Credit Card Fraud Detection using Data Science

Chandramouli Yalamanchili
DSC680 - T302 Applied Data Science (2215-1)
https://chandu85.github.io/data-science/

Abstract

We have witnessed an enormous evolution in credit card processing over last few years, issuing chip-based credit cards, starting mobile device-based wallets like Apple Pay are some of the significant changes done to secure credit card transactions.

Despite financial institutions (banks) working hard to eliminate fraud in credit card transactions, credit card fraud has been continuously rising over the last few years. Fraudsters are getting smarter and using latest technologies to steal cardholder's information, either through hacking or through social engineering.

Increasing fraud in the industry makes fraud prediction very critical to be able to identify and stop fraud in real time, and data science plays a significant role in analyzing and being able to predict fraud based on transactional and cardholder information. The scope of this project is to build a machine learning model that would detect fraud efficiently in real time with less human support.

Domain Introduction

This project will work in the domain of credit card transaction processing with the main focus on building a credit card fraud detection and prevention.

Credit card processing is one of the fast-growing industries due to rapid advances in technology and with more and more customers switching to use credit cards instead of cash for purchases. Innovations like mobile wallets provided by Apple, Google, and other major technology firms have played an enormous role in increased usage of credit cards in recent years.

On a very high level, credit card transactions can be of two types, card present, and the card not present transactions. Card present transactions are the transactions from retail stores or gas stations where cardholder is present during the transaction, and that makes fraud a little bit difficult as the fraudster has to either steal the physical card or copy the card details, to create a duplicate card. Fraud in card present transactions has reduced in recent years due to the introduction of chip cards (challenging to copy and reproduce) and increased usage of mobile wallets which have the same security as chip cards. That leaves us with the card not present transactions, where we are seeing an increased number of fraudulent transactions in recent years. These are usually e-commerce or online portal-based transactions. In this case, fraudsters needed very less information about the physical card and cardholder to perform the transactions.

Fraud transactions can be of different types, below are some examples of fraudulent transaction types:

- Merchant fraud - Merchant POS device is compromised and used to run fraudulent transactions.
- Application Fraud - Fraudster applying for a new credit card on behalf of the cardholder.
- Counterfeit Card Fraud - Usually committed through skimming. Information from the card is stolen and used to create a fake magnetic stripe card with stolen data.

- Lost/Stolen Fraud – Transactions are performed using the cards that are either stolen from the cardholder or lost by the cardholder.
- Not Received as Issued (NRI) - Fraudsters intercepts the mail and steal the credit cards issued to the cardholder.

Any fraudulent transaction will add liability to different parties in the transaction flow like the merchant, merchant processor, networks like Visa/MasterCard, issuing processor, issuing bank and even cardholder depending on who was the weak link for that transaction.

Although financial institutions are working hard to eliminate fraud in credit card transactions, it has been continuously rising as fraudsters are using the latest technologies to steal cardholder's information, either through hacking or through social engineering.

We can detect these fraudulent transactions by analyzing parameters from different segments of information like transactional information, historical information, etc. Also, considering the liability burden on banks, it is critical to be able to identify these fraudulent transactions in real time.

Capabilities of data science will add a great value to predict fraudulent transactions in real time and help financial institutions in preventing fraud. There are several techniques within data science to predict fraud. The goal of this project is to build a deep neural network to detect the fraud depending on the transactional features provided to the model.

Credit Card Fraud Statistics

There have been massive data breaches in the past few years, and these data breaches will make cardholder information available to fraudsters, subsequently increasing fraud. Below are some of the well-known data breaches happened in recent years:

- Yahoo data breach in 2016 has impacted 3 billion user accounts.
- Equifax data breach in 2017 has affected close to 150 million users.
- Identity theft has been close to 400,000, during 2012 – 2016.

Below statistics from figure 1 shows the change in the trend of the fraudulent transaction from the card present transaction until 2015 to the card not-present transactions after 2015.

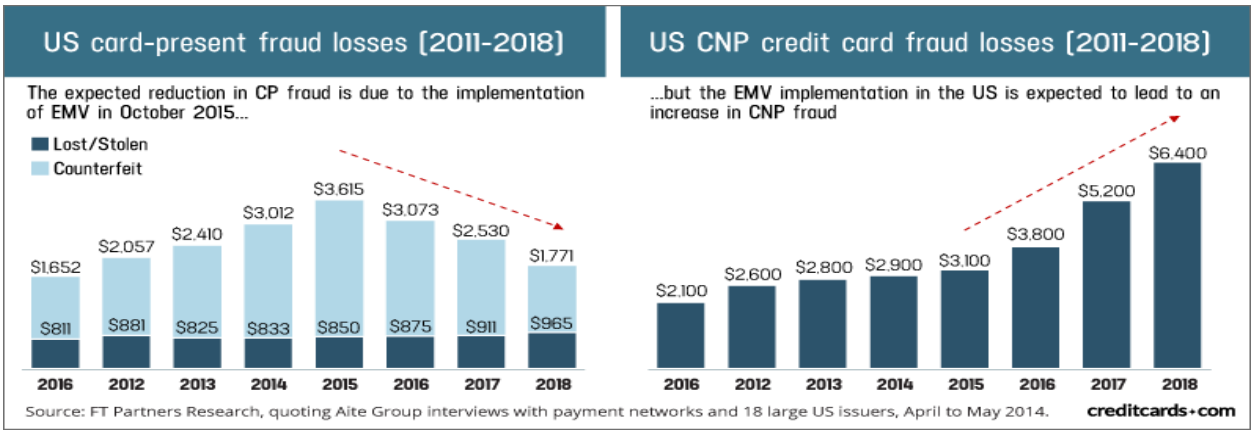


Figure 1: This image depicts the trend of reducing fraudulent transaction in card present transactions and increase of fraud using card not present transaction after 2015.[11]

Conventional Fraud detection applications

Currently, we have so many fraud prediction applications some of them are rules-based, and some of them are score based. These solutions use transactional and historical information to come up with fraud prediction. Below are a few drawbacks I have noticed with these types of applications:

- Rules-based - Complex to build and manage the rules, we have seen clients setting up bad rules resulting in false positives impacting cardholders.
- Score based - These applications use cardholder's shopping pattern, distance from the location of the previous transaction, etc. to come up fraud score indicating how risky the transaction is. Primary issue I have noticed with this solution is that a new model would take a minimum of 3 months to be ready for production.
- Common - Both of the applications are highly dependent on human support who has very good domain knowledge.
- Common - Some of the fraud prediction applications predict the fraud after the transaction got processed, even though it would stop subsequent fraudulent transactions, cardholder is already impacted for that first fraudulent transaction.

How can Data Science help in preventing Fraud?

We need a robust fraud detection system that can accommodate all of the complexities involved with credit card transactions like high volume processing, volatility, variety of transactions, and criticality, and be able to consider the vast number of attributes available in transactional or historical data and predict fraudulent transactions with high precision in real time.

The current solution of static rules-based fraud prediction tools won't stand a chance before rapidly evolving credit card industry as well as increasing fraud in the industry.

There are several machine learning algorithms that can be used to implement fraud prediction, in this project I have chosen Deep Neural Network (DNN) to see how much a deep learning machine learning model can help in detecting fraudulent transactions.

Project Details

Data

I have used the below credit card transactions dataset from Kaggle.

Dataset Link - https://www.kaggle.com/mlg-ulb/creditcardfraud

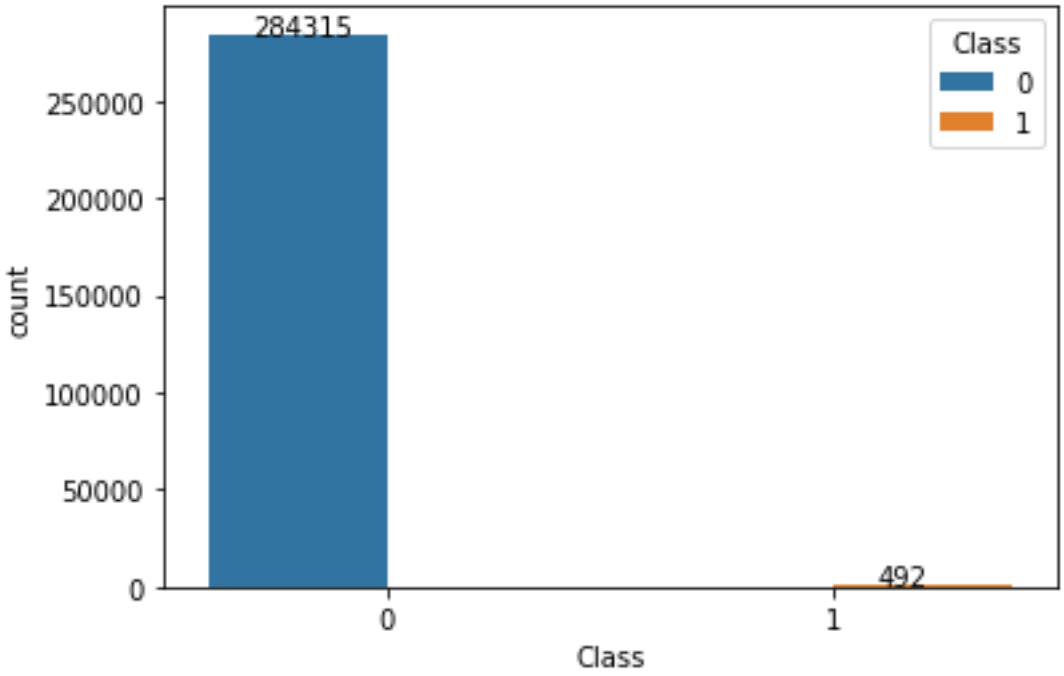
Feature Details

- This dataset has total of 284,807 credit card transactions from September 2013.
- Out of 248,807, 492 transactions are fraudulent, which accounts for only a 0.172% of positive classes.
- This dataset has only few columns in clear, rest of the columns have been PCA transformed due to the confidentiality of the data. Below are the details for the columns that are in clear.
- Time – This feature contains the seconds elapsed between each transaction and the first transaction in the dataset.
- Amount – This is the unaltered amount field on the transaction record.
- Fraud indicator/Class – This is the response variable that indicates whether a transaction is fraud (1) or not (0).
- We have 28 features that are PCA transformed due to the data compliance requirements.

Exploratory Data Analysis

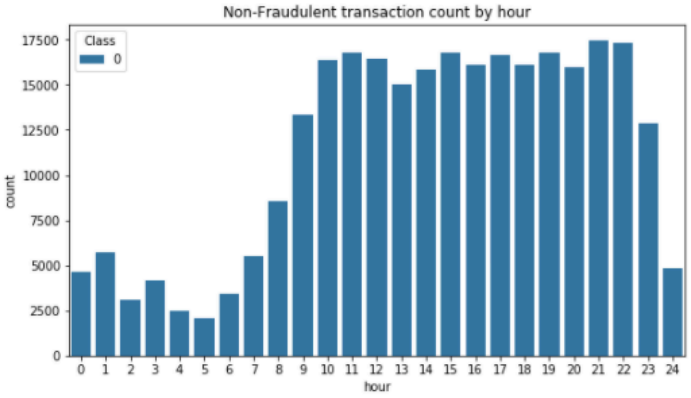
I have reviewed the data to see if I can confirm the general understand I have with respect to credit card transactions or the fraudulent transaction.

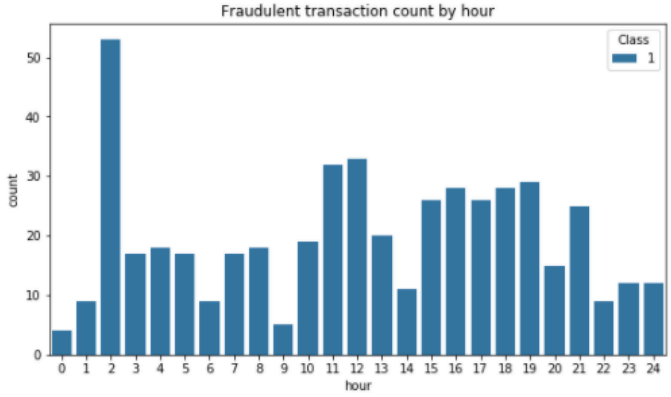
1.) Total distribution of transactions by fraud indicator.



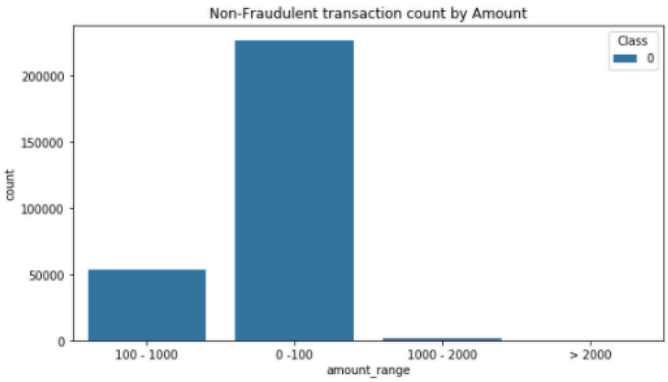
2.) I have derived the hour of the day by using the elapsed time feature, and by assuming that the transactions were captured starting from mid-night. As per the below chart, I can confirm below two observations are matching with my expectation in general:

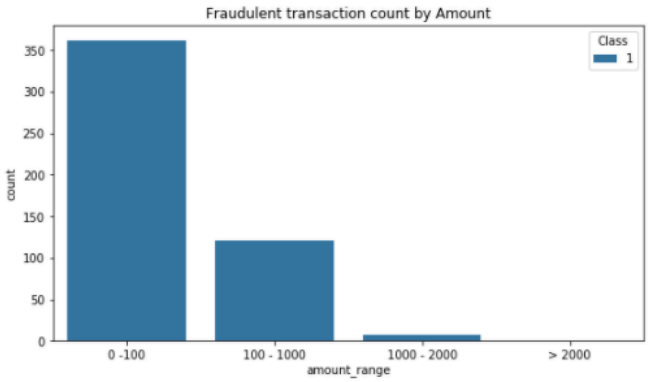
- Credit card Transactions are high during day hours.
- Fraudulent Transactions are high during night hours.



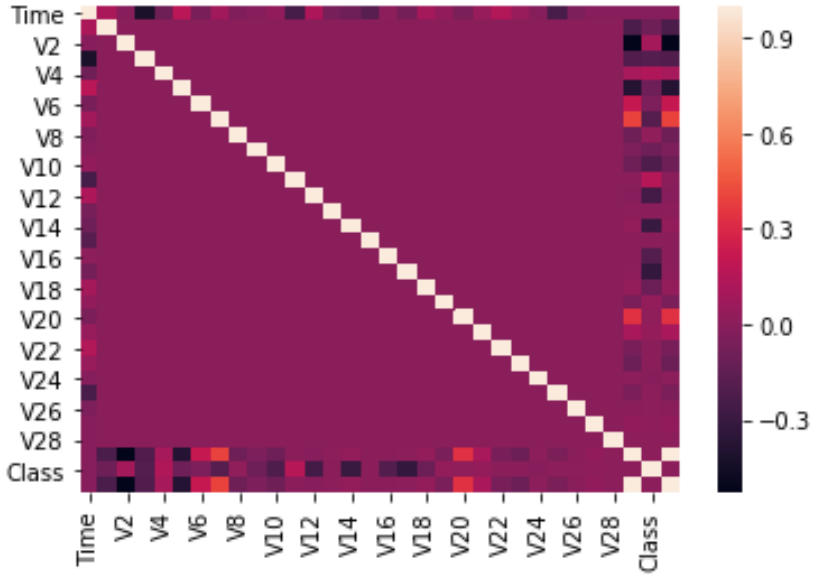


3.) I have used amount ranges to understand the distribution of the transactions by the transaction amounts. It seems like overall most of the transactions range from 0 - 100 amount. I was surprised to see considerable number of fraudulent transactions in 100-1000 amount range as well.





4.) Correlation - I have plotted the correlation between different features in the dataset, but I have seen only very few features with some correlation with the target variable, Class in this dataset.



Data Preparation

- I have normalized the amount and time fields to bring them to the range of -1 to +1 to be able to use these two features as input to the neural network.
- I have dropped the existing Amount and Time fields from the dataset as I will be using the normalized features instead.
- I have used sklearn's preprocessing package StandardScaler to standardize the time and amount features.

Modeling

Deep Neural Network with unbalanced sample data

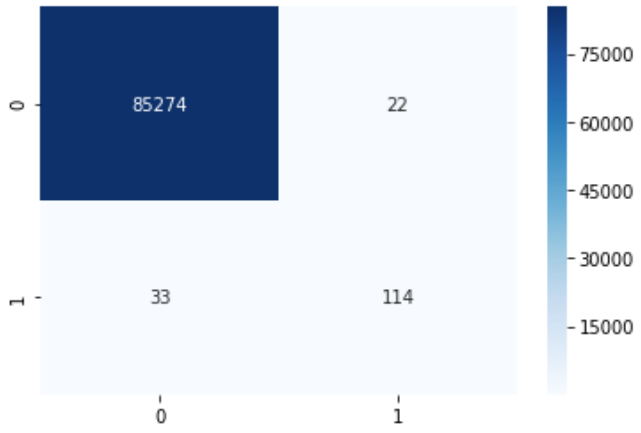
- Using Keras, I have built the neural network with 6 layers, including 4 hidden layers.

```
Model: "sequential_4"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense_16 (Dense)             (None, 16)                496
_________________________________________________________________
dense_17 (Dense)             (None, 24)                408
_________________________________________________________________
dropout_4 (Dropout)          (None, 24)                0
_________________________________________________________________
dense_18 (Dense)             (None, 20)                500
_________________________________________________________________
dense_19 (Dense)             (None, 24)                504
_________________________________________________________________
dense_20 (Dense)             (None, 1)                 25
=================================================================
Total params: 1,933
Trainable params: 1,933
Non-trainable params: 0
_________________________________________________________________
```

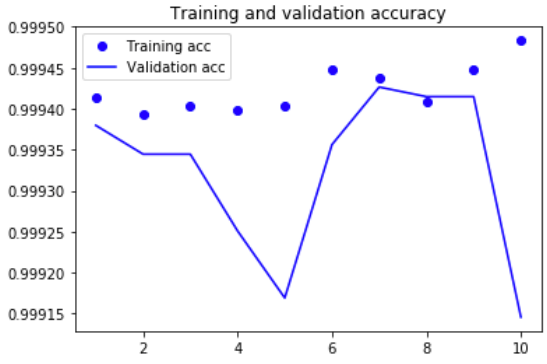
- Compiled the model using adam optimizer, binary_crossentropy loss parameter, and using the accuracy metric.
- Trained the model using training set for 10 epochs.
- Below are the model performance metrics at the end of training

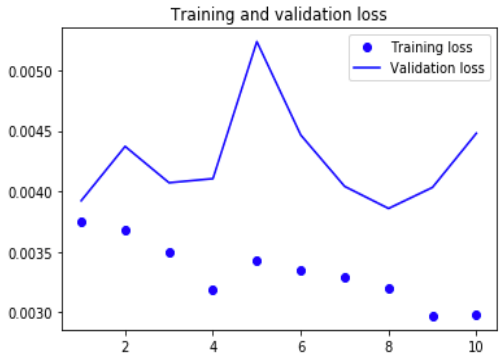
Evaluation Metric	Value
Loss	0.45%
Accuracy	99.91%
Accuracy Score	99.94%
Precision Score	83.82%
Recall Score	77.55%
F1 Score	80.57%

- As we can see from the metrics above, even though the accuracy is high, the precision is not that great because of more negative (class=0) samples in the dataset.
- Below is the confusion matrix:



- Due to the use of unbalanced dataset, the model has shown the signs of over-fitting, the validation results showed more losses than the training data.





Deep Neural Network using re-sampling technique

- Using Keras, I have built the neural network with 6 layers, including 4 hidden layers.

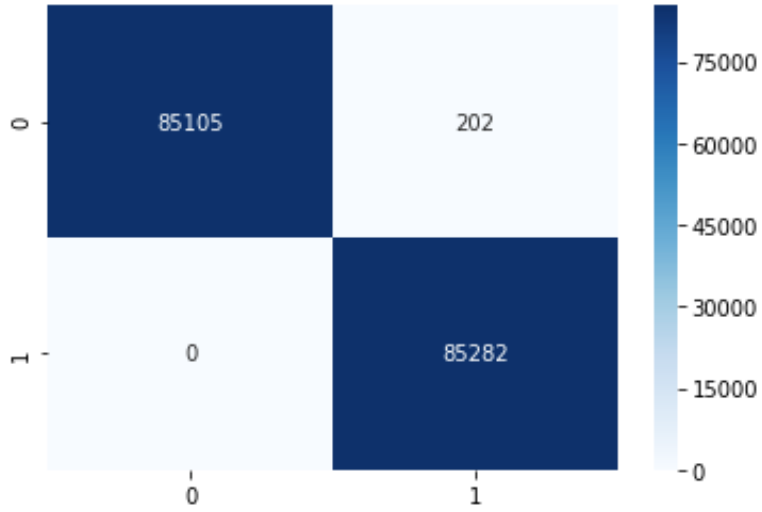
```
Model: "sequential_4"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense_16 (Dense)             (None, 16)                496
_________________________________________________________________
dense_17 (Dense)             (None, 24)                408
_________________________________________________________________
dropout_4 (Dropout)          (None, 24)                0
_________________________________________________________________
dense_18 (Dense)             (None, 20)                500
_________________________________________________________________
dense_19 (Dense)             (None, 24)                504
_________________________________________________________________
dense_20 (Dense)             (None, 1)                 25
=================================================================
Total params: 1,933
Trainable params: 1,933
Non-trainable params: 0
_________________________________________________________________
```

- Compiled the model using adam optimizer, binary_crossentropy loss parameter, and using the accuracy metric.
- Trained the model using training set for 50 epochs. (but I have noticed that 20 epochs are enough as the model stopped making improvements beyond 20 epochs).
- Using SMOTE, the data oversampling technique, I have re-created the training and testing data by making sure both classes are in equal proportions.
- Below are the model performance metrics at the end of training

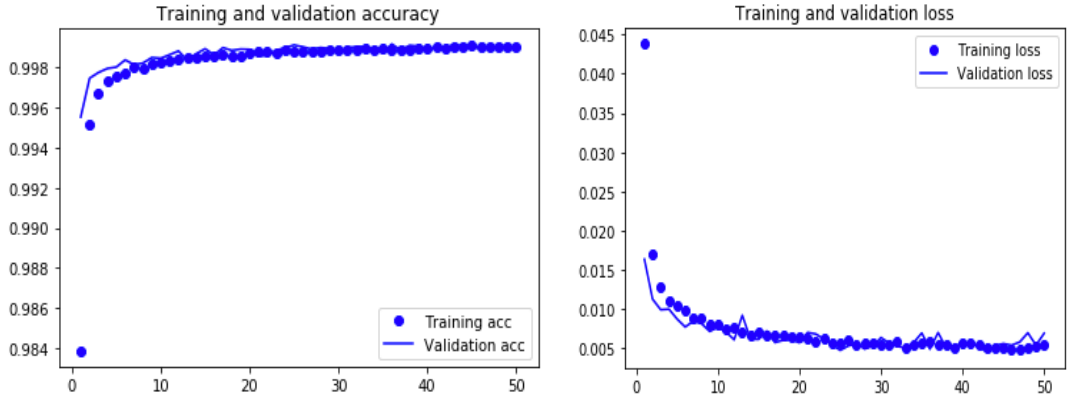
Evaluation Metric	Value
Loss	0.69%
Accuracy	99.88%
Accuracy Score	99.88%
Precision Score	99.76%

Recall Score	100%
F1 Score	99.88%

- As we can see from the metrics above, the model performance has improved significantly.
- Below is the confusion matrix:



- Below accuracy and loss curves shows that model is fitting fine with the new sample data.



Future Steps

As next steps, I would like to review different options available from references to see what can be done to integrate this DNN model into the transaction processing. I want to evaluate if there are any low-latency integration options that can detect the fraudulent transactions in real-time before the transaction decision is sent back to the merchant POS (Point Of Sale) device.

Conclusion

Credit card fraud is a growing concern in today's world with both credit card usage and credit card fraud on the rise. As shown in the stats, the number of data breaches, identity theft cases, and credit card fraudulent transactions are rising at an alarming level.

Data science plays a significant role in improving the fraud prediction tools we are currently using, by analyzing credit card transactions and being able to predict fraud based on transactional data, cardholder data, and historical data.

Machine learning algorithms running on neural networks can be effective as proven in this project and they should replace the current static rules-based fraud detection products to improve the fraud prediction precision and to stay up to speed with market trends in detecting new strategies of fraudsters and stopping them.

One common issue we will have with credit card transactions is less number of transactions that the number of fraudulent transactions will be very low when compared to the overall transaction volume. So, we have to use the relevant oversampling techniques to be able to train the models efficiently to do a better job at catching fraudulent transactions.

Acknowledgements

Thanks to Professor Fadi Alsaleem for providing continuous constructive feedback and peers for their valuable inputs and discussions.

I also thank all the authors of the reference papers and articles.

References

As I am looking to build a machine learning model to predict the fraud detection, I am going to review the work that is done already in this area and use the best options available to achieve high accuracy models to predict fraudulent transaction. Below are several references I am planning to refer as part this project:

1. Chan, P. K., Fan, W., Prodromidis, A. L., & Stolfo, S. J. (1999). Distributed data mining in credit card fraud detection. IEEE Intelligent systems, (6), 67- 74.
    - https://cs.fit.edu/~pkc/papers/ieee-is99.pdf
    - I have reviewed this reference to understand how they handled the skewed data for their experiment as credit card transaction data is highly skewed (legitimate transactions are very high in number compared to fraudulent transactions).
2. Brause, R., Langsdorf, T., & Hepp, M. (1999). Neural data mining for credit card fraud detection. In Proceedings 11th International Conference on Tools with Artificial Intelligence(pp. 103-106). IEEE.
    - http://sphinx.rbi.informatik.uni-frankfurt.de/asa/papers/ICTAI99.pdf
    - I have reviewed this reference to understand different types of advanced data mining techniques they have applied as well what type of neural network algorithms they have used.
3. Ghosh, S., & Reilly, D. L. (1994, January). Credit card fraud detection with a neural-network. In System Sciences, 1994. Proceedings of the Twenty- Seventh Hawaii International Conference on (Vol. 3, pp. 621-630). IEEE.
    - http://bit.csc.lsu.edu/~jianhua/quang.pdf
    - I have reviewed this reference to understand the P-RCE neural network that was used to implement FDS system in a financial institute. I will review this paper further to see if I can use any of the techniques mentioned as part of the machine learning model or neural network I will be building as part of this project.
4. Chan, P. K., & Stolfo, S. J. (1998, August). Toward Scalable Learning with Non-Uniform Class and Cost Distributions: A Case Study in Credit Card Fraud Detection. In KDD (Vol. 98, pp. 164-168).
    - https://www.aaai.org/Papers/KDD/1998/KDD98-026.pdf
    - This reference, similar to first one talks about dealing with the skewed data, I have reviewed this reference to understand if I can use any of the data distribution techniques they have used in their experiment.
5. Maes, S., Tuyls, K., Vanschoenwinkel, B., & Manderick, B. (2002, January). Credit card fraud detection using Bayesian and neural networks. In Proceedings of the 1st international naiso congress on neuro fuzzy technologies (pp. 261-270).
    - https://arxiv.org/pdf/1908.11553.pdf
    - This reference also concentrates on how to improve the minority sample in the input data as well as to reduce the noise. I have reviewed this reference to see if I can apply of these techniques into my project to achieve model with high accuracy.
6. Masoumeh Zareapoor, & Pourya Shamsolmoali (2015). Application of Credit Card Fraud Detection: Based on Bagging Ensemble Classifier. Procedia Computer Science (Vol. 48, pp. 679-685). https://doi.org/10.1016/j.procs.2015.04.201.
    - https://cyberleninka.org/article/n/324468.pdf

- This paper has applied different mining techniques to compare the performance of different methods, I I have reviewed this reference to ensure that I have used the right model for my project that would yield high accuracy.

7. Stolfo, S., Fan, D. W., Lee, W., Prodromidis, A., & Chan, P. (1997, July). Credit card fraud detection using meta-learning: Issues and initial results. In AAAI-97 Workshop on Fraud Detection and Risk Management.
    - https://www.aaai.org/Papers/Workshops/1997/WS-97-07/WS97-07-015.pdf
    - This research paper, similar to the previous one tested several machine learning algorithms as well as more importantly using the meta-learning strategies. I have reviewed this reference to get more understanding about the meta-learning strategies and see if I can use any of these strategies in my project as well as I have same data related issues as this project refers to.
8. Luke Sun (July 2020). Credit Card Fraud Detection.
    - https://towardsdatascience.com/credit-card-fraud-detection-9bc8db79b956
    - This article speaks about the data sampling issues as well as building different models to compare the performance of different modeling techniques. I have reviewed this reference to gain understanding on building the sample data as well as to build the machine learning models.
9. Dorronsoro, J. R., Ginel, F., Sgnchez, C., & Cruz, C. S. (1997). Neural fraud detection in credit card operations. IEEE transactions on neural networks, 8(4), 827-834.
    - https://repositorio.uam.es/bitstream/handle/10486/663701/neural_dorronsoro_ITNN_1997_ps.pdf;jsessionid=28C549CC8D6DFFC1AB4F2A16D511F89F?sequence=1
    - This paper talks about the Minerva fraud detection system, a real time fraud detection system using neural networks. I have reviewed this reference to understand how they are using neural network for fraud detection. Also one more interesting factor for this paper is their mainframe implementation, I wanted to understand how they are integrating the model to the IBM mainframe components so that I can see if it's feasible for our application at work as well.
10. Sánchez, D., Vila, M. A., Cerda, L., & Serrano, J. M. (2009). Association rules applied to credit card fraud detection. Expert systems with applications, 36(2), 3630-3640.
    - http://didawiki.cli.di.unipi.it/lib/exe/fetch.php/dm/ar-creditcard-fraudedetection.pdf
    - This paper reviews the use of association rules to determine fraudulent transactions. This paper probably will not be of much help for this particular project as most of the data I have is transformed. I have reviewed this reference to understand if I can gather any of the insights from this work for my project, also to see if this is something I can implement in my work environment.
11. Brian Fung March 2018, Equifax's massive 2017 data breach keeps getting worse. Retrieved May 16, 2019 from https://www.washingtonpost.com/news/the-switch/wp/2018/03/01/equifax-keeps-finding-millions-more-people-who-were-affected-by-its-massive-data-breach/?noredirect=on&utm_term=.0b854d8c3526

Appendix

No supplemental data at this time.