id/x partners

Rakamin Academy

CREDIT RISK DETECTION USING MACHINE LEARNING

BACKGROUND

The development of credit risk detection stemmed from the aftermath of the 2008 financial crisis, as regulatory reforms and technological advancements prompted financial institutions to enhance their risk assessment methods. By merging traditional financial data with alternative sources and embracing machine learning and AI, credit risk detection systems emerged to provide a comprehensive view of borrowers' credit profiles and behavior. These systems enable more informed lending decisions, reduce default risks, and foster fairer lending practices. Looking forward, ongoing advancements in AI and data sharing are expected to drive even more sophisticated risk assessment techniques, bolstering financial stability and efficiency.

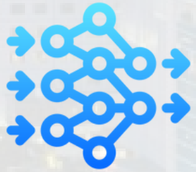
PROCESS CREATING MODEL

Data Wrangling

01.

Exploratory Data Analysis (EDA)

02.



Modeling

03.

Evaluate Model

04.

id/x partners

Rakamin Academy

Data Wrangling

DATA UNDERSTANDING

- You can get the data used in building the credit risk detection model on the following page:

Dataset | Data Dictionary

- Dataset contains 466285 Records
- 72 Columns
- Type of data type is float64(46), int64(4), object(22)

id/x partners

Rakamin Academy

Data Wrangling

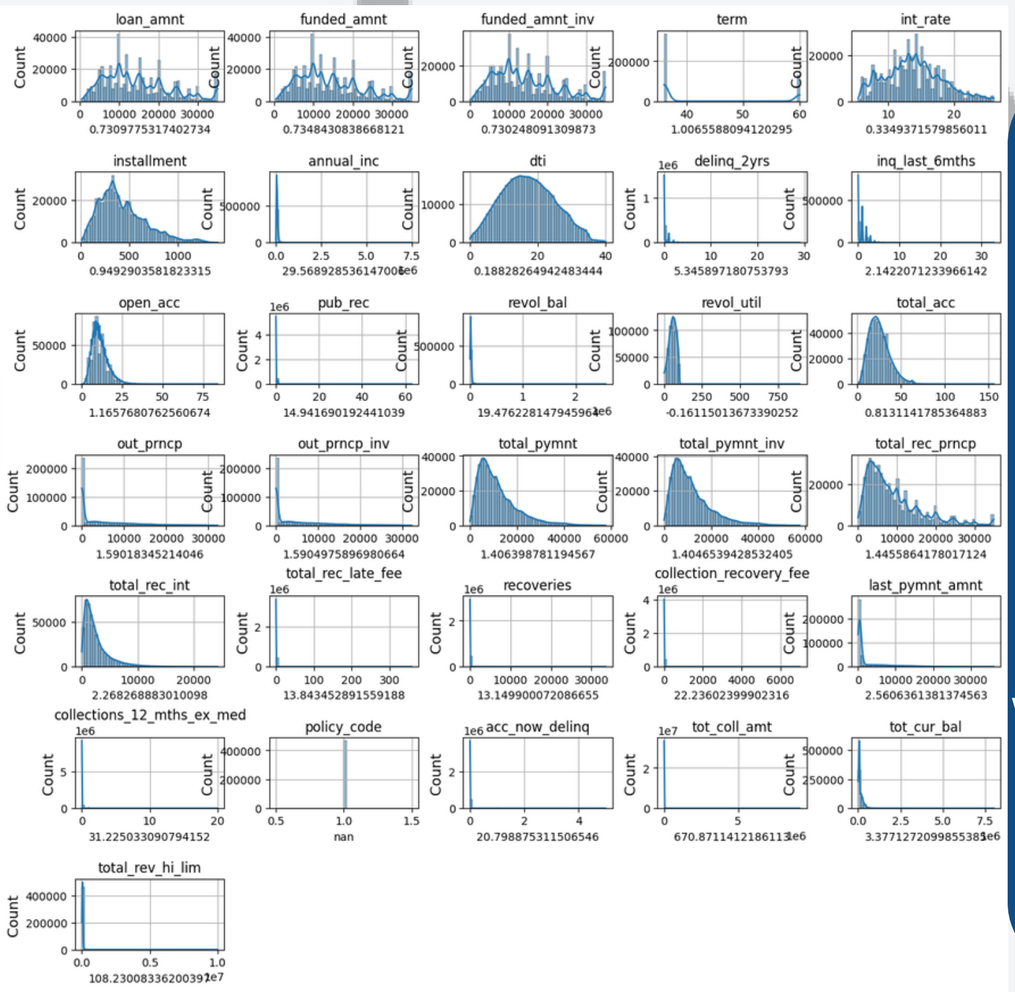
DATA CLEANING

inq_last_12m	100.000000
total_bal_il	100.000000
dti_joint	100.000000
verification_status_joint	100.000000
annual_inc_joint	100.000000
open_acc_6m	100.000000
open_il_6m	100.000000
open_il_12m	100.000000
open_il_24m	100.000000
mths_since_rcnt_il	100.000000
il_util	100.000000
open_rv_24m	100.000000
total_cu_tl	100.000000
inq_fi	100.000000
max_bal_bc	100.000000
all_util	100.000000
open_rv_12m	100.000000
mths_since_last_record	86.566585
mths_since_last_major_derog	78.773926
desc	72.981975
mths_since_last_delinq	53.690554
next_pymnt_d	48.728567
tot_cur_bal	15.071469
tot_coll_amt	15.071469
total_rev_hi_lim	15.071469
emp_title	5.916553
emp_length	4.505399
last_pymnt_d	0.080637

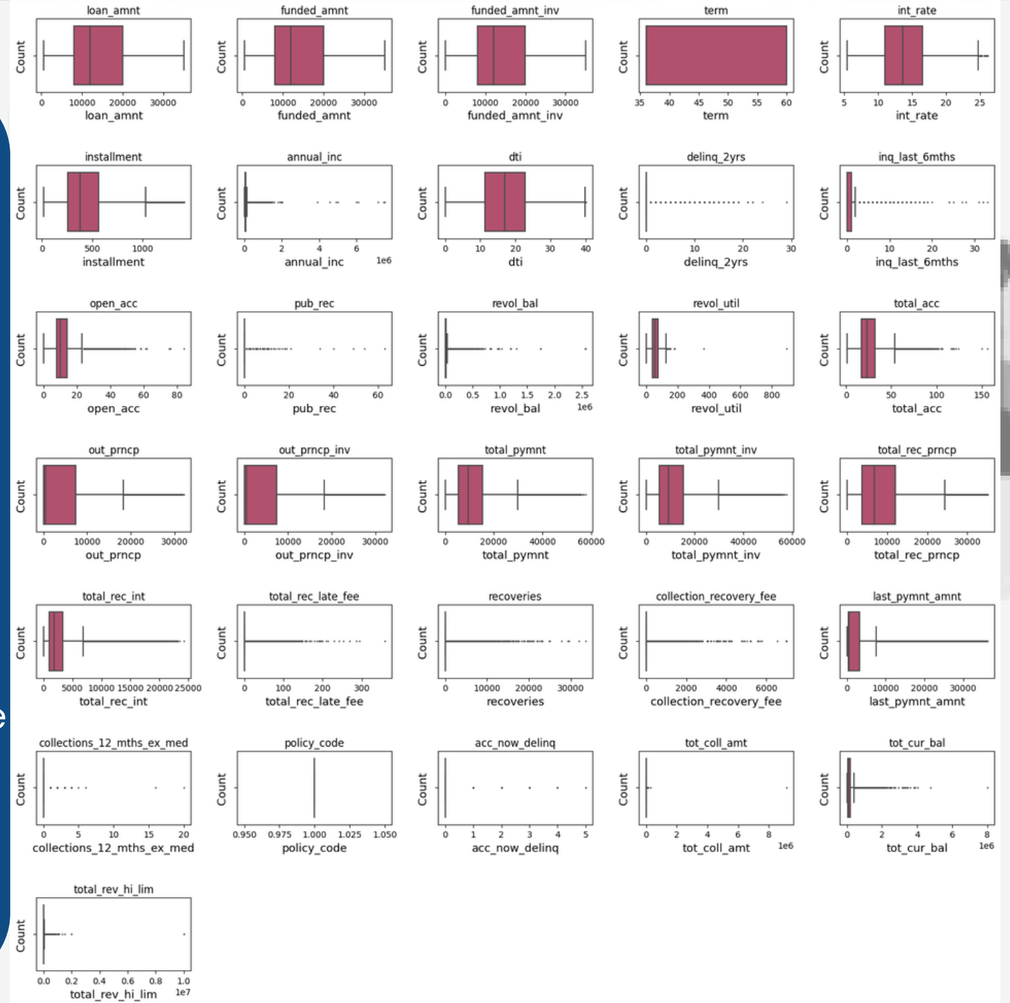
- Look the percentages of every feature how many features have missing value
- Drop the features where that features have more than 40% of missing value and also drop feature where wasn't needed

Exploratory Data Analysis (EDA)
Univariat

NUMERICAL DATA



It can be seen from the distribution of data and checking outliers by depicting histograms and boxplots for each numerical feature. There are many features that have an abnormal distribution, tend to experience positive skew and have a lot of possible outliers, which will cause distortion in the data. It would be nice if the feature would be dropped and features that have not so many outliers would be removed using the IQR technique.



id/x partners

Rakamin Academy

Exploratory Data Analysis (EDA)
Univariat

CATEGORICAL DATA

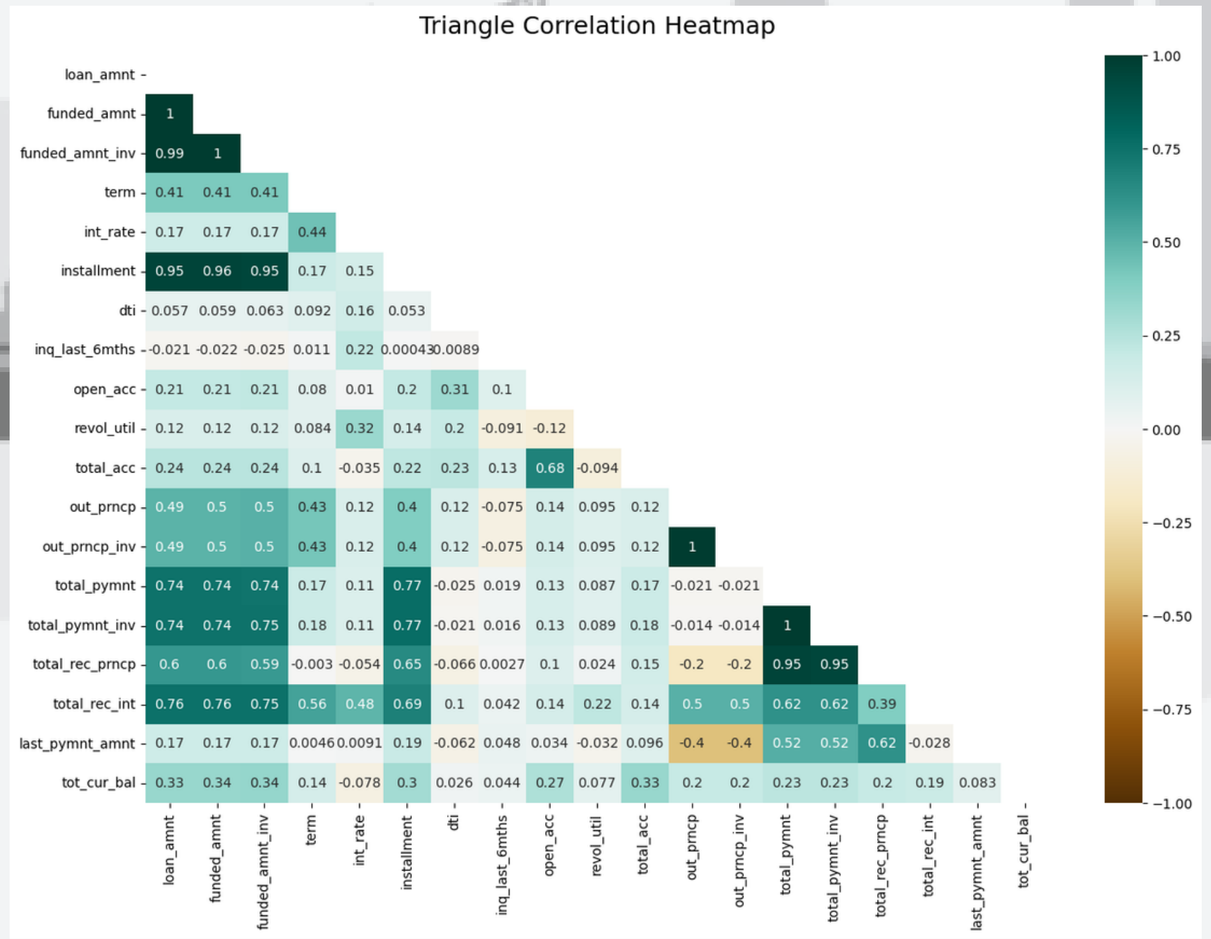
```
    # Count the value of categorical features
    cat_cols = df.select_dtypes(include=["object","category"]).columns.tolist()
    for col in cat_cols:
        print(df[col].value_counts())
        print("\n")
```

[26]

```
grade
B    136929
C    125293
D     76888
A     74867
E     35757
F     13229
G      3322
Name: count, dtype: int64


sub_grade
B3    31686
B4    30505
C1    26953
C2    26740
B2    26610
C3    25317
B5    25252
```

- Look the value counts of categorical features
- Drop the features who has only one value

Exploratory Data Analysis (EDA)
Multivariate

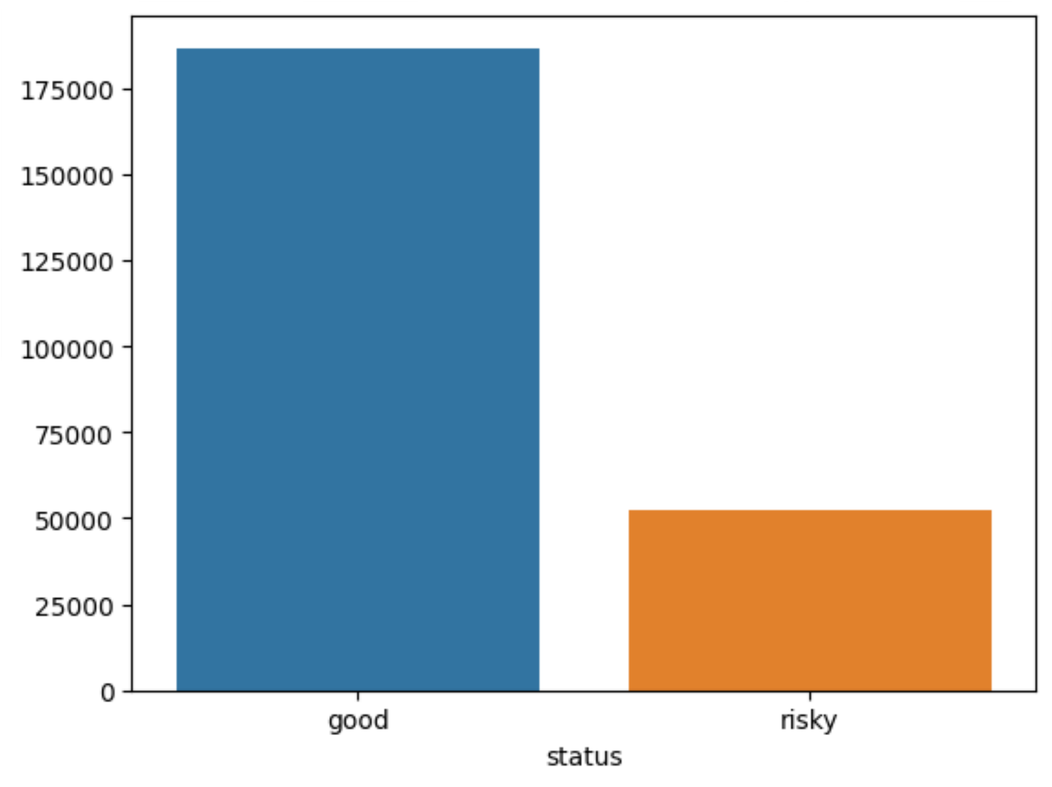


id/x partners

Rakamin Academy

Exploratory Data Analysis (EDA)

CHOOSING THE TARGET



Our aim was to predict the riskiness of loans, necessitating an understanding of the historical outcomes of each loan, whether they ended in default, being charged off, or fully paid. However, certain loan statuses like "Current" and "In Grace Period" carry uncertainty in their final outcomes, making them unsuitable for analysis. Even though "Late" status is also uncertain, I personally consider investing in late loans undesirable, so I will categorize them as unfavorable loans. To categorize the loan outcomes, we will designate loans as low-risk or good loans if they fall under statuses like "Fully Paid" or "Does not meet the credit policy. Status:Fully Paid." Conversely, loans deemed risky or bad will encompass statuses such as "Charged Off," "Late (31-120 days)," "Late (16-30 days)," "Default," and "Does not meet the credit policy. Status:Charged Off."

MODELING

```
# Seperate X and y label
X = df.drop('status', axis=1)
y = df['status']
```

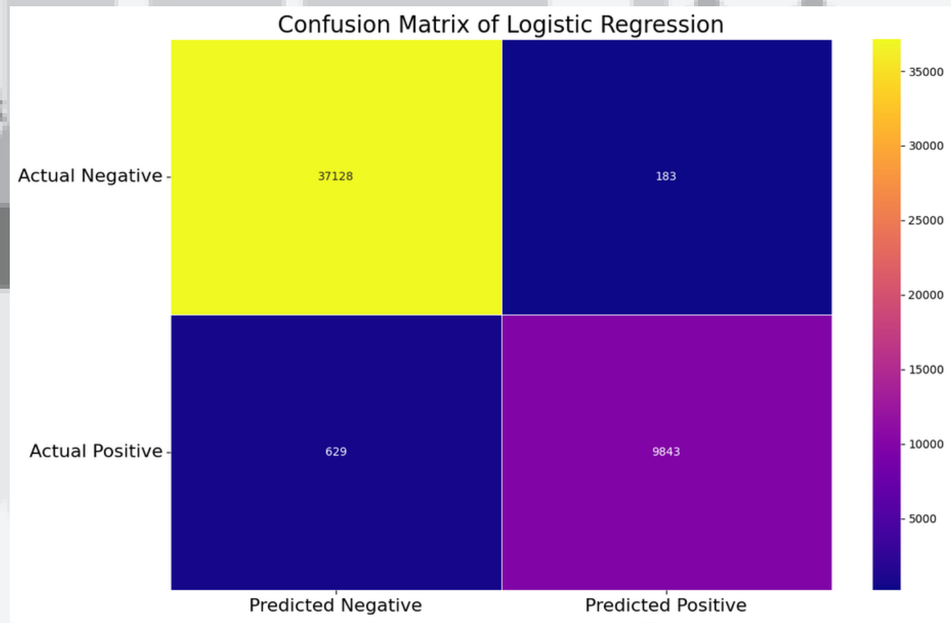
```
# Split and scalling data
X_train, X_test, y_train, y_test = train_test_split(X , y, shuffle = True, test_size = 0.2, random_state = 42)

# Scaling data
scaler = MinMaxScaler()
scaler.fit(X_train)

# The transofrmation of X
X_train_scaled = scaler.transform(X_train)
X_test_scaled = scaler.transform(X_test)

# See the end of dimensions data
print('Dimensi feature data train =', X_train_scaled.shape)
print('Dimensi target data train =', y_train.shape)
print('Dimensi feature data test =', X_test_scaled.shape)
print('Dimensi target data test =', y_test.shape)
```

```
lrmodel = LogisticRegression()
lrmodel.fit(X_train_scaled, y_train)
```

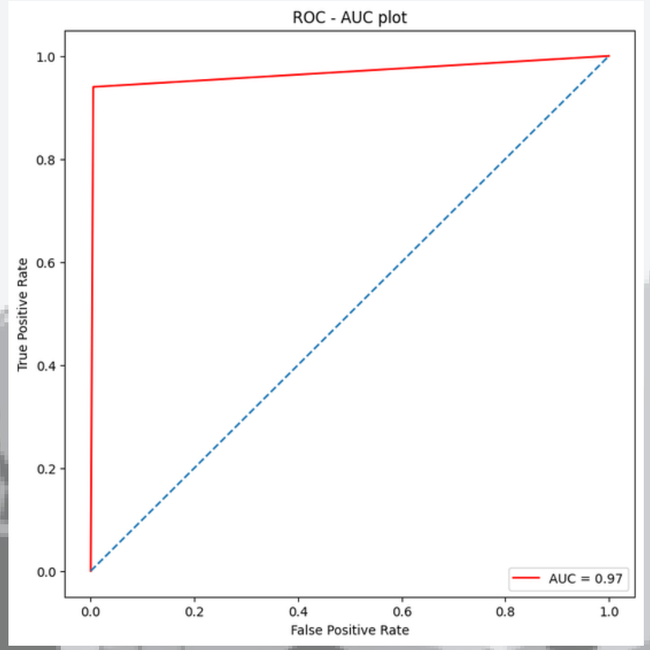
```
y_predLr = lrmodel.predict(X_test_scaled)
```

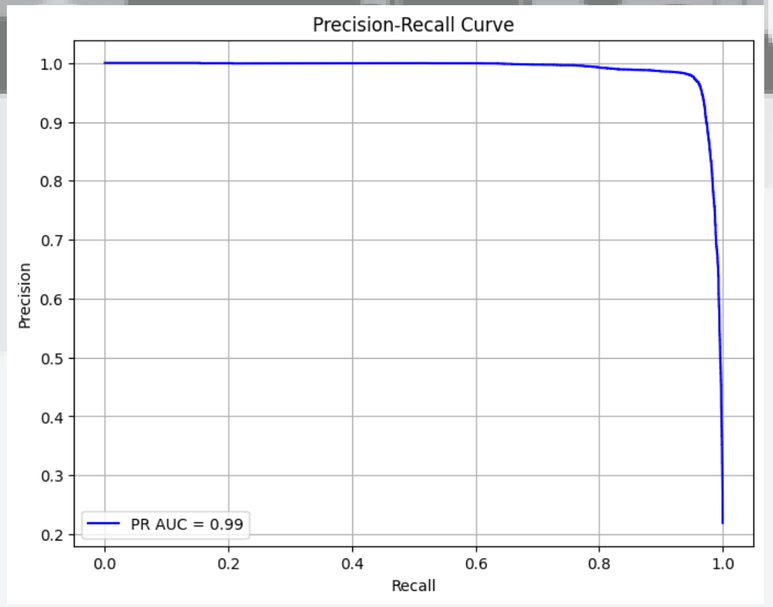
EVALUATING

	precision	recall	f1-score	support
0	0.98	1.00	0.99	37311
1	0.98	0.94	0.96	10472
accuracy			0.98	47783
macro avg	0.98	0.97	0.97	47783
weighted avg	0.98	0.98	0.98	47783



EVALUATING


ROC - AUC plot
True Positive Rate
False Positive Rate
AUC = 0.97
0.0
0.2
0.4
0.6
0.8
1.0


Precision-Recall Curve
Precision
Recall
PR AUC = 0.99
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1.0

EVALUATING

Based on the confusion matrix results, the classification model demonstrates high performance in predicting both classes. For the "0" class (considered as the positive outcome), the model achieves remarkable precision and recall scores of 0.98 and 1.00, respectively, along with an impressive F1-score of 0.99. This indicates the model's ability to accurately identify instances where the outcome is truly "0" and its exceptional capability in avoiding false negatives. For the "1" class (considered as the negative outcome), the model maintains a high precision of 0.98 and a respectable recall of 0.94, yielding an F1-score of 0.96. This signifies the model's proficiency in correctly classifying instances as "1" while moderately handling false positives. The overall accuracy of the model is reported at 0.98, indicating the correct classification of approximately 98% of instances across both classes. In conclusion, the model demonstrates a strong ability to effectively distinguish between the two classes, supported by consistently high precision, recall, and F1-score values, contributing to its reliable performance in this classification task.