D606 Task 2: Implements a Data Analytics Approach

By: Kevin Sandoval

A.

The research question I proposed in Task 1 was whether I could find important variables that contribute to whether a customer would default on their loans, and then create a machine learning model that helps predict the loan defaulters. This can have significant implications for the success of the company. Using the underlying data to help predict whether a customer would default could assist the company in their decision-making. My hypothesis for this result is that there will be variables that, when used in a Random Forest Classifier model, will effectively predict loan defaulters. The null hypothesis would be that there is no significant relationship between the variables and whether a customer defaults.

B.

I decided to use one of the WGU-supplied datasets for this analysis. The “Loan Default Data” dataset was chosen. The data consists of a variety of customer information, both personal to the customer, such as income, age, or gender, as well as the customer’s information regarding the company, such as loan type, loan purpose, and interest rate. One advantage of using this data is that it was easy to obtain, as WGU supplies it. One disadvantage is not knowing if the data is robust, real, or randomly generated. Real data can often contain patterns, which is what this analysis is trying to find, but generated data at random can potentially be too random and not have any meaningful underlying connections.

C.

My process for the data extraction and preparation portion of the assessment was the longest and most arduous process. I first read in the data and checked for any duplicate values. From the screenshot below, we can see that no duplicate values were detected. One advantage is that removing duplicate values will increase the model's effectiveness. A disadvantage could be that the amount of data removed could be impactful if there were a lot of duplicated data.

A screenshot of a computer program

AI-generated content may be incorrect.

The next step was to check for any missing values in the data. One advantage of doing this is ensuring that no missing data affects how the model gets created. A potential disadvantage is that how the missing data is treated could affect the data quality and thus the model's quality.

A screenshot of a computer code

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

From this output, we can see a lot of missing values that will need to be dealt with. To figure out how to deal with the missing values for the quantitative variables, such as “income”, I decided to look at their respective histograms. I am only including one histogram in this document, as it is repetitive to include all of them, but all quantitative variables were checked.

A screenshot of a graph

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

From this example, I saw that income was skewed towards the low end, so I decided to impute the missing values with the median in this case. An advantage of this is that it is an easy computation, and choosing the median won’t affect the data too much. A disadvantage is that the data is made up, so it could affect the model if the model finds any trends in the data related to the made-up data. The final screenshot shows no more missing values in the “income” column.

For the qualitative variables with missing data, I decided to look at their distributions as well and then choose how to treat them. For the case of “loan\_limit”, the distribution and treatment of missing values are shown below.

A screenshot of a computer

AI-generated content may be incorrect.

A screen shot of a graph

AI-generated content may be incorrect.

In this case, we could see from the first code snippet that the value\_counts function returned a significant skew towards “cf.” As such, I decided to impute the missing values with the mode. An advantage of this is that it is easy to do and shouldn’t change the distribution too much, as there isn’t much missing data. A disadvantage is that imputing data is not real; thus, when creating the model, it could affect performance. The code snippet for that is also shown above. I included a line of code that returns the count of missing values, and we can see that it returned zero, so the missing values have been treated.

I performed that technique on all the quantitative and qualitative columns, and the result is shown below.

A screenshot of a computer program

AI-generated content may be incorrect.

Now that the missing values have been treated, I decided to check for any outliers in the quantitative data and then treat them as necessary. Like the previous part, I will include 1 example in this report, but I will not include every column to prevent repetitiveness. I’m choosing to use ‘loan\_amount’ as the example.

A screenshot of a computer

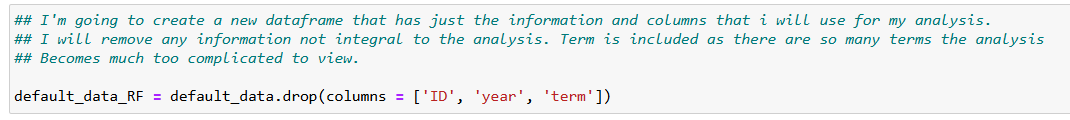
AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

In the first code snippet I calculated the ‘whiskers’ of both the left and right side using the .describe() function to find the quartiles and then multiplied by 1.5 to get the whisker values. In this case I decided to drop all the outliers outside the right-side whisker. The resulting plot still has a few outliers, but I’m choosing to retain them due to diminishing returns on removing data. But now we can see that the boxplot looks much better. An advantage to removing outliers is that the model will focus on the more common cases, rather than being skewed by some edge cases. A disadvantage to this is that removing data will sometimes make the model less effective, and it won’t account for the high loan amounts in this case.

The next step in the data preparation process was to drop unnecessary columns. I chose to drop ‘ID’, ‘year’, and ‘term’. I wanted to include ‘term’, but there were so many unique values as a qualitative variable that there were diminishing returns on the analysis.

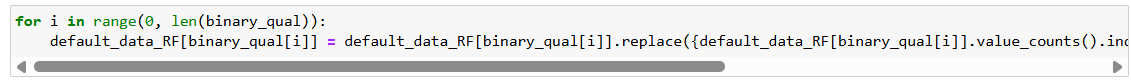


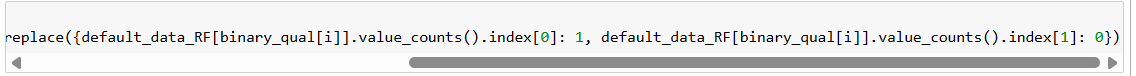
The next step was to re-label the qualitative variables. I wrote a section of code to get a list of the categorical variables that have only 1 unique value, 2 unique values, and more than 2 unique values. Snippet is shown below.

A screenshot of a computer code

AI-generated content may be incorrect.

For the binary categorical variables, I remapped the most common result to a value of ‘1’ and the less common result to ‘0’. The advantage of doing this is that it makes processing a categorical variable possible for model creation. A disadvantage would be that it is harder to understand what each value means without a list of what the 1s and 0s mean.





I used one-hot encoding to remap the categorical variables with more than two unique values. The code is shown below.

A screenshot of a computer code

AI-generated content may be incorrect.

A screenshot of a computer program

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

I included a screenshot of some new columns showcasing the change in the categorical variables. The advantage of doing this is that it makes processing a categorical variable possible for model creation. A disadvantage would be that it is harder to understand what each value means without a list of what the 1s and 0s mean.

For the next part of the preparation, I used a heatmap to show the correlation between each variable. An advantage of this is that it can help reduce overfitting as well as reduce any multicollinearity issues that may be in the data. A disadvantage is that removing columns could remove important and insightful data.

A screenshot of a computer

AI-generated content may be incorrect.

From this, we can see a strong correlation between property value and loan amount. This blends into the next part of the preparation, where I ran into a problem with the data. A lot of the missing data from ‘rate\_of\_interest’, ‘interest\_rate\_spread’, ‘upfront\_charges’ directly correlated to a ‘1’ on status. Because of this, the model started performing at a ‘perfect’ rate because it knew exactly how to predict. To combat this, I dropped ‘rate\_of\_interest’, ‘interest\_rate\_spread’, ‘upfront\_charges’, and ‘LTV’. Afterwards, I replotted the heatmap.

A screenshot of a computer screen

AI-generated content may be incorrect.

As we can see, loan\_amount’ is highly correlated with ‘income’ and ‘property\_value’, so I also chose to drop ‘loan\_amount’. The heatmap of the categorical variables is shown below.

A screenshot of a computer

AI-generated content may be incorrect.

The numbers are hard to discern, but based on the color, it looks like ‘Security\_Type’ and ‘Secured\_by’ both have high correlation with other variables, so I will remove those and replot them.

A screenshot of a computer

AI-generated content may be incorrect.

The new plot looks much better, so I will retain the remaining variables.

That was the last step in the preparation. The edited dataset will be included in the submission.

D.

To start the analysis portion of the assignment, my first step was to create a dataframe of the explanatory variables and a dataframe of the target variable, ‘Status’. The advantage to this is that it allows me to more easily access the target and explanatory variables when creating the training, validation, and test sets. The only disadvantage I could see would be that it could complicate the process slightly, and if the new dataframes are named something odd, it could cause confusion.

A close-up of a white background

AI-generated content may be incorrect.

As mentioned above, the next step was to create the training, validation, and test sets. I created the initial split and then took the test split and split it into the validation and test split. The splits will be 70-15-15. An advantage of this is that it allows the model to train itself on the data and be evaluated afterwards properly. A potential disadvantage could be the size of the test data. Since I’m also using validation data, the test data size is cut in half, which could affect the analysis.

A screenshot of a computer program

AI-generated content may be incorrect.

The next step in the analysis is to create the model, train it on the training set, and then create the predicted values. Afterwards, I created a confusion matrix to visualize the model’s effectiveness, a classification report, and the Gini Importance of each variable. The code for the Gini importance was taken from GeeksforGeeks' “*Feature Importance with Random Forests”*. I also included an ROC curve and the model's AUC score. The results are shown below.

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer program

AI-generated content may be incorrect.

A graph with blue lines

AI-generated content may be incorrect.

A chart with numbers and a few labels

AI-generated content may be incorrect.

A screen shot of a graph

AI-generated content may be incorrect.

A white rectangular object with black text

AI-generated content may be incorrect.

These results show that the model performs very well for ‘0’ or the non-default prediction. The model struggled with the recall and F1 scores for predicting that a customer will default. We can also see from the AUC score that the model had an accuracy rating of around 86%, which is relatively high. We can also clearly see through the output and the graph which variables have the most significant feature importance to the model. An advantage of this approach is seeing how the model performs initially. A disadvantage is that this doesn’t help showcase how to improve the model, and it is up to the user to try to interpret the results.

For the last part of the analysis, I implemented a grid search function to find the optimal parameters for the model creation. I then re-ran the model and used the same confusion matrix, Gini Importance, classification report, ROC curve, and AUC to evaluate the model. The code for the Gini importance was taken from GeeksforGeeks' “*Feature Importance with Random Forests”*.

A screenshot of a computer program

AI-generated content may be incorrect.

A screenshot of a computer screen

AI-generated content may be incorrect.

A screenshot of a computer program

AI-generated content may be incorrect.

A graph with blue lines

AI-generated content may be incorrect.

A chart with numbers and a few lines

AI-generated content may be incorrect.

A screen shot of a graph

AI-generated content may be incorrect.

A white box with black text

AI-generated content may be incorrect.

The code snippets show that the new model was created using the parameters provided from the grid search. The results show that the model may have improved slightly. An advantage to this technique is that it can improve the model's effectiveness. A disadvantage is that the grid search is computationally extensive and takes significant time to run. In my case, the grid search took about 5 minutes to finish, and the results were negligible, so it wasted time.

E.

My hypothesis, mentioned in Part A, is proven correct. The results show that the model created was 86% accurate at predicting loan defaults. Thus, there is a significant relationship between some variables and defaulting.

Through the analysis, I found that some data needed to be omitted due to data quality issues, and some variables had strong correlations with each other that also needed to be dropped. The Gini Importance helped determine which variables impacted the model the most, which helped answer the first part of the question, which was to find the impactful variables to the default rate. Overall, the analysis was successful, but it has some limitations. For one, I had to impute a lot of missing values initially. This can lead to some false conclusions in the data. Another issue was that many of the missing values came in bunches that were directly correlated to the status of a customer. I initially tried dropping the missing values, but it also dropped all the ‘1’ results in the Status column. This made the model not work as intended, and to work around this, I had to drop a few of the columns entirely to get the model to function.

This analysis is helpful in the company at hand. I recommend using this model to help predict customers who won’t default, rather than those who will. The model performed better when classifying the customers that way, and I think its utility also fits the company’s desire. Another recommendation I would have is to try to find the missing data in the “rate\_of\_interest”, “interest\_rate\_spread”, and “upfront\_charges” columns. For some reason, 99% of the missing data was directly correlated to the customer's status, as they had defaulted. I would explore how and why that missing data may have happened. Those variables seem like they would be beneficial to the analysis, but were entirely unusable in this case because of the missing data.

My last recommendation would be to examine the top 5 variables regarding Gini Importance. There is a clear top 5 in terms of the value, and they are “credit\_type\_EQUI”, “property\_value”, “dtir1”, “income”, and “Credit\_Score”. This would imply that a customer's specific credit types could be the best indicator for predicting what will happen with their loan. The other variables are also essential and should be investigated further.

F.

GeeksforGeeks (April 5th, 2024) *Feature Importance with Random Forests* Retrieved May 6th, 2024 From <https://www.geeksforgeeks.org/feature-importance-with-random-forests/>