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 approximately 227,000 data points, each having 30 features (1 Time feature, 1 Amount feature, and 28 anonymized features). The dataset is labeled such that a transaction is marked with a 1 if it is fraudulent and 0 if it is not, making the task a binary classification problem.

The dataset is highly imbalanced, with only about 0.001729% of the data points labeled as fraudulent. This extreme imbalance poses a significant challenge for the classification task, as the model may become biased towards predicting the majority class (non-fraudulent transactions).

The 28 other features are anonymized, and their exact meanings are unknown. However, these features do not contain any missing values, ensuring that each feature is ready for use in machine learning algorithms without additional preprocessing for imputation or scaling.

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 employed an oversampling technique to balance the classes. We used a custom oversampling function that generates synthetic samples for the minority class (fraudulent transactions) based on the k-nearest neighbors algorithm. This approach involves generating new fraudulent samples by interpolating between existing fraudulent samples and their nearest neighbors.

B. Logistic Regression

Logistic regression is one of the most straightforward classification algorithms, making it a suitable initial approach for binary classification tasks. This model estimates the probability that a given input belongs to a particular class using a logistic function.

1) Hyperparameters: We use completely standard hyperparameters without any fine tuning. We tried some different solvers and maximumt iterations but the outcome didn't change much. ¿ The hyperparameters we used are:

- Solver: lbfgs
- Max iterations: 100

The biggest impact on the performance was the oversampling.

2) Model Description: Logistic regression operates by applying the logistic function to a linear combination of input features, yielding a probability score between 0 and 1. This probability is then used to classify the input into one of the two classes. Given its simplicity and ease of implementation, logistic regression serves as an effective starting point for classification problems.

C. 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 64 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 64. Because we implemented an early stopping mechanism, the model stopped at most after 24 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 roc-auc score as the evaluation metric for early stopping.

D. 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.