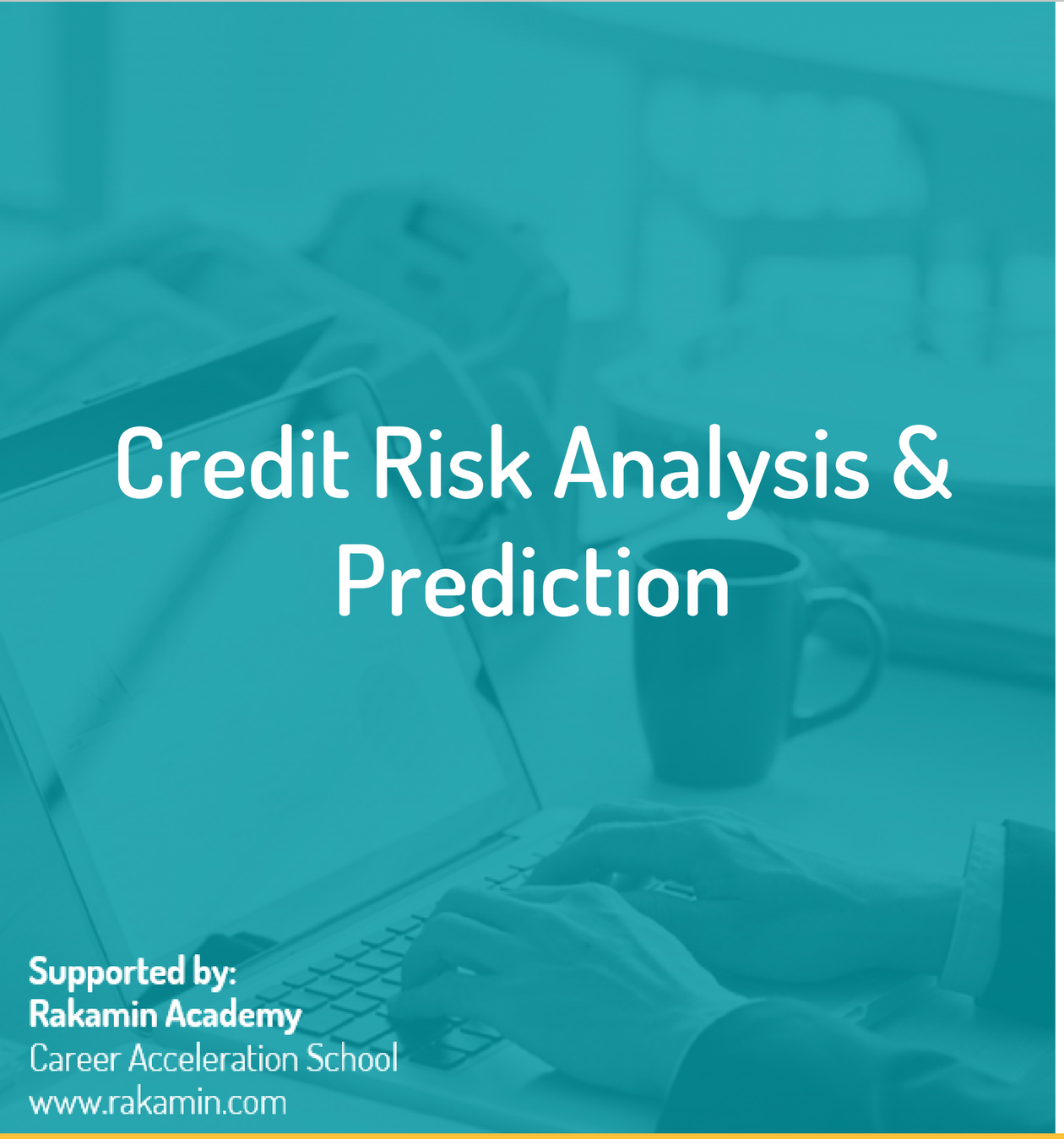


Credit Risk Analysis & Prediction

Supported by:
Rakamin Academy
Career Acceleration School
www.rakamin.com

Rakamin Academy

id/x partners

ID/X Partners Data Scientist Virtual Internship Experience

Muhammad Randa Yandika

https://randayandika.github.io/
https://linkedin.com/in/muhammad-randa-yandika

Summary

Objective

- To build a model that can predict credit risk for loan applicants, based on Loan dataset.

Outcome:

- Improved accuracy of credit risk predictions: By building a model to predict credit risk, the company can improve the accuracy of their lending decisions.
- Faster lending decisions: With a model to predict credit risk, the company can automate part of the lending decision-making process.
- Increased efficiency: Automating part of the lending decision-making process can also increase efficiency, as it reduces the need for manual review of every loan application.

Business Understanding

- The company deals with loan applications: The dataset provided by ID/X Partners, consists of accepted and rejected loan data, indicating that the company is in the business of processing loan applications.

- Credit risk is a concern for the company: The fact that the company wants to build a model to predict credit risk suggests that credit risk is a concern for the company. This is likely because the company wants to avoid lending money to individuals who are unlikely to pay it back, as this could result in financial losses.

- The model will help the company make lending decisions: The model being built will likely be used to inform lending decisions. By predicting credit risk, the model will help the company determine whether to approve or reject loan applications, and may also inform the terms of the loan (such as interest rate).

Data Understanding

- Dataset have have 74 columns with 466.285 entries and data type from each column
- Have 22 categorical feature and 52 numerical feature
- Have a data type like int64, float64, and object

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 466285 entries, 0 to 466284
Data columns (total 74 columns):
 #   Column                      Non-Null Count   Dtype
---  ------                      --------------   -----
 0   id                          466285 non-null  int64
 1   member_id                   466285 non-null  int64
 2   loan_amnt                   466285 non-null  int64
 3   funded_amnt                 466285 non-null  int64
 4   funded_amnt_inv             466285 non-null  float64
 5   term                        466285 non-null  object
 6   int_rate                    466285 non-null  float64
 7   installment                 466285 non-null  float64
 8   grade                       466285 non-null  object
 9   sub_grade                   466285 non-null  object
 10  emp_title                   438697 non-null  object
 11  emp_length                  445277 non-null  object
 12  home_ownership              466285 non-null  object
 13  annual_inc                  466281 non-null  float64
 14  verification_status         466285 non-null  object
 15  issue_d                     466285 non-null  object
 16  loan_status                 466285 non-null  object
 17  pymnt_plan                  466285 non-null  object
 18  url                         466285 non-null  object
 19  desc                        125983 non-null  object
 20  purpose                     466285 non-null  object
 21  title                       466265 non-null  object
 22  zip_code                    466285 non-null  object
 23  addr_state                  466285 non-null  object
 24  dti                         466285 non-null  float64
 25  delinq_2yrs                 466256 non-null  float64
 26  earliest_cr_line            466256 non-null  object
 27  inq_last_6mths              466256 non-null  float64
 28  mths_since_last_delinq      215934 non-null  float64
 29  mths_since_last_record      62638 non-null   float64
 30  open_acc                    466256 non-null  float64
 31  pub_rec                     466256 non-null  float64
 32  revol_bal                   466285 non-null  int64
 33  revol_util                  465945 non-null  float64
 34  total_acc                   466256 non-null  float64
 35  initial_list_status         466285 non-null  object
 36  out_prncp                   466285 non-null  float64
 37  out_prncp_inv               466285 non-null  float64
 38  total_pymnt                 466285 non-null  float64
 39  total_pymnt_inv             466285 non-null  float64
 40  total_rec_prncp             466285 non-null  float64
 41  total_rec_int               466285 non-null  float64
 42  total_rec_late_fee          466285 non-null  float64
 43  recoveries                  466285 non-null  float64
 44  collection_recovery_fee     466285 non-null  float64
 45  last_pymnt_d                465909 non-null  object
 46  last_pymnt_amnt             466285 non-null  float64
 47  next_pymnt_d                239071 non-null  object
 48  last_credit_pull_d          466243 non-null  object
 49  collections_12_mths_ex_med  466140 non-null  float64
 50  mths_since_last_major_derog 98974 non-null   float64
 51  policy_code                 466285 non-null  int64
 52  application_type            466285 non-null  object
 53  annual_inc_joint            0 non-null       float64
 54  dti_joint                   0 non-null       float64
 55  verification_status_joint   0 non-null       float64
 56  acc_now_delinq              466256 non-null  float64
 57  tot_coll_amt                396009 non-null  float64
 58  tot_cur_bal                 396009 non-null  float64
 59  open_acc_6m                 0 non-null       float64
 60  open_il_6m                  0 non-null       float64
 61  open_il_12m                 0 non-null       float64
 62  open_il_24m                 0 non-null       float64
 63  mths_since_rcnt_il          0 non-null       float64
 64  total_bal_il                0 non-null       float64
 65  il_util                     0 non-null       float64
 66  open_rv_12m                 0 non-null       float64
 67  open_rv_24m                 0 non-null       float64
 68  max_bal_bc                  0 non-null       float64
 69  all_util                    0 non-null       float64
 70  total_rev_hi_lim            396009 non-null  float64
 71  inq_fi                      0 non-null       float64
 72  total_cu_tl                 0 non-null       float64
 73  inq_last_12m                0 non-null       float64
```

Statistical Descriptive

Numerical Features

```
df.describe()
```

	id	member_id	loan_amnt	funded_amnt	funded_amnt_inv	int_rate	installment	annual_inc	dti	delinq_2yrs
count	4.662850e+05	4.662850e+05	466285.000000	466285.000000	466285.000000	466285.000000	466285.000000	4.662810e+05	466285.000000	466256.000000
mean	1.307973e+07	1.459766e+07	14317.277577	14291.801044	14222.329888	13.829236	432.061201	7.327738e+04	17.218758	0.284678
std	1.089371e+07	1.168237e+07	8286.509164	8274.371300	8297.637788	4.357587	243.485550	5.496357e+04	7.851121	0.797365
min	5.473400e+04	7.047300e+04	500.000000	500.000000	0.000000	5.420000	15.670000	1.896000e+03	0.000000	0.000000
25%	3.639987e+06	4.379705e+06	8000.000000	8000.000000	8000.000000	10.990000	256.690000	4.500000e+04	11.360000	0.000000
50%	1.010790e+07	1.194108e+07	12000.000000	12000.000000	12000.000000	13.660000	379.890000	6.300000e+04	16.870000	0.000000
75%	2.073121e+07	2.300154e+07	20000.000000	20000.000000	19950.000000	16.490000	566.580000	8.896000e+04	22.780000	0.000000
max	3.809811e+07	4.086083e+07	35000.000000	35000.000000	35000.000000	26.060000	1409.990000	7.500000e+06	39.990000	29.000000

8 rows × 52 columns

Categorical Features

	count	unique	top	freq
term	466285	2	36 months	337953
grade	466285	7	B	136929
sub_grade	466285	35	B3	31686
emp_title	438697	205475	Teacher	5399
emp_length	445277	11	10+ years	150049
home_ownership	466285	6	MORTGAGE	235875
verification_status	466285	3	Verified	168055
issue_d	466285	91	Oct-14	38782
loan_status	466285	9	Current	224226
pymnt_plan	466285	2	n	466276
url	466285	466285	https://www.lendingclub.com/browse/loanDetail....	1
desc	125983	124436		234
purpose	466285	14	debt_consolidation	274195
title	466265	63099	Debt consolidation	164075
zip_code	466285	888	945xx	5304
addr_state	466285	50	CA	71450
earliest_cr_line	466256	664	Oct-00	3674
initial_list_status	466285	2	f	303005
last_pymnt_d	465909	98	Jan-16	179620
next_pymnt_d	239071	100	Feb-16	208393
last_credit_pull_d	466243	103	Jan-16	327699
application_type	466285	1	INDIVIDUAL	466285

Data Preparation & Cleansing

Dataset is messy, before we use the data for EDA and Visualization we need to do some things like:

- Drop columns that only have null values, 17 columns with such condition.
- Drop columns with too much unique data.
- Drop columns with only have 1 unique data.

After the cleansing process, 49 out of 74 columns remain.

and create a order for the values of these 3 columns

```
grade_order = ['A','B','C','D','E','F','G']
df_clean['grade'] = pd.Categorical(df_clean['grade'], categories=grade_order, ordered=True)
```

```
subgrade_order = ['A1', 'A2', 'A3', 'A4', 'A5', 'B1', 'B2', 'B3', 'B4', 'B5', 'C1',
       'C2', 'C3', 'C4', 'C5', 'D1', 'D2', 'D3', 'D4', 'D5', 'E1', 'E2',
       'E3', 'E4', 'E5', 'F1', 'F2', 'F3', 'F4', 'F5', 'G1', 'G2', 'G3',
       'G4', 'G5']
df_clean['sub_grade'] = pd.Categorical(df_clean['sub_grade'], categories=subgrade_order, ordered=True)
```

```
length_order = ['< 1 year','1 year', '2 years', '3 years', '4 years', '5 years', '6 years', '7 years', '8 years', '9 years', '10+
df_clean['emp_length'] = pd.Categorical(df_clean['emp_length'], categories=length_order, ordered=True)
```

Data Preparation & Cleansing

Feature and Target:

We need to change target column value into boolean, in accordance with the conditions of the loan status. where ['Charged Off', 'Default' , 'Does not meet the credit policy. Status:Charged Off', 'Late (31-120 days)','Late (16-30 days)' is a bad status. and the rest is good status

Before:

```
Current                                                 224226
Fully Paid                                              184739
Charged Off                                              42475
Late (31-120 days)                                        6900
In Grace Period                                           3146
Does not meet the credit policy. Status:Fully Paid        1988
Late (16-30 days)                                         1218
Default                                                    832
Does not meet the credit policy. Status:Charged Off        761
Name: loan_status, dtype: int64
```

After:

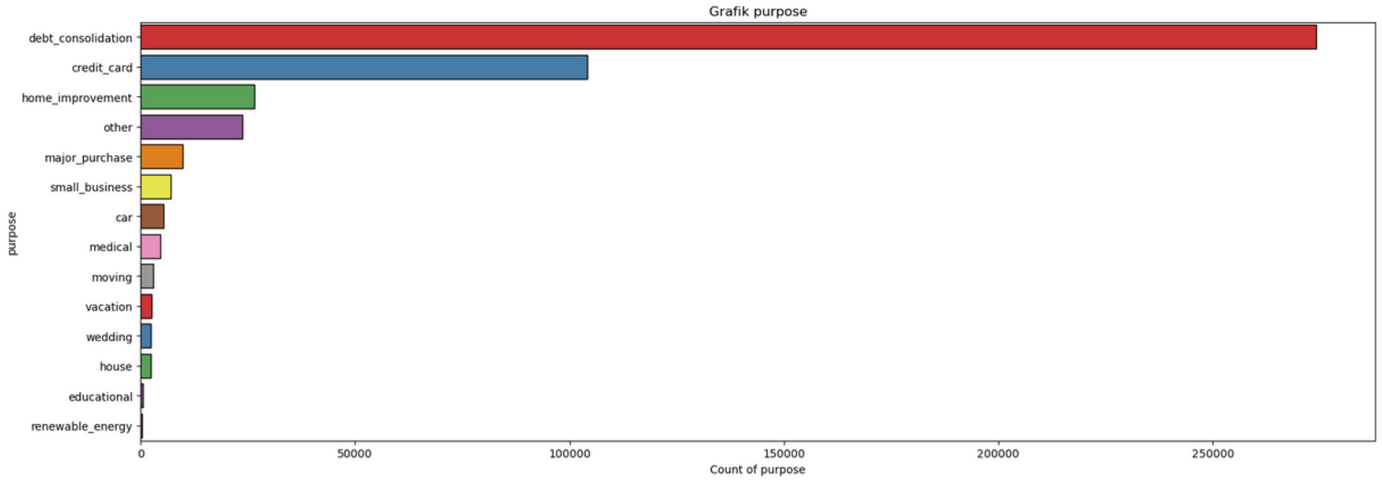
```
status_counts = df_clean["target"].value_counts()

# Print the value counts
print(status_counts)
```

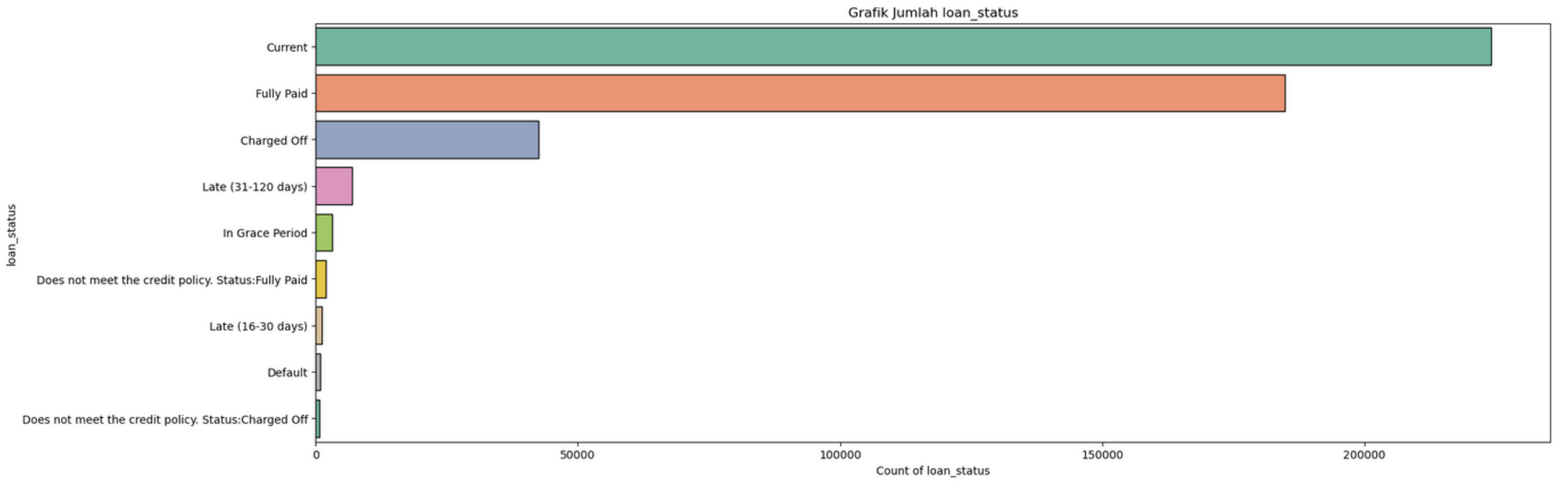
```
0    414099
1     52186
Name: target, dtype: int64
```

0 = Low Risk / Good Loan, 1= High Risk/Bad Loan

Categorical Feature Univariate Analysis

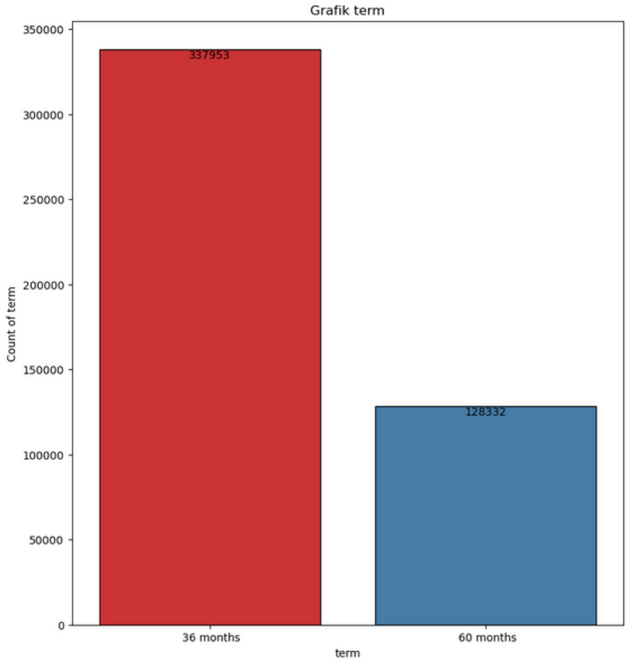
The most reason to take a loan is debt consolidation and the most loan status is Current.

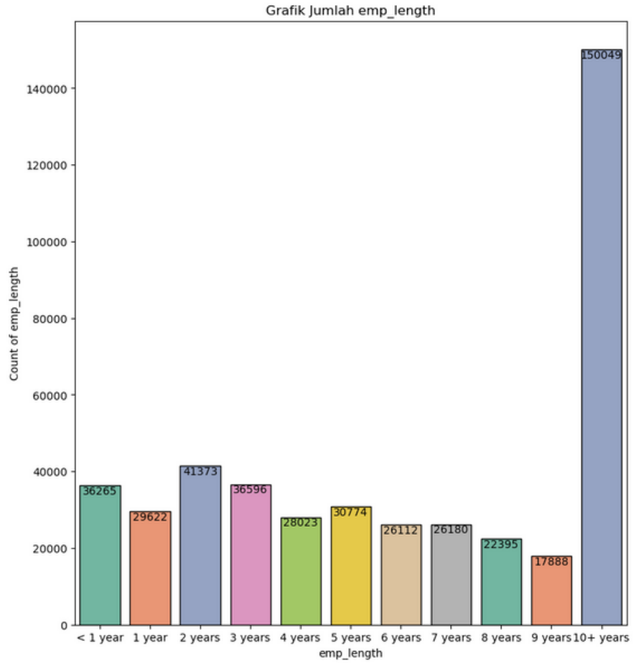


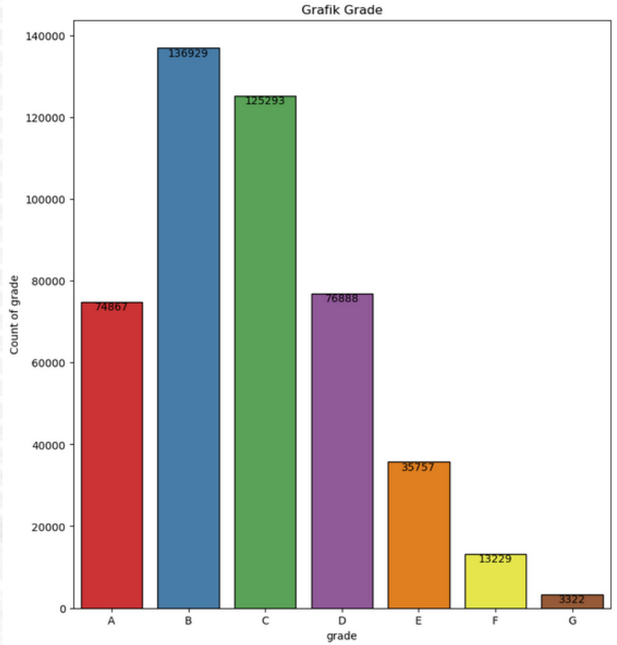


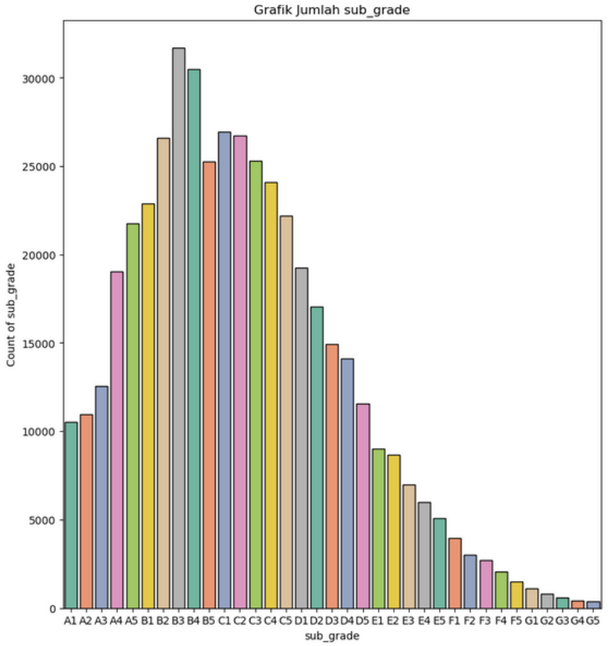
Categorical Feature Univariate Analysis

- The term with the most amount is within 36 months.
- The most employment length is more than 10 years.
- Grade B is the most Grade.

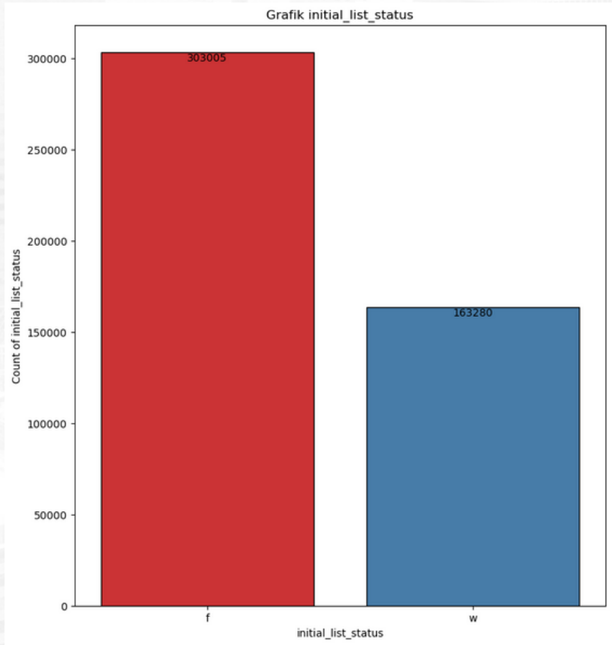


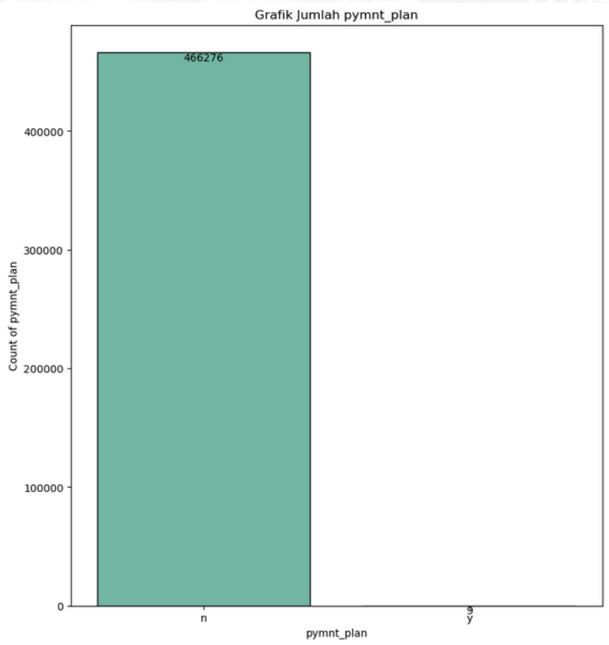


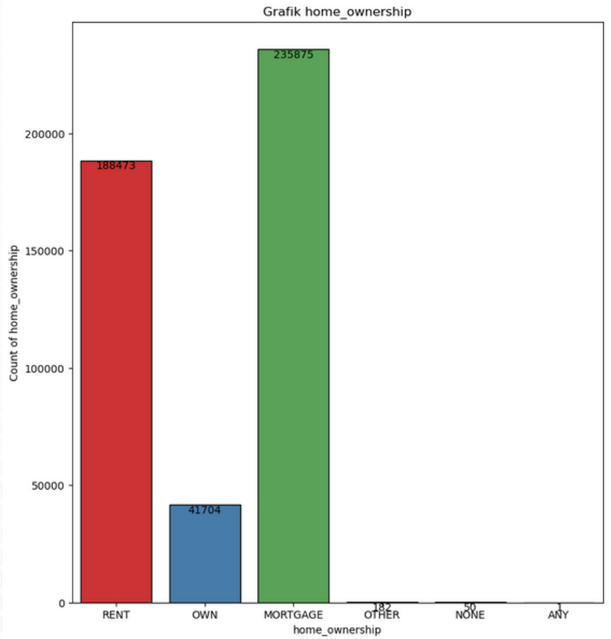


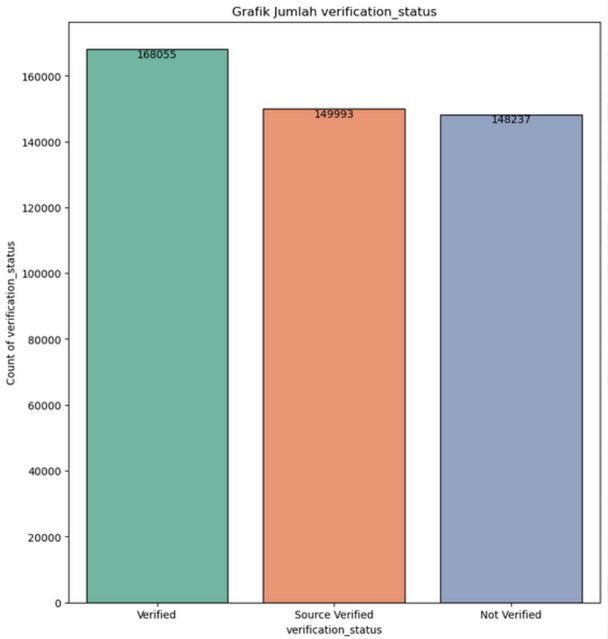


Categorical Feature Univariate Analysis



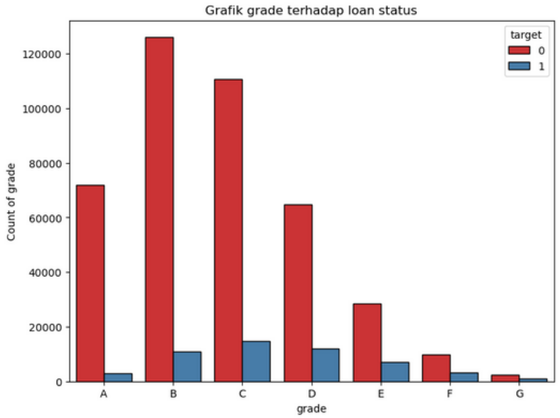


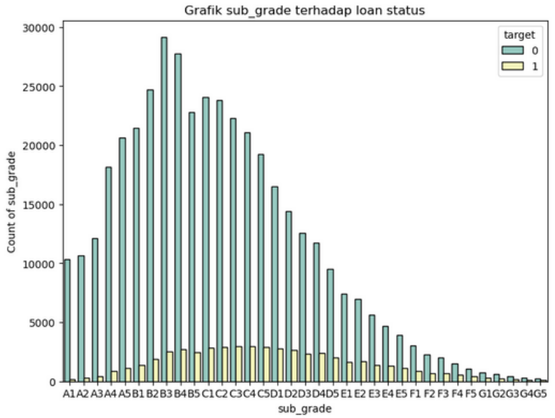


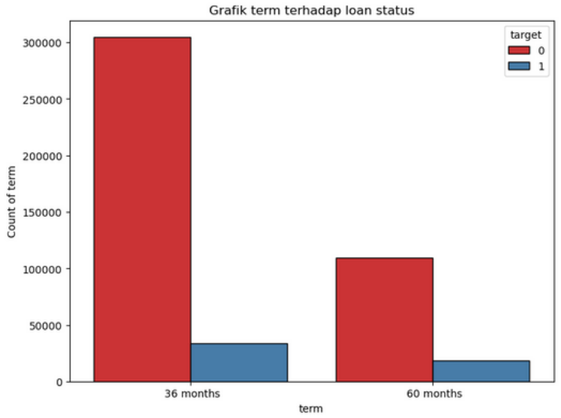


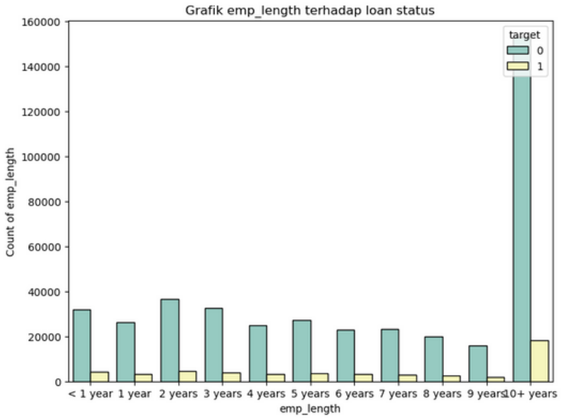
Categorical Feature Univariate Analysis

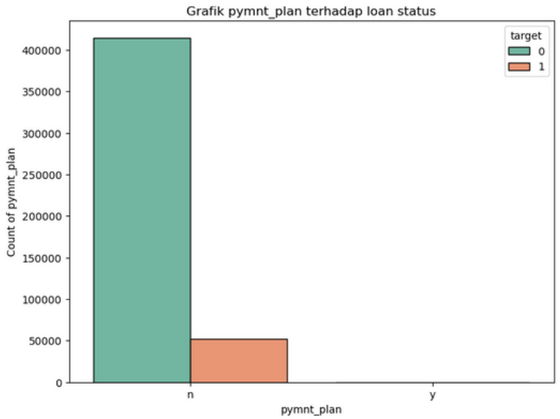
These are some bar graphs according to loan status conditions

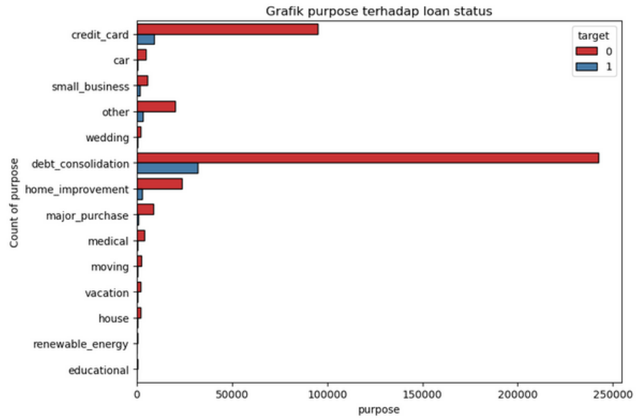


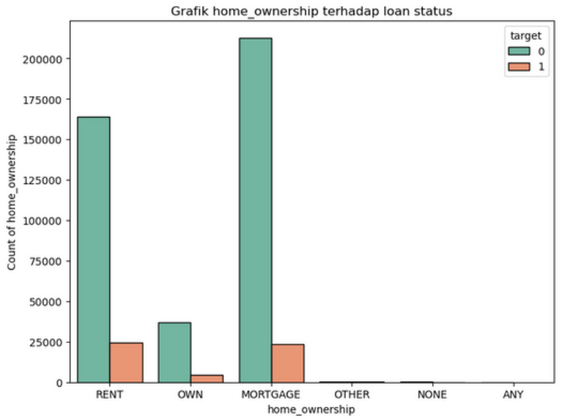


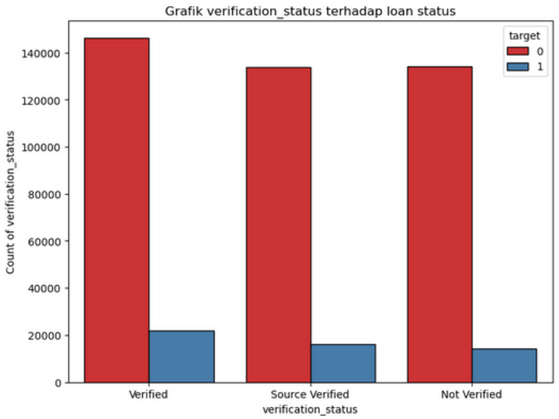






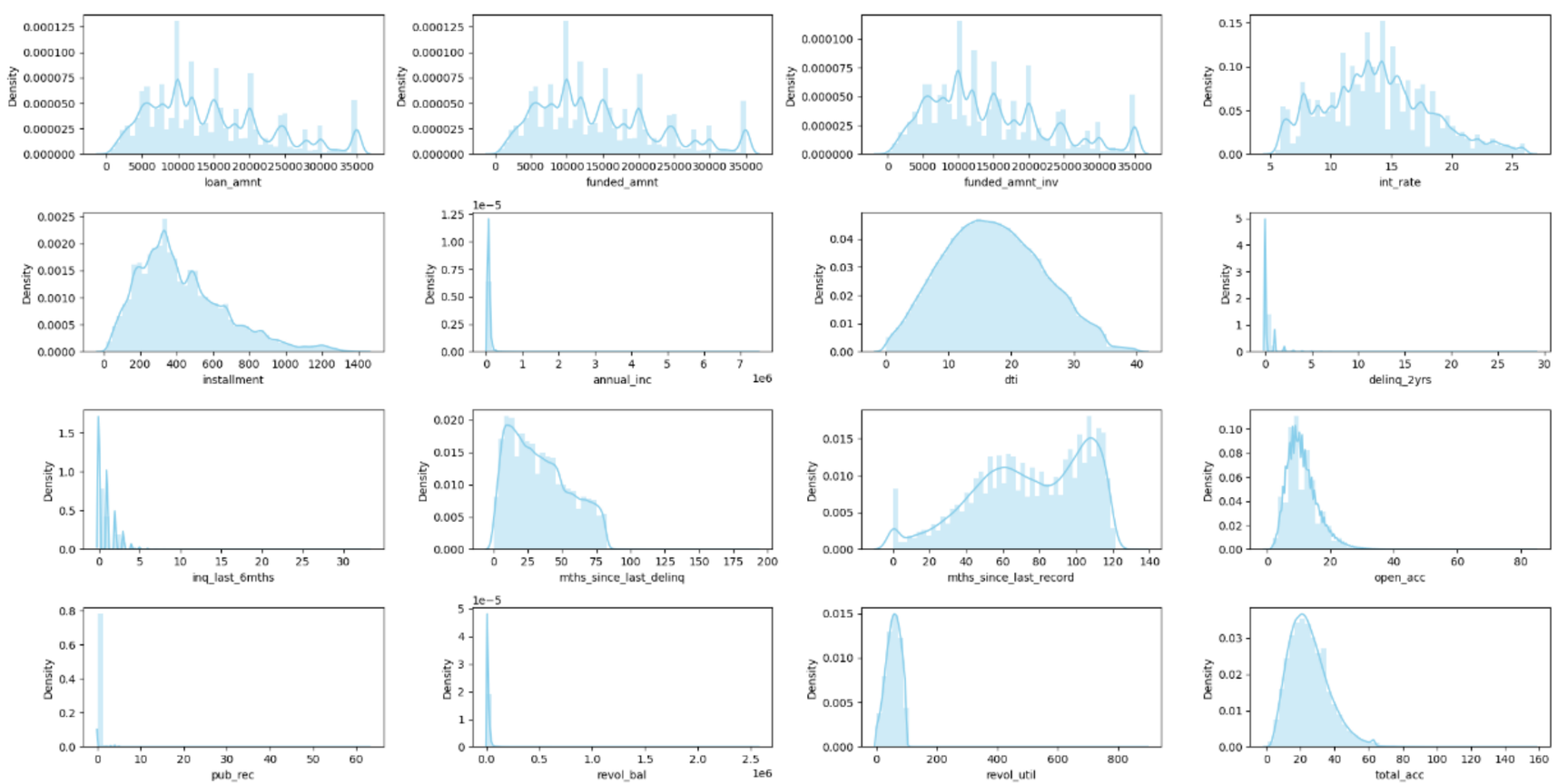




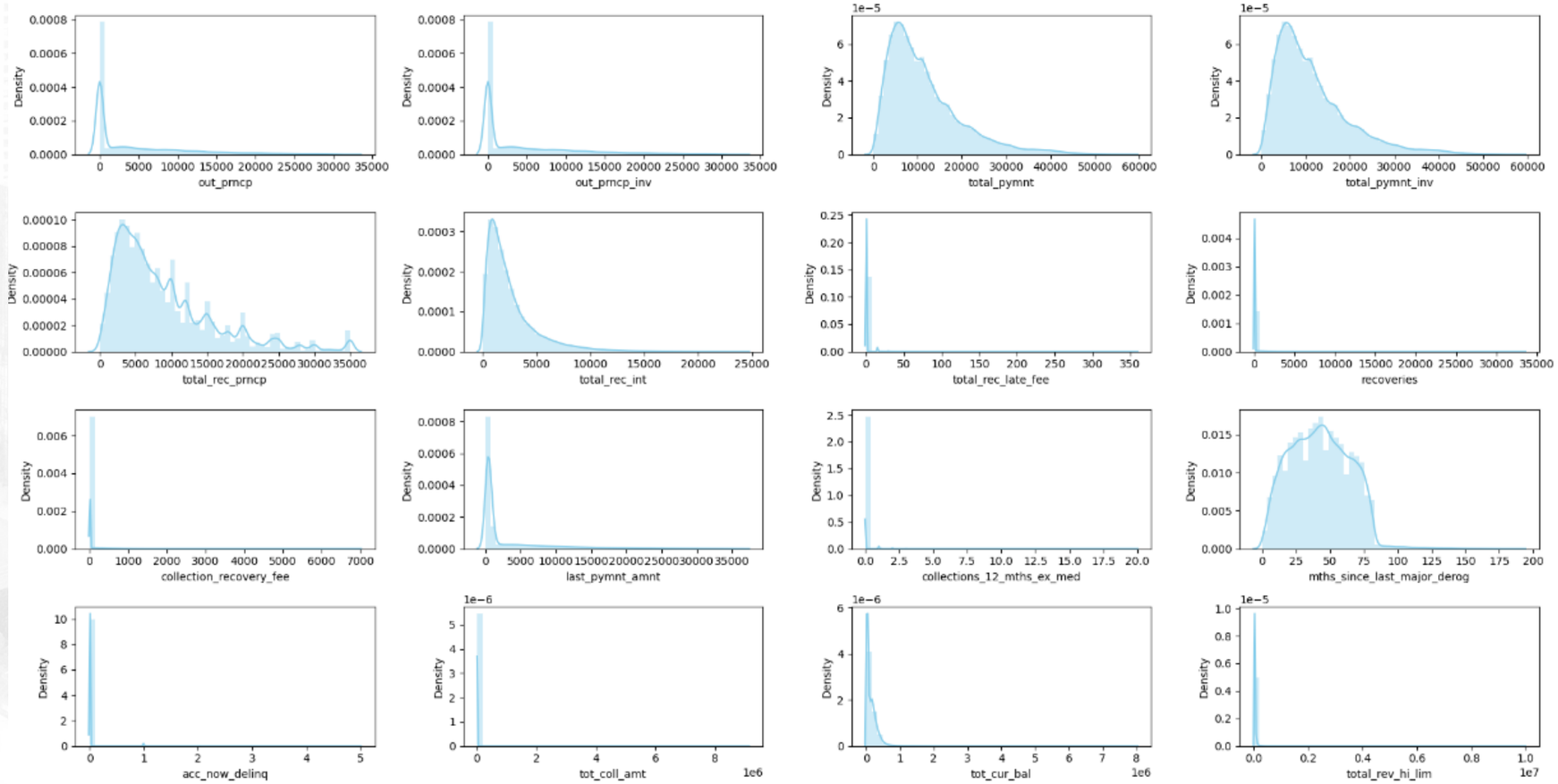


Numerical Feature Univariate Analysis

There are several features with skew distributions

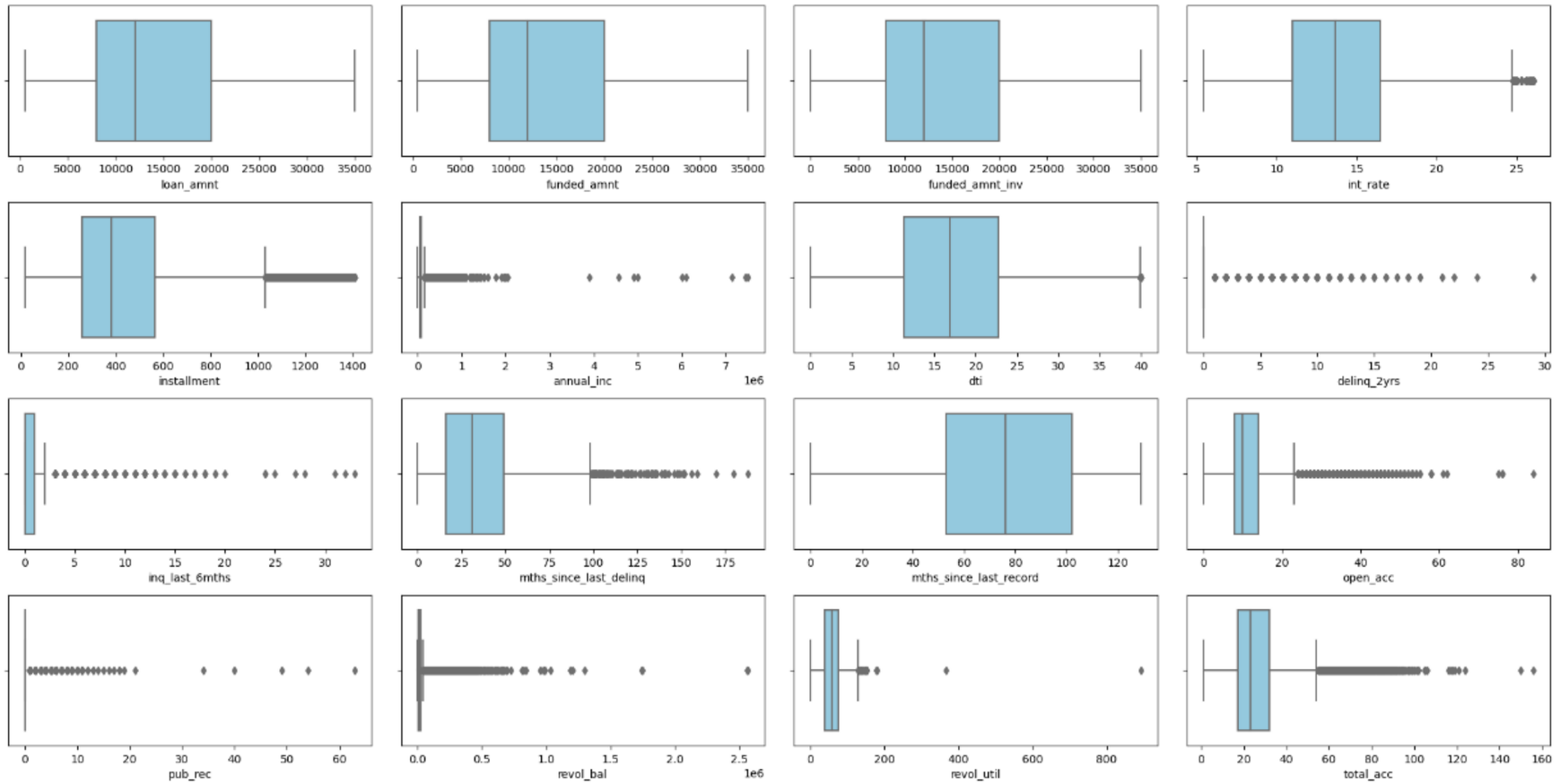


Numerical Feature Univariate Analysis

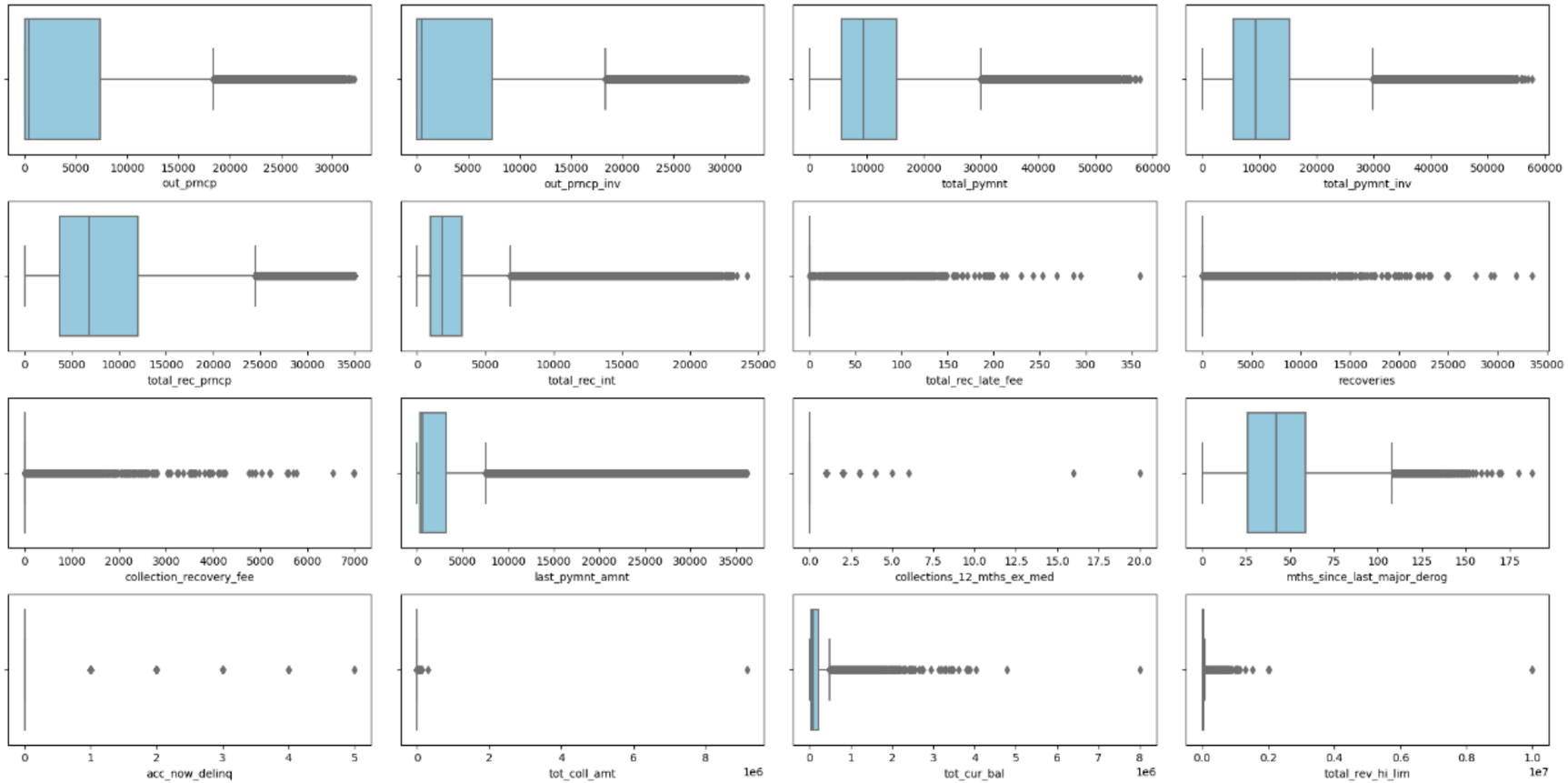


Numerical Feature Univariate Analysis

there are some outliers which can be removed

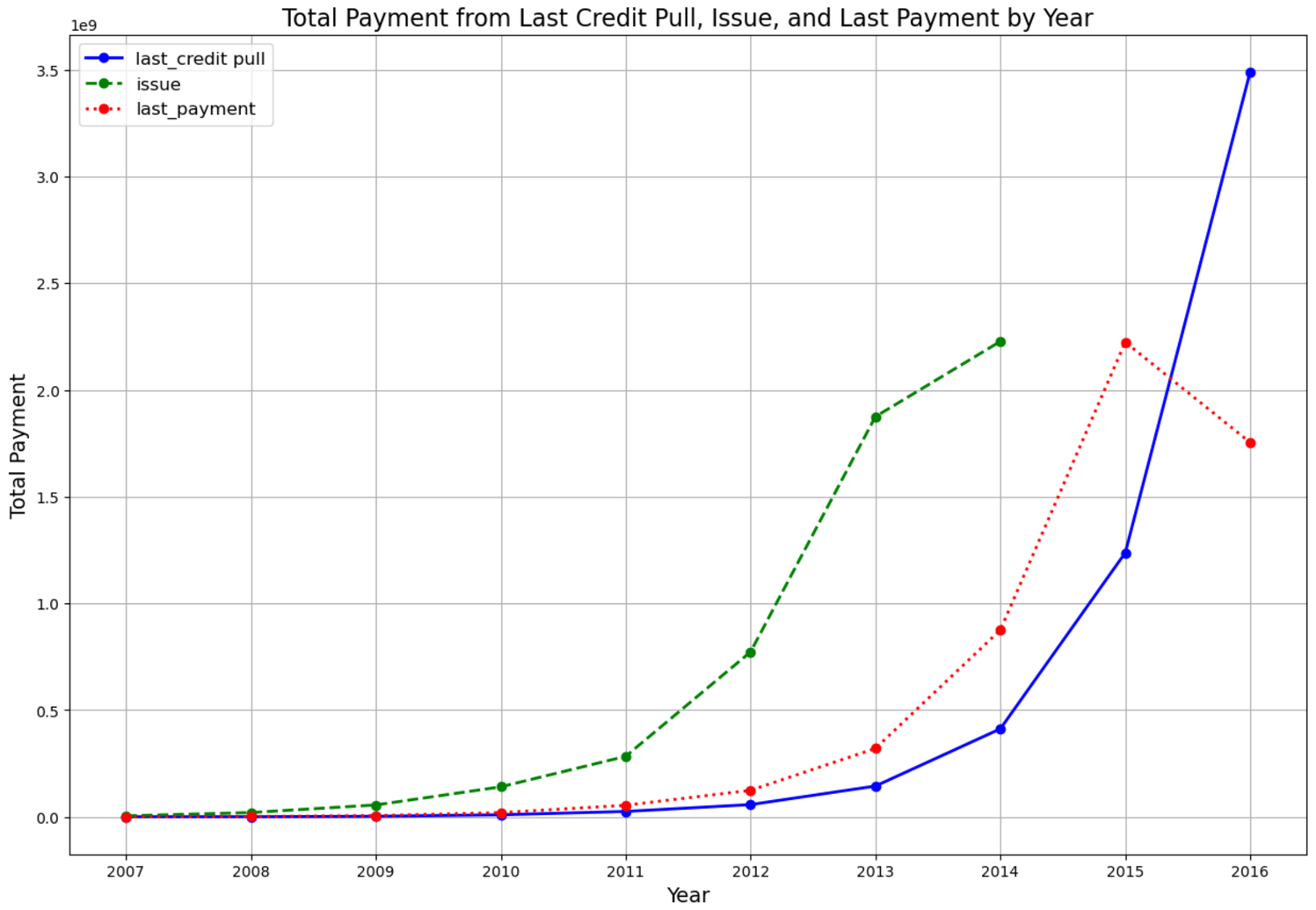


Numerical Feature Univariate Analysis



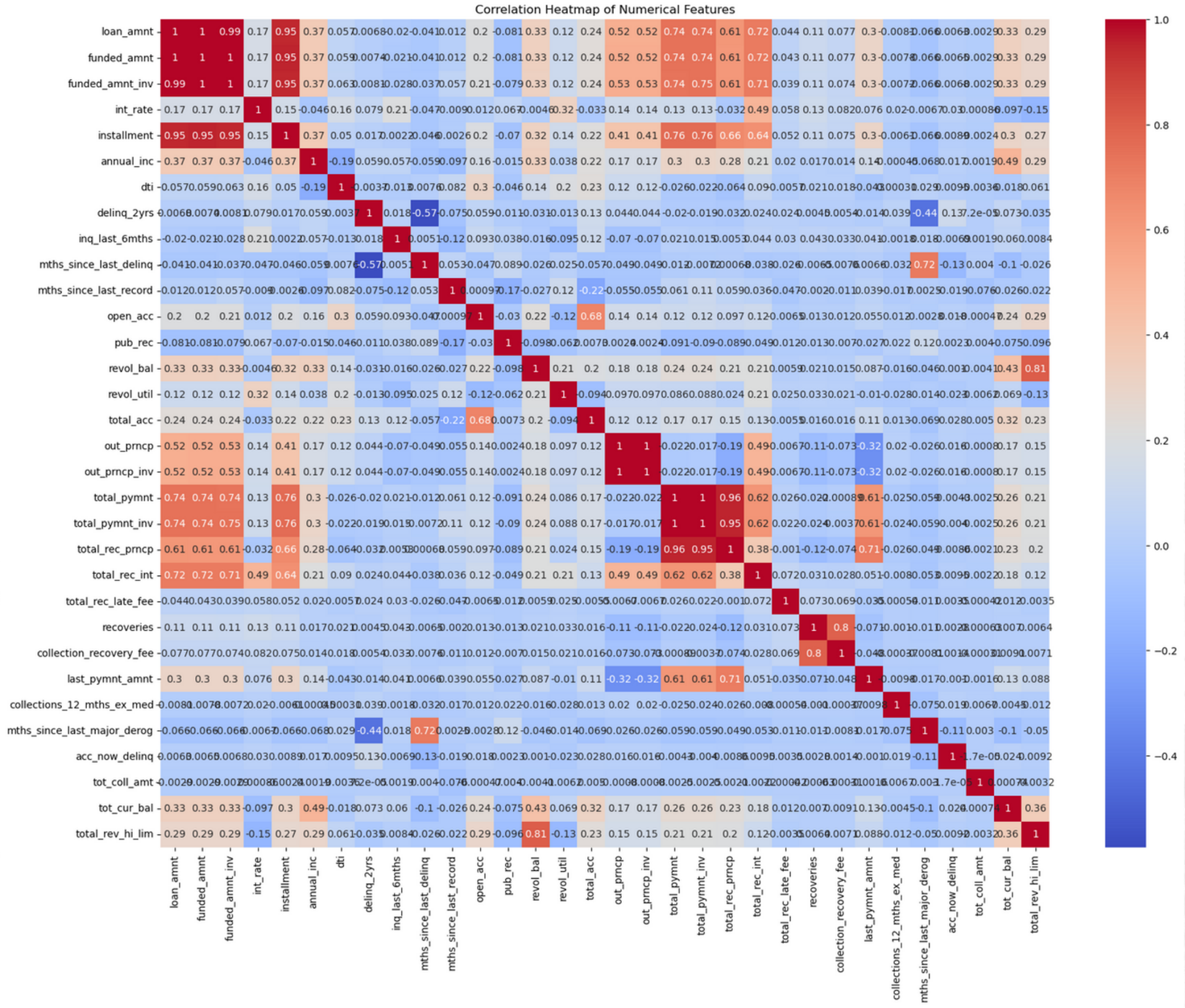
Time Series Line Plot

there was an increase in total payments for these 3 features from 2011 to 2015



Multivariate Analysis

After check feature correlation, we must remove 1 from 2 or more features/predictors are highly correlated with each other called Multicolliear Features



Data Preprocessing

Before modelling process, we need to clean dataset:

- Null Values Imputation (Mean and Median to Numerical data, Mode to Categorical Data)
- Drop a column that has too many null data.
- Check Duplicated Data.
- Drop columns we dont need to modelling process like, id and member id
- Remove Outlier Data

Feature Engineering

- Change value in Date Feature into numerical.
- Example: From 1 issue_d column into 2 Column issue_d_month and issue_d_year. Value change Jan-13 to 1 and 2013.

Data Preprocessing

Label Encoding:

- Change Categorical Data into numerical data.
- Encode using One-Hot Encoding Method to several Categorical Columns.

Handling Imbalanced Data

- SMOTE
- Undersampling

Handling Imbalanced Data with SMOTE

Modelling with SMOTE

```
: X_train, X_val, y_train, y_val = train_test_split(X, y, test_size=0.2, random_state=42)
```

```
: # Overampling with smote
from imblearn.over_sampling import SMOTE
from imblearn import over_sampling
X_over_smote, y_over_smote = over_sampling.SMOTE().fit_resample(X_train, y_train)
```

```
: X_train_, X_test, y_train_, y_test = train_test_split(X_over_smote, y_over_smote, test_size=0.2, random_state=42)
```

- First, we split X & y into data train and data validation with 80:20 ratio.
- And then from the train data it will be resample using SMOTE to produce X_over_smote, y_over_smote.
- We split again from X_over_smote, y_over_smote with same ratio into new data train and data test.
- From this step, we can predict from 3 data, data train, data test, and data val.
- Data val is original data that is not carried out any processing so as to produce appropriate predictions

Modelling Process

In Modelling process, we use some of Machine Learning Algorithm.

- Logistic Regression
- Random Forest
- Decision Tree

For Modelling process, we split data with ratio 80 data train :20 data test, and we check shape of data train and test

Modelling Process

This is result of modelling process using 3 different method with with a data train and data test ratio is 80:20.
Random Forest with SMOTE is the highest model performance compared with other model

Modelling With Imbalanced Data		
	Accuracy Train	Accuracy Test
Decision Tree	1.0000	0.9844
Random Forest	1.0000	0.9907
Logistic Regression	0.9486	0.9489

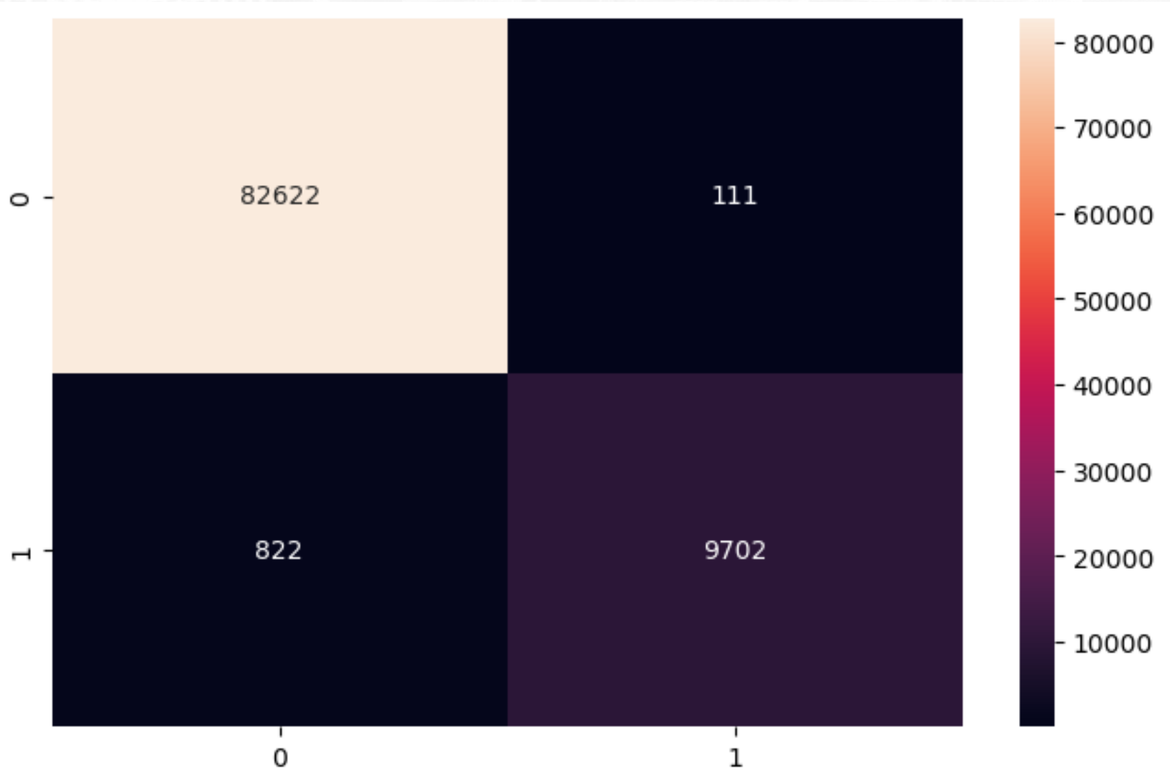
Modelling With SMOTE			
	Accuracy Train	Accuracy Test	Accuracy Validation
Decision Tree	1.0000	0.9894	0.9828
Random Forest	1.0000	0.9940	0.9894
Log Reg	0.8894	0.9123	0.9010

Modelling With Undersampling		
	Accuracy Train	Accuracy Test
Decision Tree	1.0000	0.9501
Random Forest	1.0000	0.9825
Logistic Regression	0.8249	0.8118

Random Forest with SMOTE Evaluation

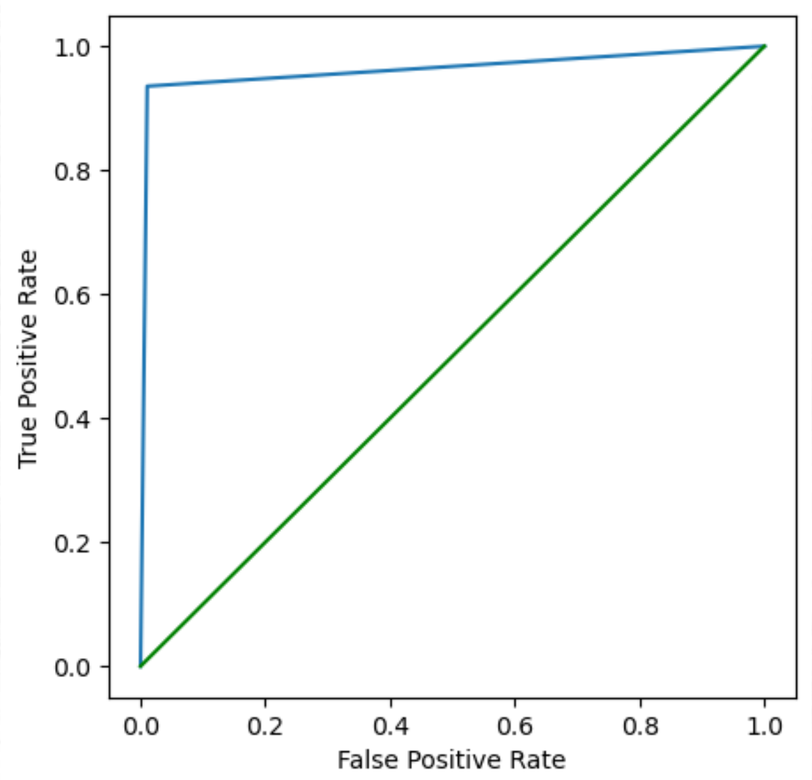
The model can predict as many as 82622 True negative cases and 9702 True positive

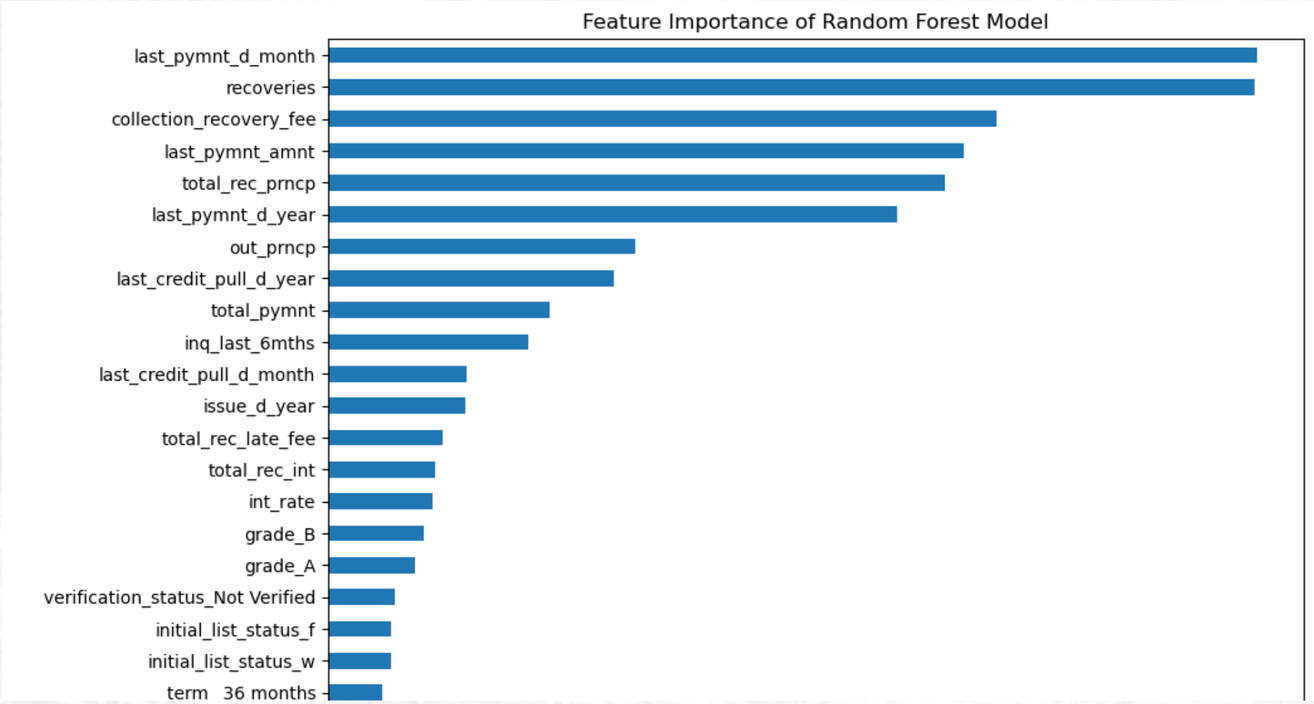
	precision	recall	f1-score	support
0	0.99	1.00	0.99	82733
1	0.99	0.92	0.95	10524
accuracy			0.99	93257
macro avg	0.99	0.96	0.97	93257
weighted avg	0.99	0.99	0.99	93257



Model Evaluation

ROC AUC Curve & Feature Importances





Conclusion & Recommendation

- Tree-Based Models can work well and produce good performance in the modeling process.
- Random Forest with SMOTE can predict 98,94% with data validation.
- Based on Feature Importances, last payment date (month), recoveries, and collection recovery fee is the biggest factor in predicting good or bad customers in making loan.
- This model can provide valuable insights and information for lenders to make informed lending decisions, while also helping to reduce the risk of default and improve the overall efficiency of the lending process.
- To improve the result, you can do hyperparameter tuning to model.
- You can use some important feature to reduce the computational run when train the model


Thank You