Machine Learning Engineer Nanodegree

Capstone Project

Nitin Mahajan
June 24th , 2018

I. Definition

Project Overview

Credit Card fraud losses are a huge problem. The Nilson Report, a publication covering global payment systems, reported recently that global card fraud losses equaled $22.8 billion in 2016, an increase of 4.4 percent over 2015. That amount does not include costs incurred by retailers, card issuers, and acquirers for their operations and chargeback management.

By 2021, card fraud worldwide is expected to reach a total of $32.96 billion. It is important that card companies can identify fraudulent card transactions so that customers are not charged for items that they did not purchase. Better tactics and technology to combat criminals attempting such transactions will define fraud fighting for merchants in coming years.

Fraud detection is one of the most important applications for machine learning in finance. Institutions are now using machine learning to help detect frauds. Supervised learning is an area of machine learning where based on historical data, which is labelled the machine learns the pattern in the data. Based on the learning, the machine is then able to make predictions for the labels of unseen & unlabelled data.

This project will detail the process of building a supervised learning machine learning model to classify fraudulent and non-fraudulent transactions. The data has for the project has anonymized credit card transactions labelled as fraudulent or genuine. The project data has been obtained from Kaggle.

Problem Statement

The goal of this project is to classify the transactions into fraudulent or genuine transactions. Our main focus would be to identify the fraudulent transactions as these are the transactions we would want to separate from the genuine transactions. Below steps would be involved to solve this binary classification problem-

1. Download the data from Kaggle
2. Data exploration would be done to understand the data better and do any further processing as required
3. Train multiple classifiers i.e. –
    a. Logistic Regression
    b. Adaboost

    c. Ridge Classifier
    d. Light Gradient Boost
4. Select the best model by comparing the Area under precision recall curve(AUPRC) and further try improving the results.

Metrics

The datasets 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.

Given the imbalance ratio, we would be using Area Under the Precision-Recall Curve (AUPRC) to measure accuracy. The closer to 1 the AUPRC is, the better the model is. A model with AUPRC score of 1 implies a perfect classifier.

Precision is the proportion of test cases predicted to be fraud that were indeed fraudulent (i.e. the true positive predictions), while recall or sensitivity is the proportion of fraud cases that were identified as fraud. This precision-recall curve(AUPRC) tells us the relationship between correct fraud predictions and the proportion of fraud cases that were detected (e.g. if all or most fraud cases were identified, we also have many non-fraud cases predicted as fraud and vice versa).

II. Analysis

Data Exploration

The dataset for the project is a set of credit card transactions over a period of 2 days with each transaction labelled as ‘fraudulent’ or ‘genuine’. To maintain confidentiality this data is anonymized & the original features were not provided.

The data has only numerical input variables which are the result of a PCA transformation. Again, due to confidentiality issues, there is no background information available for this data. Features V1, V2, ... V28 are the principal components obtained with PCA, the only features which have not been transformed with PCA are 'Time' and 'Amount'.

Feature 'Time' contains the seconds elapsed between each transaction and the first transaction in the dataset. The time between first and last transaction is 172,792 seconds or 48 hours. So the data provided has the credit card transactions which occurred over a period of 2 days.

The feature 'Amount' is the transaction Amount, this feature can be used for example-dependant cost-sensitive learning. The average transaction amount for fraudulent & genuine transactions is 122.21 and 88.29 respectively.

Feature 'Class' is the response variable and it takes value 1 in case of fraud and 0 otherwise.

Exploratory Visualization

Distribution of Class

Figure-1 below, shows the distribution of the class i.e. no. of fraudulent transactions vs others. It can be clearly seen that the class i.e. 1 or the fraudulent transactions is the minority class and the dataset is highly unbalanced.

```
Total records count  284807
Fraud records count  492
Non Fraud records count  284315
```

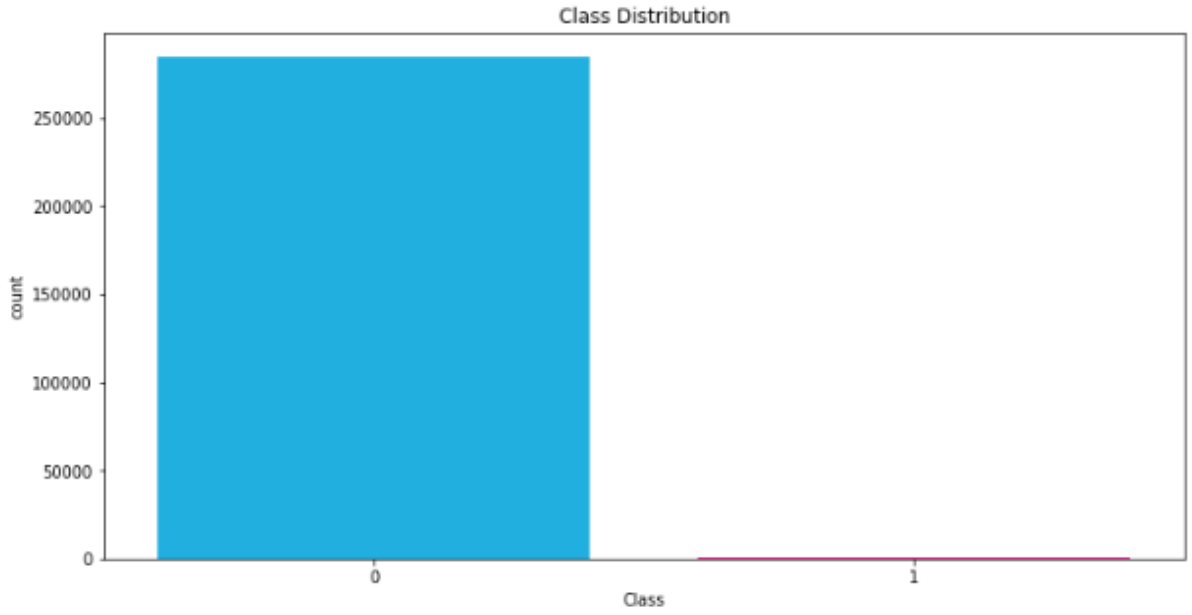

Figure-1

Explore Amount

Figure-2 below shows the spread of transaction amounts for fraudulent and normal transactions. Most of the Fraudulent transactions are low value as compared to non-fraudulent or normal transactions. The maximum amount for a fraudulent transaction is 2,125.

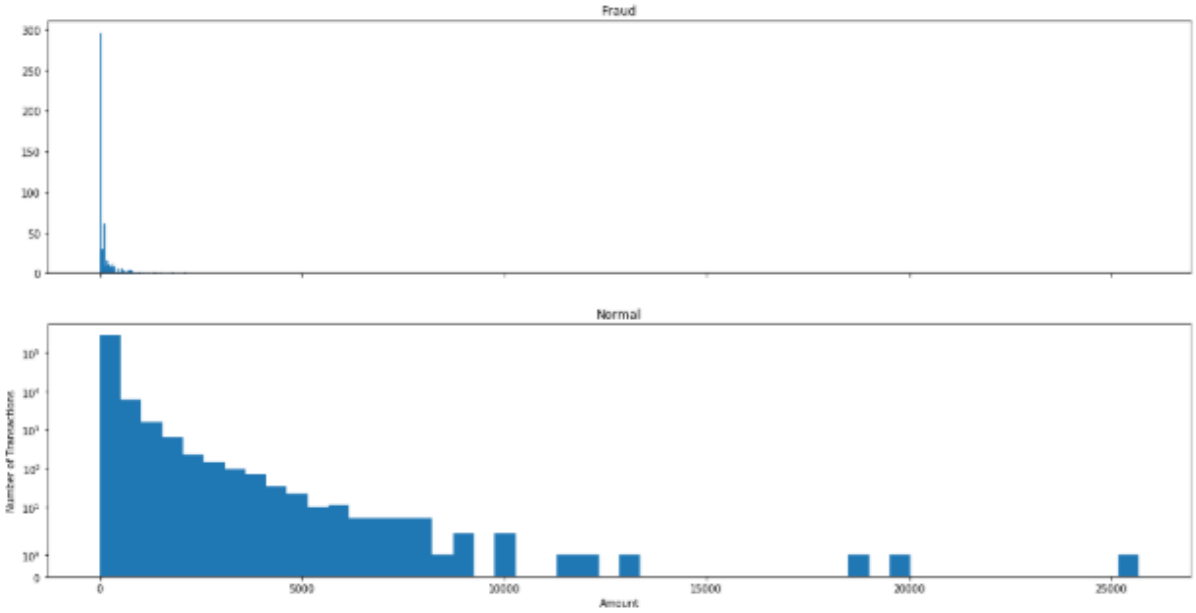

Figure-2

Explore Time vs Amount

Figure-3 below shows the spread of amount values for fraudulent & normal transactions over a period of 48 hours. The way the normal transactions occur over a period of 48 hour looks cyclical but not for Fraud transactions. There is no other significant reading from the below plot.

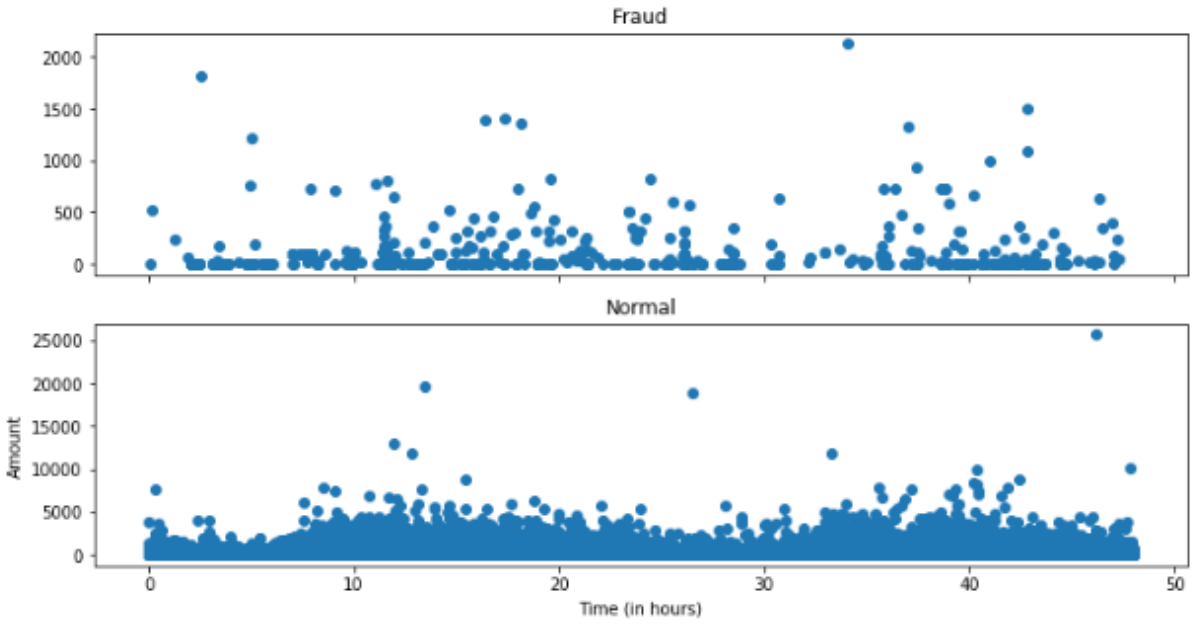

Figure-3

Algorithms and Techniques

This is a binary classification problem with a highly unbalanced dataset. Based on the nature of problem 4 machine learning algorithms namely, Logistic regression, Adaboost, Ridge Classifier and Light gradient boosting algorithms were tested.

Logistic Regression

Logistic regression is named for the function used at the core of the method, the logistic function also called as the sigmoid function. It's an S-shaped curve that can take any real-valued number and map it into a value between 0 and 1, but never exactly at those limits.

The coefficients or the weights of the logistic regression algorithm must be estimated from the training data. This is done using Maximum-likelihood estimation (a common learning algorithm used by a variety of machine learning algorithms).

The best coefficients would result in a model that would predict a value very close to 1 (e.g. non-fraud) for the default class and a value very close to 0 (e.g. fraud) for the other class. The intuition for maximum-likelihood for logistic regression is that a search procedure seeks values for the coefficients (Beta values) that minimize the error in the probabilities predicted by the model to those in the data (e.g. probability of 1 if the data is the primary class).

Adaboost (Adaptive Boosting)

Boosting is a general ensemble method that creates a strong classifier from a number of weak classifiers. AdaBoost can be used to boost the performance of any machine learning algorithm. It is best used with weak learners. These are models that achieve accuracy just above random chance on a classification problem.

AdaBoost is best used to boost the performance of decision trees on binary classification problems. This is done by building a model from the training data, then creating a second model that attempts to correct the errors from the first model. Models are added until the training set is predicted perfectly or a maximum number of models are added.

Ridge Classifier

Ridge regression is a method that seeks to reduce the mean square error, by adding some bias and, at the same time, reducing the variance. It learns weights & bias, adds a penalty for large variations in weights parameters. This penalizing of the parameter is called as regularization which prevents overfitting by restricting the model, typically to reduce the complexity.

The influence of the regularization term is controlled by the α parameter. Higher α means more regularization and simpler model. We would be using the default alpha value for our model.

Light Gradient Boosting or LGB

LightGBM is a fast, distributed as well as high-performance gradient boosting framework that makes the use of a learning algorithm that is tree-based, and is used for ranking, classification as well as many other machine learning tasks. While other algorithms grow trees horizontally, Light GBM grows tree vertically meaning that Light GBM grows tree leaf-wise while other algorithms grow level-wise. It, in order to grow, will choose the leaf that has a max delta loss. When growing the same leaf, Leaf-wise algorithm can reduce more loss when compared to a level-wise algorithm.

Benchmark

The biggest challenge with this project is highly unbalanced data and accuracy score or the confusion matrix is not helpful metrics for such data. For this imbalanced dataset the metrics, area under the precision recall curve(AUPRC) would be used.

For benchmark, a simple classification model like Logistic regression model was run as the benchmark model and it produced an AUPRC of 0.52. This would be compared against different models with a goal of making sure that any models producing results are worth considering only if the AUPRC score is greater than 0.52.

III. Methodology

Data Preprocessing

Since the dataset was highly imbalanced, to balance the dataset SMOTE algorithm was applied. SMOTE or synthetic minority oversampling technique is an approach which works from the perspective of existing minority instances and synthesises new instances at some distance from them towards one of their neighbours.

Class Distribution	*Fraudulent*	*Non-Fraudulent*
Before Oversampling	492 (0.173%)	284807 (99.837%)
After Oversampling(SMOTE)	284315 (50%)	284315 (50%)

Table-1

Implementation

Implement multiple models (Initial run- unbalanced dataset)

Implementation was done by training 4 different algorithms and comparing there performance using the AUPRC metric. Below table compares the performance of the 4 algorithms. This was done by dividing the data into training and testing sets where the data kept for testing was 30% of overall data.

Note: No oversampling done

Model	*AUPRC*	*Precision*	*Recall*	*Train time*	*Test Time*
Adaboost	0.8242	0.8248	0.7635	61.88	0.76
Ridge Classifier	0.7658	0.8133	0.4121	0.38	.01
Logistic Regression	0.6019	0.7164	0.6486	6.49	.01
Light Gradient Boosting	0.8820	0.9516	0.7973	12.8	0.61

Table-2

It was very evident from initial results that the 'Light Gradient Boosting' or LGB has the best metrics for the AUPRC score of 0.8820. The model gives the best precision of .9516 at 0.7973 recall. All the 3 metrics are by far the best for this model.

Overcome the imbalance – Apply oversampling

We have already tried penalizing the dominating class by using class weights. We can now further try and improve results by using resampling the training data using oversampling or under-sampling techniques. I'm focussing on resampling by oversampling the minority class by using a technique called SMOTE (Synthetic Minority Over-Sampling Technique) as explained in the section above.

Below are the results for various models after oversampling the data

Model	***AUPRC***	***Precision***	***Recall***	***Train time***	***Test Time***
Adaboost	0.9985	0.9876	0.9781	167.99	1.71
Ridge Classifier	0.9885	0.9875	0.8763	0.36	.01
Logistic Regression	0.9931	0.9832	0.9487	7.05	.01
Light Gradient Boosting	**0.9992**	**0.9978**	**0.9850**	**7.83**	**0.61**

Table-3

Refinement

Applying class weights

Class weightss were applied on 'Logistic regression' and 'Ridge Classifier' to further improve the performance by putting more weight or emphasis on the class 1 or the frauds. Below table has updated metrics for these 2 classifiers only.

Both the models had a better overall performance but the improvement in Logistic regression was significant. The AUPRC score for logistic regression improved from 0.6019 to 0.7814 after applying class_weights. Even then the performance of 'Light Gradient Boosting' was far better than any of the algorithms.

Model	*AUPRC*	*Precision*	*Recall*	*Train time*	*Test Time*
Adaboost	0.8242	0.8248	0.7635	61.88	0.76
Ridge Classifier	0.7706	0.8627	0.5946	0.36	.01
Logistic Regression	0.7814	0.7530	0.8446	7.05	.01
Light Gradient Boosting	**0.8820**	**0.9516**	**0.7973**	**12.8**	**0.61**

Table-4

Applying class weights (with oversampled data, from SMOTE)

With the dataset balanced, class weights were applied to the Logistic regression and the Ridge classifier. The overall AUPRC and the Recall improved for both the models but the precision dropped.

Model	*AUPRC*	*Precision*	*Recall*	*Train time*	*Test Time*
Adaboost	0.9985	0.9876	0.9781	167.99	1.71
Ridge Classifier	0.9725	0.7369	.9826	.7523	.01
Logistic Regression	0.9940	0.9068	0.9840	6.51	.01

Light Gradient Boosting	**0.9992**	**0.9978**	**0.9850**	**7.83**	**0.61**

Table-5

IV. Results

Model Evaluation and Validation

Comparing the 4 algorithms, **Light Gradient Boosting(LGB)** algorithm has outperformed all the other models namely -Logistic Regression, Adaboost and Ridge Classifier for all the metrics namely precision, recall and the AUPRC.

Adaboost model has a lower but a comparable AUPRC score of 99.85% but it takes much longer to train/test compared to the LGB. LGB produces better results in $1/20^{th}$ of the time when compared to Adaboost. So, working on Adaboost to further refine the results is not reasonable.

Justification

The best model i.e. LGB model on the oversampled data has a precision of 99.78% at 98.5% recall with AUPRC score of 99.92% which is very good and a significant improvement when compared to the benchmark model which has an AUPRC score of of 52%(77% precision at 63% recall).

V. Conclusion

Free-Form Visualization

Below figure illustrates the improvement in overall performance as compared to the benchmark model. The figure on left shows the area under the precision recall curve for the benchmark model and the one on right is for the selected LGB model.

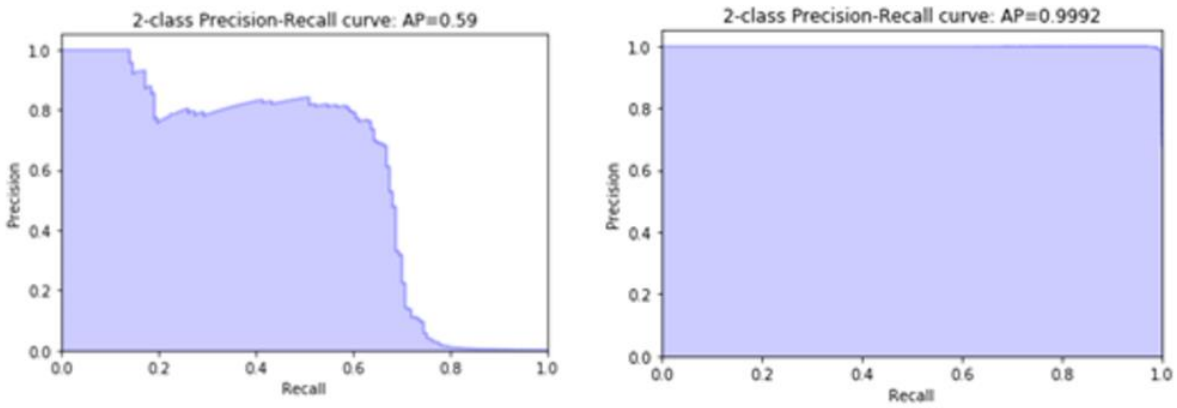

Figure-4

LGB Model predicted 83,972 fraudulent transactions correctly and 1,317 fraudulent transactions were missed and incorrectly predicted as non fraud. Only 170 of non Fraudulent transactions were wrongfully identified as fraud transactions.

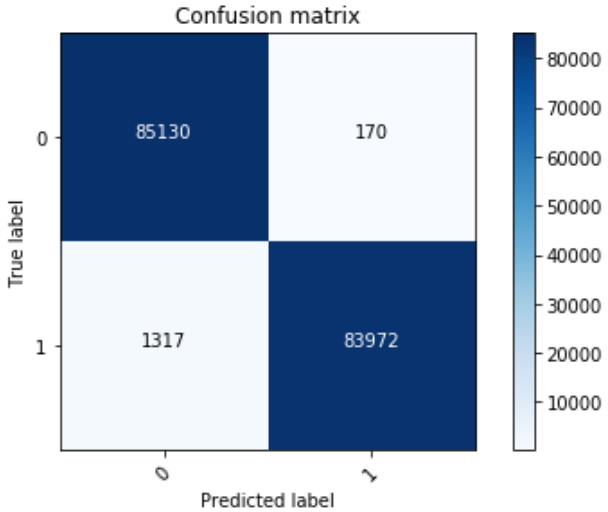

Figure-5

Reflection

In my view the major challenge for this project was the imbalnced classes which was very well taken care by the oversampling technique called SMOTE. Post oversampling the same algorithms performed much better. I also tried applying class weights to some of the models which did make sense and did show significant improvement for Logistic regressions pre-oversampling.

Improvement

- A definite and further improvement would be possible if we have better description of the fields in order to do more meaningful analysis of the data.
- Feature selection may be done and focus on the most relevant features may help improving the results
- More data and more examples of frauds would help in improving the results and we can try and apply deep learning techniques to further try and improve results

---

References

https://medium.com/data-science-group-iitr/logistic-regression-simplified-9b4efe801389

https://machinelearningmastery.com/boosting-and-adaboost-for-machine-learning/

https://www.coursera.org/lecture/python-machine-learning/linear-regression-ridge-lasso-and-polynomial-regression-M7yUQ?authMode=signup

https://www.techleer.com/articles/489-lightgbm-a-light-gradient-boosting-machine/