SpringBoard Capstone Project 1 Final Report

Abuabbas Qureshi

The project I worked on was based off a Lending Club loan application dataset. This purpose of this project was to investigate the risk factors for an applicant and to determine which of the applicants could be risky clients to grant a loan to. This was done primarily through exploratory data analysis and touched upon basic machine learning concepts to derive an answer to the main question. While pursuing the question, I was interested in finding out some more supplementary answers to questions such as what kind of applicants who have had delinquencies in the past still prove to be good clients, what kind of applicants who are high earners still prove to be risky clients. I was very interested in finding out whether location would be a factor in determining the risk factors in a loan applicant. I was also curious to see whether the 2008 financial crisis has anything to do with the success or failure of a loan application. Some of these queries would be whether a high income applicant suddenly fell on hard times or not, or whether a low income applicant has a better chance of prospering now.

To achieve this, I had to perform some data munging to be able to perform feature selection better. Apart from data munging, there was plenty of data wrangling to get an in depth analysis on this data set. The variables were plenty however upon starting this project I saw that there were many of them containing missing values over 70%, because of this I had to drop the null values or determine whether I had to replace them with an average value. The TARGET variable was determined from the ‘loan\_status” column, I differentiated between a ‘Good Loan’ and  a ‘Bad Loan’ according to the values in the column. I categorized “Fully Paid” and “Current” loans as good loans and the other values to be bad loans.

For the data wrangling portion of the first capstone project I had performed various methods to achieve clarity in my analysis. Some of the values in certain variables were rounded up to help get an easier reading.

17 columns have missing data over 90%, and from the top 20, 3 have 75%+ missing data. These columns would not do the prediction model any favors, hence they were removed from the model data. I determined that 70% or more missing values in a column would be of no use, therefore they were removed from the model.

I had also used a heatmap to get a visual idea of how many missing values were in each column. This gave me a better idea in realizing the big picture of missing values.

For the outliers there were plenty and I had dug deeper to see what they were. For the categorical outliers it seemed like many were of the same kind but worded differently, I had either grouped them together if there enough of them or simply filtered them out for the label encoding. The numeric outliers I had mostly filled the missing values with either mean or medians depending on the variable. Otherwise some missing values also were filled simply with ‘None’ in certain cases such as employment status being NaN. Some of the columns were removed after determining that their correlation was not significant enough.

Data visualization is a key aspect of exploratory data analysis and this is why graphs and charts had become handy in understanding the dataset. I had used various kinds of graphs and plots to display my skill in matplotlib and pandas. The first one was a pie chart with percentages showing the good loans and the bad loans, it showed a large number of good loans to bad loans ratio giving the reader a general understanding that the company does a fairly good job granting loans to good applicants. I also used a distribution plot to display that the loans applied for were mostly granted by the lender. Bar plots were a handy tool to use in order to get a visual idea of certain variables. This would be in the case of loan purpose, showing a stark majority that debt consolidation was the key reason for applying for a loan, primarily credit card debt. This shows the nature of the loan applicant as well, carrying a habit of taking on debt and paying it off with more debt.  I had used a heat map to see correlations between these variables and found that loan amount and annual income had the most correlation.

I had also used a KDE plot to check the density and distribution of interest rates in the loans that were granted and from those loans which of them were good loans and which were bad loans. The good loans had mostly around 5 to 6 percent interest rates and a few just shy of 15%, while most of the failed loans had around 15 to 16 percent in interest rates granted to them.

For the location, I had grouped the states to regions first, this way I can see if there is a correlation to the culture of loan acquisition and region. The western states were more prone to acquiring a loan but so was New York from the north east region. The graph I used for this was a stacked line graph linked with the time the loan was applied for. This also showed a good idea of when the loans were applied for, picking up around 2012 and shooting up by 2014. This shows that Lending Club was not a company used for recovery directly from the 2008 financial crisis but may be a company sought after for help a few years after recovering from the crisis.

During the exploratory data analysis part of this project I was also able to apply the inferential statistics knowledge I acquired from the curriculum to understand my dataset better. There were about 840,000 rows with 74 variables that allowed me to get a thorough and accurate solution to the problem I was solving.

The main answer I was looking for was an algorithm to help the company get a near accurate approximation of whether a loan applicant would prove to be a good candidate to pay back or a bad candidate. To do this, I had to first get an understanding of the dataset and look into the variables to determine which of them would be significant to my cause and which of them could be taken out. From the significant variables I had analyzed them further through bivariate analysis techniques and pair the values together to see how they would impact the algorithm. With this, I was able to determine whether there was a correlation with location, job title, purpose, home ownership, interest rates, and debt to income ratios to the success of a loan application.

I had set a target variable to pitch against the variables and compare correlations. This target variable was based off of the loan status column that showed whether a loan was paid off, charged off, late by 30 or 60 days etc. I had set the paid off value as a good loan and the rest as bad loans.

One of the first variables I wanted to analyze was the ‘Loan Status’ category. This would give a very good head start in understanding the big picture of the loan applicants and their background, along with what kind of market Lending Club is attracting. To start this I plotted a bar graph to get a visual representation of the loan statuses. This bar graph clearly showed a vast majority honored their loan and they paid back in time. There was a chunk that did not pay back in time or at all. I also saw the categories in this variable and then determined which of these would be considered a good loan and which of these would be considered a bad loan. Upon reviewing the categories I decided that ‘Fully Paid’ should be the only one considered a good loan, the rest would be considered a bad loan, these would categories such as ‘charged off’, ‘late 120 days’ etc. After determining the categories I had formally split them into good loans and bad loans by code. After splitting the good and bad loans I wanted to get another picture of the loan status with this new category. To understand this better I then plotted a pie chart that also displayed a percentage of good loans and bad loans. From the findings, I saw that around 7% of the loans were unsuccessful and 93% were good loans. This helped me decide that I can use loan status as a target variable with the good loan and bad loan categories, that means I will be doing a classification model for this project.

Another key factor in this dataset I felt was the “dti” column that stands for Debt to Income. The debt to income ratio would help me understand whether someone is taking on a high liability with a lower income, indicating a higher risk. After having python describe the variable, I found that the average debt to income ratio was about 18 with a standard deviation of 17. This shows that the typical applicant is incurring more debt than their income.

Another feature that carries some understanding in my initial impressions was ‘purpose’. This would allow me to understand the motive behind the loan application. For this I plotted another bar graph with the value counts. This showed that a very strong majority of the applications requested a loan for credit card debt consolidation. Along with this, there were plenty of entries that had implied credit card debt consolidation in other linguistic ways such as ‘credit consolidation’ or ‘cc’ etc. Other than that some of the purposes were home renovations or other. Apart from the fact that I realized that this variable would carry a light weight on the algorithms, I realized that the nature and market of the loan applications were quite debt ridden, reflecting the nature of the loan applicants where they are finding a solution for debt with another debt.

The loans applied for against the loans granted by investors had approximately the same outcome reflecting that most of the applications were being approved without any issue, showing Lending Club would be an ideal loan granting company. Home ownership also showed that a majority of the loan applicants were mortgage holders and second came renting tenants. A mortgage would suggest a good credit score and stronger foundation of debt management compared to someone who rents or definitely compared to someone who is neither renting or owning their home.

Another variable that carries significant weight is “delinq\_2yrs” which indicates whether there were any delinquencies in the last two years for the loan applicant. The value count displayed that 716,961 applicants had no delinquencies and 113,224 had 1 delinquency in the last two years. These were the top two figures that mirrored the ratio for good loans and bad loans I mentioned above. To compliment this variable, there is another variable labeled ‘months since last delinquency’ that shows how many months it has been since the loan applicant’s last delinquency. The statistics on this variable say the average months is 34, with one fourth of the applicants have delinquencies in the last 15 months.

Annual Income has been an interesting variable as well in this dataset. The values recorded have astronomical differences. Although the average annual income is around $75,000, the minimum is $0 and maximum is around $9,000,000. These statistics show three pictures of the loan applicants. The average income shows there are a lot of standard applicants with true intentions. However, there are plenty of $0 earners and then going up to over 9 million dollars. This would suggest that there are applicants who either earn nothing and may not be able to pay back their loans easily or they do not report their income being cash they still carry a high risk of not paying back. On the other end of the spectrum, earners over a million dollars could be strategic defaulters. Strategic defaulters are those who get a loan and not pay back to write off a loss to avoid paying higher taxes.

Following the financial analysis, I started to look towards geography for a better understanding. The findings showed that location was insignificant because the ratio between the loans granted and loans paid back were the same between all states and regions. Job title also showed very little significance. However the delinquencies in the past 2 years, purpose, and debt to income ratio showed better significance with our algorithm as they had showed a clear picture of the nature of the applicant and his financial habits. There was a higher correlation between these variables and the target variable that I had set. The purpose column mostly had one common value written in different ways, which was that its purpose was to pay off credit card debts. This made better sense as seeing it in the correlation values.

For the machine learning aspect of the project, I applied three algorithms to build the predictive models. These were Logistic Regression, Decision Trees, and Random Forests. These are classification based algorithms that helped me achieve my goal in this project. Before applying the algorithms, I had decided to first label encode the values so to make it easier for the algorithms to perform its calculations. This was done with the ‘preprocessing’ feature in sklearn. From this, 7 columns had been label encoded. From here I had then performed the train test split method on the dataset creating a training set and a testing set.

The first algorithm I applied was the Logistic Regression model. This model should not be confused with the Linear Regression Model, as Logistic Regression is more of a classification algorithm and not a regression algorithm. For Logistic Regression the trade-off parameter that determines the strength of the regularization is called C,, and higher values of C correspond to less regularization. In other words when you use a high value for the parameter C, logistic regression will try to fit the training set as best as possible, while with low values of the parameter C, the models put more emphasis on finding a coefficient vector “W” that is close to zero. . Another interesting aspect of how the parameter C acts, using low values of C will cause the algorithms to try to adjust to the majority of data points, while using a higher value of C stressed the importance that each individual data point be classified correctly. So a higher C value could result in overfitting. Luckily, for our model the algorithm was more straightforward hence we did not need to tweak the C value at all. The training set and the testing set produced excellent results with 96% accuracy on both, so the logistic regression model was deemed acceptable.

After this I had applied the Decision Trees model. This model is widely used for classification and regression tasks, Essentially, they learn a hierarchy of if/else questions, leading to a decision. When building the tree, the algorithm searches over all possible tests and finds the one that is more informative about the target variable. The parameters that control model complexity in decision trees are the pre-pruning parameters that stop the building of the tree before it is fully developed. Usually, picking one of the pre-pruning strategies is sufficient to prevent overfitting. This would be done by either setting max\_depth, max\_leaf\_nodes, or min\_samples\_leaf. This model produced around a 95% accuracy which was deemed good from my mentor.

Following these two algorithms, I decided to try out the Random Forest model on this training and testing set. A drawback of decision trees is that they tend to overfit the training data, random forests are one way to address this problem. A random forest is essentially a collection of decision trees, where each tree is slightly different from the others. The idea behind random forests is that each tree might do a relatively good job of predicting, but will likely over fit on part of the data. If we build many tree, all of which work well and over fit in different ways, we can reduce the amount of overfitting by averaging their results. This reduction in overfitting, while retaining the predictive power of the trees can be shown using rigorous mathematics. To implement this strategy, we need to build many decision trees, each should do an acceptable job of predicting the target and should also be different from the other trees. Random forests get their name from injecting randomness into the tree building to ensure each tree is different. There are two ways in which the trees in a random forest are randomized, by selecting the data points used to build a tree and by selecting the features in each split test. The Random Forest model after decision trees produced an even better result with around 96% accuracy. These models are good for classification problems and the results clearly state that as well. After applying these models, I had posted a ROC curve to get a visual work of all this. The curve gave out a good result since a visual technique to analyse this would be that the curve should be curving towards the upper left corner of the graph and this was the case over here.

The Lending Club dataset has proven to be quite sufficient in this entire project as I did not have to look to other sources to build an effective predictive model. Although there were plenty of variables that needed to be dropped because they either lacked entries or lacked any significance towards the algorithms, I still had a multitude of variables that were effective in this project. With a 95% average accuracy I feel I have put together a good collection of algorithms for building the right predictive model to help a lending company such as Lending Club etc.