**Machine Learning for Public Policy**

**Homework 2**

**Diego Escobar**

**Parts 1 and 2:**

For the first part I generate summary tables for a list of variables, histograms, tabulations, correlation tables, scatter plots and compute the means for different crossing of values of categorical variables and display them in a heat map.

Table 1: Summary Statistics

|  |  |  |  |
| --- | --- | --- | --- |
|  | Serious Fin. Distress | Debt Ratio | Number Of Dependents |
| count | 41016 | 41016 | 39979 |
| mean | 0.1614 | 331.4581 | 0.773231 |
| std | 0.367904 | 1296.11 | 1.121269 |
| min | 0 | 0 | 0 |
| 25% | 0 | 0.176375 | 0 |
| 50% | 0 | 0.369736 | 0 |
| 75% | 0 | 0.866471 | 1 |
| max | 1 | 106885 | 13 |

Figure 1: Age Histogram

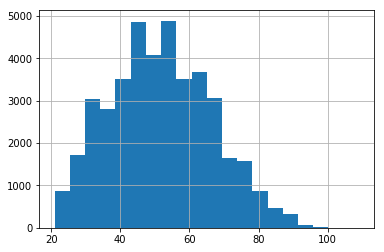


Table 2: Number of dependents

|  |  |
| --- | --- |
| Number of Dependents | Frequency |
| 0 | 23503 |
| 1 | 7211 |
| 2 | 5539 |
| 3 | 2666 |
| 4 | 786 |
| 5 | 201 |
| 6 | 51 |
| 7 | 12 |
| 8 | 7 |
| 9 | 2 |
| 13 | 1 |

Figure 2: Correlations

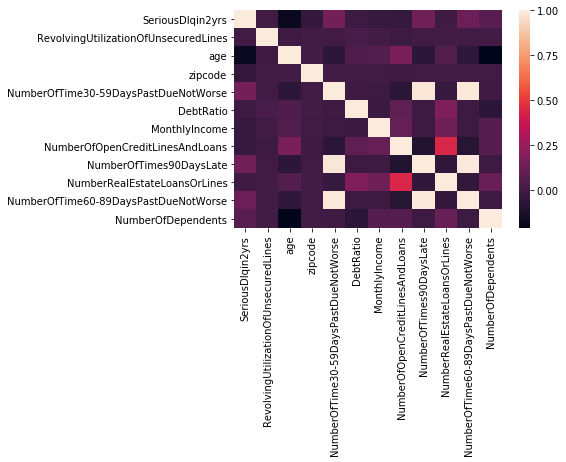


Figure 3: Scatter Plot

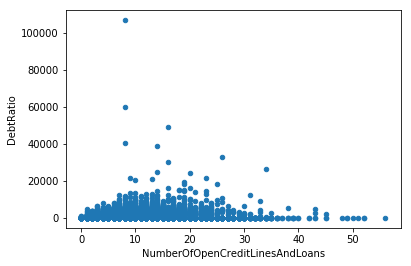
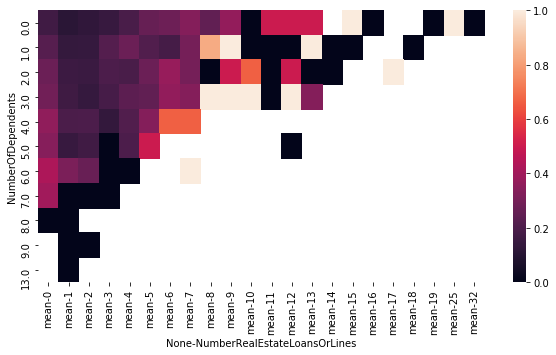


Figure 4: Proportion of Financial Distress by N° Dependents and N° Loans/Lines



**Part 3:**

For this part I simply wrote functions that replace outliers with a desired value. For this case I used 3 times the standard deviation of each variable as outlier and replaced it with that maximum value (equivalent to winsorizing). I also implemented functions to generate dummy variables indicating missing values for each variable and that also replace with zero the respective observation, and a function that takes null values and replaces them with the mean or median.

**Part 4:**

For this I created a function that creates bins based on previously specified values (e.g. age 0 to 25, 25 to 50, and so on) and a function that splits the data into N bins. I then construct dummy (indicator) variables to identify the categories to which each bin belongs. For the purpose of this example I converted age to bins as specified in the first case, and the remaining continuous variables to equally sized bins. Note that the advantage of the first method is that it can generate more meaningful separations of the data, but it only makes sense if we can give some interpretation to the cutoff values chosen for the bins.

**Part 5:**

For the purpose of having a more complete pipeline I implemented two different models: decision tree and random forest. In the first case the variables are simple selected in order to maximize the information gain, and I arbitrarily decided to set a maximum tree depth of 10 (ideally one would choose this though cross-validation). The second case is simply an ensemble of the first one, and the procedure is similar but a subsample of the data is used on each split to avoid overfitting.

**Part 6:**

In order to evaluate the models, I compute the “score” parameter that measures the accuracy of the model. Prior to doing this I randomly split the sample into training and testing data (2/3 and 1/3 of the data, respectively). This avoid the problem of evaluating the prediction accuracy in the same sample where it was estimated, which would in turn generally favor selecting as many variables as possible to get a better fit, and thus overfitting the data.

In the case of the tree the accuracy on the test data is **94.23.** The accuracy of the random forest is slightly higher, reaching **94.78.** The reason why the scores are relatively similar is that choosing a limited maximum depth for the tree is an alternative method to avoid overfitting. As a comparison, setting that parameter to 100 for instance leads to an accuracy of 91.48 on the test data.