

# MODELLING CLIENTS' PAYMENT DIFFICULTIES

Muhammad Ananda Taj Bara

Virtual Internship Experience Data Scientist at Home Credit  
Indonesia by Rakamin Academy Batch 25

# BUSINESS QUESTION

01

BERAPAKA PERSEN  
PELANGGAN YANG  
MENGALAMI KESULITAN  
PEMBAYARAN PADA HOME  
CREDIT INDONESIA ?

02

KRITERIA APA SAJA YANG  
MENGALAMI KESULITAN  
PEMBAYARAN PADA PELANGGAN  
DENGAN PINJAMAN PADA SAAT  
DI TERIMA OLEH HOME CREDIT  
INDONESIA ?

# BUSINESS QUESTION

01

MENGIDENTIFIKASI  
PENYEBAB PELANGGAN  
KESULITAN DALAM  
PEMBAYARAN PADA HOME  
CREDIT INDONESIA

02

MEMBUAT MACHINE LEARNING  
UNTUK MENGIDENTIFIKASI  
PADA PELANGGAN YANG  
MENGALAMI KESULITAN  
PEMBAYARAN PADA HOME  
CREDIT INDONESIA

# DATA PREPROCESSING

## ❖ Check & Handling Missing Values

```
✓ 5d # drop column missing values > 50% rows
for col in df:
    if df[col].isna().sum() > (df.shape[0])/2:
        df = df.drop(col, 1)

<ipython-input-304-c8b6dd3ec0d8>:4: FutureWarning: In a future version of pandas all arguments of DataFrame.drop except for the argument 'labels' will be keyword-only
df = df.drop(col, 1)
```

```
✓ [311] # handle missing values categorical columns with mode
0d for col in df[catcol]:
    df[col].fillna(df[col].mode()[0], inplace=True)
```

```
✓ 4d [313] # handle missing values numerical column with median
for col in df[numcol2]:
    df[col].fillna(df[col].median(), inplace=True)
```

# DATA PREPROCESSING

## ❖ Check & Handling Outliers

```
✓ [315] # handle outlier using z-score
1d
    from scipy import stats
    print(f'jumlah baris sebelum di filter : {len(df)}')

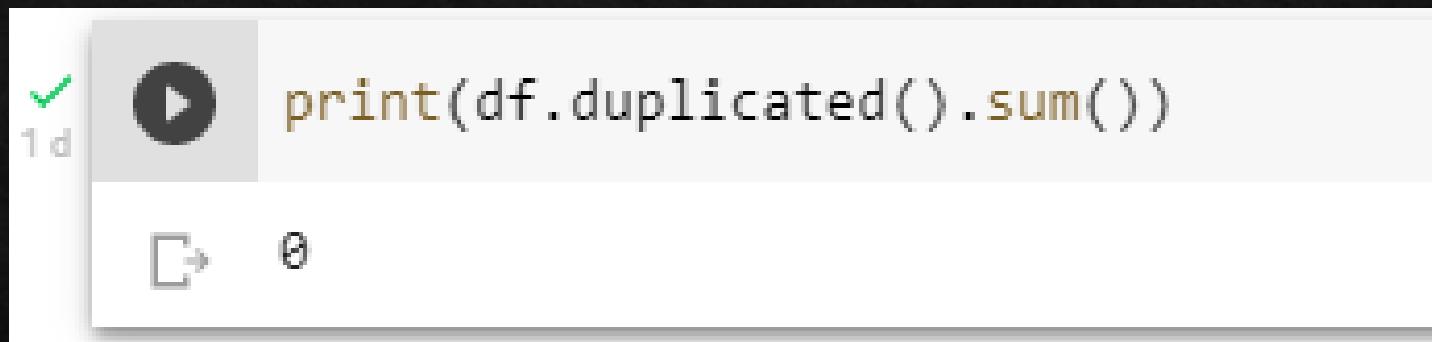
    filtered_entries = np.array([True]*len(df))
    for col in numcol2:
        zscore = abs(stats.zscore(df[col]))
        filtered_entries = (zscore < 3) & filtered_entries
    df_filtered = df[filtered_entries]

    print(f"jumlah baris setelah di filter: {len(df_filtered)})"

jumlah baris sebelum di filter : 307511
jumlah baris setelah di filter: 248189
```

# DATA PREPROCESSING

- ❖ Check & Handling Duplicated Values



A screenshot of a Jupyter Notebook cell. The cell contains the following code:

```
print(df.duplicated().sum())
```

The cell has a green checkmark icon and a play button icon. The output area shows the result of the code execution:

```
0
```

# FEATURE ENGINEERING

## ❖ Feature Scaling

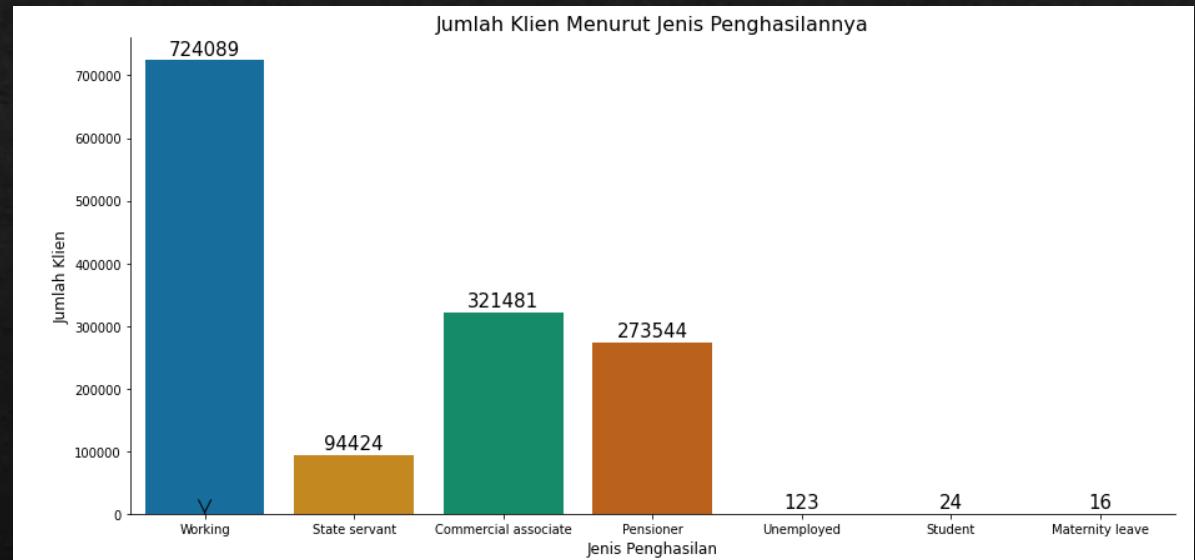
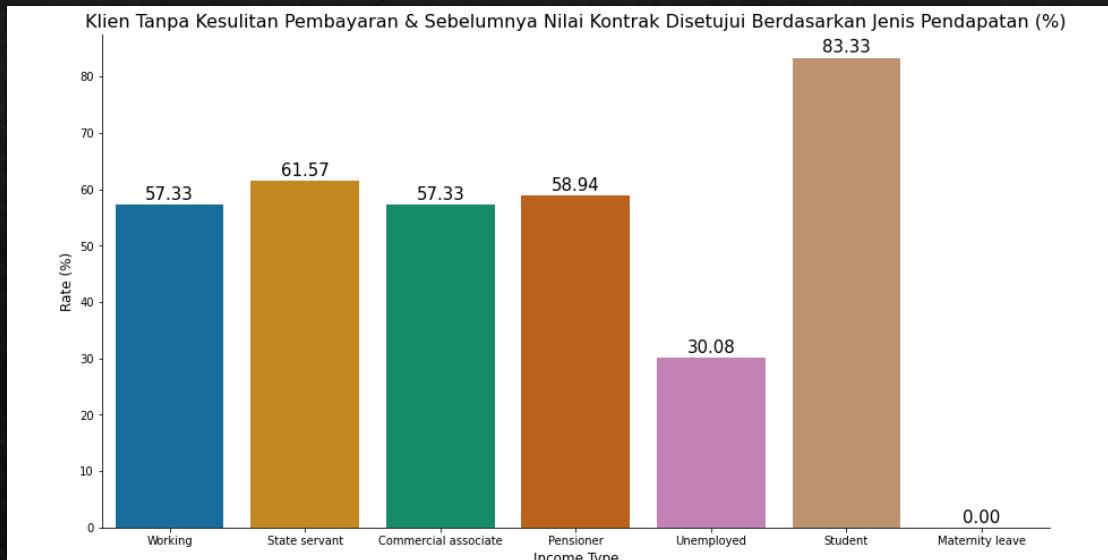
```
✓ [39] # scaling for numerical feature
      from sklearn.preprocessing import MinMaxScaler, StandardScaler
      for col in numcol2:
          df_filtered[col] = MinMaxScaler().fit_transform(df_filtered[col].values.reshape(len(df_filtered), 1))
```

# FEATURE ENGINEERING

## ❖ Feature Encoding

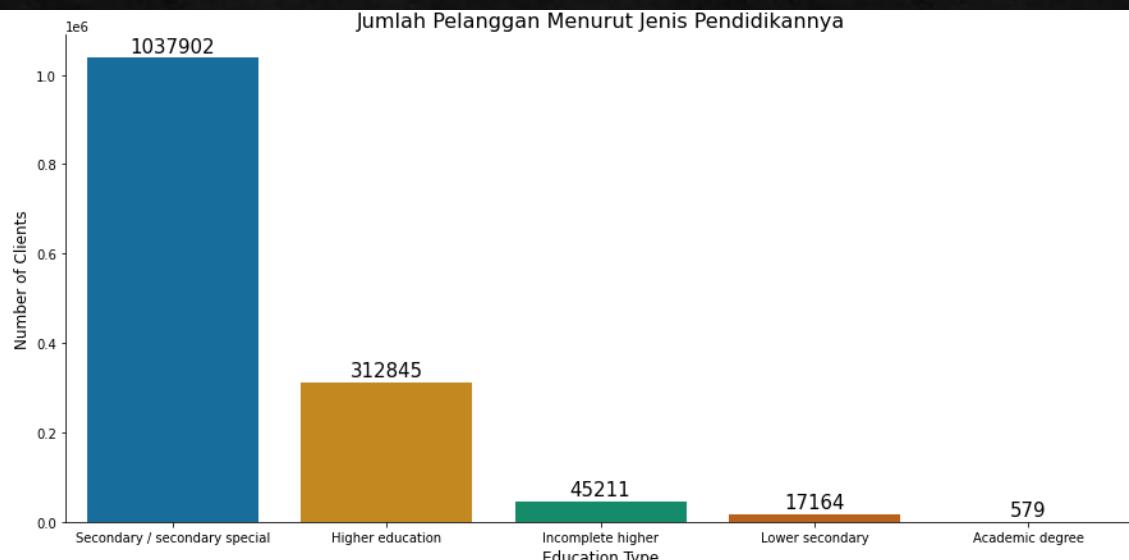
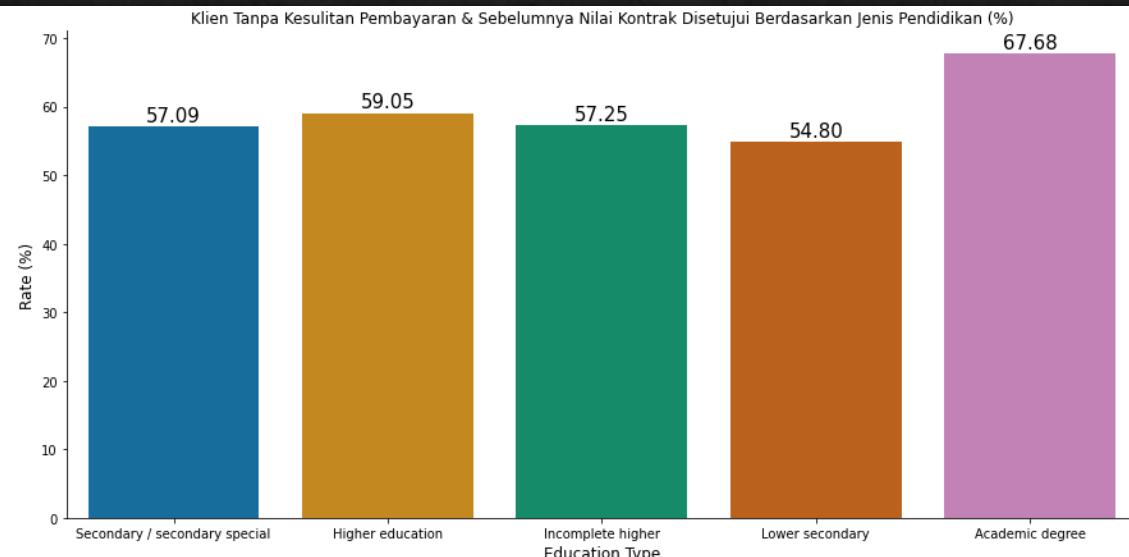
```
✓ 0 d   # encoding for categorical feature
    for col in catcol2:
        onehots = pd.get_dummies(df_filtered[col], prefix=col)
        df_filtered2 = df_filtered.join(onehots)
    df_filtered2
```

# DATA VISUALIZATION & INSIGHT



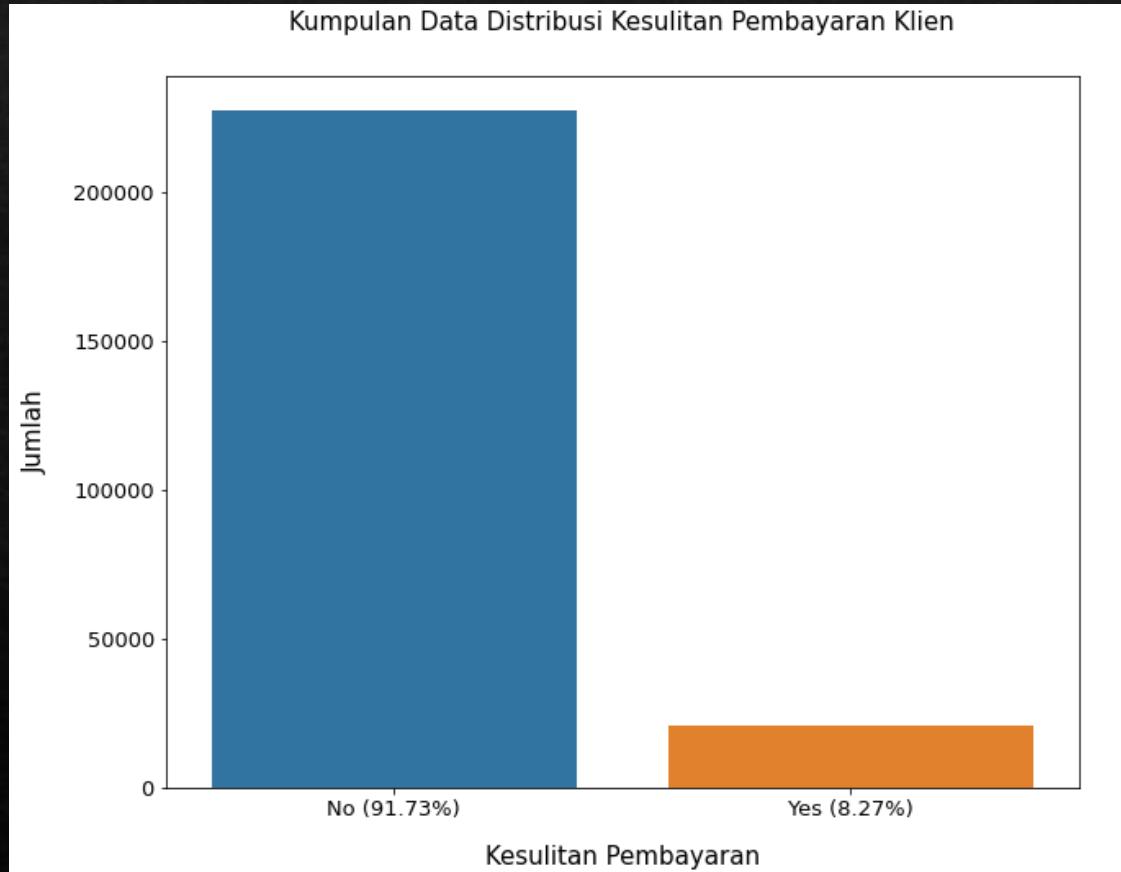
Berdasarkan diagram diatas, pelanggan berstatus **Student** memiliki rate tertinggi sebesar 83.3% untuk pelanggan yang pinjaman sebelumnya diterima & tidak mengalami kesulitan pembayaran. Namun, total dari pelanggan berstatus **student** mengajukan pinjaman hanya 24 orang atau sekitar 0.0017% dari total pelanggan yang mengajukan pinjamanan sehingga harus dilakukan promosi dan kampanye kepada pelanggan yang berstatus **Student** agar semakin banyak yang tertarik untuk mengajukan pinjaman di Home Credit Indonesia.

# DATA VISUALIZATION & INSIGHT



Berdasarkan diagram, Pelanggan dengan status akademik adalah yang tertinggi 67,68% untuk nasabah kredit diperoleh sebelumnya dan tidak berpengalaman kesulitan pembayaran Namun, berbagai pelanggan dengan status akademik hanya 579 orang atau sekitar 0,04% dari seluruh nasabah yang mengajukan pinjaman. Sehingga perlu promosi atau kampanye untuk klien dengan gelar akademik lainnya agar semakin banyak orang yang tertarik Ajukan pinjaman Home Credit Indonesia

# CHECK & HANDLING IMBALANCED CLASS



```
[47] # handling using SMOTE
from imblearn.over_sampling import SMOTE
oversample = SMOTE()
x_smote, y_smote = oversample.fit_resample(x, y)
y_smote.value_counts()

TARGET
0      227672
1      227672
dtype: int64
```

```
[42] yes = df_filtered2['TARGET'].value_counts()[1]
no = df_filtered2['TARGET'].value_counts()[0]
yes_per = yes / df_filtered2.shape[0] * 100
no_per = no / df_filtered2.shape[0] * 100

print('{} dari {} dengan kesulitan pembayaran dan itu adalah {:.2f}% dari dataset.'.format(yes, df_filtered2.shape[0], yes_per))
print('{} dari {} tanpa kesulitan pembayaran dan itu adalah {:.2f}% dari dataset.'.format(no, df_filtered2.shape[0], no_per))

20517 dari 248189 dengan kesulitan pembayaran dan itu adalah 8.27% dari dataset.
227672 dari 248189 tanpa kesulitan pembayaran dan itu adalah 91.73% dari dataset.
```

# Data Modelling & Evaluation

Pada hasil dari evaluasi (atas) model dari Random Forest memiliki nilai Accuracy, Recall, & AUC yang lebih tinggi dari model lainnya. Namun, Namun, pada hasil evaluasi overfitting (bawah) model Random Forest mengalami overfitting, yakni score data trainingnya lebih besar daripada score data testingnya. Maka dari itu Namun, pada hasil evaluasi overfitting (bawah) model Random Forest mengalami overfitting, yakni score data trainingnya lebih besar daripada score data testingnya.

```
[95] dfeval = pd.DataFrame({'Model':['Logistic Regression', 'Random Forest', 'XGBoost'],
                           'Accuracy' : [0.8629, 0.9521, 0.9331],
                           'Recall' : [0.9059, 0.9863, 0.8799],
                           'AUC' : [0.86, 0.95, 0.93]},)
dfeval
```

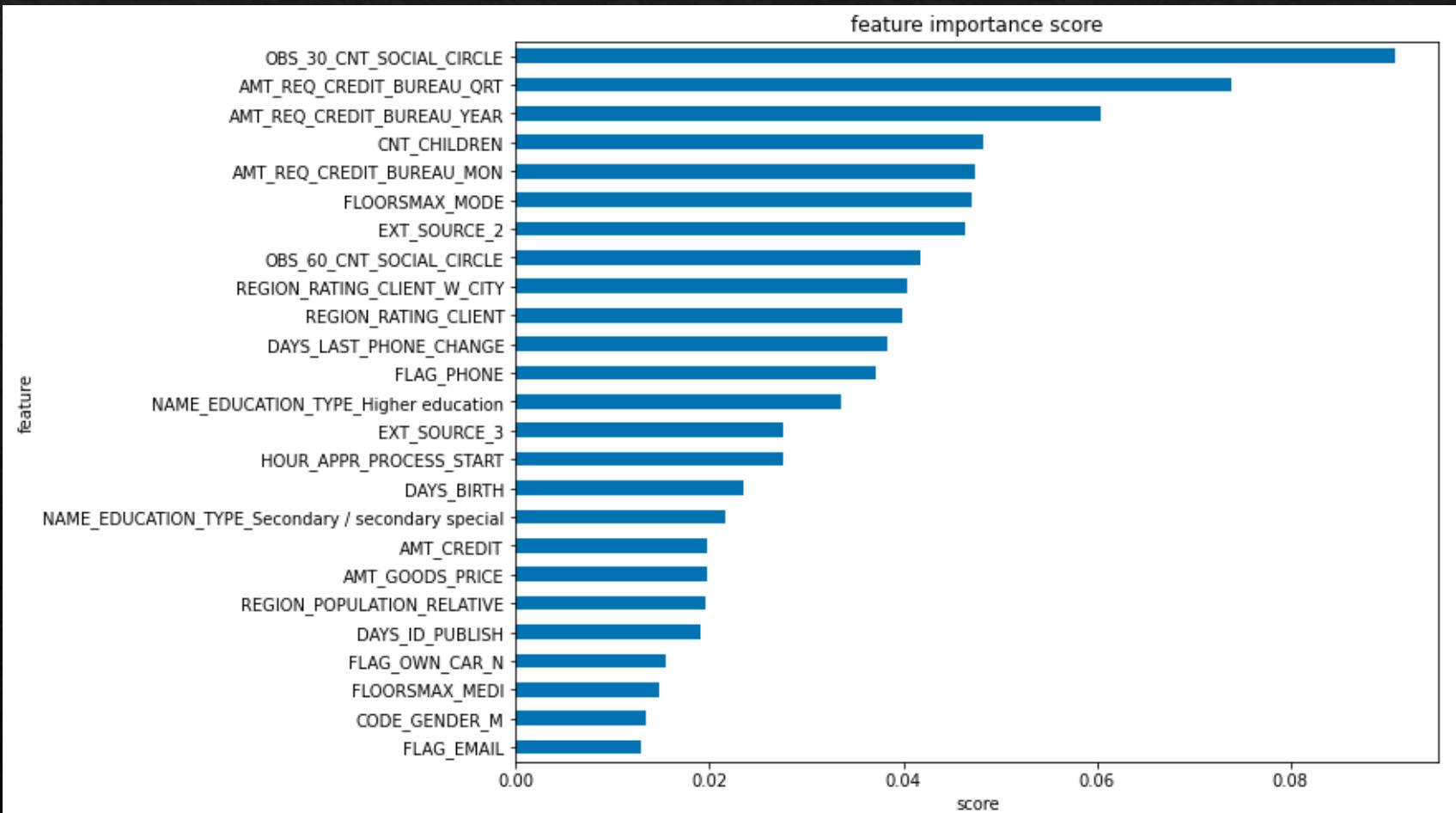
	Model	Accuracy	Recall	AUC
0	Logistic Regression	0.8629	0.9059	0.86
1	Random Forest	0.9521	0.9863	0.95
2	XGBoost	0.9331	0.8799	0.93

```
[96] dfoverfit = pd.DataFrame({'Model':['Logistic Regression', 'Random Forest', 'XGBoost'],
                               'Training Score' : [0.8609, 1.0000, 0.9326],
                               'Test Score' : [0.8629, 0.9521, 0.9331]})
```

	Model	Training Score	Test Score
0	Logistic Regression	0.8609	0.8629
1	Random Forest	1.0000	0.9521
2	XGBoost	0.9326	0.9331

# Data Modelling & Evaluation



# SUMMARY & RECOMENDATION

- ❖ 8.27% client mengalami kesulitan pembayaran di Home Credit Indonesia
- ❖ Student Client memiliki rate tertinggi sebesar 83.3% untuk client yang pinjaman sebelumnya diterima & tidak mengalami kesulitan pembayaran. Namun, total Student Client yang mengajukan pinjaman hanya 24 orang atau sekitar 0.0017% dari total client yang mengajukan pinjaman. Sehingga diperlukan promosi atau campaign kepada Student Client lainnya agar semakin banyak yang tertarik untuk mengajukan pinjaman di Home Credit Indonesia.
- ❖ Client dengan gelar akademik memiliki rate tertinggi sebesar 67,68 % untuk client yang pinjaman sebelumnya diterima & tidak mengalami kesulitan pembayaran. Namun, total Client dengan gelar akademik yang mengajukan pinjaman hanya 579 orang atau sekitar 0,04% dari total client yang mengajukan pinjaman. Sehingga diperlukan promosi atau campaign kepada Client dengan gelar akademik lainnya agar semakin banyak yang tertarik untuk mengajukan pinjaman di Home Credit Indonesia.
- ❖ Model terbaik yang dipilih adalah model Logistic Regression dan XGBoost yang memiliki nilai Accuracy, Recall, & AUC cukup baik dan tidak terjadi overfitting.