TUGAS

EXPLORATORY DATA ANALYSIS

Pada kesempatan ini Saya diberikan tugas untuk melakukan eksplorasi pada data Principal. Principal sendiri adalah nasabah korporasi existingBRI. Dalam hal ini korporasi yang dimaksud adalah korporasi Pertamina dan korporasi Sampoerna. Hasil eksplorasi ini nantinya dapat membantu dalam memprediksi potensi simpanan dan pinjaman kedua korporasi tersebut. Potensi simpanan menggunakan rekening giro operasional untuk diberikan rekomendasi top up dan untuk membuka rekening baru. Sedangkan potensi pinjaman dari oustanding pinjaman untuk diberikan rekomendasi top up dan untuk membuka rekening pinjaman. Biasanya hasil peramalan dari modelnya berupa data hasil prediksi selama 12 bulan.

Hal-hal yang Saya kerjakan dalam melakukan EDA untuk Principal antara lain sebagai berikut.

1) Data Preparation

2) Basic Data Exploration

3) Perhitungan Korelasi Pearson

4) Pembuatan Heatmap untuk setiap korporasi

5) Features Selection

6) Pembuatan Scatterplot hasil dari features selection

7) Membuat rekomendasi fitur yang sebaiknya digunakan serta interpretasinya

Berikut syntax yang telah Saya buat untuk dapat melakukan eksplorasi data principal berdasarkan ketujuh poin diatas.

## Impor modul-modul yang sekiranya akan digunakan

# ====================================================

from fbprophet import Prophet

from fbprophet.diagnostics import cross\_validation, performance\_metrics

from fbprophet.plot import plot\_cross\_validation\_metric

from time import time

from datetime import datetime, timedelta, date

from dateutil.relativedelta import relativedelta

from pyspark.sql import SparkSession, functions as F, window as W

from pyspark.sql.types import \*

from time import time

import numpy as np

import seaborn as sns

import matplotlib.pyplot as plt

from matplotlib.ticker import PercentFormatter

from statsmodels.tsa.stattools import adfuller, kpss

from statsmodels.tsa.stattools import acf, pacf

from statsmodels.graphics.tsaplots import plot\_acf, plot\_pacf

from sklearn.metrics import mean\_squared\_error

from permetrics.regression import Metrics

import pandas as pd

pd.options.display.html.table\_schema = True

pd.options.display.max\_columns = 999

pd.options.display.max\_rows = 999

import warnings

warnings.filterwarnings("ignore")

## Pengaturan Spark

#======================================================================

spark = SparkSession\

.builder\

.appName('EDA Prinsipal')\

.config('spark.dynamicAllocation.enabled','false')\

.config('spark.executor.instances','4')\

.config('spark.executor.cores','5')\

.config('spark.executor.memory','10g')\

.config('spark.yarn.executor.memoryOverhead','5g')\

.config('spark.network.timeout',60)\

.enableHiveSupport()\

.getOrCreate()

## Membuat timer selama running

#======================================================================

def set\_timer():

global START\_TIME

START\_TIME = time()

def get\_timer():

t = datetime(1,1,1)+timedelta(seconds=int(time()-START\_TIME))

return "{}:{}:{}".format(str(t.hour).zfill(2),str(t.minute).zfill(2),str(t.second).zfill(2))

set\_timer()

## Data Preparation

#======================================================================

spark\_cif = spark.read.table('datamart.bribrain\_mas\_brimopp\_sum\_trx\_sampoerna\_pertamina')

#spark\_uniq = spark\_cif.dropDuplicates(['corporate\_code','ds']) #tidak ada duplicate

len(spark\_cif.columns)

## Basic Data Exploration

#======================================================================

data\_df = spark\_cif.toPandas().reset\_index(drop=True)

data\_df = data\_df.fillna('NaN')

data\_df.info()

kolom\_toconvert = ['sum\_cbal\_base\_giro',

'avg\_ratas\_saldo\_giro',

'sum\_outstanding',

'sum\_plafon',

'sum\_amt\_credit\_giro',

'avg\_amt\_credit\_giro',

'sum\_amt\_debit\_giro',

'avg\_amt\_debit\_giro']

data\_df[kolom\_toconvert] = data\_df[kolom\_toconvert].astype('float')

data\_df['year'] = data\_df['ds'].apply(lambda x: x[0:4])

data\_df['month'] = data\_df['ds'].apply(lambda x: x[4:])

data\_df['date'] = data\_df['ds'].apply(lambda x: x[0:4]+'-'+x[4:])

data\_df['date'] = data\_df['date'].apply(lambda x: pd.to\_datetime(x).strftime('%Y-%m'))

data\_df.info()

## Menghitung Nilai Korelasi Pearson

#======================================================================

'''df['nama\_kolom'].sum().reset\_index() === jika ada yg kebetulan mempunyai corporate\_code,group\_name dan date

yg sama, maka tiap baris yg sama dalam suatu kolom akan dijumlahkan(digabungkan menjadi 1 baris) '''

grup\_cif\_ds = data\_df.groupby(['corporate\_code','group\_name','date']).sum().reset\_index()

len(data\_df)

grup\_cif\_ds.info()

len(grup\_cif\_ds)

# jumlah baris tidak berubah karena kelompok date tidak memiliki duplikat

df\_corr = grup\_cif\_ds.copy()

df\_corr.set\_index('date',inplace=True)

df\_corr.index = pd.to\_datetime(df\_corr.index)

df\_corr.index = df\_corr.index.to\_period('M').to\_timestamp('M')

corr\_results = df\_corr.drop(['corporate\_code','group\_name'],axis=1).corr()

corr\_results.shape

## Membuat Heatmap

#======================================================================

mask = np.triu(np.ones\_like(corr\_results, dtype=bool))

sns.heatmap(corr\_results,cmap='coolwarm', annot=True,annot\_kws={"size": 8},mask=mask)

print("VCIF yang harus di eksplor="+str(len(pd.unique(df\_corr['corporate\_code']))))

def heatmap\_func(data,cif):

mask = np.triu(np.ones\_like(data, dtype=bool))

fig, ax = plt.subplots(figsize=(7,5))

ax = sns.heatmap(data, cmap='coolwarm',annot=True, annot\_kws={"size": 10},mask=mask)

plt.title('Correlation Heat Map pada {}'.format(cif),pad=20)

plt.show()

for cif in df\_corr['corporate\_code'].unique():

corr\_temp = df\_corr[df\_corr['corporate\_code']==cif].drop(['corporate\_code','group\_name'],axis=1).corr()

heatmap\_func(corr\_temp,cif)

## SELEKSI FITUR DAN PEMBUATAN SCATTER PLOT

#======================================================================

df\_plot=grup\_cif\_ds.copy().drop(['group\_name','date'],axis=1)

df\_plot.set\_index('corporate\_code',inplace=True)

print("VCIF yang harus di eksplor="+str(len(pd.unique(df\_corr['corporate\_code']))))

#1

df\_plot\_NAN=df\_plot.loc['NaN']

#Set figure size (width,height) in inches

plt.figure(figsize=(5,3))

sns.scatterplot(data=df\_plot\_NAN, x="avg\_ratas\_saldo\_giro",y="avg\_amt\_credit\_giro")

plt.show()

#2

df\_plot\_VCIF1017=df\_plot.loc['VCIF1017']

df\_plot\_VCIF1017=df\_plot\_VCIF1017.drop(['sum\_cbal\_base\_giro','sum\_freq\_debit\_giro','sum\_outstanding','sum\_plafon'],axis=1)

sns.pairplot(df\_plot\_VCIF1017,corner=True)

df\_plot\_VCIF1017.columns

#3

df\_plot\_VCIF1018=df\_plot.loc['VCIF1018']

df\_plot\_VCIF1018=df\_plot\_VCIF1018.drop(['sum\_cbal\_base\_giro','avg\_ratas\_saldo\_giro','sum\_plafon'],axis=1)

sns.pairplot(df\_plot\_VCIF1018,corner=True)

df\_plot\_VCIF1018.columns

#4

df\_plot\_VCIF1019=df\_plot.loc['VCIF1019']

df\_plot\_VCIF1019=df\_plot\_VCIF1019.drop(['sum\_cbal\_base\_giro','avg\_ratas\_saldo\_giro','sum\_plafon'],axis=1)

sns.pairplot(df\_plot\_VCIF1019,corner=True)

df\_plot\_VCIF1019.columns

#5

df\_plot\_VCIF1160=df\_plot.loc['VCIF1160']

df\_plot\_VCIF1160=df\_plot\_VCIF1160

sns.pairplot(df\_plot\_VCIF1160,corner=True)

df\_plot\_VCIF1160.columns

#6 (1 fitur)

df\_plot\_VCIF1231=df\_plot.loc['VCIF1231']

df\_plot\_VCIF1231\_copy=df\_plot\_VCIF1231.copy()

df\_plot\_VCIF1231=df\_plot\_VCIF1231['sum\_freq\_debit\_giro']

sns.pairplot(df\_plot\_VCIF1231\_copy,

x\_vars=[ 'sum\_cbal\_base\_giro','avg\_ratas\_saldo\_giro', 'sum\_outstanding', 'sum\_plafon','sum\_freq\_credit\_giro', 'sum\_amt\_credit\_giro', 'avg\_amt\_credit\_giro','sum\_freq\_debit\_giro', 'sum\_amt\_debit\_giro', 'avg\_amt\_debit\_giro'],

y\_vars=['sum\_freq\_debit\_giro'])

#7

df\_plot\_VCIF1233=df\_plot.loc['VCIF1233']

df\_plot\_VCIF1233\_copy=df\_plot\_VCIF1233.copy()

df\_plot\_VCIF1233=df\_plot['avg\_ratas\_saldo\_giro']

sns.pairplot(df\_plot\_VCIF1233\_copy,

x\_vars=['sum\_cbal\_base\_giro', 'avg\_ratas\_saldo\_giro', 'sum\_outstanding', 'sum\_plafon','sum\_freq\_credit\_giro', 'sum\_amt\_credit\_giro', 'avg\_amt\_credit\_giro','sum\_freq\_debit\_giro', 'sum\_amt\_debit\_giro', 'avg\_amt\_debit\_giro'],

y\_vars=['avg\_ratas\_saldo\_giro'])

#8 df\_plot\_VCIF1234=0

#9 df\_plot\_VCIF1235=0

#10 df\_plot\_VCIF1236=0'''

#11

df\_plot\_VCIF1237=df\_plot.loc['VCIF1237']

df\_plot\_VCIF1237=df\_plot\_VCIF1237[['sum\_cbal\_base\_giro','avg\_ratas\_saldo\_giro','sum\_amt\_credit\_giro','sum\_amt\_debit\_giro']]

sns.pairplot(df\_plot\_VCIF1237,corner=True)

df\_plot\_VCIF1237.columns

'''#12 df\_plot\_VCIF1239=0

#13 df\_plot\_VCIF1240=0

#14 df\_plot\_VCIF1242=0

#15

df\_plot\_VCIF1243=df\_plot.loc['VCIF1243']

df\_plot\_VCIF1243=df\_plot\_VCIF1243['sum\_freq\_credit\_giro']

sns.pairplot(df\_plot,

x\_vars=['corporate\_code', 'group\_name', 'sum\_cbal\_base\_giro','avg\_ratas\_saldo\_giro', 'sum\_outstanding', 'sum\_plafon','sum\_freq\_credit\_giro', 'sum\_amt\_credit\_giro', 'avg\_amt\_credit\_giro','sum\_freq\_debit\_giro', 'sum\_amt\_debit\_giro', 'avg\_amt\_debit\_giro'],

y\_vars=['sum\_freq\_credit\_giro'])

#16

df\_plot\_VCIF1246=df\_plot['avg\_ratas\_saldo\_giro']

'''

#17

df\_plot\_VCIF284=df\_plot.loc['VCIF284']

sns.pairplot(df\_plot\_VCIF284,corner=True)

df\_plot\_VCIF284.columns

#18 df\_plot\_VCIF285=0

#19 df\_plot\_VCIF286=0

#20

df\_plot\_VCIF488=df\_plot['sum\_freq\_debit\_giro']

#21

df\_plot\_VCIF489=df\_plot['sum\_freq\_debit\_giro']

#22

df\_plot\_VCIF490=df\_plot['sum\_amt\_debit\_giro']

#23 df\_plot\_VCIF491=0

#24

df\_plot\_VCIF492=df\_plot.loc['VCIF492']

df\_plot\_VCIF492=df\_plot\_VCIF492[['sum\_amt\_credit\_giro','sum\_freq\_debit\_giro','sum\_amt\_debit\_giro']]

sns.pairplot(df\_plot\_VCIF492,corner=True)

'''#25

df\_plot\_VCIF493=df\_plot.loc['VCIF493']

df\_plot\_VCIF493=df\_plot['avg\_ratas\_saldo\_giro']

#26 df\_plot\_VCIF495=0

#27 df\_plot\_VCIF496=0

'''

#28

df\_plot\_VCIF497=df\_plot.loc['VCIF497']

df\_plot\_VCIF497=df\_plot\_VCIF497[['sum\_freq\_credit\_giro','sum\_amt\_credit\_giro','avg\_amt\_credit\_giro','sum\_amt\_debit\_giro','avg\_amt\_debit\_giro']]

sns.pairplot(df\_plot\_VCIF497,corner=True)

#29 df\_plot\_VCIF498=0

#30

df\_plot\_VCIF499=df\_plot.loc['VCIF499']

sns.scatterplot(data=df\_plot\_VCIF499,x="sum\_amt\_credit\_giro",y="sum\_amt\_debit\_giro")

#31

df\_plot\_VCIF500=df\_plot.loc['VCIF500']

df\_plot\_VCIF500=df\_plot\_VCIF500.drop(['sum\_cbal\_base\_giro','sum\_outstanding','sum\_plafon','sum\_freq\_debit\_giro'],axis=1)

sns.pairplot(df\_plot\_VCIF500,corner=True)

df\_plot\_VCIF500.columns

#32 df\_plot\_VCIF510=0

#33 df\_plot\_VCIF604=0

#34 df\_plot\_VCIF676=0

#35 df\_plot\_VCIF751=0

#36

df\_plot\_VCIF761=df\_plot.loc['VCIF761']

df\_plot\_VCIF761=df\_plot\_VCIF761[['sum\_cbal\_base\_giro','avg\_ratas\_saldo\_giro']]

sns.pairplot(df\_plot\_VCIF761,corner=True)

sns.scatterplot(data=df\_plot\_VCIF761,x="sum\_cbal\_base\_giro",y="avg\_ratas\_saldo\_giro")

'''

Feature Selection

=====================

def correlation(data, threshold,cif):

global corr\_results, corr\_features

col\_corr = set() # mengumpulkan nama kolomnya

for i in range(len(corr\_results.columns)):

for j in range(i):

if abs(corr\_results.iloc[i, j]) < threshold: # di absolute kan karena nilai korelasi (-) hanya menunjukkan arah

colname = corr\_results.columns[i] # getting the name of column

col\_corr.add(colname)

return col\_corr

for cif in df\_corr['corporate\_code'].unique():

corr\_temp = df\_corr[df\_corr['corporate\_code']==cif].drop(['corporate\_code','group\_name'],axis=1).corr()

corr\_features = correlation(corr\_temp, 0.2,cif)

print(cif+':'+str(corr\_features))

len(set(corr\_features))

len(df\_corr.columns)

'''

print("Running took {}".format(get\_timer()))

## Rangkuman Fitur

#======================================================================

df=spark\_cif.toPandas()

df\_summary=pd.DataFrame()

df\_summary['corporate\_code']=df['corporate\_code'].unique()

df\_summary1=pd.DataFrame()

df\_summary1['avg\_ratas\_saldo vs sum\_outstanding']=df\_plot.groupby('corporate\_code')[['avg\_ratas\_saldo\_giro','sum\_outstanding']].corr().iloc[::2,-1]

df\_summary1['avg\_ratas\_saldo vs sum\_amt\_credit']=df\_plot.groupby('corporate\_code')[['avg\_ratas\_saldo\_giro','sum\_amt\_credit\_giro']].corr().iloc[::2,-1]

df\_summary1['avg\_ratas\_saldo vs sum\_amt\_debit']=df\_plot.groupby('corporate\_code')[['avg\_ratas\_saldo\_giro','sum\_amt\_debit\_giro']].corr().iloc[::2,-1]

df\_summary1=df\_summary1.reset\_index(drop=True)

df\_summary2=pd.DataFrame()

df\_summary2['sum\_outstanding vs sum\_amt\_credit']=df\_plot.groupby('corporate\_code')[['sum\_outstanding','sum\_amt\_credit\_giro']].corr().iloc[::2,-1]

df\_summary2['sum\_outstanding vs sum\_amt\_dedit']=df\_plot.groupby('corporate\_code')[['sum\_outstanding','sum\_amt\_debit\_giro']].corr().iloc[::2,-1]

df\_summary2=df\_summary2.reset\_index(drop=True)

df\_summary=pd.concat([df\_summary1,df\_summary2],axis=1)

#======================================================================

df = spark\_cif.toPandas().reset\_index(drop=True)

df\_summary['corporate\_code']=df['corporate\_code'].unique()

df\_summary.set\_index('corporate\_code',inplace=True)

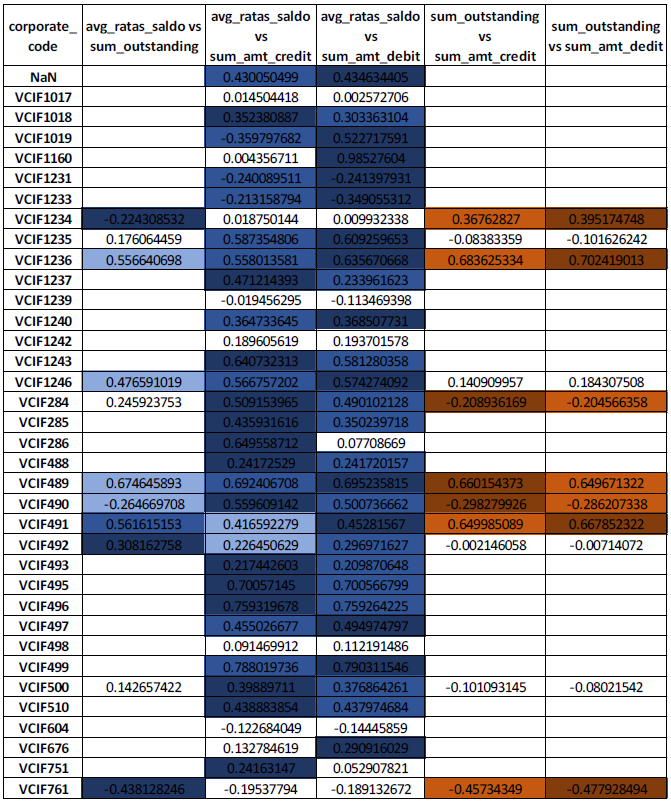
df\_summary

df\_summary.to\_csv('./*Merchant* Score/BRIMOPP/Network Analysis Supplier & Buyer/Playground/MIDAH/tabel\_rangkuman\_korelasi.csv',sep=',')

Detail penjabaran syntax dan outputnya dapat dilihat pada dokumentasi berikut :

<https://docs.google.com/document/d/1fWXACwGWv29GIcudu-6KcD0KbHQzbRui/edit?usp=sharing&ouid=112484286986617075679&rtpof=true&sd=true>

Jika mengacu pada hasil akhir dari EDA yang telah dilakukan, kesimpulan yang Saya dapatkan adalah memberikan rekomendasi fitur-fitur yang sebaiknya digunakan.



Rekomendasi fitur yang dapat digunakan :

|  |  |  |
| --- | --- | --- |
| **ratas\_saldo + debit** | **+ credit** | **+ outstanding** |
| NaN | NaN | VCIF1246 |
| VCIF1019 | VCIF1019 | VCIF489 |
| VCIF1160 | VCIF1231 | VCIF490 |
| VCIF1231 | VCIF1235 |  |
| VCIF1235 | VCIF1240 |  |
| VCIF1240 | VCIF1246 |  |
| VCIF1246 | VCIF489 |  |
| VCIF489 | VCIF490 |  |
| VCIF490 | VCIF497 |  |
| VCIF497 | VCIF499 |  |
| VCIF499 |  |  |
| VCIF676 |  |  |

|  |  |  |
| --- | --- | --- |
| **ratas\_saldo + credit** | **+ debit** | **+ outstanding** |
| VCIF1018 | VCIF1018 | VCIF284 |
| VCIF1233 | VCIF1233 |  |
| VCIF1237 | VCIF1237 |  |
| VCIF1243 | VCIF1243 |  |
| VCIF284 | VCIF284 |  |
| VCIF285 | VCIF285 |  |
| VCIF286 | VCIF488 |  |
| VCIF488 | VCIF493 |  |
| VCIF493 | VCIF495 |  |
| VCIF495 | VCIF496 |  |
| VCIF496 | VCIF500 |  |
| VCIF500 | VCIF510 |  |
| VCIF510 |  |  |
| VCIF751 |  |  |

|  |  |
| --- | --- |
| **ratas\_saldo + outstanding + debit + credit** |  |
| VCIF492 |  |

|  |  |
| --- | --- |
| **outstanding + debit** | **+ credit + ratas\_saldo** |
| VCIF1234 | VCIF1234 |
| VCIF1236 | VCIF491 |
| VCIF491 | VCIF761 |
| VCIF761 | **+ ratas\_saldo + credit** |
|  | VCIF1236 |

**Interpretasi :**

Untuk 12 Corporate yang awal (NaN,VCIF1019, … , VCIF676) direkomendasikan untuk menggunakan 2 fitur yaitu avg\_ratas\_saldo\_giro dan sum\_amt\_debit. Hal ini didasarkan pada nilai korelasi yang cukup kuat antara kedua fitur tersebut dibanding nilai korelasi dari kombinasi dengan fitur lainnya. Selanjutnya jika dibutuhkan 3 fitur untuk ditambahkan maka akan direkomendasikan untuk menambahkan fitur sum\_amt\_credit dengan dasar yang sama. Begitu pula jika dibutuhkan tambahan 4 fitur, maka fitur yang direkomendasikan adalah fitur sum\_outstanding dengan dasar yang sama.

Namun pada penambahan fitur sum\_amt\_credit, tidak semua corporate yang awal dapat ditambahkan fitur sum\_amt\_credit. Hal ini dikarenakan tidak semua corporate yang awal memiliki nilai korelasi yang cukup bagus dengan sum\_amt\_credit ( r ≥ 0,2). Seperti misalnya, pada penambahan fitur sum\_amt\_credit tidak dimasukkan VCIF676 karena jika kita lihat nilai korelasi antara ratas\_saldo\_giro dengan sum\_amt\_credit nilainya masih sangat rendah yakni hanya sebesar 0,133. Sehingga kita tidak bisa menambahkan fitur sum\_amt\_credit untuk VCIF676. Namun, mungkin saja jika dibutuhkan 3 fitur atau lebih untuk VCIF676 bisa menggunakan fitur lain diluar 4 fitur yang sedang dianalisis. Hal ini berlaku juga untuk penambahan fitur keempat (sum\_outstanding) yang tidak menyertakan semua corporate yang awal.

Penjelasan di atas juga berlaku untuk rekomendasi fitur-fitur lainnya.