*To Default or Not to Default: An Analytical Approach to Predict Default Cases*

Group project MIS 64037/74037

John, Asher

Liu, Tammy

Lu, Tongxiang

Group participants and contribution:

Asher contributes to the report and PowerPoint;

Tammy contributes to the code;

Laura contributes to the PowerPoint.

**Introduction**

The banking industry has evolved significantly since its inception almost 4000 years ago. The last two decades have witnessed a great deal of upheavals and revolutionary changes in the banking industry. The 2008 banking crisis and three major banks collapsing in the last two months have forced banking industry to become judicious and discreet in issuing new loans. Handling loan applications is one of the most daunting jobs that banks and bankers perform. Managing and predicting default and loss cases can be quite challenging for banks. With the rise and availability of big data the challenge of handling loan application has become manageable to a large extent. Though data availability can help banks in decisions about issuing loans still there are many hurdles they should overcome before issuing a loan. One of the most pressing concerns for banks is predicting and managing default and loss cases. This requires accurate modeling and analysis of vast amounts of data. This study uses bank data to build a predictive model for default and loss cases. By analyzing and selecting from a long list of variables and cases, the study develops a predictive model that can help banks manage their risk exposure.

**The Data**

The data in this study is from a bank (we will call this bank ABC Bank) and provided to us by the professor. The data consists of two sets; first set (training data) is for the purpose of training our predictive model and the second set (test set) is for testing the model that was trained on the first set. The training data consists of 762 variables and 80,000 cases. The test set includes 25471obseravtions and 761 variables as it does not include the “Loss” column! There is no other information about the dataset as we do not know what the columns represent. Anyway, this is a big dataset and none of the group members had handled a dataset with so many variables and observation!

***Data Preparation***

The first step any data analyst must perform on dataset is to prepare the data for the final analysis. A preparatory analysis of the data was conducted, and outliers and missing values were found. The first screening was done on missing values. We found more than 40,000 observation that had missing values. As this data was a large dataset, we decided to delete all the observation with the missing values. After deleting the missing values, we were left with 762 variables and 39430 observations. The second screening process was conducted for outliers in the data. The outliers were detected using an approach which detected an outlier as 2 times the median of the vector of the maximum number from each row. We found 4000 rows that contained outliers and we decided to remove all those rows. Our decision to remove rows instead of columns is based on the justifications that as we do not know what these columns stand for some variables might have significant information about the target variable and we did not want to lose that information; so, we decided to delete rows instead of columns.

Next step in cleaning our data was to reduce the dimensionality or number of independent variables (IVs) in the data. The first technique we used was to look at the bivariate correlations between the IVs. We decided to delete the second variable in each highly (0.9 or above) correlated pair. This step helped us remove 379 IVs. After removing these variables, we decided to look at the bivariate correlation between IVs and DV. In this case we deleted the IVs that had very low correlation with the DV. Our threshold for removing an IV was if it had a correlation below 0.04 with the DV. After removing these IVs, we were left with 58 IVs.

Graphical user interface, text

Description automatically generated

Figure-1: Identifying Missing values.

**The Approach**

The first step we took after cleaning the dataset we decided to build a model on the training set. We used LASSO and PLS. LASSO gave us 35 new features that were based on the original 58 features that entered the model. The accuracy on the training measured using kappa standard was 0, which is a s good as a random guess. The second model we built was based on PLS. This model gave us 16 features that were based on the original 58 IVs. The accuracy of this model on the training set was as good as a random guess, so we decided not to use any of these models. For building our final training model we decided to test and try different models. We built Decision Tree, Random Forest, SVM, and Neural Network (NN) models. We tested all these models and decided to keep the NN model for our final analysis.

For the NN model training data provided was split into training and testing data. This was done to test the accuracy and the predictive power of the model before applying it to the “real’ test data provided by the instructor. We built and tested different NN models. In the first model three hidden layers, with 60 neurons in the first two layers, and 30 neurons in the third layer were used. The “relu” function was used for activation in the hidden layers. We used “sigmoid function” in the output layer as it was a binary classification problem of “Default” or “No Default”. The optimizer we used in the model was “adam” and the “binary-crossentropy” loss function was used. Another model with one hidden layer and 150 units; with other functions similar to the three-layer model was also built and tested. We decided to keep the model with one hidden layer and 150 units as our final model. The reason for keeping the model one hidden layer and 150 units was that it had the best precision rate. This architecture was used to train and evaluate the final model. In the training model we used 50 epochs with batch sizes of 32. Code for the model is given in Figure-2 on the next page.

Graphical user interface, text, application

Description automatically generated

Figure-2: Final Neural Network Model

The final model has an MAE of 1.345821; MSE of 7.247086 and Loss of 7.247086; Figure-3 show all these values on both the training and validation data. To check and cross validate the prediction power of the model it was applied to the test data that we had left out of the training data. On the test data the performance of the model was like the training data. Based on the test data model has an MAE of 1.345821; MSE of 7.247086 and R-Squared of 0.2916.

**Final Model**

The objective of this study was to develop a predictive model which can help the bankers at Bank ABC to decide about a new applicant whether they will default or not, and if they default how much will the bank lose. We tested different models and decided to use a Neural Network model as the model of choice for this purpose. We have chosen the NN model as it has the best ‘Specificity’ rate compared to other models (some models had 0 specificity!). Figure-4 provides the snapshot of the code for the applying the NN model on ‘test set’. We could not measure the performance of our model on the ’test set’ as it did not have the ‘Loss’ column.

Chart

Description automatically generated

Figure-3: Performance of the Model on Training and Validation

Graphical user interface, text, application, email

Description automatically generated

Figure-4: Code for the Test Data

**Conclusion**

The model built in this analysis is a model of good fit and has good predictive power. The model built through Neural Network has an accuracy of 82% in predicting the default cases. The analysis suggests that there is a strong association between ‘Loss’ and some IVs. Some of the limitations of this analysis are that analysts were not very familiar with the data as there was no metadata and we did know what different variables represented. Though the objective was to develop a good predictive model for predicting default cases yet, we cannot predict them with 100% accuracy as the quality of the data and lack of any metadata prevented us from developing a more robust model.