Introduction to Machine Learning (SS 2024)
Programming Project

Author 1
Last name: Plunser
First name: Fabio
Matrikel Nr.:

Author 2
Last name: Barbist
First name: Dominik
Matrikel Nr.:

I. Introduction

We selected the Transaction dataset which includes 22000 Data points each having 30 features(1 Time + 29 other features). The Data were labeled with a 1 if the transaction was fraudulent and 0 if it was not. The dataset is highly imbalanced with only 0.xxx% of the data points being labeled as fraudulent. The nature of the task is a classification problem either a transaction is fraudulent or not. The 29 other features are anonymized, and we do not know what they represent. The dataset has no missing values and no data imbalances.

- What is the nature of your task (regression/classification)? Is it about classifying types of birds, or deciding the number of cookies an employee receives?
- Describe the dataset (number of features, number of instances, types of features, missing data, data imbalances, or any other relevant information).

II. Implementation / ML Process

- Did you need to pre-process the dataset (e.g. augmenting data points, extracting features, reducing the dimensionality, etc.)? If so, describe how you did this.
- Specify the method (e.g. linear regression, or neural network, etc.). You do not have to describe the algorithm in detail, but rather the algorithm family and the properties of the algorithm within that family, e.g. which distance functions for a decision tree, what architecture (layers and activations) for a neural network, etc.
- State (in 2-5 lines) what makes the algorithm you chose suitable for this problem. What are the reasons for choosing your ML method over others?
- If you used a method that was not covered in the VO, describe how it is different from the closest method described in the VO.
- How did you choose hyperparameters (other design choices) and what are the values of the hyperparameters you chose for your final model? How did you make sure that the choice of hyperparameters works well?

A. Preprocessing

Due to the imbalanced nature of the dataset, we used an oversampling technique to balance the dataset. Meaning we duplicated the fraudulent transactions to a certain percentage of the training data. Around 30%-40% showed the best results. We tested two different methods, logistic regression, and a neural network because the data were labeled, and the task was a classification problem.

1) Logistic Regression:

2) Neural Network: We used a neural network with 3 hidden layers and 1 output layer. The input layer has 30 neurons, and all hidden layers have 124 neurons. We used the ReLU activation function for all hidden layers and the sigmoid activation function for the output layer. We used the Adam optimizer and binary cross-entropy as the loss function. Furthermore, we trained the model for 40 epochs with a batch size of 32. Because we implemented an early stopping mechanism, the model stopped at most after 18 epochs. We used a dropout layer with a dropout rate of 0.4 to prevent overfitting. For the learning rate, we started with 0.001 and decreased it by a factor of 0.1 if the validation loss did not decrease for 5 epochs. For the early stopping mechanism, we used a patience value of 10 epochs. We used the F1 score as the evaluation metric because the dataset is imbalanced.

3) Choice: We chose the neural network over logistic regression because the neural network showed better results. The neural network was able to learn the underlying patterns in the data better than logistic regression. See the Results' section for more details.

III. Results

- Describe the performance of your model (in terms of the metrics for your dataset) on the training and validation sets with the help of plots or/and tables.
- You must provide at least two separate visualizations (plot or tables) of different things, i.e. don't use a table and a bar plot of the same metrics. At least three visualizations are required for the 3 person team.

IV. Discussion

- Analyze the results presented in the report (comment on what contributed to the good or bad results). If your method does not work well, try to analyze why this is the case.
- Describe very briefly what you tried but did not keep for your final implementation (e.g. things you tried but that did not work, discarded ideas, etc.).
- How could you try to improve your results? What else would you want to try?

V. Conclusion

- Finally, describe the test-set performance you achieved. Do not optimize your method based on the test set performance!
- Write a 5-10 line paragraph describing the main take-away of your project.