

DAFTAR PUSTAKA

- W. Kumara, P. Kumara and F. Sourjah, "Value Customer Reviews in E-Commerce Platforms using Sentiment Analysis Tactics: A Review," 2022 3rd International Conference for Emerging Technology (INCET), 2022, pp. 1-6
- Chenyue Wang, Xiaodong Zhu, and Lirong Yan, "Sentiment Analysis for E-Commerce Reviews Based on Deep Learning Hybrid Model," 2022 5th International Conference on Signal Processing and Machine Learning, 2022, pp. 38–46.
- J. Clement, "Retail e-commerce sales growth worldwide 2017-2023," <https://www.statista.com/statistics/288487/forecast-of-global-b2c-ecommerce-growth/>, 2020]
- D. Kowanda, M. Firdaus, R. B. F. Pasaribu and S. Nawangsari, "Lesson from Tokopedia.com: E-commerce success factor analysis: A case study from Indonesian unicorn," 2018 International Conference on Information Management and Processing (ICIMP), 2018, pp. 61-65.
- M. A. Casas-Valadez, A. Faz-Mendoza, C. E. Medina-Rodríguez, R. Castañeda-Miranda, N. K. Gamboa-Rosales and J. R. López-Robles, "The synergic relationship between e-commerce and Sentiment Analysis: A content analysis of published articles between 2007 and 2020," 2020 International Conference on Data Analytics for Business and Industry: Way Towards a Sustainable Economy (ICDABI), 2020, pp. 1-5.
- O. Appel, F. Chiclana, J. Carter and H. Fujita, "A hybrid approach to sentiment analysis," 2016 IEEE Congress on Evolutionary Computation (CEC), 2016, pp. 4950-4957.
- W. Wu, Y. Wang, S. Xu and K. Yan, "SFNN: Semantic Features Fusion Neural Network for Multimodal Sentiment Analysis," 2020 5th International Conference on Automation, Control and Robotics Engineering (CACRE), 2020, pp. 661-665.
- Narayana Kamath, Cannannore Nidhi & Bukhari, Syed & Dengel, Andreas, "Comparative Study between Traditional Machine Learning and Deep

- Learning Approaches for Text Classification, " 2018 18th ACM Symposium on Document Engineering (DocEng18), 2018, pp. 1-11.
- N. Srivats Athindran, S. Manikandaraj and R. Kamaleshwar, "Comparative Analysis of Customer Sentiments on Competing Brands using Hybrid Model Approach," 2018 3rd International Conference on Inventive Computation Technologies (ICICT), 2018, pp. 348-353.
- S. Juanita, K. Adiyarta and M. Syafrullah, "Sentiment analysis on E-Marketplace User Opinions Using Lexicon-Based and Naïve Bayes Model," 2022 9th International Conference on Electrical Engineering, Computer Science and Informatics (EECSI), 2022, pp. 379-382.
- A. N. Rohman, R. Luviana Musyarofah, E. Utami and S. Raharjo, "Natural Language Processing on Marketplace Product Review Sentiment Analysis," 2020 2nd International Conference on Cybernetics and Intelligent System (ICORIS), 2020, pp. 1-5.
- T. Sutabri, S. J. Putra, M. R. Effendi, M. N. Gunawan and D. Napitupulu, "Sentiment Analysis for Popular e-traveling Sites in Indonesia using Naive Bayes," 2018 6th International Conference on Cyber and IT Service Management (CITSM), 2018, pp. 1-4.
- F. WU, Z. SHI, Z. DONG, C. PANG and B. ZHANG, "Sentiment Analysis of Online Product Reviews Based On SenBERT-CNN," 2020 International Conference on Machine Learning and Cybernetics (ICMLC), 2020, pp. 229-234.
- M. M. Dwi Putra, W. Fawwaz Al Maki and A. Romadhony, "Sentiment Analysis on Marketplace Review using Hybrid Lexicon and SVM Method," 2021 9th International Conference on Information and Communication Technology (ICoICT), 2021, pp. 66-70.
- D. Ananda, T. A. Taqiyuddin, I. Nugraha Faqih, R. Badrahadipura and A. A. Pravitasari, "Application of Bidirectional Gated Recurrent Unit (BiGRU) in Sentiment Analysis of Tokopedia Application Users," 2021 International Conference on Artificial Intelligence and Big Data Analytics, 2021, pp. 1-4.

- N. S. Kustanto, I. Nurma Yulita and I. Sarathan, "Sentiment Analysis of Indonesia's National Health Insurance Mobile Application using Naïve Bayes Algorithm," 2021 International Conference on Artificial Intelligence and Big Data Analytics, Bandung, Indonesia, 2021, pp. 38-42, doi: 10.1109/ICAIBDA53487.2021.9689696.
- S. N. Singh and T. Sarraf, "Sentiment Analysis of a Product based on User Reviews using Random Forests Algorithm," 2020 10th International Conference on Cloud Computing, Data Science & Engineering (Confluence), Noida, India, 2020, pp. 112-116, doi: 10.1109/Confluence47617.2020.9058128.
- M. Mahyarani, A. Adiwijaya, S. A. Faraby and M. Dwifabri, "Implementation of Sentiment Analysis Movie Review based on IMDB with Naive Bayes Using Information Gain on Feature Selection," 2021 3rd International Conference on Electronics Representation and Algorithm (ICERA), Yogyakarta, Indonesia, 2021, pp. 99-103, doi: 10.1109/ICERA53111.2021.9538763.
- R. Uma, A. S. H, P. Jawahar and B. V. Rishitha, "Support Vector Machine and Convolutional Neural Network Approach to Customer Review Sentiment Analysis," 2022 1st International Conference on Computational Science and Technology (ICCST), CHENNAI, India, 2022, pp. 239-243, doi: 10.1109/ICCST55948.2022.10040381.
- M. Kamruzzaman, M. Hossain, M. R. I. Imran and S. C. Bakchy, "A Comparative Analysis of Sentiment Classification Based on Deep and Traditional Ensemble Machine Learning Models," 2021 International Conference on Science & Contemporary Technologies (ICSCT), Dhaka, Bangladesh, 2021, pp. 1-5, doi: 10.1109/ICSCT53883.2021.9642583.

LAMPIRAN

35861	masak cuma pkek kresek,-1								
35862	soalnya pengalaman saya dengan jasa pengiriman agak kurang enk,-1								
35863	dan barang saya hancur di bkin di jln sma mereka terus nggak mau di gnti rugi,-1								
35864	sama fontnya nggak norak,-1								
35865	terpaksa ditelen saja,-1								
35866	kok tetumbenan bikin keyboard low profile seperti gini,-1								
35867	nggak ada bentuk kaya megatron,-1								
35868	kenapa ya usb through nya tidak berfungsi,-1								
35869	sudah di chat seller bilang nya stock banyak untuk tactilenya dikirim nya gl switch - tanpa konfirmasi,-1								
35870	di komplain,-1								
35871	tetapi didiemin hampir dua hari kurang beberapa jam terus komplain di terima bilang nya masuk indo cu								
35872	jadi ya terpaksa diterima saja,-1								
35873	kecewa banget sih,-1								
35874	dua di komplain harus banget sampai dua hari baru diproses,-1								
35875	pesanan yang ke dua kalinya ajib banget sprti yang pertama,-1								
35876	pilihan warnanya sudah otomatis jadi nggak bisa dicustom,-1								
35877	sic parah juga ya,-1								
35878	dan seller slow respons,-1								
35879	kadang nggak ke pencet padahal yang lain key nya normal,-1								
35880	enternya agak macet,-1								
35881	jadi nggak rata,-1								
35882	cuma kecewa dengan keyboard nya,-1								
35883	semua ke reset lagi,-1								
35884	harus setting ulang setiap nyalain pc,-1								
35885	karena benar2 merepotkan sekali,-1								
35886	berisik banget ini keyboard,-1								
35887	mau maks dibiain pun nggak bisa juga,-1								
35888	sudah pemakaian seminggu dan gue kecewa berat,-1								
35889	suaranya doang kenceng tombol nya keras tetapi jeroannya payah,-1								
35890	ini barang belum siap dilempar pasar,-1								
35891	cuma bad di pengiriman saja,-1								
35892	ada kesalahan layout keys,-1								
35893	kesulitan yang saya alami adalah sering nya tertekan tombol lain yang tidak saya pijat contoh tertekanny								
35894	2 tidak memiliki aplikasi pendukung sehingga ketika anda mematikan pc atau mencabut nya dari lubang								
35895	kurang lebih seperti suara fan pada laptop,-1								
35896	sayangnya kartu garansinya nggak di isi dan nggak ada stempel nya,-1								
35897	kekurangan cuma nggak ada software yang buat ngatur lampunya,-1								
35898	awalnya ada yang part bonus tidak terkirim,-1								
	< > negative +								

500	tk bk dulu,0								
501	untuk produk sih lima cuma untuk kurirnya kalau blh minus saya kasi minus,0								
502	storage tiga puluh dua gb,0								
503	plastiknya sudah kebuka,0								
504	coba persen lecet karena bajet minim,0								
505	unit ke dua dari store ini,0								
506	terus dari pesan tiga jam kmudian ud dpn rmhh,0								
507	belum coba tes,0								
508	tetapi cuma ada iring nya saja,0								
509	tetapi barang yang ada battery nya macam hp dan remote mobil jangan masukan,0								
510	yups nggak karatan,0								
511	jadi lumayan merepotkan,0								
512	sedikit masukkan,0								
513	next order lagi ðŸ™,0								
514	namanya teko bunyi tetapi pas masak air nggak bunyi pertama tamanya doang bunyi,0								
515	imei terdaftar no cacat,0								
516	cuma bisa foto kotaknya saja,0								
517	sudah ditulis jangan robek penyok dan wajib segel plastik tulisan samsung kiri kanan,0								
518	ada keanehan pas ngatur rgb di software steelseries nya untuk aplikasi csgo,0								
519	tahu bocor pas mau dipakai atau diisi air,0								
520	hanya kurang bonus powerbank,0								
521	masih di simpan,0								
522	mulai order tanggal enam belas okt pengiriman ke tangsel pas enam hari,0								
523	gimana caranya ya,0								
524	mana resinya h satu pengiriman,0								
525	adminnya konfirmasi dulu kalau soal warna yang nggak ada,0								
526	cuma yang pojok cacat sedikit,0								
527	ni order yang ke dua,0								
528	layanan betanggung jawab dan responsif,0								
529	barang sampai tetapi kok wattnya gede ya sampai enam ratus di desknya cuma empat ratus lima puluh,0								
530	takut dapat barang yang nggak mulus,0								
531	dikirim ch atau a,0								
532	saya punya preferensi utensil yang agak besar,0								
533	tetapi gatau kenapa sampai rumah basah kena hujan,0								
534	casing delapan atau sepuluh,0								
535	mungkin karna gratis ongkir kalee,0								
536	untuk ngambil bahan tepung melengkung,0								
537	barang dipakai normal,0								

	A	B	C	D	E	F	G	H
1	text,label							
2	bagus,1							
3	sudah tiba dan prosesnya cepat,1							
4	barang bagus dan tanpa cacat,1							
5	plastikan dan langsung dikasih kardus pengiriman,1							
6	barang sesuai gambar pengoriman cepat,1							
7	dan ternyata harga di pasar jauh lebih murah,1							
8	keren,1							
9	cepat sampainya,1							
10	belum dicoba semoga pisau nya tajam dan barang awet,1							
11	sesuai dengan gambar dan fungsinya,1							
12	tajam pisau nya,1							
13	pengiriman oke,1							
14	sepertinya akan berfungsi dengan baik,1							
15	untung nggak pecah ðŸ˜†,1							
16	barang bagus sesuai iklan,1							
17	response dan pengiriman cepat,1							
18	good,1							
19	produk sesuai dan lengkap,1							
20	bagus barang than pecah karena dari bahan plasrtik yang bagus,1							
21	alhamdulillah produk da sampai,1							
22	barang bagus,1							
23	suka ðŸ˜’,1							
24	sangat bagus,1							
25	smuanya ðŸ˜’,1							
26	top banget,1							
27	nggak nyangka ada bonus yang nggak dijelasin di deskripsi,1							
28	puas deh ðŸ˜’,1							
29	proses cepat dan semoga barang bermanfaat,1							
30	lumayan bagus itemnya dan sesuai deskripsi,1							
31	semoga awet ya,1							
32	thanks seller juga tokopedia,1							
33	barang sesuai foto,1							
34	smoga berfungsi baik ðŸ˜—,1							
35	packing rapi aman juga karena ditambah plastic wrap dan pengiriman cepat,1							
36	thanks seller,1							
37	packing rapi,1							
38	kecepatan pengiriman oke,1							

positive



:92589	biar bintang saja yang berbicara wkwkw,1				
:92590	barang berfungsi dengan baik sesuai dengan pesanan,1				
:92591	worth it banget buat dibeli,1				
:92592	keyboard nya nyaman,1				
:92593	barang nya recommended banget bagus,1				
:92594	barang mantap mantap laa,1				
:92595	keyboard mechanical ini benar - benar nyaman digunakan dan sedikit membuat sec				
:92596	keyboard mechanical pertama yang gua punya,1				
:92597	dapat wrist rest dan yang penting build quality nya mantap dan harga bersahabat,1				
:92598	produk bagus dan berkualitas sesuai dengan harga,1				
:92599	penggunaan juga nyaman,1				
:92600	dan paling mantul,1				
:92601	pengiriman dapat dibantu dengan sangat cepat di weekend,1				
:92602	produk yang sangat uwu dengan suara mechanical yang lumayan tacky tetapi enak c				
:92603	lima atau lima untuk produk,1				
:92604	lima atau lima untuk penjual,1				
:92605	lumayan lah ðŸ‘œ,1				
:92606	dan sampai tujuan dengan baik,1				
:92607	pedagang sangat responsif dan juga cepat tanggap,1				
:92608	mantap bro ðŸ‘Ÿ,1				
:92609	barang nya bagus hanya packing saja harus di rapiin,1				
:92610	original keyboard,1				
:92611	product mantap lancar,1				
:92612	feels clicky nya dapat banget,1				
:92613	batang cepat sampai dan kualitas produk nya oke,1				
:92614	kliknya satisfying banget,1				
:92615	ya dengan kelas harganya sudah the best ini,1				
:92616	tetapi penjual baik sudah dikirim ulang,1				
:92617	magnet di handrestnya kuat,1				
:92618	semua tombol dan antighostingnya berfungsi dengan baik,1				
:92619	rgb nya juga nggak ada masalah,1				
:92620	thanks rexeus ðŸ™†,1				
:92621	sayang banget max lima,1				
:92622	tetapi tetap nyaman untuk range harga segini,1				
:92623	datengnya dalam waktu cuma dua hari kerja saja,1				
:92624	dan paket datang dalam keadaan mulus walaupun kardus sedikit penyok,1				
:92625	button berfungsi semua terima kasih,1				
:92626	untuk harga promo sangat recommended untuk di beli,1				
	< > positive (+)				


```

In [7]: # import kebutuhan cross validasi
from sklearn.model_selection import KFold # import KFold
import numpy as np

# bikin array yg bisa scalar
data_train = np.array(data_train)
label_train = np.array(label_train)

# definisikan berapa banyak fold di cross validasi
kf = KFold(n_splits=10)
kf.get_n_splits(data_train)

# tampilkan konfig cv (shuffle false krna sudah random diatas)
print(kf)

KFold(n_splits=10, random_state=None, shuffle=False)

In [8]: from statistics import mode
from sklearn import metrics

# fungsi hitung rerata/mean probabilitas
def mean(lst):
    return sum(lst) / len(lst)

n_fold = 1
for train_index, test_index in kf.split(data_train):
    print('urutan fold cv ke-', n_fold)
    #print('TRAIN: ', train_index, 'TEST: ', test_index)

    # inisialisasi data training dan test tiap iterasi cross validasi
    X_train_i, X_test_i = data_train[train_index], data_train[test_index]
    y_train_i, y_test_i = label_train[train_index], label_train[test_index]

    # vectorkan kata2 dan bikin label X, y
    # print(X_train[n_fold].tolist())
    X_train = X_train_i.tolist()
    X_test = X_test_i.tolist()
    y_train = y_train_i.tolist()
    y_test = y_test_i.tolist()

    vectorizer = TfidfVectorizer(min_df=1)
    X_train = vectorizer.fit_transform(X_train)
    X_test = vectorizer.transform(X_test)

    # training NB
    klasifikasi_NB = MultinomialNB()
    model_NB = klasifikasi_NB.fit(X_train, y_train) #training

    # training random forest
    klasifikasi_RF = RandomForestClassifier()
    model_RF = klasifikasi_RF.fit(X_train, y_train) #training

    # training SVM
    pipeline_SVM = Pipeline([('clf', OneVsRestClassifier(CalibratedClassifierCV(LinearSVC(class_weight="balanced"))))])
    model_SVM = pipeline_SVM.fit(X_train, y_train) #training

    # SS = sentiment selector
    # SV = soft voting
    i = 0
    nb_result = []
    rf_result = []
    svm_result = []
    voting_data_SS = []
    voting_result_SS = []
    voting_data_SV = []
    voting_result_SV = []
    for item in y_test:
        # prediksi NB
        prediksi_NB = model_NB.predict(X_test[i])
        val_proba_NB = probPredict(model_NB.predict_proba(X_test[i]))
        # simpan hasil NB untuk precision, recall, f1score
        nb_result.append(prediksi_NB[0])

        # prediksi RF
        prediksi_RF = model_RF.predict(X_test[i])
        val_proba_RF = probPredict(model_RF.predict_proba(X_test[i]))
        # simpan hasil RF untuk precision, recall, f1score
        rf_result.append(prediksi_RF[0])

        # prediksi SVM
        prediksi_SVM = model_SVM.predict(X_test[i])
        val_proba_SVM = probPredict(model_SVM.predict_proba(X_test[i]))
        # simpan hasil SVM untuk precision, recall, f1score
        svm_result.append(prediksi_SVM[0])

    # S: bagian sentiment selector
    # gabung tiap hasil prediksi metode
    all_hasil_prediksi = [prediksi_NB[0], prediksi_RF[0], prediksi_SVM[0]]
    voting_data_SS.append(all_hasil_prediksi)
    # lakukan sentiment voting
    hasil_akhir = mode(all_hasil_prediksi)
    # simpan label dan hasil prediksi
    voting_result_SS.append(hasil_akhir)
    # E: bagian sentiment selector

    # S: bagian soft voting
    # gabung tiap hasil probabilitas (urutan -1,0,1)
    proba_negative = [val_proba_NB[0], val_proba_RF[0], val_proba_SVM[0]]
    proba_neutral = [val_proba_NB[1], val_proba_RF[1], val_proba_SVM[1]]
    proba_positif = [val_proba_NB[2], val_proba_RF[2], val_proba_SVM[2]]

```

```

proba_neutral = [vai_proba_NB[1], vai_proba_RF[1], vai_proba_SVM[1]]
proba_positive = [vai_proba_NB[2], vai_proba_RF[2], vai_proba_SVM[2]]
# hitung rerata/mean probabilitas tiap kelas
mean_negative = mean(proba_negative)
mean_neutral = mean(proba_neutral)
mean_positive = mean(proba_positive)
# pilih kelas dengan proba tertinggi
urutan_kelas = ['-1', '0', '1']
urutan_proba = [mean_negative, mean_neutral, mean_positive]
# simpan hasil proba
voting_data_SV.append(urutan_proba)
# cari urutan yg paling max nilainya
max_value = max(urutan_proba)
max_index = urutan_proba.index(max_value)
# simpan hasil soft voting
voting_result_SV.append(urutan_kelas[max_index])
# E: bagian soft voting

# pindah data test berikutnya
i += 1

# tampilkan hasil NB
print('-----')
print('Naive Bayes')
print('1. accuracy model NB saja :', metrics.accuracy_score(y_test, nb_result))
print('2. precision model NB saja :', metrics.precision_score(y_test, nb_result,average='macro'))
print('3. recall model NB saja :', metrics.recall_score(y_test, nb_result,average='macro'))
print('4. f1-score model NB saja :', metrics.f1_score(y_test, nb_result,average='macro'))

# tampilkan hasil RF
print('-----')
print('Random Forest')
print('1. accuracy model RF saja :', metrics.accuracy_score(y_test, rf_result))
print('2. precision model RF saja :', metrics.precision_score(y_test, rf_result,average='macro'))
print('3. recall model RF saja :', metrics.recall_score(y_test, rf_result,average='macro'))
print('4. f1-score model RF saja :', metrics.f1_score(y_test, rf_result,average='macro'))

# tampilkan hasil SVM
print('-----')
print('Support Vector Machine')
print('1. accuracy model SVM saja :', metrics.accuracy_score(y_test, svm_result))
print('2. precision model SVM saja :', metrics.precision_score(y_test, svm_result,average='macro'))
print('3. recall model SVM saja :', metrics.recall_score(y_test, svm_result,average='macro'))
print('4. f1-score model SVM saja :', metrics.f1_score(y_test, svm_result,average='macro'))

#print confusion matrix Sentiment Selector
print('-----')
print('Sentiment selector (SS)')
print('Confusion matrix SS: \n', metrics.confusion_matrix(y_test, voting_result_SS))
print('1. Akurasi SS: ', metrics.accuracy_score(y_test, voting_result_SS))
print('2. Precision SS: ', metrics.precision_score(y_test, voting_result_SS,average='macro'))
print('3. Recall SS: ', metrics.recall_score(y_test, voting_result_SS,average='macro'))
print('4. F1-score SS: ', metrics.f1_score(y_test, voting_result_SS,average='macro'))

print('-----')
#print confusion matrix Soft Voting
print('Soft Voting (SV)')
print('Confusion matrix SV: \n', metrics.confusion_matrix(y_test, voting_result_SV))
print('1. Akurasi SV: ', metrics.accuracy_score(y_test, voting_result_SV))
print('2. Precision SV: ', metrics.precision_score(y_test, voting_result_SV,average='macro'))
print('3. Recall SV: ', metrics.recall_score(y_test, voting_result_SV,average='macro'))
print('4. F1-score SV: ', metrics.f1_score(y_test, voting_result_SV,average='macro'))

print('\n')
n_fold += 1

```

```
urutan fold cv ke- 1
```

```

-----
Naive Bayes
1. accuracy model NB saja : 0.6266666666666667
2. precision model NB saja : 0.6252161881701833
3. recall model NB saja : 0.6166380789022299
4. f1-score model NB saja : 0.6195054945054945
-----
Random Forest
1. accuracy model RF saja : 0.6733333333333333
2. precision model RF saja : 0.7043610113733657
3. recall model RF saja : 0.6855536497045932
4. f1-score model RF saja : 0.6747397929073448
-----
Support Vector Machine
1. accuracy model SVM saja : 0.6466666666666666
2. precision model SVM saja : 0.6379813527529515
3. recall model SVM saja : 0.6350485991995426
4. f1-score model SVM saja : 0.6360515726369385

```


jupyter 0-random-sampling Last Checkpoint: 12/26/2022 (unsaved changes) Python 3 (ipykernel)

File Edit View Insert Cell Kernel Widgets Help Not Trusted

```
In [28]: import csv # import paket csv
from sklearn.utils import shuffle

import pandas as pd
```

```
In [29]: # random pick negative dataset
dataset_negative = pd.read_csv("dataset/negative.csv")
dataset_neutral = pd.read_csv("dataset/neutral.csv")
dataset_positive = pd.read_csv("dataset/positive.csv")
```

```
In [51]: # random sample
random_sample_negative = dataset_negative.sample(n=500)
random_sample_neutral = dataset_neutral.sample(n=500)
random_sample_positive = dataset_positive.sample(n=500)

print(random_sample_negative)
print(random_sample_neutral)
print(random_sample_positive)
```

	text	label
5732	bahannya kasar	-1.0
25221	chat lama diblis padahal sedang online	-1.0
9994	pakingnya kurang bagus waktu saya terima kondi...	-1.0
20982	masa hp sama charger nggak sesuai	-1.0
7793	bahan tidak sama dengan aslinya	-1.0
...
10617	tutup nya patah	-1.0
6717	NaN	NaN
6988	trlalu tipis	-1.0
31381	didiskripsi tidak di kasih keterangan silent	-1.0
20050	gue agak kecewa sama penjual lapak ini karena ...	-1.0

[500 rows x 2 columns]

	text	label
54111	tk bk dulu	0.0
18814	untuk produk sih lima cuma untuk kurirnya kala...	0.0
61145	storage tiga puluh dua gb	0.0
18979	plastiknya sudah kebuka	0.0
77628	coba persen lecet karena bajet minim	0.0
...
14122	segampang itu tanggung jawab penjual	0.0
33188	barang sesuai cuma ada sedikit lecet2 bagian b...	0.0
81900	tetapi kalau order	0.0
9684	aneh nya yang punya rumah terima saja itu barang	0.0
33730	mohon diperhatikan lagi dibagian pengemasan	0.0

[500 rows x 2 columns]

	text	label
299972	sangat rekomendasi belanja toko ini untuk yang...	1.0
3675	mantaaf	1.0
88166	mantap si ini mah sungguh cocok anak kost banget	1.0
341502	karena respons dan after salesnya bagus banget	1.0
25623	recommended 🤔👍	1.0
...
251993	sukses untuk tokonya dan recommended banget un...	1.0
324007	cuma layarnya agak lecet dikit tetapi bodynya ...	1.0
189812	kemarin sore order siang ini barang sudah sampai	1.0
155063	sikatnya high quality banget	1.0
218022	nokia is the best	1.0

[500 rows x 2 columns]

```
In [52]: # combine
frames = [random_sample_negative, random_sample_neutral, random_sample_positive]
result = pd.concat(frames)
display(result)
```

	text	label
5732	bahannya kasar	-1.0
25221	chat lama diblis padahal sedang online	-1.0
9994	pakingnya kurang bagus waktu saya terima kondi...	-1.0
20982	masa hp sama charger nggak sesuai	-1.0
7793	bahan tidak sama dengan aslinya	-1.0
...
251993	sukses untuk tokonya dan recommended banget un...	1.0
324007	cuma layarnya agak lecet dikit tetapi bodynya ...	1.0
189812	kemarin sore order siang ini barang sudah sampai	1.0
155063	sikatnya high quality banget	1.0
218022	nokia is the best	1.0

1500 rows x 2 columns

```
In [53]: # write to csv
result.to_csv('dataset/sample.csv', index=False)
```

```

In [11]: # gabung tiap hasil prediksi metode
all_hasil_prediksi = [prediksi_NB[0], prediksi_RF[0], prediksi_SVM[0]]
print(all_hasil_prediksi)

['-1', '1', '-1']

In [12]: # pilih hasil sentiment / sentiment selector
from statistics import mode
hasil_akhir = mode(all_hasil_prediksi)
print(hasil_akhir)

-1

In [11]: # gabung tiap hasil probabilitas (urutan -1,0,1)
proba_negative = [val_proba_NB[0], val_proba_RF[0], val_proba_SVM[0]]
proba_neutral = [val_proba_NB[1], val_proba_RF[1], val_proba_SVM[1]]
proba_positive = [val_proba_NB[2], val_proba_RF[2], val_proba_SVM[2]]

print("probabilitas negative NB, RF, SVM:", proba_negative)
print("probabilitas neutral NB, RF, SVM:", proba_neutral)
print("probabilitas positive NB, RF, SVM:", proba_positive)

probabilitas negative NB, RF, SVM: [0.20701917007469325, 0.24, 0.1276168941706923]
probabilitas neutral NB, RF, SVM: [0.04012670478797156, 0.01, 0.027583988723366475]
probabilitas positive NB, RF, SVM: [0.7528541251373356, 0.75, 0.8447991171059412]

In [12]: # hitung rerata/mean probabilitas tiap kelas
def mean(lst):
    return sum(lst) / len(lst)

mean_negative = mean(proba_negative)
mean_neutral = mean(proba_neutral)
mean_positive = mean(proba_positive)

print("mean negative: ", mean_negative)
print("mean neutral: ", mean_neutral)
print("mean positive: ", mean_positive)

mean negative: 0.19154535474846188
mean neutral: 0.025903564503779345
mean positive: 0.782551080747759

In [13]: # pilih kelas dengan proba tertinggi
urutan_kelas = [-1, 0, 1]
urutan_proba = [mean_negative, mean_neutral, mean_positive]

# cari urutan yg paling max nilainya
max_value = max(urutan_proba)
max_index = urutan_proba.index(max_value)

# tampilkan hasil
print("probabilitas: ", urutan_proba)
print("hasil sentiment akhir: ", urutan_kelas[max_index])

probabilitas: [0.19154535474846188, 0.025903564503779345, 0.782551080747759]
hasil sentiment akhir: 1

```

```
In [1]: # import csv
import csv

# import random data
from sklearn.utils import shuffle
from sklearn.model_selection import ShuffleSplit

# Making the Cross Validation
from sklearn.model_selection import cross_val_score
from sklearn.model_selection import cross_validate

# import metode
from sklearn.naive_bayes import MultinomialNB #metode naive bayes
from sklearn.ensemble import RandomForestClassifier #metode random forest
#pipeline u/ SVM
from sklearn.pipeline import Pipeline
from sklearn.calibration import CalibratedClassifierCV
from sklearn.svm import LinearSVC
from sklearn.multiclass import OneVsRestClassifier
# from sklearn import svm

# import vektorisasi
from sklearn.feature_extraction.text import TfidfVectorizer
```

```
In [2]: # buka dataset
dataset = []
with open("dataset/sample.csv", 'r', encoding="utf8") as csvfile:
    r = csv.reader(csvfile, delimiter=',')
    next(r)
    for row in r:
        dataset.append([row[0], row[1]])
```

```
In [3]: c = shuffle(dataset, random_state=0)
print(c)
```

```
[['pas awal dapat unit yang masuk fastboot', '0'], ['ditarik juga aman2 ajaa nggak kaku dan lancar2', '1'], ['terima kasih gan sukses selalu', '1'], ['wifi suka eror', '-1'], ['kecptan pengiriman', '0'], ['trusted 🍀 🍀', '1'], ['nggak confirm sebelum nya 🍀', '-1'], ['aman smuaa', '0'], ['saya pake bluetooth orico lima puluh original tetap terasa delay', '-1'], ['paket sudah sampai cuma belum dibuka', '-1'], ['ramah dan pengiriman nya super cepat', '1'], ['habis bongkar nggak benar ngasih lem', '-1'], ['bukannya diperbaiki', '-1'], ['pengiriman standar empat hari', '0'], ['original ibox 🍀', '1'], ['mantap sesuai deskripsi nya', '1'], ['kondisi sempurna', '1'], ['ternyata pipa itu mah bukan selang', '0'], ['lagi proses bikin telor puyuhny', '1'], ['produk sudah sampai dan sesuai dengan keterangan pada etalase tokopedia', '1'], ['dikasih bh seratus dan tiga utools ijo semua', '1'], ['di sarankan tidak membeli', '-1'], ['barang yang di pesan juga sesuai', '1'], ['mungkin stoknya kosong', '0'], ['pengiriman satu daerah saja hampir dua minggu', '0'], ['akan selalu langganan beli hape di toko ini', '1'], ['a pada admin', '0'], ['apa bedanya dengan penipu', '-1'], ['carbon active dan filter tanki bawah', '0'], ['barang sampai tetapi kok wlatnya gede ya sampai enam ratus di desknya cuma empat ratus lima puluh', '0'], ['screen scratch', '-1'], ['benar - benari dikasih yang terbaik', '1'], ['pengiriman cepat banget dan product ok', '1'], ['produk barang bagus dan sesuai deskripsi', '1'], ['yang waswas ingin iphone', '1'], ['dapat bh sembilan puluh sembilan lagi 🍀 🍀 🍀', '1'], ['blur atau tidak bisa auto capture foto', '-1'], ['sangat lambat', 'Hati-hati kalau mau beli barangnya', '0'], ['warna paling nggak lak u yang dikirim', '-1'], ['juga rugi waktu belanja di toko ini', '-1'], ['tetapi proses pengiriman lama', '0'], ['namanya teko bunyi tetapi pas masak air nggak bunyi pertama tamanya doang bunyi', '0'], ['packing dengan bubble wrap 🍀', '0'], ['skor ijo semua', '0'], ['i ring sama screenprotectornya nggak ada', '-1'], ['sesuai deskripsi dan harganya juga terjangkau', '1'], ['respons kurang pengiriman lambat pdahal pesan dari pagi dan pake go send baru kali ini di kirim esok hari', '-1'], ['ngalirnya lama', '-1'], ['keep up the good work kak', '1'], ['ini sudah ke enam kali beli hp disini', '1'], ['ini sudah
```

```
In [4]: # pisahkan fitur dan label
data_train = []
label_train = []
for row in c:
    data_train.append(row[0])
    label_train.append(row[1])
print(data_train)
print(label_train)
```

```
[['pas awal dapat unit yang masuk fastboot', 'ditarik juga aman2 ajaa nggak kaku dan lancar2', 'terima kasih gan sukses selalu', 'wifi suka eror', 'kecptan pengiriman', 'trusted 🍀 🍀', 'nggak confirm sebelum nya 🍀', 'aman smuaa', 'saya pake bluetooth orico lima puluh original tetap terasa delay', 'paket sudah sampai cuma belum dibuka', 'ramah dan pengiriman nya super cepat', 'habis bongkar nggak benar ngasih lem', 'bukannya diperbaiki', 'pengiriman standar empat hari', 'original ibox 🍀', 'mantap sesuai deskripsi nya', 'kondisi sempurna', 'ternyata pipa itu mah bukan selang', 'lagi proses bikin telor puyuhny', 'produk sudah sampai dan sesuai dengan keterangan pada etalase tokopedia', 'dikasih bh seratus dan tiga utools ijo semua', 'di sarankan tidak membeli', 'barang yang di pesan juga sesuai', 'mungkin stoknya kosong', 'pengiriman satu daerah saja hampir dua minggu', 'akan selalu langganan beli hape di toko ini', 'a pada admin', 'apa bedanya dengan penipu', 'carbon active dan filter tanki bawah', 'barang sampai tetapi kok wlatnya gede ya sampai enam ratus di desknya cuma empat ratus lima puluh', 'screen scratch', 'benar - benari dikasih yang terbaik', 'pengiriman cepat banget dan product ok', 'produk barang bagus dan sesuai deskripsi', 'yang waswas ingin iphone', 'dapat bh sembilan puluh sembilan lagi 🍀 🍀 🍀', 'blur atau tidak bisa auto fokus saat mau capture foto', 'sering belanja disini belum sama sekali mengecewakan', 'warna paling nggak laku yang dikirim', 'juga rugi waktu belanja di toko ini', 'tetapi proses pengiriman lama', 'namanya teko bunyi tetapi pas masak air nggak bunyi pertama tamanya doang bunyi', 'packing dengan bubble wrap 🍀', 'skor ijo semua', 'i ring sama screenprotectornya nggak ada', 'sesuai deskripsi dan harganya juga terjangkau', 'respons kurang pengiriman lambat pdahal pesan dari pagi dan pake go send baru kali ini pesan hari ini di kirim esok hari', 'ngalirnya lama', 'keep up the good work kak', 'ini sudah ke enam kali beli hp disini', 'ini sudah sore belum dikirim2 juga', 'dikira hitam glossy ternyata hitam doff', 'harus sering dibersihkan atau diganti filternya', 'pesanan dua belas pcs minta warna kotak empat warna di kirim sewarna pink semua dan sendok garpu tipis ada motif bukannya tebal polos', 'barang utuh dan berfungsi dengan baik', 'penjual salahinnya ke pihak kurir', 'tetapi packing nya nggak rap
```

```
In [5]: # vektoran kata2 dan bikin Label X, y
vectorizer = TfidfVectorizer(min_df=1)
X = vectorizer.fit_transform(data_train)
y = label_train

print(X)
```

```
(0, 640) 0.499786900412221
(0, 1199) 0.42768663692965125
(0, 2263) 0.20594605468906588
(0, 2195) 0.38219643545815635
(0, 410) 0.31009617197558653
(0, 86) 0.42768663692965125
```

```
(0, 1480) 0.3158065970924757
(1, 1099) 0.4205307924981186
(1, 407) 0.1681954393131743
(1, 895) 0.4205307924981186
(1, 1370) 0.17487866736819618
(1, 24) 0.4205307924981186
(1, 49) 0.4205307924981186
(1, 877) 0.23853093866057196
(1, 567) 0.4205307924981186
(2, 1841) 0.49279911026589873
(2, 1996) 0.49279911026589873
(2, 676) 0.43254589170782975
(2, 922) 0.39844793242220916
(2, 2088) 0.41041609441811905
(3, 624) 0.6511335389718353
(3, 1994) 0.49793405428799975
(3, 2246) 0.5727806102274804
(4, 1533) 0.4704532010835067
(4, 942) 0.8824249461513889
:
:
:
(1488, 1835) 0.40514286160070894
(1488, 2051) 0.3835719147474365
(1488, 2150) 0.35639573567485144
(1488, 1009) 0.33849524413733184
(1488, 1116) 0.2912902696858784
(1488, 756) 0.26107323436460417
(1488, 251) 0.45385257219183094
(1488, 2128) 0.2444547446422938
(1488, 1370) 0.16847956201016012
(1489, 294) 0.5362910775595401
(1489, 715) 0.654543449494369
(1489, 2145) 0.5328834327074163
(1490, 1201) 0.364924439920135
(1490, 405) 0.2750541833872989
(1490, 847) 0.23972506860152853
(1490, 913) 0.670045658601237
(1490, 1101) 0.2750541833872989
(1490, 794) 0.23972506860152853
(1490, 929) 0.25210525215491225
(1490, 1452) 0.23851311599115368
(1490, 441) 0.17350753929217058
(1491, 1666) 0.5751076946879814
(1491, 1340) 0.5497196826649268
(1491, 1094) 0.4885492774192129
(1491, 1533) 0.35830017238987305
```

```
In [6]: # uji akurasi Naive Bayes
model_NB = MultinomialNB()
# cv = ShuffleSplit(n_splits=5)
# scores_cv_NB = cross_val_score(klasifikasi_NB, X, y, cv=cv)
scores_cv_NB = cross_val_score(model_NB, X, y, cv=10, scoring='accuracy')

print("Cross Validation Naive Bayes")
print("1. Hasil tiap fold NB : ", scores_cv_NB)
print("2. Mean/rerata akurasi NB : ", scores_cv_NB.mean())
print("3. Max akurasi NB : ", scores_cv_NB.max())

Cross Validation Naive Bayes
1. Hasil tiap fold NB : [0.6          0.6          0.57718121 0.61744966 0.70469799 0.66442953
 0.65771812 0.58389262 0.63758389 0.65100671]
2. Mean/rerata akurasi NB : 0.6293959731543625
3. Max akurasi NB : 0.7046979865771812
```

```
In [7]: # uji akurasi random forest
model_RF = RandomForestClassifier()
# cv = ShuffleSplit(n_splits=5)
# scores_cv_RF = cross_val_score(klasifikasi_RF, X, y, cv=cv)
scores_cv_RF = cross_val_score(model_RF, X, y, cv=10, scoring='accuracy')

print("Cross Validation Random Forest")
print("1. Hasil tiap fold RF : ", scores_cv_RF)
print("2. Mean/rerata akurasi RF : ", scores_cv_RF.mean())
print("3. Max akurasi RF : ", scores_cv_RF.max())

Cross Validation Random Forest
1. Hasil tiap fold RF : [0.64          0.60666667 0.65771812 0.6442953 0.69127517 0.68456376
 0.65100671 0.60402685 0.61073826 0.60402685]
2. Mean/rerata akurasi RF : 0.6394317673378077
3. Max akurasi RF : 0.6912751677852349
```

```
In [8]: # uji akurasi support vector machine
pipeline_SVM = Pipeline([('clf', OneVsRestClassifier(CalibratedClassifierCV(LinearSVC(class_weight="balanced")))))
model_SVM = pipeline_SVM.fit(X,y) #training
# klasifikasi_SVM = svm.SVC(kernel='linear', C=1)
# cv = ShuffleSplit(n_splits=5)
# scores_cv_SVM = cross_val_score(klasifikasi_SVM, X, y, cv=cv)
scores_cv_SVM = cross_val_score(model_SVM, X, y, cv=10, scoring='accuracy')

print("Cross Validation Support Vector Machine")
print("1. Hasil tiap fold SVM : ", scores_cv_SVM)
print("2. Mean/rerata akurasi SVM : ", scores_cv_SVM.mean())
print("3. Max akurasi SVM : ", scores_cv_SVM.max())

Cross Validation Support Vector Machine
1. Hasil tiap fold SVM : [0.62          0.63333333 0.63087248 0.59731544 0.69127517 0.69798658
 0.63758389 0.59060403 0.61744966 0.65100671]
2. Mean/rerata akurasi SVM : 0.6367427293064877
3. Max akurasi SVM : 0.697986577181208
```

Jupyter 2-uji-prediksi Last Checkpoint: 04/04/2023 (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3 (ipykernel) O

```

In [12]: # import csv
import csv

# import random data
from sklearn.utils import shuffle

# Making the Cross Validation
from sklearn.model_selection import cross_val_score
from sklearn.model_selection import cross_validate

# import metode
from sklearn.naive_bayes import MultinomialNB #metode naive bayes
from sklearn.ensemble import RandomForestClassifier #metode random forest
# pipeline u/ SVM
from sklearn.pipeline import Pipeline
from sklearn.calibration import CalibratedClassifierCV
from sklearn.svm import LinearSVC
from sklearn.multiclass import OneVsRestClassifier
from sklearn import svm

# import vektorisasi
from sklearn.feature_extraction.text import TfidfVectorizer

```

```

In [13]: # buka dataset
dataset = []
with open('dataset/sample.csv', 'r', encoding="utf8") as csvfile:
    r = csv.reader(csvfile, delimiter=',')
    next(r)
    for row in r:
        dataset.append([row[0],row[1]])

```

```

In [14]: c = shuffle(dataset, random_state=0)
print(c)

```

```

[['pas awal dapat unit yang masuk fastboot', '0'], ['ditarik juga aman2 ajaa nggak kaku dan lancar2', '1'], ['terima kasih gan sukses selalu', '1'], ['wifi suka eror', '-1'], ['kecepatan pengiriman', '0'], ['trusted', 'E', '1'], ['nggak confirm sebelum nya', '-1'], ['aman smuaa', '0'], ['saya pake bluetooth orico lima puluh original tetap terasa delay', '-1'], ['paket sudah sampai cuma belum dibuka', '-1'], ['ramah dan pengiriman nya super cepat', '1'], ['habis bongkar nggak benar ngasih lem', '-1'], ['bukannya diperbaiki', '-1'], ['pengiriman standar empat hari', '0'], ['original ibox', 'E', '1'], ['mantap sesuai deskripsi nya', '1'], ['kondisi sempurna', '1'], ['ternyata pipa itu mah bukan selang', '0'], ['lagi proses bikin telor puyuhny', '1'], ['produk sudah sampai dan sesuai dengan keterangan pada etalase tokopedia', '1'], ['dikasih bh seratus dan tiga utools ijo semua', '1'], ['di sarankan tidak membeli', '-1'], ['barang yang di pesan juga sesuai', '1'], ['mungkin stoknya kosong', '0'], ['pengiriman satu daerah saja hampir dua minggu', '0'], ['akan selalu langganan beli hape di toko ini', '1'], ['a pada admin', '0'], ['apa bedanya dengan penipu', '-1'], ['carbon active dan filter tanki bawah', '0'], ['barang sampai tetapi kok wattnya gede ya sampai enam ratus di desknya cuma empat ratus lima puluh', '0'], ['screen scratch', '-1'], ['benar - benari dikasih yang terbaik', '1'], ['pengiriman cepat banget dan product ok', '1'], ['produk barang bagus dan sesuai deskripsi', '1'], ['yang waswas ingin iphone', '1'], ['dapat bh sembilan puluh sembilan lagi', 'E', 'E', '1'], ['blur atau tidak bisa auto fokus saat mau capture foto', '-1'], ['sering belanja disini belum sama sekali mengecewakan', '-1'], ['warna paling enak lak u yang dikirim', '-1'], ['juga rugi waktu belanja di toko ini', '-1'], ['tetapi proses pengiriman lama', '0'], ['namanya teko bunyi tetapi pas masak air nggak bunyi pertama tamanya doang bunyi', '0'], ['packing dengan bubble wrap', 'E', '0'], ['skor ijo semua', '0'], ['i ring sama screenprotectornya nggak ada', '-1'], ['sesuai deskripsi dan harganya juga terjangkau', '1'], ['respons kurang pengiriman lambat pdahal pesan dari pagi dan pake go send baru kali ini di kirim esok hari', '-1'], ['ngalirnya lama', '-1'], ['keep up the good work kak', '1'], ['ini sudah ke enam kali beli hp disini', '1'], ['ini sudah

```

```

In [15]: # pisahkan fitur dan label
data_train = []
label_train = []
for row in c:
    data_train.append(row[0])
    label_train.append(row[1])
print(data_train)
print(label_train)

```

```

[['pas awal dapat unit yang masuk fastboot', 'ditarik juga aman2 ajaa nggak kaku dan lancar2', 'terima kasih gan sukses selalu u', 'wifi suka eror', 'kecepatan pengiriman', 'trusted', 'aman smuaa', 'saya pake bluetooth orico lima puluh original tetap terasa delay', 'paket sudah sampai cuma belum dibuka', 'ramah dan pengiriman nya super cepat', 'habis bongkar nggak benar ngasih lem', 'bukannya diperbaiki', 'pengiriman standar empat hari', 'original ibox', 'mantap sesuai deskripsi nya', 'kondisi sempurna', 'ternyata pipa itu mah bukan selang', 'lagi proses bikin telor puyuhny', 'produk sudah sampai dan sesuai dengan keterangan pada etalase tokopedia', 'dikasih bh seratus dan tiga utools ijo semua', 'di sarankan tidak membeli', 'barang yang di pesan juga sesuai', 'mungkin stoknya kosong', 'pengiriman satu daerah saja hampir dua minggu', 'akan selalu langganan beli hape di toko ini', 'a pada admin', 'apa bedanya dengan penipu', 'carbon active dan filter tanki bawah', 'barang sampai tetapi kok wattnya gede ya sampai enam ratus di desknya cuma empat ratus lima puluh', 'screen scratch', 'benar - benari dikasih yang terbaik', 'pengiriman cepat banget dan product ok', 'produk barang bagus dan sesuai deskripsi', 'yang waswas ingin iphone', 'dapat bh sembilan puluh sembilan lagi', 'blur atau tidak bisa auto fokus saat mau capture foto', 'sering belanja disini belum sama sekali mengecewakan', 'warna paling enak lak u yang dikirim', 'juga rugi waktu belanja di toko ini', 'tetapi proses pengiriman lama', 'namanya teko bunyi tetapi pas masak air nggak bunyi pertama tamanya doang bunyi', 'packing dengan bubble wrap', 'skor ijo semua', 'i ring sama screenprotectornya nggak ada', 'sesuai deskripsi dan harganya juga terjangkau', 'respons kurang pengiriman lambat pdahal pesan dari pagi dan pake go send baru kali ini pesan hari ini di kirim esok hari', 'ngalirnya lama', 'keep up the good work kak', 'ini sudah ke enam kali beli hp disini', 'ini sudah sore belum dikirim2 juga', 'dikira hitam glossy ternyata hitam doff', 'harus sering dibersihkan atau diganti filtenya', 'pesanan dua belas pcs minta warna kotak empat warna di kirim sewarna pink semua dan sendok garpu tipis ada motif bukan tebal polos', 'barang utuh dan berfungsi dengan baik', 'penjual salahinya ke pihak kurir', 'tetapi packing nya nggak rapi

```

```

In [16]: # vektoran kata2 dan bikin label X, y
vectorizer = TfidfVectorizer(min_df=1)
X = vectorizer.fit_transform(data_train)
y = label_train

from collections import Counter

print(*Counter(y))

0 1 -1

```

```

In [17]: # fungsi untuk jabarkan probabilitas tiap Label

```

```
def probPredict(varproba):
    alist = []
    for seq in varproba:
        for val in seq:
            alist.append(val)
    return alist
```

```
In [18]: # kalimat test (Langsung vektorkan)
kalimat_test = "kurang bagus"
text_test = vectorizer.transform([kalimat_test])
```

```
In [19]: # training NB
klasifikasi_NB = MultinomialNB()
model_NB = klasifikasi_NB.fit(X,y) #training
# prediksi menggunakan model yg telah di training
prediksi_NB = model_NB.predict(text_test)
val_proba_NB = probPredict(model_NB.predict_proba(text_test))

# tampilkan hasil
print('Naive Bayes')
print('1. data text test :', kalimat_test)
print('2. hasil vektor :', text_test) #hasil vektor
print('3. urutan kelas/label :', model_NB.classes_)
print('4. probabilitas tiap kelas/label :', val_proba_NB) #probabilitas tiap Label -1,0,1
print('5. hasil akhir prediksi :', prediksi_NB) #hasil klasifikasi Label

Naive Bayes
1. data text test : kurang bagus
2. hasil vektor : (0, 1071) 0.7071067811865475
(0, 103) 0.7071067811865475
3. urutan kelas/label : ['-1' '0' '1']
4. probabilitas tiap kelas/label : [0.5259982360543245, 0.21359515353963177, 0.260406610406604296]
5. hasil akhir prediksi : ['-1']
```

```
In [20]: # training random forest
klasifikasi_RF = RandomForestClassifier()
model_RF = klasifikasi_RF.fit(X,y) #training
# prediksi menggunakan model yg telah di training
prediksi_RF = model_RF.predict(text_test)
val_proba_RF = probPredict(model_RF.predict_proba(text_test))

# tampilkan hasil
print('Random Forest')
print('1. data text test :', kalimat_test)
print('2. hasil vektor :', text_test) #hasil vektor
print('3. urutan kelas/label :', model_RF.classes_)
print('4. probabilitas tiap kelas/label :', val_proba_RF) #probabilitas tiap Label -1,0,1
print('5. hasil akhir prediksi :', prediksi_RF) #hasil klasifikasi Label
```

```
Random Forest
1. data text test : kurang bagus
2. hasil vektor : (0, 103) 0.7071067811865475
(0, 1071) 0.7071067811865475
3. urutan kelas/label : ['-1' '0' '1']
4. probabilitas tiap kelas/label : [0.46, 0.12, 0.42]
5. hasil akhir prediksi : ['-1']
```

```
In [21]: # training SVM
pipeline_SVM = Pipeline(['clf', OneVsRestClassifier(CalibratedClassifierCV(LinearSVC(class_weight="balanced")))]))
model_SVM = pipeline_SVM.fit(X,y) #training
# prediksi menggunakan model yg telah di training
prediksi_SVM = model_SVM.predict(text_test)
val_proba_SVM = probPredict(model_SVM.predict_proba(text_test))

# tampilkan hasil
print('Support Vector Machine')
print('1. data text test :', kalimat_test)
print('2. hasil vektor :', text_test) #hasil vektor
print('3. urutan kelas/label :', model_SVM.classes_)
print('4. probabilitas tiap kelas/label :', val_proba_SVM) #probabilitas tiap Label -1,0,1
print('5. hasil akhir prediksi :', prediksi_SVM) #hasil klasifikasi Label
```

```
Support Vector Machine
1. data text test : kurang bagus
2. hasil vektor : (0, 103) 0.7071067811865475
(0, 1071) 0.7071067811865475
3. urutan kelas/label : ['-1' '0' '1']
4. probabilitas tiap kelas/label : [0.8107934473305507, 0.0740586447396042, 0.1151479079298451]
5. hasil akhir prediksi : ['-1']
```



```

In [11]: from statistics import mode

# fungsi hitung rerata/mean probabilitas
def mean(lst):
    return sum(lst) / len(lst)

# SS = sentiment selector
# SV = soft voting
i = 0
nb_result = []
rf_result = []
svm_result = []
voting_data_SS = []
voting_result_SS = []
voting_data_SV = []
voting_result_SV = []
for item in y_test:
    # prediksi NB
    prediksi_NB = model_NB.predict(X_test[i])
    val_proba_NB = probPredict(model_NB.predict_proba(X_test[i]))
    nb_result.append(prediksi_NB[0])

    # prediksi RF
    prediksi_RF = model_RF.predict(X_test[i])
    val_proba_RF = probPredict(model_RF.predict_proba(X_test[i]))
    rf_result.append(prediksi_RF[0])

    # prediksi SVM
    prediksi_SVM = model_SVM.predict(X_test[i])
    val_proba_SVM = probPredict(model_SVM.predict_proba(X_test[i]))
    svm_result.append(prediksi_SVM[0])

    # S: bagian sentiment selector
    # gabung tiap hasil prediksi metode
    all_hasil_prediksi = (prediksi_NB[0], prediksi_RF[0], prediksi_SVM[0])
    voting_data_SS.append(all_hasil_prediksi)
    # lakukan sentiment voting
    hasil_akhir = mode(all_hasil_prediksi)
    # simpan label dan hasil prediksi
    voting_result_SS.append(hasil_akhir)
    # E: bagian sentiment selector

    # S: bagian soft voting
    # gabung tiap hasil probabilitas (urutan -1,0,1)
    proba_negative = [val_proba_NB[0], val_proba_RF[0], val_proba_SVM[0]]
    proba_neutral = [val_proba_NB[1], val_proba_RF[1], val_proba_SVM[1]]
    proba_positive = [val_proba_NB[2], val_proba_RF[2], val_proba_SVM[2]]
    # hitung rerata/mean probabilitas tiap kelas
    mean_negative = mean(proba_negative)
    mean_neutral = mean(proba_neutral)
    mean_positive = mean(proba_positive)
    # pilih kelas dengan proba tertinggi
    urutan_kelas = ['-1', '0', '1']
    urutan_proba = [mean_negative, mean_neutral, mean_positive]
    # simpan hasil proba
    voting_data_SV.append(urutan_proba)
    # cari urutan yg paling max nilainya
    max_value = max(urutan_proba)
    max_index = urutan_proba.index(max_value)
    # simpan hasil soft voting
    voting_result_SV.append(urutan_kelas[max_index])
    # E: bagian soft voting

    # pindah data test berikutnya
    i += 1

print('Naive Bayes')
print('probabilitas tiap kelas/label :', val_proba_NB) #probabilitas tiap label -1,0,1
print('hasil akhir prediksi :', prediksi_NB) #hasil klasifikasi label
print('Random Forest')
print('probabilitas tiap kelas/label :', val_proba_RF) #probabilitas tiap label -1,0,1
print('hasil akhir prediksi :', prediksi_RF) #hasil klasifikasi label
print('Support Vector Machine')
print('probabilitas tiap kelas/label :', val_proba_SVM) #probabilitas tiap label -1,0,1
print('hasil akhir prediksi :', prediksi_SVM) #hasil klasifikasi label

print('Sentiment Selector')
print('hasil tiap algoritma :', all_hasil_prediksi)
print('hasil akhir :', hasil_akhir)
print('Soft Voting')
print("probabilitas negative NB, RF, SVM:", proba_negative)
print("probabilitas neutral NB, RF, SVM:", proba_neutral)
print("probabilitas positive NB, RF, SVM:", proba_positive)
print("mean negative: ", mean_negative)
print("mean neutral: ", mean_neutral)
print("mean positive: ", mean_positive)
print("probabilitas: ", urutan_proba)
print("hasil sentiment akhir: ", urutan_kelas[max_index])
print('actual label: ', y_test)
print('\nvoting data SS: ', voting_data_SS)
print('\nvoting result SS: ', voting_result_SS)
print('\nvoting data SV: ', voting_data_SV)
print('\nvoting result SV: ', voting_result_SV)

Naive Bayes
probabilitas tiap kelas/label : [0.3580184891972121, 0.47717682295098757, 0.16480468785180002]
hasil akhir prediksi : ['0']
Random Forest
probabilitas tiap kelas/label : [0.24, 0.74, 0.02]

```



```

In [12]: # hitung akurasi manual model sentiment voting
# hasil_akurasi_SS = sum(1 for x,y in zip(y_test,voting_result_SS) if x == y) / float(len(y_test))
# hasil_akurasi_SV = sum(1 for x,y in zip(y_test,voting_result_SV) if x == y) / float(len(y_test))
# print('akurasi sentiment selector: ', hasil_akurasi_SS)
# print('akurasi soft voting: ', hasil_akurasi_SV)

# hitung hasil confusion matrix lib
from sklearn import metrics

# contoh perhitungan:
# Predicted 0 1
# Actual
# 0          6  4
# 1          2  8
# Accuracy: (6+8) / (6+4+2+8) = 0.7
# Precision: 8 / (8+4) = 0.667
# Recall: 8 / (2+8) = 0.8

print('Naive Bayes')
print('Confusion matrix NB saja \n', metrics.confusion_matrix(y_test, nb_result))
print('accuracy model NB saja:', metrics.accuracy_score(y_test, nb_result))
print('precision model NB saja:', metrics.precision_score(y_test, nb_result, average='macro'))
print('recall model NB saja:', metrics.recall_score(y_test, nb_result, average='macro'))
print('f1-score model NB saja:', metrics.f1_score(y_test, nb_result, average='macro'))

print('Random Forest')
print('Confusion matrix RF saja \n', metrics.confusion_matrix(y_test, rf_result))
print('accuracy model RF saja:', metrics.accuracy_score(y_test, rf_result))
print('precision model RF saja:', metrics.precision_score(y_test, rf_result, average='macro'))
print('recall model RF saja:', metrics.recall_score(y_test, rf_result, average='macro'))
print('f1-score model RF saja:', metrics.f1_score(y_test, rf_result, average='macro'))

print('Support Vector Machine')
print('Confusion matrix SVM saja \n', metrics.confusion_matrix(y_test, svm_result))
print('accuracy model SVM saja:', metrics.accuracy_score(y_test, svm_result))
print('precision model SVM saja:', metrics.precision_score(y_test, svm_result, average='macro'))
print('recall model SVM saja:', metrics.recall_score(y_test, svm_result, average='macro'))
print('f1-score model SVM saja:', metrics.f1_score(y_test, svm_result, average='macro'))

print('Sentiment selector (SS)')
print('Confusion matrix SS: \n', metrics.confusion_matrix(y_test, voting_result_SS))
print('Akurasi SS: ', metrics.accuracy_score(y_test, voting_result_SS))
print('Precision SS: ', metrics.precision_score(y_test, voting_result_SS, average='macro'))
print('Recall SS: ', metrics.recall_score(y_test, voting_result_SS, average='macro'))
print('F1-score SS: ', metrics.f1_score(y_test, voting_result_SS, average='macro'))

print('Soft Voting (SV)')
print('Confusion matrix SV: \n', metrics.confusion_matrix(y_test, voting_result_SV))
print('Akurasi SV: ', metrics.accuracy_score(y_test, voting_result_SV))

Naive Bayes
Confusion matrix NB saja
[[ 89 35 18]
 [ 62 67 26]
 [ 18 15 118]]
accuracy model NB saja: 0.6116071428571429
precision model NB saja: 0.6092239511042931
recall model NB saja: 0.61349186085129317
f1-score model NB saja: 0.6063293119760669
-----
Random Forest
Confusion matrix RF saja
[[ 83 46 13]
 [ 32 102 21]
 [ 13 38 100]]
accuracy model RF saja: 0.6361607142857143
precision model RF saja: 0.6476977511635371
recall model RF saja: 0.6349410713372309
f1-score model RF saja: 0.6382698899959794
-----
Support Vector Machine
Confusion matrix SVM saja
[[ 88 39 15]
 [ 48 82 25]
 [ 16 24 111]]
accuracy model SVM saja: 0.6272321428571429
precision model SVM saja: 0.6265213158495692
recall model SVM saja: 0.6279499685573384
f1-score model SVM saja: 0.6268018200657747
Sentiment selector (SS)
Confusion matrix SS:
[[ 93 38 11]
 [ 50 82 23]
 [ 19 23 109]]
Akurasi SS: 0.6339285714285714
Precision SS: 0.6365794699128032
Recall SS: 0.6352720467216888
F1-score SS: 0.6345580914574708

Soft Voting (SV)
Confusion matrix SV:
[[ 90 41 11]
 [ 43 90 22]
 [ 15 25 111]]
Akurasi SV: 0.6495535714285714
Precision SV: 0.6510548394548395
Recall SV: 0.6498491053133585
F1-score SV: 0.6506700543673337

```

```
In [6]: # split dataset 30/70 (tidak perlu random lg, karna sudah dilakukan diatas)
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3)
print(X_train, y_train)
```

```
(0, 1610) 0.4706969556307691
(0, 810) 0.42785449752311877
(0, 847) 0.28106489330407525
(0, 274) 0.33842245819557437
(0, 1785) 0.30982834512039887
(0, 123) 0.2594520478064838
(0, 1370) 0.39148075624138334
(0, 410) 0.2920471184444331
(1, 1418) 1.0
(2, 1576) 0.6677514913040581
(2, 1891) 0.3229916702981135
(2, 1768) 0.3611622796655271
(2, 1989) 0.31323891581868785
(2, 676) 0.47034829237787656
(3, 463) 0.6323078451572941
(3, 405) 0.47658884682315317
(3, 1744) 0.4852948872039865
(3, 1533) 0.37086214755618635
(4, 1192) 0.4283009942779617
(4, 1864) 0.4957409229332678
(4, 711) 0.4957409229332678
(4, 1782) 0.3110858017632518
(4, 764) 0.3308662664893856
(4, 1460) 0.3446639966688467
(5, 245) 0.6942054507281734
:
:
(1043, 1997) 0.24424431398547333
(1043, 839) 0.2090091379975724
(1043, 1014) 0.22201339308238005
(1043, 1505) 0.23124005890066565
(1043, 520) 0.24424431398547333
(1043, 244) 0.23124005890066565
(1043, 2098) 0.23124005890066565
(1043, 1113) 0.2090091379975724
(1043, 1110) 0.2148566350684556
(1043, 249) 0.23124005890066565
(1043, 1311) 0.15633888857395942
(1043, 294) 0.3106412607462004
(1043, 1744) 0.178394793262269
(1043, 2136) 0.17377396200967143
(1043, 1071) 0.13462948224018403
(1043, 8) 0.1142936870231182
(1043, 1452) 0.14510704239879435
```