Capstone #2 Project Milestone Report

Bank Loan Bot

By: Matthew Wallace

Background

I am a data scientist working on a contract to hire position working on a project for a bank called First Calgary Financial. A model will be created to predict whether or not an individual or group will receive a loan or not. This model will be programmed into a bot used specifically to assess a person’s eligibility for a loan. This saves a lot of man power in studying the application and assessing it with human eyes. In this case, if a person is predicted to receive a loan by a bot, then that person automatically receives it. Human eyes are focused on applications where the bots disqualify them from getting a loan.

Project Objective

The objective of this project is to predict whether or not a person or group of people will be granted a loan from the bank.

Strategy

This is a classification problem, so classification models will be used. Three models will be created; a decision tree, a random forest and logistic regression. Each of the three models will be evaluated using an accuracy score. A confusion matrix will be created. The best model will be chosen based on metrics given in their classification reports. Accuracy will be considered, but this will be in conjunction with the ROC\_AUC score; the most important measure in this project. The importance will be based on the model’s ability to separate true positives from false positives. An accuracy score can give too much credit to a model where the majority of the data points are false negatives, which are easily classified.

Dataset Import and Description

The dataset was downloaded into csv files from Kaggle. The dataset will be uploaded in two sets; training data and testing data. The training data contains 12 independent variables as well as the target variable; “loan status.” The testing data will contain more rows containing 12 independent variables only. These independent variables are listed as columns within the training and testing data. These column labels are: Loan ID, Gender, Married, Dependents, Education, Self-Employed, Applicant Income, Coapplicant Income, Loan Amount, Loan Amount Term, Credit History, Property Area.

Unfortunately, the test data has no target variable. This data was to be used as a submission of ones predictions from creating their models using the training data for a competition assessment. Therefore, the training data was used as the entire data set containing 614 observations. The data types for the columns were mixed. Four columns were float64, one was int64 and the remaining eight were of object data type.

Data Attribute Observations

**Loan\_ID** – *loan identification.*

**Gender** – *Male or female.*

**Married** – *Whether the applicant is married or not married.*

**Dependents** – *The number of dependents an applicant has*

**Education** – *This states whether the applicant is considered educated or not educated.*

**Self-Employed** – *States whether or not the applicant runs their own business*

**ApplicantIncome** – *Annual income of the applicant.*

**CoapplicantIncome**- *Income of the coapplicant.*

**LoanAmount** – *States the amount of money the applicants are asking for as a loan.*

**Loan\_Amount\_Term** – *States the loan time in months the applicants are applying for.*

**Credit\_History** – Credit history of the applicants; *1 for good credit and 0 for bad credit.*

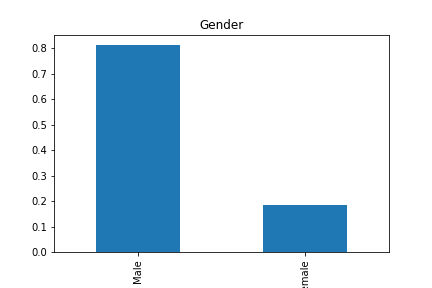
**Property\_Area** – *The applicants’ residential property area.*

**Loan\_Status** *– Whether or not the loan was approved. This is the target variable.*

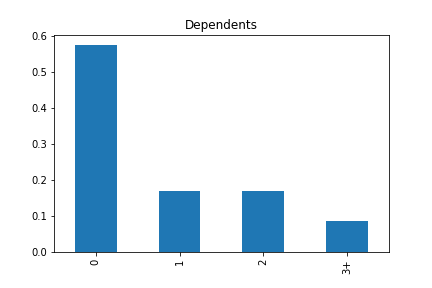
Exploratory Data Analysis

Although there are missing values, the way these missing values were imputed comes from information gained through EDA. It was found with many categorical features that it made the most sense to impute the mode based on the frequency distributions. For other categorical features, the median was imputed due to the observation of outliers. The wrangling and cleaning of the data is discussed below based on the insights gained from EDA.

Frequency bar charts were created to visualize all the numerical columns. There were significantly more males than females.



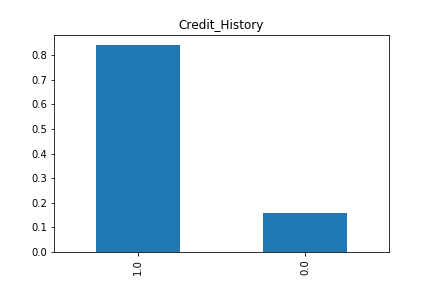
This data was generated over 40 years ago where men were the bread winners. Thus, most women had no concern loans. The majority of applicants had no dependents.



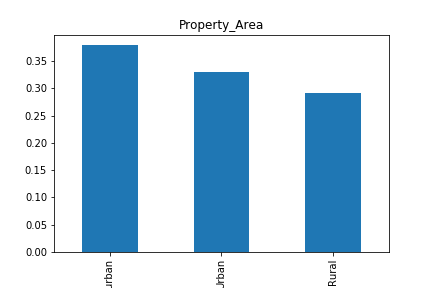
Most applicants were not self-employed.



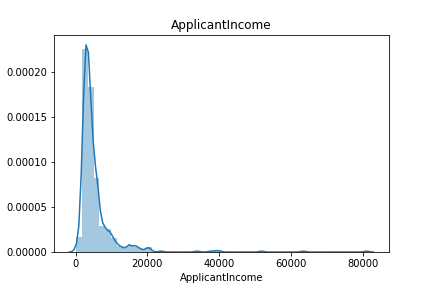
Most applicants had good credit history.

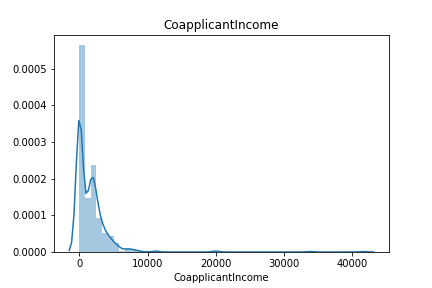


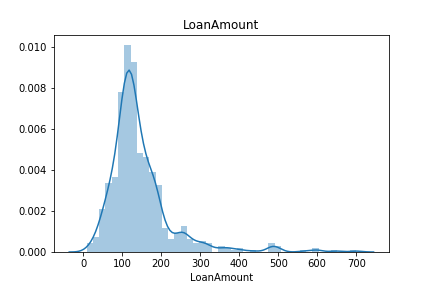
All applicants came from a variety of residence where there were slightly more living in semi-urban and urban areas than in rural areas.



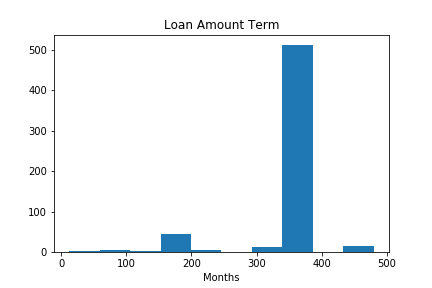
The applicants’ incomes generally were much higher than the co-applicants’ income, but the distribution was the same. Like the loan amount, the applicant and co-applicant incomes had normal distributions with a tail to the right. Thus there were a lot of high outliers to the right. These high outliers project the fact that there is no ceiling when it comes to earning potential.



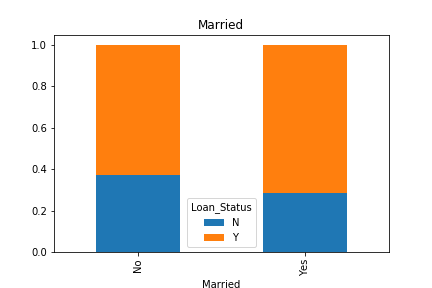


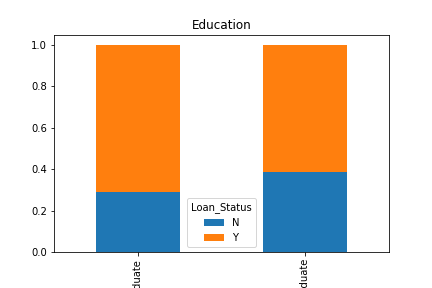


The loan amount term was also variable from 12 months to 480 months with a large mode situated at 360 months.

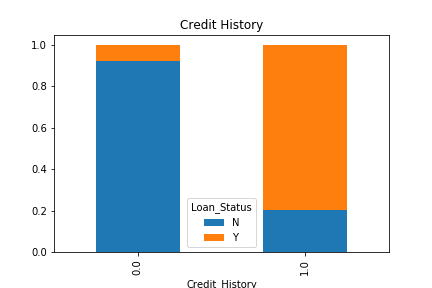


Cross-tab bar charts were created to visualize any relationship between the frequencies of loan approval in comparison to the various categorical variables. There is a slightly higher frequency of loan approval among married applicants and educated applicants. The banks see these applicants as more stable therefore there is less risk of the banks not getting their money back.

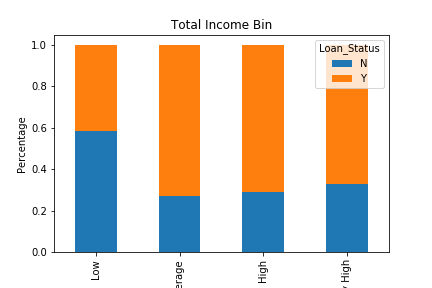




There is a significantly higher frequency of loan approval to applicants who have good credit outlining the importance of having good credit in order to get a loan from a bank. Banks don’t like lending money to applicants with bad credit because of a lack of will/ability to pay debts.



Applicants and coapplicants with a total income that is average or higher stand a significantly higher chance at getting a loan. This makes sense considering that it’s more risky to lend money to people of lower income. However, income is not as important as credit history because income says nothing about an applicant’s will to pay back a loan. Asking loan amounts that are lower stand a higher chance of getting loan approval. This makes sense because for banks, shorter loan terms mean higher risk.



A Seaborn correlation heatmap shows two strong correlations. One shows that there’s a weak correlation between applicant and coapplicant income and another stronger correlation between credit history and loan approval. 

Data Wrangling and Cleaning

Three general steps were taken to wrangle and clean the data. The first was imputing values. The second was quantizing the categorical features and the third was changing the datatypes of all the variables to float64.

Step 1:

The following variables had the mode imputed to replace missing values: Gender, Dependents, Credit\_History, Self\_Employed, Married, LoanAmountTerm. The LoanAmount variable had missing values as well, but based on the distribution the median was imputed.

Step 2:

Variables such as Loan\_Status, Education, Self\_Employed, Gender, Married, Dependents and Property\_Area were quantized and converted from categorical variables to numerical. All these variable except property area and dependents were binary. Dependents had 5 classes; 0, 1, 2, 3, 3+. Replacing the ‘3+’ entry with ‘3’ allowed me to express this variable numerically, but 1 class had to be dropped; the ‘3’ and ‘3+’ were merged into a single class. Property\_Area was expressed as extent of urbanization from rural (0) to Urban (2).

Step 3:

Once all variables are complete and numerically expressed, they were all converted to float64 type.

Using Logistic Regression Model To Predict Loan\_Status

**Preparation Of Data For Modelling**

**Step 1:**

The training set was split up into X and y variables. The loan\_status column was dropped

and saved as a series named y.

**Step 2:**

The input variables; X\_train, X\_test, y\_train, y\_test were created from X and y created in step 1 using test\_train\_split from sklearn.model\_selection.

Evaluation Of The Model

Two metrics were used to evaluate the model and a confusion matrix was created. The first

Was just the accuracy\_score measurement from sklearn.metrics. The second metric used was

An ROC curve generated from a function created where the data from test\_train\_split was

taken in as arguments and the same logistic regression model was run. The accuracy score

of the model was about 0.78, which was in agreement with the accuracy score generated from

the function. The ROC curve below illustrates a reasonable job at the models ability to separate a true positive from a false positive.

The Best Models Used

A LogisticRegression() from sklearn.linear\_model. The best logistic regression model and it’s parameters are defined as such:

LogisticRegression(C=0.26826957952797276, class\_weight=None, dual=False,

fit\_intercept=True, intercept\_scaling=1, l1\_ratio=None,

max\_iter=100, multi\_class='auto', n\_jobs=None, penalty='l2',

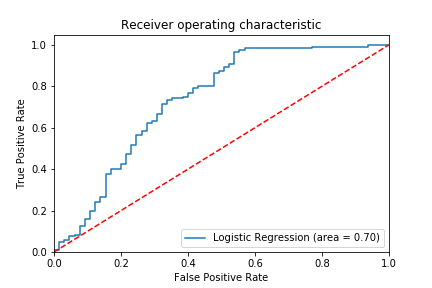
random\_state=None, solver='lbfgs', tol=0.0001, verbose=0,

warm\_start=False)

The most important parameter is the C parameter, which represents the extent of

regularization. C is the inverse of the regularization and therefore the lower the value of C,

the higher the amount of regularization. Using GridSearchCV, this hyperparameter was tuned and found to be around 0.27. The data was fairly linear in relation to the target variable so not much regularization was needed.



Accuracy of logistic regression classifier on test set: 0.78

Confusion matrix:

[[ 25 33]

[ 8 119]]

Classification report:

precision recall f1-score support

0.0 0.76 0.43 0.55 58

1.0 0.78 0.94 0.85 127

accuracy 0.78 185

macro avg 0.77 0.68 0.70 185

weighted avg 0.77 0.78 0.76 185

A random forest classifier was chosen as well because random forests are generally robust and are good at explaining variations in data due the ensembling of many decision trees. The basic model is defined as such:

RandomForestClassifier(bootstrap=True, ccp\_alpha=0.0, class\_weight=None,

criterion='gini', max\_depth=3, max\_features='auto',

max\_leaf\_nodes=None, max\_samples=None,

min\_impurity\_decrease=0.0, min\_impurity\_split=None,

min\_samples\_leaf=1, min\_samples\_split=2,

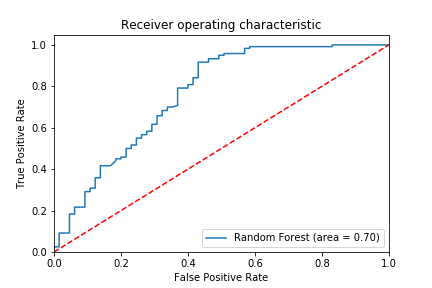
min\_weight\_fraction\_leaf=0.0, n\_estimators=61,

n\_jobs=None, oob\_score=False, random\_state=None,

verbose=0, warm\_start=False)

The parameters that were tuned were the number of trees (n\_estimators) and the number of nodes (depth). This was carried out using both GridSearchCV and RandomSearchCV. It was

found that RandomSearchCV produced varying number of trees. The best random forests were found to contain 41 and 141 trees. The best estimator as shown above come from using GridSearchCV. When a decision tree was used, the ROC score was around 0.66 when only the credit variable was considered, which is an indication of the importance of credit history.



Accuracy of random forest classifier on test set: 0.78

Confusion matrix:

[[ 27 38]

[ 2 118]]

Classification report:

precision recall f1-score support

0.0 0.93 0.42 0.57 65

1.0 0.76 0.98 0.86 120

accuracy 0.78 185

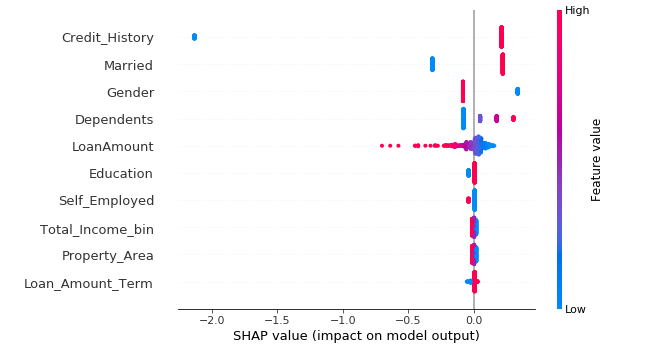
macro avg 0.84 0.70 0.71 185

weighted avg 0.82 0.78 0.76 185

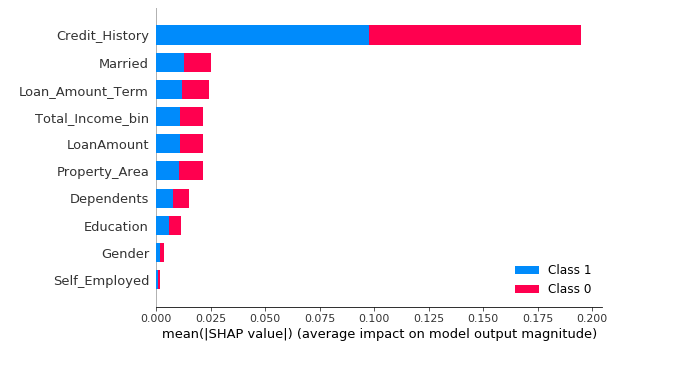
Feature Importance And Explanation

Shap values for each feature was measured for both models. A shap value is a data point’s contribution to the value of the target variable. Data points where certain features like credit history have low values like 0 will have negative shap values. They will contribute to a target variable concluding as a 0. Data points for credit history having 1’s indicating good credit history will have positive shap values. The diagrams below are the shap values for each feature taken from both models.

**Shap Values For Logistic Regression**



**SHAP Values For Random Forest Classifier**



SHAP values for both the random forest classifier and the logistic regression were displayed. Both models stand in agreement with the two most important features being credit history and married. However, in the random forest classifier, the magnitude of importance for Married, Loan\_Amount\_Term, Total\_Income\_bin, LoanAmount and Property\_Area were given some importance and the magnitude of that importance for these variables was comparable. Married is an important feature. At the time this data was generated, it indicates an applicant’s character and stability. Banks find it less risky lending money to people who are stable.

It was good to use a random forest as well as a logistic regression because not everything was explained reasonably with the logistic regression only. For example, the crosstab bar charts from the EDA clearly showed that having dependents does not have any clear effect on the target variable alone. The logistic regression model will fit a linear relationship between number of dependents and its contribution to the target value. The random forest on the other hand placed less priority on dependents, which makes more sense.

The Total\_Income\_bin feature was engineered and given more priority in the random forest model. This was a good thing because the random forest captured the pattern of having less chance of getting approval if the total income between the applicant and co-applicant was low. By today’s standards, capacity is an important factor in determining how much risk a bank has to take in lending people money. However, the ability to pay back a loan is not as important to a bank as is the will to pay back a loan. This is explicitly measured in credit history.

The logistic regression model did a good job at explaining the relationship between the loan amount and the target variable. Banks take less risk lending less money to an applicant, which increases his chances of getting a loan.

Property area is important to a bank. At the time of this data, rural property must have been cheaper on average than property in the city. Therefore, people buying more expensive property need to ask for more money, which decreases their chances of getting a loan.

Conclusions

From the results, it is clear that at this time a random forest is the best model here. The f1-scores are slightly higher. However, a random forest is good at explaining the data based on ensembling of many different trees. A logistic regression model is capable of beating the random forest model, however this is going to require more feature engineering. Many of the categorical variables like credit history, education and property area was over simplified. Only total income bin portrayed an applicant’s capacity to pay back a loan. This was not clear when provided only with the applicant and co-applicant incomes. If a random forest model is the best model here, then the logistic regression model should prioritize and explain the variables in a similar fashion.

Improvements And Future Scope Of Work

Other models like XG boost can be further tuned to make improvements on the accuracy of the model specifically improving the ratio of true positives to false positives. XG boost models have the potential to outperform a random forest. Therefore the goal is to develop a model where the false positives are reduced.