IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661,p-ISSN: 2278-8727, Volume 22, Issue 3, Ser. II (May - June 2020), PP 38-45
www.iosrjournals.org

Binary Classification Model for Fraudulent Credit Card Transactions

Aishwarya V
Department of Computer Science and Engineering
Sir M Visveswaraya Institute of Technology

Abstract: *Financial Fraud has been around for centuries since the time internet banking has taken off and is increasing substantially with the innovation of technology and the global superhighways of communication as a consequence of losing out billions of dollars world-wide each year. Unusually, large transactions or the ones that happen in atypical locations evidently deserve additional verification which could be detected by looking at on-surface and evident signals. So as to overcome these fraudulent acts we employ a fraud detection system which not only detects fraud but further makes it essential to model past credit card transactions with the ones that are atypical or turned out to be fraud.*

In this research paper, my objective is to identify the typical patterns found in transactions by employing a supervised learning approach and trainabinary classification model that can identify the features of these transactions and sort the data into one of two transaction classes: fraudulent or valid, based on provided, historical data. I have explained a Linear learning technique which could have two applications: Regression and Binary Classification that uses Amazon Sagemaker and an extensive review is done on Binary Classification on the basis of quantitative measurements such as accuracy, detection rate and false alarm rates.

Keywords: Financial Fraud Transactions, Supervised Learning, Binary Classifier, Linear learner, Amazon Sagemaker.

Date of Submission: 18-05-2020 Date of Acceptance: 03-06-2020

I. Introduction

Payment fraud is about an individual finding a card information and using it for buying something for themselves. The way that a bank identifies pseudo like this is by looking at the transactions and noting if something is atypical or outside of your spending patterns.

Assuming that I have only bought groceries in the past few months, possibly it would be suspicious if I were to move out for lots and lots of clothing which also depends on the individual's holidays and his motto. Pertaining to an increase in Credit Card holders, the cases of fraud in Credit Cards have been in rise and with the emerging technology it is a growing dispute all over the world today. The explosion of payment/credit card fraud is not only hit due to an unvariable escalation of card usage but to not paying enough attention for detection and prevention which risks the banks long after the criminal has committed the crime.

As stated, Credit cards being one of the most illustrous targets of fraud, although not the only one, frauds can yet emerge with any category of credit products such as personal loans, retails and home loans. Concerning the current payment fraud, criminals are beginning to alter their approach to rise, besides initiating several high-tech frauds such as APT, Phishing, Spoofing, etc., consequently a segment of the criminals is reverting back to it's archaic school class of malicious activities, like-wise lost and stolen, most instances in combination with social engineering and as e-commerce remains on the rise, CNP fraud remains a significant factor for fraud losses. In a 2016 case-study, it was estimated that credit card fraud was responsible for over 20 billion dollars in loss, worldwide. Accurately detecting cases of fraud is an ongoing area of research.

This paper aims to put forward some the essential terms used in fraud and reviews the main class of payment/credit card fraud and the decision techniques that can be deployed.

Some of the common re-markable strategies of Credit Card fraud are referred to as below:

o Card ID Theft: Card ID theft can occur when criminals gain or have access to your bank or credit card accounts—usually because of a data breach, phishing scam, or malware attack—and start making charges to those accounts where your name can be used with it.

o Assumed identity: This a scenario where a criminal may use a temporary address and an invalid name to secure the credit card. The banks could perhaps use a number of systems for protection from the fraud by consulting the customers to provide account details and thereby check to ascertain they are genuine.

o Counterfiet Card Fraud: It is committed to intimate something authentic with the intent to steal/skim-meaning to copy a magnetic stripe on a legitimate card(your card details). This fake strip is later used to produce a fraudulent card that is fully functional.

o Application Fraud: Application fraud is in conjunction with identity theft creating fraudulent accounts by stealing someone else's supporting documents and substantiating it with theirs. Although banks have adopted various safeguarding measures in place to halt this type of fraud from happening.

o Account Takeover: Account takeover fraud is emphasizing one of the most common forms of credit card fraud. A criminal by some means acquires all of your information and relevant documents online and tries to contact the credit card company and they pretend to be you requesting them to change the address.

Several approaches put forward to the above strategies are mentioned below:

o Genetic Algorithm: A genetic algorithm is a search heuristic which reflects the process of natural selection for obtaining the best possible solution to the problem. In regard with fraud, it's main significance is to secure an efficient e-payment system to provide some evidence whether the given transaction is valid or fraudulent.

o Support Vector Machine: Support vector machine is employed to predict and classify the data based on fraudulent or non-fraudulent and gives us a unique solution. It learns the behaviour of fraud and genuine transactions and models new transaction as to which class it belongs. It requires training the machine to feed supervised data i.e. data with results already known. The proposed method gives higher accuracy of detection and is also scalable for handling large volumes of transactions.

- Bayesian Network: A Bayesian network is an acyclic graph where the nodes correspond to random variables and edges correspond to conditional dependence of attributes. The training data in Bayesian network is split into two tasks: The first approach is to find regions of clusterization for each of the attributes.
The second approach is to build statistical data for Bayesian Network for implementing real time credit card fraud detection and prevention.

- Artificial Neural Network: Artificial Neural Networks algorithm can be employed to model any complex transactional patterns. The neural network consents to many inputs, sums them and connects them in a systematic way and generates a result either as model prediction or as inputs to other neurons. The artificial neural network trains either supervised methods where the outcome is already known for a given transaction or unsupervised method where we are provided with no actual results to compare it the atypical ones and thus are not sure about the results.

- Decision Trees: A decision tree technique is a support tool structure that uses a tree-like a model of decisions each bearing an internal node that represents the chances of outcome of the test

II. Literature Review

The subject of fraud detection has been of a wide range globally, that the entire textbooks, training programs, and comparabaly companies have been devoted to it exclusively. A good deal of study and research has gone into the essence of 'Credit Card Fraud Detection'. With the ascent in machine learning techniques and artificial intelligence, researchers have tandem reviewed papers and have implemented patterns to evade future fraudsters from committing the fraud.

Bhattacharya S, Jha S, Tharakunnel K have summarized a comparative study on data mining approaches that tend to ascertain the productivity of the few available techniques that help to combat the fraudsters. This research wrapped-up the contrast of various machine learning techniques that are accessible to prognosticate the credit card fraud. . The analysis specifying the comparison between various different techniques like SVM, logistic regression, neural networks constituted that neural networks performed the best in combating the fraud.

E. Duman, in [6] aimed to denote the advantages/superiority of applying the data mining techniques including Decision Trees & Support Vector Machine (SVM) to the credit card fraud detection problem to lessen the banks risk. The results obtained manifested that the classifiers & other Decision Tree approaches outmatched SVM approaches in solving the problem under investigation.

Wang'[4] in their paper proposed an approach to credit card fraud detection employing an outlier mining based on a distance sum. Outlier mining known to be a field of data mining is basically accustomed to be used in monetary and internet fields that deals with detecting objects which are pulled apart from the main system i.e. those transactions that aren't authentic.

Geoffrey F.Miller, Peter M.Todd [9] have stated the problem associated with intuitive network design by humans and have put forward the idea of an automated evolutionary deign method that is revolving on genetic algorithms and as a solution to it they have modeled a system which would have applications in biological, neurological and psychological modelling as well as the engineering and design applications using automated network design. Their research aims to free the network design process from the constraints of human biases.

Pooja Chougule and others showed that how k-means algorithm grouped the transactions based on the distinct attribute values by proposing a simple K-means and Simple Genetic Algorithm for fraud detection. Genetic algorithm was employed for optimization as with the increase in size of the input k-means algorithm produced outliers.

Numerous papers have been focusing on perceiving fraudulent transactions that are based on deep neural networks. Prior to these techniques, the models are computationally expensive but perform better on larger datasets. This approach may lead to obtaining great results, as we saw in some papers, but what if same results, or even better, can be achieved with less amount of resources? Our main goal is to show that different machine learning algorithms can give decent results with appropriate preprocessing.

Deep learning models which abide for data hungry learn through experience and perform the best when it is fed with more and more data. Also a lot unstructured and structured data is available with the banks, which the deep learning models can tackle on. Therefore, lately deep learning has been used exclusively in data mining research for fraud detection.

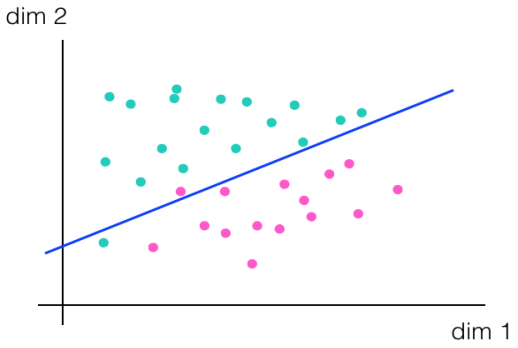
III. Methodology

To approach the task of fraud detection, we directed the credit card data initially obtained from Kaggle. The acquired data included thousands of transactions with anonymized features and class labels that were fraudulent or valid.

In this paper, I have employed a Machine learning workflow that works well for loading and exploring the data and progressing onto splitting that data and parsing it into a binary classification model. This experiment emphasizes on training and deploying a built-in linear model using Amazon SageMaker's Linearlearner. The Amazon Sagemaker linear learner encompasses algorithm that are used extensively in banking, fraud/risk management, insurance, and healthcare.

The Sagemaker's Linearlearner has two main applications:

One for regression tasks in which a linear line is to fit some data points and predict the output value. Second application is to do Binary classification in which the Linearlearner basically learns a line to separate two classes of data which aims to effectively output labels as either the class label 1 that falls above the line or zero that fall on or below the line.



This is a simple, clearly expressed two-dimensional case, say a plain or a multi-dimensional line for multiple features that can be used as a binary classifier to predict the two transaction classes valid zero, or fraudulent one. To do this we present a step-by-step methodology on Linearlearner Estimator:

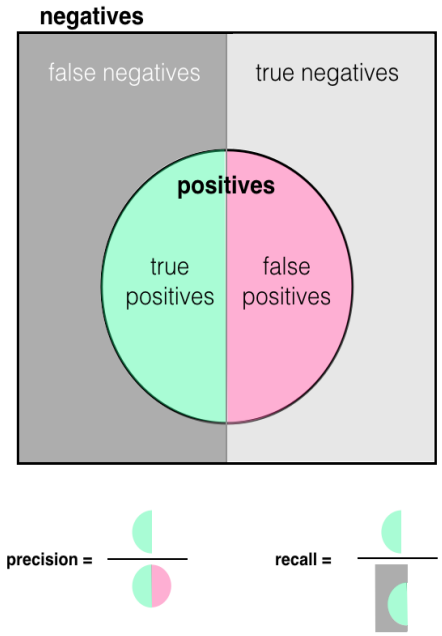
I have inputted the linear learner estimator algorithm from SageMaker and specified the s3 bucket to place the saved model attributes. In this case specifying a prefix 'credit card' and a full s3 which points to the directory credit card.

```
from sagemaker import LinearLearner
prefix='credit card'
output path='s3://{}/{}'.format(bucket, prefix)

linear=LinearLearner(role=role,
 train_instance_count=1,
 train_instance_type='xyxyxyxy',
 predictor_type='binary classifier',
            output_path=output_path,
            sagemaker_session=sagemker_session,
 epochs=15)
```

The estimator in the above algorithm shows a binary classifier and have passed a number of epochs say 15, which is the number of times that the algorithm will iterate over the entire training test as it learns.

The default Linearlearner could get a high accuracy, yet sometimes it tends to misclassify the fraudulent invalid data points. To improve the model, say when we envision designing the model for use in bank applications, we are aware that users do not want any valid transactions to be categorized as fraudulent i.e, we want to have as few false positives, zeroes classified as ones as possible. And if an application catches almost all cases of fraud even if it means a higher number of false positives then we optimize for as few false negative cases as possible. To train according to a specific product demands and goods, we could direct for a metric like precision or recall that could help us decrease the number of false positives and false negatives.



A model with high recall has as many true positives as few false negatives as possible, which is the number of true positive over the number of true positives plus false negatives. Hence, it will be the highest when the number of false negatives is zero.

To tune for a higher precision, we aim for a specific metric, thus Linearlearner offers the hyperparameter 'binary classification model selection criteria'.

Here, I have created my model and passed in a role, in this case a model selection criteria to 'precision at target recall'.

```
linear_recall=Linearlearner(role=role,
                train_instance_count=1,                    train_instance_type='xyxyxy',
        predictor_type='binaryclassifier',
                output_path=output_path,
sagemaker_session=sagemaker_session,
epochs=15,
binary_classification_model_selection_criteria='precision at target recall',
target_reacall=0.9,
positive_example_weight_mult='balanced'
)
```

This is the weight assigned to positive and fraudulent examples when training a binary classifier. The weight of negative examples or valid data is fixed at 1.

However, training machine learning models aims for a certain level of recall based on the constraints that you give it during the training process.

IV. Implementation

In my experiment I have used SageMaker for classification tasks in specific to credit cards and fraudulent transactions where the sagemaker interface makes the algorithm scalable and easy to update.

To deploy the trained model, I have employed the linear estimator and have been directed to a lot of data about that epic. The required status of this job could be checked upon in the sagemaker console. This can perceive you to some of the hyperparameters that are either default values or values that you specify:

key	**value**
epochs	15
feature_dim	30
mini_batch_size	1000
predictor_type	binary_classifier

To monitor the training job is to do so by looking at the logs that are provided with a lot of data printed out.

We can thus expand this metrics to see the binary classification entropy loss. After successfully training the linear estimator, we deploy it and create a linear predictor to make predictions. As mentioned earlier, it is now evident that precision and recall are the two metrics for measuring the "success" or performance of the trained model.

V. Results

The code loads and unzips the data and reads into a csv file of all the transaction data.

```
Data shape (rows, cols):  (284807, 31)
```

	Time	V1	V2	V3	V4	V5	V6	V7	V8	V9	...	V21
0	0.0	-1.359807	-0.072781	2.536347	1.378155	-0.338321	0.462388	0.239599	0.098698	0.363787	...	-0.018307
1	0.0	1.191857	0.266151	0.166480	0.448154	0.060018	-0.082361	-0.078803	0.085102	-0.255425	...	-0.225775
2	1.0	-1.358354	-1.340163	1.773209	0.379780	-0.503198	1.800499	0.791461	0.247676	-1.514654	...	0.247998
3	1.0	-0.966272	-0.185226	1.792993	-0.863291	-0.010309	1.247203	0.237609	0.377436	-1.387024	...	-0.108300
4	2.0	-1.158233	0.877737	1.548718	0.403034	-0.407193	0.095921	0.592941	-0.270533	0.817739	...	-0.009431

5 rows × 31 columns

V22	V23	V24	V25	V26	V27	V28	Amount	Class
0.277838	-0.110474	0.066928	0.128539	-0.189115	0.133558	-0.021053	149.62	0
-0.638672	0.101288	-0.339846	0.167170	0.125895	-0.008983	0.014724	2.69	0
0.771679	0.909412	-0.689281	-0.327642	-0.139097	-0.055353	-0.059752	378.66	0
0.005274	-0.190321	-1.175575	0.647376	-0.221929	0.062723	0.061458	123.50	0
0.798278	-0.137458	0.141267	-0.206010	0.502292	0.219422	0.215153	69.99	0

With the distribution of the transaction, the total percentage of fraudulent data can be competed.

```
Fraudulent percentage =  0.001727485630620034
Total # of fraudulent pts:  492.0
Out of (total) pts:  284807
```

To test whether the above result makes sense, we define test/training and split the data into expected number of points and the labels are indeed class labels of (0,1).

```
Training data pts:  199364
Test data pts:  85443

First item:
 [ 1.19907000e+05 -6.11711999e-01 -7.69705324e-01 -1.49759145e-01
 -2.24876503e-01  2.02857736e+00 -2.01988711e+00  2.92491387e-01
 -5.23020325e-01  3.58468461e-01  7.00499612e-02 -8.54022784e-01
  5.47347360e-01  6.16448382e-01 -1.01785018e-01 -6.08491804e-01
 -2.88559430e-01 -6.06199260e-01 -9.00745518e-01 -2.01311157e-01
 -1.96039343e-01 -7.52077614e-02  4.55360454e-02  3.80739375e-01
  2.34403159e-02 -2.22068576e+00 -2.01145578e-01  6.65013699e-02
  2.21179560e-01  1.79000000e+00]
Label:  0.0

Tests passed!
```

Following the evaluation function we test the end points of the model metrics, thus calculating false negatives and positives as well as recall, precision and accuracy.

```
Metrics for simple, LinearLearner.

prediction (col)    0.0  1.0
actual (row)
0.0               85269   33
1.0                  32  109

Recall:     0.773
Precision:  0.768
Accuracy:   0.999
```

The model gets a very high accuracy of 99.9% but it still misclassifies about 30(each) of our valid and fraudulent cases, which results in a much lower values for recall and precision.
To enhance this, I have deployed a tune predictor and hypothesized that a tuned model optimized for a higher recall would have fewer false negatives.

```
Metrics for tuned (recall), LinearLearner.

prediction (col)    0.0   1.0
actual (row)
0.0              81913  3389
1.0                 10   131

Recall:     0.929
Precision:  0.037
```

To improve managing the class imbalance, we account for a model that is tuned to get a higher recall which aims to reduce the number of false negatives. The Linearlearner algorithm offers the hyperparameter 'positive_weight_example-mult' which is the weight assigned to positive samples when training a binary classifier.

```
Metrics for balanced, LinearLearner.

prediction (col)    0.0   1.0
actual (row)
0.0               84277  1025
1.0                  12   129

Recall:     0.915
Precision:  0.112
Accuracy:   0.988
```

The above trained model is tuned for the best possible precision with recall fixed at about 90%.

```
Metrics for tuned (precision), LinearLearner.

prediction (col)    0.0  1.0
actual (row)
0.0               85276   26
1.0                  31  110

Recall:     0.780
Precision:  0.809
Accuracy:   0.999
```

This is a scenario wherein a performance on the training set will be within 5-10% of the performance on the test set. For an instance, if you get 80% on a training set, it is evident that you will get an accuracy between 70-90% on the test set.

The Linearlearner model is very well suited for a binary classification task that involves design decisions and managing class imbalance in the training set.

VI. Conclusion

This paper focused on how to train and deploy a model using a Linearlearner algorithm and SageMaker to aim for a certain level of recall thereby employing a binary classification method which wrapped-up design decisions and helped manage class imbalance in the training set.

SageMaker contributed a simple way to tune models based on the constraints given to it during the training process.

Following the mechanism of machine learning workflow, the credit card transaction data was loaded, explored and was prepared for model training. It was further trained, deployed and later evaluated several models in accordance with different design considerations and improved it over time.

References

[1]. "Research on Credit Card Fraud Detection Model Based on Distance Sum – by Wen-Fang YU and Na Wang" published by 2009 International Joint Conference on Artificial Intelligence

[2]. C. Wang, Y. Wang, Z. Ye, L. Yan, W. Cai, S. Pan, "Credit card fraud detection based on whale algorithm optimized BP neural network", 2018 13th International Conference on Computer Science & Education (ICCSE) pp. 1-4. IEEE.

[3]. "P. M.,. S. H. Geoffrey F.Miller, "Designing Neural networks using genetic algorithms," [Online]. Available: "neural networks-pdf".

[4]. A. Mishra, C. Ghorpade, "Credit Card Fraud Detection on the Skewed Data Using Various Classification and Ensemble Techniques" 2018 IEEE International Students' Conference on Electrical, Electronics and Computer Science (SCEECS) pp. 1-5. IEEE.

[5]. "Credit Card Fraud Detection-by Ishu Trivedi, Monika, Mrigya, Mridushi" published by International Journal of Advanced Research in Computer and Communication Engineering Vol. 5, Issue 1, January 2016

[6]. David J.Wetson,David J.Hand,M Adams,Whitrow and Piotr Jusczak "Plastic Card Fraud Detection using Peer Group Analysis" Springer, Issue 2010.

[7]. Global Facts (2019). Topic: Startups worldwide. [online] Available at: https://www.statista.com/topics/4733/startups-worldwide/ [Accessed 10 Jan. 2019].

[8]. Global Facts (2019). Topic: Startups worldwide. [online] Available at: https://www.statista.com/topics/4733/startups-worldwide/ [Accessed 10 Jan. 2019].

Aishwarya V. "Binary Classification Model for Fraudulent Credit Card Transactions." *IOSR Journal of Computer Engineering (IOSR-JCE)*, 22(3), 2020, pp. 38-45.