Making Credit Risk Customer Detection

ID/X Partners

Data Scientist Virtual Internship Program
Andi Eka Nugraha

Created by:
Andi Eka Nugraha
an.ekanugraha@gmail.com
linkedin/andi-eka-nugraha

Bachelor in Physics with major expertise in Instrumentation & Robotics and has attended a Datascience bootcamp for 4 months. Experienced in programming microcontrollers and machine learning to process data or images, as well as creating robotic systems that can support human work. Able to understand business, especially for data analysis, studying statistics and machine learning, as well as the ability to create regression models, classification, and clustering. Skills in identifying and analyzing patterns in data and presenting analytical results well.

Metodology

1. Business & Problem Understanding
2. Data Collection & Preparation
3. Exploratory Data Analysis
4. Feature Enginering
5. Data Preprocessing
6. Modeling
7. Business Recommendation

01

Business & Problem Understanding

Business & Problem Understanding

We as data scientists are asked to create a model that can predict the customer's credit risk to avoid company losses. The dataset consists of various customers who have made loans along with information on the credit conditions of each customer.

Goals

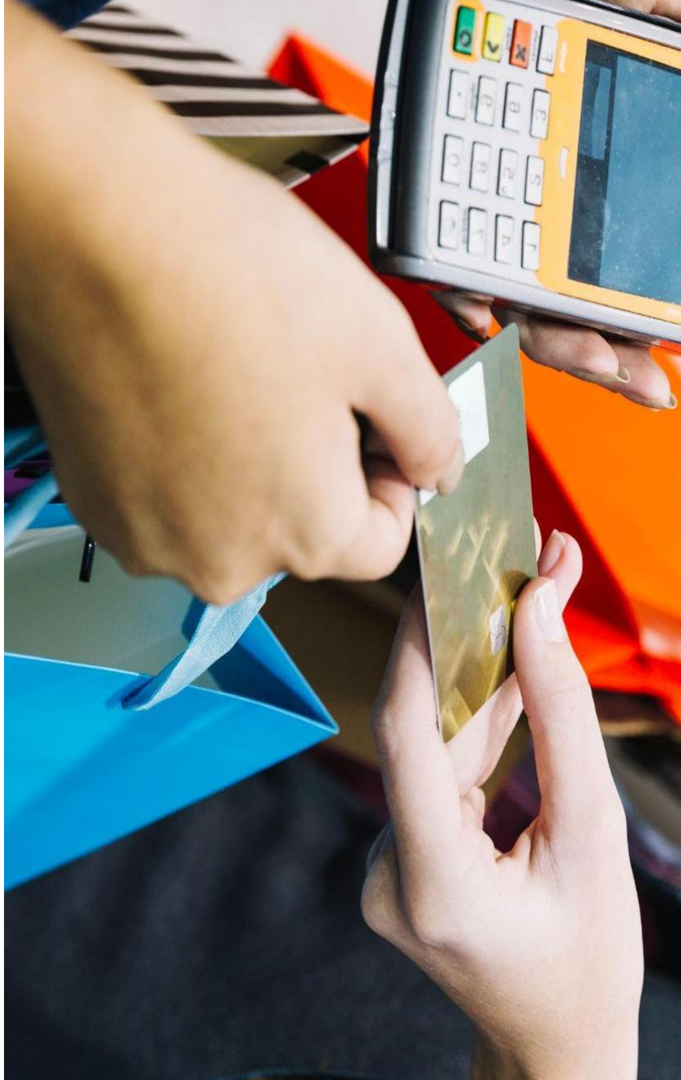
Create a model that can predict the customer's credit risk to avoid company losses

Objective

- Determine the data used for modeling
- Make customer segmentation based on credit risk

Business Metrics

credit risk



02

Data Collection & Preparation

Feature Needed

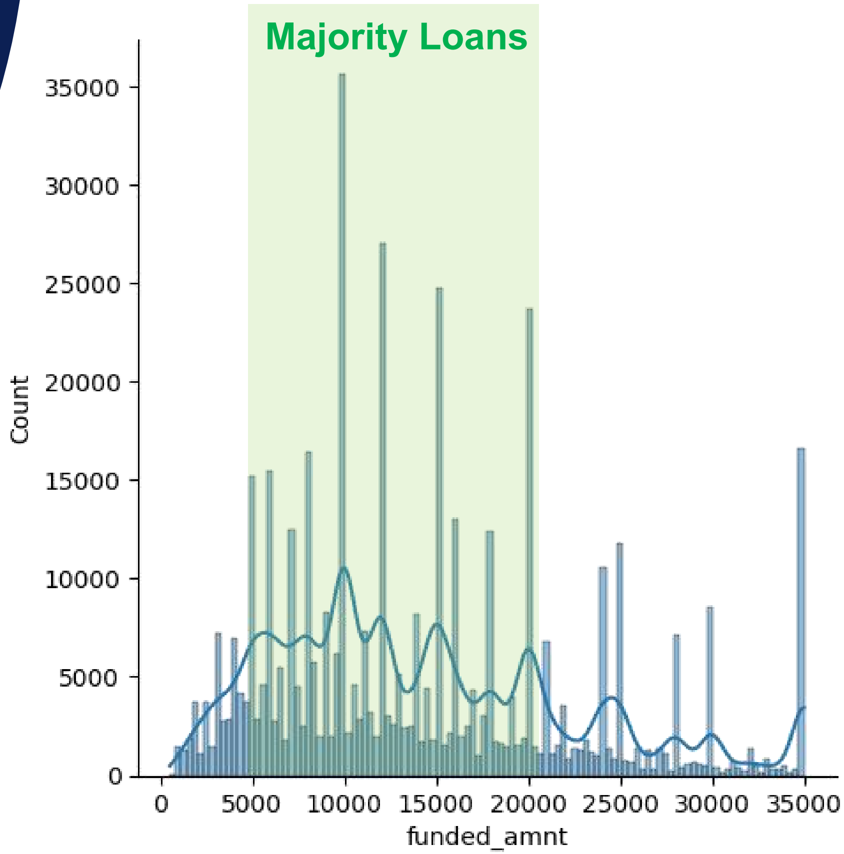
To predict the credit risk model, customer data features are needed at the time of registration and target features which are the customer's lending conditions

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 466285 entries, 0 to 466284
Data columns (total 27 columns):
 #   Column                      Non-Null Count   Dtype
---  ------                      --------------   -----
 0   id                          466285 non-null  int64
 1   member_id                   466285 non-null  int64
 2   acc_now_delinq              466256 non-null  float64
 3   addr_state                  466285 non-null  object
 4   annual_inc                  466281 non-null  float64
 5   application_type            466285 non-null  object
 6   collection_recovery_fee     466285 non-null  float64
 7   collections_12_mths_ex_med  466140 non-null  float64
 8   delinq_2yrs                 466256 non-null  float64
 9   desc                        125983 non-null  object
 10  emp_length                  445277 non-null  object
 11  emp_title                   438697 non-null  object
 12  funded_amnt                 466285 non-null  int64
 13  grade                       466285 non-null  object
 14  sub_grade                   466285 non-null  object
 15  home_ownership              466285 non-null  object
 16  initial_list_status         466285 non-null  object
 17  installment                 466285 non-null  float64
 18  int_rate                    466285 non-null  float64
 19  issue_d                     466285 non-null  object
 20  loan_status                 466285 non-null  object
 21  pub_rec                     466256 non-null  float64
 22  purpose                     466285 non-null  object
 23  term                        466285 non-null  object
 24  title                       466265 non-null  object
 25  url                         466285 non-null  object
 26  zip_code                    466285 non-null  object
dtypes: float64(8), int64(3), object(16)
memory usage: 96.1+ MB
```

For details, see Jupyter Notebook here

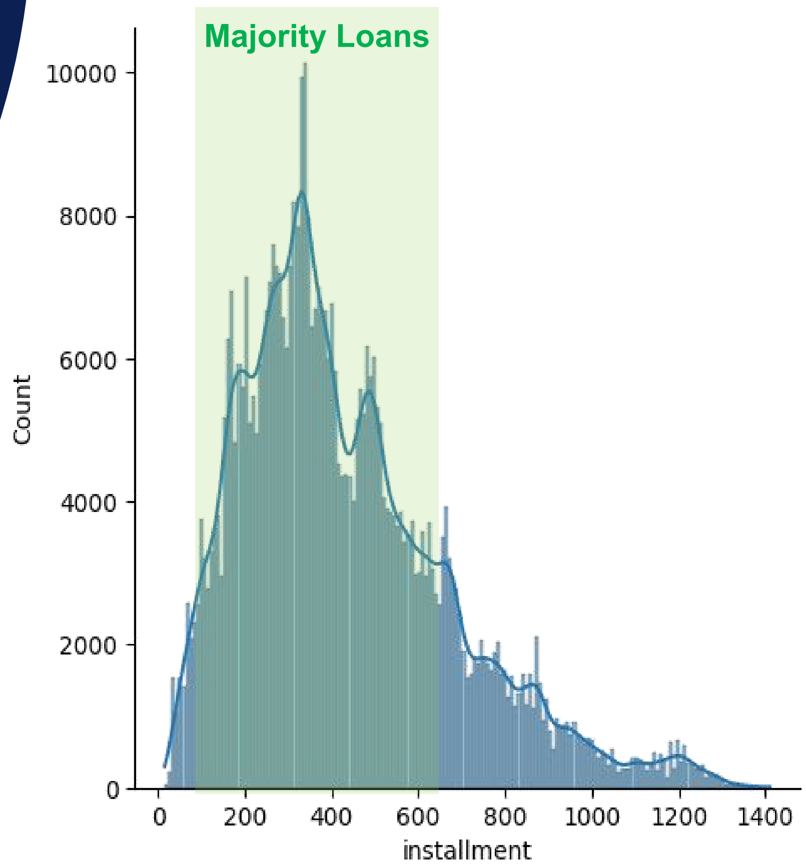
03

Exploratory Data Analysis



The majority of loan values range from 4500 to 20000, this can be analyzed for the type of customer who borrows with that amount and conducts special marketing with certain segments of society to increase sales of lending services

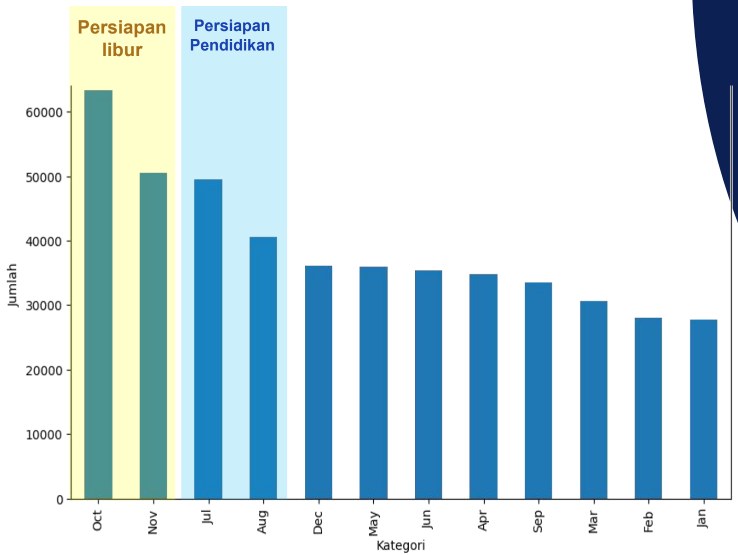
For details, see Jupyter Notebook here

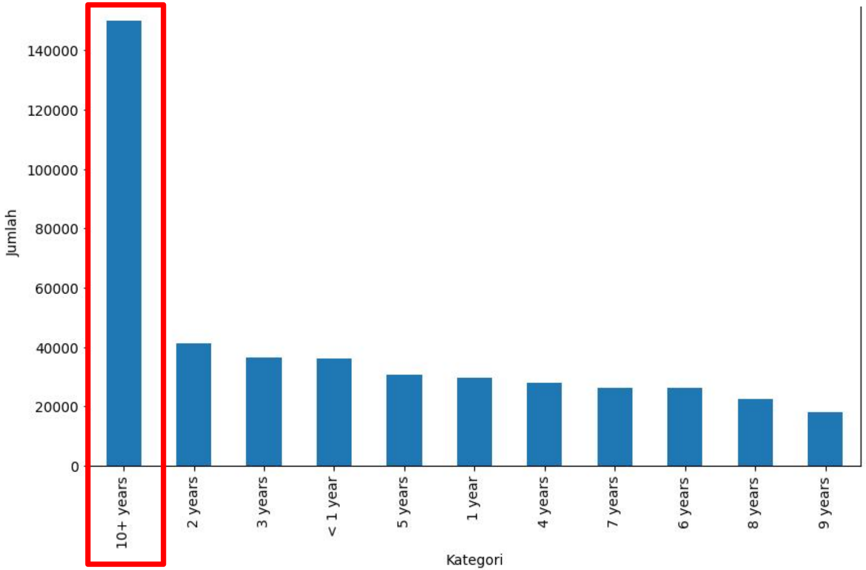


The number of installments that are in demand has a range of 150 to 600. This can be used as a marketing strategy and special offers to increase the number of customers who make loans

For details, see Jupyter Notebook here

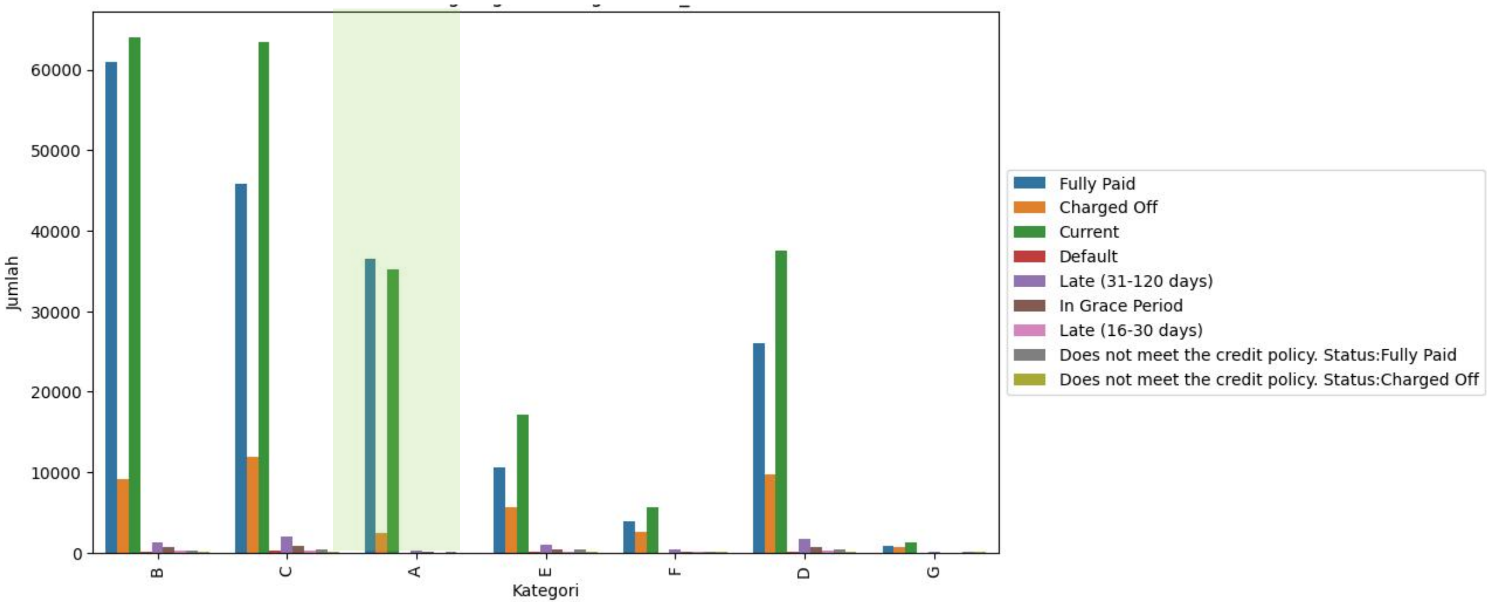
The number of customers making loans is high during **October** and **November** because it is towards the end of the year, and **July** is close to the moment of class promotion and college registration. This moment can be used as a marketing strategy to take advantage of this momentum.





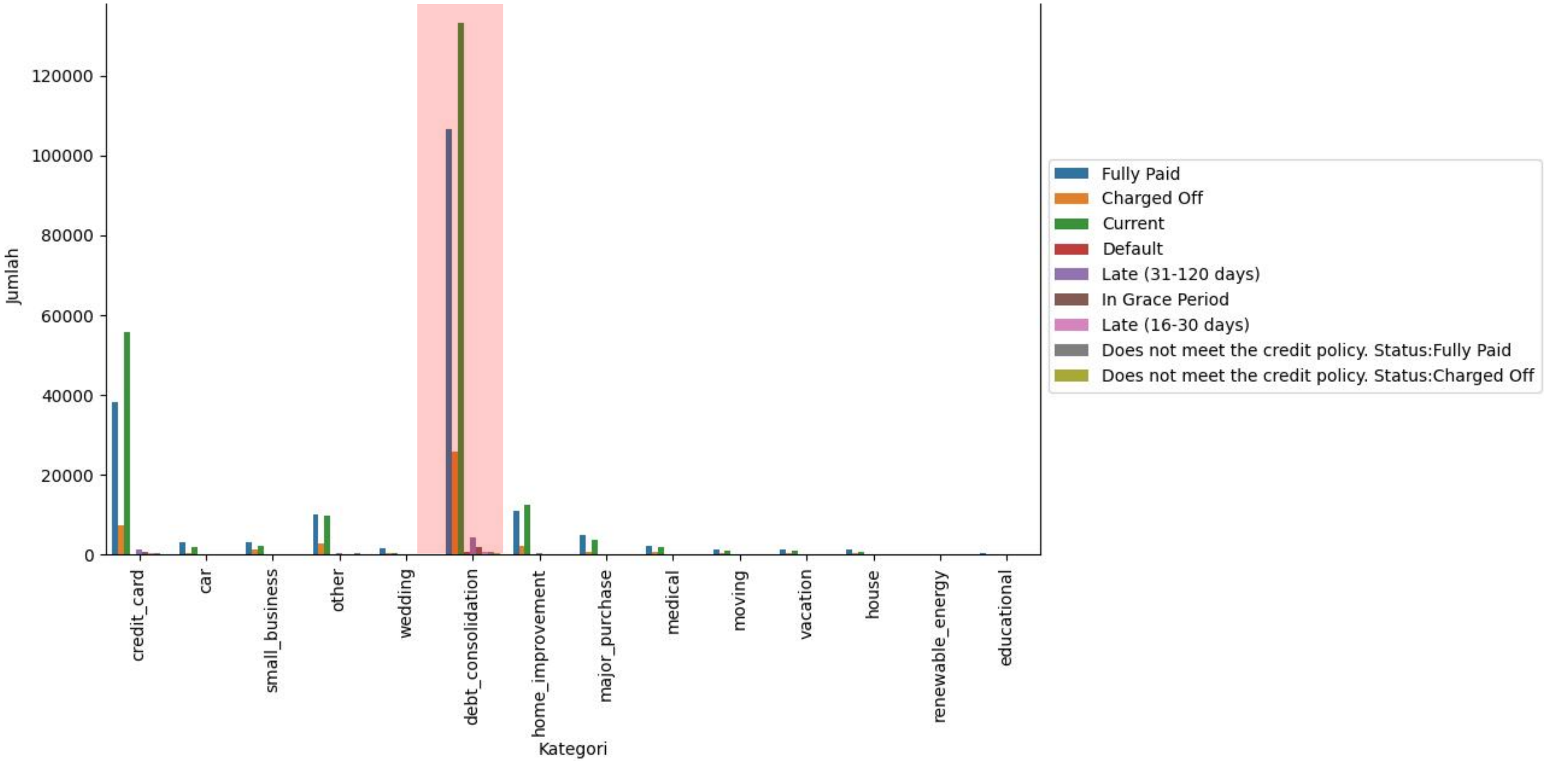
The number of customers who have worked for more than **10 years** is the majority of customers, this is one of the things that can be followed up further to maximize marketing

For details, see Jupyter Notebook here



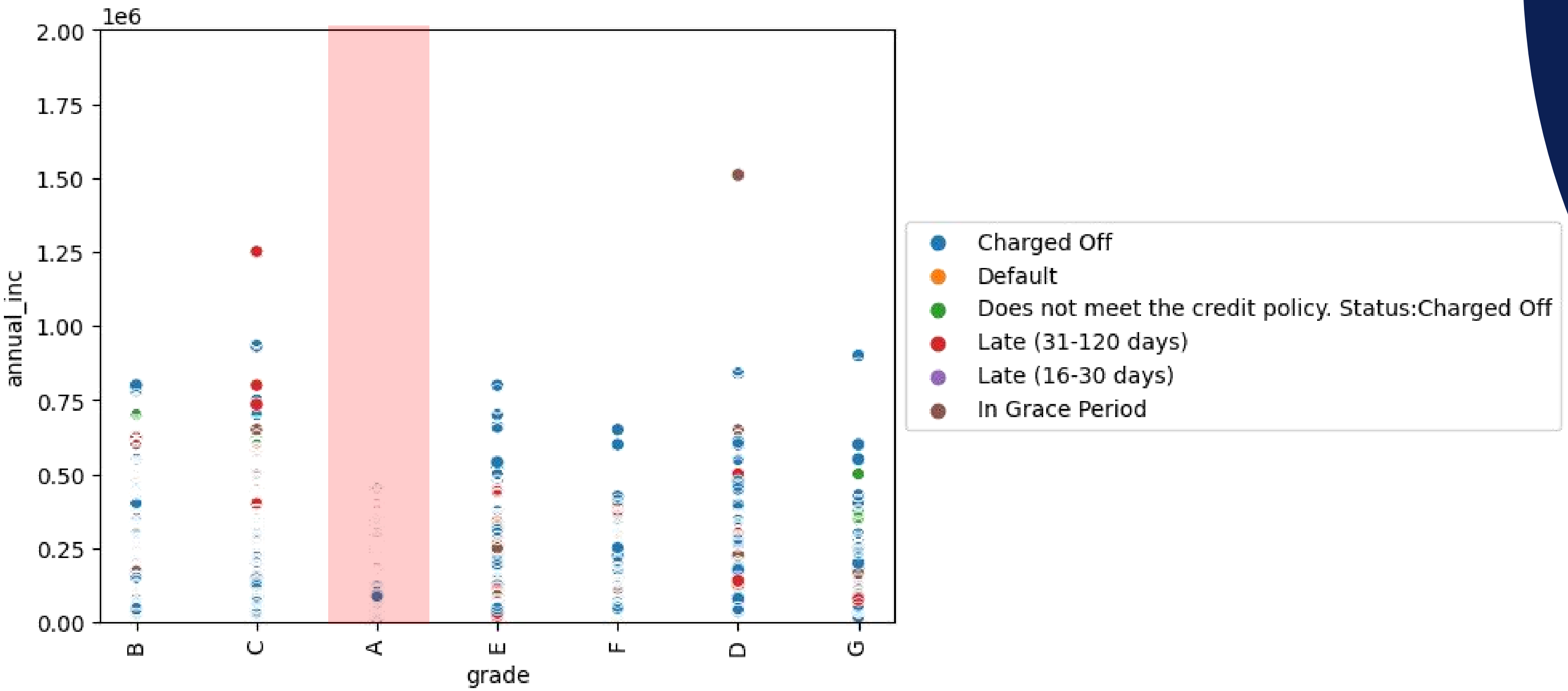
The ratio of borrowers who fail to pay compared to those who do not default has a worrying score in category D. There needs to be more stringent selection for low-grade customers

For details, see Jupyter Notebook here



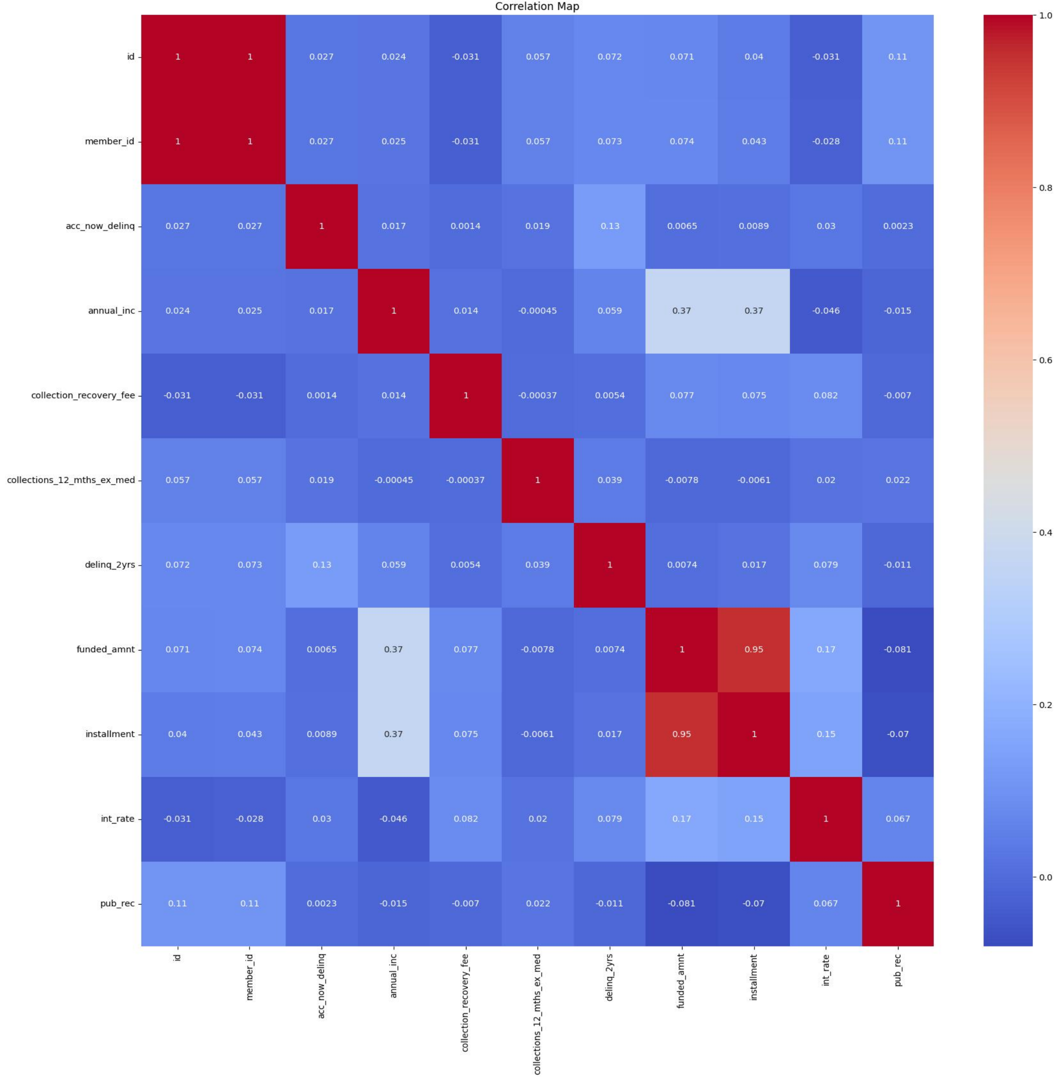
The majority of customers who make loans have the goal of carrying out debt consolidation, this requires a more stringent selection of customers with the aim of debt consolidation to reduce the company's loss ratio

For details, see Jupyter Notebook here



Customers have a tendency to make loans along with increasing income, this is likely to support lifestyles or increasing expenses, customers who have grades lower than A have a high risk of default. A special strategy is needed to reduce this risk.

For details, see Jupyter Notebook here



Redundant values occur in the funded_amnt column with instalments, you should consider dropping one of them

For details, see Jupyter Notebook here

	count	mean	std	min	25%	50%	75%	max
id	466285.0	1.307973e+07	1.089371e+07	54734.00	3639987.00	10107897.00	20731209.00	38098114.00
member_id	466285.0	1.459766e+07	1.168237e+07	70473.00	4379705.00	11941075.00	23001541.00	40860827.00
acc_now_delinq	466256.0	4.002093e-03	6.863680e-02	0.00	0.00	0.00	0.00	5.00
annual_inc	466281.0	7.327738e+04	5.496357e+04	1896.00	45000.00	63000.00	88960.00	7500000.00
collection_recovery_fee	466285.0	8.961534e+00	8.549144e+01	0.00	0.00	0.00	0.00	7002.19
collections_12_mths_ex_med	466140.0	9.085253e-03	1.086484e-01	0.00	0.00	0.00	0.00	20.00
delinq_2yrs	466256.0	2.846784e-01	7.973651e-01	0.00	0.00	0.00	0.00	29.00
funded_amnt	466285.0	1.429180e+04	8.274371e+03	500.00	8000.00	12000.00	20000.00	35000.00
installment	466285.0	4.320612e+02	2.434855e+02	15.67	256.69	379.89	566.58	1409.99
int_rate	466285.0	1.382924e+01	4.357587e+00	5.42	10.99	13.66	16.49	26.06
pub_rec	466256.0	1.605642e-01	5.108626e-01	0.00	0.00	0.00	0.00	63.00

All columns are tail skew to the right of the chart except funded_amnt. skew data with characteristics such as showing the accumulation of data on the left side of the graph and most likely contains outliers

For details, see Jupyter Notebook here

04

Feature Enginering

State Economic Quality Segmentation

```python
def get_quality(state):
    high_quality = ['CA', 'MA', 'WA', 'NY', 'VA']
    medium_quality = ['CO', 'OR', 'MN', 'UT', 'IL', 'WI', 'MD', 'CT', 'NJ']

    if state in high_quality:
        return 'Kualitas Tinggi'
    elif state in medium_quality:
        return 'Kualitas Menengah'
    else:
        return 'Kualitas Rendah'
```

Regional segmentation based on the country's economic class into 3 categories: High Quality, Medium Quality and Low Quality

For details, see Jupyter Notebook here

Loan Segmentation Purposes

```
def segment_loan_purpose(purpose):
    personal_categories = ['credit_card', 'car', 'small_business', 'other']
    major_expense_categories = ['wedding', 'debt_consolidation', 'home_improvement', 'major_purchase']
    special_purpose_categories = ['medical', 'moving', 'vacation', 'house']

    if purpose in personal_categories:
        return 'Personal'
    elif purpose in major_expense_categories:
        return 'Major Expense'
    elif purpose in special_purpose_categories:
        return 'Special Purpose'
    else:
        return 'Other'
```

Segmenting lending objectives into three objectives: Personal, Major Expense, Special Purpose, and Others

For details, see Jupyter Notebook here

Target Segmentation

```
# Daftar status yang akan diubah menjadi 0 (Current dan Fully Paid)
status_to_zero = ['Current', 'Fully Paid','Does not meet the credit policy. Status:Fully Paid']

# Mengganti status menjadi 0 atau 1 berdasarkan kondisi
df['loan_status'] = df['loan_status'].replace(status_to_zero, 0)
df['loan_status'] = df['loan_status'].replace({status: 1 for status in df['loan_status'].unique() if status != 0})
```

Do target binning, bad credit quality is given 1 and good quality is given 0

Handling Date Column

```
df['issue_d']= df['issue_d'].str[:3]
```

take the first 3 letters in the date column to take the month

For details, see Jupyter Notebook here

Drop Column

- Drop unnecessary columns

```
drop = ['addr_state','purpose']
df = df.drop(drop, axis=1)
```

- Drop High Cardinality

```
drop = ['desc','url','emp_title','title','zip_code']
df = df.drop(drop, axis=1)
```

- Drop Low Cardinality

```
drop = ['application_type']
df = df.drop(drop, axis=1)
```

- Drop Null Values

```
df.dropna(subset=['emp_length'],axis=0, inplace=True)
df.dropna(subset=['collections_12_mths_ex_med'],axis=0, inplace=True)
```

- Drop Redundant Columns

```
drop = ['sub_grade','funded_amnt']
df = df.drop(drop, axis=1)
```

- Drop Columns Identity

```
drop = ['member_id','id']
df = df.drop(drop, axis=1)
```

- Drop Duplicate

```
df.drop_duplicates(inplace=True)
```

For details, see Jupyter Notebook here

05

Data Preprocessing

Numeric Transformation

```
data_skew = df.select_dtypes(include='number')

scaler = QuantileTransformer()
df[data_skew.columns] =scaler.fit_transform(df[data_skew.columns])
```

Because the data contains many outlier values, a Quantile Transformer scaler transformation is performed

Label Encoding

```
data = df['emp_length']
df['emp_length'] = label_encoding_with_changes(data)
data = df['grade']
df['grade'] = label_encoding_with_changes(data)
data = df['term']
df['term'] = label_encoding_with_changes(data)
data = df['Kualitas_Negara']
df['Kualitas_Negara'] = label_encoding_with_changes(data)
```

Performs label encoding on column which is character ranking

For details, see Jupyter Notebook here

One Hot Encoding

```
x = x.select_dtypes(include='object')
```

```
df_encoded = pd.get_dummies(x, columns=x.columns)
```

Doing one hot encoding on the category column without ranking

Split Data Set

```
X = dataset.drop('loan_status', axis=1)
y = dataset['loan_status']

# Split the dataset into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)
```

For details, see Jupyter Notebook here

06

Modeling

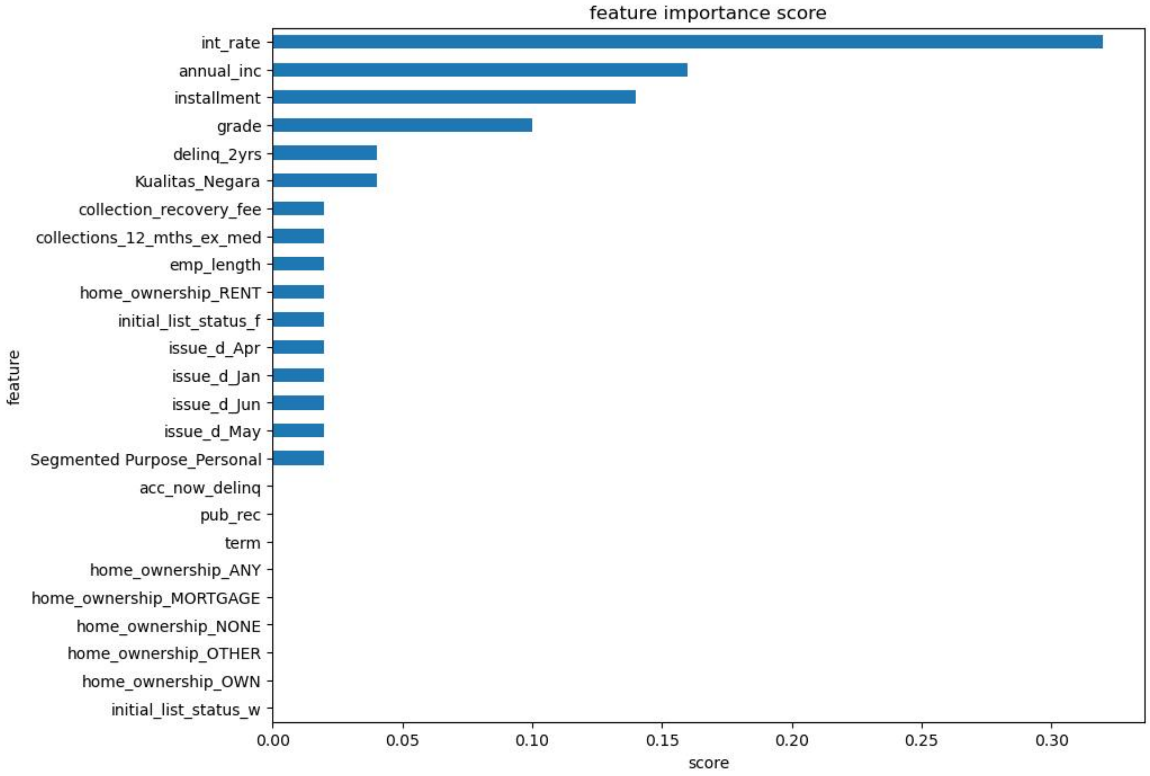
For details, see Jupyter Notebook here

Metric	AdaBoostClassifier	RandomForestClassifier	xgboost
Accuracy	0.93	0.93	0.93
F1-Score	0.59	0.59	0.59
roc_auc (test-proba)	0.80	0.78	0.80
roc_auc (train-proba)	0.81	1.00	0.84

The model used is a model that is resistant to outliers, the evaluation results show that the model used is the Ada Boost Classifier.

For details, see Jupyter Notebook here

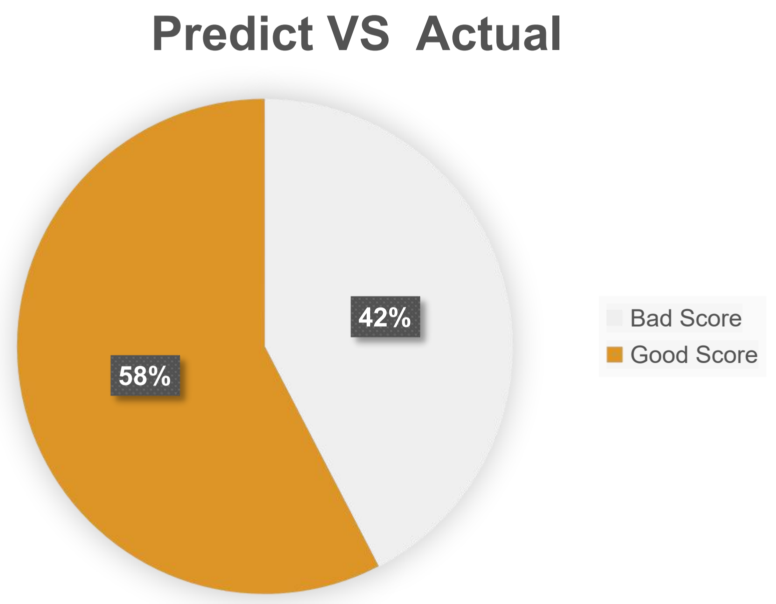
Feature Importance



For details, see Jupyter Notebook here

Model Testing

Of the 52,343 customers who have a bad credit score, the model successfully predicts 42%. So it is estimated that the model can reduce the number of customers who have bad credit scores by 42%.



07

Business Insight

- Make attractive offers regarding monthly installments at an installment rate of 150 to 600
- Conducting intensive marketing targets to customers with a length of service of over 10 years
- Doing massive marketing during the month of preparation for long holidays and preparation for education
- Conduct a special review of customers with a grade below A
- Conduct a special review of customers whose credit scores are indicated by the model

Thank YOu