Springboard Capstone I

Loan Eligibility Predictive Model

by

Vaishali Kalekar

Table of Contents

Purpose

Aim is to build a loan eligibility predictive model to predict if a person will be approved a loan. Loan companies could use this predictive analysis to automate their decision making process.

Data Wrangling / Cleaning

Data Acquisition

The training and test data for the model is acquired from kaggle.
https://www.kaggle.com/wendykan/lending-club-loan-data#loan.csv

Storage and access

The data is stored locally as well as in the project's Github repository. Through the jupyter notebook, the .csv data file is opened and data is loaded in pandas dataframe.

Description of Data

Once the data is loaded into dataframe, we see that the data set for this project consists of 2260668 observations and 145 features. Loan_status is our target feature and has multiple values. We will be focusing only on loan applications that are 'Fully Paid' or 'Charged Off' and not those that are active or delayed in payments for a certain period of time. About 80% are 'Fully Paid' and 20% are 'Charged Off'. So our sample is biased in this case.

Data Types

Further investigation reveals that the data types in the raw data has some issues. Some features have data values/ data types combinations that may be viewed as less efficient. For performance and efficiency purposes, we will be cleaning up those data values. For e.g. cleaning up the employment length feature by removing 'years' and '+' strings and then converting it into integer. The values are -10+ years, < 1 year, 2 years, 3 years etc. We will be storing these as 10, 0, 2, 3 respectively. We will also be formatting data from Oct-2013 to 2015-10-01 format. Also the term has values '36 Months' and '60 Months'. We will be changing those to 36 and 60 respectively.

Missing Values

We see there are 56 out of 143 features have 50% or more missing values. It will be difficult to project the missing values, so we will drop all those features with a missing percentage >50%. For e.g. few of those features are-

Index	Name	Total Missing Values	Percentage %
0	url	1303607	100.0
1	next_pymnt_d	1303607	100.0
2	orig_projected_additional_accrued_interest	1300174	99.74
3	deferral_term	1298272	99.59
4	hardship_last_payment_amount	1298272	99.59
5	hardship_payoff_balance_amount	1298272	99.59
6	hardship_end_date	1298272	99.59
7	hardship_dpd	1298272	99.59

Next we drop the features that Lending Club added as part of processing the loan application. We will only concentrate on features that the loan applicant provides as part of the loan application.

Outliers

We figure out the outliers with The interquartile range (IQR) approach, also called the midspread or middle 50%. It is a measure of the dispersion similar to standard deviation or variance, but is much more robust against outliers. The interquartile range shows how the data is spread about the median. IQR is somewhat similar to Z-score in terms of finding the distribution of data and then keeping some threshold to identify the outlier. Next we remove the outliers identified by the Interquartile Rule for Outliers.

EDA and Feature Analysis

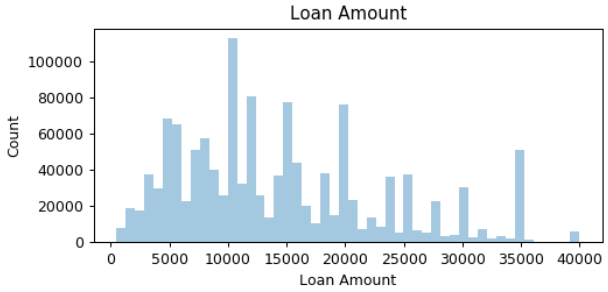
EDA

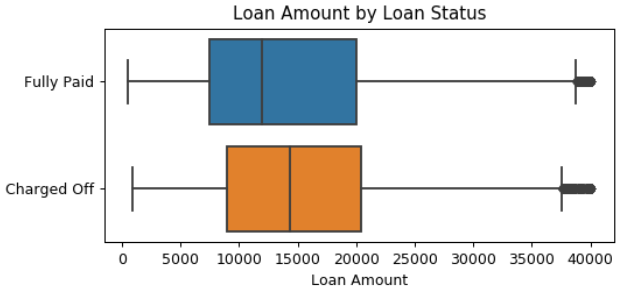
Univariate and Bivariate Feature Analysis

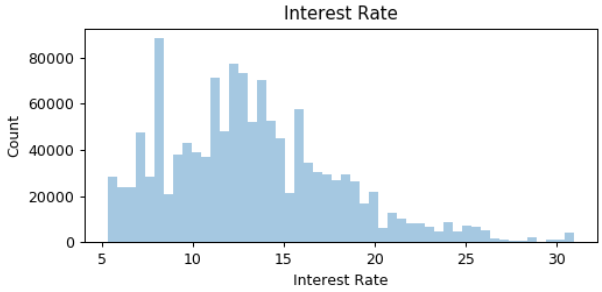
We inspect each feature individually. We will perform univariate and bivariate analysis using the following steps-

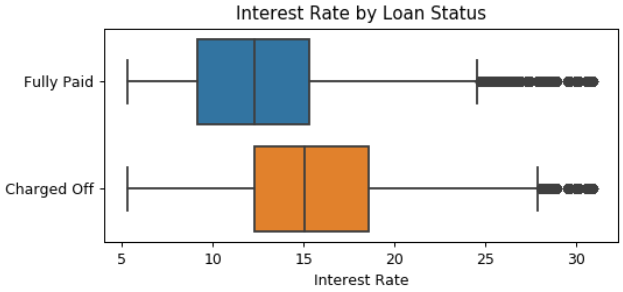
1. Summary statistics
2. Plot individually
3. Plot against our target variable- loan_status
4. Modify the feature to make it ready for modelling, if necessary

Upon completion of feature analysis as mentioned above, we make some interesting discoveries such as out of all charged off loans, 5 year loans are almost **twice** as likely to go **bad** as 3 year loans! charged off loans have a higher loan amount. int_rate ranges from 5 to 25%! charged off loans have higher installments!





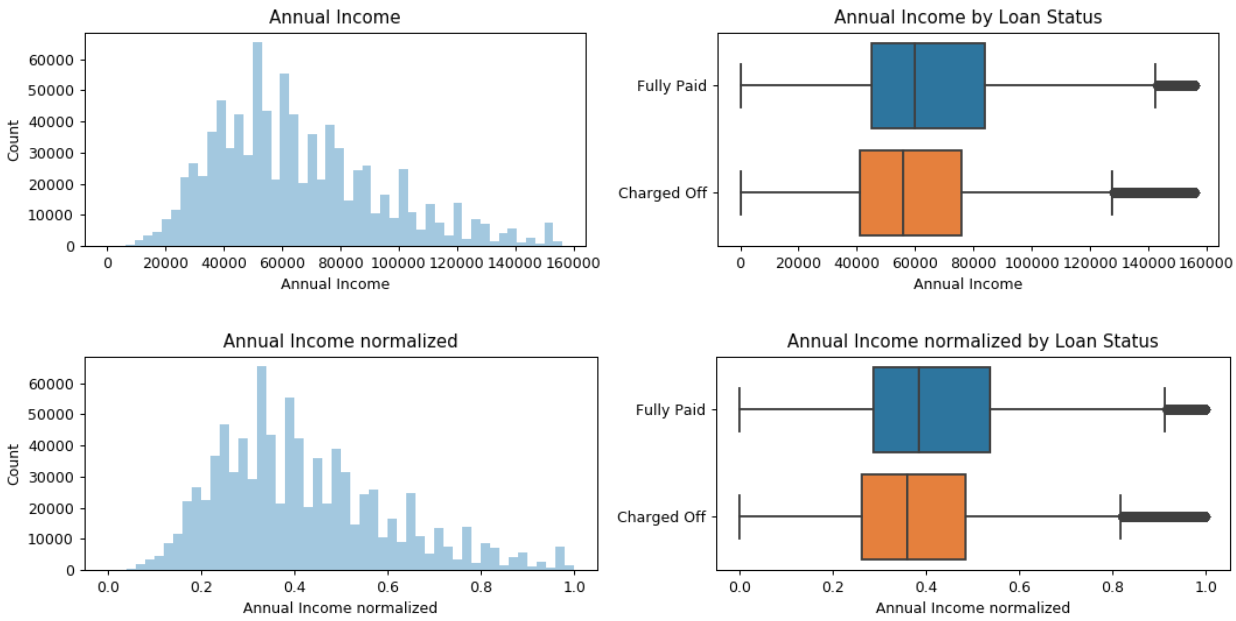

Interest Rate
Count
80000
60000
40000
20000
0
5
10
15
20
25
30
Interest Rate


Interest Rate by Loan Status
Fully Paid
Charged Off
5
10
15
20
25
30
Interest Rate

Feature Engineering and Feature Selection

Feature Scaling

From the univariate analysis of annual_income, we see that values range from 0 to 165k with a median of 60,000. Since the values vary so much we will transform the data using minmax scaler. We will create a new feature called 'annual_inc_norm' and store these values there and then drop the feature annual_inc. This will scale the annual income data between 0 and 1.



From the two plots above we see that for both raw data and scaled data the underlying distribution has remained the same.

Feature Selection

Pearson Correlation

We find Pearson correlation coefficients between pairs of features from our feature set. Correlation could be either positive or negative. We set the threshold of 0.3 and remove a feature where correlation coefficient > 0.3.

RFE

Here we determined which predictor variables will contribute the most to the predictive power of a machine learning algorithm. We used RFE here. The Recursive Feature Elimination (or RFE) works by recursively removing attributes and building a model on those attributes that remain.

It uses the model accuracy to identify which attributes (and combination of attributes) contribute the most to predicting the target attribute.

You used RFE with the Logistic Regression classifier to select the top 4 features. From the results we see, RFE chose the top 4 features as 'term','int_rate','mort_acc', 'annual_inc_norm'

Inferential Statistics - Test 1

Term

We perform a statistical analysis to establish whether term has a significant impact on the charge offs aka our target variable loan_status

Test:

We will use Z statistic test for this problem
Explanation- We take a random sample from the data set with sample size =100 (50+50). Our target variable has 2 values. So this is a binomial distribution. However with n=100, Central Limit Theorem applies. The sampling distribution will follow normal distribution.

Hypothesis:

Null Hypothesis H0: There are no differences in both sample proportions i.e. term has no impact on the charge back

Alternate hypothesis H1: Both sample proportions are different i.e. term has an impact on the charge backs

Significance:

Confidence Level : 90%

Analysis Plan:

Reject the null hypothesis if p value< a
If p val > 1, we will fail to reject the null hypothesis

Note: the test is appropriate because the sampling method was simple random sampling, the samples were independent, each population was at least 10 times larger than its sample, and each sample included at least 10 successes and 10 failures.

Interpretation:

After running the test, we observed that p value < a, so we reject the null hypothesis.
That means Both sample means are different i.e. with 90% confidence we can say that term has significant impact on the charge offs.

Inferential Statistics - Test 2

Interest_rate

We perform a statistical analysis to establish whether interest_rate has a significant impact on the charge offs aka our target variable loan_status

Test:

We will use Z statistic test for this problem
Explanation- We take a random sample from the data set with sample size =100 (50+50). Our target variable has 2 values. So this is a binomial distribution. However with n=100, Central Limit Theorem applies. The sampling distribution will follow normal distribution.

Hypothesis:

Null Hypothesis H0: There are no differences in both sample proportions i.e. interrest_rate has no impact on the charge back

Alternate hypothesis H1: Both sample proportions are different i.e. interrest_rate has an impact on the charge backs

Significance:

Confidence Level : 95%

Analysis Plan:

- Reject the null hypothesis if p value< a
- If p val > 1, we will fail to reject the null hypothesis

Note: the test is appropriate because the sampling method was simple random sampling, the samples were independent, each population was at least 10 times larger than its sample, and each sample included at least 10 successes and 10 failures.

Interpretation:

After running the test, we observed that p value < a, so we reject the null hypothesis.
That means Both sample means are different i.e. with 95% confidence we can say that interrest_rate has significant impact on the charge offs.

Machine Learning Modeling

Overview and Approach

The aim is to find y , a target variable based on knowing a list of features X. We are trying to predict 'Loan Status'. Loan_status feature is converted to target feature with values 0 = 'Fully paid' and 1 = 'Charged Off'. Since the target variable takes only 2 values 0 and 1, this will be a binary classification problem.
First we use 'Logistic Regression'.

Training and Test Datasets

When fitting models, we would like to ensure two things:

- We have found the best model (in terms of model parameters).
- The model is highly likely to generalize i.e. perform well on unseen data.

We will be using a combination of 2 approaches-

- Holdout
  The given data set is divided into 2 partitions as test and train 20% and 80% respectively. The train set will be used to train the model and the unseen test data will be used to test its predictive power.

- 10 fold Cross Validation
  The 80% train data set is randomly partitioned into 10 mutually exclusive subsets, each approximately equal size. Training and testing iterated over these 10 folds.

Finally the trained model is tested on the 20% holdout set.

First, we try a basic Logistic Regression:

- Split the data into a training and test (hold-out) set
- Train on the training set, and test for accuracy on the testing set

Choice of Performance Metric

As we have seen earlier, we have a target feature with class imbalance. F1 measure is a better metric in case of imbalanced class. In addition False Negatives and False Positives are important for our project. For these reasons we will be using F1 weighted average since it gives a better measure of the incorrectly classified cases than the Accuracy Metric.

Training the Model

We have established above that this is a binary classification problem. We choose our model as a logistic regression. We start off with including all the features filtered out/ selected after feature selection and feature engineering step. After training and testing we note down the performance metric. Next we move to Logistics regression with features given by RFE. Then lastly we do the logistics regression with k=10 fold cross validation.

Results Summary

We are looking at leveraging the best model to predict whether the loan application will end with ‘Fully Paid’ or ‘Charged Off’ status. Classification models generate the probabilities associated with each outcome. Internally the threshold for scikitlearn models is set at 0.5. So anything above that would be termed as ‘1’ or in our case 'a bad loan’. In future, before the deployment depending on business use case, this threshold could be decided upon i.e. it could be anything other than 0.5 and model could be used to identify the cases above the threshold.

Comparison of Base Model to “Final” Model

Base Model - LogisticRegression

Accuracy Score = 0.8082622721248344

#	Precision	Recall	F1-Score
0	0.81	0.99	0.89
1	0.48	0.05	0.08
Micro Avg	0.81	0.81	0.81
Macro Avg	0.65	0.52	0.49
Weighted Avg	0.75	0.81	0.74

Final Models

SGD

After hyperparameter tuning and scaling all the features using standardscaler.

Accuracy score on training set	0.7031064515959886
Accuracy score on test data set	0.7053174398676887

#	Precision	Recall	F1-Score
0	0.65	0.75	0.70
1	0.64	0.53	0.58
Micro avg	0.65	0.65	0.65
Macro avg	0.65	0.64	0.64
Weighted avg	0.65	0.65	0.64

Logistics Regression

After hyperparameter tuning

Accuracy score on training set	0.6948059891541704
Accuracy score on test data set	0.6975098108396514

#	Precision	Recall	F1-Score
0	0.65	0.75	0.69
1	0.64	0.52	0.57
Micro avg	0.64	0.64	0.64
Macro avg	0.64	0.63	0.63

Weighted avg	0.64	0.64	0.64

Random Forest Classification

Accuracy Score	0.6135567219623709

#	Precision	Recall	F1-Score
0	0.73	0.62	0.67
1	0.48	0.60	0.53
Micro avg	0.61	0.61	0.61
Macro avg	0.60	0.61	0.60
Weighted avg	0.64	0.61	0.62

Also if we compare the model training time

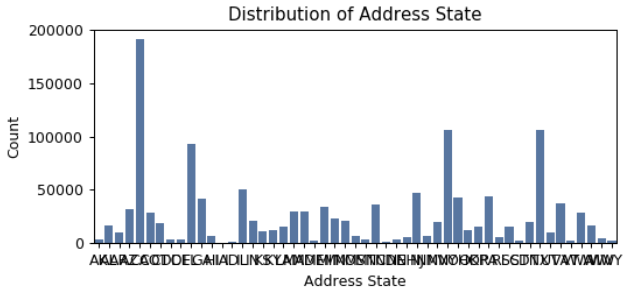
Model	Performance
SGD	[Parallel(n_jobs=1)]: Done 15 out of 15 | elapsed: **8.7s** finished
Logistic Regression	[Parallel(n_jobs=1)]: Done 15 out of 15 | elapsed: **9.2s** finished

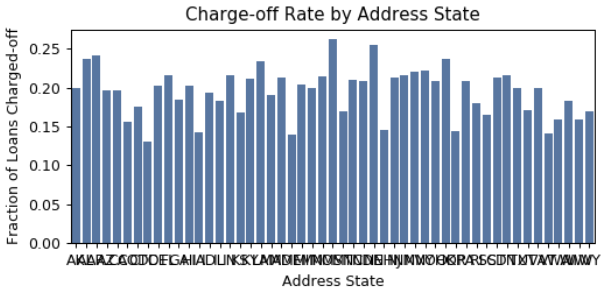
There is a huge improvement from base model performance to both our final models.
we see that SGD while on comparing performance/classification metric performed similarly.
However it is slightly better when we compare model training time with Logistics Regression.

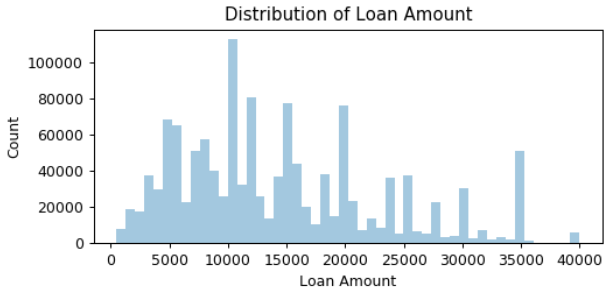
Future Recommendations/ Improvements

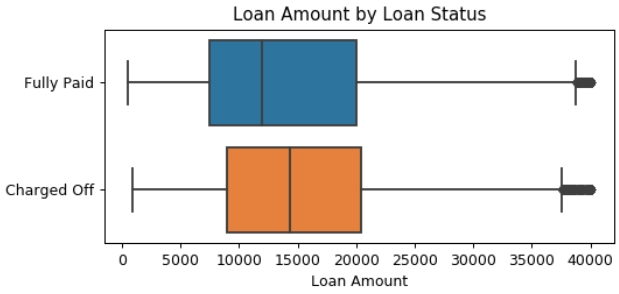
Since our train and test data come from the same file, we can safely say that the samples taken from the train and test data are coming from the same distribution. This is usually a contributing factor to getting a nicely performing ML model. However this may not be the case in the real world. So there is a possibility that the model may not predict as efficiently with the real world data. So we will need to iteratively keep training the model to include as diverse training cases as possible. This is a good scope of future improvement with respect to model efficiency in cases where the model performs poorly.

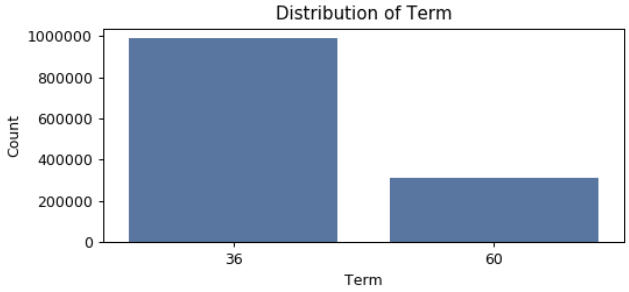
Appendix : Distributions of feature set

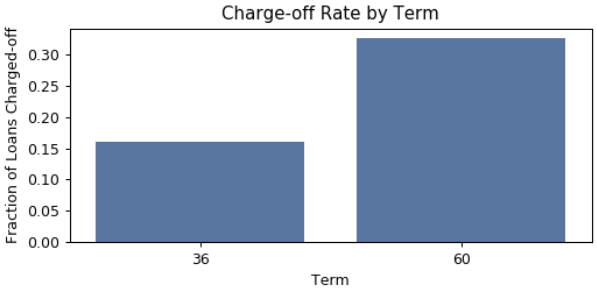


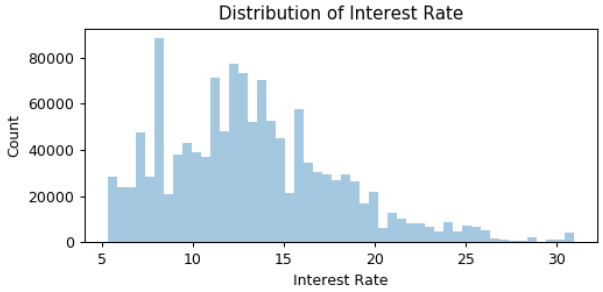


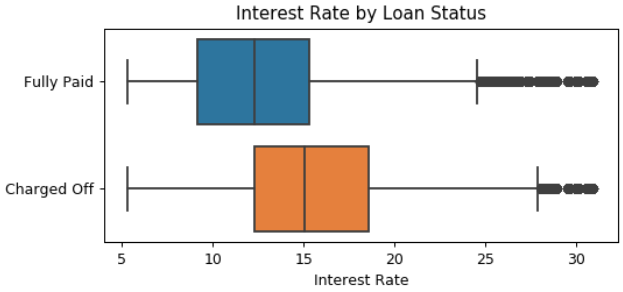


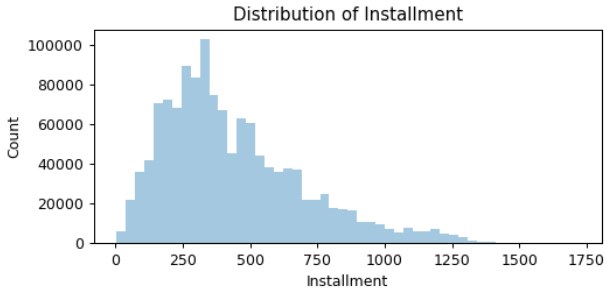


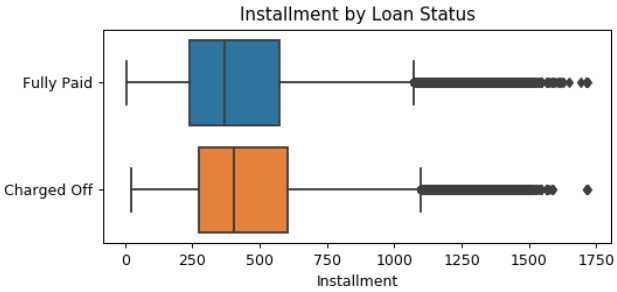


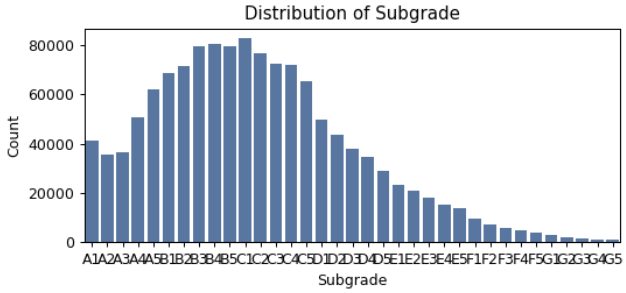


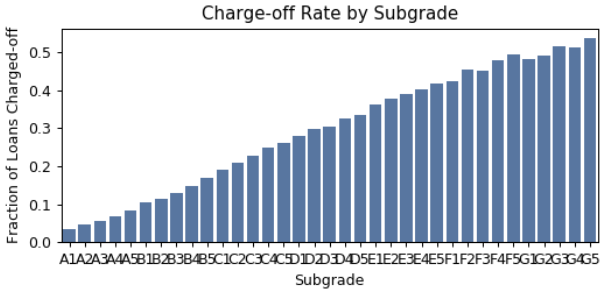

Distribution of Interest Rate
Count
Interest Rate


Interest Rate by Loan Status
Fully Paid
Charged Off
Interest Rate


Distribution of Installment
Count
Installment

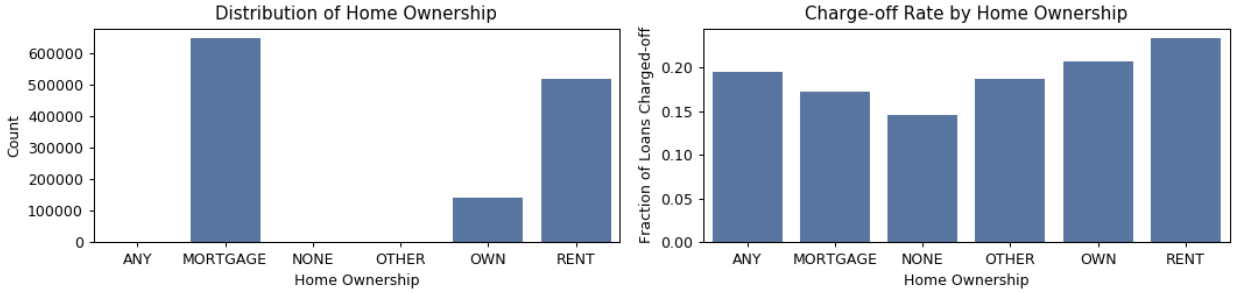

Installment by Loan Status
Fully Paid
Charged Off
Installment


Distribution of Subgrade
Count
Subgrade


Charge-off Rate by Subgrade
Fraction of Loans Charged-off
Subgrade


Distribution of Employment Length
Count
Employment Length


Charge-off Rate by Employment Length
Fraction of Loans Charged-off
Employment Length


Distribution of Home Ownership
Count
600000
500000
400000
300000
200000
100000
0
ANY
MORTGAGE
NONE
OTHER
OWN
RENT
Home Ownership
Charge-off Rate by Home Ownership
Fraction of Loans Charged-off
0.20
0.15
0.10
0.05
0.00
ANY
MORTGAGE
NONE
OTHER
OWN
RENT
Home Ownership

Pearson Correlation Coefficients Matrix: