Microsoft Data Science Challenge: Loan Granting Binary Classification

Problem

This competition concerns loan data. When a customer applies for a loan, banks and other credit providers use statistical models to determine whether or not to grant the loan based on the likelihood of the loan being repaid. The factors involved in determining this likelihood are complex, and extensive statistical analysis and modelling are required to predict the outcome for each individual case. You must implement a model that predicts loan repayment or default based on the data provided.

In this competition, you must explore and cleanse a dataset consisting of over 111,000 loan records to determine the best way to predict whether a loan applicant will fully repay or default on a loan. You must then build a machine learning model and publish it as a web service that returns the unique loan ID and a loan status label that indicates whether the loan will be fully paid or charged off.

Data

The dataset used in this competition consists of synthetic data that was generated specifically for use in this project. The data is designed to exhibit similar characteristics to genuine loan data.

The dataset consists of the following fields:

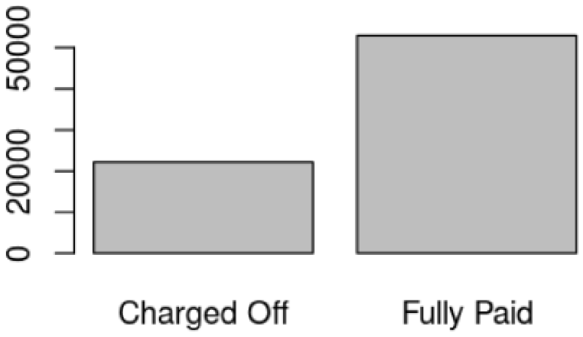
- Loan ID: A unique Identifier for the loan information.
- Customer ID: A unique identifier for the customer. Customers may have more than one loan.
- Loan Status: A categorical variable indicating if the loan was paid back or defaulted(Charged off).
- Current Loan Amount: This is the loan amount that was either completely paid off, or the amount that was defaulted.
- Term: A categorical variable indicating if it is a short term or long term loan.
- Credit Score: A value between 0 and 800 indicating the riskiness of the borrowers credit history.
- Years in current job: A categorical variable indicating how many years the customer has been in their current job.
- Home Ownership: Categorical variable indicating home ownership. Values are "Rent", "Home Mortgage", and "Own". If the value is OWN, then the customer is a home owner with no mortgage
- Annual Income: The customer's annual income
- Purpose: A description of the purpose of the loan.
- Monthly Debt: The customer's monthly payment for their existing loans
- Years of Credit History: The years since the first entry in the customer’s credit history • Months since last delinquent: Months since the last loan delinquent payment
- Number of Open Accounts: The total number of open credit cards
- Number of Credit Problems: The number of credit problems in the customer records.

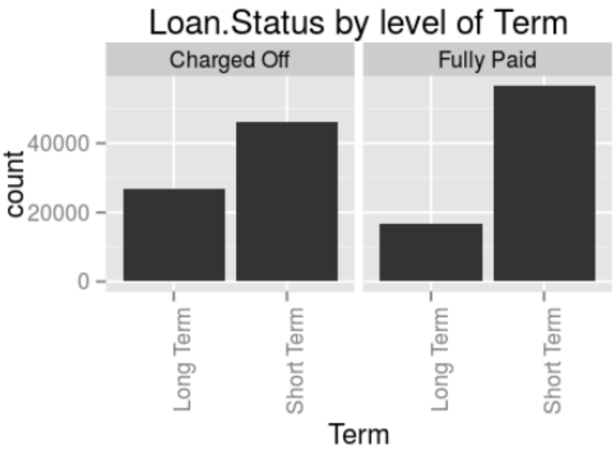
- Current Credit Balance: The current total debt for the customer
- Maximum Open Credit: The maximum credit limit for all credit sources.
- Bankruptcies: The number of bankruptcies
- Tax Liens: The number of tax liens.

The full set of loan records was split into training, public test, and private test datasets. You can use the training data to build and refine your model, and then submit your web service to be tested using the public test dataset. Model is evaluated against the private set upon completion of the competition.

Data Exploration and Visualization

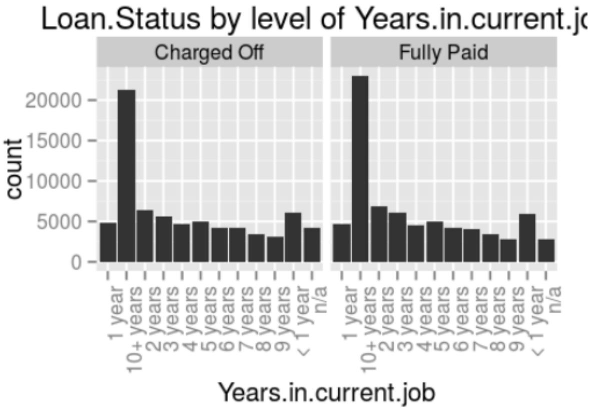
Label classes are highly imbalanced as the defaulters constitute only 30% of the dataset. This may result in biased models, for example 70% accuracy can be achieved by simply predicting 'Fully Paid' for all the rows! And algorithm which is attempting to maximize the accuracy will bias itself towards 'Fully paid'. So we need to consider using the SMOTE module to balance the classes so that we can have unbiased models.



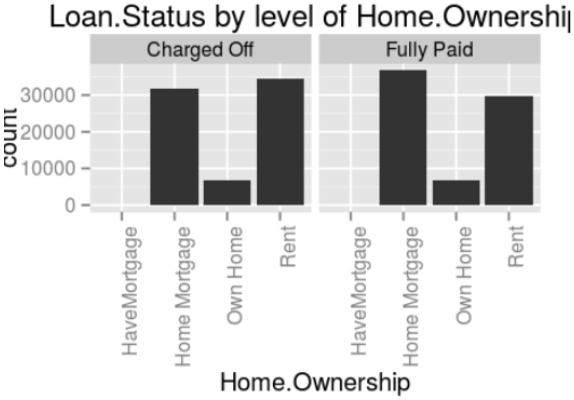


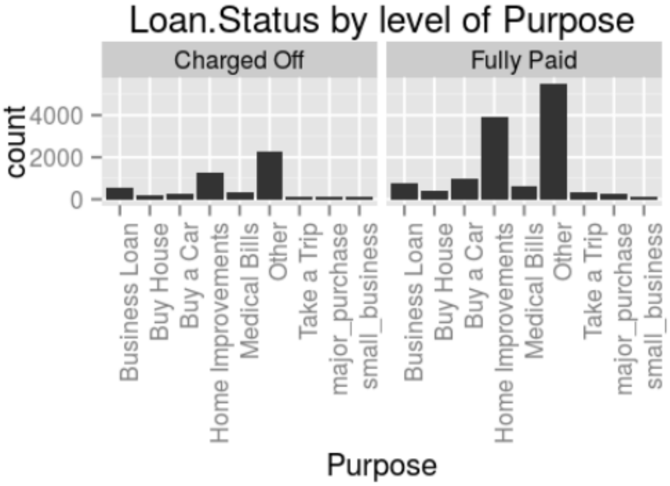
It can be clearly seen that Long term Loans have a distinctively higher ratio of "Charged Off". So the information regarding the duration of loan is a good predictor its status.

As both labels have very similar distributions against the years in current job variable. This variable does not appear to be a good predictor of label class.



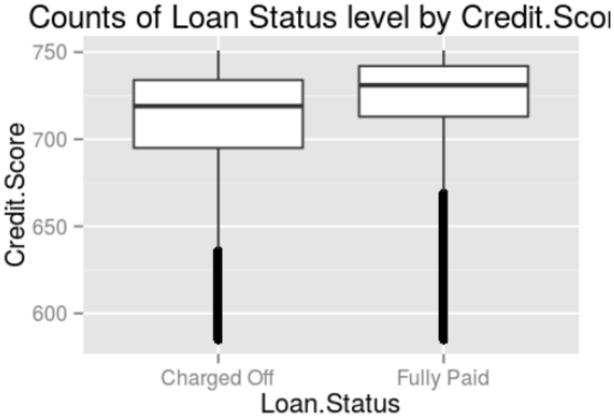
It can be observed that people who live in a rented house are marginally more likely to be defaulters while those who have mortgaged their houses are more likely to pay the loans. This variable can be a reasonably good predictor of label.

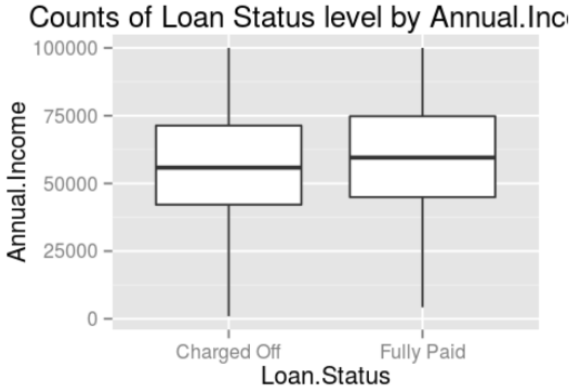




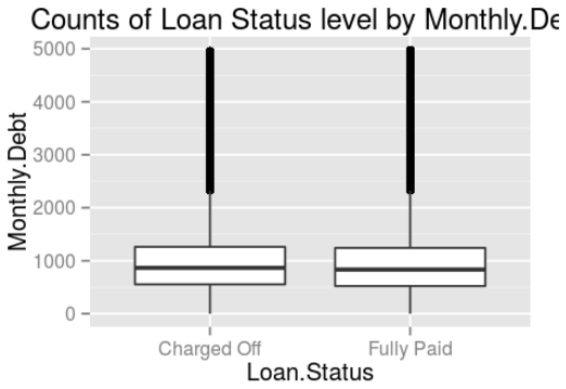
Again the distributions of both labels are vary similarly over this variable. This overlap will make this variable a poor predictor of the label.

There is significant overlap but it is observable that cases with higher credit score are more likely to pay back their loans. So this may be a reasonably good predictor.

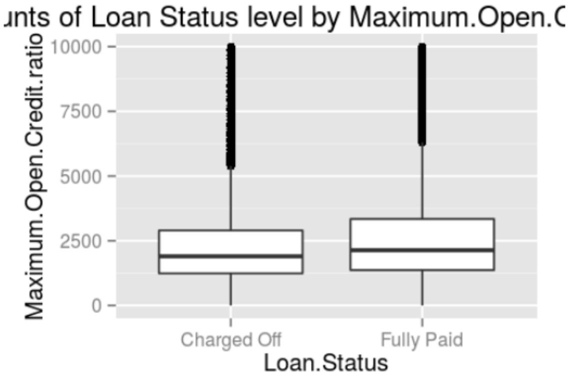




Again there is long overlap but the cases with loan Fully paid have a slightly higher median value for annual income. Appears to be a fairly good predictor.



Both label classes have almost identical distribution of Monthly debt, with Fully paid have slightly lower median value for Monthly debt. Huge overlap may render this variable useless.



Again there is a lot of overlap, with Fully Paid cases tending to have slightly higher Maximum Open Credit. This might be a slightly helpful predictor.

Removing Duplicates

9674 rows are repeated in the training dataset and a row is repeated maximum of two times. We need to remove these repeated rows because they tend to bias the algorithm. Among the two repetitions we need to decide if we should keep the first occurrence or the second occurrence because they may have different information. Analyzing the data reveals that the second repetition usually has less missing values.

	First occurrence	Second occurrence
Rows with Missing Credit Score	5249	0

But we are not sure that same would apply for the test data which is not available to us, so we need to go for a general approach. An R script was programmed that extracts non-missing features from both occurrences.

Following steps are performed:

- First occurrence of rows with all features identical are removed.
- Among repeated record with discrepancy in some attributes, R code amalgamates the records to extract maximum information.

Datatypes and Formatting

Observing the datatypes reveals that ‘Monthly Dept’ and ‘Maximum open credit’ are string features, but they should be numerical. Edit metadata module was used to convert these to numerical, but it resulted in error in some rows. Examining error reveals:

- Monthly Debt has $ preceding value in some rows.
- Maximum open credit has #value which indicates missing value.

This also shed lights on the kind of errors that we can face on the unseen test data so we need to devise a generic treatment for numeric variables.

Generalized Treatment:

- Remove $ and # from all numeric columns.
- Introduce missing value if the value is not convertible to a number.

Repeated categories

- House ownership: 'HaveMortgage' and 'Home Mortgage' are same.
- Purpose: ‘Other’ and ‘other’ are same.

These categories were joined.

Feature Engineering

Indicator Columns were created for the following scenarios:

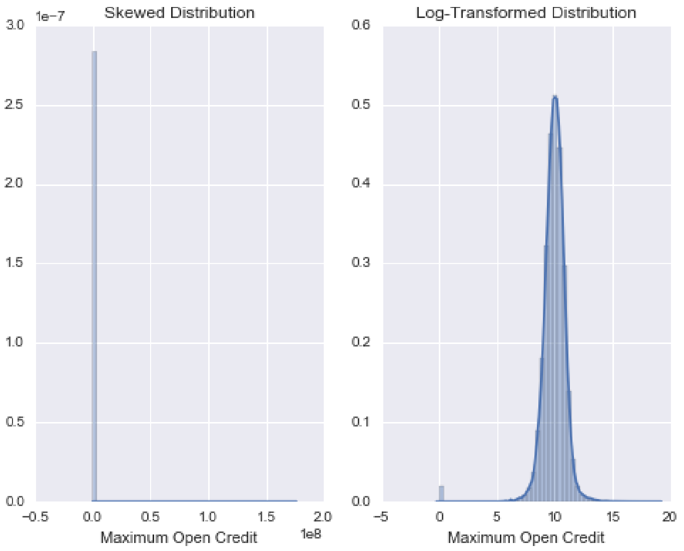
- 16000 rows have both credit score and annual income missing. These rows have higher ratio of charged off than rest of the data.
- 8000 values have current loan amount 999999999, all fully paid.
- 4000 rows have credit score greater than 800(apparently a trailing zero). All of them are charged off.
- Repeated values were all fully paid.

Following Ratios were created. These ratios had stronger correlation with the label class than involved features.

- utilization = Current.Credit.Balance/(Maximum.Open.Credit+1)
- debit.ratio = Monthly.Debt/(Annual.Income/12)
- LoanTOvalue = Current.Loan.Amount / (Annual.Income-(12 * Monthly.Debt))

Transformations

- Some features were transformed using Ln(x+1) to improve pearson correlation and chi-squared test scores, and to make the distribution of the variable more normal.



The numeric features were normalized so that high values of some variables may not result in bias. Tanh and Z score normalization were tried and Z score resulted in models with better AUC

Cross-validation AUC for various models.

	Boosted Decision Tree	Logistic Regression
Tanh Normalization	0.803	0.809
Z score Normalization	0.824	0.825

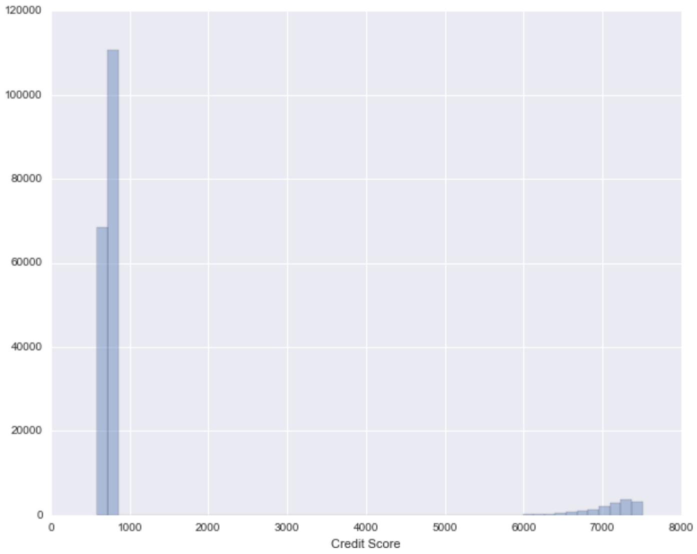
It was also observed that Math transformations must be performed before missing value treatment the model performed better. AUC was increased from 0.805 to 0.813. This is because with more normally distributed variables the missing value treatments like replacing with central measures result in better estimates.

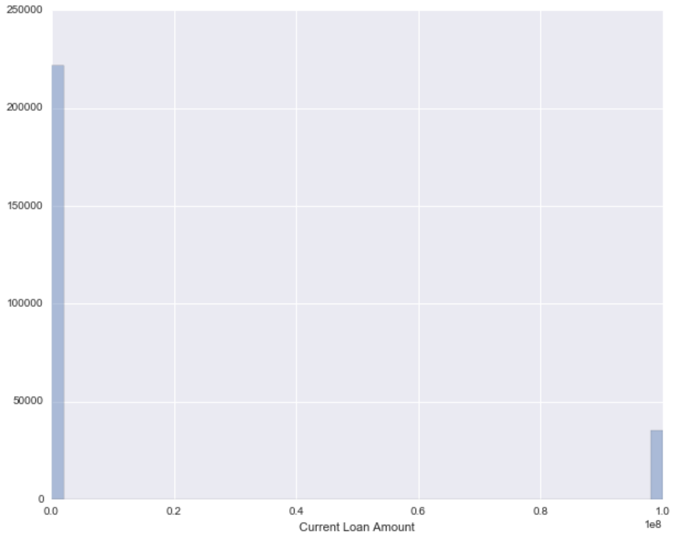
Balancing Classes

Since the label classes were not balanced, we used built in minority oversampling technique called SMOTE. While SMOTE module did not have any significant influence of accuracy of hold-out dataset, it deteriorated the performance on test data. So we decided not to use SMOTE.

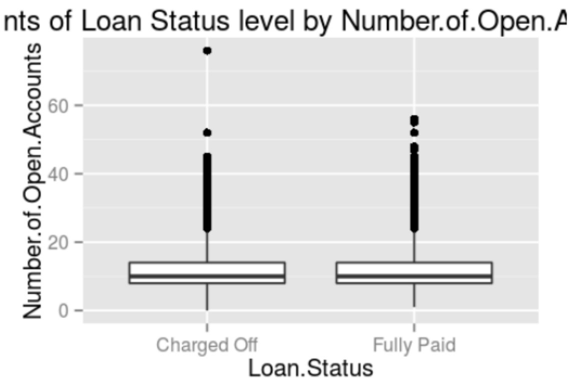
Outlier Treatment

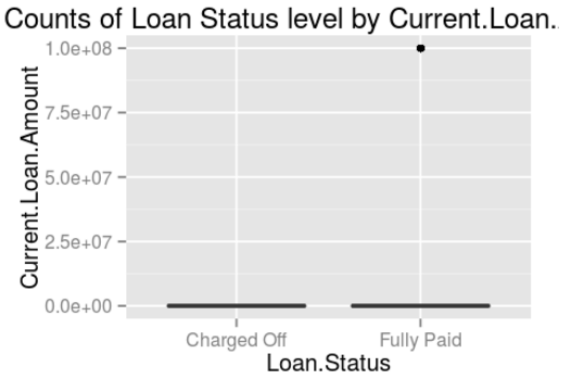
Credit Score clearly has some outlier values to the extreme right. The variable description tells us that maximum valid value for credit score is 800. Observing the outliers closely revealed that these are a result of mistakenly appended zero. So dividing them by 10 will give us the actual credit score. Interestingly all these 4000 outlier rows are defaulters. To preserve this information and feed it to the algorithm we created an indicator column specifying if the row had an invalid credit score value.





Current Loan amount also has conspicuous outliers. These 999999 values correspond to missing data. Thus they are treated along with other missing data. Interestingly all these 8000 rows with missing Current loan amount were fully paid! So an indicator column was also created for them.





As shown above, bar plots were used to identify outliers. These plots can also reveal if extreme values tend to occur with any particular label class only. Rows with outliers were not used in training the model.

Missing Data Treatment

Identifying missing values

- 8000 values have current loan amount 999999999, which were treated as missing values.
- 16000 rows have both credit score and annual income missing.
- Some string features had Na, n/a which were converted to missing values.
- Numeric features that had string values were treated as missing, if the numeric value could not be recovered from it.

Following treatments that were evaluated for all features with missing values:

- Replacing with medians. Since means tend to be drawn away by extreme values.
- Replacing with Median and creating indicator columns. For cases were the fact that rows were missing tended to convey some information about the label class, E.g. all missing values in 'Current Loan Amount' corresponded to Fully Paid label class.
- Using MICE. Imputation was made with-out using the label class because that would have resulted in bias.
- Replacement value. E.g. missing values in 'Current Loan Amount' were all fully paid. So instead of using the median for the whole column we tried to use the median of 'Fully Paid cases'.
- Removing entire rows. Since significant data was missing, this treatment usually resulted in inferior performance.

'Months since Last Delinquent' feature had missing values referring to people who never experienced delinquency. Feature was modified in a way that increased its correlation with the label. Reciprocal of the feature was taken and missing values were replaced by 0. So that the people who had recent delinquents had highest values for this feature and those with no delinquents (i.e. missing values) had 0 value for this feature.

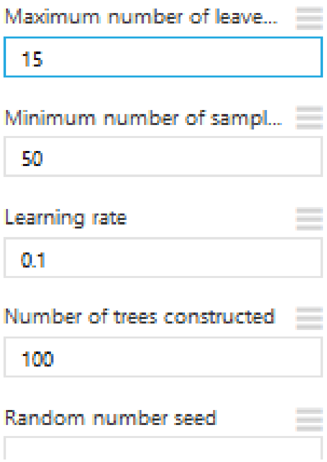
- Months.since.last.delinquent = ifelse(Months.since.last.delinquent == NA,0,1/Months.since.last.delinquent+1))

Model Selection

Due to little correlation between features and labels and significant overlap it was obvious that non-linear modelling techniques would be a better option. Various Models were compared on AUC score in cross-validation on training data. Boosted Decision trees, Decision Forest and Linear regression were the best performers respectively. Boosted Decision Trees outperformed others distinctively in Test scores, and this model resulted in my best private Test score in the competition.

Tuning Model Parameters

Tune model hyper parameters module was used to get a rough estimate of parameter values and then manual selection was made by monitoring model performance. The parameters which performed best by striking a balance between simplicity and overfitting are shown below.



Maximum number of leave...
15

Minimum number of sampl...
50

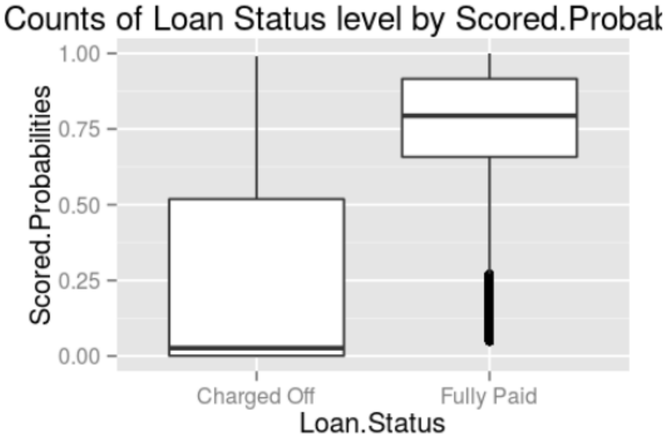
Learning rate
0.1

Number of trees constructed
100

Random number seed

Evaluation

This was a difficult machine learning problem considering the given unclean dataset with no clear correlations and significant overlap within features for both label classes. The figure below shows the scored probabilities that are produced by the prediction model for the hold out data set, plotted against the true label class. Ideally there should be a sharp cut out point, but the scores overlap. So it will always be a big trade-off between sensitivity and specificity.



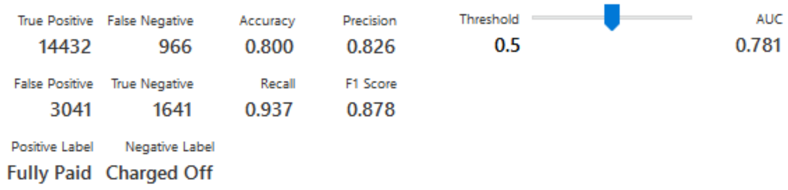
Relatively simpler model performed better than complex models.
The best private score: **72.725262% Accuracy**

Some Interesting Alternate Models

We also tried to build some hybrid and segregated models to see if they would result in better performance. They did not show significant improvements. But they were good practice and can serve as interesting approaches while encountering such problems in the future.

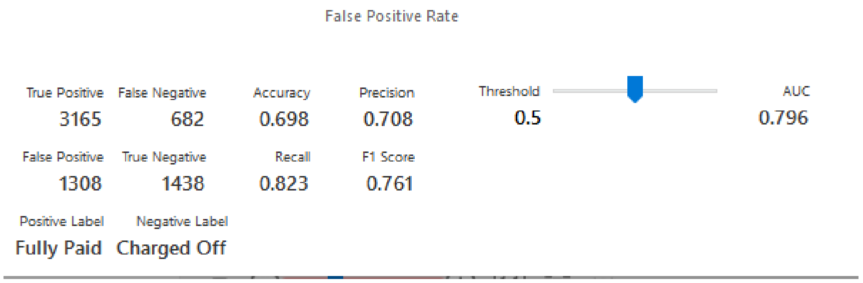
Trying different models for short term and long term Loans.

Short Term only:



True Positive	False Negative	Accuracy	Precision	Threshold	AUC
14432	966	0.800	0.826	0.5	0.781
False Positive	True Negative	Recall	F1 Score		
3041	1641	0.937	0.878		
Positive Label	Negative Label				
Fully Paid	Charged Off				

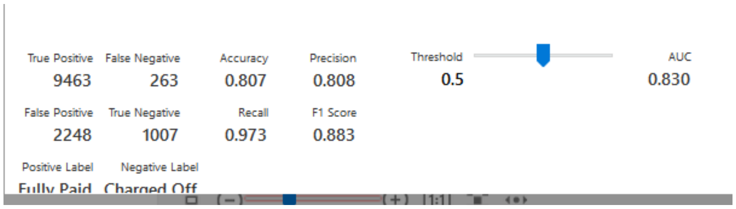
Long Term only:



True Positive	False Negative	Accuracy	Precision	Threshold	AUC
3165	682	0.698	0.708	0.5	0.796
False Positive	True Negative	Recall	F1 Score		
1308	1438	0.823	0.761		
Positive Label	Negative Label				
Fully Paid	Charged Off				

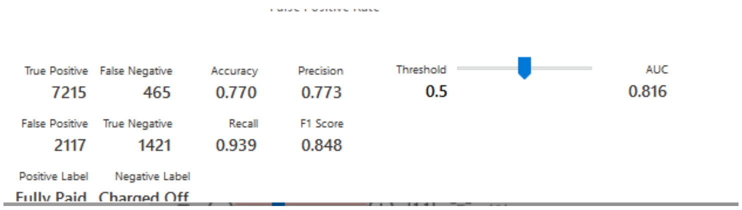
Trying Different Models for Home owner ship:

Home Mortgage:



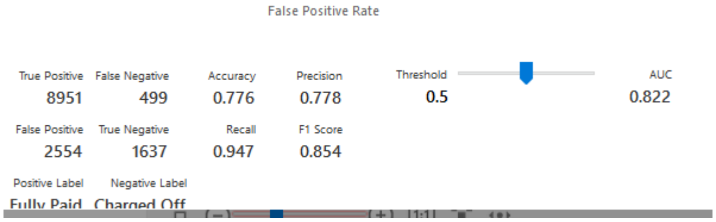
True Positive	False Negative	Accuracy	Precision	Threshold	AUC
9463	263	0.807	0.808	0.5	0.830
False Positive	True Negative	Recall	F1 Score		
2248	1007	0.973	0.883		
Positive Label	Negative Label				
Fully Paid	Charged Off				

Rent:



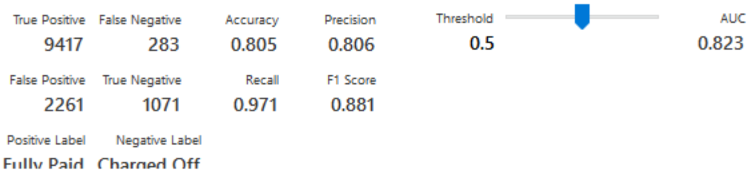
True Positive	False Negative	Accuracy	Precision	Threshold	AUC
7215	465	0.770	0.773	0.5	0.816
False Positive	True Negative	Recall	F1 Score		
2117	1421	0.939	0.848		
Positive Label	Negative Label				
Fully Paid	Charged Off				

Rent + own Home:



True Positive	False Negative	Accuracy	Precision	Threshold	AUC
8951	499	0.776	0.778	0.5	0.822
False Positive	True Negative	Recall	F1 Score		
2554	1637	0.947	0.854		
Positive Label	Negative Label				
Fully Paid	Charged Off				

Have Mortgage+Home Mortgage:



True Positive	False Negative	Accuracy	Precision	Threshold	AUC
9417	283	0.805	0.806	0.5	0.823
False Positive	True Negative	Recall	F1 Score		
2261	1071	0.971	0.881		
Positive Label	Negative Label				
Fully Paid	Charged Off				

We also attempted to build regression models to predict the missing ‘Annual Income’ and ‘Credit Score’ values but its performance was inferior to other missing data treatments.