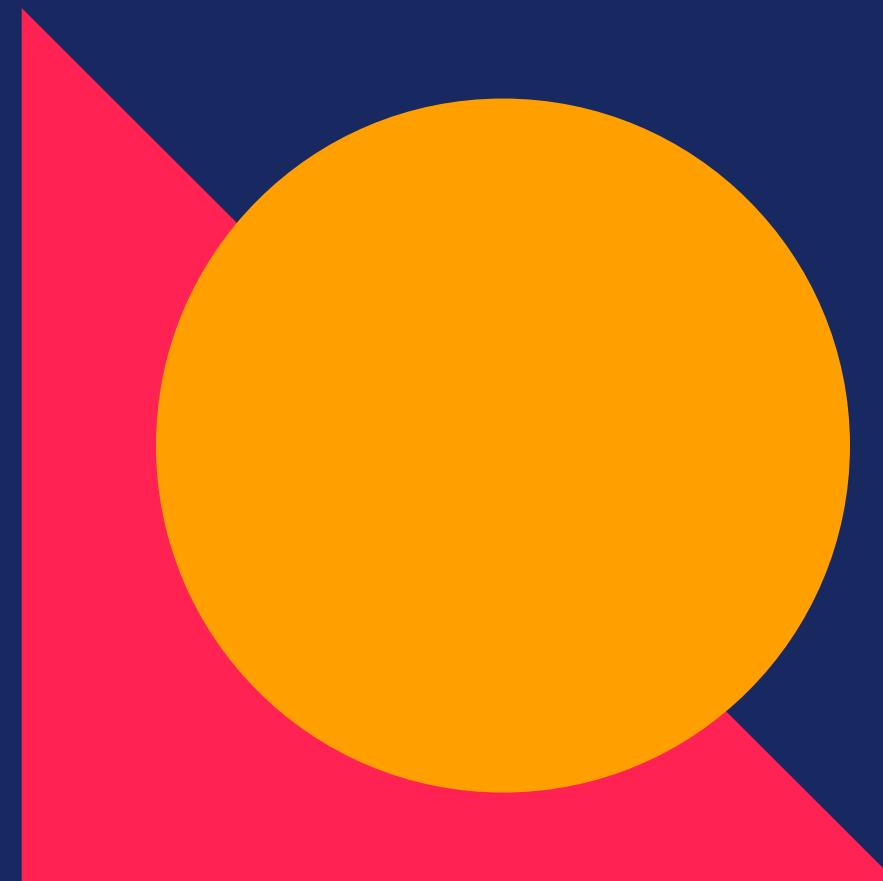


Nando Dyas Arya



Dashboard 1

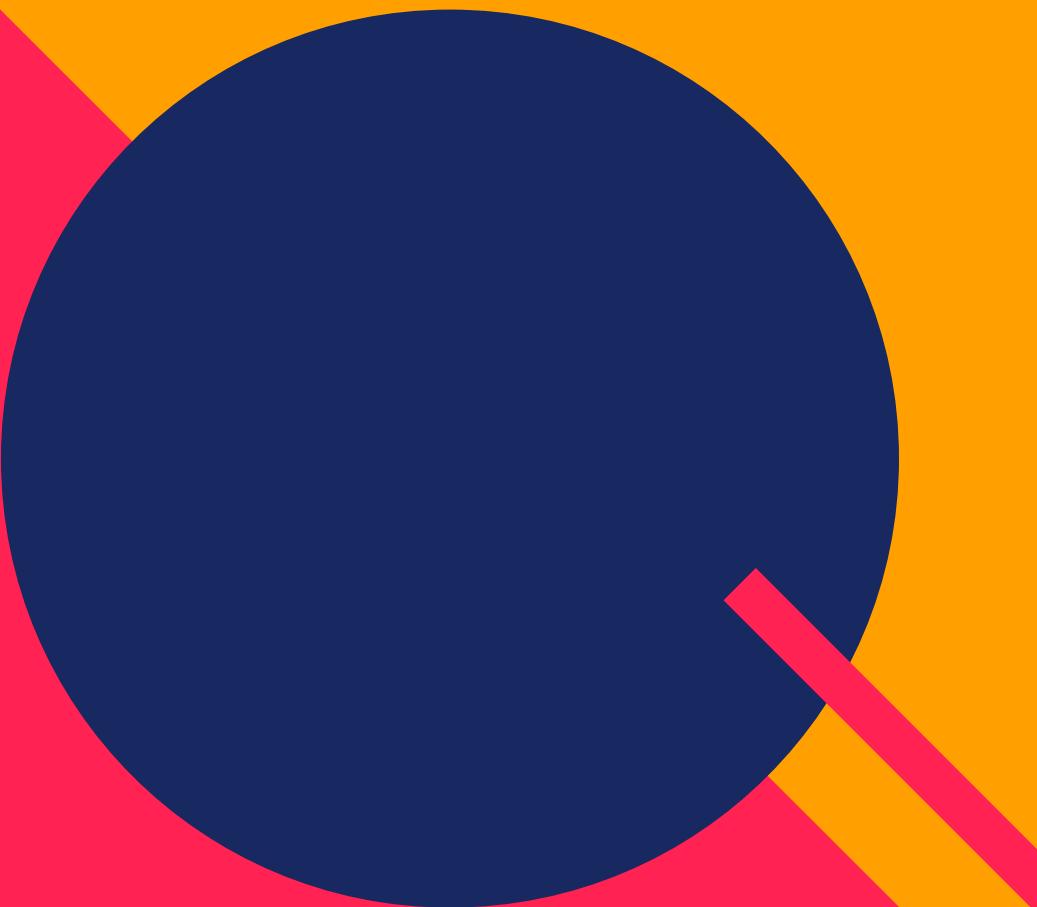
FINAL PROJECT CELERATES



Discussion Points

- Data Preprocessing
- Modelling
- Data Testing

FIRST TOPIC PART 1



Data
Preprocessing



The image shows two identical Google Colab notebooks side-by-side. Both notebooks have a dark theme and are titled "Final Project Celerates.ipynb".

Code in Cell [3]:

```
#import data path
df_male = pd.read_csv('/content/drive/MyDrive/Celerates BDBI/Data Set final project/application_history_m.csv', delimiter=';')
df_female = pd.read_csv('/content/drive/MyDrive/Celerates BDBI/Data Set final project/application_history_f.csv', delimiter=';')
df_credit_history = pd.read_csv('/content/drive/MyDrive/Celerates BDBI/Data Set final project/credit_history.csv', sep=',')
```

Code in Cell [4]:

```
#merge data f and m
df_fm = pd.concat([df_male, df_female])
df_fm.head()
```

Output of df_fm.head():

	Id_customer	JK	KepemilikanMobil	KepemilikanProperti	JmlAnak	Pendapatan	TipePendapatan	TingkatPendidikan	StatusKeluarga	TipeRumah	FlagMobile	FlagWor
0	5008804	Laki-laki	Ya	Ya	0	427500	Bekerja	PG	M	RA	1	
1	5008805	Laki-laki	Ya	Ya	0	427500	Bekerja	PG	M	RA	1	
2	5008806	Laki-laki	Ya	Ya	0	112500	Bekerja	G	M	MH	1	
3	5008815	Laki-laki	Ya	Ya	0	270000	Bekerja	PG	M	MH	1	
4	5112956	Laki-laki	Ya	Ya	0	270000	Bekerja	PG	M	MH	1	

Code in Cell [5]:

```
#df_last_credit_history = df_last_credit_history[df_last_credit_history['Overdue'] != 'Tidak memiliki pinjaman']
df_last_credit_history = df_credit_history.groupby('Id_customer').max() #maximum value
df_last_credit_history.head()
```

Output of df_last_credit_history.head():

	Overdue
5001711	Tidak memiliki pinjaman
5001712	0
5001713	Tidak memiliki pinjaman
5001714	Tidak memiliki pinjaman
5001715	Tidak memiliki pinjaman

Code in Cell [6]:

```
#delete values Tidak memiliki pinajaman on field overdue
df_last_credit_history = df_last_credit_history[df_last_credit_history['Overdue'] != 'Tidak memiliki pinjaman']
```

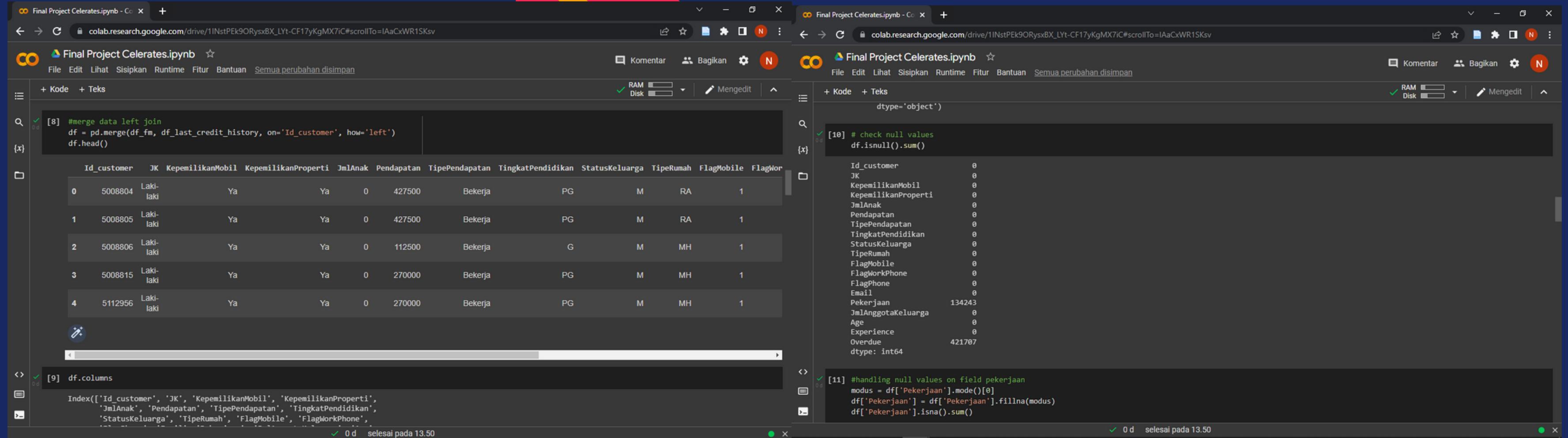
Code in Cell [7]:

```
#checking values on Overdue
df_last_credit_history.value_counts()
```

Output of df_last_credit_history.value_counts():

Overdue	Count
0	18086
1	2470
2	164
5	106
3	35

Read data and get max values from credit history



The image shows two side-by-side Google Colab notebooks. Both notebooks have the title "Final Project Celerates.ipynb".

Notebook 1 (Left):

- Cell [8]:

```
#merge data left join
df = pd.merge(df_fm, df_last_credit_history, on='Id_customer', how='left')
df.head()
```
- Output:

	Id_customer	JK	KepemilikanMobil	KepemilikanProperti	JmlAnak	Pendapatan	TipePendapatan	TingkatPendidikan	StatusKeluarga	TipeRumah	FlagMobile	FlagWorkPhone
0	5008804	Laki-laki	Ya	Ya	0	427500	Bekerja	PG	M	RA	1	
1	5008805	Laki-laki	Ya	Ya	0	427500	Bekerja	PG	M	RA	1	
2	5008806	Laki-laki	Ya	Ya	0	112500	Bekerja	G	M	MH	1	
3	5008815	Laki-laki	Ya	Ya	0	270000	Bekerja	PG	M	MH	1	
4	5112956	Laki-laki	Ya	Ya	0	270000	Bekerja	PG	M	MH	1	
- Cell [9]:

```
df.columns
```
- Output:

```
Index(['Id_customer', 'JK', 'KepemilikanMobil', 'KepemilikanProperti', 'JmlAnak', 'Pendapatan', 'TipePendapatan', 'TingkatPendidikan', 'StatusKeluarga', 'TipeRumah', 'FlagMobile', 'FlagWorkPhone', 'Pekerjaan', 'JmlAnggotaKeluarga', 'Age', 'Experience', 'Overdue'], dtype='object')
```

Notebook 2 (Right):

- Cell [10]:

```
# check null values
df.isnull().sum()
```
- Output:

```
Id_customer          0
JK                  0
KepemilikanMobil      0
KepemilikanProperti    0
JmlAnak              0
Pendapatan            0
TipePendapatan        0
TingkatPendidikan      0
StatusKeluarga        0
TipeRumah              0
FlagMobile             0
FlagWorkPhone          0
FlagPhone              0
Email                 0
Pekerjaan           134243
JmlAnggotaKeluarga      0
Age                  0
Experience            0
Overdue             421707
dtype: int64
```
- Cell [11]:

```
#handling null values on field pekerjaan
modus = df['Pekerjaan'].mode()[0]
df['Pekerjaan'] = df['Pekerjaan'].fillna(modus)
df['Pekerjaan'].isna().sum()
```

Join male and female data to credit history

Handling null values on field pekerjaan

```
[12] df.isnull().sum()
0 d
Id_customer      0
JK                0
KepemilikanMobil 0
KepemilikanProperti 0
JmlAnak          0
Pendapatan        0
TipePendapatan    0
TingkatPendidikan 0
StatusKeluarga    0
TipeRumah         0
FlagMobile        0
FlagWorkPhone     0
FlagPhone         0
Email              0
Pekerjaan         0
JmlAnggotaKeluarga 0
Age                0
Experience        0
Overdue           421707
dtype: int64

[13] #handling null values on overdue
df.dropna(subset=['Overdue'], how='all', inplace=True)

[14] df.isnull().sum()
0 d
Id_customer      0
JK                0
KepemilikanMobil 0
KepemilikanProperti 0
JmlAnak          0
Pendapatan        0
TipePendapatan    0
TingkatPendidikan 0
StatusKeluarga    0
TipeRumah         0
FlagMobile        0
FlagWorkPhone     0
FlagPhone         0
Email              0
Pekerjaan         0
JmlAnggotaKeluarga 0
Age                0
Experience        0
Overdue           0
dtype: int64

[15] df
0 d
```

Removes null values in overdue as
they are not used in creating the
model

The image shows two adjacent Google Colab notebooks. Both have a dark theme and are titled 'Final Project Celerates.ipynb'. The left notebook contains the following code:

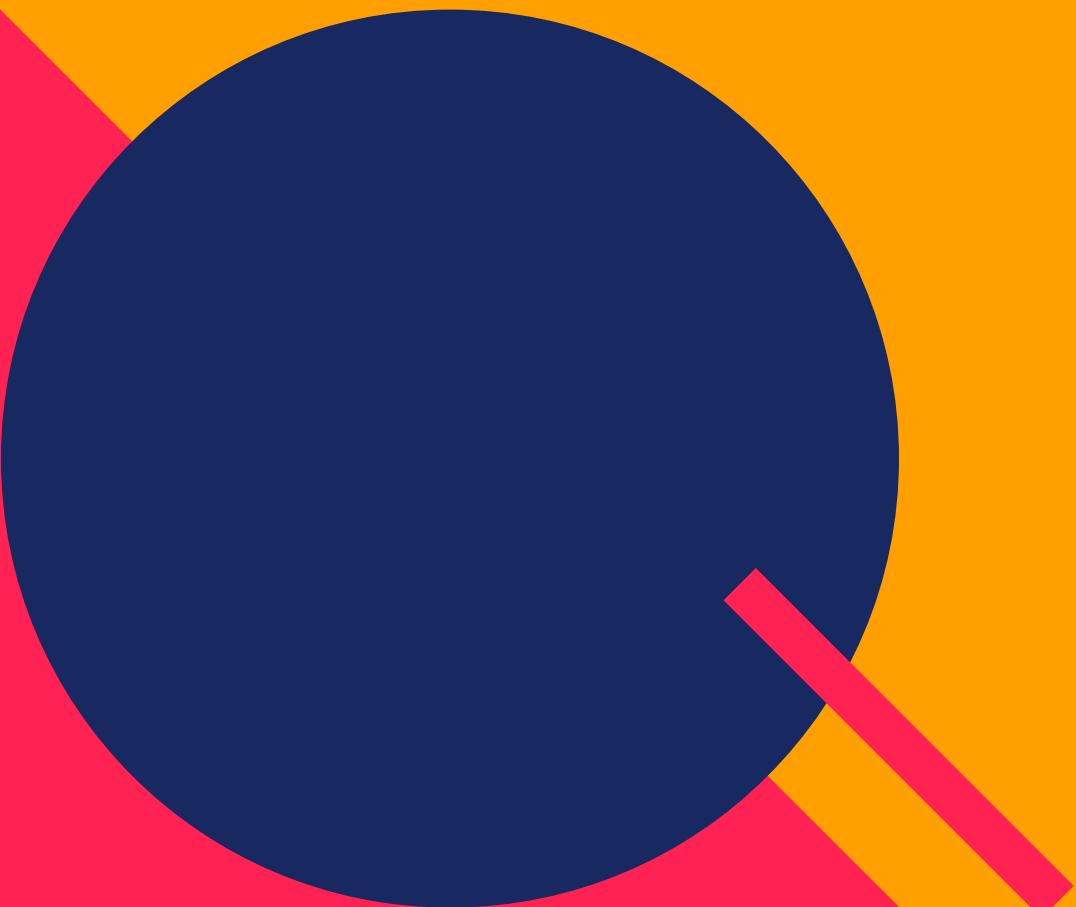
```
[17] df.shape # this data has 36350 rows and 19 column  
(16749, 19)  
[18] print(pd.value_counts(df.Overdue))  
0    14466  
1    1987  
2     154  
5      96  
3      31  
4      15  
Name: Overdue, dtype: int64  
[19] #replace values overdue  
df['Overdue'].mask(df['Overdue'] == '0', 'good score', inplace=True)  
df['Overdue'].mask(df['Overdue'] == '1', 'bad score', inplace=True)  
df['Overdue'].mask(df['Overdue'] == '2', 'bad score', inplace=True)  
df['Overdue'].mask(df['Overdue'] == '3', 'bad score', inplace=True)  
df['Overdue'].mask(df['Overdue'] == '4', 'bad score', inplace=True)  
df['Overdue'].mask(df['Overdue'] == '5', 'bad score', inplace=True)  
[20] print(pd.value_counts(df.Overdue))  
good score    14466  
bad score     2283  
Name: Overdue, dtype: int64
```

The right notebook contains the following code:

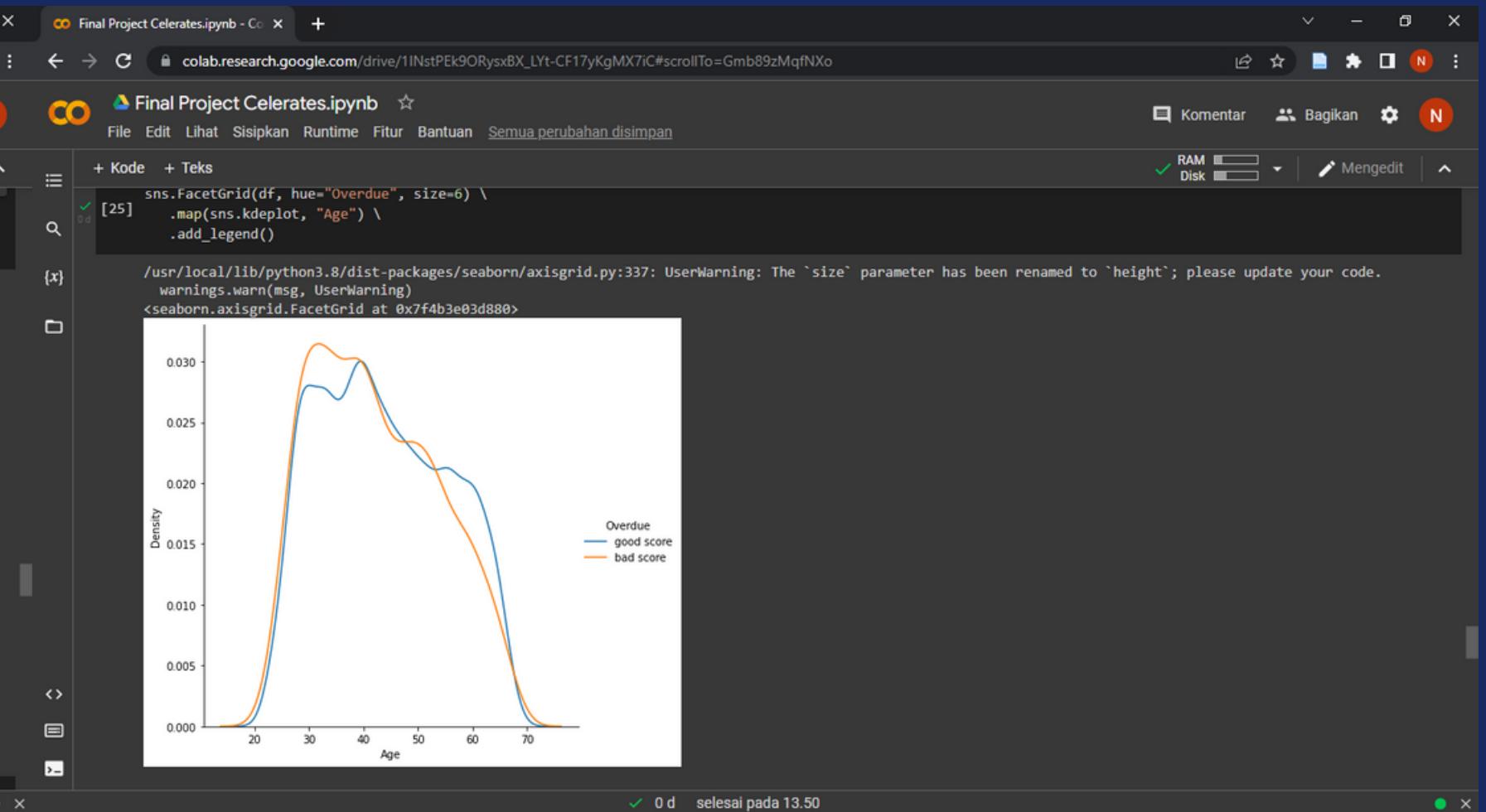
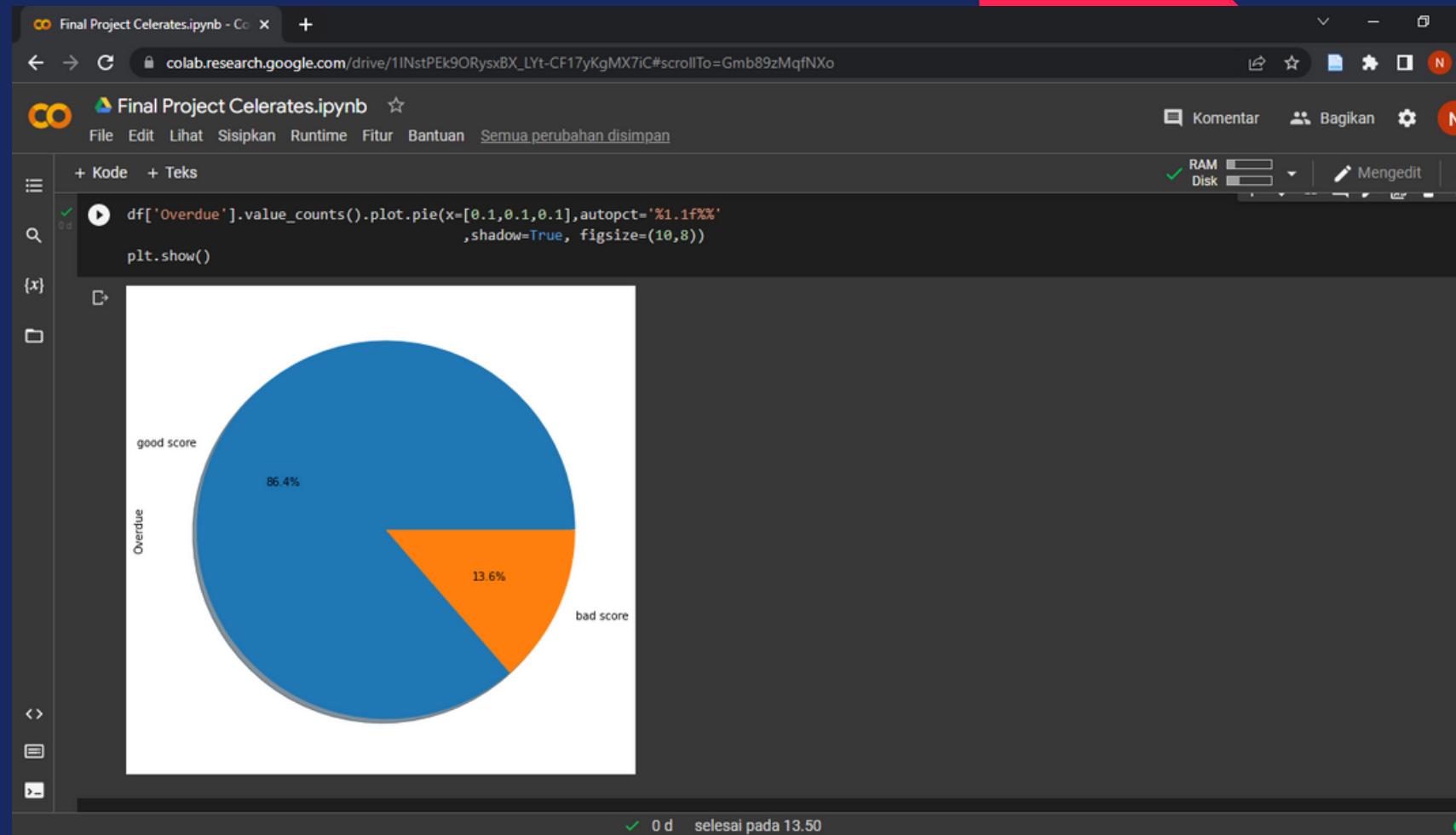
```
[21] 438455    5009319 Perempuan    Ya    Tidak    0    135000    Pensioner    G    M    MH  
16749 rows x 19 columns  
# re-check null values  
df.isnull().sum()  
Id_customer    0  
JK    0  
KepemilikanMobil    0  
KepemilikanProperti    0  
JmlAnak    0  
Pendapatan    0  
TipePendapatan    0  
TingkatPendidikan    0  
StatusKeluarga    0  
TipeRumah    0  
FlagMobile    0  
FlagWorkPhone    0  
FlagPhone    0  
Email    0  
Pekerjaan    0  
JmlAnggotaKeluarga    0  
Age    0  
Experience    0  
Overdue    0  
dtype: int64
```

Manipulate data values in overdue and
check if there is null data before
exploration

FIRST TOPIC PART 2



Exploring



Final Project Celerates.ipynb

```
/usr/local/lib/python3.8/dist-packages/pandas/io/formats/style.py:2813: RuntimeWarning: All-NaN slice encountered
  smin = np.nanmin(gmap) if vmin is None else vmin
/usr/local/lib/python3.8/dist-packages/pandas/io/formats/style.py:2814: RuntimeWarning: All-NaN slice encountered
  smax = np.nanmax(gmap) if vmax is None else vmax
```

	Id_customer	JmlAnak	FlagMobile	FlagWorkPhone	FlagPhone	Email	JmlAnggotaKeluarga	Age	Experience
Id_customer	1.00	0.03	nan	0.08	0.01	-0.05	0.02	-0.05	-0.05
JmlAnak	0.03	1.00	nan	0.07	-0.01	0.02	0.89	-0.33	-0.23
FlagMobile	nan	nan	nan	nan	nan	nan	nan	nan	nan
FlagWorkPhone	0.08	0.07	nan	1.00	0.32	-0.03	0.07	-0.18	-0.24
FlagPhone	0.01	-0.01	nan	0.32	1.00	0.02	-0.01	0.02	-0.01
Email	-0.05	0.02	nan	-0.03	0.02	1.00	0.02	-0.11	-0.08
JmlAnggotaKeluarga	0.02	0.89	nan	0.07	-0.01	0.02	1.00	-0.29	-0.22
Age	-0.05	-0.33	nan	-0.18	0.02	-0.11	-0.29	1.00	0.63
Experience	-0.05	-0.23	nan	-0.24	-0.01	-0.08	-0.22	0.63	1.00

► Imbalance Data Check and Handling

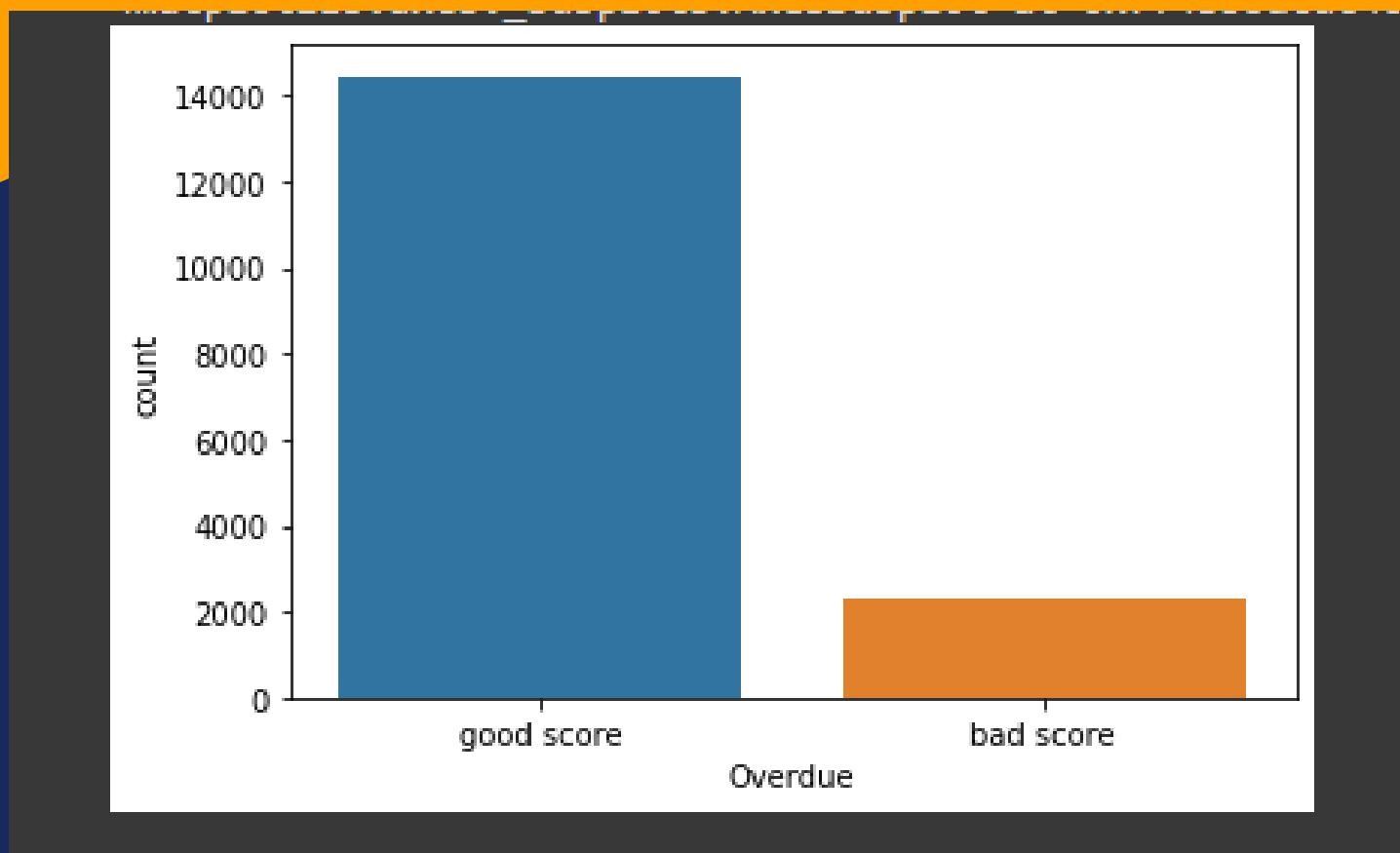
[] 415 sel tersembunyi

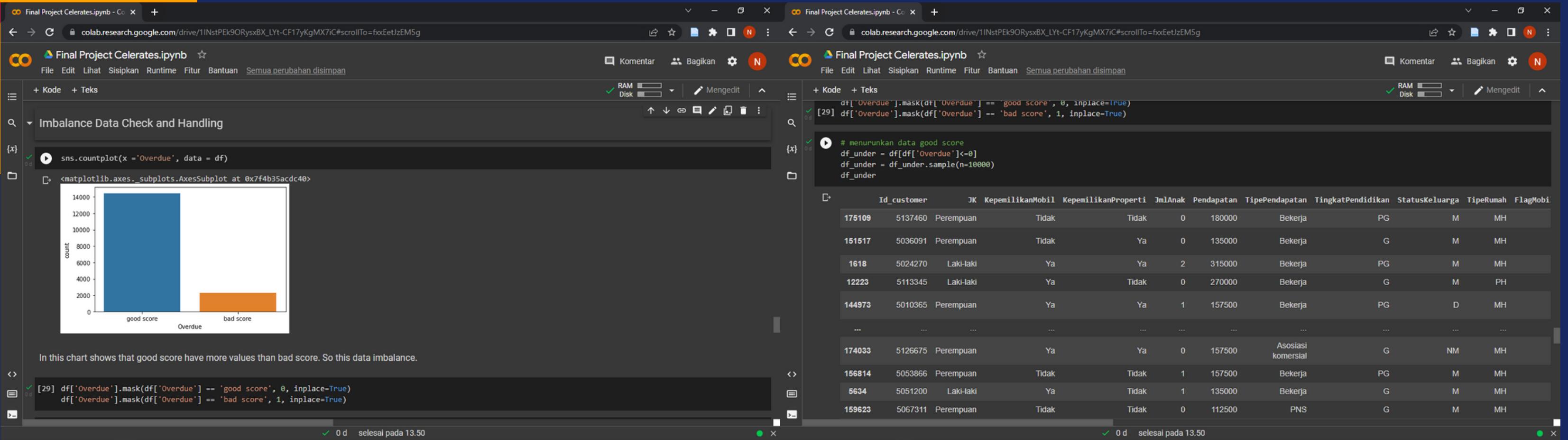
► Modelling

FIRST TOPIC PART 3



Imbalance data Cheking and Handling





there is imbalance data in the overdue field and undersampling techniques are carried out

The image shows two side-by-side Google Colab notebook interfaces. Both notebooks have the title "Final Project Celerates.ipynb". The left notebook contains Python code for concatenating data frames and performing encoding on categorical variables like gender (JK) and property ownership (KepemilikanProperti). The right notebook shows the resulting encoded data frame with numerical values for these variables. Both notebooks display a table of 10k customer records with columns including Id_customer, JK, KepemilikanMobil, KepemilikanProperti, JmlAnak, Pendapatan, TipePendapatan, TingkatPendidikan, StatusKeluarga, TipeRumah, FlagMobil, and FlagKw.

```

df = pd.concat([df_under, df[df['Overdue']>=1]], ignore_index=True)
df

[ ] #encoding data
df['JK'] = df['JK'].map({'Perempuan': 1, 'Laki-laki': 2})
df['KepemilikanMobil'] = df['KepemilikanMobil'].map({'Ya': 1, 'Tidak': 0})
df['KepemilikanProperti'] = df['KepemilikanProperti'].map({'Ya': 1, 'Tidak': 0})
df['TipePendapatan'] = df['TipePendapatan'].map({'Mahasiswa/Murid': 1, 'Pensioner': 2, 'Bekerja': 3, 'PNS': 4, 'Asosiasi komersial': 5})
df['TingkatPendidikan'] = df['TingkatPendidikan'].map({'UG': 1, 'PG': 2, 'G': 3})
df['StatusKeluarga'] = df['StatusKeluarga'].map({'NM': 1, 'M': 2, 'D': 3})
df['TipeRumah'] = df['TipeRumah'].map({'PH': 1, 'MH': 2, 'RA': 3, 'OA': 4, 'MA': 5})
df['Pekerjaan'] = df['Pekerjaan'].map({'Waiters/barmen staff': 1, 'Cooking staff': 2, 'Security staff': 3, 'Private service staff': 4, 'Realty agents': 5, 'Secretarial staff': 6, 'Other staff': 7, 'Sales staff': 8, 'Administrative staff': 9, 'Technician staff': 10, 'Driver staff': 11, 'Other workers': 12, 'Non workers': 13})
df

```

	Id_customer	JK	KepemilikanMobil	KepemilikanProperti	JmlAnak	Pendapatan	TipePendapatan	TingkatPendidikan	StatusKeluarga	TipeRumah	FlagMobil	FlagKw
0	5137460	Perempuan	Tidak	Tidak	0	180000	Bekerja	PG	M	MH		
1	5036091	Perempuan	Tidak	Ya	0	135000	Bekerja	G	M	MH		
2	5024270	Laki-laki	Ya	Ya	2	315000	Bekerja	PG	M	MH		
3	5113345	Laki-laki	Ya	Tidak	0	270000	Bekerja	G	M	PH		
4	5010365	Perempuan	Ya	Ya	1	157500	Bekerja	PG	D	MH		
...		
12278	5149056	Perempuan	Tidak	Ya	0	112500	Asosiasi komersial	G	M	MH		
12279	5149834	Perempuan	Tidak	Ya	0	157500	Asosiasi komersial	PG	M	MH		
12280	5149838	Perempuan	Tidak	Ya	0	157500	Pensioner	PG	M	MH		
12281	5150049	Perempuan	Tidak	Ya	0	283500	Bekerja	G	M	MH		
12282	5009264	Perempuan	Tidak	Tidak	0	40500	Pensioner	G	NM	MH		

Concat 10k data with credit good scoring (0) with bad scoring (1) and encode the data before building the model

Handling oversampling with SMOTE technique.

The image shows two side-by-side Jupyter Notebook environments. Both notebooks have the title "Final Project Celerates.ipynb".

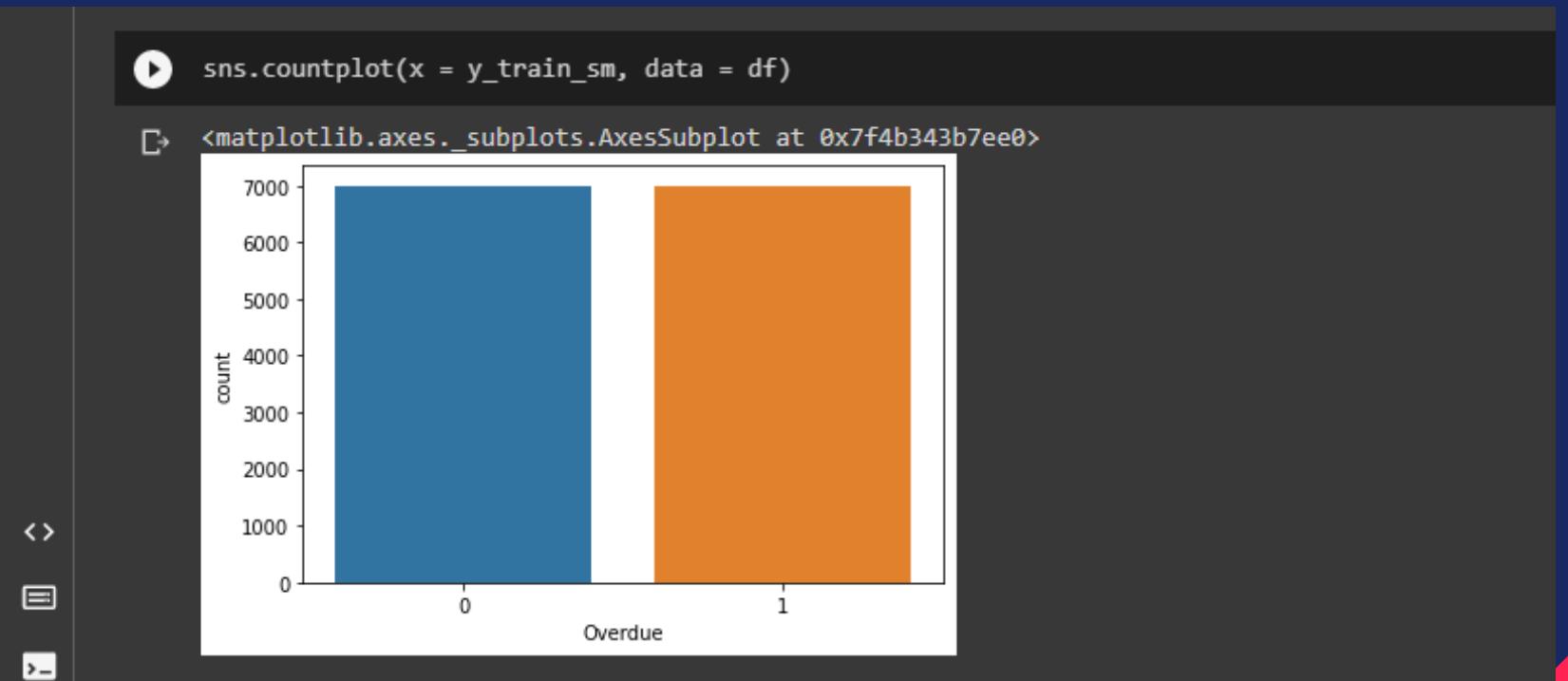
Left Notebook Content:

```
[ ] #change , to . cause make error on handling imbalance data  
df['Pendapatan'] = df['Pendapatan'].astype(str)  
df['Pendapatan'] = df['Pendapatan'].str.replace(',', '.')  
  
[ ] #change data type  
df['Pendapatan'] = df['Pendapatan'].astype(float)  
  
[ ] #split label to x & y  
y = df['Overdue']  
x = df.drop(['Overdue', 'Id_customer'], axis = 1)  
y = y.astype(int)  
  
[ ] #split data to data train and test  
from sklearn.model_selection import train_test_split  
X_train, X_test, y_train, y_test = train_test_split(x, y, test_size = 0.30 ,random_state = 42, stratify=y)  
  
[ ] #handling imbalance data  
from imblearn.over_sampling import RandomOverSampler  
from imblearn.under_sampling import RandomUnderSampler  
from imblearn.over_sampling import SMOTE  
  
[ ] smt = SMOTE(random_state=12)  
X_train_sm, y_train_sm = smt.fit_resample(X_train, y_train)
```

Right Notebook Content:

```
[ ] smt = SMOTE(random_state=12)  
X_train_sm, y_train_sm = smt.fit_resample(X_train, y_train)  
print(X_train_sm.shape)  
print(y_train_sm.shape)  
  
(14000, 17)  
(14000,)  
  
[ ] y_train_sm.value_counts()  
  
0    7000  
1    7000  
Name: Overdue, dtype: int64  
  
[ ] y_test.value_counts()  
  
0    3000  
1    685  
Name: Overdue, dtype: int64  
  
[ ] sns.countplot(x = y_train_sm, data = df)  
[ ] <matplotlib.axes._subplots.AxesSubplot at 0x7f4b343b7ee0>
```

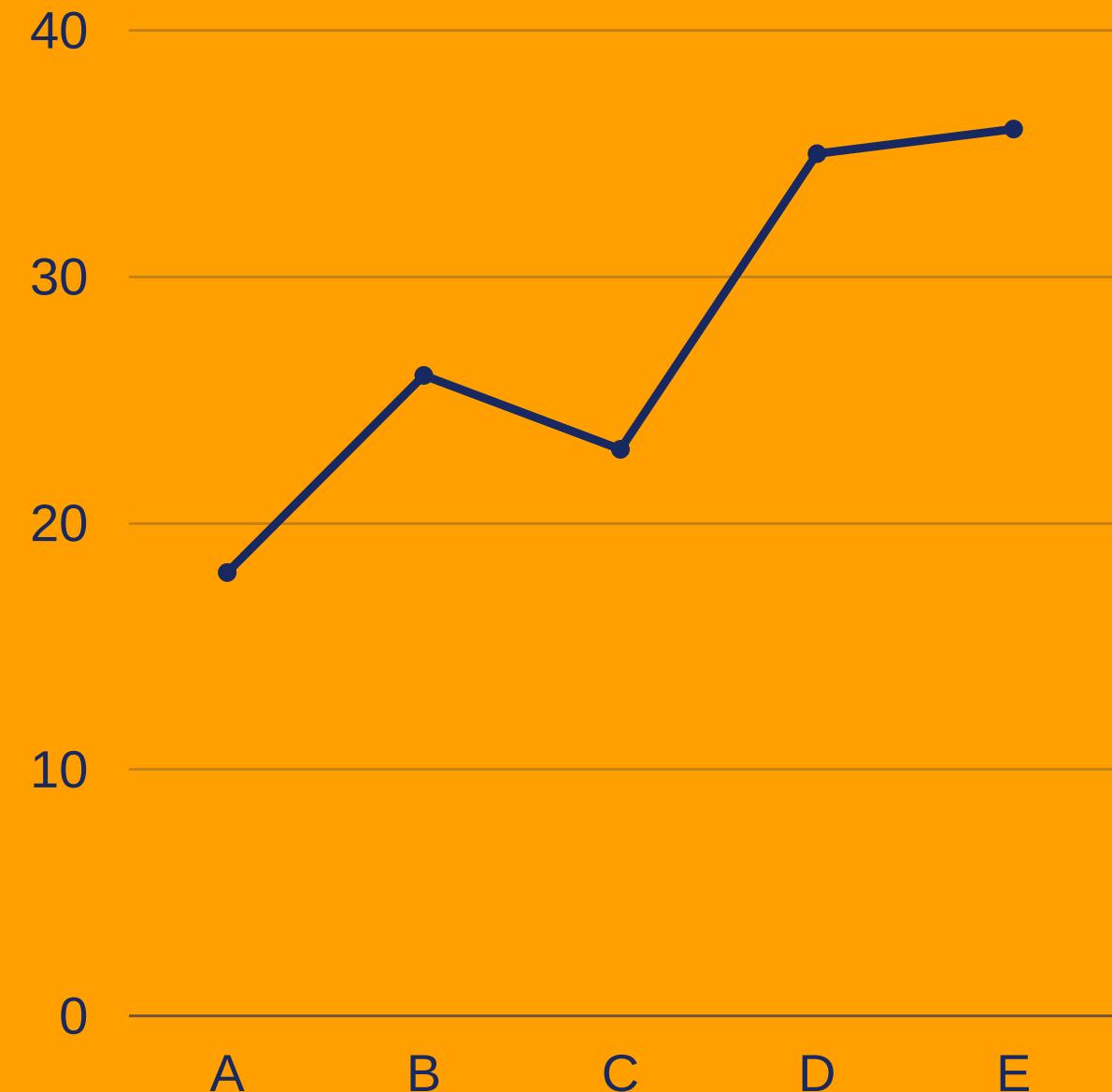
A histogram plot titled "sns.countplot(x = y_train_sm, data = df)" is displayed in the right notebook's output area. The x-axis is labeled "Overdue" and has two categories: 0 and 1. The y-axis is labeled "count" and ranges from 0 to 7000. The bar for category 0 is blue and reaches approximately 7000. The bar for category 1 is orange and reaches approximately 685.



MODELING

- Decision Tree
- SVM
- Naive Bayes
- Random Forest
- Neural Network

IN THE FIVE MODELING



Decision Tree

```
[ ] matrix = confusion_matrix(y_test, prediksi)
matrix
[ ] array([[2583,  417],
       [ 343,  342]])

[ ] accuracy = accuracy_score(y_test, prediksi)
accuracy
0.7937584803256446

[ ] print(classification_report(y_test, prediksi))

          precision    recall  f1-score   support
          0         0.88      0.86      0.87     3000
          1         0.45      0.50      0.47      685
accuracy                           0.79      3685
macro avg       0.67      0.68      0.67     3685
weighted avg    0.80      0.79      0.80     3685
```

SVM

```
[ ] confusion_matrix(y_test, prediksi1)
array([[2542,  458],
       [ 344,  341]])

[ ] accuracy_score(y_test, prediksi1)
0.7823609226594301

[ ] print(classification_report(y_test, prediksi1))

          precision    recall  f1-score   support
          0         0.88      0.85      0.86     3000
          1         0.43      0.50      0.46      685
accuracy                           0.78      3685
macro avg       0.65      0.67      0.66     3685
weighted avg    0.80      0.78      0.79     3685
```

Naive Bayes

```
[ ] confusion_matrix(y_test, prediksi2)
array([[ 726, 2274],
       [ 136,  549]])

[ ] accuracy_score(y_test, prediksi2)
0.34599728629579374

[ ] print(classification_report(y_test, prediksi2))

precision    recall  f1-score   support
          0       0.84      0.24      0.38     3000
          1       0.19      0.80      0.31      685
accuracy                           0.35     3685
macro avg       0.52      0.52      0.34     3685
weighted avg    0.72      0.35      0.36     3685
```

Random Forest

```
▶ prediksi3 = model_3.predict(X_test)
print(prediksi3.shape)
print(y_test.shape)
(3685,)
(3685,)

[ ] confusion_matrix(y_test, prediksi3)
array([[2653,  347],
       [ 351,  334]])

[ ] accuracy_score(y_test, prediksi3)
0.8105834464043419

[ ] print(classification_report(y_test, prediksi3))

precision    recall  f1-score   support
          0       0.88      0.88      0.88     3000
          1       0.49      0.49      0.49      685
accuracy                           0.81     3685
macro avg       0.69      0.69      0.69     3685
weighted avg    0.81      0.81      0.81     3685
```

Final Project Celerates.ipynb - Colab

colab.research.google.com/drive/1InstPEk9ORysxBX_LYt-CF17yKgMX7iC#scrollTo=mGwnsVVd-c5E

Final Project Celerates.ipynb ☆

File Edit Lihat Sisipkan Runtime Fitur Bantuan Semua perubahan disimpan

+ Kode + Teks

Sambungkan kembali Mengedit

Neural Network

```
[ ] from keras.optimizers import Adam
from keras.layers import Dense
from keras.models import Sequential

[ ] model_4 = Sequential()
model_4.add(Dense(32, input_shape=(17,)))
model_4.add(Dense(32, activation="relu"))
model_4.add(Dense(1, activation='sigmoid'))
model_4.compile(
    optimizer='adam',
    loss='binary_crossentropy',
    metrics=['accuracy'])

[ ] model_4.fit(X_train_sm, y_train_sm, epochs = 50)

438/438 [=====] - 1s 3ms/step - loss: 110.1879 - accuracy: 0.5005
Epoch 23/50
438/438 [=====] - 1s 3ms/step - loss: 93.5884 - accuracy: 0.5086
Epoch 24/50
438/438 [=====] - 1s 3ms/step - loss: 95.0481 - accuracy: 0.5122
Epoch 25/50
438/438 [=====] - 1s 3ms/step - loss: 94.2029 - accuracy: 0.5072
Epoch 26/50
```

0 d selesai pada 13.50

Final Project Celerates.ipynb ☆

File Edit Lihat Sisipkan Runtime Fitur Bantuan Semua perubahan disimpan

+ Kode + Teks

```
<keras.callbacks.History at 0x7f4a7a149910>
```

y_pred = model_4.predict(X_test)
import numpy as np
y_pred = np.where(y_pred >= 0.95*y_pred.max(), 1, 0)
print(classification_report(y_test, y_pred))

116/116 [=====] - 0s 1ms/step

	precision	recall	f1-score	support
0	0.81	1.00	0.90	3000
1	0.24	0.01	0.01	685
accuracy			0.81	3685
macro avg	0.52	0.50	0.45	3685
weighted avg	0.71	0.81	0.73	3685

Data Test Prediction



The image shows two side-by-side Google Colab notebooks. Both notebooks have the title "Final Project Celerates.ipynb".

Notebook 1 (Left):

- Code cell 1:

```
dft = pd.read_csv('/content/drive/MyDrive/Celerates_BDBI/Data Set final project/data_test.csv')
dft.head()
```
- Output cell 1: A table showing the first 5 rows of the dataset. The columns are: Id_customer, JK, KepemilikanMobil, KepemilikanProperti, JmlAnak, Pendapatan, TipePendapatan, TingkatPendidikan, StatusKeluarga, TipeRumah, FlagMobile, F1.
- Code cell 2:

```
[ ] modus = dft['Pekerjaan'].mode()[0]
dft['Pekerjaan'] = dft['Pekerjaan'].fillna(modus)
dft['Pekerjaan'].isna().sum()
```

Notebook 2 (Right):

- Code cell 1:

```
[ ] modus = dft['Pekerjaan'].mode()[0]
dft['Pekerjaan'] = dft['Pekerjaan'].fillna(modus)
dft['Pekerjaan'].isna().sum()
```
- Output cell 1:

```
[ ] dft['Pekerjaan'].value_counts()
```

Pekerjaan	Count
Laborers	93
Core staff	16
Managers	16
Sales staff	14
High skill tech staff	13
Drivers	12
Medicine staff	10
Cooking staff	7
Cleaning staff	5
Accountants	5
Security staff	4
Private service staff	3
Waiters/barmen staff	2
- Code cell 2:

```
[ ] dft['TingkatPendidikan'] = dft['TingkatPendidikan'].str.lower()
dft['StatusKeluarga'] = dft['StatusKeluarga'].str.lower()
dft['FlagMobile'] = dft['FlagMobile'].str.lower()
```

Read data and fill null
data on pekerjaan

Final Project Celerates.ipynb - Colab

File Edit Lihat Sisipkan Runtime Fitur Bantuan Semua perubahan disimpan Komentar Bagikan Mengedit

```
[ ] dft['TingkatPendidikan'] = dft['TingkatPendidikan'].str.lower()
dft['StatusKeluarga'] = dft['StatusKeluarga'].str.lower()
dft['TipeRumah'] = dft['TipeRumah'].str.lower()

[ ] dft['JK'] = dft['JK'].map({'Perempuan': 1, 'Laki-laki': 2})
dft['KepemilikanMobil'] = dft['KepemilikanMobil'].map({'Ya': 1, 'Tidak': 0})
dft['KepemilikanProperti'] = dft['KepemilikanProperti'].map({'Ya': 1, 'Tidak': 0})
dft['TipePendapatan'] = dft['TipePendapatan'].map({'Mahasiswa/Murid': 1, 'Pensioner': 2, 'Bekerja': 3, 'PNS': 4, 'Asosiasi komersial': 5})
dft['TingkatPendidikan'] = dft['TingkatPendidikan'].map({'undergraduate': 1, 'postgraduate': 2, 'graduate': 3})
dft['StatusKeluarga'] = dft['StatusKeluarga'].map({'belum menikah': 1, 'menikah': 2, 'cerai': 3})
dft['TipeRumah'] = dft['TipeRumah'].map({'rumah orang tua': 1, 'rumah pribadi': 2, 'sewa apartemen': 3, 'apartemen kantor': 4, 'apartemen pribadi': 5})
dft['Pekerjaan'] = dft['Pekerjaan'].map({'Waiters/barmen staff': 1, 'Cooking staff': 2, 'Security staff': 3, 'Private service staff': 4, 'Realty agents': 5, 'Other': 6})
dft
```

	ID_Customer	JK	KepemilikanMobil	KepemilikanProperti	JmlAnak	Pendapatan	TipePendapatan	TingkatPendidikan	StatusKeluarga	TipeRumah	FlagMobile	Flag	
0	5142248	1		0		1	0	225000.0		4	3	2	2
1	5036925	1		1		1	0	157500.0		5	3	2	2
2	5126080	1		0		1	1	112500.0		4	3	2	2
3	5088887	1		0		1	0	171000.0		3	3	1	3
4	5022156	1		1		1	2	180000.0		5	2	2	2
...
195	5105368	2		0		1	0	360000.0		3	3	2	2
196	5116026	1		0		1	0	135000.0		2	3	2	2

Encode the data test and predict with model_3 (Random Forest) and export the result.

Credit Risk Analysis

Annual income

26100

Income category

(All)

Gender

(All)

Credit Status

(All)

D
Q

Total Client Number

438,310

Avg Annual Income

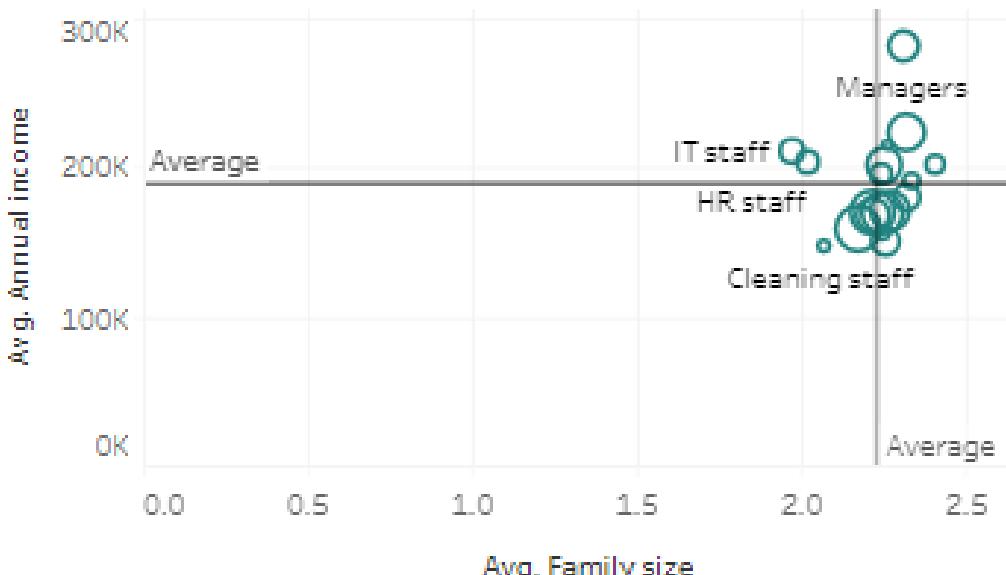
187,492

Avg Family Size

2.194

Reset Filters

Click Here!



Regulatory Understanding

In granting credit, the systems and procedures for giving attention are not only considered, but also the management of credit. Education has a significant effect on how customers understand the contents of existing regulations.

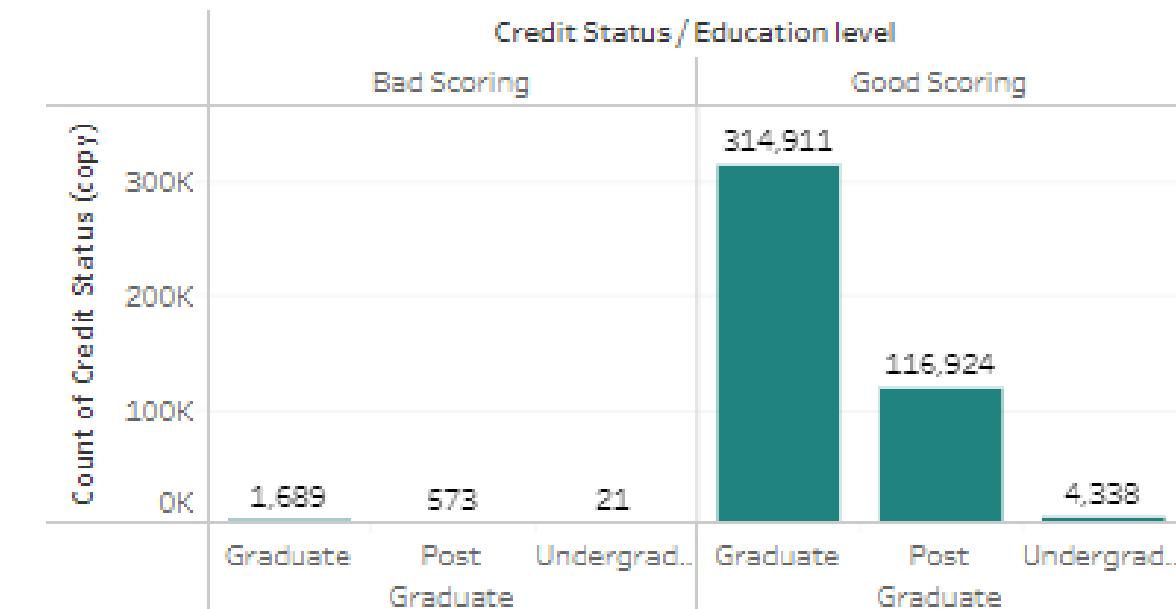
So before giving credit, an understanding is made in advance so that credit does not experience problems from either the bank or the customer.

Good Scoring
Perempuan
Married

Good Scoring
Laki-laki
Married

Problem Credit

Providing working credit capital is a bank business with a high risk. The risk in credit is the occurrence of credit problems so the background of the customer needs to be considered, such as average income and number of families. Which is a factor in whether the customer can return the credit or not.



Track Record

The provision of existing credit is also one of the considerations of whether the customer can repay the credit (good score) or not (bad score). Men and women have their own needs and status and also have relationships that can affect responsibility. The principles that must be applied in granting credit are the 5 C principles, namely character, capacity, capital, economic conditions, and collateral.

**FOR INQUIRIES
AND CONCERN'S**

NEXT DASHBOARD ON THIS LINK
https://public.tableau.com/views/FinalProjectCelerates_167204243_00480/Dashboard1?:language=en-US&:display_count=n&:origin=viz_share_link

Thank You..

