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.

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.

Data Wrangling and Cleaning

Both the training and testing data were uploaded into separate Pandas DataFrames. The thirteen columns are discussed below. The data types for the columns were mixed. Four columns were float64, one was int64 and the remaining eight were of object data type.

Throughout the course of the wrangling process, most of the columns that posed some numerical measurement were converted to float64 type. This allowed these columns to be analyzed in a Seaborn matrix plot so correlations can be observed.

Some columns had missing values entered as NaN. This was easy do deal with because all that was needed was imputation. For the categorical variables, the mode was imputed and for the numerical variables the median was imputed. There were outliers in the income columns and loan term. The mean would not have therefore sufficed as the most accurate method for filling in the missing data in these columns. The target variable; the loan status, was quantized such that a non-approval was 0 and a loan approval was 1. This was an important step so that this could be included in the Seaborn matrix chart. The dependents column was also modified to make this data type numerical. The '3+' entries were converted to '3.'

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.*

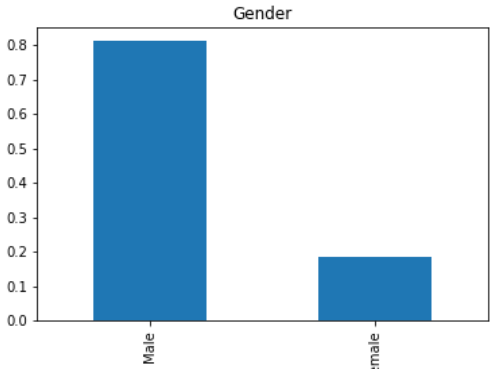
Credit_History – Credit history of the applicants; *1 for good credit and 0 for bad credit.*

Property_Area – *The applicants' residential property area.*

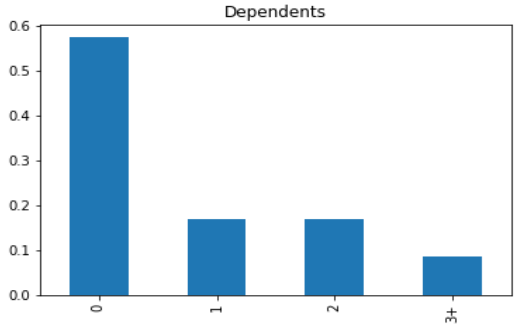
Loan_Status – *Whether or not the loan was approved. This is the target variable.*

Exploratory Data Analysis

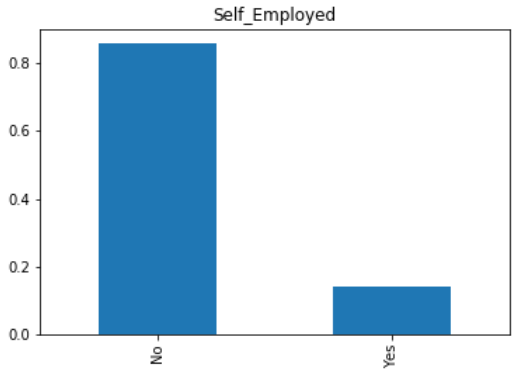
Frequency bar charts were created to visualize all the numerical columns. There were significantly more males than females.



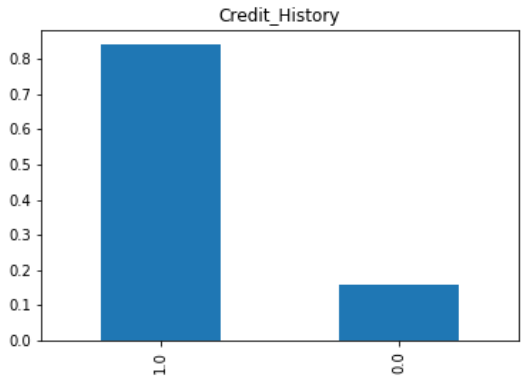
This data was generated over 40 years ago where men were the bread winners. Thus, most women had no concern loans. The majority of applicants had no dependents.



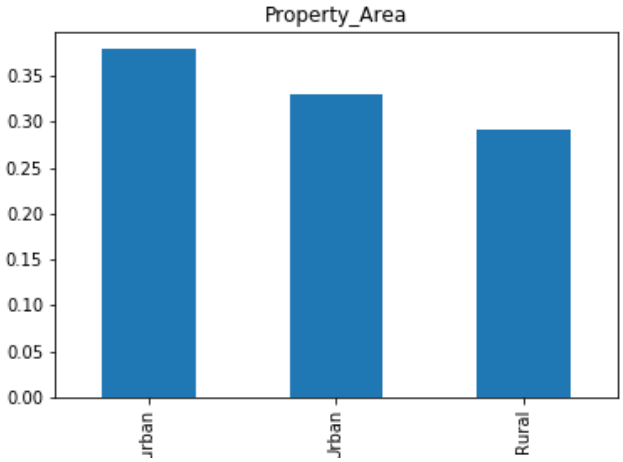
Most applicants were not self-employed.



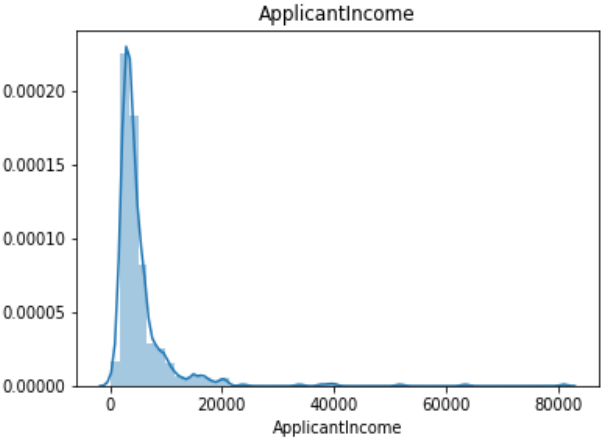
Most applicants had good credit history.

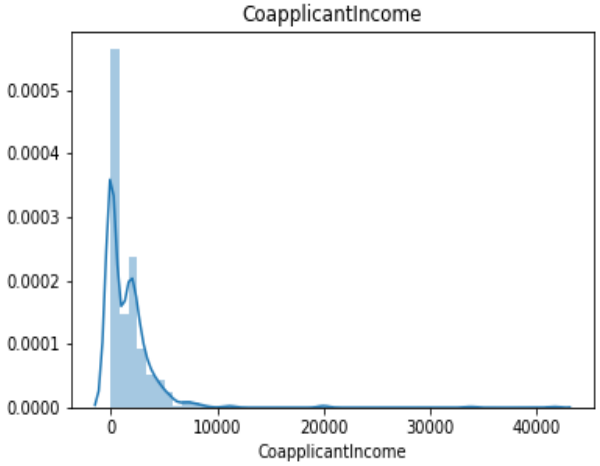


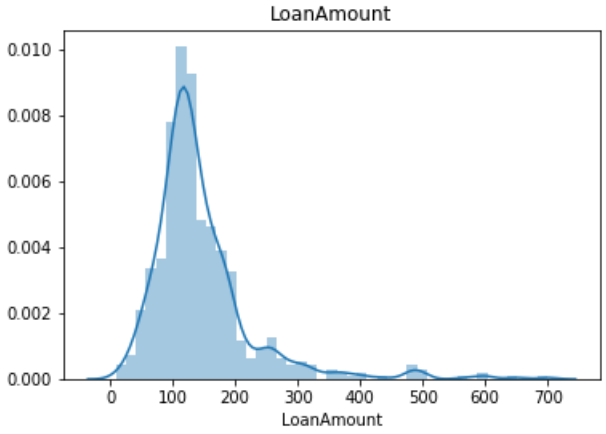
All applicants came from a variety of residence where there were slightly more living in semi-urban and urban areas than in rural areas.



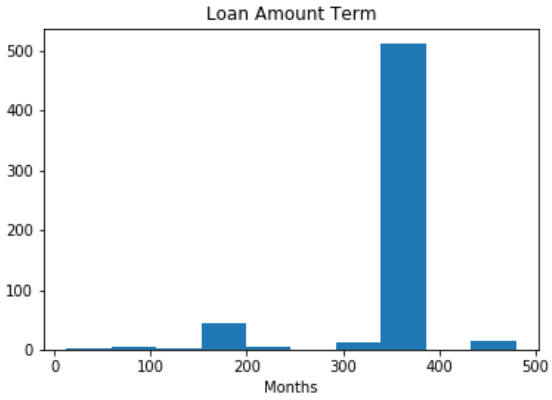
The applicants' incomes generally were much higher than the co-applicants' income, but the distribution was the same. Like the loan amount, the applicant and co-applicant incomes had normal distributions with a tail to the right. Thus there were a lot of high outliers to the right. These high outliers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


ApplicantIncome
0.00020
0.00015
0.00010
0.00005
0.00000
0
20000
40000
60000
80000
ApplicantIncome

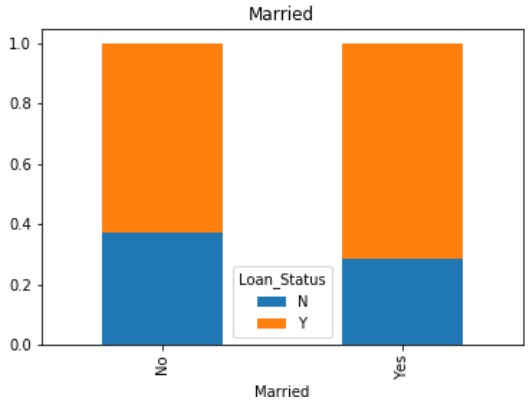

CoapplicantIncome
0.0005
0.0004
0.0003
0.0002
0.0001
0.0000
0
10000
20000
30000
40000
CoapplicantIncome

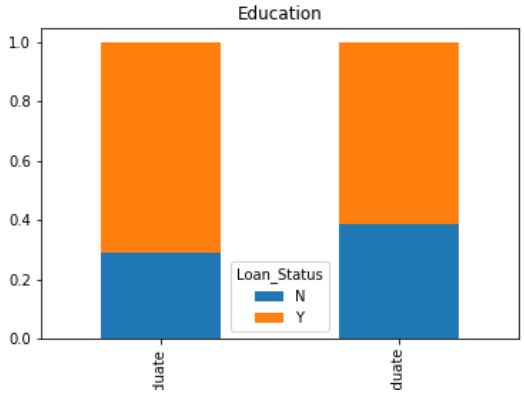

LoanAmount
0.010
0.008
0.006
0.004
0.002
0.000
0
100
200
300
400
500
600
700
LoanAmount

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



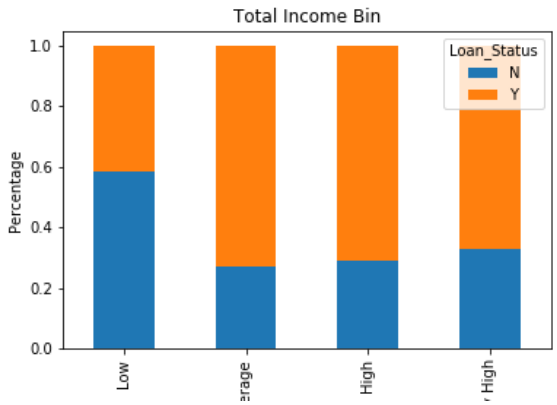
Cross-tab bar charts were created to visualize any relationship between the frequencies of loan approval in comparison to the various categorical variables. There is a slightly higher frequency of loan approval among married applicants and educated applicants. The banks see these applicants as more stable 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.

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:

Not all the columns are numerical at this point. Columns such as Married, Self_Employed, and Property_Area remained categorical. To transform these categorical variables into numerical data for the regression model, Pandas has a get_dummies function. This function pivots the categorical variables by dropping the categorical columns and replacing these dropped columns with new columns where each column represents a class. Each entry is entered as '1' or '0' to indicate a positive or negative value for that class. These dummy variables were then incorporated into the model as determinant features.

Step 3:

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.

The Model Used

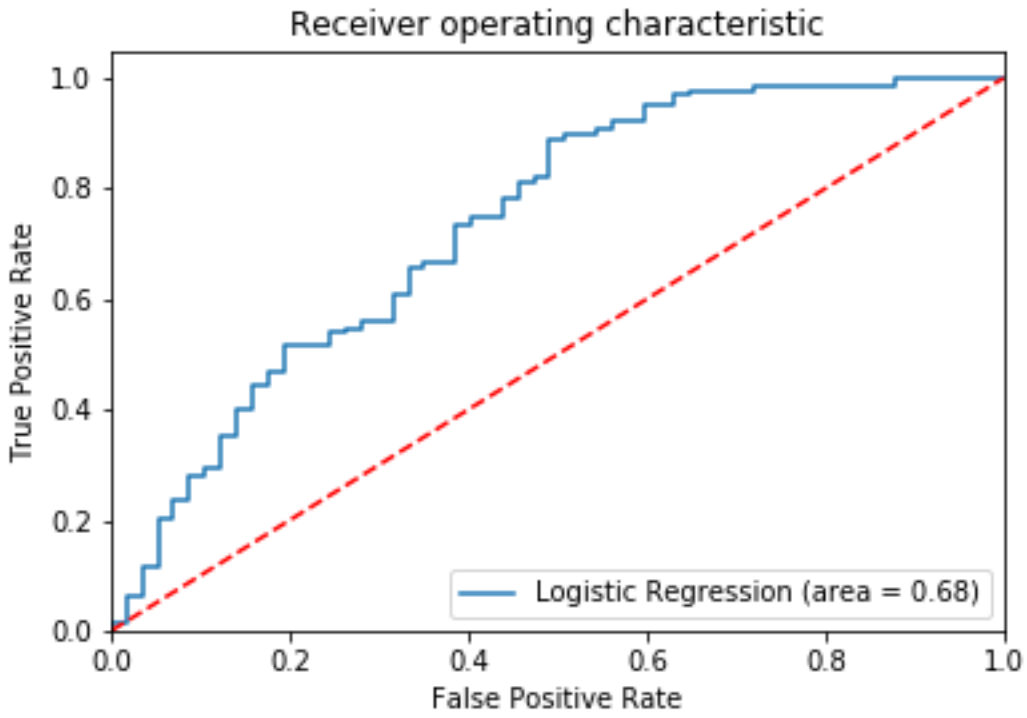
A model defined as 'Model' was created using LogisticRegression() from sklearn.linear_model. The model and it's parameters are defined as such:

```
LogisticRegression(C=1.0, 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. Subsequent models will be generated in the final submission where a grid was used so that the model was run on varying values of C.

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.



```
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
```