# 

# **ESOF 0151 - Large Scale Data Analytics**

# **Date Submitted: November 15th, 2020**

# **Project Progress Report: Stage 4**

# **Contributing Members:**

# **David Adese, Lucas Dillistone, Kevin Genereux**

**Artificial Neural Network**

The Artificial Neural Network (ANN) initially gave very good results with an accuracy score of around 93%, but on closer inspection we discovered that the high accuracy was due to our highly imbalanced data. Due to the high imbalance of fraudulent and non-fraudulent cases the ANN would always predict that there was no fraud so we had to come up with a sampling technique to help balance our data. A total of seven sampling techniques were tested with the ANN, although the Condensed Nearest Neighbor was dropped due to the time taken. A summary of the results for an ANN with a first layer with 500 nodes and a second dropoff layer are shown in the table below.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Sampling Technique** | **Overall Accuracy** | **Accuracy of predicting fraud** | **Accuracy of predicting non-fraud** | **AUC Score** |
| SMOTE (Oversampling) | 80% | 72% | 81% | 0.7604 |
| ADASYN (Oversampling) | 72% | 79% | 72% | 0.7543 |
| Edited Nearest Neighbors (Undersampling) | 93% | 37% | 95% | 0.6582 |
| One Sided Selection (Undersampling) | 97% | 15% | 100% | 0.5749 |
| Near Miss (Undersampling) | 65% | 61% | 65% | 0.6322 |
| Combination of SMOTE and Tomek (Under and Over Sampling) | 80% | 72% | 80% | 0.7609 |

Fig. 1: Summary of sampling techniques

As we can see from the above table the most accurate results come from a combination of under and over sampling techniques so we decided to make use of the combination sampler to build the final model. A full record of most of the tests performed with different sampling techniques and node amounts can be found in the Google Drive in the NN Testing file. Another issue we faced with the ANN was overfitting. When comparing the results we found that the accuracy of predicting fraud was quite high on our sampled and original training data but when testing the model that accuracy dropped. To solve this issue we attempted simplifying our ANN until we got more closely related results. Simplifying the ANN did help us with overfitting but it is still an issue and we have yet to find results that we find acceptable for a final model. At the moment this model gives us a better idea of if a transaction is fraudulent or not but it still has a ways to go and we will likely need to perform some more data transformation and hyperparameter tuning before we get to a result we are happy with. Looking at the notebooks for this competition on Kaggle there are no solutions that use an ANN so it is difficult to compare the results, but if we look back at some of the research papers we reviewed, our model is less accurate but also far less complex. This model may not be accurate enough to predict fraud on its own but if we combine it with our other models using the ensemble classifier we can produce more accurate results as a whole. The ROC curve for one of our NN models is shown below as well as some graphs showing how accuracy and loss in training and validation sets are affected by different layers and epochs.

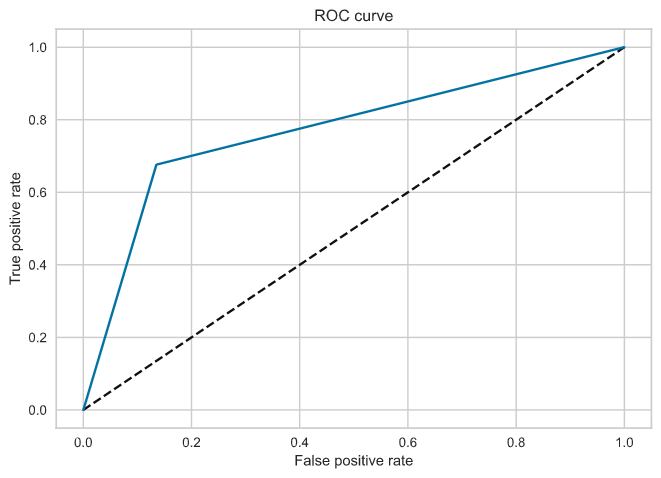


Fig. 2: ROC curve for an ANN trained over 20 epochs with an initial layer of 500 nodes and a second layer of 500 nodes

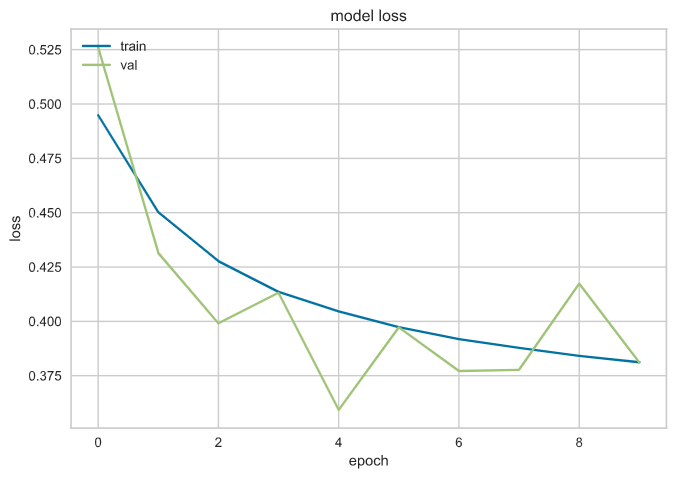
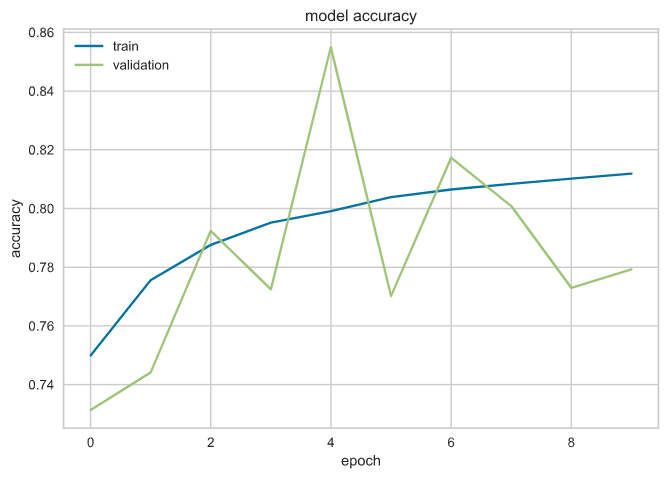


Fig. 3: Accuracy and Loss of an ANN with initial layer of 500 nodes trained over 10 epochs

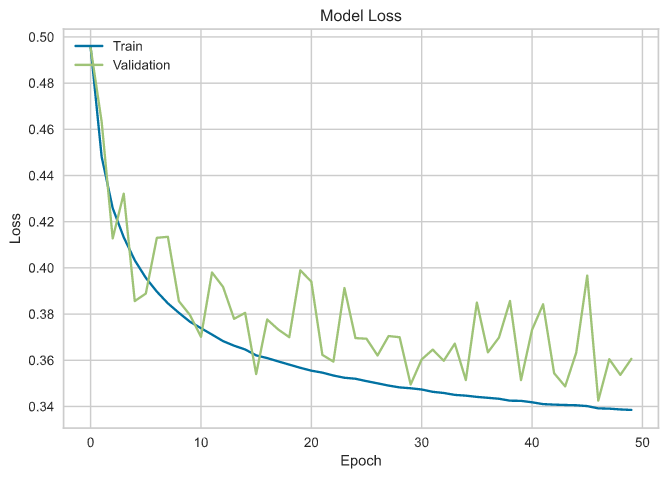
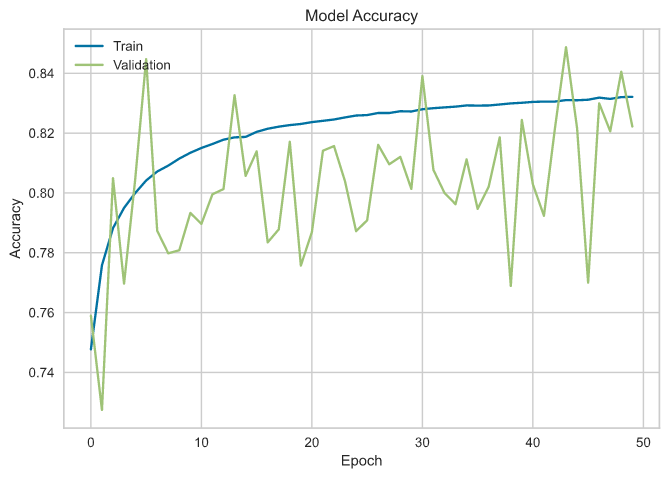


Fig. 4: Accuracy and Loss of an ANN with an initial layer of 50 nodes, a second layer of 100 nodes and a third dropout layer trained over 50 epochs

From looking at a number of models using the graphs above we found that more accurate results were given by fewer layers with amounts of nodes between 100-700. We have found some initial results from simple trial and error but we have yet to apply hyperparameter tuning so we believe that our results can be more accurate.

**XGBoost**

The XGBoost model provided somewhat accurate results from the start with testing AUC scores sitting at around 0.86. Looking at the other submissions on the Kaggle competition the highest AUC score that we have seen was around 0.96 with the application of some additional data preprocessing techniques, which we hope to integrate. Initially we had issues with unbalanced data with this model, although not quite as serious as with the ANN but through applying the same combination of over and under sampling we were able to produce more accurate results. This model can be tricky due to all of the hyperparameters that can be changed but by looking at what some other people have done along with a fair amount of trial and error and some hyperparameter tuning we can find the best parameters for the model. Although this model is already showing promising results, once we combine it with the others with the ensemble classifier it should offer us more accurate predictions. The use of the XGBoost model also allowed us to find which results were the most important similarly to what was done in the data preprocessing stage. As XGBoost uses decision trees to help train the model we decided to visualize the outcome of some of those decision trees but unfortunately unless we lowered the maximum depth of the tree the images were far too small to view.

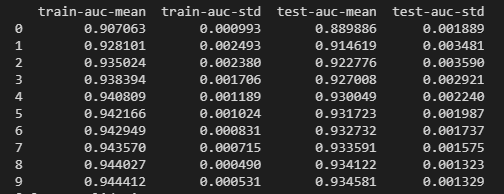


Fig. 5 effect of cross validation on XGBoost model

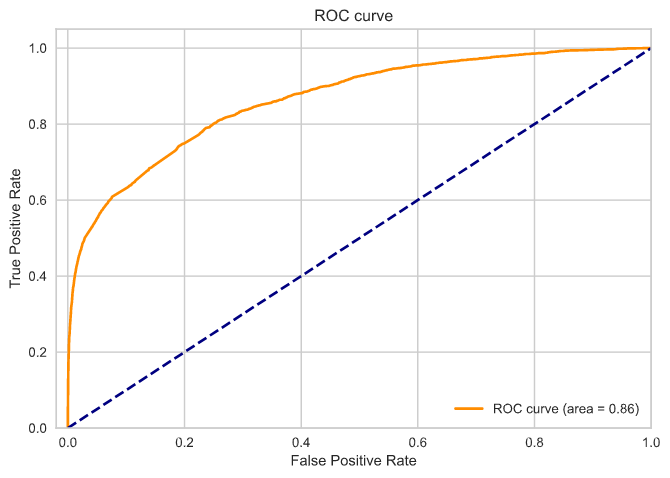


Fig. 6: ROC Curve of XGBoost model

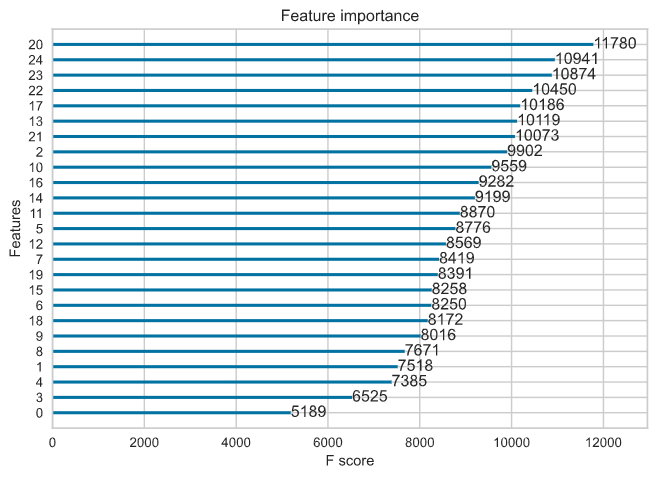


Fig. 7: Feature importance determined by XGBoost

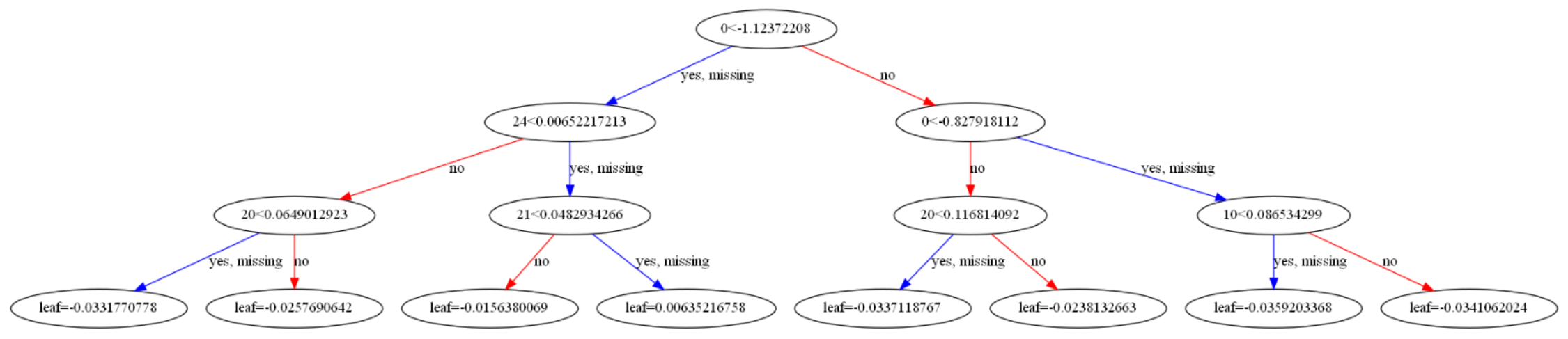


Fig.8: simplified single tree as part of XGBoost model

**AdaBoost**

For the AdaBoost model, the data is divided into 3 sets. 60 percent of the data was allocated to the training set, 20 percent to the validation set and another 20 percent to the testing set. In the training set, the model yields an accuracy of 71.5%. However, the model seems to perform better on the validation and test sets because it equally yields an accuracy of 83.5% for both sets. Other performance evaluation metrics are reported in the data below.



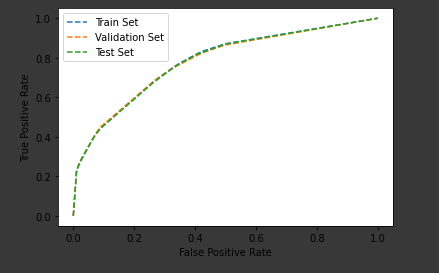




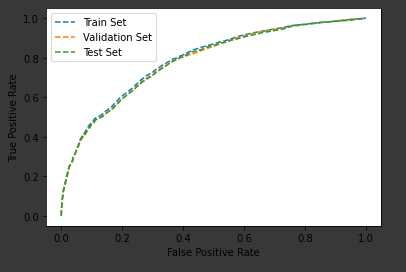
The data reduction phase happens to be the experimental area. This is where the data size is reduced but at the same time maintaining the value of the information without loss. While trying to make choices on what dimensionality reduction techniques to use. Experiments were performed on the model to gauge the effectiveness of the principal components analysis, the linear discriminant analysis as well as neither of them. The table below shows the results.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | **Principal Component Analysis (PCA)** | **Linear Discriminant Analysis (LDA)** | **None** |
| **Training Set** | **Accuracy** | 71.5% | 71.1% | 71.9% |
| **AUC Score** | 78.6% | 78.1% | 79.9% |
| **Validation Set** | **Accuracy** | 83.5% | 72.4% | 78.4% |
| **AUC Score** | 78.5% | 77.9% | 79.8% |
| **Test Set** | **Accuracy** | 83.5% | 72.3% | 78.2% |
| **AUC Score** | 78.5% | 77.9% | 79.3% |

From the data provided above, the model seems to yield a higher accuracy and AUC score when the Principal Components Analysis is used in the data reduction phase. Other areas of experimentation in progress includes feature scaling and the random sampling.



Model’s ROC Curve using LDA for Data Reduction



Model’s ROC Curve using PCA for Data Reduction

**LightGBM**

The LightGBM model received a top AUC Score of 0.7328 on the test set while it obtained an AUC score of 0.8803 on the validation set in K-Fold Cross Validation. This significant difference between the validation set performance and test set performance indicates that overfitting was involved. As shown in the table below, it was determined that SMOTE was the most effective sampling algorithm in optimizing the AUC score. In addition, applying K-Best Features as a feature selection strategy resulted in the AUC score improving from 0.7068 to 0.7382.

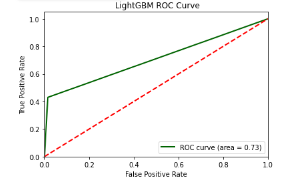


Fig. 9: Summary of the performance of various sampling techniques using the LightGBM Model. These results were obtained by using 10 iterations of KFold Cross Validation.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Sampling Technique** | **Overall Accuracy** | **Accuracy of predicting fraud** | **Accuracy of predicting non-fraud** | **AUC Score** |
| SMOTE (Oversampling) | 78% | 72% | 80% | 0.7328 |
| ADASYN (Oversampling) | 69% | 76% | 71% | 0.7543 |
| Edited Nearest Neighbors (Undersampling) | 92% | 32% | 93% | 0.6234 |
| One Sided Selection (Undersampling) | 92% | 18% | 99% | 0.5912 |
| Near Miss (Undersampling) | 65% | 61% | 65% | 0.6129 |
| Combination of SMOTE and Tomek (Under and Over Sampling) | 78% | 71% | 82% | 0.7304 |

Fig. 10: Summary of the performance of various sampling techniques using the LightGBM Model. These results were obtained by using 10 iterations of K Fold Cross Validation.