A Supervised Machine Learning Approach to Credit Card Fraud Detection

Adedamola Adesoye
College of Engineering
Northeastern University
Toronto, ON
adesoye.a@northeastern.edu

Abstract

In this report, I explored in detail, the performance of two supervised learning algorithms on a credit card fraud dataset. The two machine learning algorithms are Decision Tree (DT) and K-Nearest Neighbors (KNN). I show the unsuitability of traditional accuracy metrics for evaluating the models. I then compare the performance of both models and discuss some caveats about using this dataset in building a credit card fraud detection system.

1 Datasets

The dataset is made up of 48hours worth of transaction data made with credit cards in September 2013 by European cardholders. The dataset will be used for classification; to classify if a transaction is fraudulent or genuine.

Table 1: The basic feature of the dataset.

	Data Set Characteristics	Attribute Characteristics	Associated Task	Number of Instances	Number of Attributes
Dataset	Multivariate	Real	Classification	284,807	30

1.1 Data characteristics

Each data point represents a transaction.

28 of the attributes have been transformed using Principal Component Analysis (PCA) to maintain client confidentiality. Although PCA is commonly used to reduce the dimension of a dataset, in this case, we cannot confidently say it was also used to reduce the dimension of the original data. It is possible that the 28 Principal Components account for all the variation produced by 28 features from the original dataset and PCA was only carried out for confidentiality reasons.

The other two attributes are *Time* and *Amount*; *Amount* describes how much money was spent in a transaction while *Time* denotes the number of seconds elapsed between a transaction and the first transaction in the dataset.

All the attributes have float datatypes. The target column, *Class*, which differentiates a fraudulent (1) transaction from a genuine (0) one, is of integer data type.

The bar chart of *Class* is shown in Figure 1. Class imbalance is very evident in the dataset.

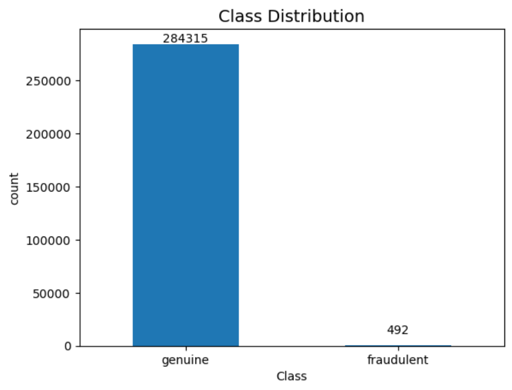

Figure 1: The *Class* distribution

1.2 Why is this an interesting dataset?

The positive class in this dataset accounts for 0.17% of the dataset. The high degree of class imbalance makes it an interesting problem. Techniques used in analyzing this data can be applied to several other classification problems.

1.3 Pre-processing

At an initial glance, the dataset has no null values. Notably, the *Amount* column has 1,825 transactions equal to "0.00". This could mean anything; it could mean a purchase with 100% discount or adding one's credit card to a subscription platform like Netflix (Sanzil et.al, 2019). Since the dataset contains some *Amount*s equal to 0.01, the value 0.00 could also represent transactions that are too small to be rounded up to 0.01.

Since the transactions are not interdependent, I dropped the 1,825 (with 27 fraudulent) transactions. This reduced the proportion of the fraudulent transactions in the new dataset to about 0.16% and total size of the data set to 282,982.

I split the data to have 75% and 25% for the training and test set respectively. The two splits mimic the proportion of the classes in the whole dataset (i.e.: 0.16% fraudulent transactions). After the split, I scaled the *Time* and *Amount* variables to ensure a standard normal distribution. I fit the scaler to the training set and used the same scale to transform both the training and test set. I chose to carry out the scaling in this manner because, in reality, the test set will serve as new, unseen data and won't contribute in the training phase.

2 Model Building and Evaluation

For brevity's sake, the only models discussed in this section are DT and KNN. Due to time constraints, no parameter tuning was carried out to build the models. The hyperparameter values were randomly selected and, fortunately, yielded decent results.

2.1 Decision Trees with Pre-pruning

Initially, I built a decision tree with a maximum depth of 4 to illustrate the unsuitability of the traditional accuracy score to evaluate its performance. It yielded an 100% accuracy score on the training set and 99.9% on the test set. Looking at the confusion matrix paints a different picture.

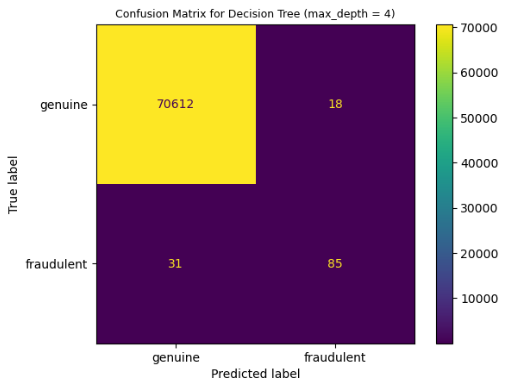

Figure 2: Confusion Matrix of Decision Tree

31 of the truly fraudulent transactions were misclassified as genuine and 18 transactions from the negative class were wrongly classified as fraudulent.

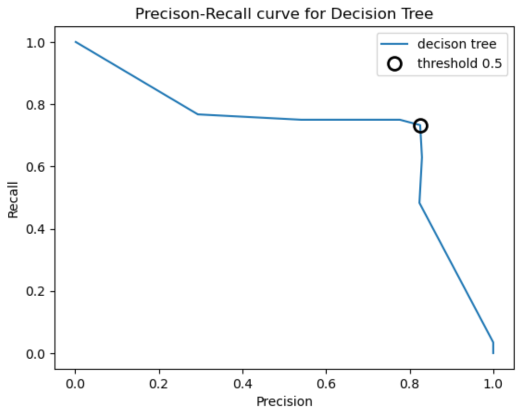

Figure 3: Precision-Recall Curve for Decision Tree

The model yielded a recall of 0.73 and a precision of 0.83 with an f1-score of 0.78. Figure 3 shows that the default probability threshold (0.5) yields the best result since it records the highest precision and recall simultaneously.

2.2 K-Nearest Neighbor (KNN)

The KNN classifier was instantiated with three neighbors. It yielded a precision, recall and f1-score of 0.93, 0.72 and 0.81 respectively.

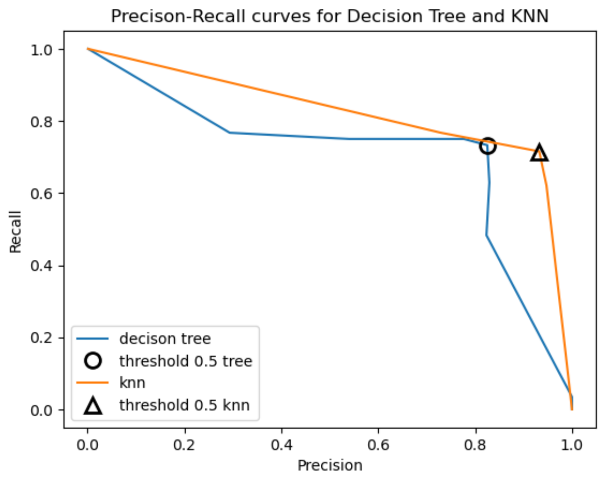

Figure 4: Precision-Recall Curves for Decision Tree and KNN

As shown in Figure 4, as the prediction probability threshold increases to the default (0.5), KNN produces a much slower decrease in recall than the Decision Tree. At 0.5 threshold, KNN yields a significantly higher precision than the Decision Tree with only a 0.01 less recall. Overall, KNN performs better than the Decision Tree built.

3 Conclusion

In this report, I have analyzed the performance of two algorithms on a credit card fraud dataset. With the decision tree, I have shown that conventional metrics, like accuracy score, are unsuitable for evaluating the models due to the high proportion of class imbalance. The confusion matrix gives a better picture of the models' performances.

Depending on the goal of the merchant, the model's hyperparameter(s) can be tuned to give a better recall or a better precision with a trade-off between the two. A higher recall will increase the number of transactions flagged as fraud thereby increasing the number of false positives and inevitably, reducing the precision. If the threat response is to block a transaction, this could lead to customer dissatisfaction since their genuine transaction attempts are denied. On the other hand, a higher precision will reduce (or keep constant) the number of fraud alerts. This will reduce the number of true positives and cost the banks and/or merchants some money since, as is by law, they are responsible for paying for the purchases made with the compromised credit card. (Hardekopf, 2022)

4 Further Scope

For a better confidence in model performance, cross-validation technique could be used to evaluate the performance of the model. Other metrics like area under the receiver operating characteristics curve can also be explored. Sampling the can also be explored. Automatic feature selection can also be carried out to improve the model's performance. Manually, the *Time* feature can be deselected before building the model since it is unlikely that the time that, for example, transaction five took place, relative to transaction one, will have any impact on its "genuineness" especially if they are by different cards in different locations.

With all that said, 48 hours' worth of data is not reliable for building an effective credit card transaction data. Also, a single transaction's information is not sufficient to label it fraudulent or genuine; it should be put into context (A. Dal Pozzolo et al., 2014). A better approach will be to consider each cardholder separately. A cardholder's spending habits may give better indications of fraud. The location where their card was used could also help to make better decisions on when to raise an alarm. In addition, the training data needs to be replaced, over time, with new data as fraudsters evolve their methods.

Acknowledgments

The learning code is adapted from: this Kaggle Notebook and references therein.

References

[1] Sanzil (2019). why there are transactions with amount 0?. https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud/discussion/86665?search=zero

[2] Hardekopf, B. (2022, November 8). Who Pays For Fraudulent Credit Card Transactions? Forbes. https://www.forbes.com/sites/billhardekopf/2022/11/08/who-pays-for-fraudulent-credit-card-transactions/?sh=5c52d9222218

[3] Dal Pozzolo, A., Caelen, O. D., Le Borgne, Y., Waterschoot, S. & Bontempi, G. (2014). Learned lessons in credit card fraud detection from a practitioner perspective. Elsevier. http://dx.doi.org/10.1016/j.eswa.2014.02.026