Detecting Fraud in Credit Card Transactions

Fatimah Alshaikh, Moustafa Makhlouf, Muhammad Haider Asif

Objective

Credit card use has been increasing throughout the years, thus increased credit card fraud. Especially after covid, fraud claims have increased a lot. Therefore, we decided to analyze a dataset of credit card transactions to learn patterns in credit. We used a dataset collected from European cardholders in September 2013 over 2 days. The goal of this project is to develop a Machine Learning model that is able to detect fraud in credit card transactions.

Data

The data is taken from a kaggle notebook, https://www.kaggle.com/mlg-ulb/creditcardfraud?select=creditcard.csv, with a usability rating of 8.5/10, and ~7500 upvotes. The data contains transactions made by credit cards in September 2013 by european cardholders. This dataset presents transactions that occurred in two days, where we have 492 frauds out of 284,807 transactions. The dataset is highly unbalanced, the positive class (frauds) account for 0.172% of all transactions. Each example has 28 anonymous features, “Delta-T”, “Amount”, and “Class”. We inserted the data into a SQL database that we created, adding an extra column id which is an auto incremented integer and is used for primary key purposes. We also made sure that there are no null fields or corrupt data points.

Models and Procedure

In order to analyze this dataset to find the best fraud detection accuracy, we tried a few models on this dataset. Our first thought looking at the dataset was to use a binary classification model, due to its binary nature. Therefore, we tried using the logistic regression model, and we achieved considerable results. However, we wanted to examine how an unsupervised learning model would train on the data and what possible insights could it provide. After that, we decided to experiment with Gaussian Naive Bayes and CNN network models.

- Supervised Learning
    - *Logistic Regression Model*
      This model was the most intuitive for us since it is a binary classification model for a binary problem. On our initial training, we achieved very high accuracy, and upon further inspection we realized that the model was classifying everything as non-fraud since the percentage of fraud transactions in the dataset is only 0.172%. Thus, the overall accuracy of the model would be above 99% if it classified everything as non-fraud. Thus we split the analysis of the accuracy based on classification, thus allowing us to analyze more specifically how the model learns to classify fraud transactions. After a couple of revisions of the model with different hyperparameters, we were able to get an accuracy of 99.91% overall (63.14% accuracy for fraudulent and 99.96% accuracy for non-fraudulent).

- *Gaussian Naive Bayes*

  The next thing we wanted to check whether we could do better by assuming that the features of the dataset, V1-V28 along with the relative time, “Delta_T” and the Amount were all independent of each other. Our model, Gaussian Naive Bayes, relied on this as the base assumption along with assuming that the likelihood of the features is also Gaussian. This reduced the complexity of our model compared to the regularized logistic regression model, but gave us a similar accuracy to the accuracy we got from the most optimal Logistic regression model. But on further optimization, we removed the one field that we knew was relative for each of the transactions to see if we can better fit the independent feature assumption. As it turned out, that was the case, and our accuracy on the fraudulent transactions jumped from 63.5% to 82.05%, which was our highest accuracy. On the other hand, this model didn’t do as well on the non-fraudulent transactions with an accuracy of 97.79%, but it has a respectable overall accuracy 97.77%, and the reduction in the overall accuracy was acceptable considering the high accuracy on the fraudulent transactions.

- Unsupervised Learning
  - KMeans

    Even though this is a classification problem with a labeled dataset, we wanted to experiment with unsupervised learning and possibly gain any insights the model would provide. However, our results were very inconclusive, and we were not able to extract any meaningful patterns. We reckon that is due to the fact that the number of fraud examples is very small thus the model (not given any labels as to whether an example is fraud or not) was not able to find an underlying pattern just based on the data. We thought that there might be some implied distinctive features of fraud transactions, but as our results show, there appears to be none.

- Deep learning
  - Convolutional neural network and a multi-layered perceptron

    Finally, we wanted to see if increasing the complexity of the model by using a multi layered perceptron or a convolutional neural network would improve our accuracy for the fraudulent transactions. The reason for this is that a MLP is able to capture non-linear and very complex relationships that might exist between the features, and a convolutional neural network can capture local as well as global patterns and therefore make better predictions. Our results showed us that with the skew of the dataset, even after optimizing the architecture and the hyper parameters both the MLP and the CNN were very unpredictable and gave varying accuracies on the fraudulent transactions. But on average the CNN gave us an accuracy of 55% on the fraudulent transactions, which was not better than our simpler models so for our final model we went for the least complex and the best performing Gaussian Naive Bayes model.

Results and Analysis

Discriminative Supervised Approach*:* Can we accurately classify transactions as fraudulent/non-fraudulent using a regularized logistic regression model?

> Given the binary nature of the classification problem, a logistic regression model performs well on this dataset with an overall accuracy of 99.91% (63.5% accuracy for fraudulent transactions). We tested the model with both Delta_T and without it, and we found that the accuracies are mostly the same.

Generative Supervised Approach: Can we accurately classify fraudulent/non-fraudulent transactions using a Naive Bayesian model?

> The model achieved the highest classification accuracy for fraud transactions of 82.05%. However, that comes at the expense of classifying non-fraud transactions, where it scored a lower accuracy than the Logistic regression model at 97.79%. The overall accuracy for this model was 97.77%. This accuracy was achieved by training the model without the Delta_T field. The model had very similar accuracies to the logistic regression model if we include the Delta_T field.

Unsupervised Approach: Can we cluster fraudulent/non-fraudulent transactions in their respective clusters using K-means clustering algorithm?

> We had very little success with KMeans. Using KMeans clusters did not provide any significant analysis of the dataset, due to the extreme imbalance of the dataset. Moreover, for a labeled dataset, unsupervised learning is not ideal. It classified the data into two large clusters that we were not able to interpret in scope of our classification problem.

Deep-Learning Approach: Can we use convolutional neural networks to accurately classify transactions as fraudulent/non-fraudulent?

> CNN's although theoretically could do well did not achieve high accuracies on the fraudulent transactions as expected even after hyper parameter and architecture tuning.

Optimization: In the most accurate model, can we drop some non-relevant features to improve accuracy?

> Due to the anonymity of the fields, we had less flexibility with dropping features. Since we know 'Delta-T' of one feature is dependent on another, we decided to drop it, which improved the accuracy of our NB model, and changed the learning behavior of the logistic model. Without Delta_T, the model achieved its highest accuracy faster and with fewer epochs. However, its highest accuracy was a bit smaller than that with Delta_T, with a difference less than 0.2%. Dropping Delta_T had the biggest noticeable effect on the NB model, where it increased the fraud accuracy by about 20%.