Credit Card Fraud Detection Model Comparison Report

Brian Faraira

5/10/2022

This program is a credit card fraud detection algorithm. It takes in transactions from credit cards in September 2013 and it learns from the training data in order to make accurate predictions. This program was made using the random forest model, logistic regression model, and the SVM model. This program was made to compare the recall scores of these algorithms and see which algorithm works the best for credit card fraud. The dataset that I used has 31 features, and features V1 to V28 are all features that have gone through PCA transformation and are withheld due to confidentiality concerns. The other features are the amount of the transaction, the time since the first transaction, and the class of the transaction. The class feature will be our label and it will contain either 0 or 1 where o is for normal and 1 is for fraud. There is a total of 284,315 normal transactions and only 492 fraudulent transactions. This is extremely unbalanced and this can lead to many issues with some models like the SVM model. In my project proposal, I was told that since there was a large imbalance in the label data that there would be a need for weighted loss functions and for a different evaluation metric from classification accuracy since it would lead to an abnormally high accuracy which was incorrect. I decided to use the precision, recall, and F1 score in order to find an accurate evaluation of the data. The precision score accounts for the false positives while the recall score accounts for the false negatives. The combination of precision and recall is the F1 score. Since we want to account for the false negatives the most it's best to use the recall score as the main evaluation metric. I also made a heatmap to map the correlations of the 31 features in the dataset to each other. This helped me see which features were irrelevant and which features were redundant. I noticed that there were

no redundant features from V1 to V28 since there was little correlation between them. The only feature that had some correlation was the amount feature with V20 and V7. This shows that these features may be redundant, but there was only a slight correlation between them. There was little correlation between the label, class, and the other features which mean that there were no irrelevant features. The model that received the highest recall score, precision score, and F1 score was the Random Forest model. The model that had the second-highest scores was the SVM model while the Logistic Regression model had the lowest scores. The algorithm should minimize the number of false negatives over the number of false positives. In real life, it is better if the algorithm predicts a transaction is fraudulent than the algorithm predicting a transaction isn't fraudulent when it is fraud. This means we would want to pick an algorithm that minimizes this which is still the random forest model according to the confusion matrix. The random forest model performed the best since it was a model that had both low bias and low variance. The low bias is due to it being a decision tree which normally has a low bias due to the complexity. The random forest model uses bootstrapping in order to compensate for the high variance which leads to the model neither underfitting nor overfitting normally. Due to the low variance and bias, the recall score of this algorithm is higher than the others. At first, when I started coding my program, I noticed an abnormally low recall score of around 50% for SVM. The SVM model then proceeded to perform better than the logistic regression model. This is due to the standard scaling I used on the data, this would have little effect on the recall score of logistic regression, but it brought the score of SVM up by a large amount. SVM is less prone to overfitting compared to logistic regression which could be the reason why logistic regression performed less accurately when compared to SVM. I then standardized my data by using the standard scaling method from sklearn which helped increase the recall score. I was motivated to do this project

idea since I had always wondered how they decide whether a transaction was fraud or real. I had received texts that restricted my credit card after I made some transactions once and I had wondered why decided that. Now I learned that the process of making a credit card fraud detection algorithm is fairly difficult and that the accuracy of these machine learning algorithms isn't always 100%. I think this problem is extremely important in daily life since it can be a measure used to help people who have had their credit card information stolen. This machine-learning algorithm can help save people from losing thousands of dollars if it is able to accurately predict if a transaction is fraudulent.