# LOAN DEFAULT MODELING

PHASE I REPORT

MINIMUM VIABLE PRODUCT

# STRATEGY

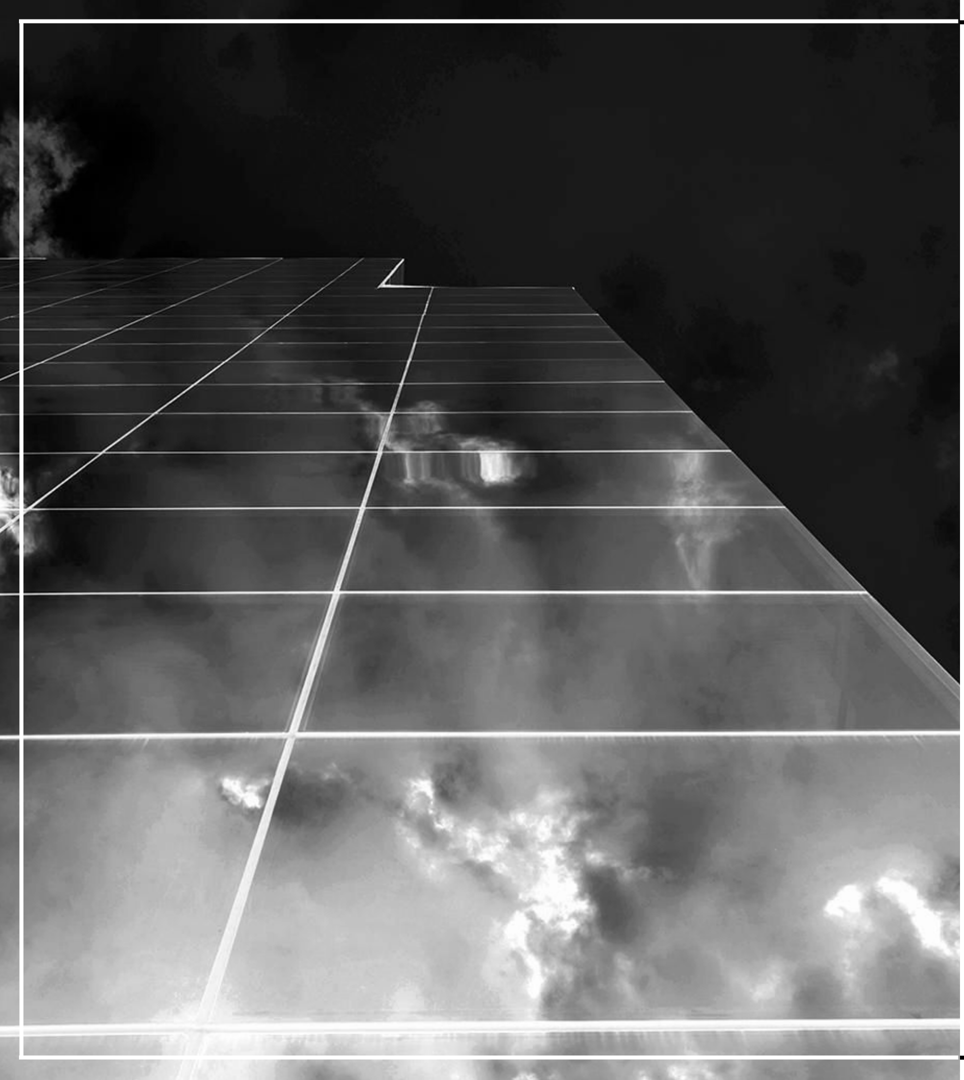
## BUSINESS INTEREST

### PHASE I

The current status of the initiative and SWOT Report

### PHASE II

Exploring the possible enhancements and goal setting



Lending Club is a peer to peer lending company based in the United States, in which investors provide funds for potential borrowers and investors earn a profit depending on the risk they take (the borrowers credit score). The company also registers its offerings as securities with the Securities and Exchange Commission (SEC), and to offer loan trading on a secondary market. Lending Club provides the "bridge" between investors and borrowers

# LENDING CLUB BUSINESS MODEL

# BUSINESS INTEREST

The client does not want to miss-classify prospective applicants during their pre-screening process

Operational risk management by early detection of applicants that could be considered at risk of default down the line

# THREE PILLAR APPROACH

## IDENTIFY

The ability to capture and create a risk profile of borrowers

## INNOVATE

Creating a product that will be able to manage the risk appetite of Lending Club

## LISTENING

Understanding the customer journey to increase revenue and increase customer inclusion

# PHASE I

## THE DATA

Asset (dataset) Overview and Understanding

## MODELING

Current state of the modeling process

## QUICK STATS

The current landscape of the asset

## SWOT

Analysis of the Phase 1 of the initiative

# THE DATA

The data was sourced via Kaggale

**ASSET ACQUISITION**

**ASSET REDUCTION**

2 Mil observation with 151 feature variables

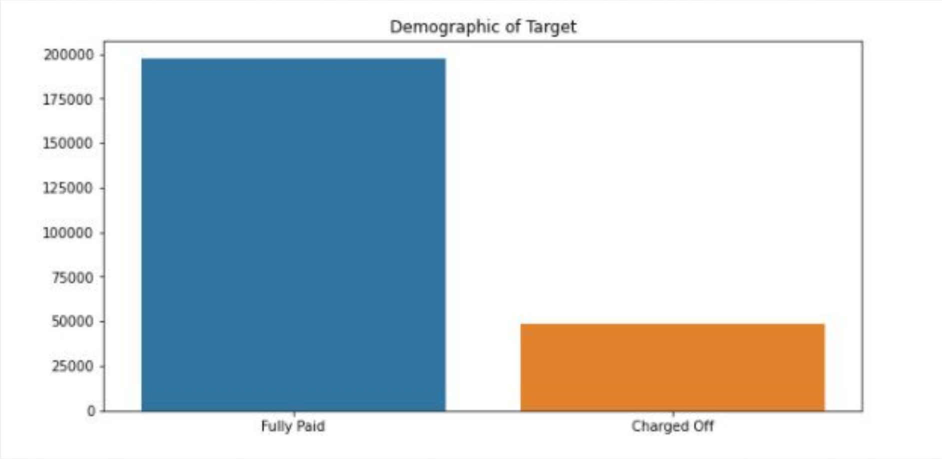
1 Mil observations with 93 feature variables

**REDUCTION 2**

**FINAL ASSET**

246,269 observations with 27 features

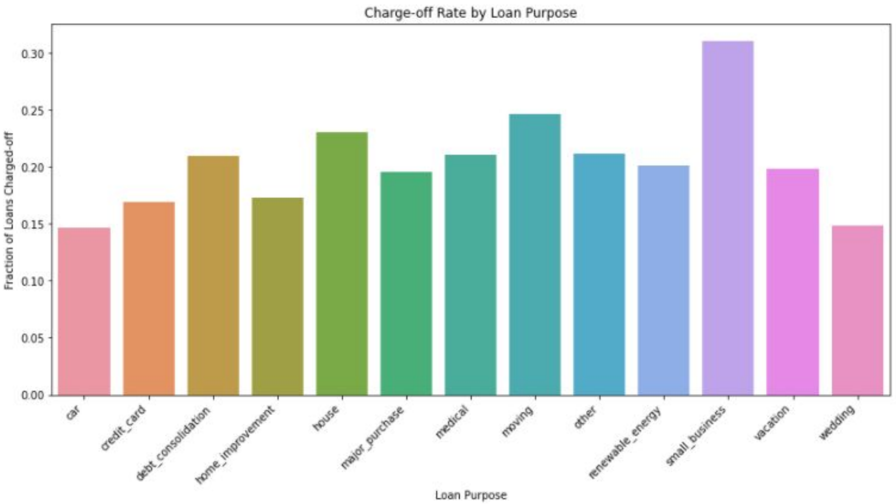
- ❏ Reduced the asset from missing values that were over 30% of the asset.
- ❏ Multiple versioning of the assets for data management.
- ❏ Assets created for business intelligence analysis
- ❏ Data preparation initiated for pre-screening rejection analysis




```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 246269 entries, 0 to 246268
Data columns (total 27 columns):
 #   Column                Non-Null Count   Dtype
---  ------                --------------   -----
 0   loan_amnt             246269 non-null  float64
 1   term                  246269 non-null  object
 2   int_rate              246269 non-null  float64
 3   installment           246269 non-null  float64
 4   grade                 246269 non-null  object
 5   sub_grade             246269 non-null  object
 6   emp_length            246269 non-null  int64
 7   home_ownership        246269 non-null  object
 8   annual_inc            246269 non-null  float64
 9   verification_status   246269 non-null  object
 10  issue_d               246269 non-null  object
 11  loan_status           246269 non-null  object
 12  purpose               246269 non-null  object
 13  addr_state            246269 non-null  object
 14  dti                   246269 non-null  float64
 15  earliest_cr_line      246269 non-null  int64
 16  open_acc              246269 non-null  float64
 17  pub_rec               246269 non-null  float64
 18  revol_bal             246269 non-null  float64
 19  revol_util            246269 non-null  float64
 20  initial_list_status   246269 non-null  object
 21  application_type      246269 non-null  object
 22  tot_cur_bal           246269 non-null  float64
 23  mort_acc              246269 non-null  float64
 24  pub_rec_bankruptcies  246269 non-null  float64
 25  annual_inc_log        246269 non-null  float64
 26  fico_score            246269 non-null  float64
dtypes: float64(14), int64(2), object(11)
memory usage: 50.7+ MB
```

## QUICK STATS:

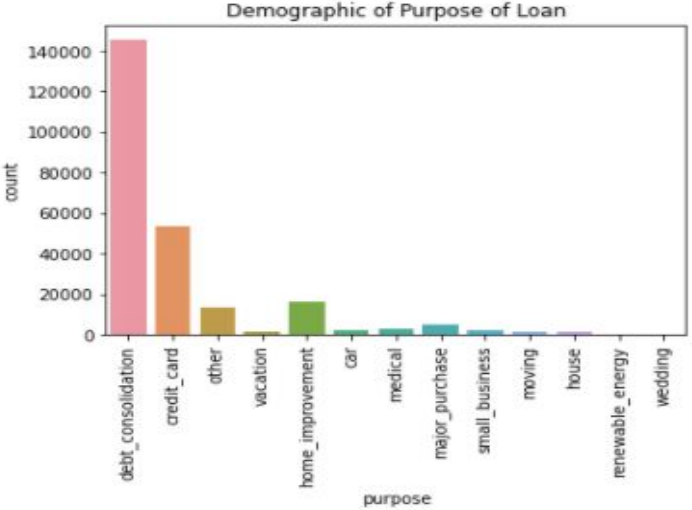
- ❏ RESPONSE VARIABLE
- ❏ PURPOSE
- ❏ HOME OWNERSHIP
- ❏ LOAN AMOUNT



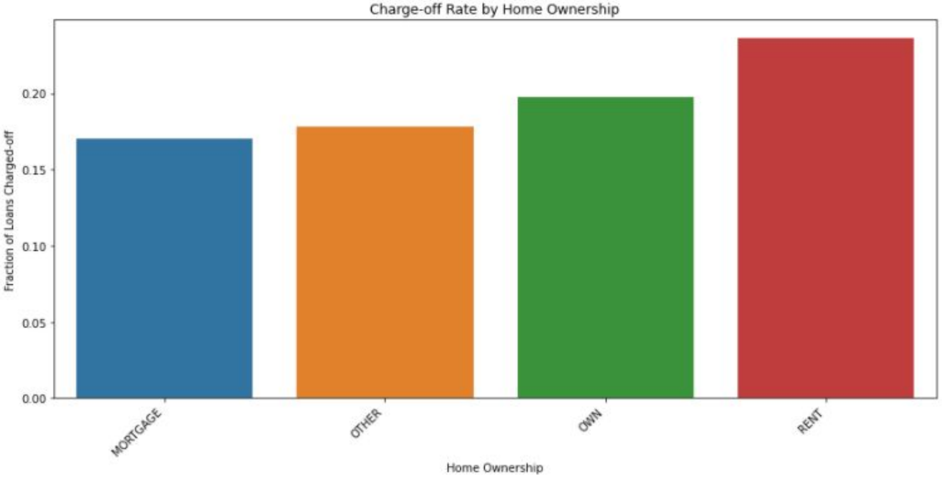

```
             count unique                top    freq
loan_status
Charged Off  48883     13  debt_consolidation   30452
Fully Paid  197386     13  debt_consolidation  115145
```

# PURPOSE OF THE LOAN

```
debt_consolidation    145597
credit_card            53371
home_improvement       16346
other                  13382
major_purchase          5160
medical                 2803
small_business          2399
car                     2390
vacation                1596
moving                  1557
house                   1360
wedding                  169
renewable_energy         139
Name: purpose, dtype: int64
```



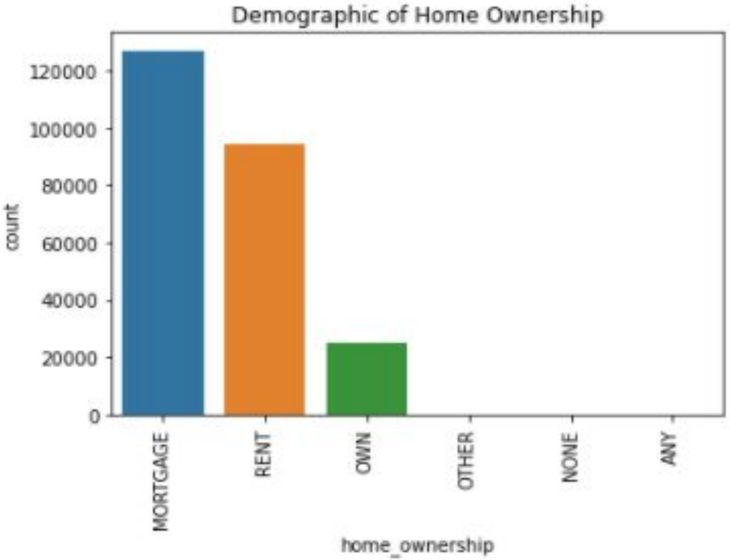

```
count                 246269
unique                    13
top       debt_consolidation
freq                  145597
Name: purpose, dtype: object
```



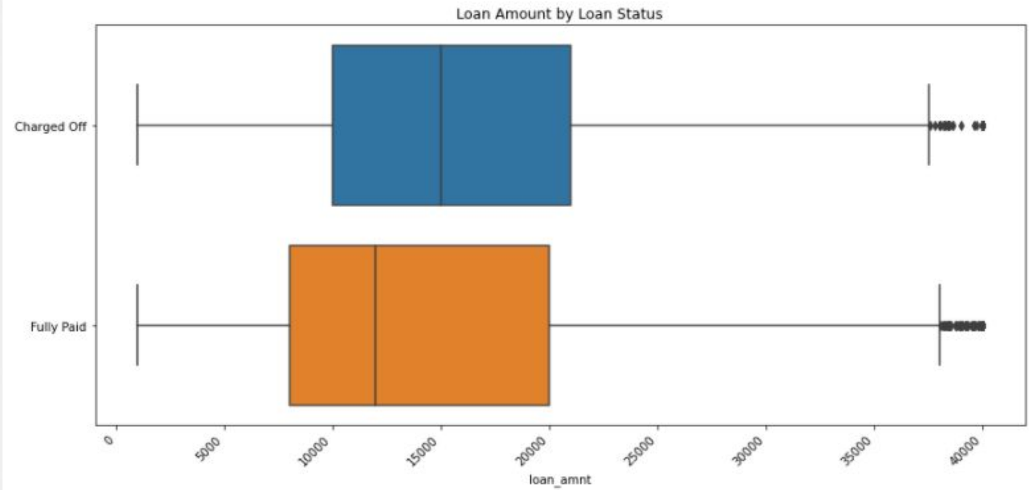

```
             count unique       top    freq
loan_status
Charged Off   48883      4      RENT   22234
Fully Paid   197386      4  MORTGAGE  105294
```

# HOME OWNERSHIP

```
MORTGAGE    126956
RENT         94084
OWN          25156
ANY             53
OTHER           11
NONE             9
Name: home_ownership, dtype: int64
```

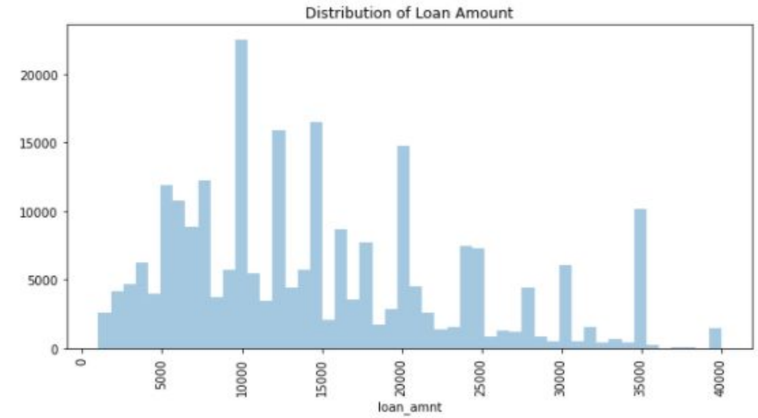


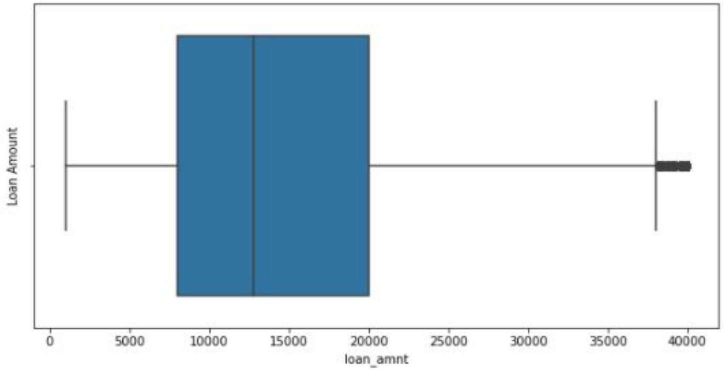

```
count        246269
unique            6
top        MORTGAGE
freq         126956
Name: home_ownership, dtype: object
```




```
              count          mean          std     min      25%      50%  \
loan_status
Charged Off   48883.0  16091.673997  8826.185411  1000.0  10000.0  15000.0
Fully Paid   197386.0  14534.892419  8756.550535  1000.0   8000.0  12000.0

                 75%      max
loan_status
Charged Off  21000.0  40000.0
Fully Paid   20000.0  40000.0
```

# LOAN AMOUNT



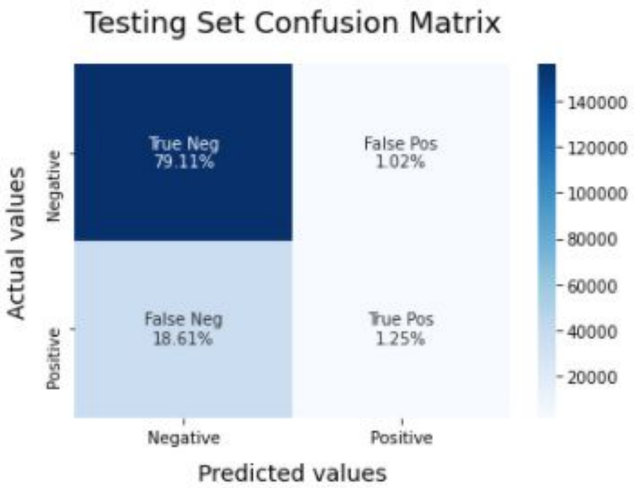

```
count    246269.000000
mean      14843.904734
std        8792.352950
min        1000.000000
25%        8000.000000
50%       12800.000000
75%       20000.000000
max       40000.000000
Name: loan_amnt, dtype: float64
```

# MODELING

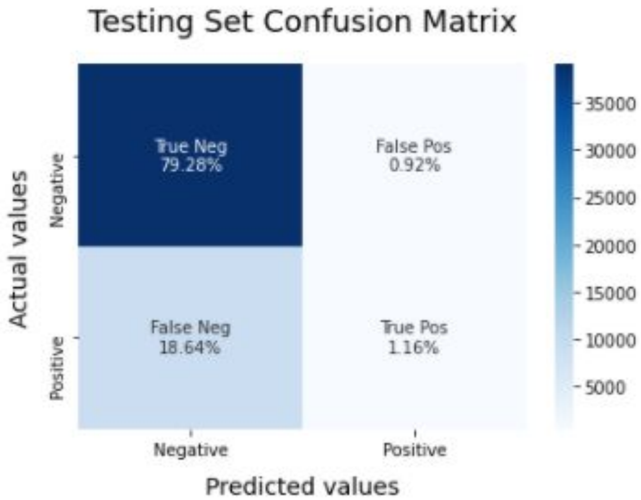
SK LEARN

PYCARET

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.81 | 0.99 | 0.89 | 157884 |
| 1 | 0.55 | 0.06 | 0.11 | 39131 |
| accuracy | | | 0.80 | 197015 |
| macro avg | 0.68 | 0.53 | 0.50 | 197015 |
| weighted avg | 0.76 | 0.80 | 0.74 | 197015 |




| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.81 | 0.99 | 0.89 | 39502 |
| 1 | 0.56 | 0.06 | 0.11 | 9752 |
| accuracy | | | 0.80 | 49254 |
| macro avg | 0.68 | 0.52 | 0.50 | 49254 |
| weighted avg | 0.76 | 0.80 | 0.73 | 49254 |




# LOGISTIC REGRESSION

## 7.2 Performance Indicators and Model Comparison

We will prioritize Recall, Accuracy, and AUC

Benchmark indicators:

- Recall: should be above .45
- Accuracy: equal or above .60
- AUC: should be equal or above .50/.60, ideally over .70 for better balance in predicting our positive class

For the purpose of this version of the initiative we will consider two out of the three benchmark indicators as a green light for model selection. Also, in consideration, if either one metric stands out we will raise the question, if we should include it to our model selection list.

# PYCARET MODEL SELECTION

| | Model | Accuracy | AUC | Recall | Prec. | F1 | Kappa | MCC | TT (Sec) |
|---|---|---|---|---|---|---|---|---|---|
| **qda** | Quadratic Discriminant Analysis | 0.1983 | 0.5000 | 0.9997 | 0.1982 | 0.3308 | -0.0000 | -0.0017 | 5.6410 |
| **nb** | Naive Bayes | 0.4492 | 0.6587 | 0.8148 | 0.2391 | 0.3697 | 0.0911 | 0.1478 | 1.9780 |
| **lr** | Logistic Regression | 0.6008 | 0.6640 | 0.6449 | 0.2809 | 0.3910 | 0.1584 | 0.1883 | 7.8160 |
| **svm** | SVM - Linear Kernel | 0.6143 | 0.0000 | 0.4326 | 0.2928 | 0.2586 | 0.0780 | 0.0972 | 18.2330 |
| **knn** | K Neighbors Classifier | 0.5768 | 0.5273 | 0.4242 | 0.2139 | 0.2844 | 0.0283 | 0.0316 | 6.6440 |
| **dt** | Decision Tree Classifier | 0.7023 | 0.5508 | 0.2997 | 0.2722 | 0.2853 | 0.0979 | 0.0981 | 5.1950 |
| **ada** | Ada Boost Classifier | 0.7717 | 0.6637 | 0.2057 | 0.3651 | 0.2630 | 0.1403 | 0.1484 | 16.9730 |
| **gbc** | Gradient Boosting Classifier | 0.7904 | 0.6920 | 0.1745 | 0.4295 | 0.2481 | 0.1509 | 0.1717 | 73.2410 |
| **et** | Extra Trees Classifier | 0.7944 | 0.6776 | 0.1140 | 0.4299 | 0.1802 | 0.1059 | 0.1368 | 54.2830 |
| **catboost** | CatBoost Classifier | 0.8036 | 0.7162 | 0.1131 | 0.5215 | 0.1858 | 0.1239 | 0.1718 | 85.6450 |
| **xgboost** | Extreme Gradient Boosting | 0.8023 | 0.7105 | 0.1113 | 0.5058 | 0.1824 | 0.1195 | 0.1647 | 94.7230 |
| **lightgbm** | Light Gradient Boosting Machine | 0.8039 | 0.7163 | 0.0891 | 0.5320 | 0.1527 | 0.1015 | 0.1551 | 6.2130 |
| **rf** | Random Forest Classifier | 0.8012 | 0.6943 | 0.0760 | 0.4906 | 0.1316 | 0.0828 | 0.1305 | 42.8570 |
| **ridge** | Ridge Classifier | 0.7768 | 0.0000 | 0.0493 | 0.2195 | 0.0804 | 0.0084 | 0.0115 | 2.1820 |
| **lda** | Linear Discriminant Analysis | 0.7768 | 0.6699 | 0.0486 | 0.2180 | 0.0795 | 0.0077 | 0.0106 | 11.7650 |

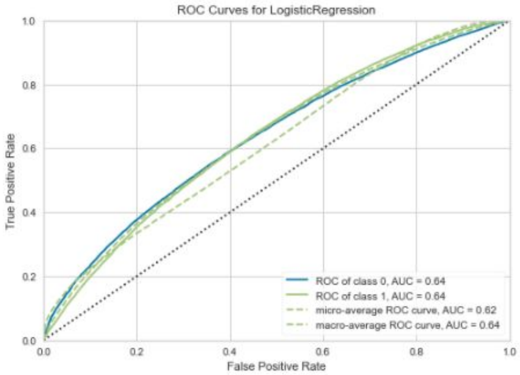
```python
lr = create_model('lr')
```

| | Accuracy | AUC | Recall | Prec. | F1 | Kappa | MCC |
|---|---|---|---|---|---|---|---|
| 0 | 0.6345 | 0.6918 | 0.6465 | 0.3026 | 0.4122 | 0.1948 | 0.2244 |
| 1 | 0.5611 | 0.6334 | 0.6468 | 0.2579 | 0.3687 | 0.1191 | 0.1488 |
| 2 | 0.6047 | 0.6718 | 0.6608 | 0.2853 | 0.3986 | 0.1683 | 0.2013 |
| 3 | 0.6264 | 0.6761 | 0.6289 | 0.2935 | 0.4003 | 0.1781 | 0.2054 |
| 4 | 0.5660 | 0.6441 | 0.6538 | 0.2618 | 0.3739 | 0.1267 | 0.1579 |
| 5 | 0.5927 | 0.6556 | 0.6438 | 0.2748 | 0.3852 | 0.1487 | 0.1789 |
| 6 | 0.5962 | 0.6618 | 0.6415 | 0.2765 | 0.3864 | 0.1513 | 0.1812 |
| 7 | 0.5697 | 0.6412 | 0.6544 | 0.2639 | 0.3761 | 0.1305 | 0.1620 |
| 8 | 0.6292 | 0.6846 | 0.6444 | 0.2984 | 0.4079 | 0.1879 | 0.2174 |
| 9 | 0.6272 | 0.6793 | 0.6277 | 0.2938 | 0.4003 | 0.1784 | 0.2055 |
| **Mean** | 0.6008 | 0.6640 | 0.6449 | 0.2809 | 0.3910 | 0.1584 | 0.1883 |
| **SD** | 0.0268 | 0.0189 | 0.0100 | 0.0154 | 0.0142 | 0.0256 | 0.0249 |

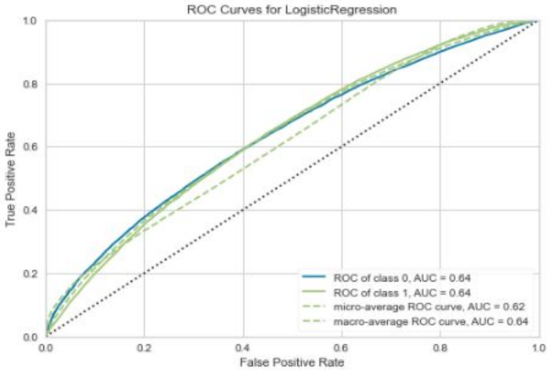
```python
tuned_lr = tune_model(lr, optimize='Recall')
```

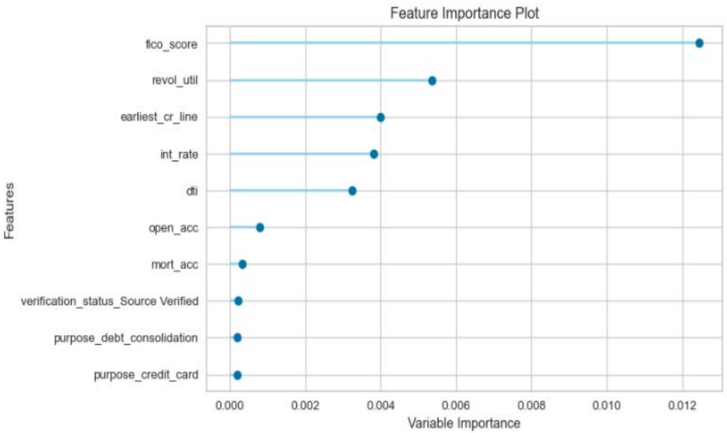
| | Accuracy | AUC | Recall | Prec. | F1 | Kappa | MCC |
|---|---|---|---|---|---|---|---|
| 0 | 0.6347 | 0.6918 | 0.6465 | 0.3026 | 0.4123 | 0.1949 | 0.2245 |
| 1 | 0.5611 | 0.6334 | 0.6468 | 0.2579 | 0.3687 | 0.1191 | 0.1488 |
| 2 | 0.6563 | 0.7050 | 0.6339 | 0.3167 | 0.4223 | 0.2148 | 0.2410 |
| 3 | 0.5888 | 0.6511 | 0.6392 | 0.2716 | 0.3813 | 0.1428 | 0.1722 |
| 4 | 0.5660 | 0.6441 | 0.6538 | 0.2618 | 0.3739 | 0.1267 | 0.1579 |
| 5 | 0.5927 | 0.6556 | 0.6438 | 0.2748 | 0.3852 | 0.1487 | 0.1789 |
| 6 | 0.5950 | 0.6610 | 0.6441 | 0.2763 | 0.3867 | 0.1513 | 0.1815 |
| 7 | 0.5697 | 0.6412 | 0.6544 | 0.2639 | 0.3761 | 0.1305 | 0.1620 |
| 8 | 0.6293 | 0.6846 | 0.6444 | 0.2985 | 0.4080 | 0.1880 | 0.2175 |
| 9 | 0.5952 | 0.6616 | 0.6444 | 0.2765 | 0.3869 | 0.1515 | 0.1819 |
| **Mean** | 0.5989 | 0.6630 | 0.6451 | 0.2801 | 0.3901 | 0.1568 | 0.1866 |
| **SD** | 0.0301 | 0.0223 | 0.0058 | 0.0184 | 0.0170 | 0.0302 | 0.0292 |

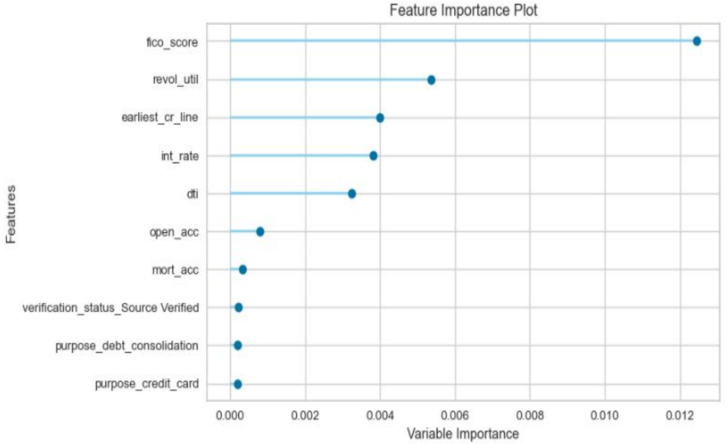
```python
#base model ROC

plot_model(lr, plot = 'auc')
```




```python
# Tuned ROC

plot_model(tuned_lr, plot = 'auc')
```




# LOGISTIC REGRESSION

# PHASE 1 SWOT ANALYSIS

## STRENGTHS

- ❏ Business Analysis
- ❏ Modeling
- ❏ Initiative Workflow and Process Management
- ❏ Leveraging External dependencies, when needed

## OPPORTUNITY

- ❏ Modeling Process and Workflow Management Enhancements
- ❏ Business Intelligence Dashboard

## WEAKNESSES

- ❏ Computing Power and Resources
- ❏ More Efficient Model Process Enhancement

## THREATS

- ❏ Resource Miss-Management
- ❏ Deadline Awareness

# PHASE 2 PRIORITY GOALS

## GOAL 1 (P1)

Complete SKLearn Modeling with additional metrics and analysis

## GOAL 2 (P2)

Finalize project analysis report for business stakeholder presentation

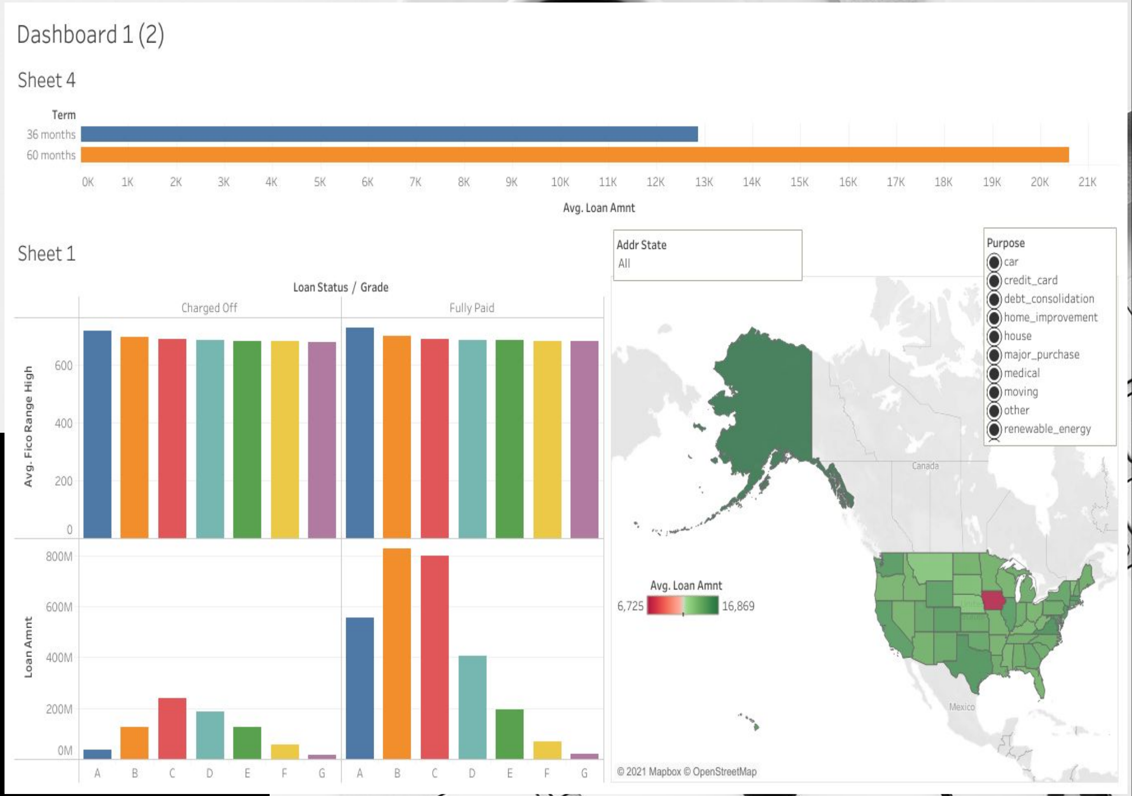
## GOAL 3 (P3)

User acceptance testing (UAT) of Tableau business intelligence dashboards for key stakeholders

## GOAL 4 - LOW PRIORITY

Front-end user interface for model deployment

# TABLEAU ANALYTICS DASHBOARD




## BUSINESS INTELLIGENCE FEATURES:

- ❏ BUSINESS PROFILE
- ❏ CUSTOMER ANALYTICS
- ❏ APPLICATION PRE-SCREEN
    - ❏ APPLICANT REJECTION ANALYTICS
    - ❏ INCLUSION OPTIMIZATION ANALYSIS
- ❏ PROJECT BENCHMARK ANALYSIS

# THANKS

Christian Corrales
Business Intelligence Analyst
(Role title for academic project)

corrales.christian1228@gmail.com
https://github.com/ccorrales1228/Capstone