04-802 E: Programming and Problem Solving for Data Analytics
Final Project: Building a Credit Approval Model using Neural Networks

[Individual Project]

Instructor: George Okeyo TAs: Janvier, Faustin and Pacifique

Name: Oluwadara Adedeji

Andrew ID: oadedeji

1.0 Background

This project uses neural network for building a credit approval model. The dataset was anonymous, and the features have been reduced to protect the confidentiality of the project. This is because the dataset relates to credit card applications. The dataset was retrieved from [1]. About 690 observations are in the dataset, and the dataset has 16 columns with 15 features and one column for the dependent variable. The model was trained using Sklearn neural network MLP classifier. This is because this is a classification task. It is assumed that "+" denotes an approval while "-" denotes disapproval. Two relevant metrics are used to evaluate the performance of the model. They are accuracy and precision. Also, cross-validation of 10 k-folds was used for this evaluation. The model was also tuned using grid search.

2.0 Problem Statement

Identifying customers that are qualified for credits can be challenging. This model helps to classify credit card applications, based on some features, although the features are not labeled to protect the confidentiality of the customers. This solution is important to reduce the risk of people defaulting on their credits. However, it is also important the model is not biased.

3.0 Methods

This section discusses the methods used in carrying out the project.

Toolkits: Numpy, Pandas, Matplotlib, Seaborn, Sklearn

3.1 Data Preparation

The data was loaded using Pandas. It was discovered that there are 16 columns and 689 rows in the dataset. There are also missing values in the dataset, but this wasn't obvious initially until all the unique values in each column was printed out and it was discovered that some values are represented as "?" in the data. All the "?" in the dataset were then replaced with null values. This made it easier to view the null values. Null values in columns with numeric values were filled with the mean of the column while the null values in the columns with categorical data were dropped. At the end of this, it was discovered that only about 2.75 percent of the dataset was dropped.

3.2 Exploratory Data Analysis

Two visualization tools name Matplotlib and Seaborn were used for visualizing the dataset.

First, some selected columns with numerical values were viewed using histograms to get the distribution of the data for that column. The bins size was selected to be 20. Also, a scatter plot was used to show how the distributed are distributed in coordinate distribution. This made it easier to view outliers based on data points that are far away from others. Furthermore, a boxplot was used to view the outliers based on data points that are out of the quartile distribution. Also, at each instance, the mean of the column is computed, to visualize how far away from the mean the points being visualized are, based on the plots.

3.3 Preprocessing, Feature Selection and Engineering

3.3.1 Dealing with null values

First, the columns which have numerical content but are not in numerical format are converted to relevant numbers. Afterwards, the missing values in these columns are filled with the mean of the columns. Also, the rows with null values, with categorical data were dropped, and this dropped only 2.75 percent of the dataset. Although using the most occurring data for the categorical data was considered, this was discarded because using this might cause bias in the data, and considering how sensitive the data is, this is not wanted.

3.3.2 Dealing with Outliers

As seen from the visualization, some columns had outliers. These outliers were treated using the Winsorize method of dealing with outliers. This method limits outliers with an upper and lower limit for the maximum and minimum points for the columns. The boundary selected were 0.01 quartile and the 0.95 quartile for each relevant column. Although this was picked arbitrarily after testing different boundaries to see which deals with the outliers better for all the columns, an alternative way to pick the limit would be to select based on the inter-quartile range for each column. After dealing with the outliers, it was observed from the boxplots that indeed the outliers have been treated.

3.3.3 Dealing with Categorical Data

The “+” column was observed to be of two values. And this was the dependent variable, hence, this was changed to numerical values using label encoding. So as not to give an unfair weight to columns which have 3 and more unique categorical values, one hot encoding was used to get the dummies of these columns. This was then merged with the other dataset.

3.3.4 Scaling

It was observed that some columns had higher values than others, thus, scaling of these columns was done. This was limited to columns which had numerical values in the loaded data before the data was formatted from categorical to numerical data. The type of scaling done is the standard scaling which makes use of the mean and the standard deviation.

3.3.5 Feature Correlation

The correlation coefficients between all the features and dependent variable “+” was generated using an heatmap, which was annotated to show the correlation coefficients. It is observed that many of the features are weakly correlated with the “+” column (i.e, they have a correlation coefficient of around 0 to +/-0.2. Also, the “ t” feature is strongly negatively correlated with “+” column.

3.3.6 Feature Selection

All the columns except the “+” are set to be the independent variable “X”, while the “+” column” is set to be the dependent “y” variable. This is later used for the machine learning model.

3.4 Model creation and Evaluation

For the model, the Sklearn Neural network is selected. Since the task is a classification task, the Multi-Layer Perceptron (MLP) classifier is used. As requested,10 folds are used for the cross-validation. The metrics used to estimate the performance are the accuracy and the precision. These are selected because they are two relevant metrics for classification tasks. They also show how well the model performed on the testing data (in cross-validation). The accuracy tests the ratio of sum of true positive and true negatives, and all the outputs derived (i.e sum of true positive, true negative, false positive, false negative). Likewise, the precision computes the total number of true positives divided by the sum of true and false positives. This is because we do not want too many false profits which increases our risk. The random_state is set so there won’t be random values everytime the code is run. Also, the maximum iteration was increased from the default so it can accommodate and not to get an error that the optimizer didn’t converge before termination of the code. It should be noted that eventhough cross_val_score was used, cross_validate can also be used to get more than one metric at a time.

3.4.1 Hyper Parameter Tuning

Some parameters were tuned on the model. These parameters are activation, solver and learning rate. The relevant metrics are retrieved from the online documentation. Grid search using 10-fold cross-validation is used to get the best parameters. These parameters are then passed into the model, and it is observed that there is an increase in the accuracy and precision of the model

4.0 Results and Discussions

It was observed that there is a better performance using hyperparameter tuning. The accuracy before tuning was around 0.81 while the performance after tuning was around 0.83 approximately. This shows an increase of 0.02 in the accuracy. Likewise, for the precision an increase from 0.689 to 0.692 is observed. These are also visualized using a plot.

5.0 Conclusion

In this project, classification of credit approval is done. First, the dataset was visualized using visualization tools. Outliers were discovered in the data, and these were dealt with. Also, one-hot encoding was done on the data and some features were also scaled to reduce imbalance. Missing values were also discovered and dealt with. The sklearn multi-layer classifier was used for this project. The relevant metrics used for model evaluation are accuracy and precision. Furthermore, cross-validation is used for this evaluation. The model was also tuned using grid search and it was observed that the tuned model performed better.

6.0 References

[1] C. Dua, Dheeru and Graff, “Credit Approval Data Set,” *University of California, School of Information and Computer Science.* https://archive.ics.uci.edu/ml/datasets/Credit+Approval (accessed Dec. 09, 2021).