1.0. Background

Credit approval is the process a business or an individual undergoes to become eligible for a loan or pay for goods and services over an extended period [1]. This work hopes to predict which individual should be approved for loans based on some related features. This is a classification problem, and a neural network-based model is used to make the prediction. The performance of a baseline model as well as a hyperparameter tuned model is computed and the best performing model is identified. Visualization of the performance of the models is also presented.

2.0 Problem Description

Financial Institutions rely on credit risk scores to make various lending decisions for unsecured credit, such as loan approval, interest charges, repayment tenure, among others [2]. At present, traditional credit evaluation methods have become deficient since, in many developing regions across the world, such as Latin America, sub-Saharan Africa etc., a large percentage of the population is unbanked, resulting in limited traditional data available for a sound credit assessment. This leads to financial exclusion of many in these regions, given that they have no credit. Therefore, it has become important to explore new alternative data sources that can be used in place of credit history for customers, to improve the risk assessment process and credit approval in a timely manner, so that more financial inclusion can be achieved. This work aims to explore the use of alternative data to determine credit worthiness/approval for customers.

3.0 Methods

3.1 Toolkits: Python **Libraries**: NumPy, Pandas, Scikit Learn, Matplotlib, Multi-layer Perceptron

3.2 Dataset

The dataset used for this work was retrieved from [3]. The dataset concerns credit card application and had all the feature names and values well anonymized to protect the confidentiality of the data. The dataset has a good mix of attributes as well as some missing values which poses an interesting challenge. I also had to add the column names using the data attributes information provided. Therefore, the dataset has a shape of 690 rows and 16 columns with 9 categorical predictor variables and 6 numerical predictor variables.

3.3 Data Preprocessing

3.3.1 Exploratory data analysis

For data processing, I checked the information about the dataset and noticed that two columns that should be numerical had object datatype. I investigated further and found the presence of question marks (?) in some more columns to represent missing values. I also checked for the presence of outliers in the dataset using two methods, boxplots and scatter plots to show it through

visualization and using interquartile range (IQR) which is a commonly used and most trusted approach in the research field [4]. I noticed that all the numerical columns have outliers ranging from 13 in the least column and 113 in the column with the highest outliers.

3.3.2 Handling missing values and outliers

For the missing categorical values, I dealt with this by replacing "?" with the mode of the column except column A4 where I used the letter "t" judging from the data attributes provided which specified that the column should have letters u, y, l & t but only 3 letters were present. For the numerical columns, I replaced the "?" with the median. I used median instead of mean because I had observed outliers in the data and this will affect the value of the mean.

To deal with outliers, I choose not to drop them because they are a lot, and my training data is small. Dropping the outliers would mean removing at least 16% or more of my training data which is a lot considering I only have 690 rows in the first place. I handled them using log transformation which de-emphasizes the effects of outliers/skewness on a dataset by getting the log of the values. This method also helps with scaling. I used one hot encoding with *get_dummies* on the categorical features and dropped the initial columns because using label encoding would give weights to each of the values and in this use case, the values in my categorical features have the same weights.

All columns except the class attribute column (A16) were set as the independent variable (predictors), while the class column was set as the dependent variable (target). I also converted the + in the target variable to 1 and the – to 0.

3.4 Modeling and Evaluation

3.4.1 Modeling

This task is a classification problem, and it is requested that I make use of a neural network-based model. I used the Multi-layer Perceptron Classifier from scikit-learn. I used this algorithm to build 2 models: the baseline model which makes use of default hyperparameters of the algorithm in the prediction and a hyperparameter-tuned model where I used grid search to find the optimal values for 3 hyperparameters. In each case, the algorithm was fitted using the predictor features with predictions done on the target feature.

3.4.2 Evaluation and Metrics

I evaluated the performance of the models using 10-fold cross-validation using Recall and F1 score as the metrics. I used recall because in this credit approval use case, recall is an important metric [5] which is more concerned with false negatives (the model predicting that someone is not going to default but they do) than false positives (the model predicting that someone is going to default but they don't). I also used F1 score as the second metric because it provides a single score to measure both Precision (ratio of true positives to the total positives predicted by a classifier, where positives denote default cases) and Recall. The cross-validate function was selected instead of the cross val score function because it can generate multiple scoring metrics. The desired metrics (i.e.,

recall, f1-score) are passed into the scoring parameter of the cross-validate function. For each metric, using 10-folds, 10 scores are returned, therefore, I got the mean in each case.

3.4.3 Hyperparameter Tuning

I used Grid Search which combines passed hyperparameters to get the best combination for the model. The best performing combination was retrieved using best_estimators_ attribute, while the best parameters were also retrieved using best_params_. I plotted the scores obtained against the model with matplotlib.

4.0 Results and Discussions

For the results, I compared the performance of both models, and observed that across both metrics used (F1 Score and Recall), the hyperparameter-tuned model performs better than the baseline model which uses default hyperparameters. For the tuned model, the recall score is 0.856 which implies that 85% of the data points/observations that was labelled in a particular class were labelled correctly. After the grid search, the best parameters were: hidden layer sizes = 150, activation function = identity and solver = sgd.

5.0 Conclusion

In conclusion, I was able to perform a classification task using a neural network-based algorithm on baseline and hyperparameter-tuned models. The performance of the models was computed with cross-validation on F1 score and Recall. The result was presented visually using a bar graph and line graph. The best performing model was identified to be the hyperparameter-tuned model and the best parameters were also identified using grid search.

6.0 References

[1] "Credit approval - benefits," *Referenceforbusiness.com*. [Online]. Available: https://www.referenceforbusiness.com/encyclopedia/Cos-Des/Credit-Approval.html. [Accessed: 09-Dec-2021].

[2] S. Deoras, "Using AI/ML to identify opportunities & challenges with credit risk scoring," *Analyticsindiamag.com*, 02-Nov-2020. [Online]. Available: https://analyticsindiamag.com/using-ai-ml-to-identify-opportunities-challenges-with-credit-risk-scoring/. [Accessed: 09-Dec-2021].

[3] "UCI machine learning repository: Credit approval data set," *Uci.edu*. [Online]. Available: https://archive.ics.uci.edu/ml/datasets/Credit+Approval. [Accessed: 09-Dec-2021].

[4] “Detect and Remove the Outliers using Python,” *Geeksforgeeks.org*, 23-Feb-2021. [Online]. Available: https://www.geeksforgeeks.org/detect-and-remove-the-outliers-using-python/. [Accessed: 09-Dec-2021].

[5] S. Beshr, “A machine learning approach to credit risk assessment,” *Towards Data Science*, 02-Nov-2020. [Online]. Available: https://towardsdatascience.com/a-machine-learning-approach-to-credit-risk-assessment-ba8eda1cd11f. [Accessed: 09-Dec-2021].