Group Coursework Submission Form

Specialist Masters Programme

Please list all names of group members: (Surname, first name) 1. Xiao, Chuqiao (220051792) 2. Wang, Ruiqi (210045131) 3. Ren, Yingying (220030041)	4. Peng, Junming (220051146) 5. Sulaiman, Muhammad (220053888) **GROUP NUMBER: 10**
MSc in: Business Analytics	
Module Code: SMM768	
Module Title: Applied Deep Learning	
Lecturer: Dr Philippe Blaettchen	**Submission Date: 8 March 2023**

Declaration:

By submitting this work, we declare that this work is entirely our own except those parts duly identified and referenced in my submission. It complies with any specified word limits and the requirements and regulations detailed in the coursework instructions and any other relevant programme and module documentation. In submitting this work we acknowledge that we have read and understood the regulations and code regarding academic misconduct, including that relating to plagiarism, as specified in the Programme Handbook. We also acknowledge that this work will be subject to a variety of checks for academic misconduct.

We acknowledge that work submitted late without a granted extension will be subject to penalties, as outlined in the Programme Handbook. Penalties will be applied for a maximum of five days lateness, after which a mark of zero will be awarded.

Marker's Comments (if not being marked on-line):

Deduction for Late Submission:

Final Mark: %

Xiao, Wang, Ren, Peng, Sulaiman (Group 10)

1. Two metrics of the model prediction and the trade-off between them (292 words）

- The hit rate is calculated by the fraction of claims marked as suspicious by the algorithm that are later found to be fraudulent. In contrast, the detection rate denotes the proportion of all fraudulent claims, inclusive of those correctly flagged and those missed by the algorithm, that are ultimately identified and prevented from being paid out. Although a high hit rate is preferable, it may lead to a diminished detection rate, and vice versa.
- Evaluating the algorithm's performance necessitates accounting for false positives and false negatives. False positives occur when lawful claims are inaccurately classified as suspicious, incurring supplementary costs for the insurer. In contrast, false negatives arise when fraudulent claims are not recognised by the algorithm, resulting in financial losses for the insurer. These errors' financial implications can be incorporated into a comprehensive cost function factoring in the hit rate and detection rate, and model parameters can be optimised to achieve the most favourable trade-off between the two.
- In conclusion, the success of a fraud detection algorithm is measured by the hit rate and detection rate, and a delicate balance between these metrics must be struck.

To evaluate a model performance, the objective is to minimize companies' loss based on the hit rate and detection rate, from a monetary perspective. The total loss is composed of the following 2 parts:

- Incorrectly identify legal claims as fraud, this will become a waste of resources since these claims should not be investigated.
    - (1 - hit rate) * (detection rate * total claims) * cost_1

- Fail to identify real fraud, this will result in paying the claim amounts to the fraudulent claims.
    - (NO. of real fraud - hit rate * (detection rate * total claims)) * cost_2

  Cost_1: The manual investigation cost per claim
  Cost_2: Average of claim amount for each claim

2. Data pre-processing (189 words)

- We proposed two ways of data pre-processing. In the first way we kept most of the features and deleted some of the unexplainable columns. In the second way, we used boxplots to compare the distribution of numerical features between fraudulent and non-fraudulent observations, then combined them

with the importance scores from the decision tree model. Then we excluded the columns that are less important to the model prediction.
In the first way we get 121 features and this pre-processed dataset is the one we use in our final model. The other way we got a dataset with 31 features, but the model performance doesn't outcompete the one trained on the previous dataset. Consequently, we choose the first dataset.

- For the feature engineering, we especially processed some special features, such as counting the days between the two columns of dates. We also did one-hot encoding for the categorical features.
- For the neural network model, we normalized the input features that are too various to be used directly. We also scaled the data so that the mean squared error between prediction and true data can be evaluated properly.

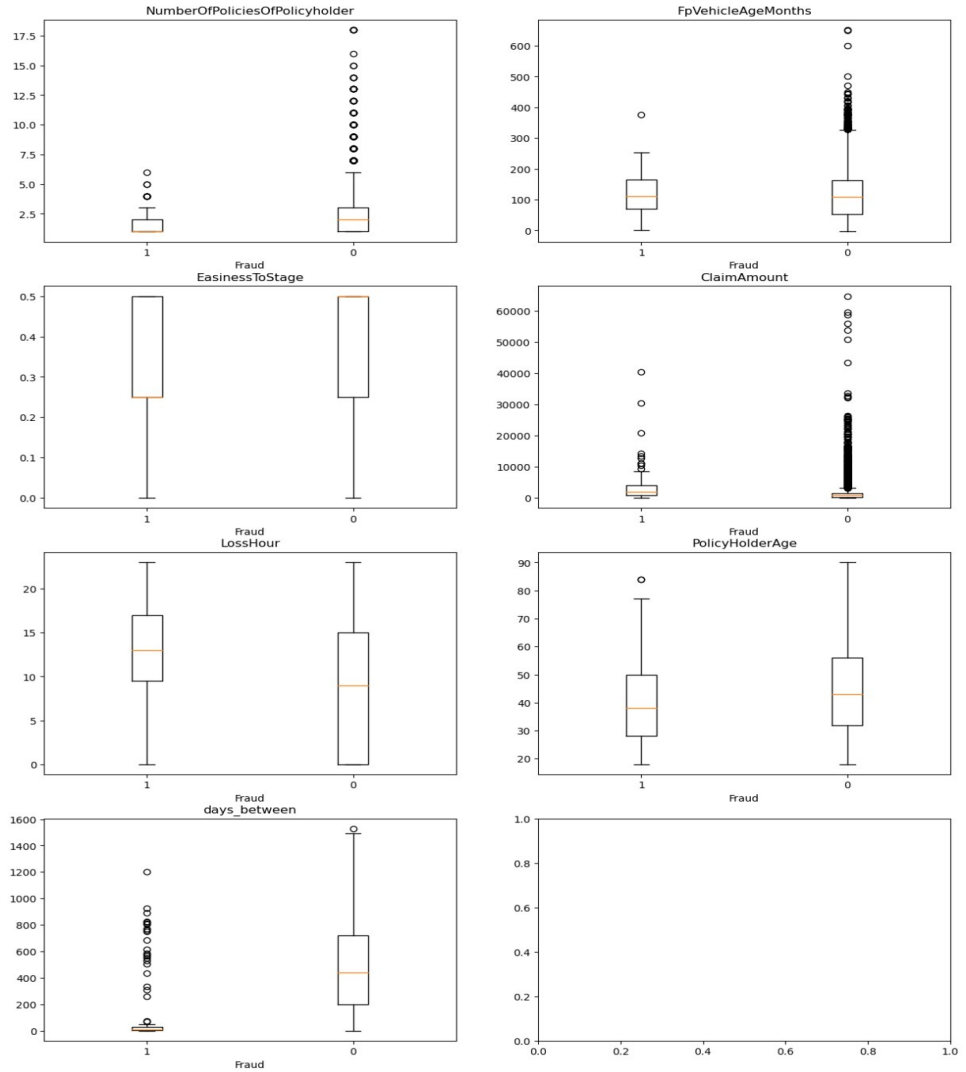
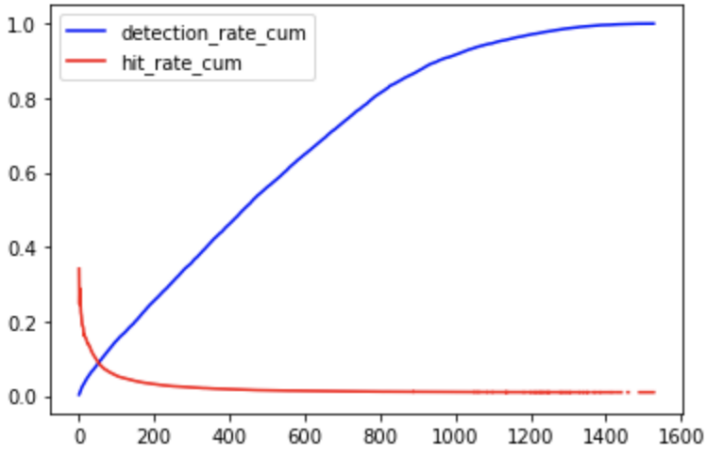


Figure 1. Boxplots of numerical features

3. Simple model (196 words)

- When we set a t value, the suspicious claims would be all the claims reported within t days, the hit rate and detection rate are cumulative rates based on

how many cumulative claims and frauds happend within t days. The cumulated detection rate would be: all claims reported within t days over total claims; the cumulative hit rate would be: all real fraud detected over all claims reported within t days. By plotting the t value against hit rate and detection rate, it shows how t affects on detection rate and hit rate.



- When t value goes up, the detection rate will increase and reach 100%, but hit rate will decrease and close to 0%. Therefore, the t value will be chosen as the interaction point between detection rate and hit rate since we want to get higher values for both rates which have opposite trends. If we choose a high detection rate, the hit rate will be low, vice versa. Therefore, the interaction point would be a balance to make both rates as high as possible, which is the optimal value.
- The detection and hit rates are 0.0888 when t value equals 51.623.

4. Decision Tree classifier (118 words)

- The Y_test and Y_pred variables contain the true labels and predicted labels using & to get only the elements that are true in both Y_test and Y_pred. Hits represents the total number of cases where the predicted label is positive and matches the true label. Frauds_detected represents the number of positive predictions that were correctly identified as frauds. Hit_rate is calculated by dividing the number of true positive predictions (hits) by the total number of positive predictions (Y_pred.sum()). Detection rate is calculated by dividing the number of correctly identified fraud cases (frauds_detected) by the total number of actual fraud cases in the test set (total_frauds).
- The hit rate is 0.095 and the detection rate is 0.012.

5. Neural network classifier (101 words)

- The neural network model is built based on a hyperparameter tuner, where we search for various optimizers, learning rate, drop-out rate and other hyperparameters' values.

- Two operations were used in order to improve the model performance on the imbalanced dataset: adding weights, increasing batch size and epoch. By implementing the above methods, the model's ability to classify fraudulent observations improves, because in each batch of a single epoch, the model is able to take more fraudulent observations into training and weigh heavier on them.
- The hit rate is 0.154 and detection rate is 0.015 for the neural network model.

6. Autoencoder classifier (250 words)

- In the data preprocessing for autoencoder, the fraudulent and non-fraudulent observations are separated first, followed by assigning fraudulent observations evenly to validation and test datasets. Then all the datasets are scaled to the range from 0 to 1.
- Two different approaches are tried in model design. Firstly, we used a traditional way of building the model, with a tuner to tune the autoencoder with ordinary dense layers with “relu” as activation functions and “mse” as the loss function.

  The other trial is to reshape the 121 features into a 11*11 matrix, and fit the data into an autoencoder with 2-dimensional convolutional layers. The noise is added to the input to train a denoise autoencoder.
- To evaluate the model performance, the loss plots on validation datasets and training datasets are shown. Then, to set a proper threshold for the classifier, the distributions of fraudulent and non-fraudulent datasets’ mean squared error (mse) are visualised into two colors. Nevertheless, no matter how we tune the classifiers the mean values of the two distributions don’t separate well and the loss plots also show similar trends. These visualisations all indicate that the model can’t clearly classify the two types of observations.
- Finally, a threshold is set based on the prediction based on the training dataset. The threshold is calculated by the mean squared error added by one standard deviation, the hit rate and detection rate are then calculated.
- The hit rate equals 0.017 and detection rate is 0.840 for the neural network model.

7. Discussion of models (299 words)

- For the simple model, we only used t value as the parameter to detect fraud which would be biased and inaccurate since it is possible that people encounter an incident and claim loss in a few days after subscribing to insurance. But this model has high transparency as it only uses one simple parameter which is straightforward and can be clearly explained.
- For the decision tree model, we used all 121 parameters to detect fraud.It has a high accuracy (0.984) at predicting fraud but it does not represent the goodness of the model since the numbers of fraud only takes about 1% of total claims which means we can get 99% accuracy even if we predict all

claims as non fraud. The transparency for the decision tree model is high as it can clearly visualize how it classifies fraudulent claims and non fraudulent claims based on parameters and logical rules on the tree plot which is explainable. The hit rate equals 0.095 and detection rate 0.012.

- Compared to the decision tree model, the neural network model has a higher hit rate (0.154) and a higher detection rate (0.015) which performed better than the decision tree model.
- For the autoencoder model, the hit rate equals 0.017 and the detection rate equals 0.840 which does not perform well.
- Both the neural network model and the autoencoder model have low transparency since they both use the machine itself to build the hidden layers and find the pattern or representative regarding the fraud classification which makes them non interpretable.
- It is problematic if we cannot provide a clear explanation to our audience (e.g., customers), the model would not be reliable since your audience cannot understand how it works so companies may lose customers' trust.

8. Possible approaches to improve model performance (135 words)

- There are 3 effective approaches to deal with an imbalanced dataset. Other than the two approaches (adding the weights, changing the batch size and epoch) that have already been applied in our neural network model, oversampling is also an approach worth trying in solving this problem.
- Oversampling involves either undersampling the majority class, oversampling the minority class, or a combination of both. Undersampling reduces the size of the majority class by randomly removing samples, while oversampling increases the size of the minority class by either replicating existing samples or generating synthetic samples.
- By oversampling, the neural network model will include more fraudulent observations in the batches each time the model calculates the gradient. This helps the classifier to learn fraudulent observations' features better. As a consequence, the classifier would perform better on the imbalanced dataset.