Background and Problem Description

Many credit card applications are received by commercial banks. Many of them are turned down for a variety of reasons, such as large debt balances, insufficient income, or too many inquiries on a person's credit report. Manually assessing these programs is tedious, time-consuming, and error prone. Fortunately, with the use of machine learning, this work can be automated, and almost every commercial bank does so nowadays. We'll use machine learning techniques to create an automatic credit card approval predictor in the notebook attached and this report discuss the key findings and approaches made [1].

Methods

The approach was simple as follows: the work was divided into four important parts

Data preparation: where we downloaded and loaded crs.data dataset, then do data cleaning where we removed special character (?) and replace it by null values in the whole dataset and then transformed continuous variables (A2) that was objects into floats then save the resulted dataframe as csv for future use purposes.

Exploratory Data Analysis: where we identified the presence of outliers and Null values and visualized the distributions of the data among different categories of categorical variables (categorical variables are columns named 'A1','A4','A5','A6','A7',A16") and the ratios at which these categories contribute to the target variable(A16).This visualizations continued to the distributions of continuous variable in the columns named ('A2','A3','A8','A11','A15').

Pre-processing, Feature selection and Engineering: In this part we started replacing missing values with mean for continuous variables and mode for categorical variables. This was followed by removing outliers by Flooring and capping technique**.** In this quantile-based technique, we will do the flooring (25th percentile) for the lower values and capping (for the 75th percentile) for the higher values means that the values less than 25th percentile are mapped to the 25^{th} and those higher than 75^{th} percentile are mapped to 75^{th} percentile. Another important thing that we tried under here was to data scaling for continuous variables had forced them to be in range of 0 and 1 using minmaxscaler, so that they may have the same weight in prediction. We also performed encoding where we used label encoding for variables with two categories and hot encoding on variables with more than two categories. At last, we visualized correlation matrix using heatmap to explore the relationship between predictor variables and target variables.

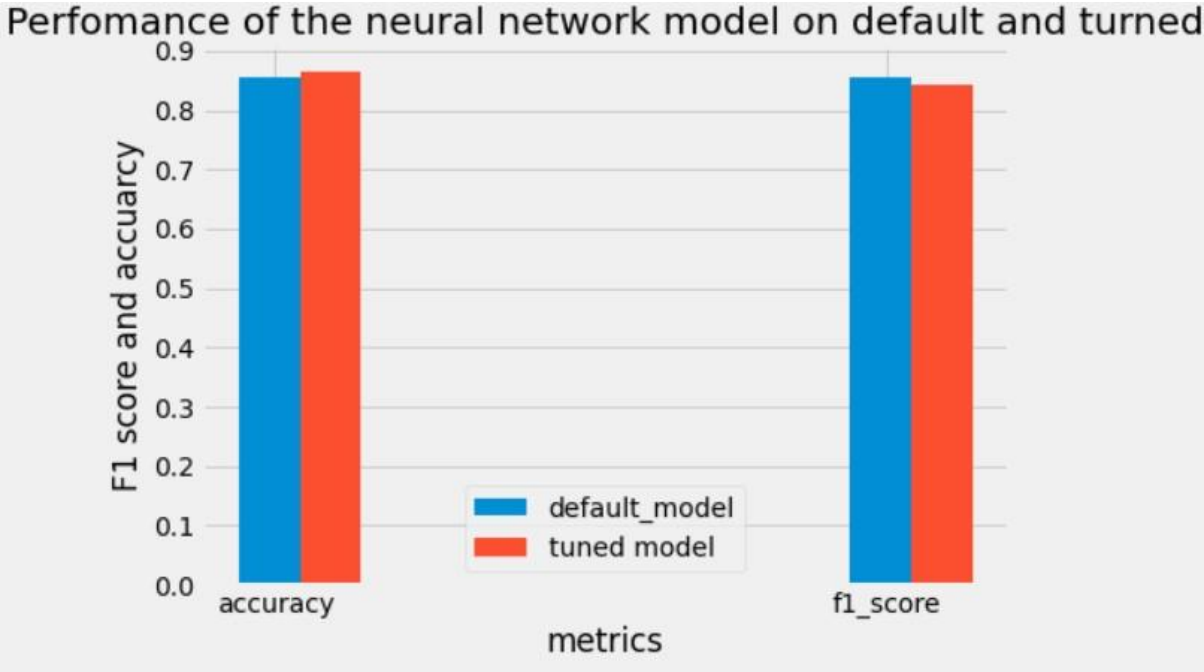
Model creation and Evaluation: Using scikit-learn we built a default neural network model with predictor variable (A1, A2, A3, A4, A5, A6, A7, A8, A9, A10, A11, A12, A13, A15) and target variable A16. And evaluated the performance using 10-fold cross validation. After these, we used hyperparameter turning with (Grid search) to find the best parameters and can give high accuracy and then use that to build a turned neural network model which is also evaluated by 10-fold cross validation. Finally, we visualized the bar graph to compare the performance of the to model using F1 score and precision. The reason for choosing F1 score is since F1-score combines the precision and recall of a classifier into a single metric by taking their harmonic mean and can be a good metrics to compare two classifiers.

Results and Discussion

After loading the dataset, we found that the dataset has 690 rows and 16 rows, according to the meta data provide this row has names A1, through A16 where A16 will be target variable and the least be predictor variables. Checking the unique values for each column we find that columns contain (?) special character which may probably stand for missing values that is why we changed them to nan .by checking info of the dataset we also find that continuous variables variable like A2, A14 are object, which is contrary to what meta data says, so we changed them to floats.

After checking for missing values, we find that we have 67 missing values that is approximately 9.7% and A1 and A2 were the having many missing values compared to other variables (12 each). we also find that the data set has 301 outliers in all variables that is probably 43.6% of our data (A15 has almost a half of these outliers). This percentage is huge, so we must pay attention on handling these outliers.

From the visualization we find that 55% of the credit card applicant are rejected means they are assigned (-) in our dataset. We also find that for A1 variable \category b account for 69% and 31% for and a contribute much to the neglection of credit card than b does. More on categorical variables are available in notebook. For numerical variables they were nonuniformly distributed and all were right skewed distributed. The model built after doing all data preparations and explanatory data analysis has achieved 85.5% accuracy for default and 86.52% for turned model.



Conclusion

The report presented Neural network machine learning model that predict credit card approval, the model achieved 86% Accuracy.

Reference

1. https://github.com/pranavtumkur/Predicting-Credit-Card-Approvals-UsingML/blob/master/notebook.ipynb