EAST WEST UNIVERSITY

Project Report

Date of Submission: - 9.09.2021

Submitted by,

Rezoanul Islam
ID: 2017-2-60-129

Anannya Das
ID: 2017-2-60-128

Md. Junayed Hossain
ID: 2017-1-60-107

Rabeya Akter
ID: 2017-2-60-144

Submitted to,

Instructor: Dr. Md Samiullah
Adjunct Faculty, Department of Computer Science and Engineering

Introduction:

Machine learning algorithms have been developed to detect fraudulent transactions throughout the financial sector. In this project, that is exactly what we are trying to do. We use nearly 285,000 credit card transaction datasets and some algorithms to identify transactions that are likely to be credit card fraud. In this project, we will build and distribute machine-learning algorithms to differentiate between the fraud and real ones.

In addition, we are exploring the use of data visualization techniques common in data science, such as parameter histograms to better understand the basic data distribution of datasets. The more we dug into this project the more we found variants of concepts related to Machine-learning. Hence, it has been a quite useful as a learning basis as well as implementing machine learning to a real life problem.

Background Study:

We have taken the dataset from Kaggle. The dataset contains transactions made by credit cards in September 2013 by European cardholders. The dataset presents transactions that occurred in two days. The dataset is highly imbalanced and the fraudulent transactions is 0.172% of all transactions.

It contains only numerical value which are the result of a PCA (Principal Component Analysis). Due to confidentiality issues, dataset did not come with the original features and more background information of the data. Only features that have not been transformed with PCA are "Time" and "Amount". "Time" contains the seconds elapsed between each transaction. The feature "Amount" is the amount of the transaction. Other features are labeled as "V1", "V2" ..."V28" which have been transformed by PCA. The feature "Class" is response variable and it takes the value 1 for fraudulent transactions and 0 for legit transactions

Our Idea & Implementation:

For data this huge, our very first approach to the problem was to under-sample the data to make it easier to calculate and train vice versa, our basic idea was using Logistic regression. The reasoning behind this approach was simply because logistic regression is intended for binary (two-class) classification problems. It will predict the probability of an instance belonging to the default class, which can be snapped into a 0 or 1 classification.

First, we cut down or so to speak omitted most of the data randomly and took a shy portion of the dataset in our analyzation. In details, we dropped the time column from the whole dataset.

```
# droping the Time column
credit_card_data = credit_card_data.drop(columns="Time", axis=1)
```

The distribution of the dataset is:

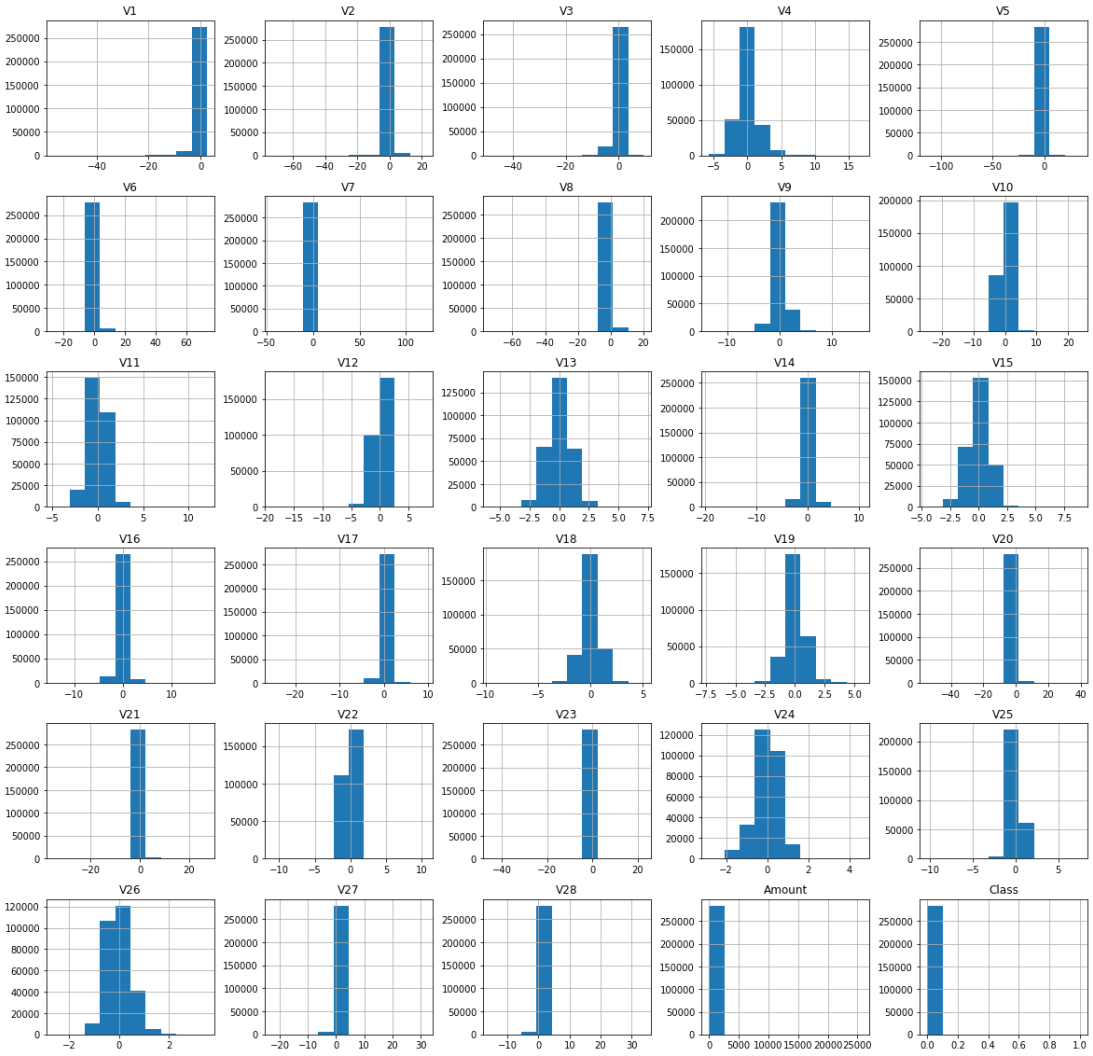

Figure: Distribution of the dataset

We can see that the whole dataset is highly imbalanced, for 284315 legit transactions, there are 492 fake transactions. To balance out the data we took a sample from the legit dataset randomly. And number of the sample data is same as the fraud dataset.

```
num_of_fraud_transaction = fraud.value_counts().sum()
legit_sample = legit.sample(n=num_of_fraud_transaction, random_state = 3)
```

That resulted a dataset of total 984 (=492+492)

After such, we took only 20% of that existing data in our implementation to test and the rest 80% we trained our algorithm.

```
X_train, X_test, Y_train, Y_test = train_test_split(X, Y, test_size = 0.2, stratify = Y, random_state = 3)
print(X.shape,X_train.shape,X_test.shape)
```

```
(984, 29) (787, 29) (197, 29)
```

Then we used logistic regression to model out the probability of each transaction whether being fraud or genuine. From that, we implemented cross validation, which resulted a mean value of 0.92.

Experimentation:

In this section, we experimented our set of data and trained the algorithm in our own implemented ways as mentioned before. In this experiment, after that we trained with the data, accuracy was 94%, but when we implemented the new data, the accuracy resulted 93%.

```
# accuracy score on testing data
X_test_prediction = model.predict(X_test)
testing_data_accuracy = accuracy_score(X_test_prediction, Y_test)
print("Accuracy on training data: ", testing_data_accuracy)
```

```
Accuracy on training data:  0.9390862944162437
```

Figure: Accuracy score on testing data

After that, we used precision, recall and f1 score, for class 0 we got 0.92 and for class 1 we got 0.96. The recalled value for class 0 is 0.96 and on the other hand for class 1 it is 0.92 . Finally to be mentioned. The f1-score remained same for both classes, which is 0.94.

```
print(classification_report(Y_test,X_test_prediction))
```

```
              precision    recall  f1-score   support

           0       0.92      0.96      0.94        99
           1       0.96      0.92      0.94        98

    accuracy                           0.94       197
   macro avg       0.94      0.94      0.94       197
weighted avg       0.94      0.94      0.94       197
```

Figure: precision, recall and f1-score report of the testing data.

Conclusion:

Credit card system is most vulnerable for frauds. These credit card frauds costs financial companies and consumers a very huge amount of money annually, fraudsters always try to find new methods and tricks to commit these illegal and outlaw actions. Online transaction fraud detection is most challenging issue for banks and financial companies. So it is much essential for banks and financial companies to have efficient fraud detection systems to reduce their losses due to these credit card fraud transactions.

In our approach, we built an efficient algorithm to detect and predict that fraud before it even happened. Our work concluded a good precision score. With the chances of credit card frauds are increasing massively with the increase in usage of credit cards for transactions, our research may help those financial institution to prevent a disastrous scenario.