DAFTAR PUSTAKA

Ali, O. (2013). Genetics of Type 2 Diabetes. *World Journal of Diabetes*, 114-123.

Ambrosius, W. T. (2007). *Methods in Molecular Biology : Topics in Biostatistics.* Totowa, New Jersey: Humana Press.

Banjarnahor, E., & Wangko, S. (2012). Sel Beta Pankreas, Sintesis dan Sekresi Insulin. *Jurnal Biomedik*, 156-162.

Barro, R. A., Sulvianti, I. D., & Afendi, F. M. (2013). Penerapan Synthetic Minority Oversampling Technique (SMOTE) Terhadap Data Tidak Seimbang pada Pembuatan Model Komposisi Jamu. *Xplore, Vol.1 (1), Edisi 9*, 1-6.

Blum, A., & Mitchell, T. (1998). Combining Labeled and Unlabeled Data with Co-Training. *Annual Conference on Computational Learning Theory*, (pp. 92-100).

Bolstad M, B., Irizarry, R. A., A strand, M., & Speed, T. P. (2003). A comparison of normalization methods for high density oligonucleotide array data based on variance and bias. *Bioinformatics*, 19(2): 185-193.

Cahyo, L. B. (2018). *Implementasi Support Vector Machine untuk Melakukan Klasifikasi pada Data Informatika (Studi Kasus: Klasifikasi Gene Expression yang terjangkit Medulloblastoma di Amerika).* Yogyakarta: Universitas Islam Indonesia.

Chandra, B. (2016). Gene Selection Methods for Microarray Data. In *Applied Computing in Medicine and Health* (pp. 45-78). New Delhi: Elsevier Inc.

Christianini, N., & Shawe-Taylor, J. (2013). An Introduction to Support Vector Machines and Other Kernel-based Learning Methods.

Corwin, E. J. (2009). *Buku Saku Patofisiologi, 3 Edisi Revisi.* Jakarta: EGC.

Cui, X., Kerr, M. K., & Churcill, G. A. (2003). Transformations for cDNA Microarray Data. *Statistical Applications in Genetics and Molecular Biology*, 1-20.

Dahlan, M. S. (2009). *Penelitian Diagnostik: Dasar-dasar Teoritis dan Aplikasi dengan SPSS dan Stata.* Jakarta: Salemba Medika.

B., & Rajapakse, J. C. (2005). Multiple SVM-RFE for Gene Selection in ancer Classification with Expression Data. *IEEE Transactions on anobioscience*, 228-234.

Feng, C., Wang, H., Lu, N., Chen, T., He, H., Lu, Y., & Tu, X. M. (2014). Log Transformation and Its Implication for Data Analysis. *Shanghai Archives of Psychiatry, Vol. 26, No.2*, 105-109.

Gardner, D. G., & Shoback, D. (2011). *Greenspan's Basic and Clinical Endocrinology. Ninth Edition.* . China: McGraw-Hill Medical.

Hammel, L. (2008). Model Assesment with ROC Curves. In J. Wang, *The Encyclopedia of Data Warehousing and Mining* (pp. 1316-1323). Hershey: Information Science Reference.

Harrington, C. A., Rosenow, C., & Retief, J. (2000). Monitoring Gene Expression Using DNA Microarrays. *Current Opinion in Microbiology*, 285-291.

Hasri, N. N., Wen, N. H., Howe, C. W., Mohamad, M. S., Deris, S., & Kasim, S. (2017). Improved Support Vector Machine Using Multiple SVM-RFE for Cancer Classification. *International Journal on Advanced Science Engineering Information Technology*, 1589-1593.

Indriani, A. (2014). Klasifikasi Data Forum dengan menggunakan Metode Naïve Bayes Classifier. *Seminar Nasional Aplikasi Teknologi Informasi*, 5-10.

Kohavi, R. (1995). A Study of Cross-Validation and Bootstrap for Accuracy Estimation and Model Selection. *Appears in the International Joint Conference on Articial Intelligence*, 1137-1145.

McDonald, J. M. (2008). *Handbook of Biological Statistics.* Baltimore: Sparky House Publishing.

Mohammed, M., Khan, M. B., & Bashier, E. B. (2016). *Machine Learning : Algorithms and Applications.* Boca Raton, Florida: CRC Press.

Morvan, M. L., & Vert, J.-P. (2017). Supervised Quantile Normalisation. *Stat. ML*, 1-13.

Munawarah, R., Soesanto, O., & Faisal, M. R. (2016). Penerapan Metode Support Vector Machine pada Diagnosa Hepatitis. *Kumpulan Jurnal Ilmu Komputer (KLIK), Volume 04, No.01 Februari*, 103-113.

Ningrum, H. C. (2018). *Perbandingan Metode Support Vector Machine (SVM) Linear, Radial Basis Function (RBF), dan Polinomial Kernel dalam Klasifikasi Bidang Studi Lanjut Pilihan Alumni UII.* Yogyakarta: niversitas Islam Indonesia.

A. S., Witarto, A. B., & Handoko, D. (2003). *Support Vector Machine : eori dan Aplikasinya dalam Bioinformatika.* IlmuKomputer.com.

Purba, D. (2009). *Perbandingan Kadar C-Peptide Pada Diabetes Melitus Tipe 2 yang Baru Didiagnosa Dengan Non Diabetes Melitus.* Meda: Universitas Sumatera Utara.

Putra, I. W., Sutarga, I. M., Kardiwinata, M. P., Suariyani, N. L., Septarini, N. W., & Subrata, I. M. (2016). *Modul Penelitian Uji Diagnostik dan Skrining.* Denpasar: Program Studi Kesehatan Masyarakat Fakultas Kedokteran Universitas Udayana.

Ramadhina, A. (2011). *Klasifikasi Dokumen BahasaIndonesia menggunakan Metode Semantic Smoothing.* Bogor: Institut Pertanian Bogor.

Rubenstein, D., Wayne, D., & Bradley, J. (2007). *Kedokteran Klinis. Edisi Keenam.* Jakarta: Erlangga.

Santoso, B. (1993). *Buku Pegangan Kuliah: Ilmu Penyakit Dalam I Seri Penyakit Endokrin dan Metabolik.* Surakarta: Universitas Negeri Sebelas Maret.

Simon, R. M., McShane, L. M., Wright, G. W., Korn, E. L., Radmacher, M. D., & Zhao, Y. (2003). *Design and Analysis of DNA Microarray Investigations.* New York: Springer.

Siswantoro, J. (2010). Automatic Gridding Citra Microarray dengan Menggunakan Image Thresholding. *Konferensi Nasional Matematika XV* (pp. 153-162). Manado: Indonesia Mathematical Society.

Taneera, J., Dhaiban, S., Mohammed, A. K., Mukhopadhyay, D., Aljaibeji, H., Sulaiman, N., . . . Salehi, A. (2019). GNAS Gene is an Important Regulator of Insulin Secretory Capacity in Pacreatic β-cells. *Elsevier*, 1-9.

Ulhaq, Z., & Adji, T. B. (2017). Integrasi Synthetic Minority Over-Sampling Technique dengan Correlated Naive Bayes Classifier (C-NBC) pada Klasifikasi Siswa Berkesulitan Belajar. *CITEE 2017* (pp. 201-205). Yogyakarta: Departemen Teknik Elektro dan Teknologi Informasi, FT UGM.

Vapnik, V., Golowich, S. E., & Smola, A. (1996). Support Vector Method for Function Approximation, Regression Estimation and Signal Processing. *NIPS '96 Proceeding of The 9th International Conference on Neural Information Processing Systems* (pp. 281-287). Denver: MIT Press Cambridge.

., & Huang, X. (2015). Multiple SVM-RFE for Multi-class Gene Selection ı DNA Microarray Data. *2015 International Joint Conference on Neural etworks (IJCNN)*.

LAMPIRAN

Lampiran 1 : Source Code Input Data

```
#Input Data
library(GEOquery)
datamentah<-getGEO(file="GSE18732_series_matrix.txt")
phenoData <-pData(datamentah)
cekdatamentah<-data.frame(na.omit(exprs(datamentah)))
```

Lampiran 2 : Source Code Transformasi Logaritma

```
#Preprocessing - Transformasi Logaritma
log2<-apply(na.omit(exprs(datamentah)),1,function(x){log(x,base = 2)})
str(log2)
head(log2)

#Preprocessing - Uji Normalitas Menggunakan Shapiro Wilk
f<- function(x) {
  if (diff(range(x)) == 0) list() else shapiro.test(x)
}
uji<-apply(log2,2,f)
uji<-data.frame(matrix(unlist(uji), nrow = 25770, byrow=T))
uji<-as.numeric(as.character(uji$X2))

normalisasidata<-t(log2)
featureName<-dimnames(normalisasidata)
featureName<-featureName[[2]]

featureName1<-dimnames(log2)
featureName1<-featureName1[[2]]
```

Lampiran 3 : Source Code Normalisasi Quantile

```
#Preprocessing - Normalisasi Quantile
library(preprocessCore)
normalisasidata<-normalize.quantiles(normalisasidata)          #melakukan
Normalisasi Quantile
colnames(normalisasidata)<-featureName #ubah nama kolom
ceknormalisasi<-data.frame(normalisasidata)
histogram_norm<-hist(ceknormalisasi$GSM465274)

#Preprocessing - Uji Shapiro Wilk Setelah Normalisasi
f<- function(x) {
  if (diff(range(x)) == 0) list() else shapiro.test(x)
}
uji2<-apply(normalisasidata,1, f)
uji2<-data.frame (matrix(unlist(uji2), nrow = 25770, byrow=T))
uji2<-as.numeric(as.character(uji2$X2))

#Mengganti Value Vector
library(plyr)
revalue<-revalue(phenoData$characteristics_ch1.5,          c("dmverified:
0"=as.numeric (0),"dmverified: 1"= as.numeric (1)))
     -data.frame(as.numeric(as.character(revalue)))
     value)[names(revalue)=="as.numeric.as.character.revalue.."]<-"y"
     -dimnames(cekdatamentah)
     -rowname[[2]]
     (revalue)<-rowname

     ungkan Dua Data Frame
```

```
inputdata<-data.frame(t(normalisasidata))
colnames(inputdata)<- featureName1
input<-data.frame(revalue, inputdata)
input.data<-data.frame(inputdata, y=revalue)
input
```

Lampiran 4 : Source Code Algoritma MSVM-RFE

```
svmRFE.wrap <- function(test.fold, X, ...) {
  # Wrapper to run svmRFE function while omitting a given test fold
  train.data = X[-test.fold, ]
  test.data  = X[test.fold, ]

  # Melakukan Rank Feature
  features.ranked = svmRFE(train.data, ...)

  return(list(feature.ids=features.ranked,
train.data.ids=row.names(train.data),
test.data.ids=row.names(test.data)))
}

svmRFE <- function(X, k=1, halve.above=5000) {
  # Seleksi Fitur menggunakan Multiple SVM Recursive Feature Elimination
  n = ncol(X) - 1

  # Melakukan Scaling Data di Depan sehingga tidak harus dilakukan di
setiap proses
  cat('Scaling data...')
  X[, -1] = scale(X[, -1])
  cat('Done!\n')
  flush.console()

  pb = txtProgressBar(1, n, 1, style=3)

  i.surviving = 1:n
  i.ranked    = n
  ranked.list = vector(length=n)

  # Mengulang proses rekursif di seluruh fitur
  while(length(i.surviving) > 0) {
    if(k > 1) {
      # Subsample to mendapatkan bobot vector multiple (i.e. mSVM-RFE)
      folds = rep(1:k, len=nrow(X))[sample(nrow(X))]
      folds = lapply(1:k, function(x) which(folds == x))

      # Memperoleh bobot untuk setiap training set
      w = lapply(folds, getWeights, X[, c(1, 1+i.surviving)])
      w = do.call(rbind, w)

      # Melakukan normalisasi di setiap bobot vector
      w = t(apply(w, 1, function(x) x / sqrt(sum(x^2))))

      # Menghitung ranking criteria
      v    = w * w
      vbar = apply(v, 2, mean)
      vsd  = apply(v, 2, sd)
      c    = vbar / vsd
    } else {
      # Hanya melakukan satu kali pass (i.e. regular SVM-RFE)
      w = getWeights(NULL, X[, c(1, 1+i.surviving)])
      c = w * w
    }

    # Melakukan ranking pada fitur
    ranking = sort(c, index.return=T)$ix
    if(length(i.surviving) == 1) {
        nking = 1


        ength(i.surviving) > halve.above) {
        Memotong fitur setengah hingga kurang dari halve.above
        eat = length(i.surviving)
        ut  = round(nfeat / 2)
```

```
      n     = nfeat - ncut

      cat('Features halved from', nfeat, 'to', n, '\n')
      flush.console()

      pb = txtProgressBar(1, n, 1, style=3)

    } else ncut = 1

    # Update feature list
    ranked.list[i.ranked:(i.ranked-ncut+1)]                                    =
i.surviving[ranking[1:ncut]]
    i.ranked    = i.ranked - ncut
    i.surviving = i.surviving[-ranking[1:ncut]]

    setTxtProgressBar(pb, n-length(i.surviving))
    flush.console()
  }

  close(pb)

  return (ranked.list)
}

getWeights <- function(test.fold, X) {
  # Fit a linear SVM model and obtain feature weights
  train.data = X
  if(!is.null(test.fold)) train.data = X[-test.fold, ]

  svmModel    =    svm(train.data[,    -1],    train.data[,    1],    cost=10,
cachesize=500,
                  scale=F, type="C-classification", kernel="linear")

  t(svmModel$coefs) %*% svmModel$SV
}
```

Lampiran 5 : Source Code Implementasi MSVM-RFE

```
#SVM-RFE
library(e1071)
source("msvmRFE.R")
svmRFE_result <- svmRFE(input, k=1, halve.above=100)

#Membuat 10 Cross Validation
nfold = 10
nrows = nrow(input)
folds = rep(1:nfold, len=nrows, byrow=T)[sample(nrows)]
folds

folds = lapply(1:nfold, function(x) which (folds==x))
folds

#Perform Feature Ranking to All-Folds
results = lapply(folds, svmRFE.wrap, input, k=10, halve.above=100)
length(results)
results
```

Lampiran 6 : Source Code Write Features untuk Mengurutkan Gen Sesuai Average Ranknya

```
WriteFeatures <- function(results, X, save=T, file='features_ranked.txt')
{
        ile feature rankings di seluruh multiple folds
        eID  =  sort(apply(sapply(results,  function(x)  sort(x$feature,
        turn=T)$ix), 1, mean), index=T)$ix
        nk    =  sort(apply(sapply(results,  function(x)  sort(x$feature,
        turn=T)$ix), 1, mean), index=T)$x
        e.name = colnames(X[, -1])[featureID]
```

```
  features.ranked          =          data.frame(FeatureName=feature.name,
FeatureID=featureID, AvgRank=avg.rank)
  if(save==T) {
    write.table(features.ranked, file=file, quote=F, row.names=F)
  } else {
    features.ranked
  }
}

#Mendapatkan Fitur Teratas Dari Setiap Folds
top.features=WriteFeatures(results, input, save=F)
head(top.features)
```

Lampiran 7 : Source Code Untuk Menginterpretasikan Gen Teratas ke dalam HGNC Symbol

```
#Nama Gene
library(biomaRt)
library(org.Hs.eg.db)
library(ensembldb)
mart<-useDataset("hsapiens_gene_ensembl", useMart("ensembl"))
genes1<-top.features$FeatureName
genes<-data.frame(sub("_at","",genes1))
names(genes)[names(genes)=="sub._at.......genes1."]<-"gene"
df<-data.frame(genes,top.features[,-4])
df$FeatureName<-NULL
df<-df[c("FeatureID", "AvgRank", "gene")]
G_list<-getBM(filters    =    "ensembl_transcript_id",    attributes    =
c("ensembl_transcript_id", "hgnc_symbol"), values = genes, mart = mart)
gene_name<-    data.frame(merge(df,G_list,by.x    =    "gene",    by.y    =
"ensembl_transcript_id"))
gene_name<-gene_name[order(gene_name$AvgRank),]
```

Lampiran 8 : Source Code Dataset Splitting ke dalam Skema 70 30 dan 80 20

```
#Dataset Splitting
table(input.data$y)
input.data$y<-factor(input.data$y, levels=c(0,1))
'%ni%' <-Negate('%in%')
options(scipen = 999)

#Preparation Training and Test data
#Skema 70 dan 30
library(caret)
set.seed(100)
trainDataIndex<-createDataPartition(input.data$y, p=0.7, list=F)
trainData<-input.data[trainDataIndex,]
testData<-input.data[-trainDataIndex,]

#Skema 80 dan 20
set.seed(100)
trainDataIndex<-createDataPartition(input.data$y, p=0.8, list=F)
trainData<-input.data[trainDataIndex,]
testData<-input.data[-trainDataIndex,]
```

n 9 : Source Code SMOTE Resampling

```
esampling
DMwR)
ainData$y)
```

```
prop.table(table(trainData$y))
smote_train<-SMOTE(y~.,data=trainData, perc.over = 100, perc.under = 200)
table(smote_train$y)
```

Lampiran 10 : Source Code Algoritma Parameter Tuning dan Generalization Error

```
FeatSweep.wrap.linear <- function(i, results, X) {
  # Wrapper untuk estimate generalization error terhadap semua folds, top
features
  svm.list = lapply(results, function(x) tune(svm,
                                              train.x      =
X[x$train.data.ids, 1+x$feature.ids[1:i]],
                                              train.y      =
X[x$train.data.ids, 1],
                                              validation.x =
X[x$test.data.ids, 1+x$feature.ids[1:i]],
                                              validation.y =
X[x$test.data.ids, 1],
                                              # Optimisasi Hyperparameter
SVM
                                              ranges       = tune(svm,
                                                                  train.x
= X[x$train.data.ids, 1+x$feature.ids[1:i]],
                                                                  train.y
= X[x$train.data.ids, 1],
                                                                  ranges
= list(cost=2^(-6:6)))$best.par,
                                              tunecontrol  =
tune.control(sampling='fix'))$perf)
  error = mean(sapply(svm.list, function(x) x$error))
  return(list(svm.list=svm.list, error=error))
}

FeatSweep.wrap.RBF <- function(i, results, X) {
  # Wrapper untuk estimate generalization error terhadap semua folds, top
features
  svm.list = lapply(results, function(x) tune(svm,
                                              train.x      =
X[x$train.data.ids, 1+x$feature.ids[1:i]],
                                              train.y      =
X[x$train.data.ids, 1],
                                              validation.x =
X[x$test.data.ids, 1+x$feature.ids[1:i]],
                                              validation.y =
X[x$test.data.ids, 1],
                                              # Optimisasi Hyperparameter
SVM
                                              ranges       = tune(svm,
                                                                  train.x
= X[x$train.data.ids, 1+x$feature.ids[1:i]],
                                                                  train.y
= X[x$train.data.ids, 1],
                                                                  ranges
= list(gamma=2^(-12:0), cost=2^(-6:6)))$best.par,
                                              tunecontrol  =
tune.control(sampling='fix'))$perf)
  error = mean(sapply(svm.list, function(x) x$error))
  return(list(svm.list=svm.list, error=error))
}

FeatSweep.wrap.polynomial <- function(i, results, X) {
  # Wrapper untuk estimate generalization error terhadap semua folds, top
features
  svm.list = lapply(results, function(x) tune(svm,
                                              train.x      =
        n.data.ids, 1+x$feature.ids[1:i]],
                                              train.y      =
        n.data.ids, 1],
                                              validation.x =
        .data.ids, 1+x$feature.ids[1:i]],
                                              validation.y =
        .data.ids, 1],
```

```
# Optimisasi Hyperparameter SVM
                                                ranges       = tune(svm,
                                                       train.x
= X[x$train.data.ids, 1+x$feature.ids[1:i]],
                                                       train.y
= X[x$train.data.ids, 1],
                                                       ranges
= list(gamma=2^(-12:0), cost=2^(-6:6), coef0=2^(-3:3),
degree=2))$best.par,
                                                tunecontrol  =
tune.control(sampling='fix'))$perf)
  error = mean(sapply(svm.list, function(x) x$error))
  return(list(svm.list=svm.list, error=error))
}
```

Lampiran 11 : Source Code Plot Error

```
PlotErrors     <-     function(errors,     errors2=NULL,     no.info=0.5,
ylim=range(c(errors,  errors2),  na.rm=T),  xlab='Number  of  Features',
ylab='10x CV Error') {
  # Melakukan plot terhadap generalization Error dan Jumlah Fitur
  AddLine <- function(x, col='black') {
    lines(which(!is.na(errors)), na.omit(x), col=col)
    points(which.min(x), min(x, na.rm=T), col='red')
    text(which.min(x),   min(x,   na.rm=T),   paste(which.min(x),   '-',
format(min(x, na.rm=T), dig=3)), pos=4, col='red', cex=0.75)
  }

  plot(errors, type='n', ylim=ylim, xlab=xlab, ylab=ylab)
  AddLine(errors)
  if(!is.null(errors2)) AddLine(errors2, 'gray30')
  abline(h=no.info, lty=3)
}
```

Lampiran 12 : Source Code Implementasi Parameter Tuning dan Generalization Error

```
#Fold untuk tuning
nfold.smote=10
nrows.smote = nrow(smote_train)
folds.smote         =         rep(1:nfold.smote,         len=nrows.smote,
byrow=T)[sample(nrows.smote)]
folds.smote

folds.smote = lapply(1:nfold.smote, function(x) which (folds.smote==x))
folds.smote

#Rearrange results untuk tuning
source("E:/Skripsi/DMT2/klasifikasi.R")

results.wrap <- function(tuning.fold, X, result, ...) {
  # Wrapper to run svmRFE function while omitting a given test fold
  train.data.tuning = X[-tuning.fold, ]
  test.data.tuning  = X[tuning.fold, ]

  # Melakukan Rank Feature
  for (i in 1:10) {
    feature.ranked.list= results[[i]]$feature.ids
    return(list(feature.ids=feature.ranked.list,
train.data.ids=row.names(train.data.tuning),
test.data.ids=row.names(test.data.tuning)))

rap    =    lapply(folds.smote,    results.wrap,    smote_train,
top.features, k=10)
rap
```

```
#Tuning Parameter untuk mendapatkan Generalization Error dan Best Parameter
SVM Kernel Linear
data_tune<-data.frame(y=smote_train[,ncol(smote_train)],smote_train[,-
ncol(smote_train)])
data_tune$y<-as.numeric(as.character(data_tune$y))

featsweep.linear = lapply(1:5, FeatSweep.wrap.linear, tuning.wrap,
data_tune)
featsweep.linear
no.info = min(prop.table((table(data_tune[,1]))))
errors = sapply(featsweep.linear, function(x) ifelse(is.null(x), NA,
x$error))

dev.new (width=5, height =5 , bg='white')
PlotErrors(errors, no.info = no.info)
dev.off()

#Tuning untuk mendapatkan Generalization Error dan Best Parameter SVM
Kernel RBF
featsweep.RBF = lapply(1:5, FeatSweep.wrap.RBF, tuning.wrap, data_tune)
featsweep.RBF
no.info = min(prop.table((table(data_tune[,1]))))
errors = sapply(featsweep.RBF, function(x) ifelse(is.null(x), NA,
x$error))

dev.new (width=5, height =5 , bg='white')
PlotErrors(errors, no.info = no.info)
dev.off()

#Tuning Parameter untuk mendapatkan Generalization Error dan Best Parameter
SVM Kernel Polynomial Orde 2
featsweep.polynomial = lapply(1:5, FeatSweep.wrap.polynomial, results,
input)
featsweep.polynomial
no.info = min(prop.table((table(input[,1]))))
errors = sapply(featsweep.polynomial, function(x) ifelse(is.null(x), NA,
x$error))

dev.new (width=5, height =5 , bg='white')
PlotErrors(errors, no.info = no.info)
dev.off()
```

Lampiran 13 : Source Code Penghitungan Akurasi, Confusion Matrix, AUC, dan Plot ROC

```
prediction<-function(svmmod, test.data){
  pred<-predict(svmmod, newdata = test.data, type="response")
  y_act<-test.data$y
  y_act

  #Menghitung Akurasi
  cat("Akurasi : ")
  print(mean(pred==y_act))
  print(caret::confusionMatrix(pred,y_act,positive="1"))

  #Plotting ROC
  pred<-as.numeric(as.character(pred))
  cat("Nilai AUC :")
  cat(InformationValue::AUROC(y_act,pred))
      (InformationValue::plotROC(y_act,pred))
```

Lampiran 14 : Source Code Implementasi Klasifikasi SVM menggunakan 3 Buah Kernel dan Evaluasi Performa untuk Dataset 1 (70 30)

```
#SVMModel
#Skema 70 Training Set, 30 Test Set
#Dataset 1 : Kernel Linear
#Dataset 1 : Kernel Linear dengan 10 Fitur Teratas
svmmod_10_linear_with_SMOTE<-
svm(y~ENST00000371095_at+ENST00000371085_at+ENST00000371082_at+ENST000003
49036_at+ENST00000265620_at+

ENST00000263080_at+ENST00000313949_at+ENST00000338783_at+ENST00000369420_
at+ENST00000369417_at,
               data=smote_train,         kernel="linear",         cost=8,
types="response")
svmmod_10_linear_no_SMOTE<-
svm(y~ENST00000371095_at+ENST00000371085_at+ENST00000371082_at+ENST000003
49036_at+ENST00000265620_at+

ENST00000263080_at+ENST00000313949_at+ENST00000338783_at+ENST00000369420_
at+ENST00000369417_at,
                              data=trainData,  kernel="linear",  cost=8,
types="response")
prediksi_with_SMOTE_linear10<-
prediction(svmmod_10_linear_with_SMOTE,testData)
prediksi_no_SMOTE_linear10<-
prediction(svmmod_10_linear_no_SMOTE,testData)

#Dataset 1 : Kernel Linear dengan 20 Fitur Teratas
svmmod_20_linear_with_SMOTE<-
svm(y~ENST00000371095_at+ENST00000371085_at+ENST00000371082_at+ENST000003
49036_at+ENST00000265620_at+

ENST00000263080_at+ENST00000313949_at+ENST00000338783_at+ENST00000369420_
at+ENST00000369417_at+

ENST00000306082_at+ENST00000381463_at+ENST00000361035_at+ENST00000371102_
at+ENST00000371101_at+

ENST00000274311_at+ENST00000325945_at+ENST00000371100_at+ENST00000371075_
at+ENST00000354359_at,
               data=smote_train,         kernel="linear",         cost=8,
types="response")
svmmod_20_linear_no_SMOTE<-
svm(y~ENST00000371095_at+ENST00000371085_at+ENST00000371082_at+ENST000003
49036_at+ENST00000265620_at+

ENST00000263080_at+ENST00000313949_at+ENST00000338783_at+ENST00000369420_
at+ENST00000369417_at+

ENST00000306082_at+ENST00000381463_at+ENST00000361035_at+ENST00000371102_
at+ENST00000371101_at+

ENST00000274311_at+ENST00000325945_at+ENST00000371100_at+ENST00000371075_
at+ENST00000354359_at,
                     data=trainData,       kernel="linear",       cost=8,
types="response")
      _with_SMOTE_linear20<-
      on(svmmod_20_linear_with_SMOTE,testData)
      _no_SMOTE_linear20<-
      on(svmmod_20_linear_no_SMOTE,testData)

       1 : Kernel linear dengan 30 Fitur Teratas
```

```
svmmod_30_linear_with_SMOTE<-
svm(y~ENST00000371095_at+ENST00000371085_at+ENST00000371082_at+ENST000003
49036_at+ENST00000265620_at+

ENST00000263080_at+ENST00000313949_at+ENST00000338783_at+ENST00000369420_
at+ENST00000369417_at+

ENST00000306082_at+ENST00000381463_at+ENST00000361035_at+ENST00000371102_
at+ENST00000371101_at+

ENST00000274311_at+ENST00000325945_at+ENST00000371100_at+ENST00000371075_
at+ENST00000354359_at+

ENST00000306120_at+ENST00000306090_at+ENST00000375594_at+ENST00000360868_
at+ENST00000358855_at+

ENST00000381936_at+ENST00000359894_at+ENST00000300947_at+ENST00000390597_
at+ENST00000380991_at,
               data=smote_train,         kernel="linear",         cost=8,
types="response")
svmmod_30_linear_no_SMOTE<-
svm(y~ENST00000371095_at+ENST00000371085_at+ENST00000371082_at+ENST000003
49036_at+ENST00000265620_at+

ENST00000263080_at+ENST00000313949_at+ENST00000338783_at+ENST00000369420_
at+ENST00000369417_at+

ENST00000306082_at+ENST00000381463_at+ENST00000361035_at+ENST00000371102_
at+ENST00000371101_at+

ENST00000274311_at+ENST00000325945_at+ENST00000371100_at+ENST00000371075_
at+ENST00000354359_at+

ENST00000306120_at+ENST00000306090_at+ENST00000375594_at+ENST00000360868_
at+ENST00000358855_at+

ENST00000381936_at+ENST00000359894_at+ENST00000300947_at+ENST00000390597_
at+ENST00000380991_at,
                                  data=trainData, kernel="linear", cost=8,
types="response")
prediksi_with_SMOTE_linear30<-
prediction(svmmod_30_linear_with_SMOTE,testData)
prediksi_no_SMOTE_linear30<-
prediction(svmmod_30_linear_no_SMOTE,testData)

#Dataset 1 : Kernel Linear dengan 40 Fitur Teratas
svmmod_40_linear_with_SMOTE<-
svm(y~ENST00000371095_at+ENST00000371085_at+ENST00000371082_at+ENST000003
49036_at+ENST00000265620_at+

ENST00000263080_at+ENST00000313949_at+ENST00000338783_at+ENST00000369420_
at+ENST00000369417_at+

ENST00000306082_at+ENST00000381463_at+ENST00000361035_at+ENST00000371102_
at+ENST00000371101_at+

ENST00000274311_at+ENST00000325945_at+ENST00000371100_at+ENST00000371075_
at+ENST00000354359_at+

)306120_at+ENST00000306090_at+ENST00000375594_at+ENST00000360868_
)000358855_at+

)381936_at+ENST00000359894_at+ENST00000300947_at+ENST00000390597_
)000380991_at+
```

```
ENST00000380986_at+ENST00000265028_at+ENST00000359623_at+ENST00000323078_
at+ENST00000283646_at+

ENST00000180173_at+ENST00000244296_at+ENST00000221293_at+ENST00000381251_
at+ENST00000304101_at,
               data=smote_train,                          kernel="radial",
cost=8,types="response")
svmmod_40_linear_no_SMOTE<-
svm(y~ENST00000371095_at+ENST00000371085_at+ENST00000371082_at+ENST000003
49036_at+ENST00000265620_at+

ENST00000263080_at+ENST00000313949_at+ENST00000338783_at+ENST00000369420_
at+ENST00000369417_at+

ENST00000306082_at+ENST00000381463_at+ENST00000361035_at+ENST00000371102_
at+ENST00000371101_at+

ENST00000274311_at+ENST00000325945_at+ENST00000371100_at+ENST00000371075_
at+ENST00000354359_at+

ENST00000306120_at+ENST00000306090_at+ENST00000375594_at+ENST00000360868_
at+ENST00000358855_at+

ENST00000381936_at+ENST00000359894_at+ENST00000300947_at+ENST00000390597_
at+ENST00000380991_at+

ENST00000380986_at+ENST00000265028_at+ENST00000359623_at+ENST00000323078_
at+ENST00000283646_at+

ENST00000180173_at+ENST00000244296_at+ENST00000221293_at+ENST00000381251_
at+ENST00000304101_at,
                              data=trainData,          kernel="radial",
cost=8,types="response")
prediksi_with_SMOTE_linear40<-
prediction(svmmod_40_linear_with_SMOTE,testData)
prediksi_no_SMOTE_linear40<-
prediction(svmmod_40_linear_no_SMOTE,testData)

#Dataset 1 : Kernel Linear dengan 50 Fitur Teratas
svmmod_50_linear_with_SMOTE<-
svm(y~ENST00000371095_at+ENST00000371085_at+ENST00000371082_at+ENST000003
49036_at+ENST00000265620_at+

ENST00000263080_at+ENST00000313949_at+ENST00000338783_at+ENST00000369420_
at+ENST00000369417_at+

ENST00000306082_at+ENST00000381463_at+ENST00000361035_at+ENST00000371102_
at+ENST00000371101_at+

ENST00000274311_at+ENST00000325945_at+ENST00000371100_at+ENST00000371075_
at+ENST00000354359_at+

ENST00000306120_at+ENST00000306090_at+ENST00000375594_at+ENST00000360868_
at+ENST00000358855_at+

ENST00000381936_at+ENST00000359894_at+ENST00000300947_at+ENST00000390597_
at+ENST00000380991_at+

)380986_at+ENST00000265028_at+ENST00000359623_at+ENST00000323078_
)000283646_at+

)180173_at+ENST00000244296_at+ENST00000221293_at+ENST00000381251_
)000304101_at+
```

```
ENST00000371761_at+ENST00000217246_at+ENST00000262662_at+ENST00000356692_
at+ENST00000361625_at+

ENST00000253023_at+ENST00000367843_at+ENST00000367840_at+ENST00000367839_
at+ENST00000324705_at,
               data=smote_train,                          kernel="linear",
cost=8,types="response")
svmmod_50_linear_no_SMOTE<-
svm(y~ENST00000371095_at+ENST00000371085_at+ENST00000371082_at+ENST000003
49036_at+ENST00000265620_at+

ENST00000263080_at+ENST00000313949_at+ENST00000338783_at+ENST00000369420_
at+ENST00000369417_at+

ENST00000306082_at+ENST00000381463_at+ENST00000361035_at+ENST00000371102_
at+ENST00000371101_at+

ENST00000274311_at+ENST00000325945_at+ENST00000371100_at+ENST00000371075_
at+ENST00000354359_at+

ENST00000306120_at+ENST00000306090_at+ENST00000375594_at+ENST00000360868_
at+ENST00000358855_at+

ENST00000381936_at+ENST00000359894_at+ENST00000300947_at+ENST00000390597_
at+ENST00000380991_at+

ENST00000380986_at+ENST00000265028_at+ENST00000359623_at+ENST00000323078_
at+ENST00000283646_at+

ENST00000180173_at+ENST00000244296_at+ENST00000221293_at+ENST00000381251_
at+ENST00000304101_at+

ENST00000371761_at+ENST00000217246_at+ENST00000262662_at+ENST00000356692_
at+ENST00000361625_at+

ENST00000253023_at+ENST00000367843_at+ENST00000367840_at+ENST00000367839_
at+ENST00000324705_at,
                                  data=trainData,         kernel="linear",
cost=8,types="response")
prediksi_with_SMOTE_linear50<-
prediction(svmmod_50_linear_with_SMOTE,testData)
prediksi_no_SMOTE_linear50<-
prediction(svmmod_50_linear_no_SMOTE,testData)

#Dataset 1 : Kernel Linear dengan 60 Fitur Teratas
svmmod_60_linear_with_SMOTE<-
svm(y~ENST00000371095_at+ENST00000371085_at+ENST00000371082_at+ENST000003
49036_at+ENST00000265620_at+

ENST00000263080_at+ENST00000313949_at+ENST00000338783_at+ENST00000369420_
at+ENST00000369417_at+

ENST00000306082_at+ENST00000381463_at+ENST00000361035_at+ENST00000371102_
at+ENST00000371101_at+

ENST00000274311_at+ENST00000325945_at+ENST00000371100_at+ENST00000371075_
at+ENST00000354359_at+

)306120_at+ENST00000306090_at+ENST00000375594_at+ENST00000360868_
)000358855_at+

)381936_at+ENST00000359894_at+ENST00000300947_at+ENST00000390597_
)000380991_at+
```

```
ENST00000380986_at+ENST00000265028_at+ENST00000359623_at+ENST00000323078_
at+ENST00000283646_at+

ENST00000180173_at+ENST00000244296_at+ENST00000221293_at+ENST00000381251_
at+ENST00000304101_at+

ENST00000371761_at+ENST00000217246_at+ENST00000262662_at+ENST00000356692_
at+ENST00000361625_at+

ENST00000253023_at+ENST00000367843_at+ENST00000367840_at+ENST00000367839_
at+ENST00000324705_at+

ENST00000312263_at+ENST00000298694_at+ENST00000298693_at+ENST00000354181_
at+ENST00000361379_at+

ENST00000358712_at+ENST00000336092_at+ENST00000382438_at+ENST00000226972_
at+ENST00000312524_at,
               data=smote_train,         kernel="linear",         cost=8,
types="response")
svmmod_60_linear_no_SMOTE<-
svm(y~ENST00000371095_at+ENST00000371085_at+ENST00000371082_at+ENST000003
49036_at+ENST00000265620_at+

ENST00000263080_at+ENST00000313949_at+ENST00000338783_at+ENST00000369420_
at+ENST00000369417_at+

ENST00000306082_at+ENST00000381463_at+ENST00000361035_at+ENST00000371102_
at+ENST00000371101_at+

ENST00000274311_at+ENST00000325945_at+ENST00000371100_at+ENST00000371075_
at+ENST00000354359_at+

ENST00000306120_at+ENST00000306090_at+ENST00000375594_at+ENST00000360868_
at+ENST00000358855_at+

ENST00000381936_at+ENST00000359894_at+ENST00000300947_at+ENST00000390597_
at+ENST00000380991_at+

ENST00000380986_at+ENST00000265028_at+ENST00000359623_at+ENST00000323078_
at+ENST00000283646_at+

ENST00000180173_at+ENST00000244296_at+ENST00000221293_at+ENST00000381251_
at+ENST00000304101_at+

ENST00000371761_at+ENST00000217246_at+ENST00000262662_at+ENST00000356692_
at+ENST00000361625_at+

ENST00000253023_at+ENST00000367843_at+ENST00000367840_at+ENST00000367839_
at+ENST00000324705_at+

ENST00000312263_at+ENST00000298694_at+ENST00000298693_at+ENST00000354181_
at+ENST00000361379_at+

ENST00000358712_at+ENST00000336092_at+ENST00000382438_at+ENST00000226972_
at+ENST00000312524_at,
                                  data=trainData, kernel="linear", cost=8,
types="response")
        _with_SMOTE_linear60<-
        on(svmmod_60_linear_with_SMOTE,testData)
        _no_SMOTE_linear60<-
        on(svmmod_60_linear_no_SMOTE,testData)
```

1 : Kernel Linear dengan 70 Fitur Teratas

```
svmmod_70_linear_with_SMOTE<-
svm(y~ENST00000371095_at+ENST00000371085_at+ENST00000371082_at+ENST000003
49036_at+ENST00000265620_at+

ENST00000263080_at+ENST00000313949_at+ENST00000338783_at+ENST00000369420_
at+ENST00000369417_at+

ENST00000306082_at+ENST00000381463_at+ENST00000361035_at+ENST00000371102_
at+ENST00000371101_at+

ENST00000274311_at+ENST00000325945_at+ENST00000371100_at+ENST00000371075_
at+ENST00000354359_at+

ENST00000306120_at+ENST00000306090_at+ENST00000375594_at+ENST00000360868_
at+ENST00000358855_at+

ENST00000381936_at+ENST00000359894_at+ENST00000300947_at+ENST00000390597_
at+ENST00000380991_at+

ENST00000380986_at+ENST00000265028_at+ENST00000359623_at+ENST00000323078_
at+ENST00000283646_at+

ENST00000180173_at+ENST00000244296_at+ENST00000221293_at+ENST00000381251_
at+ENST00000304101_at+

ENST00000371761_at+ENST00000217246_at+ENST00000262662_at+ENST00000356692_
at+ENST00000361625_at+

ENST00000253023_at+ENST00000367843_at+ENST00000367840_at+ENST00000367839_
at+ENST00000324705_at+

ENST00000312263_at+ENST00000298694_at+ENST00000298693_at+ENST00000354181_
at+ENST00000361379_at+

ENST00000358712_at+ENST00000336092_at+ENST00000382438_at+ENST00000226972_
at+ENST00000312524_at+

ENST00000361272_at+ENST00000361231_at+ENST00000298047_at+ENST00000342745_
at+ENST00000244926_at+

ENST00000356505_at+ENST00000323760_at+ENST00000219168_at+ENST00000371534_
at+ENST00000383362_at,
               data=smote_train,     kernel="linear",     cost=8,types="C-
Classification")
svmmod_70_linear_no_SMOTE<-
svm(y~ENST00000371095_at+ENST00000371085_at+ENST00000371082_at+ENST000003
49036_at+ENST00000265620_at+

ENST00000263080_at+ENST00000313949_at+ENST00000338783_at+ENST00000369420_
at+ENST00000369417_at+

ENST00000306082_at+ENST00000381463_at+ENST00000361035_at+ENST00000371102_
at+ENST00000371101_at+

ENST00000274311_at+ENST00000325945_at+ENST00000371100_at+ENST00000371075_
at+ENST00000354359_at+

ENST00000306120_at+ENST00000306090_at+ENST00000375594_at+ENST00000360868_
          0000358855_at+

          0381936_at+ENST00000359894_at+ENST00000300947_at+ENST00000390597_
          0000380991_at+
```

```
ENST00000380986_at+ENST00000265028_at+ENST00000359623_at+ENST00000323078_
at+ENST00000283646_at+

ENST00000180173_at+ENST00000244296_at+ENST00000221293_at+ENST00000381251_
at+ENST00000304101_at+

ENST00000371761_at+ENST00000217246_at+ENST00000262662_at+ENST00000356692_
at+ENST00000361625_at+

ENST00000253023_at+ENST00000367843_at+ENST00000367840_at+ENST00000367839_
at+ENST00000324705_at+

ENST00000312263_at+ENST00000298694_at+ENST00000298693_at+ENST00000354181_
at+ENST00000361379_at+

ENST00000358712_at+ENST00000336092_at+ENST00000382438_at+ENST00000226972_
at+ENST00000312524_at+

ENST00000361272_at+ENST00000361231_at+ENST00000298047_at+ENST00000342745_
at+ENST00000244926_at+

ENST00000356505_at+ENST00000323760_at+ENST00000219168_at+ENST00000371534_
at+ENST00000383362_at,
                                data=trainData,         kernel="linear",
cost=8,types="C-Classification")
prediksi_with_SMOTE_linear70<-
prediction(svmmod_70_linear_with_SMOTE,testData)
prediksi_no_SMOTE_linear70<-
prediction(svmmod_70_linear_no_SMOTE,testData)

#Dataset 1 : Kernel Linear dengan 80 Fitur Teratas
svmmod_80_linear_with_SMOTE<-
svm(y~ENST00000371095_at+ENST00000371085_at+ENST00000371082_at+ENST000003
49036_at+ENST00000265620_at+

ENST00000263080_at+ENST00000313949_at+ENST00000338783_at+ENST00000369420_
at+ENST00000369417_at+

ENST00000306082_at+ENST00000381463_at+ENST00000361035_at+ENST00000371102_
at+ENST00000371101_at+

ENST00000274311_at+ENST00000325945_at+ENST00000371100_at+ENST00000371075_
at+ENST00000354359_at+

ENST00000306120_at+ENST00000306090_at+ENST00000375594_at+ENST00000360868_
at+ENST00000358855_at+

ENST00000381936_at+ENST00000359894_at+ENST00000300947_at+ENST00000390597_
at+ENST00000380991_at+

ENST00000380986_at+ENST00000265028_at+ENST00000359623_at+ENST00000323078_
at+ENST00000283646_at+

ENST00000180173_at+ENST00000244296_at+ENST00000221293_at+ENST00000381251_
at+ENST00000304101_at+

ENST00000371761_at+ENST00000217246_at+ENST00000262662_at+ENST00000356692_
          )000361625_at+

          )253023_at+ENST00000367843_at+ENST00000367840_at+ENST00000367839_
          )000324705_at+
```

```
ENST00000312263_at+ENST00000298694_at+ENST00000298693_at+ENST00000354181_
at+ENST00000361379_at+

ENST00000358712_at+ENST00000336092_at+ENST00000382438_at+ENST00000226972_
at+ENST00000312524_at+

ENST00000372264_at+ENST00000362031_at+ENST00000355110_at+ENST00000332302_
at+ENST00000265395_at+

ENST00000376419_at+ENST00000376417_at+ENST00000277124_at+ENST00000381753_
at+ENST00000313860_at+

ENST00000383367_at+ENST00000318809_at+ENST00000308370_at+ENST00000383179_
at+ENST00000243562_at+

ENST00000252576_at+ENST00000381917_at+ENST00000375493_at+ENST00000204005_
at+ENST00000301788_at,
               data=smote_train,         kernel="linear",         cost=8,
types="response")
svmmod_80_linear_no_SMOTE<-
svm(y~ENST00000371095_at+ENST00000371085_at+ENST00000371082_at+ENST000003
49036_at+ENST00000265620_at+

ENST00000263080_at+ENST00000313949_at+ENST00000338783_at+ENST00000369420_
at+ENST00000369417_at+

ENST00000306082_at+ENST00000381463_at+ENST00000361035_at+ENST00000371102_
at+ENST00000371101_at+

ENST00000274311_at+ENST00000325945_at+ENST00000371100_at+ENST00000371075_
at+ENST00000354359_at+

ENST00000306120_at+ENST00000306090_at+ENST00000375594_at+ENST00000360868_
at+ENST00000358855_at+

ENST00000381936_at+ENST00000359894_at+ENST00000300947_at+ENST00000390597_
at+ENST00000380991_at+

ENST00000380986_at+ENST00000265028_at+ENST00000359623_at+ENST00000323078_
at+ENST00000283646_at+

ENST00000180173_at+ENST00000244296_at+ENST00000221293_at+ENST00000381251_
at+ENST00000304101_at+

ENST00000371761_at+ENST00000217246_at+ENST00000262662_at+ENST00000356692_
at+ENST00000361625_at+

ENST00000253023_at+ENST00000367843_at+ENST00000367840_at+ENST00000367839_
at+ENST00000324705_at+

ENST00000312263_at+ENST00000298694_at+ENST00000298693_at+ENST00000354181_
at+ENST00000361379_at+

ENST00000358712_at+ENST00000336092_at+ENST00000382438_at+ENST00000226972_
at+ENST00000312524_at+

ENST00000372264_at+ENST00000362031_at+ENST00000355110_at+ENST00000332302_
          )000265395_at+

          )376419_at+ENST00000376417_at+ENST00000277124_at+ENST00000381753_
          )000313860_at+
```

```
ENST00000383367_at+ENST00000318809_at+ENST00000308370_at+ENST00000383179_
at+ENST00000243562_at+

ENST00000252576_at+ENST00000381917_at+ENST00000375493_at+ENST00000204005_
at+ENST00000301788_at,
                                    data=trainData, kernel="linear", cost=8,
types="response")
prediksi_with_SMOTE_linear80<-
prediction(svmmod_80_linear_with_SMOTE,testData)
prediksi_no_SMOTE_linear80<-
prediction(svmmod_80_linear_no_SMOTE,testData)

#Dataset 1 : Kernel Linear dengan 90 Fitur Teratas
svmmod_90_linear_with_SMOTE<-
svm(y~ENST00000371095_at+ENST00000371085_at+ENST00000371082_at+ENST000003
49036_at+ENST00000265620_at+

ENST00000263080_at+ENST00000313949_at+ENST00000338783_at+ENST00000369420_
at+ENST00000369417_at+

ENST00000306082_at+ENST00000381463_at+ENST00000361035_at+ENST00000371102_
at+ENST00000371101_at+

ENST00000274311_at+ENST00000325945_at+ENST00000371100_at+ENST00000371075_
at+ENST00000354359_at+

ENST00000306120_at+ENST00000306090_at+ENST00000375594_at+ENST00000360868_
at+ENST00000358855_at+

ENST00000381936_at+ENST00000359894_at+ENST00000300947_at+ENST00000390597_
at+ENST00000380991_at+

ENST00000380986_at+ENST00000265028_at+ENST00000359623_at+ENST00000323078_
at+ENST00000283646_at+

ENST00000180173_at+ENST00000244296_at+ENST00000221293_at+ENST00000381251_
at+ENST00000304101_at+

ENST00000371761_at+ENST00000217246_at+ENST00000262662_at+ENST00000356692_
at+ENST00000361625_at+

ENST00000253023_at+ENST00000367843_at+ENST00000367840_at+ENST00000367839_
at+ENST00000324705_at+

ENST00000312263_at+ENST00000298694_at+ENST00000298693_at+ENST00000354181_
at+ENST00000361379_at+

ENST00000358712_at+ENST00000336092_at+ENST00000382438_at+ENST00000226972_
at+ENST00000312524_at+

ENST00000372264_at+ENST00000362031_at+ENST00000355110_at+ENST00000332302_
at+ENST00000265395_at+

ENST00000376419_at+ENST00000376417_at+ENST00000277124_at+ENST00000381753_
at+ENST00000313860_at+

ENST00000383367_at+ENST00000318809_at+ENST00000308370_at+ENST00000383179_
          )000243562_at+

          )252576_at+ENST00000381917_at+ENST00000375493_at+ENST00000204005_
          )000301788_at+
```

```
ENST00000361272_at+ENST00000361231_at+ENST00000298047_at+ENST00000342745_
at+ENST00000244926_at+

ENST00000356505_at+ENST00000323760_at+ENST00000219168_at+ENST00000371534_
at+ENST00000383362_at,
               data=smote_train,         kernel="linear",         cost=8,
types="response")
svmmod_90_linear_no_SMOTE<-
svm(y~ENST00000371095_at+ENST00000371085_at+ENST00000371082_at+ENST000003
49036_at+ENST00000265620_at+

ENST00000263080_at+ENST00000313949_at+ENST00000338783_at+ENST00000369420_
at+ENST00000369417_at+

ENST00000306082_at+ENST00000381463_at+ENST00000361035_at+ENST00000371102_
at+ENST00000371101_at+

ENST00000274311_at+ENST00000325945_at+ENST00000371100_at+ENST00000371075_
at+ENST00000354359_at+

ENST00000306120_at+ENST00000306090_at+ENST00000375594_at+ENST00000360868_
at+ENST00000358855_at+

ENST00000381936_at+ENST00000359894_at+ENST00000300947_at+ENST00000390597_
at+ENST00000380991_at+

ENST00000380986_at+ENST00000265028_at+ENST00000359623_at+ENST00000323078_
at+ENST00000283646_at+

ENST00000180173_at+ENST00000244296_at+ENST00000221293_at+ENST00000381251_
at+ENST00000304101_at+

ENST00000371761_at+ENST00000217246_at+ENST00000262662_at+ENST00000356692_
at+ENST00000361625_at+

ENST00000253023_at+ENST00000367843_at+ENST00000367840_at+ENST00000367839_
at+ENST00000324705_at+

ENST00000312263_at+ENST00000298694_at+ENST00000298693_at+ENST00000354181_
at+ENST00000361379_at+

ENST00000358712_at+ENST00000336092_at+ENST00000382438_at+ENST00000226972_
at+ENST00000312524_at+

ENST00000372264_at+ENST00000362031_at+ENST00000355110_at+ENST00000332302_
at+ENST00000265395_at+

ENST00000376419_at+ENST00000376417_at+ENST00000277124_at+ENST00000381753_
at+ENST00000313860_at+

ENST00000383367_at+ENST00000318809_at+ENST00000308370_at+ENST00000383179_
at+ENST00000243562_at+

ENST00000252576_at+ENST00000381917_at+ENST00000375493_at+ENST00000204005_
at+ENST00000301788_at+

ENST00000361272_at+ENST00000361231_at+ENST00000298047_at+ENST00000342745_
      )000244926_at+

      )356505_at+ENST00000323760_at+ENST00000219168_at+ENST00000371534_
      )000383362_at,
                                  data=trainData, kernel="linear", cost=8,
      esponse")
```

```
prediksi_with_SMOTE_linear90<-
prediction(svmmod_90_linear_with_SMOTE,testData)
prediksi_no_SMOTE_linear90<-
prediction(svmmod_90_linear_no_SMOTE,testData)

#Dataset 1 : Kernel Linear dengan 100 Fitur Teratas
svmmod_100_linear_with_SMOTE<-
svm(y~ENST00000371095_at+ENST00000371085_at+ENST00000371082_at+ENST000003
49036_at+ENST00000265620_at+

ENST00000263080_at+ENST00000313949_at+ENST00000338783_at+ENST00000369420_
at+ENST00000369417_at+

ENST00000306082_at+ENST00000381463_at+ENST00000361035_at+ENST00000371102_
at+ENST00000371101_at+

ENST00000274311_at+ENST00000325945_at+ENST00000371100_at+ENST00000371075_
at+ENST00000354359_at+

ENST00000306120_at+ENST00000306090_at+ENST00000375594_at+ENST00000360868_
at+ENST00000358855_at+

ENST00000381936_at+ENST00000359894_at+ENST00000300947_at+ENST00000390597_
at+ENST00000380991_at+

ENST00000380986_at+ENST00000265028_at+ENST00000359623_at+ENST00000323078_
at+ENST00000283646_at+

ENST00000180173_at+ENST00000244296_at+ENST00000221293_at+ENST00000381251_
at+ENST00000304101_at+

ENST00000371761_at+ENST00000217246_at+ENST00000262662_at+ENST00000356692_
at+ENST00000361625_at+

ENST00000253023_at+ENST00000367843_at+ENST00000367840_at+ENST00000367839_
at+ENST00000324705_at+

ENST00000312263_at+ENST00000298694_at+ENST00000298693_at+ENST00000354181_
at+ENST00000361379_at+

ENST00000358712_at+ENST00000336092_at+ENST00000382438_at+ENST00000226972_
at+ENST00000312524_at+

ENST00000372264_at+ENST00000362031_at+ENST00000355110_at+ENST00000332302_
at+ENST00000265395_at+

ENST00000376419_at+ENST00000376417_at+ENST00000277124_at+ENST00000381753_
at+ENST00000313860_at+

ENST00000383367_at+ENST00000318809_at+ENST00000308370_at+ENST00000383179_
at+ENST00000243562_at+

ENST00000252576_at+ENST00000381917_at+ENST00000375493_at+ENST00000204005_
at+ENST00000301788_at+

ENST00000361272_at+ENST00000361231_at+ENST00000298047_at+ENST00000342745_
at+ENST00000244926_at+

)356505_at+ENST00000323760_at+ENST00000219168_at+ENST00000371534_
)000383362_at+

)383360_at+ENST00000383359_at+ENST00000370109_at+ENST00000383178_
)000383177_at+
```

```
ENST00000370107_at+ENST00000371646_at+ENST00000223864_at+ENST00000371554_
at+ENST00000342173_at,
                data=smote_train,                          kernel="linear",
cost=8,types="response")
svmmod_100_linear_no_SMOTE<-
svm(y~ENST00000371095_at+ENST00000371085_at+ENST00000371082_at+ENST000003
49036_at+ENST00000265620_at+

ENST00000263080_at+ENST00000313949_at+ENST00000338783_at+ENST00000369420_
at+ENST00000369417_at+

ENST00000306082_at+ENST00000381463_at+ENST00000361035_at+ENST00000371102_
at+ENST00000371101_at+

ENST00000274311_at+ENST00000325945_at+ENST00000371100_at+ENST00000371075_
at+ENST00000354359_at+

ENST00000306120_at+ENST00000306090_at+ENST00000375594_at+ENST00000360868_
at+ENST00000358855_at+

ENST00000381936_at+ENST00000359894_at+ENST00000300947_at+ENST00000390597_
at+ENST00000380991_at+

ENST00000380986_at+ENST00000265028_at+ENST00000359623_at+ENST00000323078_
at+ENST00000283646_at+

ENST00000180173_at+ENST00000244296_at+ENST00000221293_at+ENST00000381251_
at+ENST00000304101_at+

ENST00000371761_at+ENST00000217246_at+ENST00000262662_at+ENST00000356692_
at+ENST00000361625_at+

ENST00000253023_at+ENST00000367843_at+ENST00000367840_at+ENST00000367839_
at+ENST00000324705_at+

ENST00000312263_at+ENST00000298694_at+ENST00000298693_at+ENST00000354181_
at+ENST00000361379_at+

ENST00000358712_at+ENST00000336092_at+ENST00000382438_at+ENST00000226972_
at+ENST00000312524_at+

ENST00000372264_at+ENST00000362031_at+ENST00000355110_at+ENST00000332302_
at+ENST00000265395_at+

ENST00000376419_at+ENST00000376417_at+ENST00000277124_at+ENST00000381753_
at+ENST00000313860_at+

ENST00000383367_at+ENST00000318809_at+ENST00000308370_at+ENST00000383179_
at+ENST00000243562_at+

ENST00000252576_at+ENST00000381917_at+ENST00000375493_at+ENST00000204005_
at+ENST00000301788_at+

ENST00000361272_at+ENST00000361231_at+ENST00000298047_at+ENST00000342745_
at+ENST00000244926_at+

ENST00000356505_at+ENST00000323760_at+ENST00000219168_at+ENST00000371534_
          0000383362_at+

          0383360_at+ENST00000383359_at+ENST00000370109_at+ENST00000383178_
          0000383177_at+
```

```
ENST00000370107_at+ENST00000371646_at+ENST00000223864_at+ENST00000371554_
at+ENST00000342173_at,
                                data=trainData,        kernel="linear",
cost=8,types="response")
prediksi_with_SMOTE_linear100<-
prediction(svmmod_100_linear_with_SMOTE,testData)
prediksi_no_SMOTE_linear100<-
prediction(svmmod_100_linear_no_SMOTE,testData)



#Kernel RBF
#Dataset 1 : Kernel RBF dengan 10 Fitur Teratas
svmmod_10_RBF_with_SMOTE<-
svm(y~ENST00000371095_at+ENST00000371085_at+ENST00000371082_at+ENST000003
49036_at+ENST00000265620_at+

ENST00000263080_at+ENST00000313949_at+ENST00000338783_at+ENST00000369420_
at+ENST00000369417_at,
            data=smote_train,          kernel="radial",          cost=64,
gamma=0.001953125, types="response")
svmmod_10_RBF_no_SMOTE<-
svm(y~ENST00000371095_at+ENST00000371085_at+ENST00000371082_at+ENST000003
49036_at+ENST00000265620_at+

ENST00000263080_at+ENST00000313949_at+ENST00000338783_at+ENST00000369420_
at+ENST00000369417_at,
                              data=trainData,  kernel="radial",  cost=64,
gamma=0.001953125, types="response")
prediksi_with_SMOTE_RBF10<-prediction(svmmod_10_RBF_with_SMOTE,testData)
prediksi_no_SMOTE_RBF10<-prediction(svmmod_10_RBF_no_SMOTE,testData)

#Dataset 1 : Kernel RBF dengan 20 Fitur Teratas
svmmod_20_RBF_with_SMOTE<-
svm(y~ENST00000371095_at+ENST00000371085_at+ENST00000371082_at+ENST000003
49036_at+ENST00000265620_at+

ENST00000263080_at+ENST00000313949_at+ENST00000338783_at+ENST00000369420_
at+ENST00000369417_at+

ENST00000306082_at+ENST00000381463_at+ENST00000361035_at+ENST00000371102_
at+ENST00000371101_at+

ENST00000274311_at+ENST00000325945_at+ENST00000371100_at+ENST00000371075_
at+ENST00000354359_at,
               data=smote_train,         kernel="radial",         cost=64,
gamma=0.001953125, types="response")
svmmod_20_RBF_no_SMOTE<-
svm(y~ENST00000371095_at+ENST00000371085_at+ENST00000371082_at+ENST000003
49036_at+ENST00000265620_at+

ENST00000263080_at+ENST00000313949_at+ENST00000338783_at+ENST00000369420_
at+ENST00000369417_at+

ENST00000306082_at+ENST00000381463_at+ENST00000361035_at+ENST00000371102_
at+ENST00000371101_at+

ENST00000274311_at+ENST00000325945_at+ENST00000371100_at+ENST00000371075_
ENST00000354359_at,
                              data=trainData,  kernel="radial",  cost=64,
.001953125, types="response")
_with_SMOTE_RBF20<-prediction(svmmod_20_RBF_with_SMOTE,testData)
_no_SMOTE_RBF20<-prediction(svmmod_20_RBF_no_SMOTE,testData)
```

```
#Dataset 1 : Kernel RBF dengan 30 Fitur Teratas
svmmod_30_RBF_with_SMOTE<-
svm(y~ENST00000371095_at+ENST00000371085_at+ENST00000371082_at+ENST000003
49036_at+ENST00000265620_at+

ENST00000263080_at+ENST00000313949_at+ENST00000338783_at+ENST00000369420_
at+ENST00000369417_at+

ENST00000306082_at+ENST00000381463_at+ENST00000361035_at+ENST00000371102_
at+ENST00000371101_at+

ENST00000274311_at+ENST00000325945_at+ENST00000371100_at+ENST00000371075_
at+ENST00000354359_at+

ENST00000306120_at+ENST00000306090_at+ENST00000375594_at+ENST00000360868_
at+ENST00000358855_at+

ENST00000381936_at+ENST00000359894_at+ENST00000300947_at+ENST00000390597_
at+ENST00000380991_at,
                data=smote_train,         kernel="radial",         cost=64,
gamma=0.001953125, types="response")
svmmod_30_RBF_no_SMOTE<-
svm(y~ENST00000371095_at+ENST00000371085_at+ENST00000371082_at+ENST000003
49036_at+ENST00000265620_at+

ENST00000263080_at+ENST00000313949_at+ENST00000338783_at+ENST00000369420_
at+ENST00000369417_at+

ENST00000306082_at+ENST00000381463_at+ENST00000361035_at+ENST00000371102_
at+ENST00000371101_at+

ENST00000274311_at+ENST00000325945_at+ENST00000371100_at+ENST00000371075_
at+ENST00000354359_at+

ENST00000306120_at+ENST00000306090_at+ENST00000375594_at+ENST00000360868_
at+ENST00000358855_at+

ENST00000381936_at+ENST00000359894_at+ENST00000300947_at+ENST00000390597_
at+ENST00000380991_at,
                    data=trainData,        kernel="radial",        cost=64,
gamma=0.001953125, types="response")
prediksi_with_SMOTE_RBF30<-prediction(svmmod_30_RBF_with_SMOTE,testData)
prediksi_no_SMOTE_RBF30<-prediction(svmmod_30_RBF_no_SMOTE,testData)

#Dataset 1 : Kernel RBF dengan 40 Fitur Teratas
svmmod_40_RBF_with_SMOTE<-
svm(y~ENST00000371095_at+ENST00000371085_at+ENST00000371082_at+ENST000003
49036_at+ENST00000265620_at+

ENST00000263080_at+ENST00000313949_at+ENST00000338783_at+ENST00000369420_
at+ENST00000369417_at+

ENST00000306082_at+ENST00000381463_at+ENST00000361035_at+ENST00000371102_
at+ENST00000371101_at+

ENST00000274311_at+ENST00000325945_at+ENST00000371100_at+ENST00000371075_
at+ENST00000354359_at+

                )306120_at+ENST00000306090_at+ENST00000375594_at+ENST00000360868_
                )000358855_at+

                )381936_at+ENST00000359894_at+ENST00000300947_at+ENST00000390597_
                )000380991_at+
```

```
ENST00000380986_at+ENST00000265028_at+ENST00000359623_at+ENST00000323078_
at+ENST00000283646_at+

ENST00000180173_at+ENST00000244296_at+ENST00000221293_at+ENST00000381251_
at+ENST00000304101_at,
               data=smote_train,         kernel="radial",         cost=64,
gamma=0.001953125, types="response")
svmmod_40_RBF_no_SMOTE<-
svm(y~ENST00000371095_at+ENST00000371085_at+ENST00000371082_at+ENST000003
49036_at+ENST00000265620_at+

ENST00000263080_at+ENST00000313949_at+ENST00000338783_at+ENST00000369420_
at+ENST00000369417_at+

ENST00000306082_at+ENST00000381463_at+ENST00000361035_at+ENST00000371102_
at+ENST00000371101_at+

ENST00000274311_at+ENST00000325945_at+ENST00000371100_at+ENST00000371075_
at+ENST00000354359_at+

ENST00000306120_at+ENST00000306090_at+ENST00000375594_at+ENST00000360868_
at+ENST00000358855_at+

ENST00000381936_at+ENST00000359894_at+ENST00000300947_at+ENST00000390597_
at+ENST00000380991_at+

ENST00000380986_at+ENST00000265028_at+ENST00000359623_at+ENST00000323078_
at+ENST00000283646_at+

ENST00000180173_at+ENST00000244296_at+ENST00000221293_at+ENST00000381251_
at+ENST00000304101_at,
                  data=trainData,        kernel="radial",        cost=64,
gamma=0.001953125, types="response")
prediksi_with_SMOTE_RBF40<-prediction(svmmod_40_RBF_with_SMOTE,testData)
prediksi_no_SMOTE_RBF40<-prediction(svmmod_40_RBF_no_SMOTE,testData)

#Dataset 1 : Kernel RBF dengan 50 Fitur Teratas
svmmod_50_RBF_with_SMOTE<-
svm(y~ENST00000371095_at+ENST00000371085_at+ENST00000371082_at+ENST000003
49036_at+ENST00000265620_at+

ENST00000263080_at+ENST00000313949_at+ENST00000338783_at+ENST00000369420_
at+ENST00000369417_at+

ENST00000306082_at+ENST00000381463_at+ENST00000361035_at+ENST00000371102_
at+ENST00000371101_at+

ENST00000274311_at+ENST00000325945_at+ENST00000371100_at+ENST00000371075_
at+ENST00000354359_at+

ENST00000306120_at+ENST00000306090_at+ENST00000375594_at+ENST00000360868_
at+ENST00000358855_at+

ENST00000381936_at+ENST00000359894_at+ENST00000300947_at+ENST00000390597_
at+ENST00000380991_at+

ENST00000380986_at+ENST00000265028_at+ENST00000359623_at+ENST00000323078_
)000283646_at+

)0180173_at+ENST00000244296_at+ENST00000221293_at+ENST00000381251_
)000304101_at+
```

```
ENST00000371761_at+ENST00000217246_at+ENST00000262662_at+ENST00000356692_
at+ENST00000361625_at+

ENST00000253023_at+ENST00000367843_at+ENST00000367840_at+ENST00000367839_
at+ENST00000324705_at,
               data=smote_train,         kernel="radial",         cost=64,
gamma=0.001953125, types="response")
svmmod_50_RBF_no_SMOTE<-
svm(y~ENST00000371095_at+ENST00000371085_at+ENST00000371082_at+ENST000003
49036_at+ENST00000265620_at+

ENST00000263080_at+ENST00000313949_at+ENST00000338783_at+ENST00000369420_
at+ENST00000369417_at+

ENST00000306082_at+ENST00000381463_at+ENST00000361035_at+ENST00000371102_
at+ENST00000371101_at+

ENST00000274311_at+ENST00000325945_at+ENST00000371100_at+ENST00000371075_
at+ENST00000354359_at+

ENST00000306120_at+ENST00000306090_at+ENST00000375594_at+ENST00000360868_
at+ENST00000358855_at+

ENST00000381936_at+ENST00000359894_at+ENST00000300947_at+ENST00000390597_
at+ENST00000380991_at+

ENST00000380986_at+ENST00000265028_at+ENST00000359623_at+ENST00000323078_
at+ENST00000283646_at+

ENST00000180173_at+ENST00000244296_at+ENST00000221293_at+ENST00000381251_
at+ENST00000304101_at+

ENST00000371761_at+ENST00000217246_at+ENST00000262662_at+ENST00000356692_
at+ENST00000361625_at+

ENST00000253023_at+ENST00000367843_at+ENST00000367840_at+ENST00000367839_
at+ENST00000324705_at,
                   data=trainData,        kernel="radial",        cost=64,
gamma=0.001953125, types="response")
prediksi_with_SMOTE_RBF50<-prediction(svmmod_50_RBF_with_SMOTE,testData)
prediksi_no_SMOTE_RBF50<-prediction(svmmod_50_RBF_no_SMOTE,testData)

#Dataset 1 : Kernel RBF dengan 60 Fitur Teratas
svmmod_60_RBF_with_SMOTE<-
svm(y~ENST00000371095_at+ENST00000371085_at+ENST00000371082_at+ENST000003
49036_at+ENST00000265620_at+

ENST00000263080_at+ENST00000313949_at+ENST00000338783_at+ENST00000369420_
at+ENST00000369417_at+

ENST00000306082_at+ENST00000381463_at+ENST00000361035_at+ENST00000371102_
at+ENST00000371101_at+

ENST00000274311_at+ENST00000325945_at+ENST00000371100_at+ENST00000371075_
at+ENST00000354359_at+

ENST00000306120_at+ENST00000306090_at+ENST00000375594_at+ENST00000360868_
)000358855_at+

)381936_at+ENST00000359894_at+ENST00000300947_at+ENST00000390597_
)000380991_at+
```

```
ENST00000380986_at+ENST00000265028_at+ENST00000359623_at+ENST00000323078_
at+ENST00000283646_at+

ENST00000180173_at+ENST00000244296_at+ENST00000221293_at+ENST00000381251_
at+ENST00000304101_at+

ENST00000371761_at+ENST00000217246_at+ENST00000262662_at+ENST00000356692_
at+ENST00000361625_at+

ENST00000253023_at+ENST00000367843_at+ENST00000367840_at+ENST00000367839_
at+ENST00000324705_at+

ENST00000312263_at+ENST00000298694_at+ENST00000298693_at+ENST00000354181_
at+ENST00000361379_at+

ENST00000358712_at+ENST00000336092_at+ENST00000382438_at+ENST00000226972_
at+ENST00000312524_at,
               data=smote_train,         kernel="radial",         cost=64,
gamma=0.001953125, types="response")
svmmod_60_RBF_no_SMOTE<-
svm(y~ENST00000371095_at+ENST00000371085_at+ENST00000371082_at+ENST000003
49036_at+ENST00000265620_at+

ENST00000263080_at+ENST00000313949_at+ENST00000338783_at+ENST00000369420_
at+ENST00000369417_at+

ENST00000306082_at+ENST00000381463_at+ENST00000361035_at+ENST00000371102_
at+ENST00000371101_at+

ENST00000274311_at+ENST00000325945_at+ENST00000371100_at+ENST00000371075_
at+ENST00000354359_at+

ENST00000306120_at+ENST00000306090_at+ENST00000375594_at+ENST00000360868_
at+ENST00000358855_at+

ENST00000381936_at+ENST00000359894_at+ENST00000300947_at+ENST00000390597_
at+ENST00000380991_at+

ENST00000380986_at+ENST00000265028_at+ENST00000359623_at+ENST00000323078_
at+ENST00000283646_at+

ENST00000180173_at+ENST00000244296_at+ENST00000221293_at+ENST00000381251_
at+ENST00000304101_at+

ENST00000371761_at+ENST00000217246_at+ENST00000262662_at+ENST00000356692_
at+ENST00000361625_at+

ENST00000253023_at+ENST00000367843_at+ENST00000367840_at+ENST00000367839_
at+ENST00000324705_at+

ENST00000312263_at+ENST00000298694_at+ENST00000298693_at+ENST00000354181_
at+ENST00000361379_at+

ENST00000358712_at+ENST00000336092_at+ENST00000382438_at+ENST00000226972_
at+ENST00000312524_at,
                              data=trainData,  kernel="radial",  cost=64,
gamma=0.001953125, types="response")
      _with_SMOTE_RBF60<-prediction(svmmod_60_RBF_with_SMOTE,testData)
      _no_SMOTE_RBF60<-prediction(svmmod_60_RBF_no_SMOTE,testData)
```

1 : Kernel RBF dengan 70 Fitur Teratas

```
svmmod_70_RBF_with_SMOTE<-
svm(y~ENST00000371095_at+ENST00000371085_at+ENST00000371082_at+ENST000003
49036_at+ENST00000265620_at+

ENST00000263080_at+ENST00000313949_at+ENST00000338783_at+ENST00000369420_
at+ENST00000369417_at+

ENST00000306082_at+ENST00000381463_at+ENST00000361035_at+ENST00000371102_
at+ENST00000371101_at+

ENST00000274311_at+ENST00000325945_at+ENST00000371100_at+ENST00000371075_
at+ENST00000354359_at+

ENST00000306120_at+ENST00000306090_at+ENST00000375594_at+ENST00000360868_
at+ENST00000358855_at+

ENST00000381936_at+ENST00000359894_at+ENST00000300947_at+ENST00000390597_
at+ENST00000380991_at+

ENST00000380986_at+ENST00000265028_at+ENST00000359623_at+ENST00000323078_
at+ENST00000283646_at+

ENST00000180173_at+ENST00000244296_at+ENST00000221293_at+ENST00000381251_
at+ENST00000304101_at+

ENST00000371761_at+ENST00000217246_at+ENST00000262662_at+ENST00000356692_
at+ENST00000361625_at+

ENST00000253023_at+ENST00000367843_at+ENST00000367840_at+ENST00000367839_
at+ENST00000324705_at+

ENST00000312263_at+ENST00000298694_at+ENST00000298693_at+ENST00000354181_
at+ENST00000361379_at+

ENST00000358712_at+ENST00000336092_at+ENST00000382438_at+ENST00000226972_
at+ENST00000312524_at+

ENST00000361272_at+ENST00000361231_at+ENST00000298047_at+ENST00000342745_
at+ENST00000244926_at+

ENST00000356505_at+ENST00000323760_at+ENST00000219168_at+ENST00000371534_
at+ENST00000383362_at,
               data=smote_train,         kernel="radial",         cost=64,
gamma=0.001953125, types="response")
svmmod_70_RBF_no_SMOTE<-
svm(y~ENST00000371095_at+ENST00000371085_at+ENST00000371082_at+ENST000003
49036_at+ENST00000265620_at+

ENST00000263080_at+ENST00000313949_at+ENST00000338783_at+ENST00000369420_
at+ENST00000369417_at+

ENST00000306082_at+ENST00000381463_at+ENST00000361035_at+ENST00000371102_
at+ENST00000371101_at+

ENST00000274311_at+ENST00000325945_at+ENST00000371100_at+ENST00000371075_
at+ENST00000354359_at+

ENST00000306120_at+ENST00000306090_at+ENST00000375594_at+ENST00000360868_
          )000358855_at+

          )381936_at+ENST00000359894_at+ENST00000300947_at+ENST00000390597_
          )000380991_at+
```

```
ENST00000380986_at+ENST00000265028_at+ENST00000359623_at+ENST00000323078_
at+ENST00000283646_at+

ENST00000180173_at+ENST00000244296_at+ENST00000221293_at+ENST00000381251_
at+ENST00000304101_at+

ENST00000371761_at+ENST00000217246_at+ENST00000262662_at+ENST00000356692_
at+ENST00000361625_at+

ENST00000253023_at+ENST00000367843_at+ENST00000367840_at+ENST00000367839_
at+ENST00000324705_at+

ENST00000312263_at+ENST00000298694_at+ENST00000298693_at+ENST00000354181_
at+ENST00000361379_at+

ENST00000358712_at+ENST00000336092_at+ENST00000382438_at+ENST00000226972_
at+ENST00000312524_at+

ENST00000361272_at+ENST00000361231_at+ENST00000298047_at+ENST00000342745_
at+ENST00000244926_at+

ENST00000356505_at+ENST00000323760_at+ENST00000219168_at+ENST00000371534_
at+ENST00000383362_at,
                              data=trainData,  kernel="radial",  cost=64,
gamma=0.001953125, types="response")
prediksi_with_SMOTE_RBF70<-prediction(svmmod_70_RBF_with_SMOTE,testData)
prediksi_no_SMOTE_RBF70<-prediction(svmmod_70_RBF_no_SMOTE,testData)

#Dataset 1 : Kernel RBF dengan 80 Fitur Teratas
svmmod_80_RBF_with_SMOTE<-
svm(y~ENST00000371095_at+ENST00000371085_at+ENST00000371082_at+ENST000003
49036_at+ENST00000265620_at+

ENST00000263080_at+ENST00000313949_at+ENST00000338783_at+ENST00000369420_
at+ENST00000369417_at+

ENST00000306082_at+ENST00000381463_at+ENST00000361035_at+ENST00000371102_
at+ENST00000371101_at+

ENST00000274311_at+ENST00000325945_at+ENST00000371100_at+ENST00000371075_
at+ENST00000354359_at+

ENST00000306120_at+ENST00000306090_at+ENST00000375594_at+ENST00000360868_
at+ENST00000358855_at+

ENST00000381936_at+ENST00000359894_at+ENST00000300947_at+ENST00000390597_
at+ENST00000380991_at+

ENST00000380986_at+ENST00000265028_at+ENST00000359623_at+ENST00000323078_
at+ENST00000283646_at+

ENST00000180173_at+ENST00000244296_at+ENST00000221293_at+ENST00000381251_
at+ENST00000304101_at+

ENST00000371761_at+ENST00000217246_at+ENST00000262662_at+ENST00000356692_
at+ENST00000361625_at+

)0253023_at+ENST00000367843_at+ENST00000367840_at+ENST00000367839_
)0000324705_at+

)0312263_at+ENST00000298694_at+ENST00000298693_at+ENST00000354181_
)0000361379_at+
```

```
ENST00000358712_at+ENST00000336092_at+ENST00000382438_at+ENST00000226972_
at+ENST00000312524_at+

ENST00000372264_at+ENST00000362031_at+ENST00000355110_at+ENST00000332302_
at+ENST00000265395_at+

ENST00000376419_at+ENST00000376417_at+ENST00000277124_at+ENST00000381753_
at+ENST00000313860_at+

ENST00000383367_at+ENST00000318809_at+ENST00000308370_at+ENST00000383179_
at+ENST00000243562_at+

ENST00000252576_at+ENST00000381917_at+ENST00000375493_at+ENST00000204005_
at+ENST00000301788_at,
               data=smote_train,         kernel="radial",         cost=64,
gamma=0.001953125, types="response")
svmmod_80_RBF_no_SMOTE<-
svm(y~ENST00000371095_at+ENST00000371085_at+ENST00000371082_at+ENST000003
49036_at+ENST00000265620_at+

ENST00000263080_at+ENST00000313949_at+ENST00000338783_at+ENST00000369420_
at+ENST00000369417_at+

ENST00000306082_at+ENST00000381463_at+ENST00000361035_at+ENST00000371102_
at+ENST00000371101_at+

ENST00000274311_at+ENST00000325945_at+ENST00000371100_at+ENST00000371075_
at+ENST00000354359_at+

ENST00000306120_at+ENST00000306090_at+ENST00000375594_at+ENST00000360868_
at+ENST00000358855_at+

ENST00000381936_at+ENST00000359894_at+ENST00000300947_at+ENST00000390597_
at+ENST00000380991_at+

ENST00000380986_at+ENST00000265028_at+ENST00000359623_at+ENST00000323078_
at+ENST00000283646_at+

ENST00000180173_at+ENST00000244296_at+ENST00000221293_at+ENST00000381251_
at+ENST00000304101_at+

ENST00000371761_at+ENST00000217246_at+ENST00000262662_at+ENST00000356692_
at+ENST00000361625_at+

ENST00000253023_at+ENST00000367843_at+ENST00000367840_at+ENST00000367839_
at+ENST00000324705_at+

ENST00000312263_at+ENST00000298694_at+ENST00000298693_at+ENST00000354181_
at+ENST00000361379_at+

ENST00000358712_at+ENST00000336092_at+ENST00000382438_at+ENST00000226972_
at+ENST00000312524_at+

ENST00000372264_at+ENST00000362031_at+ENST00000355110_at+ENST00000332302_
at+ENST00000265395_at+

ENST00000376419_at+ENST00000376417_at+ENST00000277124_at+ENST00000381753_
          0000313860_at+

          0383367_at+ENST00000318809_at+ENST00000308370_at+ENST00000383179_
          0000243562_at+
```

```
ENST00000252576_at+ENST00000381917_at+ENST00000375493_at+ENST00000204005_
at+ENST00000301788_at,
                                data=trainData,  kernel="radial",  cost=64,
gamma=0.001953125, types="response")
prediksi_with_SMOTE_RBF80<-prediction(svmmod_80_RBF_with_SMOTE,testData)
prediksi_no_SMOTE_RBF80<-prediction(svmmod_80_RBF_no_SMOTE,testData)

#Dataset 1 : Kernel RBF dengan 90 Fitur Teratas
svmmod_90_RBF_with_SMOTE<-
svm(y~ENST00000371095_at+ENST00000371085_at+ENST00000371082_at+ENST000003
49036_at+ENST00000265620_at+

ENST00000263080_at+ENST00000313949_at+ENST00000338783_at+ENST00000369420_
at+ENST00000369417_at+

ENST00000306082_at+ENST00000381463_at+ENST00000361035_at+ENST00000371102_
at+ENST00000371101_at+

ENST00000274311_at+ENST00000325945_at+ENST00000371100_at+ENST00000371075_
at+ENST00000354359_at+

ENST00000306120_at+ENST00000306090_at+ENST00000375594_at+ENST00000360868_
at+ENST00000358855_at+

ENST00000381936_at+ENST00000359894_at+ENST00000300947_at+ENST00000390597_
at+ENST00000380991_at+

ENST00000380986_at+ENST00000265028_at+ENST00000359623_at+ENST00000323078_
at+ENST00000283646_at+

ENST00000180173_at+ENST00000244296_at+ENST00000221293_at+ENST00000381251_
at+ENST00000304101_at+

ENST00000371761_at+ENST00000217246_at+ENST00000262662_at+ENST00000356692_
at+ENST00000361625_at+

ENST00000253023_at+ENST00000367843_at+ENST00000367840_at+ENST00000367839_
at+ENST00000324705_at+

ENST00000312263_at+ENST00000298694_at+ENST00000298693_at+ENST00000354181_
at+ENST00000361379_at+

ENST00000358712_at+ENST00000336092_at+ENST00000382438_at+ENST00000226972_
at+ENST00000312524_at+

ENST00000372264_at+ENST00000362031_at+ENST00000355110_at+ENST00000332302_
at+ENST00000265395_at+

ENST00000376419_at+ENST00000376417_at+ENST00000277124_at+ENST00000381753_
at+ENST00000313860_at+

ENST00000383367_at+ENST00000318809_at+ENST00000308370_at+ENST00000383179_
at+ENST00000243562_at+

ENST00000252576_at+ENST00000381917_at+ENST00000375493_at+ENST00000204005_
at+ENST00000301788_at+

)361272_at+ENST00000361231_at+ENST00000298047_at+ENST00000342745_
)000244926_at+

)356505_at+ENST00000323760_at+ENST00000219168_at+ENST00000371534_
)000383362_at,
```

```
              data=smote_train,        kernel="radial",        cost=64,
gamma=0.001953125, types="response")
svmmod_90_RBF_no_SMOTE<-
svm(y~ENST00000371095_at+ENST00000371085_at+ENST00000371082_at+ENST000003
49036_at+ENST00000265620_at+

ENST00000263080_at+ENST00000313949_at+ENST00000338783_at+ENST00000369420_
at+ENST00000369417_at+

ENST00000306082_at+ENST00000381463_at+ENST00000361035_at+ENST00000371102_
at+ENST00000371101_at+

ENST00000274311_at+ENST00000325945_at+ENST00000371100_at+ENST00000371075_
at+ENST00000354359_at+

ENST00000306120_at+ENST00000306090_at+ENST00000375594_at+ENST00000360868_
at+ENST00000358855_at+

ENST00000381936_at+ENST00000359894_at+ENST00000300947_at+ENST00000390597_
at+ENST00000380991_at+

ENST00000380986_at+ENST00000265028_at+ENST00000359623_at+ENST00000323078_
at+ENST00000283646_at+

ENST00000180173_at+ENST00000244296_at+ENST00000221293_at+ENST00000381251_
at+ENST00000304101_at+

ENST00000371761_at+ENST00000217246_at+ENST00000262662_at+ENST00000356692_
at+ENST00000361625_at+

ENST00000253023_at+ENST00000367843_at+ENST00000367840_at+ENST00000367839_
at+ENST00000324705_at+

ENST00000312263_at+ENST00000298694_at+ENST00000298693_at+ENST00000354181_
at+ENST00000361379_at+

ENST00000358712_at+ENST00000336092_at+ENST00000382438_at+ENST00000226972_
at+ENST00000312524_at+

ENST00000372264_at+ENST00000362031_at+ENST00000355110_at+ENST00000332302_
at+ENST00000265395_at+

ENST00000376419_at+ENST00000376417_at+ENST00000277124_at+ENST00000381753_
at+ENST00000313860_at+

ENST00000383367_at+ENST00000318809_at+ENST00000308370_at+ENST00000383179_
at+ENST00000243562_at+

ENST00000252576_at+ENST00000381917_at+ENST00000375493_at+ENST00000204005_
at+ENST00000301788_at+

ENST00000361272_at+ENST00000361231_at+ENST00000298047_at+ENST00000342745_
at+ENST00000244926_at+

ENST00000356505_at+ENST00000323760_at+ENST00000219168_at+ENST00000371534_
at+ENST00000383362_at,
                  data=trainData,        kernel="radial",        cost=64,
gamma=0.001953125, types="response")
      _with_SMOTE_RBF90<-prediction(svmmod_90_RBF_with_SMOTE,testData)
      _no_SMOTE_RBF90<-prediction(svmmod_90_RBF_no_SMOTE,testData)
```

1 : Kernel RBF dengan 100 Fitur Teratas

```
svmmod_100_RBF_with_SMOTE<-
svm(y~ENST00000371095_at+ENST00000371085_at+ENST00000371082_at+ENST000003
49036_at+ENST00000265620_at+

ENST00000263080_at+ENST00000313949_at+ENST00000338783_at+ENST00000369420_
at+ENST00000369417_at+

ENST00000306082_at+ENST00000381463_at+ENST00000361035_at+ENST00000371102_
at+ENST00000371101_at+

ENST00000274311_at+ENST00000325945_at+ENST00000371100_at+ENST00000371075_
at+ENST00000354359_at+

ENST00000306120_at+ENST00000306090_at+ENST00000375594_at+ENST00000360868_
at+ENST00000358855_at+

ENST00000381936_at+ENST00000359894_at+ENST00000300947_at+ENST00000390597_
at+ENST00000380991_at+

ENST00000380986_at+ENST00000265028_at+ENST00000359623_at+ENST00000323078_
at+ENST00000283646_at+

ENST00000180173_at+ENST00000244296_at+ENST00000221293_at+ENST00000381251_
at+ENST00000304101_at+

ENST00000371761_at+ENST00000217246_at+ENST00000262662_at+ENST00000356692_
at+ENST00000361625_at+

ENST00000253023_at+ENST00000367843_at+ENST00000367840_at+ENST00000367839_
at+ENST00000324705_at+

ENST00000312263_at+ENST00000298694_at+ENST00000298693_at+ENST00000354181_
at+ENST00000361379_at+

ENST00000358712_at+ENST00000336092_at+ENST00000382438_at+ENST00000226972_
at+ENST00000312524_at+

ENST00000372264_at+ENST00000362031_at+ENST00000355110_at+ENST00000332302_
at+ENST00000265395_at+

ENST00000376419_at+ENST00000376417_at+ENST00000277124_at+ENST00000381753_
at+ENST00000313860_at+

ENST00000383367_at+ENST00000318809_at+ENST00000308370_at+ENST00000383179_
at+ENST00000243562_at+

ENST00000252576_at+ENST00000381917_at+ENST00000375493_at+ENST00000204005_
at+ENST00000301788_at+

ENST00000361272_at+ENST00000361231_at+ENST00000298047_at+ENST00000342745_
at+ENST00000244926_at+

ENST00000356505_at+ENST00000323760_at+ENST00000219168_at+ENST00000371534_
at+ENST00000383362_at+

ENST00000383360_at+ENST00000383359_at+ENST00000370109_at+ENST00000383178_
at+ENST00000383177_at+

)370107_at+ENST00000371646_at+ENST00000223864_at+ENST00000371554_
)000342173_at,
       data=smote_train,        kernel="radial",        cost=64,
)01953125, types="response")
```

```
svmmod_100_RBF_no_SMOTE<-
svm(y~ENST00000371095_at+ENST00000371085_at+ENST00000371082_at+ENST000003
49036_at+ENST00000265620_at+

ENST00000263080_at+ENST00000313949_at+ENST00000338783_at+ENST00000369420_
at+ENST00000369417_at+

ENST00000306082_at+ENST00000381463_at+ENST00000361035_at+ENST00000371102_
at+ENST00000371101_at+

ENST00000274311_at+ENST00000325945_at+ENST00000371100_at+ENST00000371075_
at+ENST00000354359_at+

ENST00000306120_at+ENST00000306090_at+ENST00000375594_at+ENST00000360868_
at+ENST00000358855_at+

ENST00000381936_at+ENST00000359894_at+ENST00000300947_at+ENST00000390597_
at+ENST00000380991_at+

ENST00000380986_at+ENST00000265028_at+ENST00000359623_at+ENST00000323078_
at+ENST00000283646_at+

ENST00000180173_at+ENST00000244296_at+ENST00000221293_at+ENST00000381251_
at+ENST00000304101_at+

ENST00000371761_at+ENST00000217246_at+ENST00000262662_at+ENST00000356692_
at+ENST00000361625_at+

ENST00000253023_at+ENST00000367843_at+ENST00000367840_at+ENST00000367839_
at+ENST00000324705_at+

ENST00000312263_at+ENST00000298694_at+ENST00000298693_at+ENST00000354181_
at+ENST00000361379_at+

ENST00000358712_at+ENST00000336092_at+ENST00000382438_at+ENST00000226972_
at+ENST00000312524_at+

ENST00000372264_at+ENST00000362031_at+ENST00000355110_at+ENST00000332302_
at+ENST00000265395_at+

ENST00000376419_at+ENST00000376417_at+ENST00000277124_at+ENST00000381753_
at+ENST00000313860_at+

ENST00000383367_at+ENST00000318809_at+ENST00000308370_at+ENST00000383179_
at+ENST00000243562_at+

ENST00000252576_at+ENST00000381917_at+ENST00000375493_at+ENST00000204005_
at+ENST00000301788_at+

ENST00000361272_at+ENST00000361231_at+ENST00000298047_at+ENST00000342745_
at+ENST00000244926_at+

ENST00000356505_at+ENST00000323760_at+ENST00000219168_at+ENST00000371534_
at+ENST00000383362_at+

ENST00000383360_at+ENST00000383359_at+ENST00000370109_at+ENST00000383178_
at+ENST00000383177_at+

)370107_at+ENST00000371646_at+ENST00000223864_at+ENST00000371554_
)000342173_at,
                              data=trainData, kernel="radial", cost=64,
)01953125, types="response")
_with_SMOTE_RBF100<-
)n(svmmod_100_RBF_with_SMOTE,testData)
```

```
prediksi_no_SMOTE_RBF100<-prediction(svmmod_100_RBF_no_SMOTE,testData)

#Kernel Polynomial Orde 2
#Dataset 1 : Kernel Polynomial Orde 2 dengan 10 Fitur Teratas
svmmod_10_Poly2_with_SMOTE<-
svm(y~ENST00000371095_at+ENST00000371085_at+ENST00000371082_at+ENST000003
49036_at+ENST00000265620_at+

ENST00000263080_at+ENST00000313949_at+ENST00000338783_at+ENST00000369420_
at+ENST00000369417_at,
                    data=smote_train,    kernel="polynomial",    cost=32,
degree=2, coef0=0.125, gamma=0.0004882812, types="response")
svmmod_10_Poly2_no_SMOTE<-
svm(y~ENST00000371095_at+ENST00000371085_at+ENST00000371082_at+ENST000003
49036_at+ENST00000265620_at+

ENST00000263080_at+ENST00000313949_at+ENST00000338783_at+ENST00000369420_
at+ENST00000369417_at,
                    data=trainData,     kernel="polynomial",     cost=32,
degree=2, coef0=0.125, gamma=0.0004882812, types="response")
prediksi_with_SMOTE_Poly10<-
prediction(svmmod_10_Poly2_with_SMOTE,testData)
prediksi_no_SMOTE_Poly10<-prediction(svmmod_10_Poly2_no_SMOTE,testData)

#Dataset 1 : Kernel Polynomial Orde 2 dengan 20 Fitur Teratas
svmmod_20_Poly2_with_SMOTE<-
svm(y~ENST00000371095_at+ENST00000371085_at+ENST00000371082_at+ENST000003
49036_at+ENST00000265620_at+

ENST00000263080_at+ENST00000313949_at+ENST00000338783_at+ENST00000369420_
at+ENST00000369417_at+

ENST00000306082_at+ENST00000381463_at+ENST00000361035_at+ENST00000371102_
at+ENST00000371101_at+

ENST00000274311_at+ENST00000325945_at+ENST00000371100_at+ENST00000371075_
at+ENST00000354359_at,
                    data=smote_train,    kernel="polynomial",    cost=32,
degree=2, coef0=0.125, gamma=0.0004882812, types="response")
svmmod_20_Poly2_no_SMOTE<-
svm(y~ENST00000371095_at+ENST00000371085_at+ENST00000371082_at+ENST000003
49036_at+ENST00000265620_at+

ENST00000263080_at+ENST00000313949_at+ENST00000338783_at+ENST00000369420_
at+ENST00000369417_at+

ENST00000306082_at+ENST00000381463_at+ENST00000361035_at+ENST00000371102_
at+ENST00000371101_at+

ENST00000274311_at+ENST00000325945_at+ENST00000371100_at+ENST00000371075_
at+ENST00000354359_at,
                                data=trainData,      kernel="polynomial",
cost=32, degree=2, coef0=0.125, gamma=0.0004882812, types="response")
prediksi_with_SMOTE_Poly20<-
prediction(svmmod_20_Poly2_with_SMOTE,testData)
prediksi_no_SMOTE_Poly20<-prediction(svmmod_20_Poly2_no_SMOTE,testData)

#Dataset 1 : Kernel Polynomial Orde 2 dengan 30 Fitur Teratas
      )_Poly2_with_SMOTE<-
      ST00000371095_at+ENST00000371085_at+ENST00000371082_at+ENST000003
      +ENST00000265620_at+

      )263080_at+ENST00000313949_at+ENST00000338783_at+ENST00000369420_
      )000369417_at+
```

```
ENST00000306082_at+ENST00000381463_at+ENST00000361035_at+ENST00000371102_
at+ENST00000371101_at+

ENST00000274311_at+ENST00000325945_at+ENST00000371100_at+ENST00000371075_
at+ENST00000354359_at+

ENST00000306120_at+ENST00000306090_at+ENST00000375594_at+ENST00000360868_
at+ENST00000358855_at+

ENST00000381936_at+ENST00000359894_at+ENST00000300947_at+ENST00000390597_
at+ENST00000380991_at,
                      data=smote_train,    kernel="polynomial",    cost=32,
degree=2, coef0=0.125, gamma=0.0004882812, types="response")
svmmod_30_Poly2_no_SMOTE<-
svm(y~ENST00000371095_at+ENST00000371085_at+ENST00000371082_at+ENST000003
49036_at+ENST00000265620_at+

ENST00000263080_at+ENST00000313949_at+ENST00000338783_at+ENST00000369420_
at+ENST00000369417_at+

ENST00000306082_at+ENST00000381463_at+ENST00000361035_at+ENST00000371102_
at+ENST00000371101_at+

ENST00000274311_at+ENST00000325945_at+ENST00000371100_at+ENST00000371075_
at+ENST00000354359_at+

ENST00000306120_at+ENST00000306090_at+ENST00000375594_at+ENST00000360868_
at+ENST00000358855_at+

ENST00000381936_at+ENST00000359894_at+ENST00000300947_at+ENST00000390597_
at+ENST00000380991_at,
                                 data=trainData,      kernel="polynomial",
cost=32, degree=2, coef0=0.125, gamma=0.0004882812, types="response")
prediksi_with_SMOTE_Poly30<-
prediction(svmmod_30_Poly2_with_SMOTE,testData)
prediksi_no_SMOTE_Poly30<-prediction(svmmod_30_Poly2_no_SMOTE,testData)

#Dataset 1 : Kernel Polynomial Orde 2 dengan 40 Fitur Teratas
svmmod_40_Poly2_with_SMOTE<-
svm(y~ENST00000371095_at+ENST00000371085_at+ENST00000371082_at+ENST000003
49036_at+ENST00000265620_at+

ENST00000263080_at+ENST00000313949_at+ENST00000338783_at+ENST00000369420_
at+ENST00000369417_at+

ENST00000306082_at+ENST00000381463_at+ENST00000361035_at+ENST00000371102_
at+ENST00000371101_at+

ENST00000274311_at+ENST00000325945_at+ENST00000371100_at+ENST00000371075_
at+ENST00000354359_at+

ENST00000306120_at+ENST00000306090_at+ENST00000375594_at+ENST00000360868_
at+ENST00000358855_at+

ENST00000381936_at+ENST00000359894_at+ENST00000300947_at+ENST00000390597_
at+ENST00000380991_at+

)380986_at+ENST00000265028_at+ENST00000359623_at+ENST00000323078_
)000283646_at+

)180173_at+ENST00000244296_at+ENST00000221293_at+ENST00000381251_
)000304101_at,
```

```
                    data=smote_train,    kernel="polynomial",    cost=32,
degree=2, coef0=0.125, gamma=0.0004882812, types="response")
svmmod_40_Poly2_no_SMOTE<-
svm(y~ENST00000371095_at+ENST00000371085_at+ENST00000371082_at+ENST000003
49036_at+ENST00000265620_at+

ENST00000263080_at+ENST00000313949_at+ENST00000338783_at+ENST00000369420_
at+ENST00000369417_at+

ENST00000306082_at+ENST00000381463_at+ENST00000361035_at+ENST00000371102_
at+ENST00000371101_at+

ENST00000274311_at+ENST00000325945_at+ENST00000371100_at+ENST00000371075_
at+ENST00000354359_at+

ENST00000306120_at+ENST00000306090_at+ENST00000375594_at+ENST00000360868_
at+ENST00000358855_at+

ENST00000381936_at+ENST00000359894_at+ENST00000300947_at+ENST00000390597_
at+ENST00000380991_at+

ENST00000380986_at+ENST00000265028_at+ENST00000359623_at+ENST00000323078_
at+ENST00000283646_at+

ENST00000180173_at+ENST00000244296_at+ENST00000221293_at+ENST00000381251_
at+ENST00000304101_at,
                    data=trainData,     kernel="polynomial",     cost=32,
degree=2, coef0=0.125, gamma=0.0004882812, types="response")
prediksi_with_SMOTE_Poly40<-
prediction(svmmod_40_Poly2_with_SMOTE,testData)
prediksi_no_SMOTE_Poly40<-prediction(svmmod_40_Poly2_no_SMOTE,testData)

#Dataset 1 : Kernel Polynomial Orde 2 dengan 50 Fitur Teratas
svmmod_50_Poly2_with_SMOTE<-
svm(y~ENST00000371095_at+ENST00000371085_at+ENST00000371082_at+ENST000003
49036_at+ENST00000265620_at+

ENST00000263080_at+ENST00000313949_at+ENST00000338783_at+ENST00000369420_
at+ENST00000369417_at+

ENST00000306082_at+ENST00000381463_at+ENST00000361035_at+ENST00000371102_
at+ENST00000371101_at+

ENST00000274311_at+ENST00000325945_at+ENST00000371100_at+ENST00000371075_
at+ENST00000354359_at+

ENST00000306120_at+ENST00000306090_at+ENST00000375594_at+ENST00000360868_
at+ENST00000358855_at+

ENST00000381936_at+ENST00000359894_at+ENST00000300947_at+ENST00000390597_
at+ENST00000380991_at+

ENST00000380986_at+ENST00000265028_at+ENST00000359623_at+ENST00000323078_
at+ENST00000283646_at+

ENST00000180173_at+ENST00000244296_at+ENST00000221293_at+ENST00000381251_
at+ENST00000304101_at+

    )371761_at+ENST00000217246_at+ENST00000262662_at+ENST00000356692_
    )000361625_at+

    )253023_at+ENST00000367843_at+ENST00000367840_at+ENST00000367839_
    )000324705_at,
```

```
                    data=smote_train,    kernel="polynomial",    cost=32,
degree=2, coef0=0.125, gamma=0.0004882812, types="response")
svmmod_50_Poly2_no_SMOTE<-
svm(y~ENST00000371095_at+ENST00000371085_at+ENST00000371082_at+ENST000003
49036_at+ENST00000265620_at+

ENST00000263080_at+ENST00000313949_at+ENST00000338783_at+ENST00000369420_
at+ENST00000369417_at+

ENST00000306082_at+ENST00000381463_at+ENST00000361035_at+ENST00000371102_
at+ENST00000371101_at+

ENST00000274311_at+ENST00000325945_at+ENST00000371100_at+ENST00000371075_
at+ENST00000354359_at+

ENST00000306120_at+ENST00000306090_at+ENST00000375594_at+ENST00000360868_
at+ENST00000358855_at+

ENST00000381936_at+ENST00000359894_at+ENST00000300947_at+ENST00000390597_
at+ENST00000380991_at+

ENST00000380986_at+ENST00000265028_at+ENST00000359623_at+ENST00000323078_
at+ENST00000283646_at+

ENST00000180173_at+ENST00000244296_at+ENST00000221293_at+ENST00000381251_
at+ENST00000304101_at+

ENST00000371761_at+ENST00000217246_at+ENST00000262662_at+ENST00000356692_
at+ENST00000361625_at+

ENST00000253023_at+ENST00000367843_at+ENST00000367840_at+ENST00000367839_
at+ENST00000324705_at,
                    data=trainData,     kernel="polynomial",     cost=32,
degree=2, coef0=0.125, gamma=0.0004882812, types="response")
prediksi_with_SMOTE_Poly50<-
prediction(svmmod_50_Poly2_with_SMOTE,testData)
prediksi_no_SMOTE_Poly50<-prediction(svmmod_50_Poly2_no_SMOTE,testData)

#Dataset 1 : Kernel Polynomial Orde 2 dengan 60 Fitur Teratas
svmmod_60_Poly2_with_SMOTE<-
svm(y~ENST00000371095_at+ENST00000371085_at+ENST00000371082_at+ENST000003
49036_at+ENST00000265620_at+

ENST00000263080_at+ENST00000313949_at+ENST00000338783_at+ENST00000369420_
at+ENST00000369417_at+

ENST00000306082_at+ENST00000381463_at+ENST00000361035_at+ENST00000371102_
at+ENST00000371101_at+

ENST00000274311_at+ENST00000325945_at+ENST00000371100_at+ENST00000371075_
at+ENST00000354359_at+

ENST00000306120_at+ENST00000306090_at+ENST00000375594_at+ENST00000360868_
at+ENST00000358855_at+

ENST00000381936_at+ENST00000359894_at+ENST00000300947_at+ENST00000390597_
at+ENST00000380991_at+

)380986_at+ENST00000265028_at+ENST00000359623_at+ENST00000323078_
)000283646_at+

)180173_at+ENST00000244296_at+ENST00000221293_at+ENST00000381251_
)000304101_at+
```

```
ENST00000371761_at+ENST00000217246_at+ENST00000262662_at+ENST00000356692_
at+ENST00000361625_at+

ENST00000253023_at+ENST00000367843_at+ENST00000367840_at+ENST00000367839_
at+ENST00000324705_at+

ENST00000312263_at+ENST00000298694_at+ENST00000298693_at+ENST00000354181_
at+ENST00000361379_at+

ENST00000358712_at+ENST00000336092_at+ENST00000382438_at+ENST00000226972_
at+ENST00000312524_at,
                     data=smote_train,    kernel="polynomial",    cost=32,
degree=2, coef0=0.125, gamma=0.0004882812, types="response")
svmmod_60_Poly2_no_SMOTE<-
svm(y~ENST00000371095_at+ENST00000371085_at+ENST00000371082_at+ENST000003
49036_at+ENST00000265620_at+

ENST00000263080_at+ENST00000313949_at+ENST00000338783_at+ENST00000369420_
at+ENST00000369417_at+

ENST00000306082_at+ENST00000381463_at+ENST00000361035_at+ENST00000371102_
at+ENST00000371101_at+

ENST00000274311_at+ENST00000325945_at+ENST00000371100_at+ENST00000371075_
at+ENST00000354359_at+

ENST00000306120_at+ENST00000306090_at+ENST00000375594_at+ENST00000360868_
at+ENST00000358855_at+

ENST00000381936_at+ENST00000359894_at+ENST00000300947_at+ENST00000390597_
at+ENST00000380991_at+

ENST00000380986_at+ENST00000265028_at+ENST00000359623_at+ENST00000323078_
at+ENST00000283646_at+

ENST00000180173_at+ENST00000244296_at+ENST00000221293_at+ENST00000381251_
at+ENST00000304101_at+

ENST00000371761_at+ENST00000217246_at+ENST00000262662_at+ENST00000356692_
at+ENST00000361625_at+

ENST00000253023_at+ENST00000367843_at+ENST00000367840_at+ENST00000367839_
at+ENST00000324705_at+

ENST00000312263_at+ENST00000298694_at+ENST00000298693_at+ENST00000354181_
at+ENST00000361379_at+

ENST00000358712_at+ENST00000336092_at+ENST00000382438_at+ENST00000226972_
at+ENST00000312524_at,
                              data=trainData,      kernel="polynomial",
cost=32, degree=2, coef0=0.125, gamma=0.0004882812, types="response")
prediksi_with_SMOTE_Poly60<-
prediction(svmmod_60_Poly2_with_SMOTE,testData)
prediksi_no_SMOTE_Poly60<-prediction(svmmod_60_Poly2_no_SMOTE,testData)

#Dataset 1 : Kernel Polynomial Orde 2 dengan 70 Fitur Teratas
svmmod_70_Poly2_with_SMOTE<-
        ST00000371095_at+ENST00000371085_at+ENST00000371082_at+ENST000003
        +ENST00000265620_at+

        )263080_at+ENST00000313949_at+ENST00000338783_at+ENST00000369420_
        )000369417_at+
```

```
ENST00000306082_at+ENST00000381463_at+ENST00000361035_at+ENST00000371102_
at+ENST00000371101_at+

ENST00000274311_at+ENST00000325945_at+ENST00000371100_at+ENST00000371075_
at+ENST00000354359_at+

ENST00000306120_at+ENST00000306090_at+ENST00000375594_at+ENST00000360868_
at+ENST00000358855_at+

ENST00000381936_at+ENST00000359894_at+ENST00000300947_at+ENST00000390597_
at+ENST00000380991_at+

ENST00000380986_at+ENST00000265028_at+ENST00000359623_at+ENST00000323078_
at+ENST00000283646_at+

ENST00000180173_at+ENST00000244296_at+ENST00000221293_at+ENST00000381251_
at+ENST00000304101_at+

ENST00000371761_at+ENST00000217246_at+ENST00000262662_at+ENST00000356692_
at+ENST00000361625_at+

ENST00000253023_at+ENST00000367843_at+ENST00000367840_at+ENST00000367839_
at+ENST00000324705_at+

ENST00000312263_at+ENST00000298694_at+ENST00000298693_at+ENST00000354181_
at+ENST00000361379_at+

ENST00000358712_at+ENST00000336092_at+ENST00000382438_at+ENST00000226972_
at+ENST00000312524_at+

ENST00000361272_at+ENST00000361231_at+ENST00000298047_at+ENST00000342745_
at+ENST00000244926_at+

ENST00000356505_at+ENST00000323760_at+ENST00000219168_at+ENST00000371534_
at+ENST00000383362_at,
                    data=smote_train,    kernel="polynomial",    cost=32,
degree=2, coef0=0.125, gamma=0.0004882812, types="response")
svmmod_70_Poly2_no_SMOTE<-
svm(y~ENST00000371095_at+ENST00000371085_at+ENST00000371082_at+ENST000003
49036_at+ENST00000265620_at+

ENST00000263080_at+ENST00000313949_at+ENST00000338783_at+ENST00000369420_
at+ENST00000369417_at+

ENST00000306082_at+ENST00000381463_at+ENST00000361035_at+ENST00000371102_
at+ENST00000371101_at+

ENST00000274311_at+ENST00000325945_at+ENST00000371100_at+ENST00000371075_
at+ENST00000354359_at+

ENST00000306120_at+ENST00000306090_at+ENST00000375594_at+ENST00000360868_
at+ENST00000358855_at+

ENST00000381936_at+ENST00000359894_at+ENST00000300947_at+ENST00000390597_
at+ENST00000380991_at+

ENST00000380986_at+ENST00000265028_at+ENST00000359623_at+ENST00000323078_
          )000283646_at+

          )0180173_at+ENST00000244296_at+ENST00000221293_at+ENST00000381251_
          )000304101_at+
```

```
ENST00000371761_at+ENST00000217246_at+ENST00000262662_at+ENST00000356692_
at+ENST00000361625_at+

ENST00000253023_at+ENST00000367843_at+ENST00000367840_at+ENST00000367839_
at+ENST00000324705_at+

ENST00000312263_at+ENST00000298694_at+ENST00000298693_at+ENST00000354181_
at+ENST00000361379_at+

ENST00000358712_at+ENST00000336092_at+ENST00000382438_at+ENST00000226972_
at+ENST00000312524_at+

ENST00000361272_at+ENST00000361231_at+ENST00000298047_at+ENST00000342745_
at+ENST00000244926_at+

ENST00000356505_at+ENST00000323760_at+ENST00000219168_at+ENST00000371534_
at+ENST00000383362_at,
                    data=trainData,     kernel="polynomial",     cost=32,
degree=2, coef0=0.125, gamma=0.0004882812, types="response")
prediksi_with_SMOTE_Poly70<-
prediction(svmmod_70_Poly2_with_SMOTE,testData)
prediksi_no_SMOTE_Poly70<-prediction(svmmod_70_Poly2_no_SMOTE,testData)

#Dataset 1 : Kernel Polynomial Orde 2 dengan 80 Fitur Teratas
svmmod_80_Poly2_with_SMOTE<-
svm(y~ENST00000371095_at+ENST00000371085_at+ENST00000371082_at+ENST000003
49036_at+ENST00000265620_at+

ENST00000263080_at+ENST00000313949_at+ENST00000338783_at+ENST00000369420_
at+ENST00000369417_at+

ENST00000306082_at+ENST00000381463_at+ENST00000361035_at+ENST00000371102_
at+ENST00000371101_at+

ENST00000274311_at+ENST00000325945_at+ENST00000371100_at+ENST00000371075_
at+ENST00000354359_at+

ENST00000306120_at+ENST00000306090_at+ENST00000375594_at+ENST00000360868_
at+ENST00000358855_at+

ENST00000381936_at+ENST00000359894_at+ENST00000300947_at+ENST00000390597_
at+ENST00000380991_at+

ENST00000380986_at+ENST00000265028_at+ENST00000359623_at+ENST00000323078_
at+ENST00000283646_at+

ENST00000180173_at+ENST00000244296_at+ENST00000221293_at+ENST00000381251_
at+ENST00000304101_at+

ENST00000371761_at+ENST00000217246_at+ENST00000262662_at+ENST00000356692_
at+ENST00000361625_at+

ENST00000253023_at+ENST00000367843_at+ENST00000367840_at+ENST00000367839_
at+ENST00000324705_at+

ENST00000312263_at+ENST00000298694_at+ENST00000298693_at+ENST00000354181_
at+ENST00000361379_at+

)358712_at+ENST00000336092_at+ENST00000382438_at+ENST00000226972_
)000312524_at+

)372264_at+ENST00000362031_at+ENST00000355110_at+ENST00000332302_
)000265395_at+
```

```
ENST00000376419_at+ENST00000376417_at+ENST00000277124_at+ENST00000381753_
at+ENST00000313860_at+

ENST00000383367_at+ENST00000318809_at+ENST00000308370_at+ENST00000383179_
at+ENST00000243562_at+

ENST00000252576_at+ENST00000381917_at+ENST00000375493_at+ENST00000204005_
at+ENST00000301788_at,
                    data=smote_train,    kernel="polynomial",    cost=32,
degree=2, coef0=0.125, gamma=0.0004882812, types="response")
svmmod_80_Poly2_no_SMOTE<-
svm(y~ENST00000371095_at+ENST00000371085_at+ENST00000371082_at+ENST000003
49036_at+ENST00000265620_at+

ENST00000263080_at+ENST00000313949_at+ENST00000338783_at+ENST00000369420_
at+ENST00000369417_at+

ENST00000306082_at+ENST00000381463_at+ENST00000361035_at+ENST00000371102_
at+ENST00000371101_at+

ENST00000274311_at+ENST00000325945_at+ENST00000371100_at+ENST00000371075_
at+ENST00000354359_at+

ENST00000306120_at+ENST00000306090_at+ENST00000375594_at+ENST00000360868_
at+ENST00000358855_at+

ENST00000381936_at+ENST00000359894_at+ENST00000300947_at+ENST00000390597_
at+ENST00000380991_at+

ENST00000380986_at+ENST00000265028_at+ENST00000359623_at+ENST00000323078_
at+ENST00000283646_at+

ENST00000180173_at+ENST00000244296_at+ENST00000221293_at+ENST00000381251_
at+ENST00000304101_at+

ENST00000371761_at+ENST00000217246_at+ENST00000262662_at+ENST00000356692_
at+ENST00000361625_at+

ENST00000253023_at+ENST00000367843_at+ENST00000367840_at+ENST00000367839_
at+ENST00000324705_at+

ENST00000312263_at+ENST00000298694_at+ENST00000298693_at+ENST00000354181_
at+ENST00000361379_at+

ENST00000358712_at+ENST00000336092_at+ENST00000382438_at+ENST00000226972_
at+ENST00000312524_at+

ENST00000372264_at+ENST00000362031_at+ENST00000355110_at+ENST00000332302_
at+ENST00000265395_at+

ENST00000376419_at+ENST00000376417_at+ENST00000277124_at+ENST00000381753_
at+ENST00000313860_at+

ENST00000383367_at+ENST00000318809_at+ENST00000308370_at+ENST00000383179_
at+ENST00000243562_at+

ENST00000252576_at+ENST00000381917_at+ENST00000375493_at+ENST00000204005_
     )000301788_at,
                                data=trainData,      kernel="polynomial",
 degree=2, coef0=0.125, gamma=0.0004882812, types="response")
_with_SMOTE_Poly80<-
)n(svmmod_80_Poly2_with_SMOTE,testData)
_no_SMOTE_Poly80<-prediction(svmmod_80_Poly2_no_SMOTE,testData)
```

```
#Dataset 1 : Kernel Polynomial Orde 2 dengan 90 Fitur Teratas
svmmod_90_Poly2_with_SMOTE<-
svm(y~ENST00000371095_at+ENST00000371085_at+ENST00000371082_at+ENST000003
49036_at+ENST00000265620_at+

ENST00000263080_at+ENST00000313949_at+ENST00000338783_at+ENST00000369420_
at+ENST00000369417_at+

ENST00000306082_at+ENST00000381463_at+ENST00000361035_at+ENST00000371102_
at+ENST00000371101_at+

ENST00000274311_at+ENST00000325945_at+ENST00000371100_at+ENST00000371075_
at+ENST00000354359_at+

ENST00000306120_at+ENST00000306090_at+ENST00000375594_at+ENST00000360868_
at+ENST00000358855_at+

ENST00000381936_at+ENST00000359894_at+ENST00000300947_at+ENST00000390597_
at+ENST00000380991_at+

ENST00000380986_at+ENST00000265028_at+ENST00000359623_at+ENST00000323078_
at+ENST00000283646_at+

ENST00000180173_at+ENST00000244296_at+ENST00000221293_at+ENST00000381251_
at+ENST00000304101_at+

ENST00000371761_at+ENST00000217246_at+ENST00000262662_at+ENST00000356692_
at+ENST00000361625_at+

ENST00000253023_at+ENST00000367843_at+ENST00000367840_at+ENST00000367839_
at+ENST00000324705_at+

ENST00000312263_at+ENST00000298694_at+ENST00000298693_at+ENST00000354181_
at+ENST00000361379_at+

ENST00000358712_at+ENST00000336092_at+ENST00000382438_at+ENST00000226972_
at+ENST00000312524_at+

ENST00000372264_at+ENST00000362031_at+ENST00000355110_at+ENST00000332302_
at+ENST00000265395_at+

ENST00000376419_at+ENST00000376417_at+ENST00000277124_at+ENST00000381753_
at+ENST00000313860_at+

ENST00000383367_at+ENST00000318809_at+ENST00000308370_at+ENST00000383179_
at+ENST00000243562_at+

ENST00000252576_at+ENST00000381917_at+ENST00000375493_at+ENST00000204005_
at+ENST00000301788_at+

ENST00000361272_at+ENST00000361231_at+ENST00000298047_at+ENST00000342745_
at+ENST00000244926_at+

ENST00000356505_at+ENST00000323760_at+ENST00000219168_at+ENST00000371534_
at+ENST00000383362_at,
                    data=smote_train,    kernel="polynomial",    cost=32,
degree=2, coef0=0.125, gamma=0.0004882812, types="response")
        )_Poly2_no_SMOTE<-
        ST00000371095_at+ENST00000371085_at+ENST00000371082_at+ENST000003
        +ENST00000265620_at+

        )263080_at+ENST00000313949_at+ENST00000338783_at+ENST00000369420_
        )000369417_at+
```

```
ENST00000306082_at+ENST00000381463_at+ENST00000361035_at+ENST00000371102_
at+ENST00000371101_at+

ENST00000274311_at+ENST00000325945_at+ENST00000371100_at+ENST00000371075_
at+ENST00000354359_at+

ENST00000306120_at+ENST00000306090_at+ENST00000375594_at+ENST00000360868_
at+ENST00000358855_at+

ENST00000381936_at+ENST00000359894_at+ENST00000300947_at+ENST00000390597_
at+ENST00000380991_at+

ENST00000380986_at+ENST00000265028_at+ENST00000359623_at+ENST00000323078_
at+ENST00000283646_at+

ENST00000180173_at+ENST00000244296_at+ENST00000221293_at+ENST00000381251_
at+ENST00000304101_at+

ENST00000371761_at+ENST00000217246_at+ENST00000262662_at+ENST00000356692_
at+ENST00000361625_at+

ENST00000253023_at+ENST00000367843_at+ENST00000367840_at+ENST00000367839_
at+ENST00000324705_at+

ENST00000312263_at+ENST00000298694_at+ENST00000298693_at+ENST00000354181_
at+ENST00000361379_at+

ENST00000358712_at+ENST00000336092_at+ENST00000382438_at+ENST00000226972_
at+ENST00000312524_at+

ENST00000372264_at+ENST00000362031_at+ENST00000355110_at+ENST00000332302_
at+ENST00000265395_at+

ENST00000376419_at+ENST00000376417_at+ENST00000277124_at+ENST00000381753_
at+ENST00000313860_at+

ENST00000383367_at+ENST00000318809_at+ENST00000308370_at+ENST00000383179_
at+ENST00000243562_at+

ENST00000252576_at+ENST00000381917_at+ENST00000375493_at+ENST00000204005_
at+ENST00000301788_at+

ENST00000361272_at+ENST00000361231_at+ENST00000298047_at+ENST00000342745_
at+ENST00000244926_at+

ENST00000356505_at+ENST00000323760_at+ENST00000219168_at+ENST00000371534_
at+ENST00000383362_at,
                                  data=trainData,      kernel="polynomial",
cost=32, degree=2, coef0=0.125, gamma=0.0004882812, types="response")
prediksi_with_SMOTE_Poly90<-
prediction(svmmod_90_Poly2_with_SMOTE,testData)
prediksi_no_SMOTE_Poly90<-prediction(svmmod_90_Poly2_no_SMOTE,testData)

#Dataset 1 : Kernel Polynomial Orde 2 dengan 100 Fitur Teratas
svmmod_100_Poly2_with_SMOTE<-
svm(y~ENST00000371095_at+ENST00000371085_at+ENST00000371082_at+ENST000003
49036_at+ENST00000265620_at+

                    0263080_at+ENST00000313949_at+ENST00000338783_at+ENST00000369420_
                    0000369417_at+

                    0306082_at+ENST00000381463_at+ENST00000361035_at+ENST00000371102_
                    0000371101_at+
```

```
ENST00000274311_at+ENST00000325945_at+ENST00000371100_at+ENST00000371075_
at+ENST00000354359_at+

ENST00000306120_at+ENST00000306090_at+ENST00000375594_at+ENST00000360868_
at+ENST00000358855_at+

ENST00000381936_at+ENST00000359894_at+ENST00000300947_at+ENST00000390597_
at+ENST00000380991_at+

ENST00000380986_at+ENST00000265028_at+ENST00000359623_at+ENST00000323078_
at+ENST00000283646_at+

ENST00000180173_at+ENST00000244296_at+ENST00000221293_at+ENST00000381251_
at+ENST00000304101_at+

ENST00000371761_at+ENST00000217246_at+ENST00000262662_at+ENST00000356692_
at+ENST00000361625_at+

ENST00000253023_at+ENST00000367843_at+ENST00000367840_at+ENST00000367839_
at+ENST00000324705_at+

ENST00000312263_at+ENST00000298694_at+ENST00000298693_at+ENST00000354181_
at+ENST00000361379_at+

ENST00000358712_at+ENST00000336092_at+ENST00000382438_at+ENST00000226972_
at+ENST00000312524_at+

ENST00000372264_at+ENST00000362031_at+ENST00000355110_at+ENST00000332302_
at+ENST00000265395_at+

ENST00000376419_at+ENST00000376417_at+ENST00000277124_at+ENST00000381753_
at+ENST00000313860_at+

ENST00000383367_at+ENST00000318809_at+ENST00000308370_at+ENST00000383179_
at+ENST00000243562_at+

ENST00000252576_at+ENST00000381917_at+ENST00000375493_at+ENST00000204005_
at+ENST00000301788_at+

ENST00000361272_at+ENST00000361231_at+ENST00000298047_at+ENST00000342745_
at+ENST00000244926_at+

ENST00000356505_at+ENST00000323760_at+ENST00000219168_at+ENST00000371534_
at+ENST00000383362_at+

ENST00000383360_at+ENST00000383359_at+ENST00000370109_at+ENST00000383178_
at+ENST00000383177_at+

ENST00000370107_at+ENST00000371646_at+ENST00000223864_at+ENST00000371554_
at+ENST00000342173_at,
                      data=smote_train,   kernel="polynomial",   cost=32,
degree=2, coef0=0.125, gamma=0.0004882812, types="response")
svmmod_100_Poly2_no_SMOTE<-
svm(y~ENST00000371095_at+ENST00000371085_at+ENST00000371082_at+ENST000003
49036_at+ENST00000265620_at+

ENST00000263080_at+ENST00000313949_at+ENST00000338783_at+ENST00000369420_
      0000369417_at+

      0306082_at+ENST00000381463_at+ENST00000361035_at+ENST00000371102_
      0000371101_at+
```

```
ENST00000274311_at+ENST00000325945_at+ENST00000371100_at+ENST00000371075_
at+ENST00000354359_at+

ENST00000306120_at+ENST00000306090_at+ENST00000375594_at+ENST00000360868_
at+ENST00000358855_at+

ENST00000381936_at+ENST00000359894_at+ENST00000300947_at+ENST00000390597_
at+ENST00000380991_at+

ENST00000380986_at+ENST00000265028_at+ENST00000359623_at+ENST00000323078_
at+ENST00000283646_at+

ENST00000180173_at+ENST00000244296_at+ENST00000221293_at+ENST00000381251_
at+ENST00000304101_at+

ENST00000371761_at+ENST00000217246_at+ENST00000262662_at+ENST00000356692_
at+ENST00000361625_at+

ENST00000253023_at+ENST00000367843_at+ENST00000367840_at+ENST00000367839_
at+ENST00000324705_at+

ENST00000312263_at+ENST00000298694_at+ENST00000298693_at+ENST00000354181_
at+ENST00000361379_at+

ENST00000358712_at+ENST00000336092_at+ENST00000382438_at+ENST00000226972_
at+ENST00000312524_at+

ENST00000372264_at+ENST00000362031_at+ENST00000355110_at+ENST00000332302_
at+ENST00000265395_at+

ENST00000376419_at+ENST00000376417_at+ENST00000277124_at+ENST00000381753_
at+ENST00000313860_at+

ENST00000383367_at+ENST00000318809_at+ENST00000308370_at+ENST00000383179_
at+ENST00000243562_at+

ENST00000252576_at+ENST00000381917_at+ENST00000375493_at+ENST00000204005_
at+ENST00000301788_at+

ENST00000361272_at+ENST00000361231_at+ENST00000298047_at+ENST00000342745_
at+ENST00000244926_at+

ENST00000356505_at+ENST00000323760_at+ENST00000219168_at+ENST00000371534_
at+ENST00000383362_at+

ENST00000383360_at+ENST00000383359_at+ENST00000370109_at+ENST00000383178_
at+ENST00000383177_at+

ENST00000370107_at+ENST00000371646_at+ENST00000223864_at+ENST00000371554_
at+ENST00000342173_at,
                                  data=trainData,     kernel="polynomial",
cost=32, degree=2, coef0=0.125, gamma=0.0004882812, types="response")
prediksi_with_SMOTE_Poly100<-
prediction(svmmod_100_Poly2_with_SMOTE,testData)
prediksi_no_SMOTE_Poly100<-prediction(svmmod_100_Poly2_no_SMOTE,testData)

#Klasifikasi dengan Perulangan Menggunakan Subset Gen Terbaik
      1 :Skema 70 30
      1 : Untuk subset gen terbaik SVM Linear

      1: Untuk 60 Fitur Teratas Kernel Linier Menggunakan SMOTE
     tProgressBar(min = 0, max = 1000, style = 3)
     .linear.iterasi <- NULL
```

```
for (j in 1:1000) {
  acc_cv<-NULL
  folds.smote<-createFolds(smote_train$y, 10, list=FALSE)
  for(i in 1:10){
    index=which(folds.smote==i)
    training_folds = smote_train[-index,]
    test_folds = smote_train[index,]
    svmmod_60_linear_with_SMOTE<-
svm(y~ENST00000371095_at+ENST00000371085_at+ENST00000371082_at+ENST000003
49036_at+ENST00000265620_at+

ENST00000263080_at+ENST00000313949_at+ENST00000338783_at+ENST00000369420_
at+ENST00000369417_at+

ENST00000306082_at+ENST00000381463_at+ENST00000361035_at+ENST00000371102_
at+ENST00000371101_at+

ENST00000274311_at+ENST00000325945_at+ENST00000371100_at+ENST00000371075_
at+ENST00000354359_at+

ENST00000306120_at+ENST00000306090_at+ENST00000375594_at+ENST00000360868_
at+ENST00000358855_at+

ENST00000381936_at+ENST00000359894_at+ENST00000300947_at+ENST00000390597_
at+ENST00000380991_at+

ENST00000380986_at+ENST00000265028_at+ENST00000359623_at+ENST00000323078_
at+ENST00000283646_at+

ENST00000180173_at+ENST00000244296_at+ENST00000221293_at+ENST00000381251_
at+ENST00000304101_at+

ENST00000371761_at+ENST00000217246_at+ENST00000262662_at+ENST00000356692_
at+ENST00000361625_at+

ENST00000253023_at+ENST00000367843_at+ENST00000367840_at+ENST00000367839_
at+ENST00000324705_at+

ENST00000312263_at+ENST00000298694_at+ENST00000298693_at+ENST00000354181_
at+ENST00000361379_at+

ENST00000358712_at+ENST00000336092_at+ENST00000382438_at+ENST00000226972_
at+ENST00000312524_at,
                                    data=training_folds,
kernel="linear", cost=8, types="response")
    pred<-predict(svmmod_60_linear_with_SMOTE,   newdata   =   testData,
type="response")
    y_act<-testData$y
    #Menghitung Akurasi
    acc_cv[i] = mean(pred==y_act)
  }
  accuracy.linear = mean(as.numeric(acc_cv))
  accuracy.linear.iterasi[j]=mean(as.numeric(accuracy.linear))
  accuracy.linear.result=mean(as.numeric(accuracy.linear.iterasi))
  setTxtProgressBar(pb,j)
}
accuracy.linear.result
```

```
 1 : Untuk 70 Fitur Teratas Kernel Linier Menggunakan SMOTE
tProgressBar(min = 0, max = 1000, style = 3)
.linear.iterasi <- NULL
ɲ 1:1000) {
<-NULL
```

```
  folds.smote<-createFolds(smote_train$y, 10, list=FALSE)
  for(i in 1:10){
    index=which(folds.smote==i)
    training_folds = smote_train[-index,]
    test_folds = smote_train[index,]
    svmmod_70_linear_with_SMOTE<-
svm(y~ENST00000371095_at+ENST00000371085_at+ENST00000371082_at+ENST000003
49036_at+ENST00000265620_at+

ENST00000263080_at+ENST00000313949_at+ENST00000338783_at+ENST00000369420_
at+ENST00000369417_at+

ENST00000306082_at+ENST00000381463_at+ENST00000361035_at+ENST00000371102_
at+ENST00000371101_at+

ENST00000274311_at+ENST00000325945_at+ENST00000371100_at+ENST00000371075_
at+ENST00000354359_at+

ENST00000306120_at+ENST00000306090_at+ENST00000375594_at+ENST00000360868_
at+ENST00000358855_at+

ENST00000381936_at+ENST00000359894_at+ENST00000300947_at+ENST00000390597_
at+ENST00000380991_at+

ENST00000380986_at+ENST00000265028_at+ENST00000359623_at+ENST00000323078_
at+ENST00000283646_at+

ENST00000180173_at+ENST00000244296_at+ENST00000221293_at+ENST00000381251_
at+ENST00000304101_at+

ENST00000371761_at+ENST00000217246_at+ENST00000262662_at+ENST00000356692_
at+ENST00000361625_at+

ENST00000253023_at+ENST00000367843_at+ENST00000367840_at+ENST00000367839_
at+ENST00000324705_at+

ENST00000312263_at+ENST00000298694_at+ENST00000298693_at+ENST00000354181_
at+ENST00000361379_at+

ENST00000358712_at+ENST00000336092_at+ENST00000382438_at+ENST00000226972_
at+ENST00000312524_at+

ENST00000361272_at+ENST00000361231_at+ENST00000298047_at+ENST00000342745_
at+ENST00000244926_at+

ENST00000356505_at+ENST00000323760_at+ENST00000219168_at+ENST00000371534_
at+ENST00000383362_at,
                                    data=training_folds,
kernel="linear", cost=8,types="C-Classification")
    pred<-predict(svmmod_70_linear_with_SMOTE,   newdata   =   testData,
type="response")
    y_act<-testData$y
    #Menghitung Akurasi
    acc_cv[i] = mean(pred==y_act)
  }
  accuracy.linear = mean(as.numeric(acc_cv))
  accuracy.linear.iterasi[j]=mean(as.numeric(accuracy.linear))
  accuracy.linear.result=mean(as.numeric(accuracy.linear.iterasi))
       ProgressBar(pb,j)

       .linear.result

        1 : Untuk 60 Fitur Teratas Kernel Linier Tanpa Menggunakan SMOTE
       tProgressBar(min = 0, max = 1000, style = 3)
```

```
accuracy.linear.iterasi <- NULL
for (j in 1:1000) {
  acc_cv<-NULL
  folds.train<-createFolds(trainData$y, 10, list=FALSE)
  for(i in 1:10){
    index=which(folds.train==i)
    training_folds = trainData[-index,]
    test_folds = trainData[index,]
    svmmod_60_linear_no_SMOTE<-
svm(y~ENST00000371095_at+ENST00000371085_at+ENST00000371082_at+ENST000003
49036_at+ENST00000265620_at+

ENST00000263080_at+ENST00000313949_at+ENST00000338783_at+ENST00000369420_
at+ENST00000369417_at+

ENST00000306082_at+ENST00000381463_at+ENST00000361035_at+ENST00000371102_
at+ENST00000371101_at+

ENST00000274311_at+ENST00000325945_at+ENST00000371100_at+ENST00000371075_
at+ENST00000354359_at+

ENST00000306120_at+ENST00000306090_at+ENST00000375594_at+ENST00000360868_
at+ENST00000358855_at+

ENST00000381936_at+ENST00000359894_at+ENST00000300947_at+ENST00000390597_
at+ENST00000380991_at+

ENST00000380986_at+ENST00000265028_at+ENST00000359623_at+ENST00000323078_
at+ENST00000283646_at+

ENST00000180173_at+ENST00000244296_at+ENST00000221293_at+ENST00000381251_
at+ENST00000304101_at+

ENST00000371761_at+ENST00000217246_at+ENST00000262662_at+ENST00000356692_
at+ENST00000361625_at+

ENST00000253023_at+ENST00000367843_at+ENST00000367840_at+ENST00000367839_
at+ENST00000324705_at+

ENST00000312263_at+ENST00000298694_at+ENST00000298693_at+ENST00000354181_
at+ENST00000361379_at+

ENST00000358712_at+ENST00000336092_at+ENST00000382438_at+ENST00000226972_
at+ENST00000312524_at,
                                    data=training_folds,  kernel="linear",
cost=8, types="response")
    pred<-predict(svmmod_60_linear_no_SMOTE,    newdata    =    testData,
type="response")
    y_act<-testData$y
    #Menghitung Akurasi
    acc_cv[i] = mean(pred==y_act)
  }
  accuracy.linear = mean(as.numeric(acc_cv))
  accuracy.linear.iterasi[j]=mean(as.numeric(accuracy.linear))
  accuracy.linear.result=mean(as.numeric(accuracy.linear.iterasi))
  setTxtProgressBar(pb,j)
}
accuracy.linear.result
```

```
 1 : Untuk 70 Fitur Teratas Kernel Linier tanpa Menggunakan SMOTE
tProgressBar(min = 0, max = 1000, style = 3)
.linear.iterasi <- NULL
ɲ 1:1000) {
<-NULL
```

```
  folds.train<-createFolds(trainData$y, 10, list=FALSE)
  for(i in 1:10){
    index=which(folds.train==i)
    training_folds = trainData[-index,]
    test_folds = trainData[index,]
    svmmod_70_linear_no_SMOTE<-
svm(y~ENST00000371095_at+ENST00000371085_at+ENST00000371082_at+ENST000003
49036_at+ENST00000265620_at+

ENST00000263080_at+ENST00000313949_at+ENST00000338783_at+ENST00000369420_
at+ENST00000369417_at+

ENST00000306082_at+ENST00000381463_at+ENST00000361035_at+ENST00000371102_
at+ENST00000371101_at+

ENST00000274311_at+ENST00000325945_at+ENST00000371100_at+ENST00000371075_
at+ENST00000354359_at+

ENST00000306120_at+ENST00000306090_at+ENST00000375594_at+ENST00000360868_
at+ENST00000358855_at+

ENST00000381936_at+ENST00000359894_at+ENST00000300947_at+ENST00000390597_
at+ENST00000380991_at+

ENST00000380986_at+ENST00000265028_at+ENST00000359623_at+ENST00000323078_
at+ENST00000283646_at+

ENST00000180173_at+ENST00000244296_at+ENST00000221293_at+ENST00000381251_
at+ENST00000304101_at+

ENST00000371761_at+ENST00000217246_at+ENST00000262662_at+ENST00000356692_
at+ENST00000361625_at+

ENST00000253023_at+ENST00000367843_at+ENST00000367840_at+ENST00000367839_
at+ENST00000324705_at+

ENST00000312263_at+ENST00000298694_at+ENST00000298693_at+ENST00000354181_
at+ENST00000361379_at+

ENST00000358712_at+ENST00000336092_at+ENST00000382438_at+ENST00000226972_
at+ENST00000312524_at+

ENST00000361272_at+ENST00000361231_at+ENST00000298047_at+ENST00000342745_
at+ENST00000244926_at+

ENST00000356505_at+ENST00000323760_at+ENST00000219168_at+ENST00000371534_
at+ENST00000383362_at,
                                  data=training_folds,  kernel="linear",
cost=8,types="C-Classification")
    pred<-predict(svmmod_70_linear_no_SMOTE,    newdata    =    testData,
type="response")
    y_act<-testData$y
    #Menghitung Akurasi
    acc_cv[i] = mean(pred==y_act)
  }
  accuracy.linear = mean(as.numeric(acc_cv))
  accuracy.linear.iterasi[j]=mean(as.numeric(accuracy.linear))
  accuracy.linear.result=mean(as.numeric(accuracy.linear.iterasi))
       ProgressBar(pb,j)

       .linear.result


       ubset gen terbaik SVM RBF
```

```
#Dataset 1 : Untuk 60 Fitur teratas Kernel RBF Menggunakan SMOTE
accuracy.RBF.iterasi <- NULL
for (j in 1:1000) {
  acc_cv<-NULL
  folds.smote<-createFolds(smote_train$y, 10, list=FALSE)
  for(i in 1:10){
    index=which(folds.smote==i)
    training_folds = smote_train[-index,]
    test_folds = smote_train[index,]
    svmmod_60_RBF_with_SMOTE<-
svm(y~ENST00000371095_at+ENST00000371085_at+ENST00000371082_at+ENST000003
49036_at+ENST00000265620_at+

ENST00000263080_at+ENST00000313949_at+ENST00000338783_at+ENST00000369420_
at+ENST00000369417_at+

ENST00000306082_at+ENST00000381463_at+ENST00000361035_at+ENST00000371102_
at+ENST00000371101_at+

ENST00000274311_at+ENST00000325945_at+ENST00000371100_at+ENST00000371075_
at+ENST00000354359_at+

ENST00000306120_at+ENST00000306090_at+ENST00000375594_at+ENST00000360868_
at+ENST00000358855_at+

ENST00000381936_at+ENST00000359894_at+ENST00000300947_at+ENST00000390597_
at+ENST00000380991_at+

ENST00000380986_at+ENST00000265028_at+ENST00000359623_at+ENST00000323078_
at+ENST00000283646_at+

ENST00000180173_at+ENST00000244296_at+ENST00000221293_at+ENST00000381251_
at+ENST00000304101_at+

ENST00000371761_at+ENST00000217246_at+ENST00000262662_at+ENST00000356692_
at+ENST00000361625_at+

ENST00000253023_at+ENST00000367843_at+ENST00000367840_at+ENST00000367839_
at+ENST00000324705_at+

ENST00000312263_at+ENST00000298694_at+ENST00000298693_at+ENST00000354181_
at+ENST00000361379_at+

ENST00000358712_at+ENST00000336092_at+ENST00000382438_at+ENST00000226972_
at+ENST00000312524_at,
                        data=training_folds,   kernel="radial",   cost=64,
gamma=0.001953125, types="response")
    pred<-predict(svmmod_60_RBF_with_SMOTE,    newdata    =    testData,
type="response")
    y_act<-testData$y
    #Menghitung Akurasi
    acc_cv[i] = mean(pred==y_act)
  }
  accuracy.RBF = mean(as.numeric(acc_cv))
  accuracy.RBF.iterasi[j]=mean(as.numeric(accuracy.RBF))
  accuracy.RBF.result=mean(as.numeric(accuracy.RBF.iterasi))
  setTxtProgressBar(pb,j)
}
```

```
.RBF.result

 1 : Untuk 70 Fitur teratas Kernel RBF Menggunakan SMOTE
.RBF.iterasi <- NULL
ɲ 1:1000) {
<-NULL
```

```
  folds.smote<-createFolds(smote_train$y, 10, list=FALSE)
  for(i in 1:10){
    index=which(folds.smote==i)
    training_folds = smote_train[-index,]
    test_folds = smote_train[index,]
    svmmod_70_RBF_with_SMOTE<-
svm(y~ENST00000371095_at+ENST00000371085_at+ENST00000371082_at+ENST000003
49036_at+ENST00000265620_at+

ENST00000263080_at+ENST00000313949_at+ENST00000338783_at+ENST00000369420_
at+ENST00000369417_at+

ENST00000306082_at+ENST00000381463_at+ENST00000361035_at+ENST00000371102_
at+ENST00000371101_at+

ENST00000274311_at+ENST00000325945_at+ENST00000371100_at+ENST00000371075_
at+ENST00000354359_at+

ENST00000306120_at+ENST00000306090_at+ENST00000375594_at+ENST00000360868_
at+ENST00000358855_at+

ENST00000381936_at+ENST00000359894_at+ENST00000300947_at+ENST00000390597_
at+ENST00000380991_at+

ENST00000380986_at+ENST00000265028_at+ENST00000359623_at+ENST00000323078_
at+ENST00000283646_at+

ENST00000180173_at+ENST00000244296_at+ENST00000221293_at+ENST00000381251_
at+ENST00000304101_at+

ENST00000371761_at+ENST00000217246_at+ENST00000262662_at+ENST00000356692_
at+ENST00000361625_at+

ENST00000253023_at+ENST00000367843_at+ENST00000367840_at+ENST00000367839_
at+ENST00000324705_at+

ENST00000312263_at+ENST00000298694_at+ENST00000298693_at+ENST00000354181_
at+ENST00000361379_at+

ENST00000358712_at+ENST00000336092_at+ENST00000382438_at+ENST00000226972_
at+ENST00000312524_at+

ENST00000361272_at+ENST00000361231_at+ENST00000298047_at+ENST00000342745_
at+ENST00000244926_at+

ENST00000356505_at+ENST00000323760_at+ENST00000219168_at+ENST00000371534_
at+ENST00000383362_at,
                      data=training_folds,   kernel="radial",   cost=64,
gamma=0.001953125, types="response")
    pred<-predict(svmmod_70_RBF_with_SMOTE,    newdata    =    testData,
type="response")
    y_act<-testData$y
    #Menghitung Akurasi
    acc_cv[i] = mean(pred==y_act)
  }
  accuracy.RBF = mean(as.numeric(acc_cv))
  accuracy.RBF.iterasi[j]=mean(as.numeric(accuracy.RBF))
  accuracy.RBF.result=mean(as.numeric(accuracy.RBF.iterasi))
      ProgressBar(pb,j)

      .RBF.result

       1 : Untuk 40 Fitur teratas Kernel RBF tanpa Menggunakan SMOTE
      .RBF.iterasi <- NULL
```

```
for (j in 1:1000) {
  acc_cv<-NULL
  folds.train<-createFolds(trainData$y, 10, list=FALSE)
  for(i in 1:10){
    index=which(folds.smote==i)
    training_folds = trainData[-index,]
    test_folds = trainData[index,]
    svmmod_40_RBF_no_SMOTE<-
svm(y~ENST00000371095_at+ENST00000371085_at+ENST00000371082_at+ENST000003
49036_at+ENST00000265620_at+

ENST00000263080_at+ENST00000313949_at+ENST00000338783_at+ENST00000369420_
at+ENST00000369417_at+

ENST00000306082_at+ENST00000381463_at+ENST00000361035_at+ENST00000371102_
at+ENST00000371101_at+

ENST00000274311_at+ENST00000325945_at+ENST00000371100_at+ENST00000371075_
at+ENST00000354359_at+

ENST00000306120_at+ENST00000306090_at+ENST00000375594_at+ENST00000360868_
at+ENST00000358855_at+

ENST00000381936_at+ENST00000359894_at+ENST00000300947_at+ENST00000390597_
at+ENST00000380991_at+

ENST00000380986_at+ENST00000265028_at+ENST00000359623_at+ENST00000323078_
at+ENST00000283646_at+

ENST00000180173_at+ENST00000244296_at+ENST00000221293_at+ENST00000381251_
at+ENST00000304101_at,
                                  data=training_folds,     kernel="radial",
cost=64, gamma=0.001953125, types="response")
    pred<-predict(svmmod_40_RBF_no_SMOTE,     newdata     =     testData,
type="response")
    y_act<-testData$y
    #Menghitung Akurasi
    acc_cv[i] = mean(pred==y_act)
  }
  accuracy.RBF = mean(as.numeric(acc_cv))
  accuracy.RBF.iterasi[j]=mean(as.numeric(accuracy.RBF))
  accuracy.RBF.result=mean(as.numeric(accuracy.RBF.iterasi))
  setTxtProgressBar(pb,j)
}
accuracy.RBF.result

#Dataset 1 : Untuk 60 Fitur teratas Kernel RBF tanpa Menggunakan SMOTE
accuracy.RBF.iterasi <- NULL
for (j in 1:1000) {
  acc_cv<-NULL
  folds.train<-createFolds(trainData$y, 10, list=FALSE)
  for(i in 1:10){
    index=which(folds.smote==i)
    training_folds = trainData[-index,]
    test_folds = trainData[index,]
    svmmod_60_RBF_no_SMOTE<-
svm(y~ENST00000371095_at+ENST00000371085_at+ENST00000371082_at+ENST000003
49036_at+ENST00000265620_at+

    )263080_at+ENST00000313949_at+ENST00000338783_at+ENST00000369420_
    )000369417_at+

    )306082_at+ENST00000381463_at+ENST00000361035_at+ENST00000371102_
    )000371101_at+
```

```
ENST00000274311_at+ENST00000325945_at+ENST00000371100_at+ENST00000371075_
at+ENST00000354359_at+

ENST00000306120_at+ENST00000306090_at+ENST00000375594_at+ENST00000360868_
at+ENST00000358855_at+

ENST00000381936_at+ENST00000359894_at+ENST00000300947_at+ENST00000390597_
at+ENST00000380991_at+

ENST00000380986_at+ENST00000265028_at+ENST00000359623_at+ENST00000323078_
at+ENST00000283646_at+

ENST00000180173_at+ENST00000244296_at+ENST00000221293_at+ENST00000381251_
at+ENST00000304101_at+

ENST00000371761_at+ENST00000217246_at+ENST00000262662_at+ENST00000356692_
at+ENST00000361625_at+

ENST00000253023_at+ENST00000367843_at+ENST00000367840_at+ENST00000367839_
at+ENST00000324705_at+

ENST00000312263_at+ENST00000298694_at+ENST00000298693_at+ENST00000354181_
at+ENST00000361379_at+

ENST00000358712_at+ENST00000336092_at+ENST00000382438_at+ENST00000226972_
at+ENST00000312524_at,
                                    data=training_folds,   kernel="radial",
cost=64, gamma=0.001953125, types="response")
    pred<-predict(svmmod_60_RBF_no_SMOTE,     newdata     =     testData,
type="response")
    y_act<-testData$y
    #Menghitung Akurasi
    acc_cv[i] = mean(pred==y_act)
  }
  accuracy.RBF = mean(as.numeric(acc_cv))
  accuracy.RBF.iterasi[j]=mean(as.numeric(accuracy.RBF))
  accuracy.RBF.result=mean(as.numeric(accuracy.RBF.iterasi))
  setTxtProgressBar(pb,j)
}
accuracy.RBF.result

#Dataset 1 : Untuk 70 Fitur teratas Kernel RBF tanpa Menggunakan SMOTE
accuracy.RBF.iterasi <- NULL
for (j in 1:1000) {
  acc_cv<-NULL
  folds.train<-createFolds(trainData$y, 10, list=FALSE)
  for(i in 1:10){
    index=which(folds.train==i)
    training_folds = trainData[-index,]
    test_folds = trainData[index,]
    svmmod_70_RBF_no_SMOTE<-
svm(y~ENST00000371095_at+ENST00000371085_at+ENST00000371082_at+ENST000003
49036_at+ENST00000265620_at+

ENST00000263080_at+ENST00000313949_at+ENST00000338783_at+ENST00000369420_
at+ENST00000369417_at+

                                    )306082_at+ENST00000381463_at+ENST00000361035_at+ENST00000371102_
                                    )000371101_at+

                                    )274311_at+ENST00000325945_at+ENST00000371100_at+ENST00000371075_
                                    )000354359_at+
```

```
ENST00000306120_at+ENST00000306090_at+ENST00000375594_at+ENST00000360868_
at+ENST00000358855_at+

ENST00000381936_at+ENST00000359894_at+ENST00000300947_at+ENST00000390597_
at+ENST00000380991_at+

ENST00000380986_at+ENST00000265028_at+ENST00000359623_at+ENST00000323078_
at+ENST00000283646_at+

ENST00000180173_at+ENST00000244296_at+ENST00000221293_at+ENST00000381251_
at+ENST00000304101_at+

ENST00000371761_at+ENST00000217246_at+ENST00000262662_at+ENST00000356692_
at+ENST00000361625_at+

ENST00000253023_at+ENST00000367843_at+ENST00000367840_at+ENST00000367839_
at+ENST00000324705_at+

ENST00000312263_at+ENST00000298694_at+ENST00000298693_at+ENST00000354181_
at+ENST00000361379_at+

ENST00000358712_at+ENST00000336092_at+ENST00000382438_at+ENST00000226972_
at+ENST00000312524_at+

ENST00000361272_at+ENST00000361231_at+ENST00000298047_at+ENST00000342745_
at+ENST00000244926_at+

ENST00000356505_at+ENST00000323760_at+ENST00000219168_at+ENST00000371534_
at+ENST00000383362_at,
                                    data=training_folds,   kernel="radial",
cost=64, gamma=0.001953125, types="response")
    pred<-predict(svmmod_70_RBF_no_SMOTE,     newdata     =     testData,
type="response")
    y_act<-testData$y
    #Menghitung Akurasi
    acc_cv[i] = mean(pred==y_act)
  }
  accuracy.RBF = mean(as.numeric(acc_cv))
  accuracy.RBF.iterasi[j]=mean(as.numeric(accuracy.RBF))
  accuracy.RBF.result=mean(as.numeric(accuracy.RBF.iterasi))
  setTxtProgressBar(pb,j)
}
accuracy.RBF.result

#Dataset 1 : Untuk 80 Fitur teratas Kernel RBF tanpa Menggunakan SMOTE
accuracy.RBF.iterasi <- NULL
for (j in 1:1000) {
  acc_cv<-NULL
  folds.train<-createFolds(trainData$y, 10, list=FALSE)
  for(i in 1:10){
    index=which(folds.train==i)
    training_folds = trainData[-index,]
    test_folds = trainData[index,]
    svmmod_80_RBF_no_SMOTE<-
svm(y~ENST00000371095_at+ENST00000371085_at+ENST00000371082_at+ENST000003
49036_at+ENST00000265620_at+

    )263080_at+ENST00000313949_at+ENST00000338783_at+ENST00000369420_
    )000369417_at+

    )306082_at+ENST00000381463_at+ENST00000361035_at+ENST00000371102_
    )000371101_at+
```

```
ENST00000274311_at+ENST00000325945_at+ENST00000371100_at+ENST00000371075_
at+ENST00000354359_at+

ENST00000306120_at+ENST00000306090_at+ENST00000375594_at+ENST00000360868_
at+ENST00000358855_at+

ENST00000381936_at+ENST00000359894_at+ENST00000300947_at+ENST00000390597_
at+ENST00000380991_at+

ENST00000380986_at+ENST00000265028_at+ENST00000359623_at+ENST00000323078_
at+ENST00000283646_at+

ENST00000180173_at+ENST00000244296_at+ENST00000221293_at+ENST00000381251_
at+ENST00000304101_at+

ENST00000371761_at+ENST00000217246_at+ENST00000262662_at+ENST00000356692_
at+ENST00000361625_at+

ENST00000253023_at+ENST00000367843_at+ENST00000367840_at+ENST00000367839_
at+ENST00000324705_at+

ENST00000312263_at+ENST00000298694_at+ENST00000298693_at+ENST00000354181_
at+ENST00000361379_at+

ENST00000358712_at+ENST00000336092_at+ENST00000382438_at+ENST00000226972_
at+ENST00000312524_at+

ENST00000372264_at+ENST00000362031_at+ENST00000355110_at+ENST00000332302_
at+ENST00000265395_at+

ENST00000376419_at+ENST00000376417_at+ENST00000277124_at+ENST00000381753_
at+ENST00000313860_at+

ENST00000383367_at+ENST00000318809_at+ENST00000308370_at+ENST00000383179_
at+ENST00000243562_at+

ENST00000252576_at+ENST00000381917_at+ENST00000375493_at+ENST00000204005_
at+ENST00000301788_at,
                                data=training_folds,     kernel="radial",
cost=64, gamma=0.001953125, types="response")
    pred<-predict(svmmod_80_RBF_no_SMOTE,     newdata     =     testData,
type="response")
    y_act<-testData$y
    #Menghitung Akurasi
    acc_cv[i] = mean(pred==y_act)
  }
  accuracy.RBF = mean(as.numeric(acc_cv))
  accuracy.RBF.iterasi[j]=mean(as.numeric(accuracy.RBF))
  accuracy.RBF.result=mean(as.numeric(accuracy.RBF.iterasi))
  setTxtProgressBar(pb,j)
}
accuracy.RBF.result


#Untuk subset gen dengan akurasi terbaik SVM Polynomial
#Dataset 1 : Untuk 40 Fitur Teratas Kernel Polynomial
accuracy.polynomial.iterasi <- NULL
        n 1:1000) {
        <-NULL
        smote<-createFolds(smote_train$y, 10, list=FALSE)
        in 1:10){
        x=which(folds.smote==i)
        ning_folds = smote_train[-index,]
```

```
    test_folds = smote_train[index,]
    svmmod_40_Poly2_with_SMOTE<-
svm(y~ENST00000371095_at+ENST00000371085_at+ENST00000371082_at+ENST000003
49036_at+ENST00000265620_at+

ENST00000263080_at+ENST00000313949_at+ENST00000338783_at+ENST00000369420_
at+ENST00000369417_at+

ENST00000306082_at+ENST00000381463_at+ENST00000361035_at+ENST00000371102_
at+ENST00000371101_at+

ENST00000274311_at+ENST00000325945_at+ENST00000371100_at+ENST00000371075_
at+ENST00000354359_at+

ENST00000306120_at+ENST00000306090_at+ENST00000375594_at+ENST00000360868_
at+ENST00000358855_at+

ENST00000381936_at+ENST00000359894_at+ENST00000300947_at+ENST00000390597_
at+ENST00000380991_at+

ENST00000380986_at+ENST00000265028_at+ENST00000359623_at+ENST00000323078_
at+ENST00000283646_at+

ENST00000180173_at+ENST00000244296_at+ENST00000221293_at+ENST00000381251_
at+ENST00000304101_at,
                        data=training_folds,        kernel="polynomial",
cost=32,  degree=2,  coef0=0.125,  gamma=0.0004882812,  types="response",
trControl = train.control)
    pred<-predict(svmmod_40_Poly2_with_SMOTE,    newdata    =    testData,
type="response")
    y_act<-testData$y
    #Menghitung Akurasi
    acc_cv[i] = mean(pred==y_act)
  }
  accuracy.polynomial = mean(as.numeric(acc_cv))
  accuracy.polynomial.iterasi[j]=mean(as.numeric(accuracy.polynomial))

accuracy.polynomial.result=mean(as.numeric(accuracy.polynomial.iterasi))
  setTxtProgressBar(pb,j)
}
accuracy.polynomial.result

#Dataset 1 : Untuk 70 Fitur Teratas Kernel Polynomial Menggunakan SMOTE
accuracy.polynomial.iterasi <- NULL
for (j in 1:1000) {
  acc_cv<-NULL
  folds.smote<-createFolds(smote_train$y, 10, list=FALSE)
  for(i in 1:10){
    index=which(folds.smote==i)
    training_folds = smote_train[-index,]
    test_folds = smote_train[index,]
    svmmod_70_Poly2_with_SMOTE<-
svm(y~ENST00000371095_at+ENST00000371085_at+ENST00000371082_at+ENST000003
49036_at+ENST00000265620_at+

ENST00000263080_at+ENST00000313949_at+ENST00000338783_at+ENST00000369420_
at+ENST00000369417_at+

)306082_at+ENST00000381463_at+ENST00000361035_at+ENST00000371102_
)000371101_at+

)274311_at+ENST00000325945_at+ENST00000371100_at+ENST00000371075_
)000354359_at+
```

```
ENST00000306120_at+ENST00000306090_at+ENST00000375594_at+ENST00000360868_
at+ENST00000358855_at+

ENST00000381936_at+ENST00000359894_at+ENST00000300947_at+ENST00000390597_
at+ENST00000380991_at+

ENST00000380986_at+ENST00000265028_at+ENST00000359623_at+ENST00000323078_
at+ENST00000283646_at+

ENST00000180173_at+ENST00000244296_at+ENST00000221293_at+ENST00000381251_
at+ENST00000304101_at+

ENST00000371761_at+ENST00000217246_at+ENST00000262662_at+ENST00000356692_
at+ENST00000361625_at+

ENST00000253023_at+ENST00000367843_at+ENST00000367840_at+ENST00000367839_
at+ENST00000324705_at+

ENST00000312263_at+ENST00000298694_at+ENST00000298693_at+ENST00000354181_
at+ENST00000361379_at+

ENST00000358712_at+ENST00000336092_at+ENST00000382438_at+ENST00000226972_
at+ENST00000312524_at,
                        data=training_folds,        kernel="polynomial",
cost=32, degree=2, coef0=0.125, gamma=0.0004882812, types="response")
    pred<-predict(svmmod_70_Poly2_with_SMOTE,    newdata    =    testData,
type="response")
    y_act<-testData$y
    #Menghitung Akurasi
    acc_cv[i] = mean(pred==y_act)
  }
  accuracy.polynomial = mean(as.numeric(acc_cv))
  accuracy.polynomial.iterasi[j]=mean(as.numeric(accuracy.polynomial))

accuracy.polynomial.result=mean(as.numeric(accuracy.polynomial.iterasi))
  setTxtProgressBar(pb,j)
}
accuracy.polynomial.result

#Dataset 1 : Untuk 80 Fitur Teratas Kernel Polynomial Tanpa Menggunakan
SMOTE
accuracy.polynomial.iterasi <- NULL
for (j in 1:1000) {
  acc_cv<-NULL
  folds.train<-createFolds(trainData$y, 10, list=FALSE)
  for(i in 1:10){
    index=which(folds.train==i)
    training_folds = trainData[-index,]
    test_folds = trainData[index,]
    svmmod_80_Poly2_with_SMOTE<-
svm(y~ENST00000371095_at+ENST00000371085_at+ENST00000371082_at+ENST000003
49036_at+ENST00000265620_at+

ENST00000263080_at+ENST00000313949_at+ENST00000338783_at+ENST00000369420_
at+ENST00000369417_at+

ENST00000306082_at+ENST00000381463_at+ENST00000361035_at+ENST00000371102_
          0000371101_at+

          0274311_at+ENST00000325945_at+ENST00000371100_at+ENST00000371075_
          0000354359_at+
```

```
ENST00000306120_at+ENST00000306090_at+ENST00000375594_at+ENST00000360868_
at+ENST00000358855_at+

ENST00000381936_at+ENST00000359894_at+ENST00000300947_at+ENST00000390597_
at+ENST00000380991_at+

ENST00000380986_at+ENST00000265028_at+ENST00000359623_at+ENST00000323078_
at+ENST00000283646_at+

ENST00000180173_at+ENST00000244296_at+ENST00000221293_at+ENST00000381251_
at+ENST00000304101_at+

ENST00000371761_at+ENST00000217246_at+ENST00000262662_at+ENST00000356692_
at+ENST00000361625_at+

ENST00000253023_at+ENST00000367843_at+ENST00000367840_at+ENST00000367839_
at+ENST00000324705_at+

ENST00000312263_at+ENST00000298694_at+ENST00000298693_at+ENST00000354181_
at+ENST00000361379_at+

ENST00000358712_at+ENST00000336092_at+ENST00000382438_at+ENST00000226972_
at+ENST00000312524_at+

ENST00000372264_at+ENST00000362031_at+ENST00000355110_at+ENST00000332302_
at+ENST00000265395_at+

ENST00000376419_at+ENST00000376417_at+ENST00000277124_at+ENST00000381753_
at+ENST00000313860_at+

ENST00000383367_at+ENST00000318809_at+ENST00000308370_at+ENST00000383179_
at+ENST00000243562_at+

ENST00000252576_at+ENST00000381917_at+ENST00000375493_at+ENST00000204005_
at+ENST00000301788_at,
                          data=training_folds,        kernel="polynomial",
cost=32, degree=2, coef0=0.125, gamma=0.0004882812, types="response")
    pred<-predict(svmmod_80_Poly2_no_SMOTE,    newdata    =    testData,
type="response")
    y_act<-testData$y
    #Menghitung Akurasi
    acc_cv[i] = mean(pred==y_act)
  }
  accuracy.polynomial = mean(as.numeric(acc_cv))
  accuracy.polynomial.iterasi[j]=mean(as.numeric(accuracy.polynomial))

accuracy.polynomial.result=mean(as.numeric(accuracy.polynomial.iterasi))
  setTxtProgressBar(pb,j)
}
accuracy.polynomial.result
```

Lampiran 15 : Source Code Implementasi Klasifikasi SVM menggunakan 3 Buah Kernel dan Evaluasi Performa untuk Dataset 2 (80 20)

```
#Skema 80 Training Set 20 Test Set
#SVMModel
      2 : Kernel Linear
      2 : Kernel Linear dengan 10 Fitur Teratas
)_linear_with_SMOTE<-
ST00000371095_at+ENST00000371085_at+ENST00000371082_at+ENST000003
+ENST00000265620_at+
```

```
ENST00000263080_at+ENST00000313949_at+ENST00000338783_at+ENST00000369420_
at+ENST00000369417_at,
                      data=smote_train,      kernel="linear",      cost=1,
types="response")
svmmod_10_linear_no_SMOTE<-
svm(y~ENST00000371095_at+ENST00000371085_at+ENST00000371082_at+ENST000003
49036_at+ENST00000265620_at+

ENST00000263080_at+ENST00000313949_at+ENST00000338783_at+ENST00000369420_
at+ENST00000369417_at,
                      data=trainData,       kernel="linear",       cost=1,
types="response")
prediksi_with_SMOTE_linear10<-
prediction(svmmod_10_linear_with_SMOTE,testData)
prediksi_no_SMOTE_linear10<-
prediction(svmmod_10_linear_no_SMOTE,testData)

#Dataset 2 : Kernel Linear dengan 20 Fitur Teratas
svmmod_20_linear_with_SMOTE<-
svm(y~ENST00000371095_at+ENST00000371085_at+ENST00000371082_at+ENST000003
49036_at+ENST00000265620_at+

ENST00000263080_at+ENST00000313949_at+ENST00000338783_at+ENST00000369420_
at+ENST00000369417_at+

ENST00000306082_at+ENST00000381463_at+ENST00000361035_at+ENST00000371102_
at+ENST00000371101_at+

ENST00000274311_at+ENST00000325945_at+ENST00000371100_at+ENST00000371075_
at+ENST00000354359_at,
                      data=smote_train,      kernel="linear",      cost=1,
types="response")
svmmod_20_linear_no_SMOTE<-
svm(y~ENST00000371095_at+ENST00000371085_at+ENST00000371082_at+ENST000003
49036_at+ENST00000265620_at+

ENST00000263080_at+ENST00000313949_at+ENST00000338783_at+ENST00000369420_
at+ENST00000369417_at+

ENST00000306082_at+ENST00000381463_at+ENST00000361035_at+ENST00000371102_
at+ENST00000371101_at+

ENST00000274311_at+ENST00000325945_at+ENST00000371100_at+ENST00000371075_
at+ENST00000354359_at,
                      data=trainData,       kernel="linear",       cost=1,
types="response")
prediksi_with_SMOTE_linear20<-
prediction(svmmod_20_linear_with_SMOTE,testData)
prediksi_no_SMOTE_linear20<-
prediction(svmmod_20_linear_no_SMOTE,testData)

#Dataset 2 : Kernel linear dengan 30 Fitur Teratas
svmmod_30_linear_with_SMOTE<-
svm(y~ENST00000371095_at+ENST00000371085_at+ENST00000371082_at+ENST000003
49036_at+ENST00000265620_at+

ENST00000263080_at+ENST00000313949_at+ENST00000338783_at+ENST00000369420_
)000369417_at+

)306082_at+ENST00000381463_at+ENST00000361035_at+ENST00000371102_
)000371101_at+
```

```
ENST00000274311_at+ENST00000325945_at+ENST00000371100_at+ENST00000371075_
at+ENST00000354359_at+

ENST00000306120_at+ENST00000306090_at+ENST00000375594_at+ENST00000360868_
at+ENST00000358855_at+

ENST00000381936_at+ENST00000359894_at+ENST00000300947_at+ENST00000390597_
at+ENST00000380991_at,
                        data=smote_train,      kernel="linear",      cost=1,
types="response")
svmmod_30_linear_no_SMOTE<-
svm(y~ENST00000371095_at+ENST00000371085_at+ENST00000371082_at+ENST000003
49036_at+ENST00000265620_at+

ENST00000263080_at+ENST00000313949_at+ENST00000338783_at+ENST00000369420_
at+ENST00000369417_at+

ENST00000306082_at+ENST00000381463_at+ENST00000361035_at+ENST00000371102_
at+ENST00000371101_at+

ENST00000274311_at+ENST00000325945_at+ENST00000371100_at+ENST00000371075_
at+ENST00000354359_at+

ENST00000306120_at+ENST00000306090_at+ENST00000375594_at+ENST00000360868_
at+ENST00000358855_at+

ENST00000381936_at+ENST00000359894_at+ENST00000300947_at+ENST00000390597_
at+ENST00000380991_at,
                                    data=trainData, kernel="linear", cost=1,
types="response")
prediksi_with_SMOTE_linear30<-
prediction(svmmod_30_linear_with_SMOTE,testData)
prediksi_no_SMOTE_linear30<-
prediction(svmmod_30_linear_no_SMOTE,testData)

#Dataset 2 : Kernel Linear dengan 40 Fitur Teratas
svmmod_40_linear_with_SMOTE<-
svm(y~ENST00000371095_at+ENST00000371085_at+ENST00000371082_at+ENST000003
49036_at+ENST00000265620_at+

ENST00000263080_at+ENST00000313949_at+ENST00000338783_at+ENST00000369420_
at+ENST00000369417_at+

ENST00000306082_at+ENST00000381463_at+ENST00000361035_at+ENST00000371102_
at+ENST00000371101_at+

ENST00000274311_at+ENST00000325945_at+ENST00000371100_at+ENST00000371075_
at+ENST00000354359_at+

ENST00000306120_at+ENST00000306090_at+ENST00000375594_at+ENST00000360868_
at+ENST00000358855_at+

ENST00000381936_at+ENST00000359894_at+ENST00000300947_at+ENST00000390597_
at+ENST00000380991_at+

ENST00000380986_at+ENST00000265028_at+ENST00000359623_at+ENST00000323078_
at+ENST00000283646_at+

)0180173_at+ENST00000244296_at+ENST00000221293_at+ENST00000381251_
)0000304101_at,
                        data=smote_train,                    kernel="radial",
ypes="response")
```

```
svmmod_40_linear_no_SMOTE<-
svm(y~ENST00000371095_at+ENST00000371085_at+ENST00000371082_at+ENST000003
49036_at+ENST00000265620_at+

ENST00000263080_at+ENST00000313949_at+ENST00000338783_at+ENST00000369420_
at+ENST00000369417_at+

ENST00000306082_at+ENST00000381463_at+ENST00000361035_at+ENST00000371102_
at+ENST00000371101_at+

ENST00000274311_at+ENST00000325945_at+ENST00000371100_at+ENST00000371075_
at+ENST00000354359_at+

ENST00000306120_at+ENST00000306090_at+ENST00000375594_at+ENST00000360868_
at+ENST00000358855_at+

ENST00000381936_at+ENST00000359894_at+ENST00000300947_at+ENST00000390597_
at+ENST00000380991_at+

ENST00000380986_at+ENST00000265028_at+ENST00000359623_at+ENST00000323078_
at+ENST00000283646_at+

ENST00000180173_at+ENST00000244296_at+ENST00000221293_at+ENST00000381251_
at+ENST00000304101_at,
                                data=trainData,         kernel="radial",
cost=1,types="response")
prediksi_with_SMOTE_linear40<-
prediction(svmmod_40_linear_with_SMOTE,testData)
prediksi_no_SMOTE_linear40<-
prediction(svmmod_40_linear_no_SMOTE,testData)

#Dataset 2 : Kernel Linear dengan 50 Fitur Teratas
svmmod_50_linear_with_SMOTE<-
svm(y~ENST00000371095_at+ENST00000371085_at+ENST00000371082_at+ENST000003
49036_at+ENST00000265620_at+

ENST00000263080_at+ENST00000313949_at+ENST00000338783_at+ENST00000369420_
at+ENST00000369417_at+

ENST00000306082_at+ENST00000381463_at+ENST00000361035_at+ENST00000371102_
at+ENST00000371101_at+

ENST00000274311_at+ENST00000325945_at+ENST00000371100_at+ENST00000371075_
at+ENST00000354359_at+

ENST00000306120_at+ENST00000306090_at+ENST00000375594_at+ENST00000360868_
at+ENST00000358855_at+

ENST00000381936_at+ENST00000359894_at+ENST00000300947_at+ENST00000390597_
at+ENST00000380991_at+

ENST00000380986_at+ENST00000265028_at+ENST00000359623_at+ENST00000323078_
at+ENST00000283646_at+

ENST00000180173_at+ENST00000244296_at+ENST00000221293_at+ENST00000381251_
at+ENST00000304101_at+

ENST00000371761_at+ENST00000217246_at+ENST00000262662_at+ENST00000356692_
    )000361625_at+

    )253023_at+ENST00000367843_at+ENST00000367840_at+ENST00000367839_
    )000324705_at,
                data=smote_train,                   kernel="linear",
ypes="response")
```

```
svmmod_50_linear_no_SMOTE<-
svm(y~ENST00000371095_at+ENST00000371085_at+ENST00000371082_at+ENST000003
49036_at+ENST00000265620_at+

ENST00000263080_at+ENST00000313949_at+ENST00000338783_at+ENST00000369420_
at+ENST00000369417_at+

ENST00000306082_at+ENST00000381463_at+ENST00000361035_at+ENST00000371102_
at+ENST00000371101_at+

ENST00000274311_at+ENST00000325945_at+ENST00000371100_at+ENST00000371075_
at+ENST00000354359_at+

ENST00000306120_at+ENST00000306090_at+ENST00000375594_at+ENST00000360868_
at+ENST00000358855_at+

ENST00000381936_at+ENST00000359894_at+ENST00000300947_at+ENST00000390597_
at+ENST00000380991_at+

ENST00000380986_at+ENST00000265028_at+ENST00000359623_at+ENST00000323078_
at+ENST00000283646_at+

ENST00000180173_at+ENST00000244296_at+ENST00000221293_at+ENST00000381251_
at+ENST00000304101_at+

ENST00000371761_at+ENST00000217246_at+ENST00000262662_at+ENST00000356692_
at+ENST00000361625_at+

ENST00000253023_at+ENST00000367843_at+ENST00000367840_at+ENST00000367839_
at+ENST00000324705_at,
                    data=trainData,                    kernel="linear",
cost=1,types="response")
prediksi_with_SMOTE_linear50<-
prediction(svmmod_50_linear_with_SMOTE,testData)
prediksi_no_SMOTE_linear50<-
prediction(svmmod_50_linear_no_SMOTE,testData)

#Dataset 2 : Kernel Linear dengan 60 Fitur Teratas
svmmod_60_linear_with_SMOTE<-
svm(y~ENST00000371095_at+ENST00000371085_at+ENST00000371082_at+ENST000003
49036_at+ENST00000265620_at+

ENST00000263080_at+ENST00000313949_at+ENST00000338783_at+ENST00000369420_
at+ENST00000369417_at+

ENST00000306082_at+ENST00000381463_at+ENST00000361035_at+ENST00000371102_
at+ENST00000371101_at+

ENST00000274311_at+ENST00000325945_at+ENST00000371100_at+ENST00000371075_
at+ENST00000354359_at+

ENST00000306120_at+ENST00000306090_at+ENST00000375594_at+ENST00000360868_
at+ENST00000358855_at+

ENST00000381936_at+ENST00000359894_at+ENST00000300947_at+ENST00000390597_
at+ENST00000380991_at+

ENST00000380986_at+ENST00000265028_at+ENST00000359623_at+ENST00000323078_
      )000283646_at+

      )180173_at+ENST00000244296_at+ENST00000221293_at+ENST00000381251_
      )000304101_at+
```

```
ENST00000371761_at+ENST00000217246_at+ENST00000262662_at+ENST00000356692_
at+ENST00000361625_at+

ENST00000253023_at+ENST00000367843_at+ENST00000367840_at+ENST00000367839_
at+ENST00000324705_at+

ENST00000312263_at+ENST00000298694_at+ENST00000298693_at+ENST00000354181_
at+ENST00000361379_at+

ENST00000358712_at+ENST00000336092_at+ENST00000382438_at+ENST00000226972_
at+ENST00000312524_at,
                      data=smote_train,      kernel="linear",      cost=1,
types="response")
svmmod_60_linear_no_SMOTE<-
svm(y~ENST00000371095_at+ENST00000371085_at+ENST00000371082_at+ENST000003
49036_at+ENST00000265620_at+

ENST00000263080_at+ENST00000313949_at+ENST00000338783_at+ENST00000369420_
at+ENST00000369417_at+

ENST00000306082_at+ENST00000381463_at+ENST00000361035_at+ENST00000371102_
at+ENST00000371101_at+

ENST00000274311_at+ENST00000325945_at+ENST00000371100_at+ENST00000371075_
at+ENST00000354359_at+

ENST00000306120_at+ENST00000306090_at+ENST00000375594_at+ENST00000360868_
at+ENST00000358855_at+

ENST00000381936_at+ENST00000359894_at+ENST00000300947_at+ENST00000390597_
at+ENST00000380991_at+

ENST00000380986_at+ENST00000265028_at+ENST00000359623_at+ENST00000323078_
at+ENST00000283646_at+

ENST00000180173_at+ENST00000244296_at+ENST00000221293_at+ENST00000381251_
at+ENST00000304101_at+

ENST00000371761_at+ENST00000217246_at+ENST00000262662_at+ENST00000356692_
at+ENST00000361625_at+

ENST00000253023_at+ENST00000367843_at+ENST00000367840_at+ENST00000367839_
at+ENST00000324705_at+

ENST00000312263_at+ENST00000298694_at+ENST00000298693_at+ENST00000354181_
at+ENST00000361379_at+

ENST00000358712_at+ENST00000336092_at+ENST00000382438_at+ENST00000226972_
at+ENST00000312524_at,
                                  data=trainData, kernel="linear", cost=1,
types="response")
prediksi_with_SMOTE_linear60<-
prediction(svmmod_60_linear_with_SMOTE,testData)
prediksi_no_SMOTE_linear60<-
prediction(svmmod_60_linear_no_SMOTE,testData)

#Dataset 2 : Kernel Linear dengan 70 Fitur Teratas
      O_linear_with_SMOTE<-
      ST00000371095_at+ENST00000371085_at+ENST00000371082_at+ENST000003
      +ENST00000265620_at+

      O263080_at+ENST00000313949_at+ENST00000338783_at+ENST00000369420_
      O000369417_at+
```

```
ENST00000306082_at+ENST00000381463_at+ENST00000361035_at+ENST00000371102_
at+ENST00000371101_at+

ENST00000274311_at+ENST00000325945_at+ENST00000371100_at+ENST00000371075_
at+ENST00000354359_at+

ENST00000306120_at+ENST00000306090_at+ENST00000375594_at+ENST00000360868_
at+ENST00000358855_at+

ENST00000381936_at+ENST00000359894_at+ENST00000300947_at+ENST00000390597_
at+ENST00000380991_at+

ENST00000380986_at+ENST00000265028_at+ENST00000359623_at+ENST00000323078_
at+ENST00000283646_at+

ENST00000180173_at+ENST00000244296_at+ENST00000221293_at+ENST00000381251_
at+ENST00000304101_at+

ENST00000371761_at+ENST00000217246_at+ENST00000262662_at+ENST00000356692_
at+ENST00000361625_at+

ENST00000253023_at+ENST00000367843_at+ENST00000367840_at+ENST00000367839_
at+ENST00000324705_at+

ENST00000312263_at+ENST00000298694_at+ENST00000298693_at+ENST00000354181_
at+ENST00000361379_at+

ENST00000358712_at+ENST00000336092_at+ENST00000382438_at+ENST00000226972_
at+ENST00000312524_at+

ENST00000361272_at+ENST00000361231_at+ENST00000298047_at+ENST00000342745_
at+ENST00000244926_at+

ENST00000356505_at+ENST00000323760_at+ENST00000219168_at+ENST00000371534_
at+ENST00000383362_at,
                        data=smote_train, kernel="linear", cost=1,types="C-
Classification")
svmmod_70_linear_no_SMOTE<-
svm(y~ENST00000371095_at+ENST00000371085_at+ENST00000371082_at+ENST000003
49036_at+ENST00000265620_at+

ENST00000263080_at+ENST00000313949_at+ENST00000338783_at+ENST00000369420_
at+ENST00000369417_at+

ENST00000306082_at+ENST00000381463_at+ENST00000361035_at+ENST00000371102_
at+ENST00000371101_at+

ENST00000274311_at+ENST00000325945_at+ENST00000371100_at+ENST00000371075_
at+ENST00000354359_at+

ENST00000306120_at+ENST00000306090_at+ENST00000375594_at+ENST00000360868_
at+ENST00000358855_at+

ENST00000381936_at+ENST00000359894_at+ENST00000300947_at+ENST00000390597_
at+ENST00000380991_at+

ENST00000380986_at+ENST00000265028_at+ENST00000359623_at+ENST00000323078_
          )000283646_at+

          )0180173_at+ENST00000244296_at+ENST00000221293_at+ENST00000381251_
          )000304101_at+
```

```
ENST00000371761_at+ENST00000217246_at+ENST00000262662_at+ENST00000356692_
at+ENST00000361625_at+

ENST00000253023_at+ENST00000367843_at+ENST00000367840_at+ENST00000367839_
at+ENST00000324705_at+

ENST00000312263_at+ENST00000298694_at+ENST00000298693_at+ENST00000354181_
at+ENST00000361379_at+

ENST00000358712_at+ENST00000336092_at+ENST00000382438_at+ENST00000226972_
at+ENST00000312524_at+

ENST00000361272_at+ENST00000361231_at+ENST00000298047_at+ENST00000342745_
at+ENST00000244926_at+

ENST00000356505_at+ENST00000323760_at+ENST00000219168_at+ENST00000371534_
at+ENST00000383362_at,
                                  data=trainData,          kernel="linear",
cost=1,types="C-Classification")
prediksi_with_SMOTE_linear70<-
prediction(svmmod_70_linear_with_SMOTE,testData)
prediksi_no_SMOTE_linear70<-
prediction(svmmod_70_linear_no_SMOTE,testData)

#Dataset 2 : Kernel Linear dengan 80 Fitur Teratas
svmmod_80_linear_with_SMOTE<-
svm(y~ENST00000371095_at+ENST00000371085_at+ENST00000371082_at+ENST000003
49036_at+ENST00000265620_at+

ENST00000263080_at+ENST00000313949_at+ENST00000338783_at+ENST00000369420_
at+ENST00000369417_at+

ENST00000306082_at+ENST00000381463_at+ENST00000361035_at+ENST00000371102_
at+ENST00000371101_at+

ENST00000274311_at+ENST00000325945_at+ENST00000371100_at+ENST00000371075_
at+ENST00000354359_at+

ENST00000306120_at+ENST00000306090_at+ENST00000375594_at+ENST00000360868_
at+ENST00000358855_at+

ENST00000381936_at+ENST00000359894_at+ENST00000300947_at+ENST00000390597_
at+ENST00000380991_at+

ENST00000380986_at+ENST00000265028_at+ENST00000359623_at+ENST00000323078_
at+ENST00000283646_at+

ENST00000180173_at+ENST00000244296_at+ENST00000221293_at+ENST00000381251_
at+ENST00000304101_at+

ENST00000371761_at+ENST00000217246_at+ENST00000262662_at+ENST00000356692_
at+ENST00000361625_at+

ENST00000253023_at+ENST00000367843_at+ENST00000367840_at+ENST00000367839_
at+ENST00000324705_at+

ENST00000312263_at+ENST00000298694_at+ENST00000298693_at+ENST00000354181_
      )000361379_at+

      )358712_at+ENST00000336092_at+ENST00000382438_at+ENST00000226972_
      )000312524_at+
```

```
ENST00000372264_at+ENST00000362031_at+ENST00000355110_at+ENST00000332302_
at+ENST00000265395_at+

ENST00000376419_at+ENST00000376417_at+ENST00000277124_at+ENST00000381753_
at+ENST00000313860_at+

ENST00000383367_at+ENST00000318809_at+ENST00000308370_at+ENST00000383179_
at+ENST00000243562_at+

ENST00000252576_at+ENST00000381917_at+ENST00000375493_at+ENST00000204005_
at+ENST00000301788_at,
                      data=smote_train,      kernel="linear",      cost=1,
types="response")
svmmod_80_linear_no_SMOTE<-
svm(y~ENST00000371095_at+ENST00000371085_at+ENST00000371082_at+ENST000003
49036_at+ENST00000265620_at+

ENST00000263080_at+ENST00000313949_at+ENST00000338783_at+ENST00000369420_
at+ENST00000369417_at+

ENST00000306082_at+ENST00000381463_at+ENST00000361035_at+ENST00000371102_
at+ENST00000371101_at+

ENST00000274311_at+ENST00000325945_at+ENST00000371100_at+ENST00000371075_
at+ENST00000354359_at+

ENST00000306120_at+ENST00000306090_at+ENST00000375594_at+ENST00000360868_
at+ENST00000358855_at+

ENST00000381936_at+ENST00000359894_at+ENST00000300947_at+ENST00000390597_
at+ENST00000380991_at+

ENST00000380986_at+ENST00000265028_at+ENST00000359623_at+ENST00000323078_
at+ENST00000283646_at+

ENST00000180173_at+ENST00000244296_at+ENST00000221293_at+ENST00000381251_
at+ENST00000304101_at+

ENST00000371761_at+ENST00000217246_at+ENST00000262662_at+ENST00000356692_
at+ENST00000361625_at+

ENST00000253023_at+ENST00000367843_at+ENST00000367840_at+ENST00000367839_
at+ENST00000324705_at+

ENST00000312263_at+ENST00000298694_at+ENST00000298693_at+ENST00000354181_
at+ENST00000361379_at+

ENST00000358712_at+ENST00000336092_at+ENST00000382438_at+ENST00000226972_
at+ENST00000312524_at+

ENST00000372264_at+ENST00000362031_at+ENST00000355110_at+ENST00000332302_
at+ENST00000265395_at+

ENST00000376419_at+ENST00000376417_at+ENST00000277124_at+ENST00000381753_
at+ENST00000313860_at+

ENST00000383367_at+ENST00000318809_at+ENST00000308370_at+ENST00000383179_
      )000243562_at+

      )252576_at+ENST00000381917_at+ENST00000375493_at+ENST00000204005_
      )000301788_at,
                                  data=trainData, kernel="linear", cost=1,
      esponse")
```

```
prediksi_with_SMOTE_linear80<-
prediction(svmmod_80_linear_with_SMOTE,testData)
prediksi_no_SMOTE_linear80<-
prediction(svmmod_80_linear_no_SMOTE,testData)

#Dataset 2 : Kernel Linear dengan 90 Fitur Teratas
svmmod_90_linear_with_SMOTE<-
svm(y~ENST00000371095_at+ENST00000371085_at+ENST00000371082_at+ENST000003
49036_at+ENST00000265620_at+

ENST00000263080_at+ENST00000313949_at+ENST00000338783_at+ENST00000369420_
at+ENST00000369417_at+

ENST00000306082_at+ENST00000381463_at+ENST00000361035_at+ENST00000371102_
at+ENST00000371101_at+

ENST00000274311_at+ENST00000325945_at+ENST00000371100_at+ENST00000371075_
at+ENST00000354359_at+

ENST00000306120_at+ENST00000306090_at+ENST00000375594_at+ENST00000360868_
at+ENST00000358855_at+

ENST00000381936_at+ENST00000359894_at+ENST00000300947_at+ENST00000390597_
at+ENST00000380991_at+

ENST00000380986_at+ENST00000265028_at+ENST00000359623_at+ENST00000323078_
at+ENST00000283646_at+

ENST00000180173_at+ENST00000244296_at+ENST00000221293_at+ENST00000381251_
at+ENST00000304101_at+

ENST00000371761_at+ENST00000217246_at+ENST00000262662_at+ENST00000356692_
at+ENST00000361625_at+

ENST00000253023_at+ENST00000367843_at+ENST00000367840_at+ENST00000367839_
at+ENST00000324705_at+

ENST00000312263_at+ENST00000298694_at+ENST00000298693_at+ENST00000354181_
at+ENST00000361379_at+

ENST00000358712_at+ENST00000336092_at+ENST00000382438_at+ENST00000226972_
at+ENST00000312524_at+

ENST00000372264_at+ENST00000362031_at+ENST00000355110_at+ENST00000332302_
at+ENST00000265395_at+

ENST00000376419_at+ENST00000376417_at+ENST00000277124_at+ENST00000381753_
at+ENST00000313860_at+

ENST00000383367_at+ENST00000318809_at+ENST00000308370_at+ENST00000383179_
at+ENST00000243562_at+

ENST00000252576_at+ENST00000381917_at+ENST00000375493_at+ENST00000204005_
at+ENST00000301788_at+

ENST00000361272_at+ENST00000361231_at+ENST00000298047_at+ENST00000342745_
at+ENST00000244926_at+

0356505_at+ENST00000323760_at+ENST00000219168_at+ENST00000371534_
0000383362_at,
                data=smote_train,      kernel="linear",      cost=1,
esponse")
```

```
svmmod_90_linear_no_SMOTE<-
svm(y~ENST00000371095_at+ENST00000371085_at+ENST00000371082_at+ENST000003
49036_at+ENST00000265620_at+

ENST00000263080_at+ENST00000313949_at+ENST00000338783_at+ENST00000369420_
at+ENST00000369417_at+

ENST00000306082_at+ENST00000381463_at+ENST00000361035_at+ENST00000371102_
at+ENST00000371101_at+

ENST00000274311_at+ENST00000325945_at+ENST00000371100_at+ENST00000371075_
at+ENST00000354359_at+

ENST00000306120_at+ENST00000306090_at+ENST00000375594_at+ENST00000360868_
at+ENST00000358855_at+

ENST00000381936_at+ENST00000359894_at+ENST00000300947_at+ENST00000390597_
at+ENST00000380991_at+

ENST00000380986_at+ENST00000265028_at+ENST00000359623_at+ENST00000323078_
at+ENST00000283646_at+

ENST00000180173_at+ENST00000244296_at+ENST00000221293_at+ENST00000381251_
at+ENST00000304101_at+

ENST00000371761_at+ENST00000217246_at+ENST00000262662_at+ENST00000356692_
at+ENST00000361625_at+

ENST00000253023_at+ENST00000367843_at+ENST00000367840_at+ENST00000367839_
at+ENST00000324705_at+

ENST00000312263_at+ENST00000298694_at+ENST00000298693_at+ENST00000354181_
at+ENST00000361379_at+

ENST00000358712_at+ENST00000336092_at+ENST00000382438_at+ENST00000226972_
at+ENST00000312524_at+

ENST00000372264_at+ENST00000362031_at+ENST00000355110_at+ENST00000332302_
at+ENST00000265395_at+

ENST00000376419_at+ENST00000376417_at+ENST00000277124_at+ENST00000381753_
at+ENST00000313860_at+

ENST00000383367_at+ENST00000318809_at+ENST00000308370_at+ENST00000383179_
at+ENST00000243562_at+

ENST00000252576_at+ENST00000381917_at+ENST00000375493_at+ENST00000204005_
at+ENST00000301788_at+

ENST00000361272_at+ENST00000361231_at+ENST00000298047_at+ENST00000342745_
at+ENST00000244926_at+

ENST00000356505_at+ENST00000323760_at+ENST00000219168_at+ENST00000371534_
at+ENST00000383362_at,
                                data=trainData, kernel="linear", cost=1,
types="response")
prediksi_with_SMOTE_linear90<-
prediction(svmmod_90_linear_with_SMOTE,testData)
      _no_SMOTE_linear90<-
      on(svmmod_90_linear_no_SMOTE,testData)
```

2 : Kernel Linear dengan 100 Fitur Teratas

```
svmmod_100_linear_with_SMOTE<-
svm(y~ENST00000371095_at+ENST00000371085_at+ENST00000371082_at+ENST000003
49036_at+ENST00000265620_at+

ENST00000263080_at+ENST00000313949_at+ENST00000338783_at+ENST00000369420_
at+ENST00000369417_at+

ENST00000306082_at+ENST00000381463_at+ENST00000361035_at+ENST00000371102_
at+ENST00000371101_at+

ENST00000274311_at+ENST00000325945_at+ENST00000371100_at+ENST00000371075_
at+ENST00000354359_at+

ENST00000306120_at+ENST00000306090_at+ENST00000375594_at+ENST00000360868_
at+ENST00000358855_at+

ENST00000381936_at+ENST00000359894_at+ENST00000300947_at+ENST00000390597_
at+ENST00000380991_at+

ENST00000380986_at+ENST00000265028_at+ENST00000359623_at+ENST00000323078_
at+ENST00000283646_at+

ENST00000180173_at+ENST00000244296_at+ENST00000221293_at+ENST00000381251_
at+ENST00000304101_at+

ENST00000371761_at+ENST00000217246_at+ENST00000262662_at+ENST00000356692_
at+ENST00000361625_at+

ENST00000253023_at+ENST00000367843_at+ENST00000367840_at+ENST00000367839_
at+ENST00000324705_at+

ENST00000312263_at+ENST00000298694_at+ENST00000298693_at+ENST00000354181_
at+ENST00000361379_at+

ENST00000358712_at+ENST00000336092_at+ENST00000382438_at+ENST00000226972_
at+ENST00000312524_at+

ENST00000372264_at+ENST00000362031_at+ENST00000355110_at+ENST00000332302_
at+ENST00000265395_at+

ENST00000376419_at+ENST00000376417_at+ENST00000277124_at+ENST00000381753_
at+ENST00000313860_at+

ENST00000383367_at+ENST00000318809_at+ENST00000308370_at+ENST00000383179_
at+ENST00000243562_at+

ENST00000252576_at+ENST00000381917_at+ENST00000375493_at+ENST00000204005_
at+ENST00000301788_at+

ENST00000361272_at+ENST00000361231_at+ENST00000298047_at+ENST00000342745_
at+ENST00000244926_at+

ENST00000356505_at+ENST00000323760_at+ENST00000219168_at+ENST00000371534_
at+ENST00000383362_at+

ENST00000383360_at+ENST00000383359_at+ENST00000370109_at+ENST00000383178_
at+ENST00000383177_at+

)370107_at+ENST00000371646_at+ENST00000223864_at+ENST00000371554_
)000342173_at,
                data=smote_train,                kernel="linear",
ypes="response")
```

```
svmmod_100_linear_no_SMOTE<-
svm(y~ENST00000371095_at+ENST00000371085_at+ENST00000371082_at+ENST000003
49036_at+ENST00000265620_at+

ENST00000263080_at+ENST00000313949_at+ENST00000338783_at+ENST00000369420_
at+ENST00000369417_at+

ENST00000306082_at+ENST00000381463_at+ENST00000361035_at+ENST00000371102_
at+ENST00000371101_at+

ENST00000274311_at+ENST00000325945_at+ENST00000371100_at+ENST00000371075_
at+ENST00000354359_at+

ENST00000306120_at+ENST00000306090_at+ENST00000375594_at+ENST00000360868_
at+ENST00000358855_at+

ENST00000381936_at+ENST00000359894_at+ENST00000300947_at+ENST00000390597_
at+ENST00000380991_at+

ENST00000380986_at+ENST00000265028_at+ENST00000359623_at+ENST00000323078_
at+ENST00000283646_at+

ENST00000180173_at+ENST00000244296_at+ENST00000221293_at+ENST00000381251_
at+ENST00000304101_at+

ENST00000371761_at+ENST00000217246_at+ENST00000262662_at+ENST00000356692_
at+ENST00000361625_at+

ENST00000253023_at+ENST00000367843_at+ENST00000367840_at+ENST00000367839_
at+ENST00000324705_at+

ENST00000312263_at+ENST00000298694_at+ENST00000298693_at+ENST00000354181_
at+ENST00000361379_at+

ENST00000358712_at+ENST00000336092_at+ENST00000382438_at+ENST00000226972_
at+ENST00000312524_at+

ENST00000372264_at+ENST00000362031_at+ENST00000355110_at+ENST00000332302_
at+ENST00000265395_at+

ENST00000376419_at+ENST00000376417_at+ENST00000277124_at+ENST00000381753_
at+ENST00000313860_at+

ENST00000383367_at+ENST00000318809_at+ENST00000308370_at+ENST00000383179_
at+ENST00000243562_at+

ENST00000252576_at+ENST00000381917_at+ENST00000375493_at+ENST00000204005_
at+ENST00000301788_at+

ENST00000361272_at+ENST00000361231_at+ENST00000298047_at+ENST00000342745_
at+ENST00000244926_at+

ENST00000356505_at+ENST00000323760_at+ENST00000219168_at+ENST00000371534_
at+ENST00000383362_at+

ENST00000383360_at+ENST00000383359_at+ENST00000370109_at+ENST00000383178_
at+ENST00000383177_at+

)370107_at+ENST00000371646_at+ENST00000223864_at+ENST00000371554_
)000342173_at,
                data=trainData,                    kernel="linear",
ypes="response")
_with_SMOTE_linear100<-
)n(svmmod_100_linear_with_SMOTE,testData)
```

```
prediksi_no_SMOTE_linear100<-
prediction(svmmod_100_linear_no_SMOTE,testData)

#Kernel RBF
#Dataset 2 : Kernel RBF dengan 10 Fitur Teratas
svmmod_10_RBF_with_SMOTE<-
svm(y~ENST00000371095_at+ENST00000371085_at+ENST00000371082_at+ENST000003
49036_at+ENST00000265620_at+

ENST00000263080_at+ENST00000313949_at+ENST00000338783_at+ENST00000369420_
at+ENST00000369417_at,
                    data=smote_train,       kernel="radial",       cost=32,
gamma=0.0156250000, types="response")
svmmod_10_RBF_no_SMOTE<-
svm(y~ENST00000371095_at+ENST00000371085_at+ENST00000371082_at+ENST000003
49036_at+ENST00000265620_at+

ENST00000263080_at+ENST00000313949_at+ENST00000338783_at+ENST00000369420_
at+ENST00000369417_at,
                    data=trainData,        kernel="radial",        cost=32,
gamma=0.0156250000, types="response")
prediksi_with_SMOTE_RBF10<-prediction(svmmod_10_RBF_with_SMOTE,testData)
prediksi_no_SMOTE_RBF10<-prediction(svmmod_10_RBF_no_SMOTE,testData)

#Dataset 2 : Kernel RBF dengan 20 Fitur Teratas
svmmod_20_RBF_with_SMOTE<-
svm(y~ENST00000371095_at+ENST00000371085_at+ENST00000371082_at+ENST000003
49036_at+ENST00000265620_at+

ENST00000263080_at+ENST00000313949_at+ENST00000338783_at+ENST00000369420_
at+ENST00000369417_at+

ENST00000306082_at+ENST00000381463_at+ENST00000361035_at+ENST00000371102_
at+ENST00000371101_at+

ENST00000274311_at+ENST00000325945_at+ENST00000371100_at+ENST00000371075_
at+ENST00000354359_at,
                    data=smote_train,       kernel="radial",       cost=32,
gamma=0.0156250000, types="response")
svmmod_20_RBF_no_SMOTE<-
svm(y~ENST00000371095_at+ENST00000371085_at+ENST00000371082_at+ENST000003
49036_at+ENST00000265620_at+

ENST00000263080_at+ENST00000313949_at+ENST00000338783_at+ENST00000369420_
at+ENST00000369417_at+

ENST00000306082_at+ENST00000381463_at+ENST00000361035_at+ENST00000371102_
at+ENST00000371101_at+

ENST00000274311_at+ENST00000325945_at+ENST00000371100_at+ENST00000371075_
at+ENST00000354359_at,
                    data=trainData,        kernel="radial",        cost=32,
gamma=0.0156250000, types="response")
prediksi_with_SMOTE_RBF20<-prediction(svmmod_20_RBF_with_SMOTE,testData)
prediksi_no_SMOTE_RBF20<-prediction(svmmod_20_RBF_no_SMOTE,testData)

#Dataset 2 : Kernel RBF dengan 30 Fitur Teratas
svmmod_30_RBF_with_SMOTE<-
        ST00000371095_at+ENST00000371085_at+ENST00000371082_at+ENST000003
        +ENST00000265620_at+

        )263080_at+ENST00000313949_at+ENST00000338783_at+ENST00000369420_
        )000369417_at+
```

```
ENST00000306082_at+ENST00000381463_at+ENST00000361035_at+ENST00000371102_
at+ENST00000371101_at+

ENST00000274311_at+ENST00000325945_at+ENST00000371100_at+ENST00000371075_
at+ENST00000354359_at+

ENST00000306120_at+ENST00000306090_at+ENST00000375594_at+ENST00000360868_
at+ENST00000358855_at+

ENST00000381936_at+ENST00000359894_at+ENST00000300947_at+ENST00000390597_
at+ENST00000380991_at,
                    data=smote_train,       kernel="radial",       cost=32,
gamma=0.0156250000, types="response")
svmmod_30_RBF_no_SMOTE<-
svm(y~ENST00000371095_at+ENST00000371085_at+ENST00000371082_at+ENST000003
49036_at+ENST00000265620_at+

ENST00000263080_at+ENST00000313949_at+ENST00000338783_at+ENST00000369420_
at+ENST00000369417_at+

ENST00000306082_at+ENST00000381463_at+ENST00000361035_at+ENST00000371102_
at+ENST00000371101_at+

ENST00000274311_at+ENST00000325945_at+ENST00000371100_at+ENST00000371075_
at+ENST00000354359_at+

ENST00000306120_at+ENST00000306090_at+ENST00000375594_at+ENST00000360868_
at+ENST00000358855_at+

ENST00000381936_at+ENST00000359894_at+ENST00000300947_at+ENST00000390597_
at+ENST00000380991_at,
                              data=trainData,  kernel="radial",  cost=32,
gamma=0.0156250000, types="response")
prediksi_with_SMOTE_RBF30<-prediction(svmmod_30_RBF_with_SMOTE,testData)
prediksi_no_SMOTE_RBF30<-prediction(svmmod_30_RBF_no_SMOTE,testData)

#Dataset 2 : Kernel RBF dengan 40 Fitur Teratas
svmmod_40_RBF_with_SMOTE<-
svm(y~ENST00000371095_at+ENST00000371085_at+ENST00000371082_at+ENST000003
49036_at+ENST00000265620_at+

ENST00000263080_at+ENST00000313949_at+ENST00000338783_at+ENST00000369420_
at+ENST00000369417_at+

ENST00000306082_at+ENST00000381463_at+ENST00000361035_at+ENST00000371102_
at+ENST00000371101_at+

ENST00000274311_at+ENST00000325945_at+ENST00000371100_at+ENST00000371075_
at+ENST00000354359_at+

ENST00000306120_at+ENST00000306090_at+ENST00000375594_at+ENST00000360868_
at+ENST00000358855_at+

ENST00000381936_at+ENST00000359894_at+ENST00000300947_at+ENST00000390597_
at+ENST00000380991_at+

ENST00000380986_at+ENST00000265028_at+ENST00000359623_at+ENST00000323078_
      )000283646_at+

      )180173_at+ENST00000244296_at+ENST00000221293_at+ENST00000381251_
      )000304101_at,
                    data=smote_train,       kernel="radial",       cost=32,
      )156250000, types="response")
```

```
svmmod_40_RBF_no_SMOTE<-
svm(y~ENST00000371095_at+ENST00000371085_at+ENST00000371082_at+ENST000003
49036_at+ENST00000265620_at+

ENST00000263080_at+ENST00000313949_at+ENST00000338783_at+ENST00000369420_
at+ENST00000369417_at+

ENST00000306082_at+ENST00000381463_at+ENST00000361035_at+ENST00000371102_
at+ENST00000371101_at+

ENST00000274311_at+ENST00000325945_at+ENST00000371100_at+ENST00000371075_
at+ENST00000354359_at+

ENST00000306120_at+ENST00000306090_at+ENST00000375594_at+ENST00000360868_
at+ENST00000358855_at+

ENST00000381936_at+ENST00000359894_at+ENST00000300947_at+ENST00000390597_
at+ENST00000380991_at+

ENST00000380986_at+ENST00000265028_at+ENST00000359623_at+ENST00000323078_
at+ENST00000283646_at+

ENST00000180173_at+ENST00000244296_at+ENST00000221293_at+ENST00000381251_
at+ENST00000304101_at,
                              data=trainData,  kernel="radial",  cost=32,
gamma=0.0156250000, types="response")
prediksi_with_SMOTE_RBF40<-prediction(svmmod_40_RBF_with_SMOTE,testData)
prediksi_no_SMOTE_RBF40<-prediction(svmmod_40_RBF_no_SMOTE,testData)

#Dataset 2 : Kernel RBF dengan 50 Fitur Teratas
svmmod_50_RBF_with_SMOTE<-
svm(y~ENST00000371095_at+ENST00000371085_at+ENST00000371082_at+ENST000003
49036_at+ENST00000265620_at+

ENST00000263080_at+ENST00000313949_at+ENST00000338783_at+ENST00000369420_
at+ENST00000369417_at+

ENST00000306082_at+ENST00000381463_at+ENST00000361035_at+ENST00000371102_
at+ENST00000371101_at+

ENST00000274311_at+ENST00000325945_at+ENST00000371100_at+ENST00000371075_
at+ENST00000354359_at+

ENST00000306120_at+ENST00000306090_at+ENST00000375594_at+ENST00000360868_
at+ENST00000358855_at+

ENST00000381936_at+ENST00000359894_at+ENST00000300947_at+ENST00000390597_
at+ENST00000380991_at+

ENST00000380986_at+ENST00000265028_at+ENST00000359623_at+ENST00000323078_
at+ENST00000283646_at+

ENST00000180173_at+ENST00000244296_at+ENST00000221293_at+ENST00000381251_
at+ENST00000304101_at+

ENST00000371761_at+ENST00000217246_at+ENST00000262662_at+ENST00000356692_
at+ENST00000361625_at+

)253023_at+ENST00000367843_at+ENST00000367840_at+ENST00000367839_
)000324705_at,
                data=smote_train,       kernel="radial",       cost=32,
)156250000, types="response")
```

```
svmmod_50_RBF_no_SMOTE<-
svm(y~ENST00000371095_at+ENST00000371085_at+ENST00000371082_at+ENST000003
49036_at+ENST00000265620_at+

ENST00000263080_at+ENST00000313949_at+ENST00000338783_at+ENST00000369420_
at+ENST00000369417_at+

ENST00000306082_at+ENST00000381463_at+ENST00000361035_at+ENST00000371102_
at+ENST00000371101_at+

ENST00000274311_at+ENST00000325945_at+ENST00000371100_at+ENST00000371075_
at+ENST00000354359_at+

ENST00000306120_at+ENST00000306090_at+ENST00000375594_at+ENST00000360868_
at+ENST00000358855_at+

ENST00000381936_at+ENST00000359894_at+ENST00000300947_at+ENST00000390597_
at+ENST00000380991_at+

ENST00000380986_at+ENST00000265028_at+ENST00000359623_at+ENST00000323078_
at+ENST00000283646_at+

ENST00000180173_at+ENST00000244296_at+ENST00000221293_at+ENST00000381251_
at+ENST00000304101_at+

ENST00000371761_at+ENST00000217246_at+ENST00000262662_at+ENST00000356692_
at+ENST00000361625_at+

ENST00000253023_at+ENST00000367843_at+ENST00000367840_at+ENST00000367839_
at+ENST00000324705_at,
                              data=trainData,  kernel="radial",  cost=32,
gamma=0.0156250000, types="response")
prediksi_with_SMOTE_RBF50<-prediction(svmmod_50_RBF_with_SMOTE,testData)
prediksi_no_SMOTE_RBF50<-prediction(svmmod_50_RBF_no_SMOTE,testData)

#Dataset 2 : Kernel RBF dengan 60 Fitur Teratas
svmmod_60_RBF_with_SMOTE<-
svm(y~ENST00000371095_at+ENST00000371085_at+ENST00000371082_at+ENST000003
49036_at+ENST00000265620_at+

ENST00000263080_at+ENST00000313949_at+ENST00000338783_at+ENST00000369420_
at+ENST00000369417_at+

ENST00000306082_at+ENST00000381463_at+ENST00000361035_at+ENST00000371102_
at+ENST00000371101_at+

ENST00000274311_at+ENST00000325945_at+ENST00000371100_at+ENST00000371075_
at+ENST00000354359_at+

ENST00000306120_at+ENST00000306090_at+ENST00000375594_at+ENST00000360868_
at+ENST00000358855_at+

ENST00000381936_at+ENST00000359894_at+ENST00000300947_at+ENST00000390597_
at+ENST00000380991_at+

ENST00000380986_at+ENST00000265028_at+ENST00000359623_at+ENST00000323078_
at+ENST00000283646_at+

)0180173_at+ENST00000244296_at+ENST00000221293_at+ENST00000381251_
)000304101_at+

)371761_at+ENST00000217246_at+ENST00000262662_at+ENST00000356692_
)000361625_at+
```

```
ENST00000253023_at+ENST00000367843_at+ENST00000367840_at+ENST00000367839_
at+ENST00000324705_at+

ENST00000312263_at+ENST00000298694_at+ENST00000298693_at+ENST00000354181_
at+ENST00000361379_at+

ENST00000358712_at+ENST00000336092_at+ENST00000382438_at+ENST00000226972_
at+ENST00000312524_at,
                    data=smote_train,       kernel="radial",       cost=32,
gamma=0.0156250000, types="response")
svmmod_60_RBF_no_SMOTE<-
svm(y~ENST00000371095_at+ENST00000371085_at+ENST00000371082_at+ENST000003
49036_at+ENST00000265620_at+

ENST00000263080_at+ENST00000313949_at+ENST00000338783_at+ENST00000369420_
at+ENST00000369417_at+

ENST00000306082_at+ENST00000381463_at+ENST00000361035_at+ENST00000371102_
at+ENST00000371101_at+

ENST00000274311_at+ENST00000325945_at+ENST00000371100_at+ENST00000371075_
at+ENST00000354359_at+

ENST00000306120_at+ENST00000306090_at+ENST00000375594_at+ENST00000360868_
at+ENST00000358855_at+

ENST00000381936_at+ENST00000359894_at+ENST00000300947_at+ENST00000390597_
at+ENST00000380991_at+

ENST00000380986_at+ENST00000265028_at+ENST00000359623_at+ENST00000323078_
at+ENST00000283646_at+

ENST00000180173_at+ENST00000244296_at+ENST00000221293_at+ENST00000381251_
at+ENST00000304101_at+

ENST00000371761_at+ENST00000217246_at+ENST00000262662_at+ENST00000356692_
at+ENST00000361625_at+

ENST00000253023_at+ENST00000367843_at+ENST00000367840_at+ENST00000367839_
at+ENST00000324705_at+

ENST00000312263_at+ENST00000298694_at+ENST00000298693_at+ENST00000354181_
at+ENST00000361379_at+

ENST00000358712_at+ENST00000336092_at+ENST00000382438_at+ENST00000226972_
at+ENST00000312524_at,
                               data=trainData,  kernel="radial",  cost=32,
gamma=0.0156250000, types="response")
prediksi_with_SMOTE_RBF60<-prediction(svmmod_60_RBF_with_SMOTE,testData)
prediksi_no_SMOTE_RBF60<-prediction(svmmod_60_RBF_no_SMOTE,testData)

#Dataset 2 : Kernel RBF dengan 70 Fitur Teratas
svmmod_70_RBF_with_SMOTE<-
svm(y~ENST00000371095_at+ENST00000371085_at+ENST00000371082_at+ENST000003
49036_at+ENST00000265620_at+

ENST00000263080_at+ENST00000313949_at+ENST00000338783_at+ENST00000369420_
      )000369417_at+

      )306082_at+ENST00000381463_at+ENST00000361035_at+ENST00000371102_
      )000371101_at+
```

```
ENST00000274311_at+ENST00000325945_at+ENST00000371100_at+ENST00000371075_
at+ENST00000354359_at+

ENST00000306120_at+ENST00000306090_at+ENST00000375594_at+ENST00000360868_
at+ENST00000358855_at+

ENST00000381936_at+ENST00000359894_at+ENST00000300947_at+ENST00000390597_
at+ENST00000380991_at+

ENST00000380986_at+ENST00000265028_at+ENST00000359623_at+ENST00000323078_
at+ENST00000283646_at+

ENST00000180173_at+ENST00000244296_at+ENST00000221293_at+ENST00000381251_
at+ENST00000304101_at+

ENST00000371761_at+ENST00000217246_at+ENST00000262662_at+ENST00000356692_
at+ENST00000361625_at+

ENST00000253023_at+ENST00000367843_at+ENST00000367840_at+ENST00000367839_
at+ENST00000324705_at+

ENST00000312263_at+ENST00000298694_at+ENST00000298693_at+ENST00000354181_
at+ENST00000361379_at+

ENST00000358712_at+ENST00000336092_at+ENST00000382438_at+ENST00000226972_
at+ENST00000312524_at+

ENST00000361272_at+ENST00000361231_at+ENST00000298047_at+ENST00000342745_
at+ENST00000244926_at+

ENST00000356505_at+ENST00000323760_at+ENST00000219168_at+ENST00000371534_
at+ENST00000383362_at,
                  data=smote_train,       kernel="radial",       cost=32,
gamma=0.0156250000, types="response")
svmmod_70_RBF_no_SMOTE<-
svm(y~ENST00000371095_at+ENST00000371085_at+ENST00000371082_at+ENST000003
49036_at+ENST00000265620_at+

ENST00000263080_at+ENST00000313949_at+ENST00000338783_at+ENST00000369420_
at+ENST00000369417_at+

ENST00000306082_at+ENST00000381463_at+ENST00000361035_at+ENST00000371102_
at+ENST00000371101_at+

ENST00000274311_at+ENST00000325945_at+ENST00000371100_at+ENST00000371075_
at+ENST00000354359_at+

ENST00000306120_at+ENST00000306090_at+ENST00000375594_at+ENST00000360868_
at+ENST00000358855_at+

ENST00000381936_at+ENST00000359894_at+ENST00000300947_at+ENST00000390597_
at+ENST00000380991_at+

ENST00000380986_at+ENST00000265028_at+ENST00000359623_at+ENST00000323078_
at+ENST00000283646_at+

ENST00000180173_at+ENST00000244296_at+ENST00000221293_at+ENST00000381251_
          )000304101_at+

          )371761_at+ENST00000217246_at+ENST00000262662_at+ENST00000356692_
          )000361625_at+
```

```
ENST00000253023_at+ENST00000367843_at+ENST00000367840_at+ENST00000367839_
at+ENST00000324705_at+

ENST00000312263_at+ENST00000298694_at+ENST00000298693_at+ENST00000354181_
at+ENST00000361379_at+

ENST00000358712_at+ENST00000336092_at+ENST00000382438_at+ENST00000226972_
at+ENST00000312524_at+

ENST00000361272_at+ENST00000361231_at+ENST00000298047_at+ENST00000342745_
at+ENST00000244926_at+

ENST00000356505_at+ENST00000323760_at+ENST00000219168_at+ENST00000371534_
at+ENST00000383362_at,
                              data=trainData,  kernel="radial",  cost=32,
gamma=0.0156250000, types="response")
prediksi_with_SMOTE_RBF70<-prediction(svmmod_70_RBF_with_SMOTE,testData)
prediksi_no_SMOTE_RBF70<-prediction(svmmod_70_RBF_no_SMOTE,testData)

#Dataset 2 : Kernel RBF dengan 80 Fitur Teratas
svmmod_80_RBF_with_SMOTE<-
svm(y~ENST00000371095_at+ENST00000371085_at+ENST00000371082_at+ENST000003
49036_at+ENST00000265620_at+

ENST00000263080_at+ENST00000313949_at+ENST00000338783_at+ENST00000369420_
at+ENST00000369417_at+

ENST00000306082_at+ENST00000381463_at+ENST00000361035_at+ENST00000371102_
at+ENST00000371101_at+

ENST00000274311_at+ENST00000325945_at+ENST00000371100_at+ENST00000371075_
at+ENST00000354359_at+

ENST00000306120_at+ENST00000306090_at+ENST00000375594_at+ENST00000360868_
at+ENST00000358855_at+

ENST00000381936_at+ENST00000359894_at+ENST00000300947_at+ENST00000390597_
at+ENST00000380991_at+

ENST00000380986_at+ENST00000265028_at+ENST00000359623_at+ENST00000323078_
at+ENST00000283646_at+

ENST00000180173_at+ENST00000244296_at+ENST00000221293_at+ENST00000381251_
at+ENST00000304101_at+

ENST00000371761_at+ENST00000217246_at+ENST00000262662_at+ENST00000356692_
at+ENST00000361625_at+

ENST00000253023_at+ENST00000367843_at+ENST00000367840_at+ENST00000367839_
at+ENST00000324705_at+

ENST00000312263_at+ENST00000298694_at+ENST00000298693_at+ENST00000354181_
at+ENST00000361379_at+

ENST00000358712_at+ENST00000336092_at+ENST00000382438_at+ENST00000226972_
at+ENST00000312524_at+

)372264_at+ENST00000362031_at+ENST00000355110_at+ENST00000332302_
)000265395_at+

)376419_at+ENST00000376417_at+ENST00000277124_at+ENST00000381753_
)000313860_at+
```

```
ENST00000383367_at+ENST00000318809_at+ENST00000308370_at+ENST00000383179_
at+ENST00000243562_at+

ENST00000252576_at+ENST00000381917_at+ENST00000375493_at+ENST00000204005_
at+ENST00000301788_at,
                    data=smote_train,       kernel="radial",       cost=32,
gamma=0.0156250000, types="response")
svmmod_80_RBF_no_SMOTE<-
svm(y~ENST00000371095_at+ENST00000371085_at+ENST00000371082_at+ENST000003
49036_at+ENST00000265620_at+

ENST00000263080_at+ENST00000313949_at+ENST00000338783_at+ENST00000369420_
at+ENST00000369417_at+

ENST00000306082_at+ENST00000381463_at+ENST00000361035_at+ENST00000371102_
at+ENST00000371101_at+

ENST00000274311_at+ENST00000325945_at+ENST00000371100_at+ENST00000371075_
at+ENST00000354359_at+

ENST00000306120_at+ENST00000306090_at+ENST00000375594_at+ENST00000360868_
at+ENST00000358855_at+

ENST00000381936_at+ENST00000359894_at+ENST00000300947_at+ENST00000390597_
at+ENST00000380991_at+

ENST00000380986_at+ENST00000265028_at+ENST00000359623_at+ENST00000323078_
at+ENST00000283646_at+

ENST00000180173_at+ENST00000244296_at+ENST00000221293_at+ENST00000381251_
at+ENST00000304101_at+

ENST00000371761_at+ENST00000217246_at+ENST00000262662_at+ENST00000356692_
at+ENST00000361625_at+

ENST00000253023_at+ENST00000367843_at+ENST00000367840_at+ENST00000367839_
at+ENST00000324705_at+

ENST00000312263_at+ENST00000298694_at+ENST00000298693_at+ENST00000354181_
at+ENST00000361379_at+

ENST00000358712_at+ENST00000336092_at+ENST00000382438_at+ENST00000226972_
at+ENST00000312524_at+

ENST00000372264_at+ENST00000362031_at+ENST00000355110_at+ENST00000332302_
at+ENST00000265395_at+

ENST00000376419_at+ENST00000376417_at+ENST00000277124_at+ENST00000381753_
at+ENST00000313860_at+

ENST00000383367_at+ENST00000318809_at+ENST00000308370_at+ENST00000383179_
at+ENST00000243562_at+

ENST00000252576_at+ENST00000381917_at+ENST00000375493_at+ENST00000204005_
at+ENST00000301788_at,
                              data=trainData,  kernel="radial",  cost=32,
gamma=0.0156250000, types="response")
        _with_SMOTE_RBF80<-prediction(svmmod_80_RBF_with_SMOTE,testData)
        _no_SMOTE_RBF80<-prediction(svmmod_80_RBF_no_SMOTE,testData)
```

2 : Kernel RBF dengan 90 Fitur Teratas

```
svmmod_90_RBF_with_SMOTE<-
svm(y~ENST00000371095_at+ENST00000371085_at+ENST00000371082_at+ENST000003
49036_at+ENST00000265620_at+

ENST00000263080_at+ENST00000313949_at+ENST00000338783_at+ENST00000369420_
at+ENST00000369417_at+

ENST00000306082_at+ENST00000381463_at+ENST00000361035_at+ENST00000371102_
at+ENST00000371101_at+

ENST00000274311_at+ENST00000325945_at+ENST00000371100_at+ENST00000371075_
at+ENST00000354359_at+

ENST00000306120_at+ENST00000306090_at+ENST00000375594_at+ENST00000360868_
at+ENST00000358855_at+

ENST00000381936_at+ENST00000359894_at+ENST00000300947_at+ENST00000390597_
at+ENST00000380991_at+

ENST00000380986_at+ENST00000265028_at+ENST00000359623_at+ENST00000323078_
at+ENST00000283646_at+

ENST00000180173_at+ENST00000244296_at+ENST00000221293_at+ENST00000381251_
at+ENST00000304101_at+

ENST00000371761_at+ENST00000217246_at+ENST00000262662_at+ENST00000356692_
at+ENST00000361625_at+

ENST00000253023_at+ENST00000367843_at+ENST00000367840_at+ENST00000367839_
at+ENST00000324705_at+

ENST00000312263_at+ENST00000298694_at+ENST00000298693_at+ENST00000354181_
at+ENST00000361379_at+

ENST00000358712_at+ENST00000336092_at+ENST00000382438_at+ENST00000226972_
at+ENST00000312524_at+

ENST00000372264_at+ENST00000362031_at+ENST00000355110_at+ENST00000332302_
at+ENST00000265395_at+

ENST00000376419_at+ENST00000376417_at+ENST00000277124_at+ENST00000381753_
at+ENST00000313860_at+

ENST00000383367_at+ENST00000318809_at+ENST00000308370_at+ENST00000383179_
at+ENST00000243562_at+

ENST00000252576_at+ENST00000381917_at+ENST00000375493_at+ENST00000204005_
at+ENST00000301788_at+

ENST00000361272_at+ENST00000361231_at+ENST00000298047_at+ENST00000342745_
at+ENST00000244926_at+

ENST00000356505_at+ENST00000323760_at+ENST00000219168_at+ENST00000371534_
at+ENST00000383362_at,
                    data=smote_train,       kernel="radial",       cost=32,
gamma=0.0156250000, types="response")
svmmod_90_RBF_no_SMOTE<-
svm(y~ENST00000371095_at+ENST00000371085_at+ENST00000371082_at+ENST000003
      +ENST00000265620_at+

      )263080_at+ENST00000313949_at+ENST00000338783_at+ENST00000369420_
      )000369417_at+
```

```
ENST00000306082_at+ENST00000381463_at+ENST00000361035_at+ENST00000371102_
at+ENST00000371101_at+

ENST00000274311_at+ENST00000325945_at+ENST00000371100_at+ENST00000371075_
at+ENST00000354359_at+

ENST00000306120_at+ENST00000306090_at+ENST00000375594_at+ENST00000360868_
at+ENST00000358855_at+

ENST00000381936_at+ENST00000359894_at+ENST00000300947_at+ENST00000390597_
at+ENST00000380991_at+

ENST00000380986_at+ENST00000265028_at+ENST00000359623_at+ENST00000323078_
at+ENST00000283646_at+

ENST00000180173_at+ENST00000244296_at+ENST00000221293_at+ENST00000381251_
at+ENST00000304101_at+

ENST00000371761_at+ENST00000217246_at+ENST00000262662_at+ENST00000356692_
at+ENST00000361625_at+

ENST00000253023_at+ENST00000367843_at+ENST00000367840_at+ENST00000367839_
at+ENST00000324705_at+

ENST00000312263_at+ENST00000298694_at+ENST00000298693_at+ENST00000354181_
at+ENST00000361379_at+

ENST00000358712_at+ENST00000336092_at+ENST00000382438_at+ENST00000226972_
at+ENST00000312524_at+

ENST00000372264_at+ENST00000362031_at+ENST00000355110_at+ENST00000332302_
at+ENST00000265395_at+

ENST00000376419_at+ENST00000376417_at+ENST00000277124_at+ENST00000381753_
at+ENST00000313860_at+

ENST00000383367_at+ENST00000318809_at+ENST00000308370_at+ENST00000383179_
at+ENST00000243562_at+

ENST00000252576_at+ENST00000381917_at+ENST00000375493_at+ENST00000204005_
at+ENST00000301788_at+

ENST00000361272_at+ENST00000361231_at+ENST00000298047_at+ENST00000342745_
at+ENST00000244926_at+

ENST00000356505_at+ENST00000323760_at+ENST00000219168_at+ENST00000371534_
at+ENST00000383362_at,
                                data=trainData,  kernel="radial",  cost=32,
gamma=0.0156250000, types="response")
prediksi_with_SMOTE_RBF_90<-prediction(svmmod_90_RBF_with_SMOTE,testData)
prediksi_no_SMOTE_RBF_90<-prediction(svmmod_90_RBF_no_SMOTE,testData)

#Dataset 2 : Kernel RBF dengan 100 Fitur Teratas
svmmod_100_RBF_with_SMOTE<-
svm(y~ENST00000371095_at+ENST00000371085_at+ENST00000371082_at+ENST000003
49036_at+ENST00000265620_at+

)263080_at+ENST00000313949_at+ENST00000338783_at+ENST00000369420_
)000369417_at+

)306082_at+ENST00000381463_at+ENST00000361035_at+ENST00000371102_
)000371101_at+
```

```
ENST00000274311_at+ENST00000325945_at+ENST00000371100_at+ENST00000371075_
at+ENST00000354359_at+

ENST00000306120_at+ENST00000306090_at+ENST00000375594_at+ENST00000360868_
at+ENST00000358855_at+

ENST00000381936_at+ENST00000359894_at+ENST00000300947_at+ENST00000390597_
at+ENST00000380991_at+

ENST00000380986_at+ENST00000265028_at+ENST00000359623_at+ENST00000323078_
at+ENST00000283646_at+

ENST00000180173_at+ENST00000244296_at+ENST00000221293_at+ENST00000381251_
at+ENST00000304101_at+

ENST00000371761_at+ENST00000217246_at+ENST00000262662_at+ENST00000356692_
at+ENST00000361625_at+

ENST00000253023_at+ENST00000367843_at+ENST00000367840_at+ENST00000367839_
at+ENST00000324705_at+

ENST00000312263_at+ENST00000298694_at+ENST00000298693_at+ENST00000354181_
at+ENST00000361379_at+

ENST00000358712_at+ENST00000336092_at+ENST00000382438_at+ENST00000226972_
at+ENST00000312524_at+

ENST00000372264_at+ENST00000362031_at+ENST00000355110_at+ENST00000332302_
at+ENST00000265395_at+

ENST00000376419_at+ENST00000376417_at+ENST00000277124_at+ENST00000381753_
at+ENST00000313860_at+

ENST00000383367_at+ENST00000318809_at+ENST00000308370_at+ENST00000383179_
at+ENST00000243562_at+

ENST00000252576_at+ENST00000381917_at+ENST00000375493_at+ENST00000204005_
at+ENST00000301788_at+

ENST00000361272_at+ENST00000361231_at+ENST00000298047_at+ENST00000342745_
at+ENST00000244926_at+

ENST00000356505_at+ENST00000323760_at+ENST00000219168_at+ENST00000371534_
at+ENST00000383362_at+

ENST00000383360_at+ENST00000383359_at+ENST00000370109_at+ENST00000383178_
at+ENST00000383177_at+

ENST00000370107_at+ENST00000371646_at+ENST00000223864_at+ENST00000371554_
at+ENST00000342173_at,
                    data=smote_train,      kernel="radial",      cost=32,
gamma=0.0156250000, types="response")
svmmod_100_RBF_no_SMOTE<-
svm(y~ENST00000371095_at+ENST00000371085_at+ENST00000371082_at+ENST000003
49036_at+ENST00000265620_at+

ENST00000263080_at+ENST00000313949_at+ENST00000338783_at+ENST00000369420_
      0000369417_at+

      0306082_at+ENST00000381463_at+ENST00000361035_at+ENST00000371102_
      0000371101_at+
```

```
ENST00000274311_at+ENST00000325945_at+ENST00000371100_at+ENST00000371075_
at+ENST00000354359_at+

ENST00000306120_at+ENST00000306090_at+ENST00000375594_at+ENST00000360868_
at+ENST00000358855_at+

ENST00000381936_at+ENST00000359894_at+ENST00000300947_at+ENST00000390597_
at+ENST00000380991_at+

ENST00000380986_at+ENST00000265028_at+ENST00000359623_at+ENST00000323078_
at+ENST00000283646_at+

ENST00000180173_at+ENST00000244296_at+ENST00000221293_at+ENST00000381251_
at+ENST00000304101_at+

ENST00000371761_at+ENST00000217246_at+ENST00000262662_at+ENST00000356692_
at+ENST00000361625_at+

ENST00000253023_at+ENST00000367843_at+ENST00000367840_at+ENST00000367839_
at+ENST00000324705_at+

ENST00000312263_at+ENST00000298694_at+ENST00000298693_at+ENST00000354181_
at+ENST00000361379_at+

ENST00000358712_at+ENST00000336092_at+ENST00000382438_at+ENST00000226972_
at+ENST00000312524_at+

ENST00000372264_at+ENST00000362031_at+ENST00000355110_at+ENST00000332302_
at+ENST00000265395_at+

ENST00000376419_at+ENST00000376417_at+ENST00000277124_at+ENST00000381753_
at+ENST00000313860_at+

ENST00000383367_at+ENST00000318809_at+ENST00000308370_at+ENST00000383179_
at+ENST00000243562_at+

ENST00000252576_at+ENST00000381917_at+ENST00000375493_at+ENST00000204005_
at+ENST00000301788_at+

ENST00000361272_at+ENST00000361231_at+ENST00000298047_at+ENST00000342745_
at+ENST00000244926_at+

ENST00000356505_at+ENST00000323760_at+ENST00000219168_at+ENST00000371534_
at+ENST00000383362_at+

ENST00000383360_at+ENST00000383359_at+ENST00000370109_at+ENST00000383178_
at+ENST00000383177_at+

ENST00000370107_at+ENST00000371646_at+ENST00000223864_at+ENST00000371554_
at+ENST00000342173_at,
                                data=trainData,  kernel="radial",  cost=32,
gamma=0.0156250000, types="response")
prediksi_with_SMOTE_RBF100<-
prediction(svmmod_100_RBF_with_SMOTE,testData)
prediksi_no_SMOTE_RBF100<-prediction(svmmod_100_RBF_no_SMOTE,testData)

#Kernel Polynomial Orde 2
        2 : Kernel Polynomial Orde 2 dengan 10 Fitur Teratas
)_Poly2_with_SMOTE<-
ST00000371095_at+ENST00000371085_at+ENST00000371082_at+ENST000003
+ENST00000265620_at+
```

```
ENST00000263080_at+ENST00000313949_at+ENST00000338783_at+ENST00000369420_
at+ENST00000369417_at,
                    data=smote_train,    kernel="polynomial",    cost=16,
degree=2, coef0=0.125, gamma=0.015625, types="response")
svmmod_10_Poly2_no_SMOTE<-
svm(y~ENST00000371095_at+ENST00000371085_at+ENST00000371082_at+ENST000003
49036_at+ENST00000265620_at+

ENST00000263080_at+ENST00000313949_at+ENST00000338783_at+ENST00000369420_
at+ENST00000369417_at,
                    data=trainData,     kernel="polynomial",     cost=16,
degree=2, coef0=0.125, gamma=0.015625, types="response")
prediksi_with_SMOTE_Poly10<-
prediction(svmmod_10_Poly2_with_SMOTE,testData)
prediksi_no_SMOTE_Poly10<-prediction(svmmod_10_Poly2_no_SMOTE,testData)

#Kernel Polynomial Orde 2 dengan 20 Fitur Teratas
svmmod_20_Poly2_with_SMOTE<-
svm(y~ENST00000371095_at+ENST00000371085_at+ENST00000371082_at+ENST000003
49036_at+ENST00000265620_at+

ENST00000263080_at+ENST00000313949_at+ENST00000338783_at+ENST00000369420_
at+ENST00000369417_at+

ENST00000306082_at+ENST00000381463_at+ENST00000361035_at+ENST00000371102_
at+ENST00000371101_at+

ENST00000274311_at+ENST00000325945_at+ENST00000371100_at+ENST00000371075_
at+ENST00000354359_at,
                    data=smote_train,    kernel="polynomial",    cost=16,
degree=2, coef0=0.125, gamma=0.015625, types="response")
svmmod_20_Poly2_no_SMOTE<-
svm(y~ENST00000371095_at+ENST00000371085_at+ENST00000371082_at+ENST000003
49036_at+ENST00000265620_at+

ENST00000263080_at+ENST00000313949_at+ENST00000338783_at+ENST00000369420_
at+ENST00000369417_at+

ENST00000306082_at+ENST00000381463_at+ENST00000361035_at+ENST00000371102_
at+ENST00000371101_at+

ENST00000274311_at+ENST00000325945_at+ENST00000371100_at+ENST00000371075_
at+ENST00000354359_at,
                                data=trainData,      kernel="polynomial",
cost=16, degree=2, coef0=0.125, gamma=0.015625, types="response")
prediksi_with_SMOTE_Poly20<-
prediction(svmmod_20_Poly2_with_SMOTE,testData)
prediksi_no_SMOTE_Poly20<-prediction(svmmod_20_Poly2_no_SMOTE,testData)

#Dataset 2 : Kernel Polynomial Orde 2 dengan 30 Fitur Teratas
svmmod_30_Poly2_with_SMOTE<-
svm(y~ENST00000371095_at+ENST00000371085_at+ENST00000371082_at+ENST000003
49036_at+ENST00000265620_at+

ENST00000263080_at+ENST00000313949_at+ENST00000338783_at+ENST00000369420_
at+ENST00000369417_at+

)306082_at+ENST00000381463_at+ENST00000361035_at+ENST00000371102_
)000371101_at+

)274311_at+ENST00000325945_at+ENST00000371100_at+ENST00000371075_
)000354359_at+
```

```
ENST00000306120_at+ENST00000306090_at+ENST00000375594_at+ENST00000360868_
at+ENST00000358855_at+

ENST00000381936_at+ENST00000359894_at+ENST00000300947_at+ENST00000390597_
at+ENST00000380991_at,
                    data=smote_train,    kernel="polynomial",    cost=16,
degree=2, coef0=0.125, gamma=0.015625, types="response")
svmmod_30_Poly2_no_SMOTE<-
svm(y~ENST00000371095_at+ENST00000371085_at+ENST00000371082_at+ENST000003
49036_at+ENST00000265620_at+

ENST00000263080_at+ENST00000313949_at+ENST00000338783_at+ENST00000369420_
at+ENST00000369417_at+

ENST00000306082_at+ENST00000381463_at+ENST00000361035_at+ENST00000371102_
at+ENST00000371101_at+

ENST00000274311_at+ENST00000325945_at+ENST00000371100_at+ENST00000371075_
at+ENST00000354359_at+

ENST00000306120_at+ENST00000306090_at+ENST00000375594_at+ENST00000360868_
at+ENST00000358855_at+

ENST00000381936_at+ENST00000359894_at+ENST00000300947_at+ENST00000390597_
at+ENST00000380991_at,
                              data=trainData,      kernel="polynomial",
cost=16, degree=2, coef0=0.125, gamma=0.015625, types="response")
prediksi_with_SMOTE_Poly30<-
prediction(svmmod_30_Poly2_with_SMOTE,testData)
prediksi_no_SMOTE_Poly30<-prediction(svmmod_30_Poly2_no_SMOTE,testData)

#Dataset 2 : Kernel Polynomial Orde 2 dengan 40 Fitur Teratas
svmmod_40_Poly2_with_SMOTE<-
svm(y~ENST00000371095_at+ENST00000371085_at+ENST00000371082_at+ENST000003
49036_at+ENST00000265620_at+

ENST00000263080_at+ENST00000313949_at+ENST00000338783_at+ENST00000369420_
at+ENST00000369417_at+

ENST00000306082_at+ENST00000381463_at+ENST00000361035_at+ENST00000371102_
at+ENST00000371101_at+

ENST00000274311_at+ENST00000325945_at+ENST00000371100_at+ENST00000371075_
at+ENST00000354359_at+

ENST00000306120_at+ENST00000306090_at+ENST00000375594_at+ENST00000360868_
at+ENST00000358855_at+

ENST00000381936_at+ENST00000359894_at+ENST00000300947_at+ENST00000390597_
at+ENST00000380991_at+

ENST00000380986_at+ENST00000265028_at+ENST00000359623_at+ENST00000323078_
at+ENST00000283646_at+

ENST00000180173_at+ENST00000244296_at+ENST00000221293_at+ENST00000381251_
at+ENST00000304101_at,
                    data=smote_train,    kernel="polynomial",    cost=16,
, coef0=0.125, gamma=0.015625, types="response")
O_Poly2_no_SMOTE<-
ST00000371095_at+ENST00000371085_at+ENST00000371082_at+ENST000003
+ENST00000265620_at+
```

```
ENST00000263080_at+ENST00000313949_at+ENST00000338783_at+ENST00000369420_
at+ENST00000369417_at+

ENST00000306082_at+ENST00000381463_at+ENST00000361035_at+ENST00000371102_
at+ENST00000371101_at+

ENST00000274311_at+ENST00000325945_at+ENST00000371100_at+ENST00000371075_
at+ENST00000354359_at+

ENST00000306120_at+ENST00000306090_at+ENST00000375594_at+ENST00000360868_
at+ENST00000358855_at+

ENST00000381936_at+ENST00000359894_at+ENST00000300947_at+ENST00000390597_
at+ENST00000380991_at+

ENST00000380986_at+ENST00000265028_at+ENST00000359623_at+ENST00000323078_
at+ENST00000283646_at+

ENST00000180173_at+ENST00000244296_at+ENST00000221293_at+ENST00000381251_
at+ENST00000304101_at,
                                data=trainData,      kernel="polynomial",
cost=16, degree=2, coef0=0.125, gamma=0.015625, types="response")
prediksi_with_SMOTE_Poly40<-
prediction(svmmod_40_Poly2_with_SMOTE,testData)
prediksi_no_SMOTE_Poly40<-prediction(svmmod_40_Poly2_no_SMOTE,testData)


#Dataset 2 : Kernel Polynomial Orde 2 dengan 50 Fitur Teratas
svmmod_50_Poly2_with_SMOTE<-
svm(y~ENST00000371095_at+ENST00000371085_at+ENST00000371082_at+ENST000003
49036_at+ENST00000265620_at+

ENST00000263080_at+ENST00000313949_at+ENST00000338783_at+ENST00000369420_
at+ENST00000369417_at+

ENST00000306082_at+ENST00000381463_at+ENST00000361035_at+ENST00000371102_
at+ENST00000371101_at+

ENST00000274311_at+ENST00000325945_at+ENST00000371100_at+ENST00000371075_
at+ENST00000354359_at+

ENST00000306120_at+ENST00000306090_at+ENST00000375594_at+ENST00000360868_
at+ENST00000358855_at+

ENST00000381936_at+ENST00000359894_at+ENST00000300947_at+ENST00000390597_
at+ENST00000380991_at+

ENST00000380986_at+ENST00000265028_at+ENST00000359623_at+ENST00000323078_
at+ENST00000283646_at+

ENST00000180173_at+ENST00000244296_at+ENST00000221293_at+ENST00000381251_
at+ENST00000304101_at+

ENST00000371761_at+ENST00000217246_at+ENST00000262662_at+ENST00000356692_
at+ENST00000361625_at+

ENST00000253023_at+ENST00000367843_at+ENST00000367840_at+ENST00000367839_
     )0000324705_at,
                  data=smote_train,    kernel="polynomial",    cost=16,
, coef0=0.125, gamma=0.015625, types="response")
)_Poly2_no_SMOTE<-
ST00000371095_at+ENST00000371085_at+ENST00000371082_at+ENST000003
+ENST00000265620_at+
```

```
ENST00000263080_at+ENST00000313949_at+ENST00000338783_at+ENST00000369420_
at+ENST00000369417_at+

ENST00000306082_at+ENST00000381463_at+ENST00000361035_at+ENST00000371102_
at+ENST00000371101_at+

ENST00000274311_at+ENST00000325945_at+ENST00000371100_at+ENST00000371075_
at+ENST00000354359_at+

ENST00000306120_at+ENST00000306090_at+ENST00000375594_at+ENST00000360868_
at+ENST00000358855_at+

ENST00000381936_at+ENST00000359894_at+ENST00000300947_at+ENST00000390597_
at+ENST00000380991_at+

ENST00000380986_at+ENST00000265028_at+ENST00000359623_at+ENST00000323078_
at+ENST00000283646_at+

ENST00000180173_at+ENST00000244296_at+ENST00000221293_at+ENST00000381251_
at+ENST00000304101_at+

ENST00000371761_at+ENST00000217246_at+ENST00000262662_at+ENST00000356692_
at+ENST00000361625_at+

ENST00000253023_at+ENST00000367843_at+ENST00000367840_at+ENST00000367839_
at+ENST00000324705_at,
                              data=trainData,      kernel="polynomial",
cost=16, degree=2, coef0=0.125, gamma=0.015625, types="response")
prediksi_with_SMOTE_Poly50<-
prediction(svmmod_50_Poly2_with_SMOTE,testData)
prediksi_no_SMOTE_Poly50<-prediction(svmmod_50_Poly2_no_SMOTE,testData)

#Dataset 2 : Kernel Polynomial Orde 2 dengan 60 Fitur Teratas
svmmod_60_Poly2_with_SMOTE<-
svm(y~ENST00000371095_at+ENST00000371085_at+ENST00000371082_at+ENST000003
49036_at+ENST00000265620_at+

ENST00000263080_at+ENST00000313949_at+ENST00000338783_at+ENST00000369420_
at+ENST00000369417_at+

ENST00000306082_at+ENST00000381463_at+ENST00000361035_at+ENST00000371102_
at+ENST00000371101_at+

ENST00000274311_at+ENST00000325945_at+ENST00000371100_at+ENST00000371075_
at+ENST00000354359_at+

ENST00000306120_at+ENST00000306090_at+ENST00000375594_at+ENST00000360868_
at+ENST00000358855_at+

ENST00000381936_at+ENST00000359894_at+ENST00000300947_at+ENST00000390597_
at+ENST00000380991_at+

ENST00000380986_at+ENST00000265028_at+ENST00000359623_at+ENST00000323078_
at+ENST00000283646_at+

ENST00000180173_at+ENST00000244296_at+ENST00000221293_at+ENST00000381251_
at+ENST00000304101_at+

    )371761_at+ENST00000217246_at+ENST00000262662_at+ENST00000356692_
    )000361625_at+

    )253023_at+ENST00000367843_at+ENST00000367840_at+ENST00000367839_
    )000324705_at+
```

```
ENST00000312263_at+ENST00000298694_at+ENST00000298693_at+ENST00000354181_
at+ENST00000361379_at+

ENST00000358712_at+ENST00000336092_at+ENST00000382438_at+ENST00000226972_
at+ENST00000312524_at,
                    data=smote_train,    kernel="polynomial",    cost=16,
degree=2, coef0=0.125, gamma=0.015625, types="response")
svmmod_60_Poly2_no_SMOTE<-
svm(y~ENST00000371095_at+ENST00000371085_at+ENST00000371082_at+ENST000003
49036_at+ENST00000265620_at+

ENST00000263080_at+ENST00000313949_at+ENST00000338783_at+ENST00000369420_
at+ENST00000369417_at+

ENST00000306082_at+ENST00000381463_at+ENST00000361035_at+ENST00000371102_
at+ENST00000371101_at+

ENST00000274311_at+ENST00000325945_at+ENST00000371100_at+ENST00000371075_
at+ENST00000354359_at+

ENST00000306120_at+ENST00000306090_at+ENST00000375594_at+ENST00000360868_
at+ENST00000358855_at+

ENST00000381936_at+ENST00000359894_at+ENST00000300947_at+ENST00000390597_
at+ENST00000380991_at+

ENST00000380986_at+ENST00000265028_at+ENST00000359623_at+ENST00000323078_
at+ENST00000283646_at+

ENST00000180173_at+ENST00000244296_at+ENST00000221293_at+ENST00000381251_
at+ENST00000304101_at+

ENST00000371761_at+ENST00000217246_at+ENST00000262662_at+ENST00000356692_
at+ENST00000361625_at+

ENST00000253023_at+ENST00000367843_at+ENST00000367840_at+ENST00000367839_
at+ENST00000324705_at+

ENST00000312263_at+ENST00000298694_at+ENST00000298693_at+ENST00000354181_
at+ENST00000361379_at+

ENST00000358712_at+ENST00000336092_at+ENST00000382438_at+ENST00000226972_
at+ENST00000312524_at,
                                data=trainData,      kernel="polynomial",
cost=16, degree=2, coef0=0.125, gamma=0.015625, types="response")
prediksi_with_SMOTE_Poly60<-
prediction(svmmod_60_Poly2_with_SMOTE,testData)
prediksi_no_SMOTE_Poly60<-prediction(svmmod_60_Poly2_no_SMOTE,testData)

#Dataset 2 : Kernel Polynomial Orde 2 dengan 70 Fitur Teratas
svmmod_70_Poly2_with_SMOTE<-
svm(y~ENST00000371095_at+ENST00000371085_at+ENST00000371082_at+ENST000003
49036_at+ENST00000265620_at+

ENST00000263080_at+ENST00000313949_at+ENST00000338783_at+ENST00000369420_
at+ENST00000369417_at+

)306082_at+ENST00000381463_at+ENST00000361035_at+ENST00000371102_
)000371101_at+

)274311_at+ENST00000325945_at+ENST00000371100_at+ENST00000371075_
)000354359_at+
```

```
ENST00000306120_at+ENST00000306090_at+ENST00000375594_at+ENST00000360868_
at+ENST00000358855_at+

ENST00000381936_at+ENST00000359894_at+ENST00000300947_at+ENST00000390597_
at+ENST00000380991_at+

ENST00000380986_at+ENST00000265028_at+ENST00000359623_at+ENST00000323078_
at+ENST00000283646_at+

ENST00000180173_at+ENST00000244296_at+ENST00000221293_at+ENST00000381251_
at+ENST00000304101_at+

ENST00000371761_at+ENST00000217246_at+ENST00000262662_at+ENST00000356692_
at+ENST00000361625_at+

ENST00000253023_at+ENST00000367843_at+ENST00000367840_at+ENST00000367839_
at+ENST00000324705_at+

ENST00000312263_at+ENST00000298694_at+ENST00000298693_at+ENST00000354181_
at+ENST00000361379_at+

ENST00000358712_at+ENST00000336092_at+ENST00000382438_at+ENST00000226972_
at+ENST00000312524_at+

ENST00000361272_at+ENST00000361231_at+ENST00000298047_at+ENST00000342745_
at+ENST00000244926_at+

ENST00000356505_at+ENST00000323760_at+ENST00000219168_at+ENST00000371534_
at+ENST00000383362_at,
                    data=smote_train,    kernel="polynomial",    cost=16,
degree=2, coef0=0.125, gamma=0.015625, types="response")
svmmod_70_Poly2_no_SMOTE<-
svm(y~ENST00000371095_at+ENST00000371085_at+ENST00000371082_at+ENST000003
49036_at+ENST00000265620_at+

ENST00000263080_at+ENST00000313949_at+ENST00000338783_at+ENST00000369420_
at+ENST00000369417_at+

ENST00000306082_at+ENST00000381463_at+ENST00000361035_at+ENST00000371102_
at+ENST00000371101_at+

ENST00000274311_at+ENST00000325945_at+ENST00000371100_at+ENST00000371075_
at+ENST00000354359_at+

ENST00000306120_at+ENST00000306090_at+ENST00000375594_at+ENST00000360868_
at+ENST00000358855_at+

ENST00000381936_at+ENST00000359894_at+ENST00000300947_at+ENST00000390597_
at+ENST00000380991_at+

ENST00000380986_at+ENST00000265028_at+ENST00000359623_at+ENST00000323078_
at+ENST00000283646_at+

ENST00000180173_at+ENST00000244296_at+ENST00000221293_at+ENST00000381251_
at+ENST00000304101_at+

ENST00000371761_at+ENST00000217246_at+ENST00000262662_at+ENST00000356692_
      )000361625_at+

      )253023_at+ENST00000367843_at+ENST00000367840_at+ENST00000367839_
      )000324705_at+
```

```
ENST00000312263_at+ENST00000298694_at+ENST00000298693_at+ENST00000354181_
at+ENST00000361379_at+

ENST00000358712_at+ENST00000336092_at+ENST00000382438_at+ENST00000226972_
at+ENST00000312524_at+

ENST00000361272_at+ENST00000361231_at+ENST00000298047_at+ENST00000342745_
at+ENST00000244926_at+

ENST00000356505_at+ENST00000323760_at+ENST00000219168_at+ENST00000371534_
at+ENST00000383362_at,
                                  data=trainData,      kernel="polynomial",
cost=16, degree=2, coef0=0.125, gamma=0.015625, types="response")
prediksi_with_SMOTE_Poly70<-
prediction(svmmod_70_Poly2_with_SMOTE,testData)
prediksi_no_SMOTE_Poly70<-prediction(svmmod_70_Poly2_no_SMOTE,testData)

#Dataset 2 : Kernel Polynomial Orde 2 dengan 80 Fitur Teratas
svmmod_80_Poly2_with_SMOTE<-
svm(y~ENST00000371095_at+ENST00000371085_at+ENST00000371082_at+ENST000003
49036_at+ENST00000265620_at+

ENST00000263080_at+ENST00000313949_at+ENST00000338783_at+ENST00000369420_
at+ENST00000369417_at+

ENST00000306082_at+ENST00000381463_at+ENST00000361035_at+ENST00000371102_
at+ENST00000371101_at+

ENST00000274311_at+ENST00000325945_at+ENST00000371100_at+ENST00000371075_
at+ENST00000354359_at+

ENST00000306120_at+ENST00000306090_at+ENST00000375594_at+ENST00000360868_
at+ENST00000358855_at+

ENST00000381936_at+ENST00000359894_at+ENST00000300947_at+ENST00000390597_
at+ENST00000380991_at+

ENST00000380986_at+ENST00000265028_at+ENST00000359623_at+ENST00000323078_
at+ENST00000283646_at+

ENST00000180173_at+ENST00000244296_at+ENST00000221293_at+ENST00000381251_
at+ENST00000304101_at+

ENST00000371761_at+ENST00000217246_at+ENST00000262662_at+ENST00000356692_
at+ENST00000361625_at+

ENST00000253023_at+ENST00000367843_at+ENST00000367840_at+ENST00000367839_
at+ENST00000324705_at+

ENST00000312263_at+ENST00000298694_at+ENST00000298693_at+ENST00000354181_
at+ENST00000361379_at+

ENST00000358712_at+ENST00000336092_at+ENST00000382438_at+ENST00000226972_
at+ENST00000312524_at+

ENST00000372264_at+ENST00000362031_at+ENST00000355110_at+ENST00000332302_
at+ENST00000265395_at+

    )376419_at+ENST00000376417_at+ENST00000277124_at+ENST00000381753_
    )000313860_at+

    )383367_at+ENST00000318809_at+ENST00000308370_at+ENST00000383179_
    )000243562_at+
```

```
ENST00000252576_at+ENST00000381917_at+ENST00000375493_at+ENST00000204005_
at+ENST00000301788_at,
                    data=smote_train,    kernel="polynomial",    cost=16,
degree=2, coef0=0.125, gamma=0.015625, types="response")
svmmod_80_Poly2_no_SMOTE<-
svm(y~ENST00000371095_at+ENST00000371085_at+ENST00000371082_at+ENST000003
49036_at+ENST00000265620_at+

ENST00000263080_at+ENST00000313949_at+ENST00000338783_at+ENST00000369420_
at+ENST00000369417_at+

ENST00000306082_at+ENST00000381463_at+ENST00000361035_at+ENST00000371102_
at+ENST00000371101_at+

ENST00000274311_at+ENST00000325945_at+ENST00000371100_at+ENST00000371075_
at+ENST00000354359_at+

ENST00000306120_at+ENST00000306090_at+ENST00000375594_at+ENST00000360868_
at+ENST00000358855_at+

ENST00000381936_at+ENST00000359894_at+ENST00000300947_at+ENST00000390597_
at+ENST00000380991_at+

ENST00000380986_at+ENST00000265028_at+ENST00000359623_at+ENST00000323078_
at+ENST00000283646_at+

ENST00000180173_at+ENST00000244296_at+ENST00000221293_at+ENST00000381251_
at+ENST00000304101_at+

ENST00000371761_at+ENST00000217246_at+ENST00000262662_at+ENST00000356692_
at+ENST00000361625_at+

ENST00000253023_at+ENST00000367843_at+ENST00000367840_at+ENST00000367839_
at+ENST00000324705_at+

ENST00000312263_at+ENST00000298694_at+ENST00000298693_at+ENST00000354181_
at+ENST00000361379_at+

ENST00000358712_at+ENST00000336092_at+ENST00000382438_at+ENST00000226972_
at+ENST00000312524_at+

ENST00000372264_at+ENST00000362031_at+ENST00000355110_at+ENST00000332302_
at+ENST00000265395_at+

ENST00000376419_at+ENST00000376417_at+ENST00000277124_at+ENST00000381753_
at+ENST00000313860_at+

ENST00000383367_at+ENST00000318809_at+ENST00000308370_at+ENST00000383179_
at+ENST00000243562_at+

ENST00000252576_at+ENST00000381917_at+ENST00000375493_at+ENST00000204005_
at+ENST00000301788_at,
                                data=trainData,      kernel="polynomial",
cost=16, degree=2, coef0=0.125, gamma=0.015625, types="response")
prediksi_with_SMOTE_Poly80<-prediction(svmmod_80_Poly2,testData)
prediksi_no_SMOTE_Poly80<-prediction(svmmod_80_Poly2_no_SMOTE,testData)
```

2 : Kernel Polynomial Orde 2 dengan 90 Fitur Teratas

```
)_Poly2_with_SMOTE<-
ST00000371095_at+ENST00000371085_at+ENST00000371082_at+ENST000003
+ENST00000265620_at+
```

```
ENST00000263080_at+ENST00000313949_at+ENST00000338783_at+ENST00000369420_
at+ENST00000369417_at+

ENST00000306082_at+ENST00000381463_at+ENST00000361035_at+ENST00000371102_
at+ENST00000371101_at+

ENST00000274311_at+ENST00000325945_at+ENST00000371100_at+ENST00000371075_
at+ENST00000354359_at+

ENST00000306120_at+ENST00000306090_at+ENST00000375594_at+ENST00000360868_
at+ENST00000358855_at+

ENST00000381936_at+ENST00000359894_at+ENST00000300947_at+ENST00000390597_
at+ENST00000380991_at+

ENST00000380986_at+ENST00000265028_at+ENST00000359623_at+ENST00000323078_
at+ENST00000283646_at+

ENST00000180173_at+ENST00000244296_at+ENST00000221293_at+ENST00000381251_
at+ENST00000304101_at+

ENST00000371761_at+ENST00000217246_at+ENST00000262662_at+ENST00000356692_
at+ENST00000361625_at+

ENST00000253023_at+ENST00000367843_at+ENST00000367840_at+ENST00000367839_
at+ENST00000324705_at+

ENST00000312263_at+ENST00000298694_at+ENST00000298693_at+ENST00000354181_
at+ENST00000361379_at+

ENST00000358712_at+ENST00000336092_at+ENST00000382438_at+ENST00000226972_
at+ENST00000312524_at+

ENST00000372264_at+ENST00000362031_at+ENST00000355110_at+ENST00000332302_
at+ENST00000265395_at+

ENST00000376419_at+ENST00000376417_at+ENST00000277124_at+ENST00000381753_
at+ENST00000313860_at+

ENST00000383367_at+ENST00000318809_at+ENST00000308370_at+ENST00000383179_
at+ENST00000243562_at+

ENST00000252576_at+ENST00000381917_at+ENST00000375493_at+ENST00000204005_
at+ENST00000301788_at+

ENST00000361272_at+ENST00000361231_at+ENST00000298047_at+ENST00000342745_
at+ENST00000244926_at+

ENST00000356505_at+ENST00000323760_at+ENST00000219168_at+ENST00000371534_
at+ENST00000383362_at,
                    data=smote_train,    kernel="polynomial",    cost=16,
degree=2, coef0=0.125, gamma=0.015625, types="response")
svmmod_90_Poly2_no_SMOTE<-
svm(y~ENST00000371095_at+ENST00000371085_at+ENST00000371082_at+ENST000003
49036_at+ENST00000265620_at+

ENST00000263080_at+ENST00000313949_at+ENST00000338783_at+ENST00000369420_
          )000369417_at+

          )306082_at+ENST00000381463_at+ENST00000361035_at+ENST00000371102_
          )000371101_at+
```

```
ENST00000274311_at+ENST00000325945_at+ENST00000371100_at+ENST00000371075_
at+ENST00000354359_at+

ENST00000306120_at+ENST00000306090_at+ENST00000375594_at+ENST00000360868_
at+ENST00000358855_at+

ENST00000381936_at+ENST00000359894_at+ENST00000300947_at+ENST00000390597_
at+ENST00000380991_at+

ENST00000380986_at+ENST00000265028_at+ENST00000359623_at+ENST00000323078_
at+ENST00000283646_at+

ENST00000180173_at+ENST00000244296_at+ENST00000221293_at+ENST00000381251_
at+ENST00000304101_at+

ENST00000371761_at+ENST00000217246_at+ENST00000262662_at+ENST00000356692_
at+ENST00000361625_at+

ENST00000253023_at+ENST00000367843_at+ENST00000367840_at+ENST00000367839_
at+ENST00000324705_at+

ENST00000312263_at+ENST00000298694_at+ENST00000298693_at+ENST00000354181_
at+ENST00000361379_at+

ENST00000358712_at+ENST00000336092_at+ENST00000382438_at+ENST00000226972_
at+ENST00000312524_at+

ENST00000372264_at+ENST00000362031_at+ENST00000355110_at+ENST00000332302_
at+ENST00000265395_at+

ENST00000376419_at+ENST00000376417_at+ENST00000277124_at+ENST00000381753_
at+ENST00000313860_at+

ENST00000383367_at+ENST00000318809_at+ENST00000308370_at+ENST00000383179_
at+ENST00000243562_at+

ENST00000252576_at+ENST00000381917_at+ENST00000375493_at+ENST00000204005_
at+ENST00000301788_at+

ENST00000361272_at+ENST00000361231_at+ENST00000298047_at+ENST00000342745_
at+ENST00000244926_at+

ENST00000356505_at+ENST00000323760_at+ENST00000219168_at+ENST00000371534_
at+ENST00000383362_at,
                                  data=trainData,      kernel="polynomial",
cost=16, degree=2, coef0=0.125, gamma=0.015625, types="response")
prediksi_with_SMOTE_Poly90<-
prediction(svmmod_90_Poly2_with_SMOTE,testData)
prediksi_no_SMOTE_Poly90<-prediction(svmmod_90_Poly2_no_SMOTE,testData)

#Dataset 2 : Kernel Polynomial Orde 2 dengan 100 Fitur Teratas
svmmod_100_Poly2_with_SMOTE<-
svm(y~ENST00000371095_at+ENST00000371085_at+ENST00000371082_at+ENST000003
49036_at+ENST00000265620_at+

ENST00000263080_at+ENST00000313949_at+ENST00000338783_at+ENST00000369420_
at+ENST00000369417_at+

    )306082_at+ENST00000381463_at+ENST00000361035_at+ENST00000371102_
    )000371101_at+

    )274311_at+ENST00000325945_at+ENST00000371100_at+ENST00000371075_
    )000354359_at+
```

```
ENST00000306120_at+ENST00000306090_at+ENST00000375594_at+ENST00000360868_
at+ENST00000358855_at+

ENST00000381936_at+ENST00000359894_at+ENST00000300947_at+ENST00000390597_
at+ENST00000380991_at+

ENST00000380986_at+ENST00000265028_at+ENST00000359623_at+ENST00000323078_
at+ENST00000283646_at+

ENST00000180173_at+ENST00000244296_at+ENST00000221293_at+ENST00000381251_
at+ENST00000304101_at+

ENST00000371761_at+ENST00000217246_at+ENST00000262662_at+ENST00000356692_
at+ENST00000361625_at+

ENST00000253023_at+ENST00000367843_at+ENST00000367840_at+ENST00000367839_
at+ENST00000324705_at+

ENST00000312263_at+ENST00000298694_at+ENST00000298693_at+ENST00000354181_
at+ENST00000361379_at+

ENST00000358712_at+ENST00000336092_at+ENST00000382438_at+ENST00000226972_
at+ENST00000312524_at+

ENST00000372264_at+ENST00000362031_at+ENST00000355110_at+ENST00000332302_
at+ENST00000265395_at+

ENST00000376419_at+ENST00000376417_at+ENST00000277124_at+ENST00000381753_
at+ENST00000313860_at+

ENST00000383367_at+ENST00000318809_at+ENST00000308370_at+ENST00000383179_
at+ENST00000243562_at+

ENST00000252576_at+ENST00000381917_at+ENST00000375493_at+ENST00000204005_
at+ENST00000301788_at+

ENST00000361272_at+ENST00000361231_at+ENST00000298047_at+ENST00000342745_
at+ENST00000244926_at+

ENST00000356505_at+ENST00000323760_at+ENST00000219168_at+ENST00000371534_
at+ENST00000383362_at+

ENST00000383360_at+ENST00000383359_at+ENST00000370109_at+ENST00000383178_
at+ENST00000383177_at+

ENST00000370107_at+ENST00000371646_at+ENST00000223864_at+ENST00000371554_
at+ENST00000342173_at,
                      data=smote_train,   kernel="polynomial",   cost=16,
degree=2, coef0=0.125, gamma=0.015625, types="response")
svmmod_100_Poly2_no_SMOTE<-
svm(y~ENST00000371095_at+ENST00000371085_at+ENST00000371082_at+ENST000003
49036_at+ENST00000265620_at+

ENST00000263080_at+ENST00000313949_at+ENST00000338783_at+ENST00000369420_
at+ENST00000369417_at+

ENST00000306082_at+ENST00000381463_at+ENST00000361035_at+ENST00000371102_
      0000371101_at+

      0274311_at+ENST00000325945_at+ENST00000371100_at+ENST00000371075_
      0000354359_at+
```

```
ENST00000306120_at+ENST00000306090_at+ENST00000375594_at+ENST00000360868_
at+ENST00000358855_at+

ENST00000381936_at+ENST00000359894_at+ENST00000300947_at+ENST00000390597_
at+ENST00000380991_at+

ENST00000380986_at+ENST00000265028_at+ENST00000359623_at+ENST00000323078_
at+ENST00000283646_at+

ENST00000180173_at+ENST00000244296_at+ENST00000221293_at+ENST00000381251_
at+ENST00000304101_at+

ENST00000371761_at+ENST00000217246_at+ENST00000262662_at+ENST00000356692_
at+ENST00000361625_at+

ENST00000253023_at+ENST00000367843_at+ENST00000367840_at+ENST00000367839_
at+ENST00000324705_at+

ENST00000312263_at+ENST00000298694_at+ENST00000298693_at+ENST00000354181_
at+ENST00000361379_at+

ENST00000358712_at+ENST00000336092_at+ENST00000382438_at+ENST00000226972_
at+ENST00000312524_at+

ENST00000372264_at+ENST00000362031_at+ENST00000355110_at+ENST00000332302_
at+ENST00000265395_at+

ENST00000376419_at+ENST00000376417_at+ENST00000277124_at+ENST00000381753_
at+ENST00000313860_at+

ENST00000383367_at+ENST00000318809_at+ENST00000308370_at+ENST00000383179_
at+ENST00000243562_at+

ENST00000252576_at+ENST00000381917_at+ENST00000375493_at+ENST00000204005_
at+ENST00000301788_at+

ENST00000361272_at+ENST00000361231_at+ENST00000298047_at+ENST00000342745_
at+ENST00000244926_at+

ENST00000356505_at+ENST00000323760_at+ENST00000219168_at+ENST00000371534_
at+ENST00000383362_at+

ENST00000383360_at+ENST00000383359_at+ENST00000370109_at+ENST00000383178_
at+ENST00000383177_at+

ENST00000370107_at+ENST00000371646_at+ENST00000223864_at+ENST00000371554_
at+ENST00000342173_at,
                                  data=trainData,     kernel="polynomial",
cost=16, degree=2, coef0=0.125, gamma=0.015625, types="response")
prediksi_with_SMOTE_Poly100<-
prediction(svmmod_100_Poly2_with_SMOTE,testData)
prediksi_no_SMOTE_Poly100<-prediction(svmmod_100_Poly2_no_SMOTE,testData)
#Untuk subset gen terbaik SVM Linier
#Dataset 2 : Untuk 40 Fitur Teratas Kernel Linier Menggunakan SMOTE
pb <- txtProgressBar(min = 0, max = 1000, style = 3)
accuracy.linear.iterasi <- NULL
for (j in 1:1000) {
        <-NULL
        smote<-createFolds(smote_train$y, 10, list=FALSE)
        in 1:10){
        x=which(folds.smote==i)
        ning_folds = smote_train[-index,]
        _folds = smote_train[index,]
```

```
    svmmod_40_linear_with_SMOTE<-
svm(y~ENST00000371095_at+ENST00000371085_at+ENST00000371082_at+ENST000003
49036_at+ENST00000265620_at+

ENST00000263080_at+ENST00000313949_at+ENST00000338783_at+ENST00000369420_
at+ENST00000369417_at+

ENST00000306082_at+ENST00000381463_at+ENST00000361035_at+ENST00000371102_
at+ENST00000371101_at+

ENST00000274311_at+ENST00000325945_at+ENST00000371100_at+ENST00000371075_
at+ENST00000354359_at+

ENST00000306120_at+ENST00000306090_at+ENST00000375594_at+ENST00000360868_
at+ENST00000358855_at+

ENST00000381936_at+ENST00000359894_at+ENST00000300947_at+ENST00000390597_
at+ENST00000380991_at+

ENST00000380986_at+ENST00000265028_at+ENST00000359623_at+ENST00000323078_
at+ENST00000283646_at+

ENST00000180173_at+ENST00000244296_at+ENST00000221293_at+ENST00000381251_
at+ENST00000304101_at,
                          data=training_folds,            kernel="linear",
cost=1,types="response")
    pred<-predict(svmmod_40_linear_with_SMOTE,    newdata    =    testData,
type="response")
    y_act<-testData$y
    #Menghitung Akurasi
    acc_cv[i] = mean(pred==y_act)
  }
  accuracy.linear = mean(as.numeric(acc_cv))
  accuracy.linear.iterasi[j]=mean(as.numeric(accuracy.linear))
  accuracy.linear.result=mean(as.numeric(accuracy.linear.iterasi))
  setTxtProgressBar(pb,j)
}
accuracy.linear.result

#Dataset 2 : Untuk 60 Fitur Teratas Kernel Linier Menggunakan SMOTE
pb <- txtProgressBar(min = 0, max = 1000, style = 3)
accuracy.linear.iterasi <- NULL
for (j in 1:1000) {
  acc_cv<-NULL
  folds.smote<-createFolds(smote_train$y, 10, list=FALSE)
  for(i in 1:10){
    index=which(folds.smote==i)
    training_folds = smote_train[-index,]
    test_folds = smote_train[index,]
    svmmod_60_linear_with_SMOTE<-
svm(y~ENST00000371095_at+ENST00000371085_at+ENST00000371082_at+ENST000003
49036_at+ENST00000265620_at+

ENST00000263080_at+ENST00000313949_at+ENST00000338783_at+ENST00000369420_
at+ENST00000369417_at+

ENST00000306082_at+ENST00000381463_at+ENST00000361035_at+ENST00000371102_
at+ENST00000371101_at+

    )274311_at+ENST00000325945_at+ENST00000371100_at+ENST00000371075_
    )000354359_at+

    )306120_at+ENST00000306090_at+ENST00000375594_at+ENST00000360868_
    )000358855_at+
```

```
ENST00000381936_at+ENST00000359894_at+ENST00000300947_at+ENST00000390597_
at+ENST00000380991_at+

ENST00000380986_at+ENST00000265028_at+ENST00000359623_at+ENST00000323078_
at+ENST00000283646_at+

ENST00000180173_at+ENST00000244296_at+ENST00000221293_at+ENST00000381251_
at+ENST00000304101_at+

ENST00000371761_at+ENST00000217246_at+ENST00000262662_at+ENST00000356692_
at+ENST00000361625_at+

ENST00000253023_at+ENST00000367843_at+ENST00000367840_at+ENST00000367839_
at+ENST00000324705_at+

ENST00000312263_at+ENST00000298694_at+ENST00000298693_at+ENST00000354181_
at+ENST00000361379_at+

ENST00000358712_at+ENST00000336092_at+ENST00000382438_at+ENST00000226972_
at+ENST00000312524_at,
                          data=training_folds,           kernel="linear",
cost=1,types="response")
    pred<-predict(svmmod_60_linear_with_SMOTE,   newdata   =   testData,
type="response")
    y_act<-testData$y
    #Menghitung Akurasi
    acc_cv[i] = mean(pred==y_act)
  }
  accuracy.linear = mean(as.numeric(acc_cv))
  accuracy.linear.iterasi[j]=mean(as.numeric(accuracy.linear))
  accuracy.linear.result=mean(as.numeric(accuracy.linear.iterasi))
  setTxtProgressBar(pb,j)
}
accuracy.linear.result

#Dataset 2 : Untuk 70 Fitur Teratas Kernel Linier Menggunakan SMOTE
pb <- txtProgressBar(min = 0, max = 1000, style = 3)
accuracy.linear.iterasi <- NULL
for (j in 1:1000) {
  acc_cv<-NULL
  folds.smote<-createFolds(smote_train$y, 10, list=FALSE)
  for(i in 1:10){
    index=which(folds.smote==i)
    training_folds = smote_train[-index,]
    test_folds = smote_train[index,]
    svmmod_70_linear_with_SMOTE<-
svm(y~ENST00000371095_at+ENST00000371085_at+ENST00000371082_at+ENST000003
49036_at+ENST00000265620_at+

ENST00000263080_at+ENST00000313949_at+ENST00000338783_at+ENST00000369420_
at+ENST00000369417_at+

ENST00000306082_at+ENST00000381463_at+ENST00000361035_at+ENST00000371102_
at+ENST00000371101_at+

ENST00000274311_at+ENST00000325945_at+ENST00000371100_at+ENST00000371075_
at+ENST00000354359_at+

    )306120_at+ENST00000306090_at+ENST00000375594_at+ENST00000360868_
    )000358855_at+

    )381936_at+ENST00000359894_at+ENST00000300947_at+ENST00000390597_
    )000380991_at+
```

```
ENST00000380986_at+ENST00000265028_at+ENST00000359623_at+ENST00000323078_
at+ENST00000283646_at+

ENST00000180173_at+ENST00000244296_at+ENST00000221293_at+ENST00000381251_
at+ENST00000304101_at+

ENST00000371761_at+ENST00000217246_at+ENST00000262662_at+ENST00000356692_
at+ENST00000361625_at+

ENST00000253023_at+ENST00000367843_at+ENST00000367840_at+ENST00000367839_
at+ENST00000324705_at+

ENST00000312263_at+ENST00000298694_at+ENST00000298693_at+ENST00000354181_
at+ENST00000361379_at+

ENST00000358712_at+ENST00000336092_at+ENST00000382438_at+ENST00000226972_
at+ENST00000312524_at+

ENST00000361272_at+ENST00000361231_at+ENST00000298047_at+ENST00000342745_
at+ENST00000244926_at+

ENST00000356505_at+ENST00000323760_at+ENST00000219168_at+ENST00000371534_
at+ENST00000383362_at,
                          data=training_folds,            kernel="linear",
cost=1,types="response")
    pred<-predict(svmmod_70_linear_with_SMOTE,    newdata    =    testData,
type="response")
    y_act<-testData$y
    #Menghitung Akurasi
    acc_cv[i] = mean(pred==y_act)
  }
  accuracy.linear = mean(as.numeric(acc_cv))
  accuracy.linear.iterasi[j]=mean(as.numeric(accuracy.linear))
  accuracy.linear.result=mean(as.numeric(accuracy.linear.iterasi))
  setTxtProgressBar(pb,j)
}
accuracy.linear.result

#Dataset 2 : Untuk 80 Fitur Teratas Kernel Linier Menggunakan SMOTE
pb <- txtProgressBar(min = 0, max = 1000, style = 3)
accuracy.linear.iterasi <- NULL
for (j in 1:1000) {
  acc_cv<-NULL
  folds.smote<-createFolds(smote_train$y, 10, list=FALSE)
  for(i in 1:10){
    index=which(folds.smote==i)
    training_folds = smote_train[-index,]
    test_folds = smote_train[index,]
    svmmod_80_linear_with_SMOTE<-
svm(y~ENST00000371095_at+ENST00000371085_at+ENST00000371082_at+ENST000003
49036_at+ENST00000265620_at+

ENST00000263080_at+ENST00000313949_at+ENST00000338783_at+ENST00000369420_
at+ENST00000369417_at+

ENST00000306082_at+ENST00000381463_at+ENST00000361035_at+ENST00000371102_
at+ENST00000371101_at+

                    )274311_at+ENST00000325945_at+ENST00000371100_at+ENST00000371075_
                    )000354359_at+

                    )306120_at+ENST00000306090_at+ENST00000375594_at+ENST00000360868_
                    )000358855_at+
```

```
ENST00000381936_at+ENST00000359894_at+ENST00000300947_at+ENST00000390597_
at+ENST00000380991_at+

ENST00000380986_at+ENST00000265028_at+ENST00000359623_at+ENST00000323078_
at+ENST00000283646_at+

ENST00000180173_at+ENST00000244296_at+ENST00000221293_at+ENST00000381251_
at+ENST00000304101_at+

ENST00000371761_at+ENST00000217246_at+ENST00000262662_at+ENST00000356692_
at+ENST00000361625_at+

ENST00000253023_at+ENST00000367843_at+ENST00000367840_at+ENST00000367839_
at+ENST00000324705_at+

ENST00000312263_at+ENST00000298694_at+ENST00000298693_at+ENST00000354181_
at+ENST00000361379_at+

ENST00000358712_at+ENST00000336092_at+ENST00000382438_at+ENST00000226972_
at+ENST00000312524_at+

ENST00000372264_at+ENST00000362031_at+ENST00000355110_at+ENST00000332302_
at+ENST00000265395_at+

ENST00000376419_at+ENST00000376417_at+ENST00000277124_at+ENST00000381753_
at+ENST00000313860_at+

ENST00000383367_at+ENST00000318809_at+ENST00000308370_at+ENST00000383179_
at+ENST00000243562_at+

ENST00000252576_at+ENST00000381917_at+ENST00000375493_at+ENST00000204005_
at+ENST00000301788_at,
                          data=training_folds,           kernel="linear",
cost=1,types="response")
    pred<-predict(svmmod_80_linear_with_SMOTE,   newdata   =   testData,
type="response")
    y_act<-testData$y
    #Menghitung Akurasi
    acc_cv[i] = mean(pred==y_act)
  }
  accuracy.linear = mean(as.numeric(acc_cv))
  accuracy.linear.iterasi[j]=mean(as.numeric(accuracy.linear))
  accuracy.linear.result=mean(as.numeric(accuracy.linear.iterasi))
  setTxtProgressBar(pb,j)
}
accuracy.linear.result

#Dataset 2 : Untuk 80 Fitur Teratas Kernel Linier Tanpa Menggunakan SMOTE
pb <- txtProgressBar(min = 0, max = 1000, style = 3)
accuracy.linear.iterasi <- NULL
for (j in 1:1000) {
  acc_cv<-NULL
  folds.train<-createFolds(trainData$y, 10, list=FALSE)
  for(i in 1:10){
    index=which(folds.train==i)
    training_folds = trainData[-index,]
    test_folds = trainData[index,]
        od_80_linear_with_SMOTE<-
        ST00000371095_at+ENST00000371085_at+ENST00000371082_at+ENST000003
        +ENST00000265620_at+

        0263080_at+ENST00000313949_at+ENST00000338783_at+ENST00000369420_
        0000369417_at+
```

```
ENST00000306082_at+ENST00000381463_at+ENST00000361035_at+ENST00000371102_
at+ENST00000371101_at+

ENST00000274311_at+ENST00000325945_at+ENST00000371100_at+ENST00000371075_
at+ENST00000354359_at+

ENST00000306120_at+ENST00000306090_at+ENST00000375594_at+ENST00000360868_
at+ENST00000358855_at+

ENST00000381936_at+ENST00000359894_at+ENST00000300947_at+ENST00000390597_
at+ENST00000380991_at+

ENST00000380986_at+ENST00000265028_at+ENST00000359623_at+ENST00000323078_
at+ENST00000283646_at+

ENST00000180173_at+ENST00000244296_at+ENST00000221293_at+ENST00000381251_
at+ENST00000304101_at+

ENST00000371761_at+ENST00000217246_at+ENST00000262662_at+ENST00000356692_
at+ENST00000361625_at+

ENST00000253023_at+ENST00000367843_at+ENST00000367840_at+ENST00000367839_
at+ENST00000324705_at+

ENST00000312263_at+ENST00000298694_at+ENST00000298693_at+ENST00000354181_
at+ENST00000361379_at+

ENST00000358712_at+ENST00000336092_at+ENST00000382438_at+ENST00000226972_
at+ENST00000312524_at+

ENST00000372264_at+ENST00000362031_at+ENST00000355110_at+ENST00000332302_
at+ENST00000265395_at+

ENST00000376419_at+ENST00000376417_at+ENST00000277124_at+ENST00000381753_
at+ENST00000313860_at+

ENST00000383367_at+ENST00000318809_at+ENST00000308370_at+ENST00000383179_
at+ENST00000243562_at+

ENST00000252576_at+ENST00000381917_at+ENST00000375493_at+ENST00000204005_
at+ENST00000301788_at,
                                    data=training_folds,
kernel="linear", cost=1,types="response")
    pred<-predict(svmmod_80_linear_with_SMOTE,   newdata   =   testData,
type="response")
    y_act<-testData$y
    #Menghitung Akurasi
    acc_cv[i] = mean(pred==y_act)
  }
  accuracy.linear = mean(as.numeric(acc_cv))
  accuracy.linear.iterasi[j]=mean(as.numeric(accuracy.linear))
  accuracy.linear.result=mean(as.numeric(accuracy.linear.iterasi))
  setTxtProgressBar(pb,j)
}
accuracy.linear.result

#Dataset 2 : Untuk 90 Fitur Teratas Kernel Linier Tanpa Menggunakan SMOTE
        tProgressBar(min = 0, max = 1000, style = 3)
        .linear.iterasi <- NULL
        n 1:1000) {
        <-NULL
        train<-createFolds(trainData$y, 10, list=FALSE)
        in 1:10){
```

```
    index=which(folds.train==i)
    training_folds = trainData[-index,]
    test_folds = trainData[index,]
    svmmod_90_linear_no_SMOTE<-
svm(y~ENST00000371095_at+ENST00000371085_at+ENST00000371082_at+ENST000003
49036_at+ENST00000265620_at+

ENST00000263080_at+ENST00000313949_at+ENST00000338783_at+ENST00000369420_
at+ENST00000369417_at+

ENST00000306082_at+ENST00000381463_at+ENST00000361035_at+ENST00000371102_
at+ENST00000371101_at+

ENST00000274311_at+ENST00000325945_at+ENST00000371100_at+ENST00000371075_
at+ENST00000354359_at+

ENST00000306120_at+ENST00000306090_at+ENST00000375594_at+ENST00000360868_
at+ENST00000358855_at+

ENST00000381936_at+ENST00000359894_at+ENST00000300947_at+ENST00000390597_
at+ENST00000380991_at+

ENST00000380986_at+ENST00000265028_at+ENST00000359623_at+ENST00000323078_
at+ENST00000283646_at+

ENST00000180173_at+ENST00000244296_at+ENST00000221293_at+ENST00000381251_
at+ENST00000304101_at+

ENST00000371761_at+ENST00000217246_at+ENST00000262662_at+ENST00000356692_
at+ENST00000361625_at+

ENST00000253023_at+ENST00000367843_at+ENST00000367840_at+ENST00000367839_
at+ENST00000324705_at+

ENST00000312263_at+ENST00000298694_at+ENST00000298693_at+ENST00000354181_
at+ENST00000361379_at+

ENST00000358712_at+ENST00000336092_at+ENST00000382438_at+ENST00000226972_
at+ENST00000312524_at+

ENST00000372264_at+ENST00000362031_at+ENST00000355110_at+ENST00000332302_
at+ENST00000265395_at+

ENST00000376419_at+ENST00000376417_at+ENST00000277124_at+ENST00000381753_
at+ENST00000313860_at+

ENST00000383367_at+ENST00000318809_at+ENST00000308370_at+ENST00000383179_
at+ENST00000243562_at+

ENST00000252576_at+ENST00000381917_at+ENST00000375493_at+ENST00000204005_
at+ENST00000301788_at+

ENST00000361272_at+ENST00000361231_at+ENST00000298047_at+ENST00000342745_
at+ENST00000244926_at+

ENST00000356505_at+ENST00000323760_at+ENST00000219168_at+ENST00000371534_
at+ENST00000383362_at,
                                    data=training_folds,
linear", cost=1,types="response")
<-predict(svmmod_90_linear_no_SMOTE,    newdata    =    testData,
sponse")
t<-testData$y
ghitung Akurasi
cv[i] = mean(pred==y_act)
```

```
  }
  accuracy.linear = mean(as.numeric(acc_cv))
  accuracy.linear.iterasi[j]=mean(as.numeric(accuracy.linear))
  accuracy.linear.result=mean(as.numeric(accuracy.linear.iterasi))
  setTxtProgressBar(pb,j)
}
accuracy.linear.result

#Dataset 2 : Untuk 100 Fitur Teratas Kernel Linier Tanpa Menggunakan SMOTE
pb <- txtProgressBar(min = 0, max = 1000, style = 3)
accuracy.linear.iterasi <- NULL
for (j in 1:1000) {
  acc_cv<-NULL
  folds.train<-createFolds(trainData$y, 10, list=FALSE)
  for(i in 1:10){
    index=which(folds.train==i)
    training_folds = trainData[-index,]
    test_folds = trainData[index,]
    svmmod_100_linear_no_SMOTE<-
svm(y~ENST00000371095_at+ENST00000371085_at+ENST00000371082_at+ENST000003
49036_at+ENST00000265620_at+

ENST00000263080_at+ENST00000313949_at+ENST00000338783_at+ENST00000369420_
at+ENST00000369417_at+

ENST00000306082_at+ENST00000381463_at+ENST00000361035_at+ENST00000371102_
at+ENST00000371101_at+

ENST00000274311_at+ENST00000325945_at+ENST00000371100_at+ENST00000371075_
at+ENST00000354359_at+

ENST00000306120_at+ENST00000306090_at+ENST00000375594_at+ENST00000360868_
at+ENST00000358855_at+

ENST00000381936_at+ENST00000359894_at+ENST00000300947_at+ENST00000390597_
at+ENST00000380991_at+

ENST00000380986_at+ENST00000265028_at+ENST00000359623_at+ENST00000323078_
at+ENST00000283646_at+

ENST00000180173_at+ENST00000244296_at+ENST00000221293_at+ENST00000381251_
at+ENST00000304101_at+

ENST00000371761_at+ENST00000217246_at+ENST00000262662_at+ENST00000356692_
at+ENST00000361625_at+

ENST00000253023_at+ENST00000367843_at+ENST00000367840_at+ENST00000367839_
at+ENST00000324705_at+

ENST00000312263_at+ENST00000298694_at+ENST00000298693_at+ENST00000354181_
at+ENST00000361379_at+

ENST00000358712_at+ENST00000336092_at+ENST00000382438_at+ENST00000226972_
at+ENST00000312524_at+

ENST00000372264_at+ENST00000362031_at+ENST00000355110_at+ENST00000332302_
at+ENST00000265395_at+

)376419_at+ENST00000376417_at+ENST00000277124_at+ENST00000381753_
)000313860_at+

)383367_at+ENST00000318809_at+ENST00000308370_at+ENST00000383179_
)000243562_at+
```

```
ENST00000252576_at+ENST00000381917_at+ENST00000375493_at+ENST00000204005_
at+ENST00000301788_at+

ENST00000361272_at+ENST00000361231_at+ENST00000298047_at+ENST00000342745_
at+ENST00000244926_at+

ENST00000356505_at+ENST00000323760_at+ENST00000219168_at+ENST00000371534_
at+ENST00000383362_at+

ENST00000383360_at+ENST00000383359_at+ENST00000370109_at+ENST00000383178_
at+ENST00000383177_at+

ENST00000370107_at+ENST00000371646_at+ENST00000223864_at+ENST00000371554_
at+ENST00000342173_at,
                                    data=training_folds,
kernel="linear", cost=1,types="response")
    pred<-predict(svmmod_100_linear_no_SMOTE,    newdata    =    testData,
type="response")
    y_act<-testData$y
    #Menghitung Akurasi
    acc_cv[i] = mean(pred==y_act)
  }
  accuracy.linear = mean(as.numeric(acc_cv))
  accuracy.linear.iterasi[j]=mean(as.numeric(accuracy.linear))
  accuracy.linear.result=mean(as.numeric(accuracy.linear.iterasi))
  setTxtProgressBar(pb,j)
}
accuracy.linear.result


#Untuk Subset Gen Terbaik Kernel RBF
#Dataset 2 : Untuk 80 Fitur teratas Kernel RBF Menggunakan SMOTE
pb <- txtProgressBar(min = 0, max = 1000, style = 3)
accuracy.RBF.iterasi <- NULL
for (j in 1:1000) {
  acc_cv<-NULL
  folds.smote<-createFolds(smote_train$y, 10, list=FALSE)
  for(i in 1:10){
    index=which(folds.smote==i)
    training_folds = smote_train[-index,]
    test_folds = smote_train[index,]
    svmmod_80_RBF_with_SMOTE<-
svm(y~ENST00000371095_at+ENST00000371085_at+ENST00000371082_at+ENST000003
49036_at+ENST00000265620_at+

ENST00000263080_at+ENST00000313949_at+ENST00000338783_at+ENST00000369420_
at+ENST00000369417_at+

ENST00000306082_at+ENST00000381463_at+ENST00000361035_at+ENST00000371102_
at+ENST00000371101_at+

ENST00000274311_at+ENST00000325945_at+ENST00000371100_at+ENST00000371075_
at+ENST00000354359_at+

ENST00000306120_at+ENST00000306090_at+ENST00000375594_at+ENST00000360868_
at+ENST00000358855_at+

)381936_at+ENST00000359894_at+ENST00000300947_at+ENST00000390597_
)000380991_at+

)380986_at+ENST00000265028_at+ENST00000359623_at+ENST00000323078_
)000283646_at+
```

```
ENST00000180173_at+ENST00000244296_at+ENST00000221293_at+ENST00000381251_
at+ENST00000304101_at+

ENST00000371761_at+ENST00000217246_at+ENST00000262662_at+ENST00000356692_
at+ENST00000361625_at+

ENST00000253023_at+ENST00000367843_at+ENST00000367840_at+ENST00000367839_
at+ENST00000324705_at+

ENST00000312263_at+ENST00000298694_at+ENST00000298693_at+ENST00000354181_
at+ENST00000361379_at+

ENST00000358712_at+ENST00000336092_at+ENST00000382438_at+ENST00000226972_
at+ENST00000312524_at+

ENST00000372264_at+ENST00000362031_at+ENST00000355110_at+ENST00000332302_
at+ENST00000265395_at+

ENST00000376419_at+ENST00000376417_at+ENST00000277124_at+ENST00000381753_
at+ENST00000313860_at+

ENST00000383367_at+ENST00000318809_at+ENST00000308370_at+ENST00000383179_
at+ENST00000243562_at+

ENST00000252576_at+ENST00000381917_at+ENST00000375493_at+ENST00000204005_
at+ENST00000301788_at,
                                    data=training_folds,  kernel="radial",
cost=32, gamma=0.0156250000, types="response")
    pred<-predict(svmmod_80_RBF_with_SMOTE,    newdata    =    testData,
type="response")
    y_act<-testData$y
    #Menghitung Akurasi
    acc_cv[i] = mean(pred==y_act)
  }
  accuracy.RBF = mean(as.numeric(acc_cv))
  accuracy.RBF.iterasi[j]=mean(as.numeric(accuracy.RBF))
  accuracy.RBF.result=mean(as.numeric(accuracy.RBF.iterasi))
  setTxtProgressBar(pb,j)
}
accuracy.RBF.result

#Dataset 2 : Untuk 90 Fitur teratas Kernel RBF Menggunakan SMOTE
pb <- txtProgressBar(min = 0, max = 1000, style = 3)
accuracy.RBF.iterasi <- NULL
for (j in 1:1000) {
  acc_cv<-NULL
  folds.smote<-createFolds(smote_train$y, 10, list=FALSE)
  for(i in 1:10){
    index=which(folds.smote==i)
    training_folds = smote_train[-index,]
    test_folds = smote_train[index,]
    svmmod_90_RBF_with_SMOTE<-
svm(y~ENST00000371095_at+ENST00000371085_at+ENST00000371082_at+ENST000003
49036_at+ENST00000265620_at+

ENST00000263080_at+ENST00000313949_at+ENST00000338783_at+ENST00000369420_
at+ENST00000369417_at+

    )306082_at+ENST00000381463_at+ENST00000361035_at+ENST00000371102_
    )000371101_at+

    )274311_at+ENST00000325945_at+ENST00000371100_at+ENST00000371075_
    )000354359_at+
```

```
ENST00000306120_at+ENST00000306090_at+ENST00000375594_at+ENST00000360868_
at+ENST00000358855_at+

ENST00000381936_at+ENST00000359894_at+ENST00000300947_at+ENST00000390597_
at+ENST00000380991_at+

ENST00000380986_at+ENST00000265028_at+ENST00000359623_at+ENST00000323078_
at+ENST00000283646_at+

ENST00000180173_at+ENST00000244296_at+ENST00000221293_at+ENST00000381251_
at+ENST00000304101_at+

ENST00000371761_at+ENST00000217246_at+ENST00000262662_at+ENST00000356692_
at+ENST00000361625_at+

ENST00000253023_at+ENST00000367843_at+ENST00000367840_at+ENST00000367839_
at+ENST00000324705_at+

ENST00000312263_at+ENST00000298694_at+ENST00000298693_at+ENST00000354181_
at+ENST00000361379_at+

ENST00000358712_at+ENST00000336092_at+ENST00000382438_at+ENST00000226972_
at+ENST00000312524_at+

ENST00000372264_at+ENST00000362031_at+ENST00000355110_at+ENST00000332302_
at+ENST00000265395_at+

ENST00000376419_at+ENST00000376417_at+ENST00000277124_at+ENST00000381753_
at+ENST00000313860_at+

ENST00000383367_at+ENST00000318809_at+ENST00000308370_at+ENST00000383179_
at+ENST00000243562_at+

ENST00000252576_at+ENST00000381917_at+ENST00000375493_at+ENST00000204005_
at+ENST00000301788_at+

ENST00000361272_at+ENST00000361231_at+ENST00000298047_at+ENST00000342745_
at+ENST00000244926_at+

ENST00000356505_at+ENST00000323760_at+ENST00000219168_at+ENST00000371534_
at+ENST00000383362_at,
                                    data=training_folds,  kernel="radial",
cost=32, gamma=0.0156250000, types="response")
    pred<-predict(svmmod_90_RBF_with_SMOTE,    newdata    =    testData,
type="response")
    y_act<-testData$y
    #Menghitung Akurasi
    acc_cv[i] = mean(pred==y_act)
  }
  accuracy.RBF = mean(as.numeric(acc_cv))
  accuracy.RBF.iterasi[j]=mean(as.numeric(accuracy.RBF))
  accuracy.RBF.result=mean(as.numeric(accuracy.RBF.iterasi))
  setTxtProgressBar(pb,j)
}
accuracy.RBF.result
```

```
 2 : Untuk 100 Fitur teratas Kernel RBF Menggunakan SMOTE
tProgressBar(min = 0, max = 1000, style = 3)
.RBF.iterasi <- NULL
n 1:1000) {
<-NULL
smote<-createFolds(smote_train$y, 10, list=FALSE)
```

```
  for(i in 1:10){
    index=which(folds.smote==i)
    training_folds = smote_train[-index,]
    test_folds = smote_train[index,]
    svmmod_100_RBF_with_SMOTE<-
svm(y~ENST00000371095_at+ENST00000371085_at+ENST00000371082_at+ENST000003
49036_at+ENST00000265620_at+

ENST00000263080_at+ENST00000313949_at+ENST00000338783_at+ENST00000369420_
at+ENST00000369417_at+

ENST00000306082_at+ENST00000381463_at+ENST00000361035_at+ENST00000371102_
at+ENST00000371101_at+

ENST00000274311_at+ENST00000325945_at+ENST00000371100_at+ENST00000371075_
at+ENST00000354359_at+

ENST00000306120_at+ENST00000306090_at+ENST00000375594_at+ENST00000360868_
at+ENST00000358855_at+

ENST00000381936_at+ENST00000359894_at+ENST00000300947_at+ENST00000390597_
at+ENST00000380991_at+

ENST00000380986_at+ENST00000265028_at+ENST00000359623_at+ENST00000323078_
at+ENST00000283646_at+

ENST00000180173_at+ENST00000244296_at+ENST00000221293_at+ENST00000381251_
at+ENST00000304101_at+

ENST00000371761_at+ENST00000217246_at+ENST00000262662_at+ENST00000356692_
at+ENST00000361625_at+

ENST00000253023_at+ENST00000367843_at+ENST00000367840_at+ENST00000367839_
at+ENST00000324705_at+

ENST00000312263_at+ENST00000298694_at+ENST00000298693_at+ENST00000354181_
at+ENST00000361379_at+

ENST00000358712_at+ENST00000336092_at+ENST00000382438_at+ENST00000226972_
at+ENST00000312524_at+

ENST00000372264_at+ENST00000362031_at+ENST00000355110_at+ENST00000332302_
at+ENST00000265395_at+

ENST00000376419_at+ENST00000376417_at+ENST00000277124_at+ENST00000381753_
at+ENST00000313860_at+

ENST00000383367_at+ENST00000318809_at+ENST00000308370_at+ENST00000383179_
at+ENST00000243562_at+

ENST00000252576_at+ENST00000381917_at+ENST00000375493_at+ENST00000204005_
at+ENST00000301788_at+

ENST00000361272_at+ENST00000361231_at+ENST00000298047_at+ENST00000342745_
at+ENST00000244926_at+

ENST00000356505_at+ENST00000323760_at+ENST00000219168_at+ENST00000371534_
at+ENST00000383362_at+

                    0383360_at+ENST00000383359_at+ENST00000370109_at+ENST00000383178_
                    0000383177_at+

                    0370107_at+ENST00000371646_at+ENST00000223864_at+ENST00000371554_
                    0000342173_at,
```

```
                        data=training_folds,   kernel="radial",   cost=32,
gamma=0.0156250000, types="response")
    pred<-predict(svmmod_100_RBF_with_SMOTE,    newdata    =    testData,
type="response")
    y_act<-testData$y
    #Menghitung Akurasi
    acc_cv[i] = mean(pred==y_act)
  }
  accuracy.RBF = mean(as.numeric(acc_cv))
  accuracy.RBF.iterasi[j]=mean(as.numeric(accuracy.RBF))
  accuracy.RBF.result=mean(as.numeric(accuracy.RBF.iterasi))
  setTxtProgressBar(pb,j)
}
accuracy.RBF.result

#Dataset 2 : Untuk 70 Fitur teratas Kernel RBF Tanpa Menggunakan SMOTE
pb <- txtProgressBar(min = 0, max = 1000, style = 3)
accuracy.RBF.iterasi <- NULL
for (j in 1:1000) {
  acc_cv<-NULL
  folds.train<-createFolds(trainData$y, 10, list=FALSE)
  for(i in 1:10){
    index=which(folds.train==i)
    training_folds = trainData[-index,]
    test_folds = trainData[index,]
    svmmod_70_RBF_no_SMOTE<-
svm(y~ENST00000371095_at+ENST00000371085_at+ENST00000371082_at+ENST000003
49036_at+ENST00000265620_at+

ENST00000263080_at+ENST00000313949_at+ENST00000338783_at+ENST00000369420_
at+ENST00000369417_at+

ENST00000306082_at+ENST00000381463_at+ENST00000361035_at+ENST00000371102_
at+ENST00000371101_at+

ENST00000274311_at+ENST00000325945_at+ENST00000371100_at+ENST00000371075_
at+ENST00000354359_at+

ENST00000306120_at+ENST00000306090_at+ENST00000375594_at+ENST00000360868_
at+ENST00000358855_at+

ENST00000381936_at+ENST00000359894_at+ENST00000300947_at+ENST00000390597_
at+ENST00000380991_at+

ENST00000380986_at+ENST00000265028_at+ENST00000359623_at+ENST00000323078_
at+ENST00000283646_at+

ENST00000180173_at+ENST00000244296_at+ENST00000221293_at+ENST00000381251_
at+ENST00000304101_at+

ENST00000371761_at+ENST00000217246_at+ENST00000262662_at+ENST00000356692_
at+ENST00000361625_at+

ENST00000253023_at+ENST00000367843_at+ENST00000367840_at+ENST00000367839_
at+ENST00000324705_at+

ENST00000312263_at+ENST00000298694_at+ENST00000298693_at+ENST00000354181_
at+ENST00000361379_at+

    )358712_at+ENST00000336092_at+ENST00000382438_at+ENST00000226972_
    )000312524_at+

    )372264_at+ENST00000362031_at+ENST00000355110_at+ENST00000332302_
    )000265395_at+
```

```
ENST00000376419_at+ENST00000376417_at+ENST00000277124_at+ENST00000381753_
at+ENST00000313860_at,
                      data=training_folds,   kernel="radial",   cost=32,
gamma=0.0156250000, types="response")
    pred<-predict(svmmod_70_RBF_no_SMOTE,     newdata     =     testData,
type="response")
    y_act<-testData$y
    #Menghitung Akurasi
    acc_cv[i] = mean(pred==y_act)
  }
  accuracy.RBF = mean(as.numeric(acc_cv))
  accuracy.RBF.iterasi[j]=mean(as.numeric(accuracy.RBF))
  accuracy.RBF.result=mean(as.numeric(accuracy.RBF.iterasi))
  setTxtProgressBar(pb,j)
}
accuracy.RBF.result

#Dataset 2 : Untuk 90 Fitur teratas Kernel RBF Tanpa Menggunakan SMOTE
pb <- txtProgressBar(min = 0, max = 1000, style = 3)
accuracy.RBF.iterasi <- NULL
for (j in 1:1000) {
  acc_cv<-NULL
  folds.train<-createFolds(trainData$y, 10, list=FALSE)
  for(i in 1:10){
    index=which(folds.train==i)
    training_folds = trainData[-index,]
    test_folds = trainData[index,]
    svmmod_90_RBF_no_SMOTE<-
svm(y~ENST00000371095_at+ENST00000371085_at+ENST00000371082_at+ENST000003
49036_at+ENST00000265620_at+

ENST00000263080_at+ENST00000313949_at+ENST00000338783_at+ENST00000369420_
at+ENST00000369417_at+

ENST00000306082_at+ENST00000381463_at+ENST00000361035_at+ENST00000371102_
at+ENST00000371101_at+

ENST00000274311_at+ENST00000325945_at+ENST00000371100_at+ENST00000371075_
at+ENST00000354359_at+

ENST00000306120_at+ENST00000306090_at+ENST00000375594_at+ENST00000360868_
at+ENST00000358855_at+

ENST00000381936_at+ENST00000359894_at+ENST00000300947_at+ENST00000390597_
at+ENST00000380991_at+

ENST00000380986_at+ENST00000265028_at+ENST00000359623_at+ENST00000323078_
at+ENST00000283646_at+

ENST00000180173_at+ENST00000244296_at+ENST00000221293_at+ENST00000381251_
at+ENST00000304101_at+

ENST00000371761_at+ENST00000217246_at+ENST00000262662_at+ENST00000356692_
at+ENST00000361625_at+

ENST00000253023_at+ENST00000367843_at+ENST00000367840_at+ENST00000367839_
at+ENST00000324705_at+

    0312263_at+ENST00000298694_at+ENST00000298693_at+ENST00000354181_
    0000361379_at+

    0358712_at+ENST00000336092_at+ENST00000382438_at+ENST00000226972_
    0000312524_at+
```

```
ENST00000372264_at+ENST00000362031_at+ENST00000355110_at+ENST00000332302_
at+ENST00000265395_at+

ENST00000376419_at+ENST00000376417_at+ENST00000277124_at+ENST00000381753_
at+ENST00000313860_at+

ENST00000383367_at+ENST00000318809_at+ENST00000308370_at+ENST00000383179_
at+ENST00000243562_at+

ENST00000252576_at+ENST00000381917_at+ENST00000375493_at+ENST00000204005_
at+ENST00000301788_at+

ENST00000361272_at+ENST00000361231_at+ENST00000298047_at+ENST00000342745_
at+ENST00000244926_at+

ENST00000356505_at+ENST00000323760_at+ENST00000219168_at+ENST00000371534_
at+ENST00000383362_at,
                                  data=training_folds,     kernel="radial",
cost=32, gamma=0.0156250000, types="response")
    pred<-predict(svmmod_90_RBF_no_SMOTE,     newdata     =     testData,
type="response")
    y_act<-testData$y
    #Menghitung Akurasi
    acc_cv[i] = mean(pred==y_act)
  }
  accuracy.RBF = mean(as.numeric(acc_cv))
  accuracy.RBF.iterasi[j]=mean(as.numeric(accuracy.RBF))
  accuracy.RBF.result=mean(as.numeric(accuracy.RBF.iterasi))
  setTxtProgressBar(pb,j)
}
accuracy.RBF.result

#Dataset 2 : Untuk 100 Fitur teratas Kernel RBF Tanpa Menggunakan SMOTE
pb <- txtProgressBar(min = 0, max = 1000, style = 3)
accuracy.RBF.iterasi <- NULL
for (j in 1:1000) {
  acc_cv<-NULL
  folds.train<-createFolds(trainData$y, 10, list=FALSE)
  for(i in 1:10){
    index=which(folds.train==i)
    training_folds = trainData[-index,]
    test_folds = trainData[index,]
    svmmod_100_RBF_no_SMOTE<-
svm(y~ENST00000371095_at+ENST00000371085_at+ENST00000371082_at+ENST000003
49036_at+ENST00000265620_at+

ENST00000263080_at+ENST00000313949_at+ENST00000338783_at+ENST00000369420_
at+ENST00000369417_at+

ENST00000306082_at+ENST00000381463_at+ENST00000361035_at+ENST00000371102_
at+ENST00000371101_at+

ENST00000274311_at+ENST00000325945_at+ENST00000371100_at+ENST00000371075_
at+ENST00000354359_at+

ENST00000306120_at+ENST00000306090_at+ENST00000375594_at+ENST00000360868_
at+ENST00000358855_at+

    )381936_at+ENST00000359894_at+ENST00000300947_at+ENST00000390597_
    )000380991_at+

    )380986_at+ENST00000265028_at+ENST00000359623_at+ENST00000323078_
    )000283646_at+
```

```
ENST00000180173_at+ENST00000244296_at+ENST00000221293_at+ENST00000381251_
at+ENST00000304101_at+

ENST00000371761_at+ENST00000217246_at+ENST00000262662_at+ENST00000356692_
at+ENST00000361625_at+

ENST00000253023_at+ENST00000367843_at+ENST00000367840_at+ENST00000367839_
at+ENST00000324705_at+

ENST00000312263_at+ENST00000298694_at+ENST00000298693_at+ENST00000354181_
at+ENST00000361379_at+

ENST00000358712_at+ENST00000336092_at+ENST00000382438_at+ENST00000226972_
at+ENST00000312524_at+

ENST00000372264_at+ENST00000362031_at+ENST00000355110_at+ENST00000332302_
at+ENST00000265395_at+

ENST00000376419_at+ENST00000376417_at+ENST00000277124_at+ENST00000381753_
at+ENST00000313860_at+

ENST00000383367_at+ENST00000318809_at+ENST00000308370_at+ENST00000383179_
at+ENST00000243562_at+

ENST00000252576_at+ENST00000381917_at+ENST00000375493_at+ENST00000204005_
at+ENST00000301788_at+

ENST00000361272_at+ENST00000361231_at+ENST00000298047_at+ENST00000342745_
at+ENST00000244926_at+

ENST00000356505_at+ENST00000323760_at+ENST00000219168_at+ENST00000371534_
at+ENST00000383362_at+

ENST00000383360_at+ENST00000383359_at+ENST00000370109_at+ENST00000383178_
at+ENST00000383177_at+

ENST00000370107_at+ENST00000371646_at+ENST00000223864_at+ENST00000371554_
at+ENST00000342173_at,
                                  data=training_folds,     kernel="radial",
cost=32, gamma=0.0156250000, types="response")
    pred<-predict(svmmod_100_RBF_no_SMOTE,     newdata     =     testData,
type="response")
    y_act<-testData$y
    #Menghitung Akurasi
    acc_cv[i] = mean(pred==y_act)
  }
  accuracy.RBF = mean(as.numeric(acc_cv))
  accuracy.RBF.iterasi[j]=mean(as.numeric(accuracy.RBF))
  accuracy.RBF.result=mean(as.numeric(accuracy.RBF.iterasi))
  setTxtProgressBar(pb,j)
}
accuracy.RBF.result


#Dataset 2 : Untuk 90  Fitur Teratas Kernel Polynomial Menggunakan SMOTE
pb <- txtProgressBar(min = 0, max = 1000, style = 3)
accuracy.polynomial.iterasi <- NULL
      n 1:1000) {
      <-NULL
      smote<-createFolds(smote_train$y, 10, list=FALSE)
      in 1:10){
      x=which(folds.smote==i)
      ning_folds = smote_train[-index,]
```

```
    test_folds = smote_train[index,]
    svmmod_90_Poly2_with_SMOTE<-
svm(y~ENST00000371095_at+ENST00000371085_at+ENST00000371082_at+ENST000003
49036_at+ENST00000265620_at+

ENST00000263080_at+ENST00000313949_at+ENST00000338783_at+ENST00000369420_
at+ENST00000369417_at+

ENST00000306082_at+ENST00000381463_at+ENST00000361035_at+ENST00000371102_
at+ENST00000371101_at+

ENST00000274311_at+ENST00000325945_at+ENST00000371100_at+ENST00000371075_
at+ENST00000354359_at+

ENST00000306120_at+ENST00000306090_at+ENST00000375594_at+ENST00000360868_
at+ENST00000358855_at+

ENST00000381936_at+ENST00000359894_at+ENST00000300947_at+ENST00000390597_
at+ENST00000380991_at+

ENST00000380986_at+ENST00000265028_at+ENST00000359623_at+ENST00000323078_
at+ENST00000283646_at+

ENST00000180173_at+ENST00000244296_at+ENST00000221293_at+ENST00000381251_
at+ENST00000304101_at+

ENST00000371761_at+ENST00000217246_at+ENST00000262662_at+ENST00000356692_
at+ENST00000361625_at+

ENST00000253023_at+ENST00000367843_at+ENST00000367840_at+ENST00000367839_
at+ENST00000324705_at+

ENST00000312263_at+ENST00000298694_at+ENST00000298693_at+ENST00000354181_
at+ENST00000361379_at+

ENST00000358712_at+ENST00000336092_at+ENST00000382438_at+ENST00000226972_
at+ENST00000312524_at+

ENST00000372264_at+ENST00000362031_at+ENST00000355110_at+ENST00000332302_
at+ENST00000265395_at+

ENST00000376419_at+ENST00000376417_at+ENST00000277124_at+ENST00000381753_
at+ENST00000313860_at+

ENST00000383367_at+ENST00000318809_at+ENST00000308370_at+ENST00000383179_
at+ENST00000243562_at+

ENST00000252576_at+ENST00000381917_at+ENST00000375493_at+ENST00000204005_
at+ENST00000301788_at+

ENST00000361272_at+ENST00000361231_at+ENST00000298047_at+ENST00000342745_
at+ENST00000244926_at+

ENST00000356505_at+ENST00000323760_at+ENST00000219168_at+ENST00000371534_
at+ENST00000383362_at,
                                data=training_folds,
kernel="polynomial",  cost=16,  degree=2,  coef0=0.125,  gamma=0.015625,
types="response")
    <-predict(svmmod_90_Poly2_with_SMOTE,   newdata   =   testData,
    sponse")
    t<-testData$y
    ghitung Akurasi
    cv[i] = mean(pred==y_act)
```

```
  accuracy.polynomial = mean(as.numeric(acc_cv))
  accuracy.polynomial.iterasi[j]=mean(as.numeric(accuracy.polynomial))

accuracy.polynomial.result=mean(as.numeric(accuracy.polynomial.iterasi))
  setTxtProgressBar(pb,j)
}
accuracy.polynomial.result

#Dataset 2 : Untuk 100  Fitur Teratas Kernel Polynomial Menggunakan SMOTE
pb <- txtProgressBar(min = 0, max = 1000, style = 3)
accuracy.polynomial.iterasi <- NULL
for (j in 1:1000) {
  acc_cv<-NULL
  folds.smote<-createFolds(smote_train$y, 10, list=FALSE)
  for(i in 1:10){
    index=which(folds.smote==i)
    training_folds = smote_train[-index,]
    test_folds = smote_train[index,]
    svmmod_100_Poly2_with_SMOTE<-
svm(y~ENST00000371095_at+ENST00000371085_at+ENST00000371082_at+ENST000003
49036_at+ENST00000265620_at+

ENST00000263080_at+ENST00000313949_at+ENST00000338783_at+ENST00000369420_
at+ENST00000369417_at+

ENST00000306082_at+ENST00000381463_at+ENST00000361035_at+ENST00000371102_
at+ENST00000371101_at+

ENST00000274311_at+ENST00000325945_at+ENST00000371100_at+ENST00000371075_
at+ENST00000354359_at+

ENST00000306120_at+ENST00000306090_at+ENST00000375594_at+ENST00000360868_
at+ENST00000358855_at+

ENST00000381936_at+ENST00000359894_at+ENST00000300947_at+ENST00000390597_
at+ENST00000380991_at+

ENST00000380986_at+ENST00000265028_at+ENST00000359623_at+ENST00000323078_
at+ENST00000283646_at+

ENST00000180173_at+ENST00000244296_at+ENST00000221293_at+ENST00000381251_
at+ENST00000304101_at+

ENST00000371761_at+ENST00000217246_at+ENST00000262662_at+ENST00000356692_
at+ENST00000361625_at+

ENST00000253023_at+ENST00000367843_at+ENST00000367840_at+ENST00000367839_
at+ENST00000324705_at+

ENST00000312263_at+ENST00000298694_at+ENST00000298693_at+ENST00000354181_
at+ENST00000361379_at+

ENST00000358712_at+ENST00000336092_at+ENST00000382438_at+ENST00000226972_
at+ENST00000312524_at+

ENST00000372264_at+ENST00000362031_at+ENST00000355110_at+ENST00000332302_
at+ENST00000265395_at+

)376419_at+ENST00000376417_at+ENST00000277124_at+ENST00000381753_
)000313860_at+

)383367_at+ENST00000318809_at+ENST00000308370_at+ENST00000383179_
)000243562_at+
```

```
ENST00000252576_at+ENST00000381917_at+ENST00000375493_at+ENST00000204005_
at+ENST00000301788_at+

ENST00000361272_at+ENST00000361231_at+ENST00000298047_at+ENST00000342745_
at+ENST00000244926_at+

ENST00000356505_at+ENST00000323760_at+ENST00000219168_at+ENST00000371534_
at+ENST00000383362_at+

ENST00000383360_at+ENST00000383359_at+ENST00000370109_at+ENST00000383178_
at+ENST00000383177_at+

ENST00000370107_at+ENST00000371646_at+ENST00000223864_at+ENST00000371554_
at+ENST00000342173_at,
                                    data=training_folds,
kernel="polynomial",  cost=16,  degree=2,  coef0=0.125,  gamma=0.015625,
types="response")
    pred<-predict(svmmod_100_Poly2_with_SMOTE,   newdata   =   testData,
type="response")
    y_act<-testData$y
    #Menghitung Akurasi
    acc_cv[i] = mean(pred==y_act)
  }
  accuracy.polynomial = mean(as.numeric(acc_cv))
  accuracy.polynomial.iterasi[j]=mean(as.numeric(accuracy.polynomial))

accuracy.polynomial.result=mean(as.numeric(accuracy.polynomial.iterasi))
  setTxtProgressBar(pb,j)
}
accuracy.polynomial.result


#Dataset 2 : Untuk 90  Fitur Teratas Kernel Polynomial Tanpa Menggunakan
SMOTE
pb <- txtProgressBar(min = 0, max = 1000, style = 3)
accuracy.polynomial.iterasi <- NULL
for (j in 1:1000) {
  acc_cv<-NULL
  folds.train<-createFolds(trainData$y, 10, list=FALSE)
  for(i in 1:10){
    index=which(folds.train==i)
    training_folds = trainData[-index,]
    test_folds = trainData[index,]
    svmmod_90_Poly2_no_SMOTE<-
svm(y~ENST00000371095_at+ENST00000371085_at+ENST00000371082_at+ENST000003
49036_at+ENST00000265620_at+

ENST00000263080_at+ENST00000313949_at+ENST00000338783_at+ENST00000369420_
at+ENST00000369417_at+

ENST00000306082_at+ENST00000381463_at+ENST00000361035_at+ENST00000371102_
at+ENST00000371101_at+

ENST00000274311_at+ENST00000325945_at+ENST00000371100_at+ENST00000371075_
at+ENST00000354359_at+

ENST00000306120_at+ENST00000306090_at+ENST00000375594_at+ENST00000360868_
      )000358855_at+

      )381936_at+ENST00000359894_at+ENST00000300947_at+ENST00000390597_
      )000380991_at+
```

```
ENST00000380986_at+ENST00000265028_at+ENST00000359623_at+ENST00000323078_
at+ENST00000283646_at+

ENST00000180173_at+ENST00000244296_at+ENST00000221293_at+ENST00000381251_
at+ENST00000304101_at+

ENST00000371761_at+ENST00000217246_at+ENST00000262662_at+ENST00000356692_
at+ENST00000361625_at+

ENST00000253023_at+ENST00000367843_at+ENST00000367840_at+ENST00000367839_
at+ENST00000324705_at+

ENST00000312263_at+ENST00000298694_at+ENST00000298693_at+ENST00000354181_
at+ENST00000361379_at+

ENST00000358712_at+ENST00000336092_at+ENST00000382438_at+ENST00000226972_
at+ENST00000312524_at+

ENST00000372264_at+ENST00000362031_at+ENST00000355110_at+ENST00000332302_
at+ENST00000265395_at+

ENST00000376419_at+ENST00000376417_at+ENST00000277124_at+ENST00000381753_
at+ENST00000313860_at+

ENST00000383367_at+ENST00000318809_at+ENST00000308370_at+ENST00000383179_
at+ENST00000243562_at+

ENST00000252576_at+ENST00000381917_at+ENST00000375493_at+ENST00000204005_
at+ENST00000301788_at+

ENST00000361272_at+ENST00000361231_at+ENST00000298047_at+ENST00000342745_
at+ENST00000244926_at+

ENST00000356505_at+ENST00000323760_at+ENST00000219168_at+ENST00000371534_
at+ENST00000383362_at,
                          data=training_folds,        kernel="polynomial",
cost=16, degree=2, coef0=0.125, gamma=0.015625, types="response")
    pred<-predict(svmmod_90_Poly2_no_SMOTE,    newdata    =    testData,
type="response")
    y_act<-testData$y
    #Menghitung Akurasi
    acc_cv[i] = mean(pred==y_act)
  }
  accuracy.polynomial = mean(as.numeric(acc_cv))
  accuracy.polynomial.iterasi[j]=mean(as.numeric(accuracy.polynomial))

accuracy.polynomial.result=mean(as.numeric(accuracy.polynomial.iterasi))
  setTxtProgressBar(pb,j)
}
accuracy.polynomial.result
```

Lampiran 16 : Hasil Parameter Tuning dan Generalization Error Untuk Dataset I SVM Kernel Linear

```
[[1]]
[[1]]$svm.list
      n.list[[1]]
           error dispersion
       0.2099892         NA

      n.list[[2]]
          error dispersion
```

```
1 0.125 0.3225147         NA

[[1]]$svm.list[[3]]
   cost     error dispersion
1 0.125 0.2803406         NA

[[1]]$svm.list[[4]]
    cost    error dispersion
1 0.0625 0.315261         NA

[[1]]$svm.list[[5]]
   cost     error dispersion
1 0.125 0.2398028         NA

[[1]]$svm.list[[6]]
   cost     error dispersion
1 0.125 0.3145649         NA

[[1]]$svm.list[[7]]
   cost     error dispersion
1 0.125 0.1873276         NA

[[1]]$svm.list[[8]]
   cost     error dispersion
1 0.125 0.2769361         NA

[[1]]$svm.list[[9]]
   cost     error dispersion
1 0.125 0.3067519         NA

[[1]]$svm.list[[10]]
   cost     error dispersion
1 0.125 0.3063824         NA


[[1]]$error
[1] 0.2759871


[[2]]
[[2]]$svm.list
[[2]]$svm.list[[1]]
  cost      error dispersion
1    4 0.09875325         NA

[[2]]$svm.list[[2]]
  cost     error dispersion
1    1 0.1755463         NA

[[2]]$svm.list[[3]]
  cost     error dispersion
1    2 0.1132858         NA

[[2]]$svm.list[[4]]
  cost     error dispersion
1    1 0.2915211         NA

[[2]]$svm.list[[5]]
          error dispersion
     .3125511         NA

     m.list[[6]]
          error dispersion
     .1870789         NA
```

```
[[2]]$svm.list[[7]]
  cost      error dispersion
1  0.5 0.06455164         NA

[[2]]$svm.list[[8]]
   cost     error dispersion
1 0.125 0.2006421         NA

[[2]]$svm.list[[9]]
  cost     error dispersion
1 0.25 0.1793379         NA

[[2]]$svm.list[[10]]
  cost     error dispersion
1    1 0.3243813         NA


[[2]]$error
[1] 0.1947649


[[3]]
[[3]]$svm.list
[[3]]$svm.list[[1]]
  cost     error dispersion
1    4 0.1353709         NA

[[3]]$svm.list[[2]]
  cost     error dispersion
1    4 0.1441027         NA

[[3]]$svm.list[[3]]
  cost     error dispersion
1    4 0.1565363         NA

[[3]]$svm.list[[4]]
  cost     error dispersion
1 0.25 0.1932819         NA

[[3]]$svm.list[[5]]
  cost     error dispersion
1    2 0.2570434         NA

[[3]]$svm.list[[6]]
  cost     error dispersion
1 0.25 0.1925989         NA

[[3]]$svm.list[[7]]
  cost     error dispersion
1    4 0.1051849         NA

[[3]]$svm.list[[8]]
  cost     error dispersion
1    8 0.3104916         NA

[[3]]$svm.list[[9]]
  cost      error dispersion
      .07783818         NA

      n.list[[10]]
         error dispersion
      .2283864         NA
```

```
[[3]]$error
[1] 0.1800835


[[4]]
[[4]]$svm.list
[[4]]$svm.list[[1]]
  cost     error dispersion
1  0.5 0.1070418         NA

[[4]]$svm.list[[2]]
  cost    error dispersion
1    8 0.130896         NA

[[4]]$svm.list[[3]]
  cost      error dispersion
1    8 0.08126247         NA

[[4]]$svm.list[[4]]
  cost     error dispersion
1    4 0.1594306         NA

[[4]]$svm.list[[5]]
  cost     error dispersion
1    4 0.1330307         NA

[[4]]$svm.list[[6]]
  cost    error dispersion
1    8 0.170034         NA

[[4]]$svm.list[[7]]
  cost     error dispersion
1    8 0.1396072         NA

[[4]]$svm.list[[8]]
  cost     error dispersion
1    8 0.2677336         NA

[[4]]$svm.list[[9]]
  cost      error dispersion
1    8 0.03104802         NA

[[4]]$svm.list[[10]]
  cost      error dispersion
1    8 0.02561703         NA


[[4]]$error
[1] 0.1245701


[[5]]
[[5]]$svm.list
[[5]]$svm.list[[1]]
  cost     error dispersion
1    4 0.1324318         NA

m.list[[2]]
   error dispersion
.1495808         NA

m.list[[3]]
   error dispersion
```

```
1  0.5 0.1046234         NA

[[5]]$svm.list[[4]]
  cost     error dispersion
1    8 0.1979669         NA

[[5]]$svm.list[[5]]
  cost     error dispersion
1    1 0.1811061         NA

[[5]]$svm.list[[6]]
  cost     error dispersion
1    4 0.1808585         NA

[[5]]$svm.list[[7]]
  cost     error dispersion
1    8 0.1440457         NA

[[5]]$svm.list[[8]]
  cost     error dispersion
1    8 0.2697571         NA

[[5]]$svm.list[[9]]
  cost      error dispersion
1  0.5 0.02985781         NA

[[5]]$svm.list[[10]]
  cost     error dispersion
1  0.5 0.1134575         NA


[[5]]$error
[1] 0.1503686
```

Lampiran 17 : Hasil Parameter Tuning dan Generalization Error Untuk Dataset II SVM Kernel Linear

```
[[1]]
[[1]]$svm.list
[[1]]$svm.list[[1]]
   cost   error dispersion
1 0.125 0.18165         NA

[[1]]$svm.list[[2]]
    cost    error dispersion
1 0.0625 0.257277         NA

[[1]]$svm.list[[3]]
    cost    error dispersion
1 0.0625 0.312586         NA

[[1]]$svm.list[[4]]
    cost     error dispersion
1 0.0625 0.1786328         NA

[[1]]$svm.list[[5]]
     error dispersion
 0.3228888         NA

n.list[[6]]
    error dispersion
 0.326725         NA
```

```
[[1]]$svm.list[[7]]
  cost      error dispersion
1   32 0.07420165         NA

[[1]]$svm.list[[8]]
    cost     error dispersion
1 0.0625 0.3335564         NA

[[1]]$svm.list[[9]]
    cost     error dispersion
1 0.0625 0.2299429         NA

[[1]]$svm.list[[10]]
    cost     error dispersion
1 0.0625 0.2010833         NA


[[1]]$error
[1] 0.2418544


[[2]]
[[2]]$svm.list
[[2]]$svm.list[[1]]
   cost     error dispersion
1 0.125 0.1729112         NA

[[2]]$svm.list[[2]]
   cost     error dispersion
1 0.125 0.2380576         NA

[[2]]$svm.list[[3]]
  cost     error dispersion
1    8 0.2565458         NA

[[2]]$svm.list[[4]]
  cost     error dispersion
1  0.5 0.1947214         NA

[[2]]$svm.list[[5]]
  cost     error dispersion
1    2 0.3139214         NA

[[2]]$svm.list[[6]]
  cost     error dispersion
1 0.25 0.2779624         NA

[[2]]$svm.list[[7]]
  cost      error dispersion
1    1 0.06283784         NA

[[2]]$svm.list[[8]]
  cost     error dispersion
1    1 0.3587519         NA

[[2]]$svm.list[[9]]
  cost     error dispersion
1 0.25 0.2332351         NA

      n.list[[10]]
         error dispersion
      .1423181         NA
```

```
[[2]]$error
[1] 0.2251263


[[3]]
[[3]]$svm.list
[[3]]$svm.list[[1]]
  cost    error dispersion
1 0.25 0.238328         NA

[[3]]$svm.list[[2]]
  cost    error dispersion
1  0.5 0.164552         NA

[[3]]$svm.list[[3]]
  cost     error dispersion
1  0.5 0.1120403         NA

[[3]]$svm.list[[4]]
  cost     error dispersion
1  0.5 0.2381537         NA

[[3]]$svm.list[[5]]
  cost     error dispersion
1    1 0.1659295         NA

[[3]]$svm.list[[6]]
  cost   error dispersion
1  0.5 0.13964         NA

[[3]]$svm.list[[7]]
  cost      error dispersion
1  0.5 0.09394178         NA

[[3]]$svm.list[[8]]
  cost    error dispersion
1    1 0.260023         NA

[[3]]$svm.list[[9]]
  cost     error dispersion
1  0.5 0.1885458         NA

[[3]]$svm.list[[10]]
  cost     error dispersion
1    1 0.0879608         NA


[[3]]$error
[1] 0.1689115


[[4]]
[[4]]$svm.list
[[4]]$svm.list[[1]]
  cost     error dispersion
1    1 0.1371336         NA

[[4]]$svm.list[[2]]
          error dispersion
     .1605082         NA

     n.list[[3]]
          error dispersion
     .1370987         NA
```

```
[[4]]$svm.list[[4]]
  cost      error dispersion
1  0.5 0.06753403         NA

[[4]]$svm.list[[5]]
  cost     error dispersion
1  0.5 0.1187041         NA

[[4]]$svm.list[[6]]
  cost     error dispersion
1    2 0.1414636         NA

[[4]]$svm.list[[7]]
  cost      error dispersion
1  0.5 0.05796294         NA

[[4]]$svm.list[[8]]
  cost     error dispersion
1    1 0.2095888         NA

[[4]]$svm.list[[9]]
  cost     error dispersion
1    4 0.2616266         NA

[[4]]$svm.list[[10]]
  cost      error dispersion
1    1 0.03934567         NA


[[4]]$error
[1] 0.1329709


[[5]]
[[5]]$svm.list
[[5]]$svm.list[[1]]
  cost     error dispersion
1  0.5 0.1162117         NA

[[5]]$svm.list[[2]]
  cost     error dispersion
1    1 0.1765911         NA

[[5]]$svm.list[[3]]
  cost     error dispersion
1    1 0.1180387         NA

[[5]]$svm.list[[4]]
  cost      error dispersion
1  0.5 0.05565749         NA

[[5]]$svm.list[[5]]
  cost     error dispersion
1  0.5 0.1116731         NA

[[5]]$svm.list[[6]]
  cost     error dispersion
      .1491698         NA

      n.list[[7]]
          error dispersion
      .06657099         NA
```

```
[[5]]$svm.list[[8]]
  cost     error dispersion
1    1 0.2064228         NA

[[5]]$svm.list[[9]]
  cost     error dispersion
1    4 0.2878678         NA

[[5]]$svm.list[[10]]
  cost      error dispersion
1    1 0.03808899         NA


[[5]]$error
[1] 0.1327549
```

Lampiran 18 : Hasil *Parameter Tuning* dan *Generalization Error* Untuk Dataset I SVM Kernel RBF

```
[[1]]
[[1]]$svm.list
[[1]]$svm.list[[1]]
     gamma cost     error dispersion
1 0.015625  0.5 0.1873242         NA

[[1]]$svm.list[[2]]
         gamma cost    error dispersion
1 0.0004882812   64 0.252767         NA

[[1]]$svm.list[[3]]
         gamma cost     error dispersion
1 0.0004882812   64 0.2303701         NA

[[1]]$svm.list[[4]]
      gamma cost     error dispersion
1 0.0078125    2 0.2366401         NA

[[1]]$svm.list[[5]]
         gamma cost     error dispersion
1 0.0004882812   64 0.2099446         NA

[[1]]$svm.list[[6]]
         gamma cost     error dispersion
1 0.0002441406   64 0.2350155         NA

[[1]]$svm.list[[7]]
         gamma cost    error dispersion
1 0.0004882812   64 0.170137         NA

[[1]]$svm.list[[8]]
         gamma cost     error dispersion
1 0.0004882812   64 0.2332621         NA

[[1]]$svm.list[[9]]
         gamma cost     error dispersion
1 0.0004882812   64 0.2682507         NA

[[1]]$svm.list[[10]]
         gamma cost     error dispersion
441406   64 0.2740012         NA


ror
97713
```

```
[[2]]
[[2]]$svm.list
[[2]]$svm.list[[1]]
  gamma cost     error dispersion
1     1 0.25 0.1004293         NA

[[2]]$svm.list[[2]]
  gamma cost     error dispersion
1     1 0.25 0.1826284         NA

[[2]]$svm.list[[3]]
  gamma cost     error dispersion
1     1  0.5 0.1093497         NA

[[2]]$svm.list[[4]]
  gamma cost     error dispersion
1     1 0.25 0.3480401         NA

[[2]]$svm.list[[5]]
  gamma cost     error dispersion
1   0.5    2 0.3125511         NA

[[2]]$svm.list[[6]]
  gamma cost     error dispersion
1     1 0.25 0.1756818         NA

[[2]]$svm.list[[7]]
  gamma cost      error dispersion
1     1 0.25 0.07091594         NA

[[2]]$svm.list[[8]]
  gamma  cost     error dispersion
1     1 0.125 0.1800361         NA

[[2]]$svm.list[[9]]
  gamma  cost     error dispersion
1     1 0.125 0.1753595         NA

[[2]]$svm.list[[10]]
  gamma cost     error dispersion
1  0.25    4 0.3260223         NA


[[2]]$error
[1] 0.1981014


[[3]]
[[3]]$svm.list
[[3]]$svm.list[[1]]
  gamma cost     error dispersion
1  0.25    8 0.1373952         NA

[[3]]$svm.list[[2]]
  gamma cost     error dispersion
1     1 0.25 0.1554501         NA

          n.list[[3]]
          cost     error dispersion
             8 0.1613883         NA

          n.list[[4]]
```

```
  gamma cost     error dispersion
1   0.5 0.25 0.1914915         NA

[[3]]$svm.list[[5]]
  gamma cost     error dispersion
1     1    1 0.2064791         NA

[[3]]$svm.list[[6]]
  gamma cost     error dispersion
1     1    1 0.1947552         NA

[[3]]$svm.list[[7]]
  gamma cost     error dispersion
1 0.125   32 0.1168915         NA

[[3]]$svm.list[[8]]
  gamma cost     error dispersion
1  0.25   16 0.3177215         NA

[[3]]$svm.list[[9]]
  gamma cost      error dispersion
1     1  0.5 0.07482025         NA

[[3]]$svm.list[[10]]
  gamma cost     error dispersion
1  0.25 0.25 0.2438964         NA


[[3]]$error
[1] 0.1800289


[[4]]
[[4]]$svm.list
[[4]]$svm.list[[1]]
  gamma cost      error dispersion
1     1   16 0.06346612         NA

[[4]]$svm.list[[2]]
  gamma cost      error dispersion
1    0.25    2 0.04243656         NA

[[4]]$svm.list[[3]]
  gamma cost      error dispersion
1    0.25    4 0.08794751         NA

[[4]]$svm.list[[4]]
  gamma cost      error dispersion
1     0.125   16 0.07830851         NA

[[4]]$svm.list[[5]]
  gamma cost      error dispersion
1     1    8 0.03742716         NA

[[4]]$svm.list[[6]]
  gamma cost     error dispersion
1     1   16 0.2024206         NA

m.list[[7]]
cost      error dispersion
  16 0.08247584         NA

m.list[[8]]
cost     error dispersion
```

```
1     1   16 0.1562139         NA

[[4]]$svm.list[[9]]
  gamma cost      error dispersion
1     1   16 0.02590237         NA

[[4]]$svm.list[[10]]
  gamma cost     error dispersion
1     1    4 0.0378918         NA


[[4]]$error
[1] 0.08144904


[[5]]
[[5]]$svm.list
[[5]]$svm.list[[1]]
  gamma cost     error dispersion
1     1   16 0.0638007         NA

[[5]]$svm.list[[2]]
  gamma cost      error dispersion
1     1   16 0.02574187         NA

[[5]]$svm.list[[3]]
  gamma cost      error dispersion
1     1   16 0.07232554         NA

[[5]]$svm.list[[4]]
  gamma cost      error dispersion
1  0.125   16 0.07856888         NA

[[5]]$svm.list[[5]]
  gamma cost      error dispersion
1   0.125    2 0.04569092         NA

[[5]]$svm.list[[6]]
  gamma cost     error dispersion
1 0.0078125    32 0.1442385         NA

[[5]]$svm.list[[7]]
  gamma cost      error dispersion
1 0.0002441406   8 0.06424712         NA

[[5]]$svm.list[[8]]
  gamma cost     error dispersion
1     1   16 0.1322539         NA

[[5]]$svm.list[[9]]
  gamma cost      error dispersion
1 0.001953125    64 0.02366163         NA

[[5]]$svm.list[[10]]
  gamma cost      error dispersion
1 0.0004882812    64 0.04128003         NA
```

ror
918091

n 19 : Hasil *Parameter Tuning* dan *Generalization Error* Untuk I SVM Kernel RBF

```
[[1]]
[[1]]$svm.list
[[1]]$svm.list[[1]]
     gamma cost     error dispersion
1 0.015625  0.5 0.3365165         NA
[[1]]$svm.list[[2]]
   gamma  cost     error dispersion
1 0.0625 0.125 0.3009035         NA
[[1]]$svm.list[[3]]
   gamma cost     error dispersion
1 0.0625  0.5 0.1979906         NA
[[1]]$svm.list[[4]]
    gamma cost     error dispersion
1 0.03125  0.5 0.1739892         NA
[[1]]$svm.list[[5]]
   gamma cost     error dispersion
1 0.0625  0.5 0.2045426         NA
[[1]]$svm.list[[6]]
   gamma cost     error dispersion
1 0.0625 0.25 0.1271786         NA
[[1]]$svm.list[[7]]
    gamma cost     error dispersion
1 0.03125    1 0.2100092         NA
[[1]]$svm.list[[8]]
   gamma  cost     error dispersion
1 0.0625 0.125 0.1986998         NA
[[1]]$svm.list[[9]]
   gamma cost     error dispersion
1 0.0625 0.25 0.1922481         NA
[[1]]$svm.list[[10]]
   gamma cost     error dispersion
1 0.0625 0.25 0.2338777         NA


[[1]]$error
[1] 0.2175956


[[2]]
[[2]]$svm.list
[[2]]$svm.list[[1]]
      gamma cost    error dispersion
1 0.0078125    2 0.357007         NA
[[2]]$svm.list[[2]]
      gamma cost     error dispersion
1 0.0078125    1 0.2764866         NA
[[2]]$svm.list[[3]]
     gamma cost     error dispersion
1 0.015625    1 0.2023308         NA
[[2]]$svm.list[[4]]
      gamma cost     error dispersion
1 0.0078125    2 0.2217585         NA
[[2]]$svm.list[[5]]
     gamma cost     error dispersion
1 0.015625    1 0.1694048         NA
[[2]]$svm.list[[6]]
         gamma cost    error dispersion
1 0.0002441406   32 0.138273         NA
[[2]]$svm.list[[7]]
        gamma cost     error dispersion
1 0.001953125    4 0.2056046         NA
[[2]]$svm.list[[8]]
   gamma cost     error dispersion
1 0.0625 0.25 0.1819256         NA
[[2]]$svm.list[[9]]
   gamma cost     error dispersion
1 0.0625 0.25 0.1694253         NA
[[2]]$svm.list[[10]]
   gamma cost     error dispersion
1 0.0625  0.5 0.1999878         NA


[[2]]$error
[1] 0.2022204
```

```
[[3]]
[[3]]$svm.list
[[3]]$svm.list[[1]]
  gamma cost     error dispersion
1     1    1 0.1976909         NA
[[3]]$svm.list[[2]]
  gamma cost     error dispersion
1 0.125   16 0.4025775         NA
[[3]]$svm.list[[3]]
  gamma cost     error dispersion
1     1    2 0.2277305         NA
[[3]]$svm.list[[4]]
  gamma cost     error dispersion
1     1    1 0.1876323         NA
[[3]]$svm.list[[5]]
     gamma cost     error dispersion
1 0.015625    1 0.1292285         NA
[[3]]$svm.list[[6]]
    gamma cost     error dispersion
1 0.03125  0.5 0.1068312         NA
[[3]]$svm.list[[7]]
  gamma cost     error dispersion
1  0.25    1 0.3135749         NA
[[3]]$svm.list[[8]]
   gamma cost     error dispersion
1 0.0625    8 0.1070064         NA

[[3]]$svm.list[[9]]
    gamma cost     error dispersion
1 0.03125  0.5 0.1599022         NA
[[3]]$svm.list[[10]]
       gamma cost     error dispersion
1 0.00390625    4 0.1266012         NA


[[3]]$error
[1] 0.1958776


[[4]]
[[4]]$svm.list
[[4]]$svm.list[[1]]
  gamma cost   error dispersion
1  0.5    4 0.21943         NA
[[4]]$svm.list[[2]]
  gamma cost     error dispersion
1     1    2 0.1979621         NA
[[4]]$svm.list[[3]]
  gamma cost     error dispersion
1     1    1 0.1024941         NA
[[4]]$svm.list[[4]]
  gamma cost     error dispersion
1  0.015625    4 0.06688963         NA
[[4]]$svm.list[[5]]
  gamma cost      error dispersion
1 0.015625    32 0.02410529         NA
[[4]]$svm.list[[6]]
  gamma cost     error dispersion
1 0.125    1 0.1081302         NA
[[4]]$svm.list[[7]]
  gamma cost     error dispersion
1   0.25    8 0.1571119         NA
[[4]]$svm.list[[8]]
  gamma cost      error dispersion
1   0.5    1 0.08922563         NA
      n.list[[9]]
      cost     error dispersion
         1 0.1306739         NA
      n.list[[10]]
      cost      error dispersion
         2 0.09556396         NA
```

```
[[4]]$error
[1] 0.1191587


[[5]]
[[5]]$svm.list
[[5]]$svm.list[[1]]
  gamma cost     error dispersion
1     1    2 0.2094516         NA
[[5]]$svm.list[[2]]
  gamma cost     error dispersion
1   0.5    4 0.2745526         NA
[[5]]$svm.list[[3]]
  gamma cost     error dispersion
1  0.125    2 0.1038287         NA
[[5]]$svm.list[[4]]
  gamma cost      error dispersion
1  0.25    4 0.07220657         NA
[[5]]$svm.list[[5]]
  gamma cost      error dispersion
1     1    1 0.02547558         NA
[[5]]$svm.list[[6]]
  gamma cost      error dispersion
1     1    1 0.08869062         NA
[[5]]$svm.list[[7]]
  gamma cost   error dispersion
1     1    2 0.15249         NA
[[5]]$svm.list[[8]]
  gamma cost      error dispersion
1     1    1 0.08465886         NA
[[5]]$svm.list[[9]]
  gamma cost     error dispersion
1     1    4 0.1224828         NA
[[5]]$svm.list[[10]]
  gamma cost      error dispersion
1     1    2 0.09096075         NA


[[5]]$error
[1] 0.1224798
```

Lampiran 20 : Hasil *Parameter Tuning* dan *Generalization Error* Untuk Dataset I SVM Kernel Polynomial

```
[[1]]
[[1]]$svm.list
[[1]]$svm.list[[1]]
    gamma cost coef0 degree     error dispersion
1 0.03125   32 0.125      2 0.4527254         NA

[[1]]$svm.list[[2]]
   gamma cost coef0 degree     error dispersion
1 0.0625    2 0.125      2 0.3018488         NA

[[1]]$svm.list[[3]]
         gamma cost coef0 degree     error dispersion
1 0.0009765625   64 0.125      2 0.3503893         NA

[[1]]$svm.list[[4]]
     gamma cost coef0 degree     error dispersion
25   32 0.125      2 0.2018518         NA

n.list[[5]]
mma cost coef0 degree     error dispersion
125   16 0.125      2 0.2526447         NA
```

```
[[1]]$svm.list[[6]]
  gamma cost coef0 degree     error dispersion
1   0.5    2 0.125      2 0.2536132         NA

[[1]]$svm.list[[7]]
  gamma cost coef0 degree     error dispersion
1 0.125    1 0.125      2 0.2203902         NA

[[1]]$svm.list[[8]]
         gamma cost coef0 degree     error dispersion
1 0.0009765625   64 0.125      2 0.2922065         NA

[[1]]$svm.list[[9]]
  gamma cost coef0 degree     error dispersion
1  0.25    4 0.125      2 0.5380771         NA

[[1]]$svm.list[[10]]
    gamma cost coef0 degree     error dispersion
1 0.0004882812    32 0.125      2 0.1569784         NA


[[1]]$error
[1] 0.3020725


[[2]]
[[2]]$svm.list
[[2]]$svm.list[[1]]
   gamma cost coef0 degree    error dispersion
1 0.0625    4 0.125      2 0.373629         NA

[[2]]$svm.list[[2]]
      gamma cost coef0 degree     error dispersion
1 0.0078125   64 0.125      2 0.2999057         NA

[[2]]$svm.list[[3]]
  gamma cost coef0 degree    error dispersion
1  0.25    2 0.125      2 0.391061         NA

[[2]]$svm.list[[4]]
  gamma cost coef0 degree    error dispersion
1   0.5    1 0.125      2 0.270539         NA

[[2]]$svm.list[[5]]
   gamma cost coef0 degree     error dispersion
1 0.0625    2 0.125      2 0.3280833         NA

[[2]]$svm.list[[6]]
   gamma cost coef0 degree     error dispersion
1 0.0625    1 0.125      2 0.2394803         NA

[[2]]$svm.list[[7]]
   gamma cost coef0 degree     error dispersion
1 0.0625    1 0.125      2 0.2844747         NA

[[2]]$svm.list[[8]]
         gamma cost coef0 degree    error dispersion
1 0.0009765625   64 0.125      2 0.296199         NA

n.list[[9]]
amma cost coef0 degree     error dispersion
0625   64 0.125      2 0.4239396         NA

n.list[[10]]
```

```
       gamma cost coef0 degree     error dispersion
1 0.00390625   64 0.125      2 0.2309862         NA


[[2]]$error
[1] 0.3138298


[[3]]
[[3]]$svm.list
[[3]]$svm.list[[1]]
      gamma cost coef0 degree     error dispersion
1 0.0078125    8 0.125      2 0.2571741         NA

[[3]]$svm.list[[2]]
  gamma cost coef0 degree     error dispersion
1 0.125   16 0.125      2 0.2998045         NA

[[3]]$svm.list[[3]]
  gamma cost coef0 degree     error dispersion
1 0.125    2 0.125      2 0.3693786         NA

[[3]]$svm.list[[4]]
  gamma cost coef0 degree     error dispersion
1   0.5 0.25 0.125      2 0.2019233         NA

[[3]]$svm.list[[5]]
     gamma cost coef0 degree     error dispersion
1 0.015625    8 0.125      2 0.4186042         NA

[[3]]$svm.list[[6]]
    gamma cost coef0 degree     error dispersion
1 0.03125    4 0.125      2 0.2438776         NA

[[3]]$svm.list[[7]]
    gamma cost coef0 degree    error dispersion
1 0.03125    2 0.125      2 0.280932         NA

[[3]]$svm.list[[8]]
  gamma cost coef0 degree     error dispersion
1 0.125   16 0.125      2 0.2087773         NA

[[3]]$svm.list[[9]]
   gamma cost coef0 degree     error dispersion
1 0.0625    4 0.125      2 0.6397695         NA

[[3]]$svm.list[[10]]
       gamma cost coef0 degree     error dispersion
1 0.00390625   32 0.125      2 0.2318861         NA


[[3]]$error
[1] 0.3152127


[[4]]
[[4]]$svm.list
[[4]]$svm.list[[1]]
      cost coef0 degree     error dispersion
         2 0.125      2 0.3322224         NA

    n.list[[2]]
    cost coef0 degree    error dispersion
     0.5 0.125      2 0.282839         NA
```

```
[[4]]$svm.list[[3]]
  gamma cost coef0 degree     error dispersion
1     1    2 0.125      2 0.3890552         NA

[[4]]$svm.list[[4]]
  gamma cost coef0 degree     error dispersion
1  0.25    1 0.125      2 0.3080832         NA

[[4]]$svm.list[[5]]
      gamma cost coef0 degree     error dispersion
1 0.0078125   32 0.125      2 0.4069794         NA

[[4]]$svm.list[[6]]
    gamma cost coef0 degree     error dispersion
1 0.03125    2 0.125      2 0.2447717         NA

[[4]]$svm.list[[7]]
   gamma cost coef0 degree     error dispersion
1 0.0625    1 0.125      2 0.2979693         NA

[[4]]$svm.list[[8]]
         gamma cost coef0 degree     error dispersion
1 0.0004882812   64 0.125      2 0.2949576         NA

[[4]]$svm.list[[9]]
    gamma cost coef0 degree     error dispersion
1 0.03125    4 0.125      2 0.5513669         NA

[[4]]$svm.list[[10]]
   gamma cost coef0 degree     error dispersion
1 0.0625  0.5 0.125      2 0.2399656         NA


[[4]]$error
[1] 0.334821


[[5]]
[[5]]$svm.list
[[5]]$svm.list[[1]]
     gamma cost coef0 degree     error dispersion
1 0.015625    8 0.125      2 0.3115689         NA

[[5]]$svm.list[[2]]
  gamma cost coef0 degree     error dispersion
1 0.125  0.5 0.125      2 0.3149428         NA

[[5]]$svm.list[[3]]
  gamma cost coef0 degree     error dispersion
1     1  0.5 0.125      2 0.3775199         NA

[[5]]$svm.list[[4]]
  gamma cost coef0 degree     error dispersion
1   0.5    1 0.125      2 0.2661316         NA

[[5]]$svm.list[[5]]
    gamma cost coef0 degree     error dispersion
      5    2 0.125      2 0.3957291         NA

      n.list[[6]]
      gamma cost coef0 degree     error dispersion
      53125   32 0.125      2 0.2341253         NA
```

```
[[5]]$svm.list[[7]]
     gamma cost coef0 degree     error dispersion
1 0.015625   64 0.125      2 0.3105456         NA

[[5]]$svm.list[[8]]
         gamma cost coef0 degree     error dispersion
1 0.0009765625   16 0.125      2 0.2879627         NA

[[5]]$svm.list[[9]]
    gamma cost coef0 degree     error dispersion
1 0.03125    8 0.125      2 0.3333005         NA

[[5]]$svm.list[[10]]
   gamma cost coef0 degree    error dispersion
1 0.0625  0.5 0.125      2 0.229721         NA


[[5]]$error
[1] 0.3061547
```

Lampiran 21 : Hasil *Parameter Tuning* dan *Generalization Error* Untuk Dataset II SVM Kernel Polynomial

```
[[1]]
[[1]]$svm.list
[[1]]$svm.list[[1]]
    gamma cost coef0 degree     error dispersion
1 0.03125   64 0.125      2 0.4790431         NA

[[1]]$svm.list[[2]]
      gamma cost coef0 degree     error dispersion
1 0.0078125   64 0.125      2 0.3000788         NA

[[1]]$svm.list[[3]]
        gamma cost coef0 degree     error dispersion
1 0.001953125   32 0.125      2 0.3549434         NA

[[1]]$svm.list[[4]]
    gamma cost coef0 degree     error dispersion
1 0.03125    8 0.125      2 0.2022887         NA

[[1]]$svm.list[[5]]
     gamma cost coef0 degree     error dispersion
1 0.015625   64 0.125      2 0.2734348         NA

[[1]]$svm.list[[6]]
  gamma cost coef0 degree     error dispersion
1     1    1 0.125      2 0.2537182         NA

[[1]]$svm.list[[7]]
   gamma cost coef0 degree    error dispersion
1 0.0625    4 0.125      2 0.224813         NA

[[1]]$svm.list[[8]]
         gamma cost coef0 degree     error dispersion
1 0.0009765625   64 0.125      2 0.2922065         NA

[[1]]$svm.list[[9]]
  gamma cost coef0 degree     error dispersion
1  0.25    4 0.125      2 0.5380786         NA

[[1]]$svm.list[[10]]
    gamma cost coef0 degree     error dispersion
    25    16 0.125      2 0.1572367         NA


ror
75584
```

```
[[2]]
[[2]]$svm.list
[[2]]$svm.list[[1]]
   gamma cost coef0 degree    error dispersion
1 0.0625    4 0.125      2 0.373629         NA

[[2]]$svm.list[[2]]
      gamma cost coef0 degree     error dispersion
1 0.0078125   64 0.125      2 0.2999057         NA

[[2]]$svm.list[[3]]
  gamma cost coef0 degree    error dispersion
1  0.25    2 0.125      2 0.391061         NA

[[2]]$svm.list[[4]]
  gamma cost coef0 degree    error dispersion
1   0.5    1 0.125      2 0.270539         NA

[[2]]$svm.list[[5]]
   gamma cost coef0 degree     error dispersion
1 0.0625    2 0.125      2 0.3280833         NA

[[2]]$svm.list[[6]]
   gamma cost coef0 degree     error dispersion
1 0.0625    1 0.125      2 0.2394803         NA

[[2]]$svm.list[[7]]
   gamma cost coef0 degree     error dispersion
1 0.0625    1 0.125      2 0.2844747         NA

[[2]]$svm.list[[8]]
        gamma cost coef0 degree    error dispersion
1 0.0009765625   64 0.125      2 0.296199         NA

[[2]]$svm.list[[9]]
       gamma cost coef0 degree     error dispersion
1 0.00390625   64 0.125      2 0.4239396         NA

[[2]]$svm.list[[10]]
       gamma cost coef0 degree     error dispersion
1 0.00390625   64 0.125      2 0.2309862         NA


[[2]]$error
[1] 0.3138298


[[3]]
[[3]]$svm.list
[[3]]$svm.list[[1]]
      gamma cost coef0 degree     error dispersion
1 0.0078125    8 0.125      2 0.2571741         NA

[[3]]$svm.list[[2]]
  gamma cost coef0 degree     error dispersion
1 0.125   16 0.125      2 0.2998045         NA

[[3]]$svm.list[[3]]
  gamma cost coef0 degree     error dispersion
1 0.125    2 0.125      2 0.3693786         NA

[[3]]$svm.list[[4]]
  gamma cost coef0 degree     error dispersion
1   0.5 0.25 0.125      2 0.2019233         NA

[[3]]$svm.list[[5]]
    gamma cost coef0 degree     error dispersion
    25    8 0.125      2 0.4186042         NA

    n.list[[6]]
    a cost coef0 degree     error dispersion
    5    4 0.125      2 0.2438776         NA

    n.list[[7]]
```

```
    gamma cost coef0 degree    error dispersion
1 0.03125    2 0.125      2 0.280932         NA

[[3]]$svm.list[[8]]
  gamma cost coef0 degree     error dispersion
1 0.125   16 0.125      2 0.2087773         NA

[[3]]$svm.list[[9]]
   gamma cost coef0 degree     error dispersion
1 0.0625    4 0.125      2 0.6397695         NA

[[3]]$svm.list[[10]]
       gamma cost coef0 degree     error dispersion
1 0.00390625   32 0.125      2 0.2318861         NA


[[3]]$error
[1] 0.3152127


[[4]]
[[4]]$svm.list
[[4]]$svm.list[[1]]
   gamma cost coef0 degree     error dispersion
1 0.0625    2 0.125      2 0.3322224         NA

[[4]]$svm.list[[2]]
  gamma cost coef0 degree    error dispersion
1   0.5  0.5 0.125      2 0.282839         NA

[[4]]$svm.list[[3]]
  gamma cost coef0 degree     error dispersion
1     1    2 0.125      2 0.3890552         NA

[[4]]$svm.list[[4]]
  gamma cost coef0 degree     error dispersion
1  0.25    1 0.125      2 0.3080832         NA

[[4]]$svm.list[[5]]
      gamma cost coef0 degree     error dispersion
1 0.0078125   32 0.125      2 0.4069794         NA

[[4]]$svm.list[[6]]
    gamma cost coef0 degree     error dispersion
1 0.03125    2 0.125      2 0.2447717         NA

[[4]]$svm.list[[7]]
   gamma cost coef0 degree     error dispersion
1 0.0625    1 0.125      2 0.2979693         NA

[[4]]$svm.list[[8]]
         gamma cost coef0 degree     error dispersion
1 0.0004882812   64 0.125      2 0.2949576         NA

[[4]]$svm.list[[9]]
    gamma cost coef0 degree     error dispersion
1 0.03125    4 0.125      2 0.5513669         NA

[[4]]$svm.list[[10]]
   gamma cost coef0 degree     error dispersion
1 0.0625  0.5 0.125      2 0.2399656         NA


[[4]]$error
[1] 0.334821
```

```
n.list
n.list[[1]]
na cost coef0 degree     error dispersion
25    8 0.125      2 0.3115689         NA

n.list[[2]]
```