REFERENCES

Acharya, S., Chellappan, S. 2017. *Pro Tableau A Step-by-Step Guide*. Apress. India.

Annels, A. E. 1991. *Mineral Deposit Evaluation: A Practical Approach*. Chapman and Hall. London.

Azevedo, A., Santos, M. F. 2008. KDD, SEMMA and CRISP-DM: A Parallel Overview. *IADIS Multi Conference on Computer Science and Information Systems*. *June*. 182–185.

Bargawa. Waterman Sulistyana. 2018. *Perencanaan Tambang* (8th ed.). Kilau Book. Yogyakarta.

Basuki, W., Oktavia, M., Elfistoni, A. 2020. Perhitungan Kebutuhan Unit Dump Truck Berdasarkan Match Factor dan Teori Antrian pada Penambangan Batubara di PT. Kamalindo Sompurna Kecamatan Pelawan Kabupaten Sarolangun Provinsi Jambi. *Mine Magazine (MineMagz)*. 1(2).

Bhatia, P. 2019. *Data Mining and Data Warehousing Principles and Practical Techniques*. Cambridge University Press. Cambridge, United Kingdom.

Bholowalia, P., Kumar, A. 2014. EBK-Means: A Clustering Technique based on Elbow Method and K-Means in WSN. *International Journal of Computer Applications*. 105(9). 975–8887.

Bilal, M., Oyedele, L. O., Qadir, J., Munir, K., Ajayi, S. O., Akinade, O. O., Owolabi, H. A., Alaka, H. A., Pasha, M. 2016. Big Data in the Construction Industry: A Review of Present Status, Opportunities, and Future Trends. *Advanced Engineering Informatics*. 30(3). 500–521.

Bonaccorso, G. 2017. *Machine Learning Algorithms*. Packt Publishing Ltd.

BP Energy. 2020. *BP Energy Outlook 2019 Edition*. *https://www.bp.com/content/dam/bp/business-sites/en/global/corporate/pdfs/energy-economics/energy-outlook/bp-energy-outlook-2020.pdf*. Accessed September 27, 2021.

Ceder, N. 2018. *The Quick Python Book*. Manning Publications.

Chaulya, S. K., Prasad, G. M. 2016. *Sensing and Monitoring Technologies for Mines and Hazardous Areas*. Elsevier Inc. Amsterdam, Netherlands.

Chiu, C. Y., Chen, Y. F., Kuo, I. T., Ku, H. C. 2009. An Intelligent Market Segmentation System using K-Means and Particle Swarm Optimization. *Expert Systems with Applications*. 36(3). 4558–4565.

El Morr, C., Ali-Hassan, H. 2019. Anaytics in Healthcare A Practical Introduction. In *Springer Cham*.

Fraley, C., Raftery, A. E. 1998. How Many Clusters? Which Clustering Method? Answers Via Model-based Cluster Analysis. *Computer Journal*. 41(8). 586–588.

Ge, Z., Song, Z., Ding, S. X., Huang, B. 2017. Data Mining and Analytics in the Process Industry: The Role of Machine Learning. *IEEE Access*. 5. 20590–20616.

Ghandi, P., Khanna, S., Ramaswamy, S. 2016. Which Industries are the Most Digital (and Why). *Harvard Business Review*. 1. 45–48.

Han, J., Kamber, M., Pei, J. 2011. *Data Mining: Concepts and Techniques* (3rd ed.). Morgan Kaufmann. San Fransisco.

Jezard, A. 2017. *Fossil Fuels Will Still Dominate Energy in 20 Years Despite Green Power Rising*. Word Economic Forum. *https://www.weforum.org/agenda/2017/10/fossil-fuel*. Accessed October 26, 2022.

Kaplinski, O., Koseleva, N., Ropaite, G. 2016. Big Data in Civil Engineering: a State-of-the-Art Survey. *Engineering Structures and Technologies*. 8(4). 165–175.

Kaufman, L., Rousseeuw, P. J. 1990. Finding Groups in Data: An Introduction to Cluster Analysis. In *Journal of Physics A: Mathematical and Theoretical* (Vol. 44, Issue 8). Wiley Interscience. Canada.

KDnuggets. 2014. *What Main Methodology Are You Using for Your Analytics, Data Mining, or Data Science Projects? Poll*. *https://www.kdnuggets.com/polls/2014/analytics-data-mining-data-science-methodology.html*. Accessed September 23, 2021.

Kurniawan, D. 2020. *Pengenalan Machine Learning dengan Python*. PT Elex Media Komputindo. Jakarta.

Lee, S., Choi, Y. 2016. Reviews of Unmanned Aerial Vehicle (Drone) Technology Trends and Its Applications in the Mining Industry. *Geosystem Engineering*. 19(4). 197–204.

Muller, A. C., Guido, S. 2016. *Introduction to Machine Learning with Python and Scikit-Learn*. O'Reilly Media. United States of America.

Nongthombam, K. 2021. Data Analysis Using Python. *International Journal of Engineering Research and Technology (IJERT)*. 10(07). 463–468.

Pacheco, M. 2019. Green Technologies to Improve the Environment on Earth. In *IntechOpen*.

Prima, G. R. 2021. Analisa Perbandingan Nilai K Terbaik Untuk Clustering K-MEANS Menggunakan Pendekatan Elbow dan Silhoutte Pada Citra Aksara Jawa. *Thesis*. Universitas Sanata Dharma.

Provost, F., Fawcett, T. 2013. *Data Science for Bussiness What You Need to Know About Data Mining and Data-Analytics Thinking*. O'Reilly Media. United States of America.

Qi, C. chong. 2020. Big Data Management in the Mining Industry. *International Journal of Minerals, Metallurgy and Materials*. 27(2). 131–139.

Raschka, S., Patterson, J., Nolet, C. 2020. Machine Learning in Python: Main Developments and Technology Trends in Data Science, Machine Learning, and

Artificial Intelligence. *Information (Switzerland)*. 11(4). 1–44.

Revuelta, M. B. 2018. *Mineral Resources: From Explotation to Sustainable Assessment*. Springer.

Rivai, M. A., Octova, A. 2021. Analisis Optimalisasi Loss Time Alat Gali Muat untuk Mencapai Target Produksi Overburden di Pit Timur PT ALLIED INDO COAL JAYA Kota Sawahlunto, Sumatera Barat. *Jurnal Bina Tambang*. 6(4). 248–256.

Rupprecht, S. M. 2015. Owner Versus Contract Miner-A South African Update. *Journal of the Southern African Institute of Mining and Metallurgy*. 115(11). 1021–1025.

Sanchez, F., Hartlieb, P. 2020. Innovation in the Mining Industry: Technological Trends and a Case Study of the Challenges of Disruptive Innovation. *Mining, Metallurgy and Exploration*. 37(5). 1385–1399.

Schroer, C., Kruse, F., Gómez, J. M. 2021. A Systematic Literature Review on Applying CRISP-DM Process Model. *Procedia Computer Science*. 181(2019). 526–534.

Sganzerla, C., Seixas, C., Conti, A. 2016. Disruptive Innovation in Digital Mining. *Procedia Engineering*. 138. 64–71.

Sharma, S., Osei-Bryson, K. M. 2009. Framework for Formal Implementation of The Business Understanding Phase of Data Mining Projects. *Expert Systems with Applications*. 36. 4114–4124.

Sokop, R. M., Arsjad, T. T., Malingkas, G. 2018. Analisa Perhitungan Produktivitas Alat Berat Gali-Muat (Excavator) dan Alat Angkut (Dump Truck) Pada Pekerjaan Pematangan Lahan Perumahan Residence Jordan Sea. *Jurnal Tekno*. 16(70). 83–88.

Soofastaei, A. 2019. *Advanced Analytics and Artificial Intelligence Applications*. IntechOpen.

Soofastaei, A. 2021. *Data Analytics Applied to the Mining Industry*. CRC Press.

Stevens, R. 2011. *Mineral Exploration and Mining Essentials*. British Columbia Institute of Technology. Canada.

Susilo, S. F., Jamaludin, A., Purnamasari, I. 2020. Pengelompokan Desa Menggunakan K-Means Untuk Penyelenggaraan Penanggulangan Bencana Banjir. *JOINS (Journal of Information System)*. 5(2). 156–167.

Tan, P.-N., Steinbach, M., Kumar, V. 2014. *Introduction to Data Mining*. Pearson. USA.

Thinsungnoen, T., Kaoungku, N., Durongdumronchai, P., Kerdprasop, K., Kerdprasop, N. 2015. The Clustering Validity with Silhouette and Sum of Squared Errors. *Proceedings of the 3rd International Conference on Industrial Application Engineering 2015*. *January 2015*. 44–51.

US Energy Information Administration. 2022. *Coal Explained: Mining and Transportation of Coal*. *https://www.eia.gov/energyexplained/coal/mining-and-transportation.php*. Accessed February 5, 2023.

Wang, F., Franco-Penya, H. H., Kelleher, J. D., Pugh, J., Ross, R. 2017. An Analysis of the Application of Simplified Silhouette to the Evaluation of K-means Clustering Validity. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. 10358 LNAI(July). 291–305.

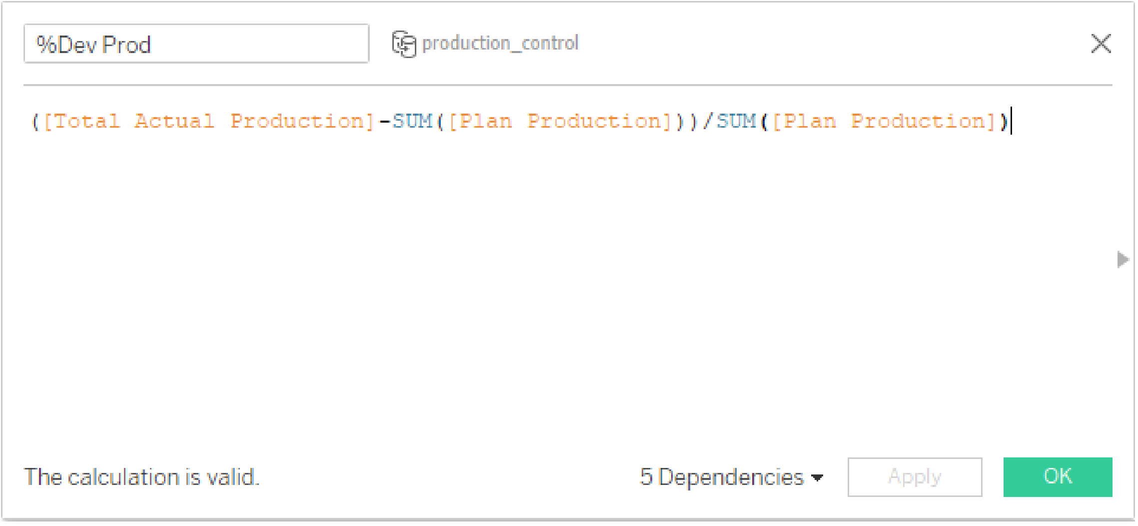
Ward, J. S., Barker, A. 2013. Undefined By Data: A Survey of Big Data Definitions. *Arxsiv Preprint*.

WorldEconomicForum. 2017. Digital Transformation Industry Mining and Metals. *Word Economic Forum*.

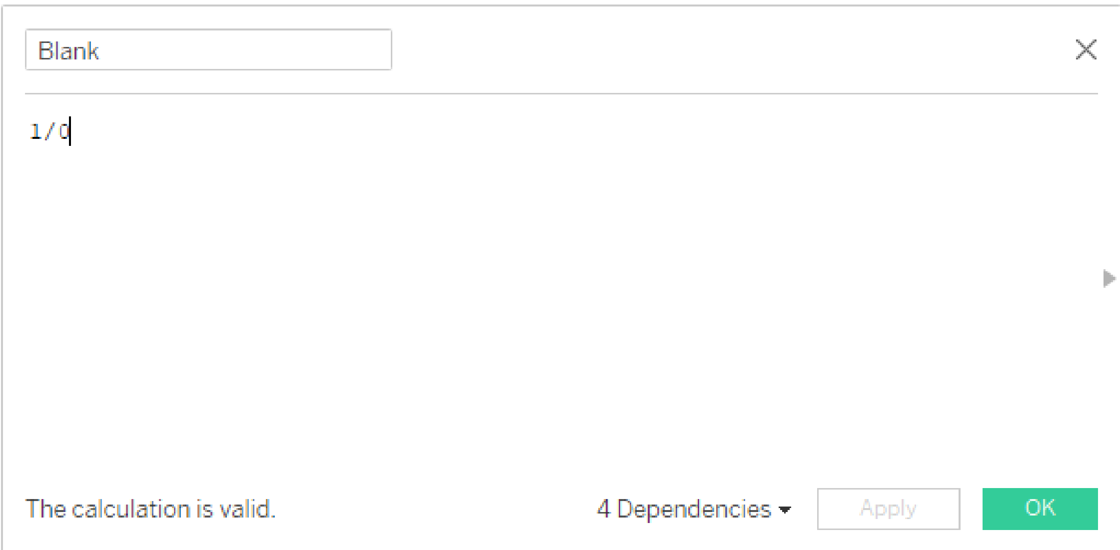
Young, A., Rogers, P. 2019. A Review of Digital Transformation in Mining. *Mining, Metallurgy and Exploration*. 36(4). 683–699.

APPENDIX A

CALCULATED FIELDS IN TABLEAU


%Dev Prod
production_control
([Total Actual Production]-SUM([Plan Production]))/SUM([Plan Production])
The calculation is valid.
5 Dependencies
Apply
OK


Actual Production
production_control
IF SUM([Hauler Production])<SUM([Loader Production])
THEN SUM([Hauler Production])
ELSE SUM([Loader Production])
END
The calculation is valid.
16 Dependencies
Apply
OK


Blank
1/0
The calculation is valid.
4 Dependencies
Apply
OK


Clustering
production_control
Results are computed along Table (across).
SCRIPT_INT("return tabpy.query('cluster_predictor',_arg1,_arg2,_arg3)['response']
[Con Dev Pa],[Con Dev Ua],[Con Dev Pdty])
Default Table Calculation
The calculation is valid.
3 Dependencies
Apply
OK

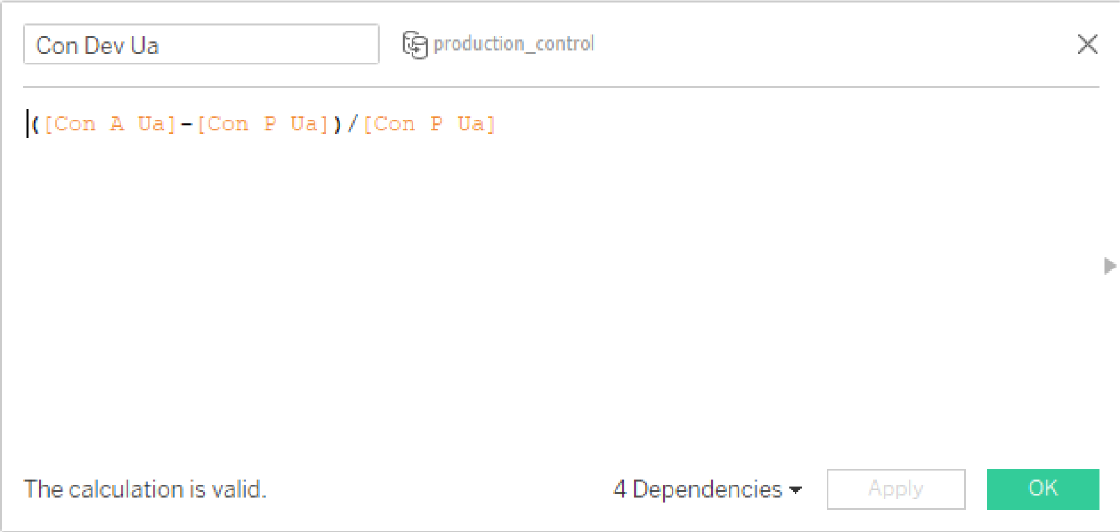

Con A Pa
production_control
SUM([A Pa]*[A Qty])/SUM([A Qty])
The calculation is valid.
10 Dependencies
Apply
OK


Con A Pdty
production_control
SUM([A Pdty]*[A Qty])/SUM([A Qty])
The calculation is valid.
7 Dependencies
Apply
OK


Con A Ua
production_control
SUM([A Ua]*[A Qty])/SUM([A Qty])
The calculation is valid.
10 Dependencies
Apply
OK


Con Dev Pa
production_control
([Con A Pa]-[Con P Pa])/[Con P Pa]
The calculation is valid.
4 Dependencies
Apply
OK

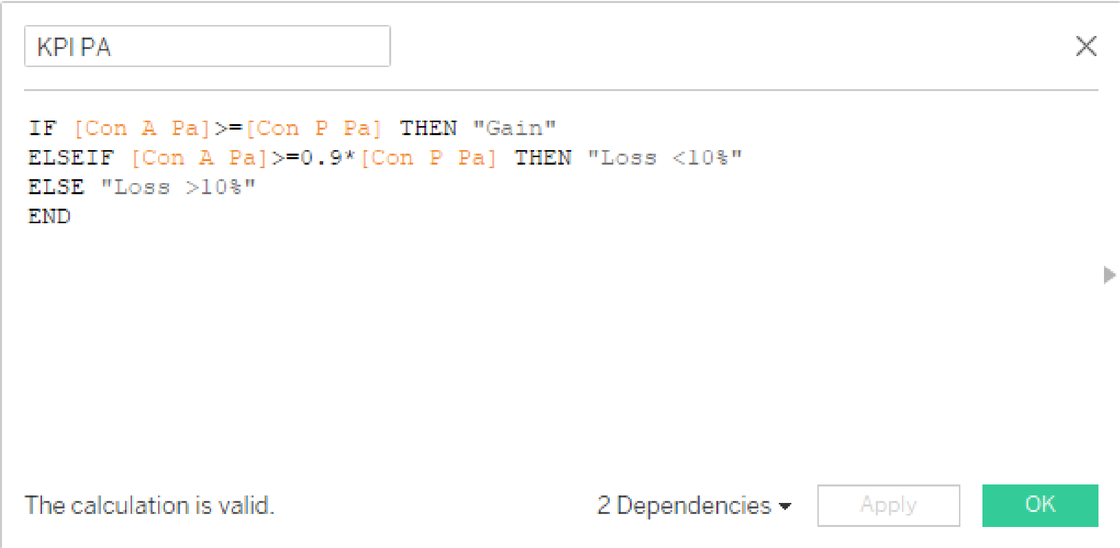

Con Dev Pdty
production_control
([Con A Pdty]-[Con P Pdty])/[Con P Pdty]
The calculation is valid.
4 Dependencies
Apply
OK


Con Dev Ua
production_control
([Con A Ua]-[Con P Ua])/[Con P Ua]
The calculation is valid.
4 Dependencies
Apply
OK


Con P Pa
production_control
SUM([P Pa]*[P Qty])/SUM([P Qty])
The calculation is valid.
10 Dependencies
Apply
OK


Con P Pdty
production_control
SUM([P Pdty]*[P Qty])/SUM([P Qty])
The calculation is valid.
7 Dependencies
Apply
OK


Con P Ua
production_control
SUM([P Ua]*[P Qty])/SUM([P Qty])
The calculation is valid.
10 Dependencies
Apply
OK


KPI PA
IF [Con A Pa]>=[Con P Pa] THEN "Gain"
ELSEIF [Con A Pa]>=0.9*[Con P Pa] THEN "Loss <10%"
ELSE "Loss >10%"
END
The calculation is valid.
2 Dependencies
Apply
OK


KPI Pdty
production_control
IF ([Con A Pdty])>=[Con P Pdty] THEN "Gain"
ELSE "Loss"
END
The calculation is valid.
2 Dependencies
Apply
OK

KPI Production — production_control

```
IF ([Actual Production])>=SUM([Plan Production]) THEN "Gain"
ELSEIF ([Actual Production])>=0.9*SUM([Plan Production]) THEN "Loss <10%"
ELSE "Loss >10%"
END
```

The calculation is valid. 6 Dependencies Apply OK

KPI UA

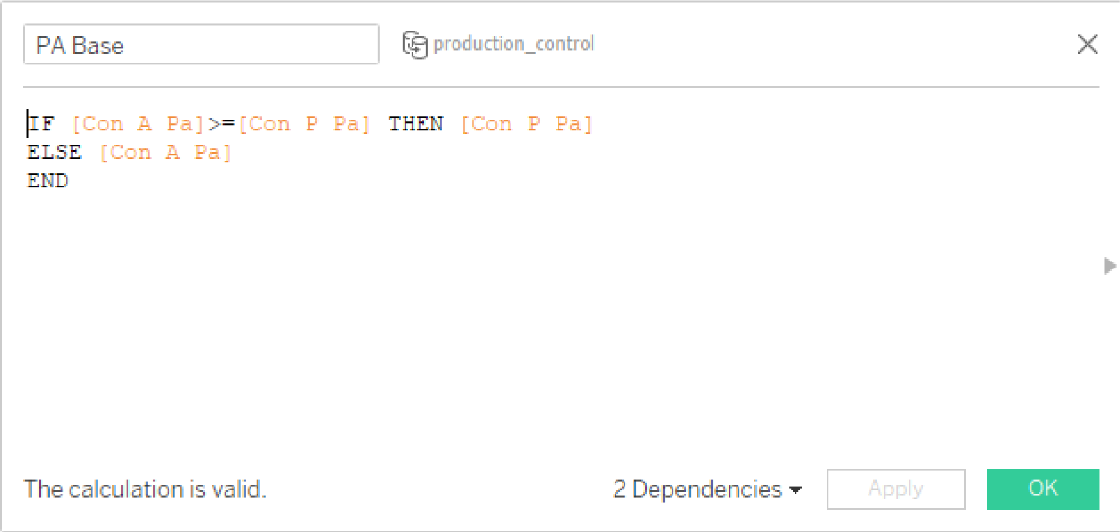
```
IF [Con A Ua]>=[Con P Ua] THEN "Gain"
ELSEIF [Con A Ua]>=0.9*[Con P Ua] THEN "Loss <10%"
ELSE "Loss >10%"
END
```

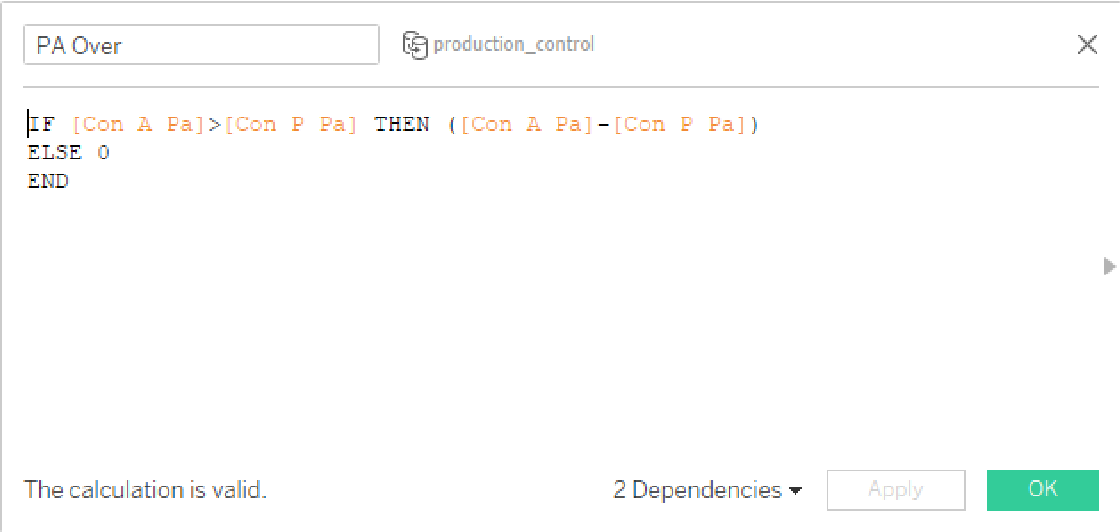
The calculation is valid. 2 Dependencies Apply OK

KPI Time Usage — production_control

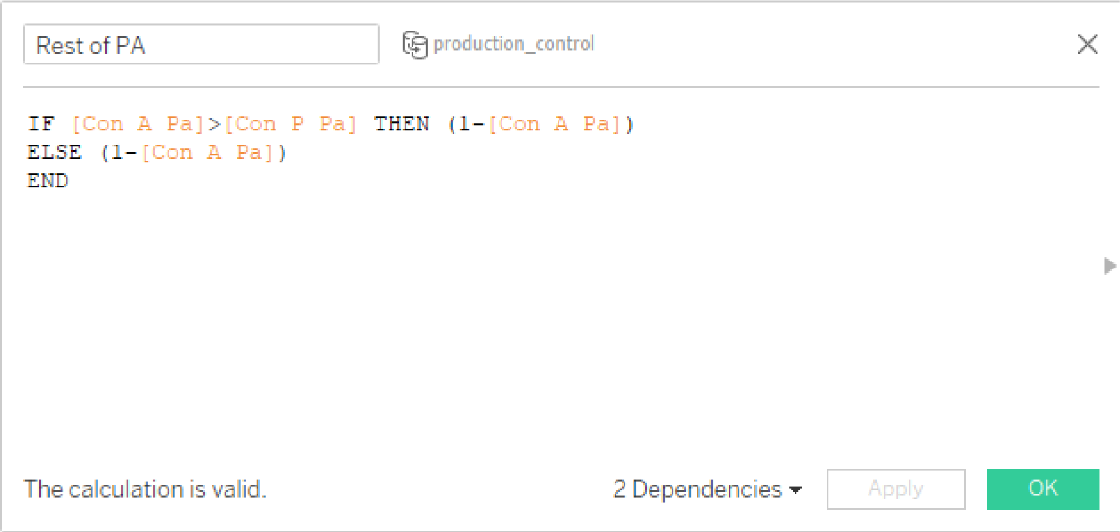
```
IF AVG([A Hr])<=AVG([P Hr]) THEN "Gain"
ELSE "Loss"
END
```

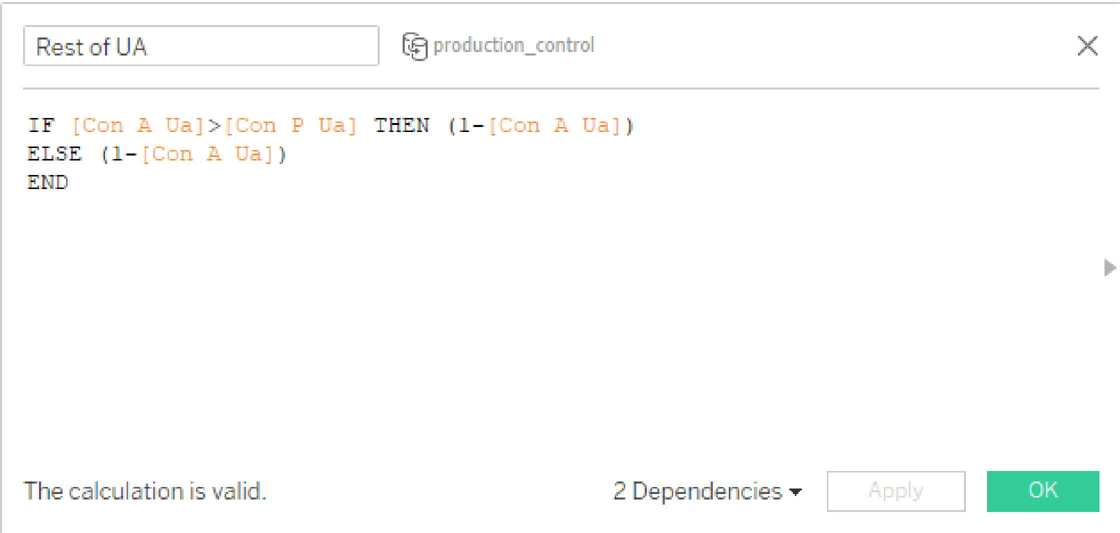
The calculation is valid. 2 Dependencies Apply OK


PA Base
production_control
IF [Con A Pa]>=[Con P Pa] THEN [Con P Pa]
ELSE [Con A Pa]
END
The calculation is valid.
2 Dependencies
Apply
OK


PA Over
production_control
IF [Con A Pa]>[Con P Pa] THEN ([Con A Pa]-[Con P Pa])
ELSE 0
END
The calculation is valid.
2 Dependencies
Apply
OK

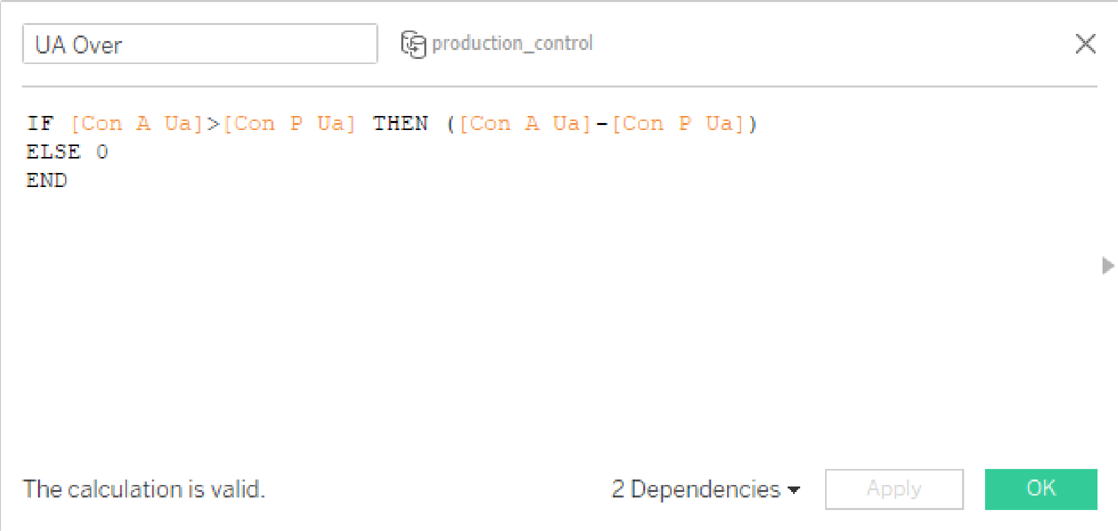

PA Under
production_control
IF [Con A Pa]>[Con P Pa] THEN 0
ELSE ([Con P Pa]-[Con A Pa])
END
The calculation is valid.
2 Dependencies
Apply
OK


Rest of PA
production_control
IF [Con A Pa]>[Con P Pa] THEN (1-[Con A Pa])
ELSE (1-[Con A Pa])
END
The calculation is valid.
2 Dependencies
Apply
OK


Rest of UA
production_control
IF [Con A Ua]>[Con P Ua] THEN (1-[Con A Ua])
ELSE (1-[Con A Ua])
END
The calculation is valid.
2 Dependencies
Apply
OK


Total Actual Production
production_control
SUM({FIXED [Date],[Class],[Con Code] : [Actual Production]})
The calculation is valid.
9 Dependencies
Apply
OK


UA Base
production_control
IF [Con A Ua]>=[Con P Ua] THEN [Con P Ua]
ELSE [Con A Ua]
END
The calculation is valid.
2 Dependencies
Apply
OK


UA Over
production_control
IF [Con A Ua]>[Con P Ua] THEN ([Con A Ua]-[Con P Ua])
ELSE 0
END
The calculation is valid.
2 Dependencies
Apply
OK


UA Under
production_control
IF [Con A Ua]>[Con P Ua] THEN 0
ELSE ([Con P Ua]-[Con A Ua])
END
The calculation is valid.
2 Dependencies
Apply
OK

Class-Extend production_control

```
IF [Class]="OB" THEN "Overburden"
ELSEIF [Class]="CG" THEN "Coal Getting"
END
```

The calculation is valid. 5 Dependencies Apply OK

Filter Daily production_control

```
[Date] = [Param Date]
```

The calculation is valid. 11 Dependencies Apply OK

Filter MTD production_control

```
[Date] <= [Param Date] AND
DATETRUNC( 'month', [Date]) = DATETRUNC('month', [Param Date])
```

The calculation is valid. 3 Dependencies Apply OK

Filter WTD — production_control

```
[Date] <= [Param Date] AND
DATETRUNC( 'week', [Date]) = DATETRUNC('week', [Param Date])
```

The calculation is valid. 3 Dependencies Apply OK

Filter YTD — production_control

```
[Date] <= [Param Date] AND
DATETRUNC( 'year', [Date]) = DATETRUNC('year', [Param Date])
```

The calculation is valid. 3 Dependencies Apply OK

Hauler Production — production_control

```
IF [Sub Class]="Hauling OB"
THEN [A Pa]*[A Ua]*[A Pdty]*[A Qty]*24
END
```

The calculation is valid. 17 Dependencies Apply OK


Loader Production
production_control
IF [Sub Class]="Loading OB"
THEN [A Pa]*[A Ua]*[A Pdty]*[A Qty]*24
ELSE NULL
END
The calculation is valid.
17 Dependencies
Apply
OK


Plan Production
production_control
IF [Sub Class]="Loading OB" OR [Sub Class]="Loading CG"
THEN [P Pa]*[P Ua]*[P Pdty]*[P Qty]*24
END
The calculation is valid.
12 Dependencies
Apply
OK

APPENDIX B

PYTHON CODE FOR DATA MINING

```python
#importing library
import pandas as pd
import numpy as np
from sklearn.cluster import KMeans
from sklearn.metrics import silhouette_score
import seaborn as sns
import matplotlib.pyplot as plt
get_ipython().run_line_magic('matplotlib', 'inline')


#importing dataset
df = pd.read_excel('data_parameter_prod.xlsx')
df.tail(5)

#check if there are any null values
df.isnull().sum()

#check data type for each column
df.info()

#descriptive statistic of numeric data
df.describe()

#Boxplot Visualization
num = ['p_pa','p_ua','p_pdty','p_qty','a_pa','a_ua','a_pdty','a_qty']
def plotBox():
    fig, ax =plt.subplots(2,4, figsize=(16,8))
    i=0;j=0;k=0
    while i<=1:
        while j<=3:
            sns.boxplot(df[num[k]], ax=ax[i, j])
            j+=1;k+=1
        j=0;i+=1;
    plt.show()
plotBox()

# check the anomaly data where actual UA > 1
df_anomaly = df.loc[df['a_ua']>1]
# show anomaly data
df_anomaly

# replace the value of actual UA>1 into plan UA
df_anomaly.loc[:,'a_ua'] = df_anomaly.loc[:,'p_ua']
# show anomaly data (after cleaning)
df_anomaly

# check descriptive statistic of anomaly data (after cleaning)
df_anomaly.describe()
```

```
# replace anomaly (UA>1) in source data with cleaned data
df.loc[df_anomaly.index] = df_anomaly

# descriptive statistic of numeric data
df.describe()

# check the anomaly data where actual PA, UA and Qty < 0
df_anomaly2 = df[(df['a_pa']<0) | (df['a_ua']<0) | (df['a_qty']<0)]
df_anomaly2

# replace the value of actual PA, UA, and Qty < 0 into absolute value (po
sitive)
df_anomaly2[['a_pa','a_ua','a_qty']]= df_anomaly2[['a_pa','a_ua','a_qty']
].abs()
# show anomaly data (after cleaning)
df_anomaly2

# replace anomaly (PA, UA, and Qty < 1) in source data with cleaned data
df.loc[df_anomaly2.index] = df_anomaly2

# final check descriptive statistic of numeric data
df.describe()

plotBox()

# save clean dataset
df.to_excel("data_parameter_cleaned.xlsx", index=False)

# create new columns containing the multiplication of the PA, UA, and Pdt
y parameters with Quantity

df['p_pa_x_qty'] = df['p_pa']*df['p_qty']
df['p_ua_x_qty'] = df['p_ua']*df['p_qty']
df['p_pdty_x_qty'] = df['p_pdty']*df['p_qty']

df['a_pa_x_qty'] = df['a_pa']*df['a_qty']
df['a_ua_x_qty'] = df['a_ua']*df['a_qty']
df['a_pdty_x_qty'] = df['a_pdty']*df['a_qty']

# show data (last rows)
df.tail()

# group data by date, contractor, and class (OB/CG) with sum aggregation
data_by_con = df.groupby(['date','con_code','class']).sum()

# repair the structure of data
data_by_con.columns.name = None
```

```
data_by_con = data_by_con.reset_index()
# show data (after grouping)
data_by_con.tail()


# create new columns containing the division of the (parameter x quantity
) column with total quantity
# this new columns is the average parameter (PA, UA, and Pdty) of each co
ntractor in a daily date

data_by_con['con_p_pa'] = data_by_con['p_pa_x_qty']/data_by_con['p_qty']
data_by_con['con_p_ua'] = data_by_con['p_ua_x_qty']/data_by_con['p_qty']
data_by_con['con_p_pdty'] = data_by_con['p_pdty_x_qty']/data_by_con['p_qt
y']

data_by_con['con_a_pa'] = data_by_con['a_pa_x_qty']/data_by_con['a_qty']
data_by_con['con_a_ua'] = data_by_con['a_ua_x_qty']/data_by_con['a_qty']
data_by_con['con_a_pdty'] = data_by_con['a_pdty_x_qty']/data_by_con['a_qt
y']

data_by_con.tail()

# create new columns containing the deviation of actual parameter with it
s plan parameter

data_by_con['dev_pa'] = (data_by_con['con_a_pa']-
data_by_con['con_p_pa'])/data_by_con['con_p_pa']*100
data_by_con['dev_ua'] = (data_by_con['con_a_ua']-
data_by_con['con_p_ua'])/data_by_con['con_p_ua']*100
data_by_con['dev_pdty'] = (data_by_con['con_a_pdty']-
data_by_con['con_p_pdty'])/data_by_con['con_p_pdty']*100

# create new dataset containing only the columns used for the modelling s
tages
df_new = data_by_con[['date','con_code','class','dev_pa','dev_ua','dev_pd
ty']]
df_new

# check if there is null value
df_new.isna().sum()

# replace null value with 0
df_new = df_new.fillna(0)

# final check descriptive statistic of numeric data
df_new.describe()
```

```
# create X that only contain 3 parameters (PA, UA, Pdty) as input data to
 model
X = df_new[['dev_pa','dev_ua','dev_pdty']].values

# iteration to get SSE and Silhouette Score for each k
clusters_inertia = []
clusters_silhouette = []
df_evaluation = []
for i in range(2,11):
    km = KMeans(n_clusters=i).fit(X)
    clusters_inertia.append(km.inertia_)
    sscore = silhouette_score(X, km.labels_, metric='euclidean')
    clusters_silhouette.append(sscore)
    df_evaluation.append([i, km.inertia_, sscore])

df_evaluation

# Visualize SSE for each k
fig, ax = plt.subplots(figsize=(8, 4))
sns.lineplot(x=list(range(2, 11)), y=clusters_inertia, ax=ax)
ax.set_title('Finding Elbow')
ax.set_xlabel('N Clusters')
ax.set_ylabel('Inertia/SSE')

# Visualize Silhouette Score for each k
fig, ax = plt.subplots(figsize=(8, 4))
sns.lineplot(x=list(range(2, 11)), y=clusters_silhouette, ax=ax)
ax.set_title('Silhouette Score')
ax.set_xlabel('N Clusters')
ax.set_ylabel('Score')

# Training KMeans model with n_clusters (k)=4

model_km = KMeans(n_clusters=4)
y_clusters = model_km.fit_predict(X)

# Save the model to file named "model_kmeans.sav"
# to use in deployment stage

import pickle

filename = 'model_kmeans.sav'
pickle.dump(model_km, open(filename, 'wb'))

get_ipython().run_line_magic('matplotlib', 'qt')

fig = plt.figure(figsize = (15,15))
ax = fig.add_subplot(111, projection='3d')
```

```
val = [100,0,0]
labels = ['x', 'y', 'z']
colors = ['r', 'g', 'b']
for v in range(3):
    x = [val[v-0], -val[v-0]]
    y = [val[v-1], -val[v-1]]
    z = [val[v-2], -val[v-2]]
    ax.plot(x,y,z,'k-', linewidth=1)
    ax.text(val[v-0], val[v-1], val[v-2],
            labels[v], color=colors[v], fontsize=20)

a = np.array(y_clusters==0)
b = np.array(y_clusters==1)
c = np.array(y_clusters==2)
d = np.array(y_clusters==3)

ax.scatter(X[a,0],X[a,1],X[a,2], s=40, color='blue', label="cluster 0")
ax.scatter(X[b,0],X[b,1],X[b,2], s=40, color='orange', label="cluster 1")
ax.scatter(X[c,0],X[c,1],X[c,2], s=40, color='green', label="cluster 2")
ax.scatter(X[d,0],X[d,1],X[d,2], s=40, color='#D12B60', label="cluster 3"
)

ax.set_xlabel('%Dev PA')
ax.set_ylabel('%Dev UA')
ax.set_zlabel('%Dev Pdty')

ax.legend()
plt.show()


# check the centroid point of each cluster
centroids = model_km.cluster_centers_
centroids

# create new column called 'label' in data containing the clusters
df_new['label'] = y_clusters


# check the contractors labelled as cluster 0
df_cluster0 = df_new[df_new['label']==0]
df_cluster0['con_code'].value_counts()


df_new[df_new['label']==0].describe()


# check the contractors labelled as cluster 1
```

```
df_cluster1 = df_new[df_new['label']==1]
df_cluster1['con_code'].value_counts()


df_new[df_new['label']==1].describe()


# check the contractors labelled as cluster 2
df_cluster1 = df_new[df_new['label']==2]
df_cluster1['con_code'].value_counts()


df_new[df_new['label']==2].describe()


# check the contractors labelled as cluster 3
df_cluster1 = df_new[df_new['label']==3]
df_cluster1['con_code'].value_counts()


df_new[df_new['label']==3].describe()
```

APPENDIX C

PYTHON CODE FOR MODEL DEPLOYMENT

```python
# import tabpy library and connect to localhost
import tabpy_client
from tabpy.tabpy_tools.client import Client
client = tabpy_client.Client('http://localhost:9004/')


# load model
import pickle
filename = 'model_kmeans.sav'
model_km = pickle.load(open(filename, 'rb'))


# function to using cluster prediction in Tableau
def cluster_predictor( _arg1, _arg2, _arg3):
    import pandas as pd
    row = {'dev_pa': _arg1[0],
           'dev_ua': _arg2[0],
           'dev_pdty': _arg3[0]}
    #Convert it into a dataframe
    x_data = pd.DataFrame(data=row,index=[0])
    #Predict the cluster
    return int(model_km.predict(x_data)[0])

 model deployment to Tableau
client.deploy('cluster_predictor', cluster_predictor,'Clustering',overrid
e = True)
```

APPENDIX D

SAMPLE DATASET PARAMETER

Table D. Sample Dataset Parameter

Index	date	con_code	site_code	class	sub_class	equipment	p_pa	p_ua	p_pdty	p_qty	a_pa	a_ua	a_pdty	a_qty
0	01/01/2022	A	P	OB	Loading OB	EX2500	0.8549	0.476662	924	2	0.75	0	0	4
1	01/01/2022	A	P	OB	Loading OB	PC2000	0.8749	0.488626	711	18	1	0	0	18
2	01/01/2022	A	P	OB	Loading OB	PC1250	0.8749	0.488626	350	2	1	0	0	0
3	01/01/2022	A	P	OB	Hauling OB	HD785	0.890905	0	101.0492	168	0.950311	0	0	167
4	01/01/2022	A	P	CG	Loading CG	PC400	0.901961	0	190	0.75	1	0	0	5
......	......	......	......	......	......	......	......	......	......	......	......	......	......	......
8877	31/05/2022	D	R	CG	Hauling CG	Scania P360	0.85	0.5	25	5	0.62	0.34	20	4
8878	31/05/2022	D	R	CG	Hauling CG	Scania P460	0.89	0.56	34	10	0.94	0.35	30	9
8879	31/05/2022	G	S	OB	Loading OB	PC2000	0.92	0.58	734	1	1	0.65125	783.7797	1
8880	31/05/2022	G	S	OB	Loading OB	EX1200	0.92	0.58	525	6	1	0.624444	441.9505	6
8881	31/05/2022	G	S	OB	Loading OB	PC800	0.92	0.58	260	2	0.9325	0.742627	186.0923	2

APPENDIX E

SAMPLE DATASET TUM (TIME USAGE MANAGEMENT)

Table E. Sample Dataset TUM (Time Usage Management)

Index	date	con_code	site_code	lag_indicator	sublead_indicator	p_hr	a_hr
1	02/01/2022	C	O	OB	Holiday	0	0
2	02/01/2022	C	O	OB	Rain	3.870968	0.457504
3	02/01/2022	C	O	OB	Slippery	0.851613	0
4	02/01/2022	C	O	OB	Shift Change	0.333333	0.417062
5	02/01/2022	C	O	OB	Safety Talk	0	0
6	02/01/2022	C	O	OB	Meal & Rest	2	1.921011
7	02/01/2022	C	O	OB	Praying	0	0.032859
8	02/01/2022	C	O	OB	Friday Praying	0	0
9	02/01/2022	C	O	OB	Blasting	0	0
10	02/01/2022	C	O	OB	Refueling	0.333333	0
11	02/01/2022	C	O	OB	Fasting	0	0
12	02/01/2022	C	O	OB	Front & Disposal Repair	0.333333	2.466983
13	02/01/2022	C	O	OB	Equipment Travel	0.166667	0.154186
14	02/01/2022	C	O	OB	Road Repair	0	0
15	02/01/2022	C	O	OB	No Work Location	0	0
…..	…..	…..	…..	…..	…..	…..	…..
67454	29/05/2022	G	S	OB	Road Repair	0	1.414444
67455	29/05/2022	G	S	OB	No Work Location	0	0
67456	29/05/2022	G	S	OB	No Operator (EXC)	0	0
67457	29/05/2022	G	S	OB	Wait Operator (EXC)	0	0.092222
67458	29/05/2022	G	S	OB	No Hauler/Truck	0	0
67459	29/05/2022	G	S	OB	Wait Hauler/Truck	0	1.414444
67460	29/05/2022	G	S	OB	Wait Support	0	0
67461	29/05/2022	G	S	OB	Internal Problem	0	0
67462	29/05/2022	G	S	OB	Customer Problem	0	0
67463	29/05/2022	G	S	OB	No Material	0	0
67464	29/05/2022	G	S	OB	Relocation	0	0
67465	29/05/2022	G	S	OB	Safety	0	0.008888
67466	29/05/2022	G	S	OB	Hazard	0	0
67467	29/05/2022	G	S	OB	Bad Visibility	0.008333	0
67468	29/05/2022	G	S	OB	External Issues	0	0.277777