by michael sitanggang

CREDIT SCORING

ID/X - RAKAMIN

FINAL PROJECT

BUSINESS UNDERSTANDING

A credit score is based on credit history: number of open accounts, total levels of debt, repayment history, and other factors. Lenders use credit scores to evaluate the probability that an individual will repay loans in a timely manner

PROBLEM SOLUTION

- minimize the possibility of default
- determine potential target markets
- determine the attributes that influence the credit score

loan_amnt
funded_amnt
funded_amnt_inv
term
int_rate
installment
grade
sub_grade
emp_title
emp_length
home_ownership
annual_inc
verification_status
issue_d
loan_status
pymnt_plan
url
desc
purpose
title
zip_code
addr_state
dti
delinq_2yrs
earliest_cr_line
inq_last_6mths
mths_since_last_delinq
mths_since_last_record
open_acc
pub_rec
revol_bal
revol_util
total_acc
initial_list_status
out_prncp
out_prncp_inv

DATA UNDERSTANDING

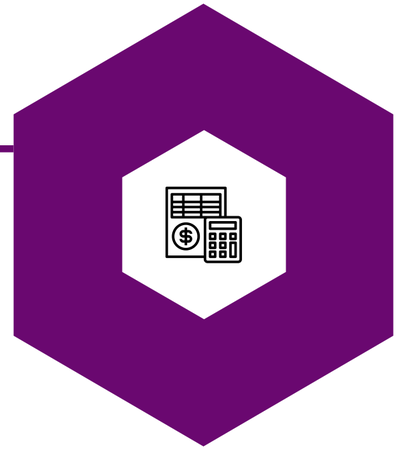
- 466285 records, 72 Columns
- Imbalanced Target Labels
- dtypes: float64(46), int64(4), object(22)

OPPORTUNITY TO ACCEPT LOAN APPLICATION

total_pymnt
total_pymnt_inv
total_rec_prncp
total_rec_int
total_rec_late_fee
recoveries
collection_recovery_fee
last_pymnt_d
last_pymnt_amnt
next_pymnt_d
last_credit_pull_d
collections_12_mths_ex_med
mths_since_last_major_derog
policy_code
application_type
annual_inc_joint
dti_joint
verification_status_joint
acc_now_delinq
tot_coll_amt
tot_cur_bal
open_acc_6m
open_il_6m
open_il_12m
open_il_24m
mths_since_rcnt_il
total_bal_il
il_util
open_rv_12m
open_rv_24m
max_bal_bc
all_util
total_rev_hi_lim
inq_fi
total_cu_tl
inq_last_12m

DATA PREPARATION

Percentage of missing value in each features



>10%

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	72.981546
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

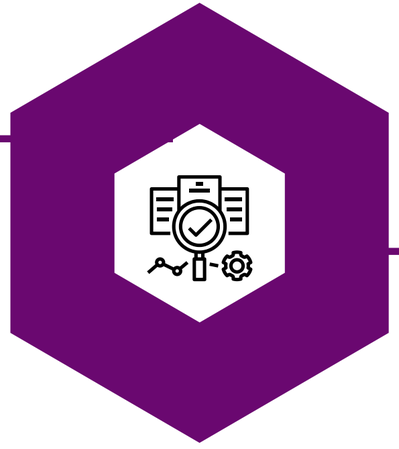
Cleaning if >=50% and imputation by median, mode

Convert data types

Diff data date-time

#Noted = difference calculated until July 2022

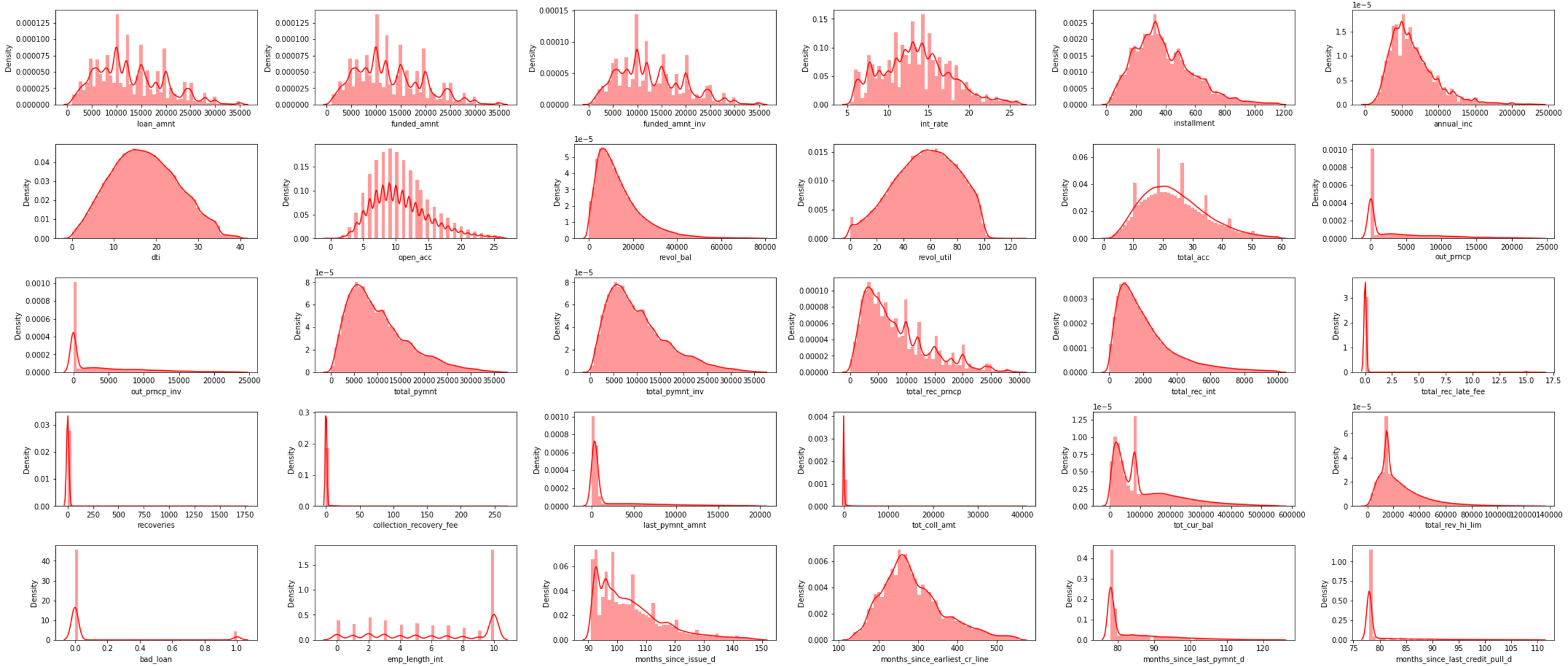
Delete Outlier



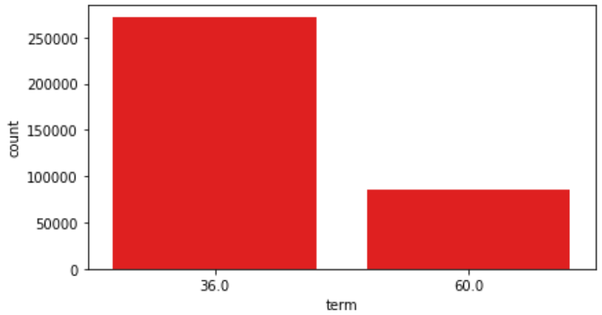
Label / Unique checking in each features

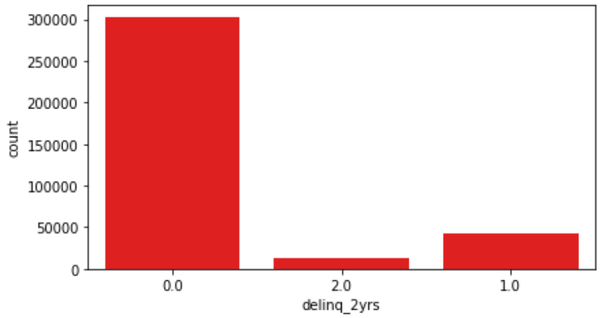
#Noted = found unreasonable categorical variables and single labels on several variables

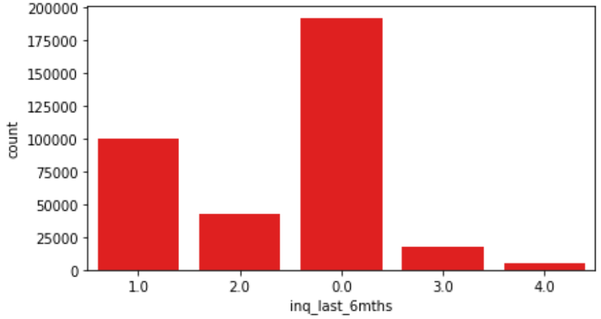
EXPLORATORY DATA ANALYSIS

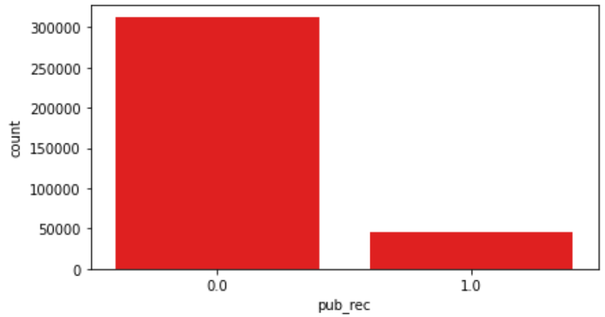


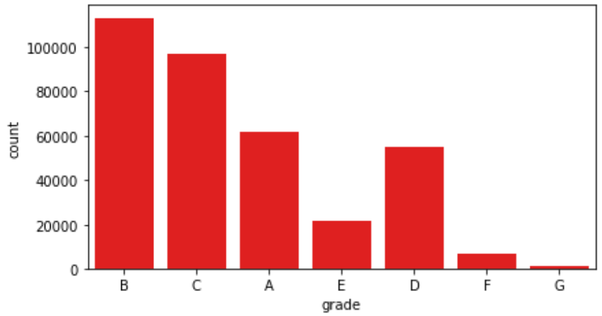
- dist plot for all numeric variables and it can be seen that some variables are skew and multimodal
- for variables with high variance and high bias will be converted into categorical form | Q1=Q2=Q3=0

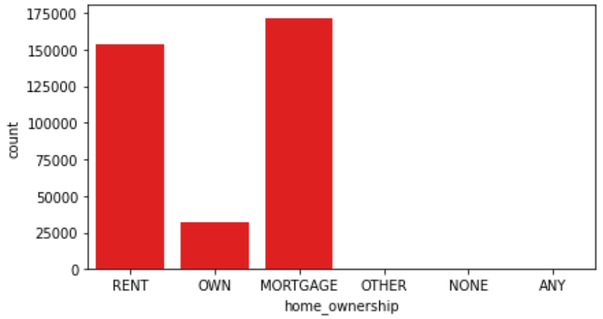


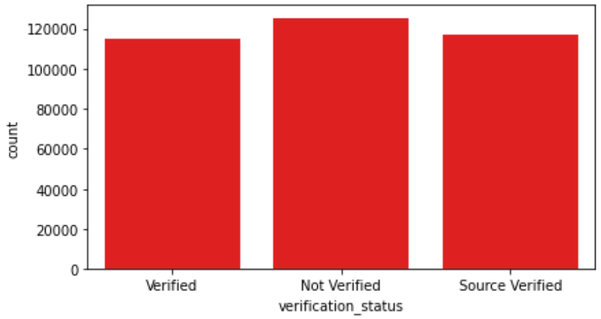


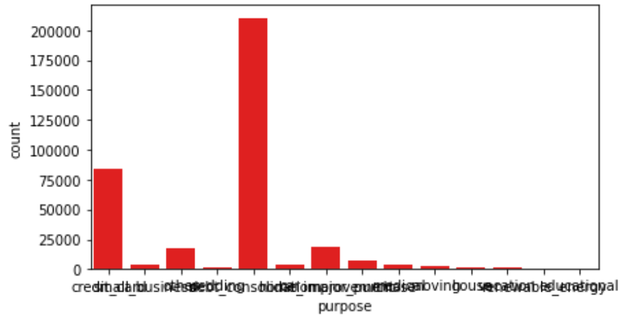


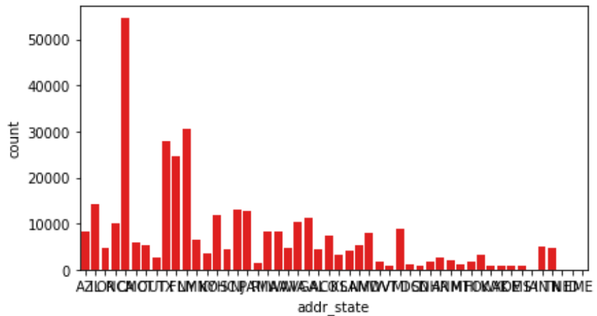


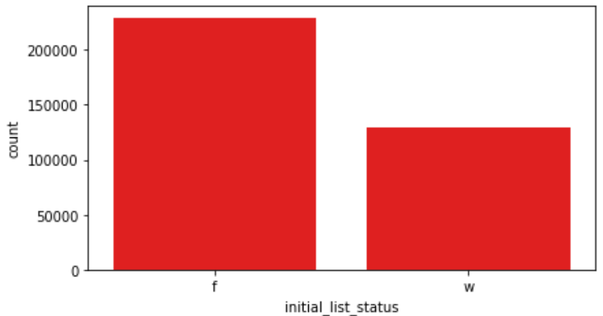


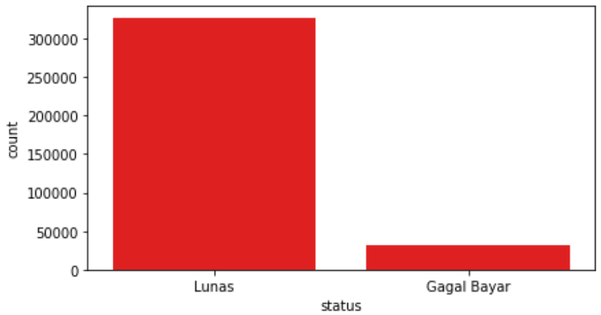








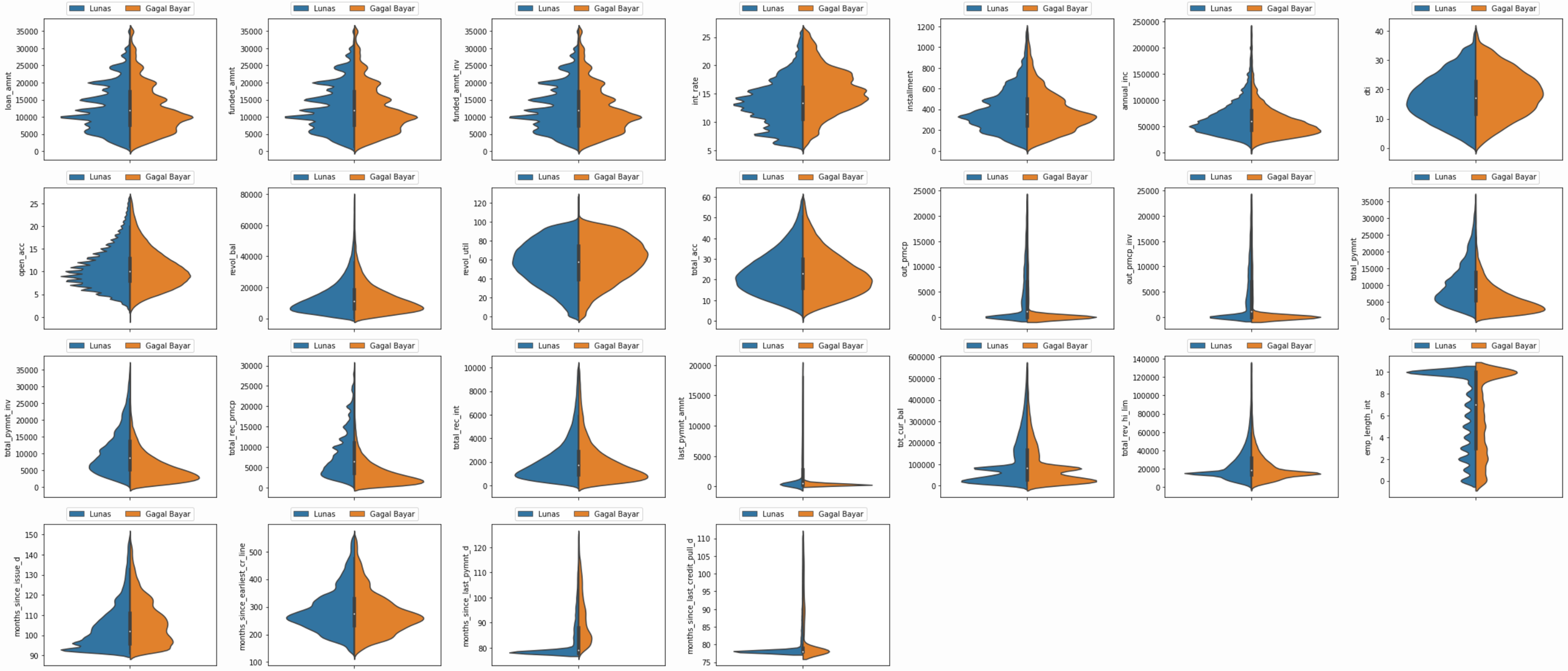




>>> It can be seen that the predictor variables with labels are not imbalanced and this will affect the accuracy of model, so a few labels are regrouped into several available labels or create other labels ;

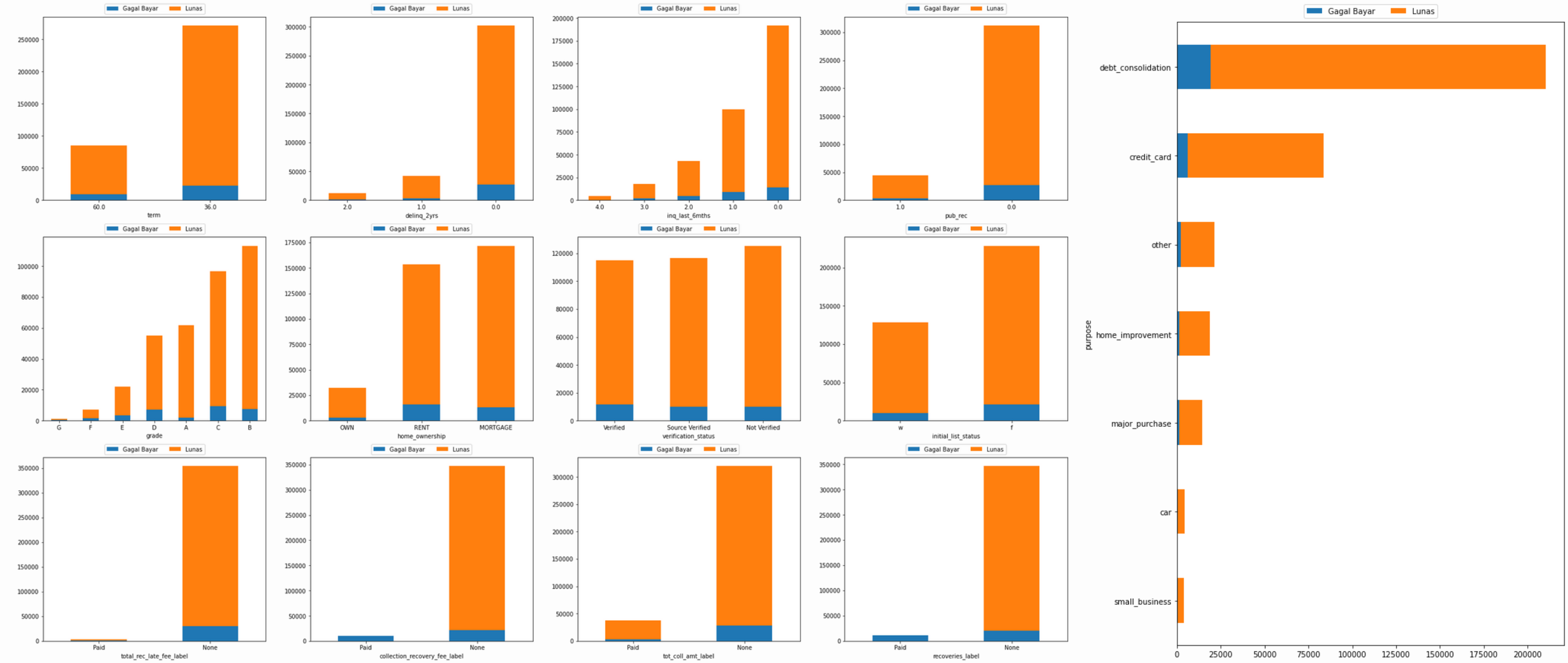
```
data['home_ownership'].replace({'NONE':'RENT', 'ANY':'RENT', 'OTHER':'RENT'},inplace=True)

data['purpose'].replace({'educational':'major_purchase',
                         'house':'major_purchase',
                         'medical':'major_purchase',
                         'moving':'major_purchase',
                         'vacation':'other',
                         'wedding':'other',
                         'renewable_energy':'home_improvement'},inplace=True)

data['addr_state'].replace({'IA':'OTHER', 'ID':'OTHER', 'NE':'OTHER', 'ME':'OTHER'},inplace=True)
```

Univariate Analysis - numeric variable



- dist plot for all numeric variables with paid and non-paid labels. The pattern of two labels so approaching
- there are several variables with high variance and high bias are maintained for log transformation

Univariate Analysis - categorical variable



- count plot for all categorical variables. There are some labels of imbalance variables and will be maintained
- for 4 plots in bottom are the conversion results from numeric variable

Bivariate Analysis - Multivariate Analysis

ANOVA TEST

Find that for all numerical predictor variables of significance < 0.05, so H0 is rejected, means that there is an average difference between the tested target labels. Conclusion = there is an effect of all numerical predictor variables on credit status.

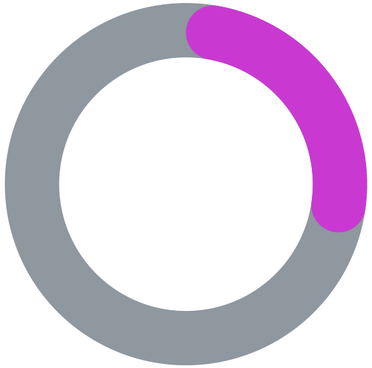
CHI SQUARED TEST

Find that almost all categorical predictor variables [except: pub_rec] sign < 0.05, so H0 is rejected, means that there is an average difference between the tested target labels. Conclusion = there is a categorical predictor variable has no effect on credit status and the variable will be deleted.

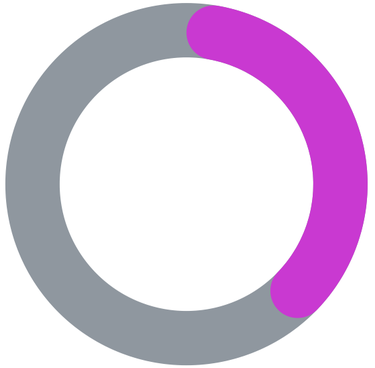
PEARSON CORR

Find that for corrrelation all numerical predictor variables and there are several variables that have a correlation < 0.7 and the variables will be deleted

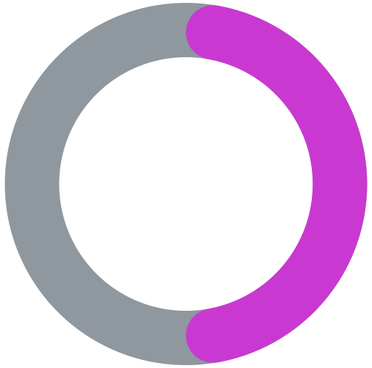
DATA PREPROCESSING



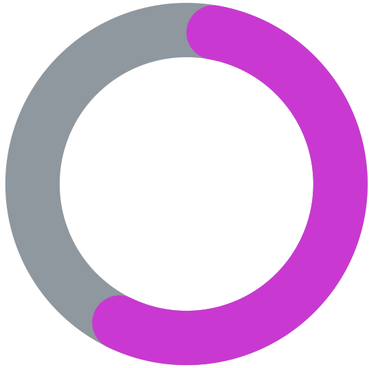
Label Encoding
&
One-Hot Encoding



Data Partitioning



Data Transformation



Undersampling for Imbalanced Target Labels

Training Model ML - Compare Tree Algorithm

```
RFclassifier = RandomForestClassifier()
RFclassifier.fit(X_train, y_train)
y_pred_rf = RFclassifier.predict(X_train)
y_pred_rf_test = RFclassifier.predict(X_test)

LRclassifier = LogisticRegression()
LRclassifier.fit(X_train, y_train)
y_pred_lr = LRclassifier.predict(X_train)
y_pred_lr_test = LRclassifier.predict(X_test)

XTclassifier = ExtraTreesClassifier()
XTclassifier.fit(X_train, y_train)
y_pred_xt = XTclassifier.predict(X_train)
y_pred_xt_test = XTclassifier.predict(X_test)

DTclassifier = DecisionTreeClassifier()
DTclassifier.fit(X_train, y_train)
y_pred_dt = DTclassifier.predict(X_train)
y_pred_dt_test = DTclassifier.predict(X_test)

GBclassifier = GradientBoostingClassifier()
GBclassifier.fit(X_train, y_train)
y_pred_gb = GBclassifier.predict(X_train)
y_pred_gb_test = GBclassifier.predict(X_test)
```

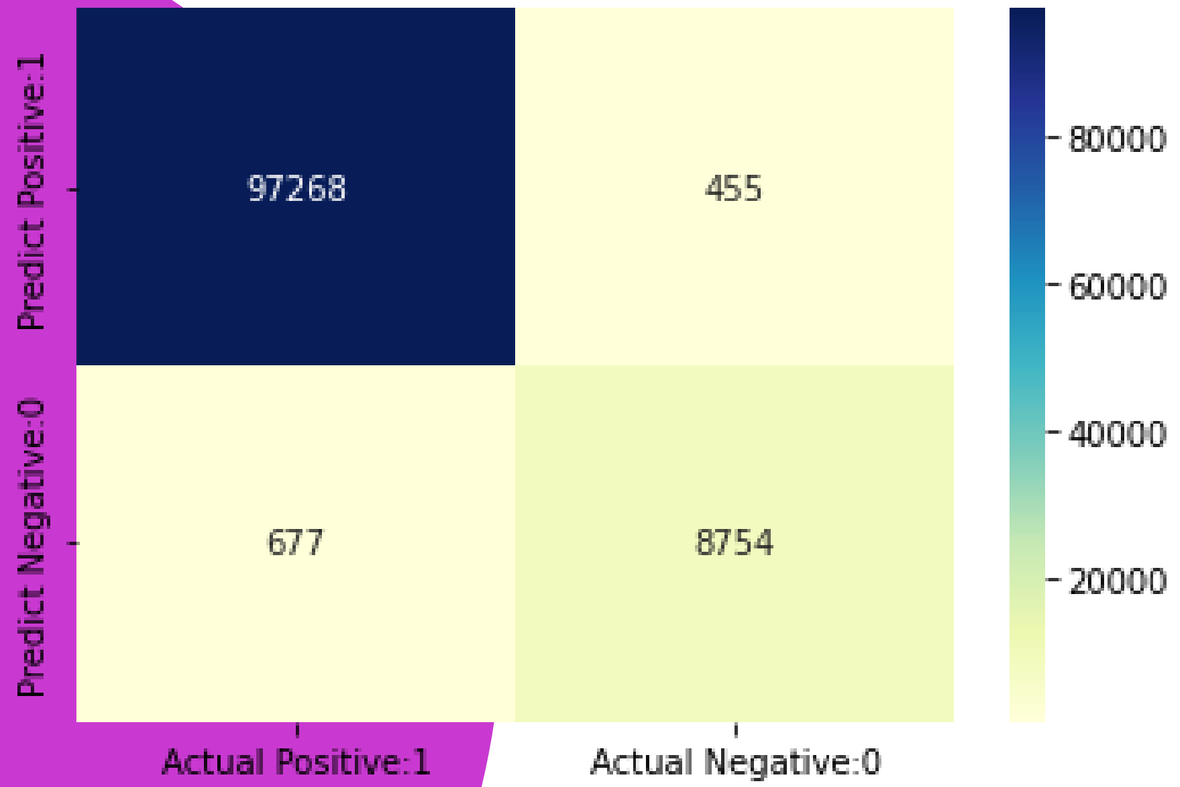
Model	Akurasi Train	Akurasi Test
RandomForest	1.000000	0.988540
Xtree	1.000000	0.976912
DecisionTree	1.000000	0.984462
GradientBoost	0.971550	0.975531
LogisticRegression	0.892719	0.936587

then choose the important variable and underfit or overfit test

```
Akurasi Train 1.0
Akurasi Test 0.9894357653470706
```

Model is not prone to underfit or overfit.

Model Evaluation

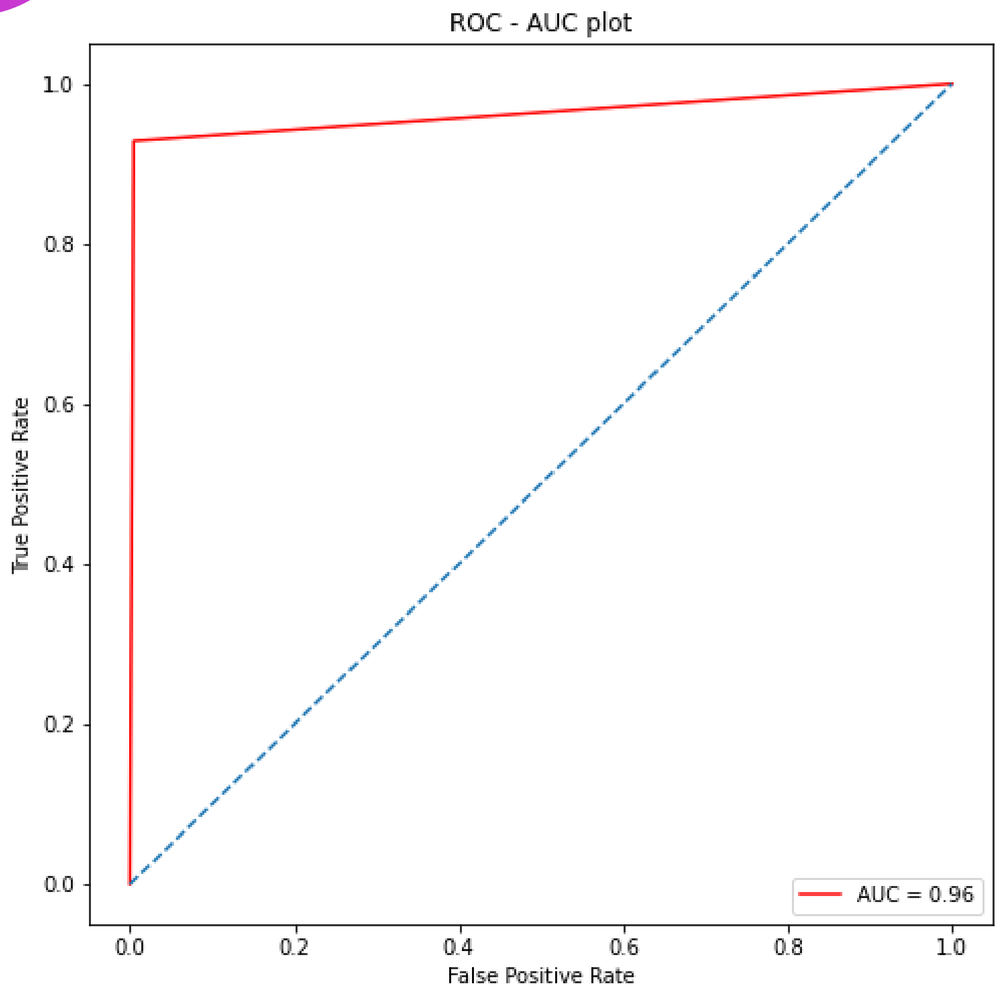


Akurasi Klasifikasi : 0.9894

Kesalahan Klasifikasi : 0.0106

Presisi : 0.9953

Sensitivitas : 0.9931



ROC- AUC more than 0.9 is considered outstanding classifier