Micah Carroll, Donald Ghazi, Christine Straub

Spring 2017

1. Problem statement and goal of Analysis

Credit card fraud is a wide-ranging term for theft and fraud committed using or involving a payment card, such as a credit card or debit card, as a fraudulent source of funds in a transaction. The purpose may be to obtain goods without paying, or to obtain unauthorized funds from an account (from Wikipedia).

The dataset 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. The goal of the analysis is to extract meaningful insights from the data and create machine learning models which can classify a given transaction as fraudulent/non-fraudulent.

2. Data pre-processing

The data provided on Kaggle is pretty clean and does not need much manipulation. Once imported from the csv provided on the Kaggle page, the data is ready to be split into training, validation and testing sets.

3. Data Modelling

The three models chosen by me for the purpose of this analysis are:

1. Logistic Regression
2. Random Forests (Decision Trees)
3. Feed-forward Neural Network

All the hyperparameters have been heuristically chosen. The training and test sets have been split in the 6:4 ratio for all the three models. Cross-validation has been performed only for feed-forward neural networks and not the other two models. 3-fold cross validation has been performed for the feed-forward neural networks. Also, the model can be easily varied to perform a k-fold cross validation for any value of k.

4. Comparison of models

I have used the precision, recall and F-1 scores for the purpose of evaluation and comparison of the three models. This is done keeping in mind the skewness of the data, as there are very few points present where the transactions are fraud. Looking just at accuracy figure won’t exactly be the right choice. To understand what precision, recall and F-1 scores are exactly, consider the following:

1. **Positive (P)**: Observation is positive (for example: is a fraudulent transaction)
2. **Negative (N)**: Observation is not positive (for example: is not a fraudulent transaction).
3. **True Positive (TP)**: Observation is positive, and is predicted to be positive.
4. **False Negative (FN)**: Observation is positive, but is predicted negative.
5. **True Negative (TN)**: Observation is negative, and is predicted to be negative.
6. **False Positive (FP)**: Observation is negative, but is predicted positive.
7. **Error**: Proportion of all predictions that are incorrect. Error is a measurement of how bad a model is.

error = (F P + F N) / (F P + F N + T P + T N) = (incorrect predictions) / (all predictions)

8. **Accuracy**: Proportion of all predictions that are correct. Accuracy is a measurment of how good a model is.

accuracy = (T P + F N) / (T P + F N + T P + T N) = (correct predictions) / (all predictions)

9. **Precision**: Proportion of all positive predictions that are correct. Precision is a measure of how many positive predictions were actual positive observations.

precision = (T P) / (T P + F P) = (positive predicted correctly) / (all positive predictions)

10. **Recall**: Proportion of all real positive observations that are correct. Precision is a measure of how many actual positive observations were predicted correctly.

recall = (T P) / (T P + F N) = (predicted to be positive) / (all positive observations)

11. **F1 Score**: The harmonic mean of precision and recall. F1 score is an ’average’ of both precision and recall. We use the harmonic mean because it is the appropriate way to average ratios (while arthmetric mean is appropriate when it conceptually makes sense to add things up).

F1 = 2 (precision * recall) / (precision + recall)

The results of the three models in terms of precision, recall and F1 score values is reported below in the table.

Model	Precision	Recall	F1-Score
Logistic Regression	0.86	0.63	0.727
Random Forest	1.0	0.95	0.974
Neural Network	1.0	0.99	0.99

Comparing the F1 scores we can clearly find that the neural network model performed the best with a F1 score of 0.99. Note, the accuracies in case of both Random Forests and feed-forward neural networks was well over 99.5 % and hence comparison just on the basis of accuracy wasn’t a great choice.

5. Discussion of results

The major reason for the failure of Random Forests was perhaps overtuning to the dataset, since there were very little positive examples. This was taken care of in the neural networks though and hence, it performed better. Also, we see that the accuracy of the models increased with the model complexity. Logistic regression and random forests, unlike feed-forward neural networks were unable to give more weightage to the positives (faudulent transactions) and hence performed poorly.

The neural network approach was by far the best, with 99% F1-Score, 99% sensitivity (successfully identifies TP 99% of the time), and 14% precision (only 14% of the transactions identified as frauds were actually frauds). The precision percentage can clearly be improved, but in credit fraud detection maybe sensitivity is the most important factor, as real problems arise when frauds are not detected. Also, having only 1 in 7 detected frauds being actual frauds is not necessarily too limiting. This data can still be used, even though it is not great.

6. Conclusion

If we see the time requied for the models to train, neural networks take really long time to train compared to the other two algorithms. And hence application of neural networks in the industry for fraud detection is only possible with great computational capabilities. We created a decent algorithm to detect frauds in credit card/online transactions, but to be able to use this on even larger corpuses of data perhaps it is necessary to use more sophisitcated algorithms (to improve precision) and greater computational capabilities