University of Missouri-Kansas City

CS5590/490 - Python and Deep Learning Programming Course

Project: Wells Fargo Hackathon - Credit Card Fraud Detection

Team: Fantastic Four

Author(s):	*Role:*
Sai Shantan Goli	Team Member
Prudhvidhar Reddy Katta	Team Member
Girish Naga Vardhan	Team Member
Vinith Kumar Chelupati	Team Member

Faculty Advisers: Yugyung Lee , Ahmed Albishri , Saeed Al-Qarni

December 13, 2021

Contents

Abstract

As digital payments continue to expand across all demographics, research shows that older adults are showing the biggest uptick in adoption during the 2020/21 period due to the pandemic. Currently, digital payment data is not analyzed specifically under the vulnerable (elder and dependent adult) financial exploitation lens. Banks are required to report elder financial abuse but, unless a customer reports fraud and files a claim, financial abuse can go undetected and repeated fraud via digital payments can continue without the banks' knowledge. Without detection models, a large amount of fraud is left unreported by consumers and elder and vulnerable adult populations will be at greater risk of being targeted and losing savings to fraudulent payments.

Banks need better methods to help protect elder and vulnerable adults against fraud in the digital payments landscape. Predictive modeling may also be applied in some form to alert consumers and bankers in advance of a fraud attempt and potentially pre-empty certain transactions and monetary losses. As the older adult segment continues to adopt digital technology, including digital payments, banks need better ways to predict and analyze transaction data to detect high risk payment patterns or transaction attributes that signal high risk for fraud, especially for older and vulnerable adult customers, which could be targeted by scammers.

To overcome this problem we uses the Machine Learning and Deep Learning techniques to identify the fraudulent transactions. With this we can mark the transaction as fraud as soon it has happened and inform the user about it and take necessary actions about it.

1 Introduction

With the increase of Digital Payments the fraudulent transactions increased a lot during the pandemic.Identifying the fraud transaction as soon as possible will benefit both the user and the bank. By using Machine Learning and Deep Learning we identify the transaction whether it is fraud or not.

This DataSet is part of Wells Fargo Hackathon, in this we were asked to analyze the columns and decide whether a transaction is fraud or not.This Hacakthons data is artificially generated using Conditional GAN. With this technique they created a new dataset using the original dataset.

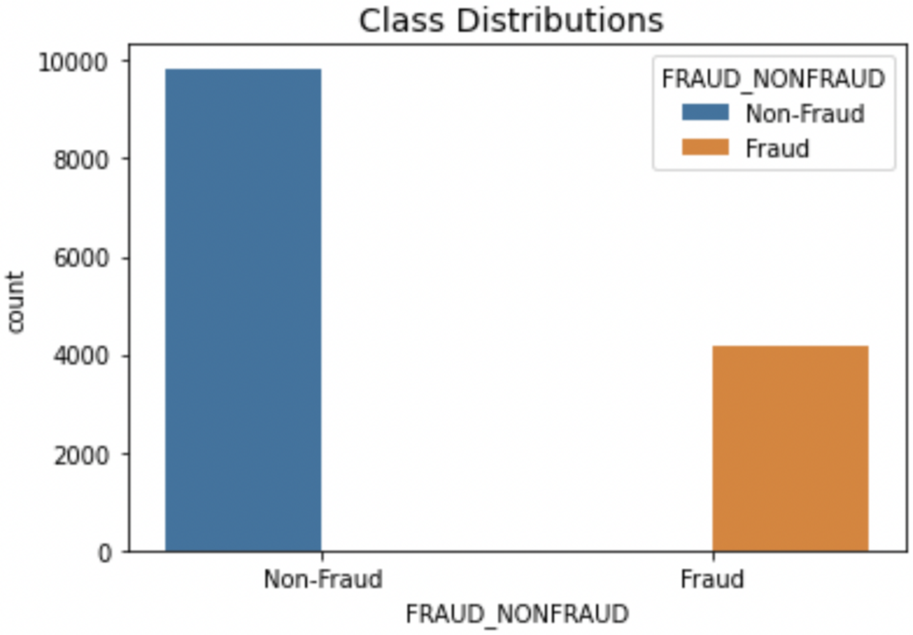
2 Problem statement

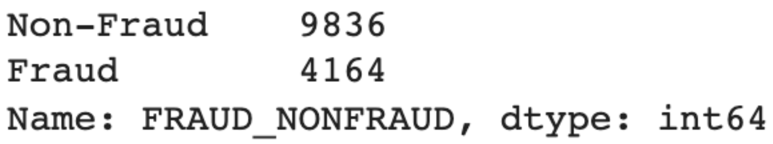
The Problem is here to Identify whether the transaction is fraud or not by using the different Data Science techniques.As this is whole new data generated artificially the co-relations between the variables is not good.We need to figure out the ways to find co-relations.

3 Related work

In the Market there are many Credit Card Fraud Detection models, but the data that was used was balanced and there were good co-relations between features. As the data changes the complete modelling part will change because model is completely dependent on the data we use. There are very good models for this type of problems.

4 Datasets





```
Non-Fraud    9836
Fraud        4164
Name: FRAUD_NONFRAUD, dtype: int64
```

As a part of hackathon we were given training file directly.It contains around 14000 rows. The target column is encoded with Fraud and Non-Fraud cases. From the above picture we can confirm that the data is highly imbalanced. The Non-Fraud cases are almost double times more than the fraud cases. With this imbalance in data training a model will not be good. And also the co-relation between the columns is not that good, so we need to find a way to decide which columns are most important and as there are around 24 columns we need to find which columns are useful and which columns are not useful. With the domain experience we can decide whether a given column is important or not. And also if there is strong co-relation between the target column and the normal column then we can say that the given column is much more important than the other columns.If the co-relation is very low or in negative quantity then we can say that these values/features are not that important.And also there are many null values in different columns in the training file. If we remove all the null values the dataset will come decrease by a huge number, so we need to find an alternative way to overcome this null value issue. Replacing all the null values with NONE might help us.But in this some are string values and some are integer fields, so we need to check what all we can replace and what we cannot.And also there are many not necessary fields like phone number update time, App last update and etc. so we can remove them as they are not useful in any ways. And also there is timestamp column at which the transaction occurred, this is also not useful. We will remove this column also.

5 Problem Solution

For this problem we introduced a solution that is mostly focused on the imbalance of data. As there are only 2 target categories we need to find a way such that we can sample the low category cases in the correct number so that it can be used to train the model. As there are many ways to overcome this issue, we can undersample the category which has more number of rows and take equal amount as the low category type, so that the data will be balanced with the both types.Then we will pass the data to the normal Machine learning algorithms and Deep learning algorithms to get the good training and testing accuracy.We can select the different types of algorithms and decide whether a algorithm is suitable or not for this type of data. usually Logistic regression will work better with this kind of data. But we cannot be sure which algorithm will work and which will not work.

6 Data Pre-Processing

As we are given the training file directly we load the excel file using the pd.read_excel() method that is available in the pandas dataframe .To this we will pass the file path of the input file. As our data contains of null values and different data types and also imbalance of data. we need to figure out how we can overcome these issues in the model building part.

First we will focus on the null values that are present in the training file. As there are many null values in the code we cannot remove all those rows directly, by doing this we will decrease the size of the training data by a lot extent. Instead of removing all the null value rows we will replace the null value cell with the NONE value. We can do this by pandas fill.na() method i.e df.fillna('None'), this method is a available in the pandas library. With this all the null values will be replaced by the NONE value in the data frame.

In the next step we will try to understand each and every feature and remove the features that are not useful in the Data frame, we can decide whether a feature is useful or not by calculating the co-relation between the features or by the expertise in the domain knowledge. With the understanding of different Credit Card Data sets we decided which columns can be used and which columns are not that useful in the Model construction.Some of the columns removed are date at which last phone number update etc.

As our data frame contains different type of data types for the each column , we need to convert all of them to Int or Float category.To do this we need to first find out what all columns are of the type string in the given data frame.so to do this we need to find out each column datatype in the data frame. We do this by iterating over data frame and finding out datatype of each column, then we store them in a list.We use LabelEncoder() method from the sklearn to encode all the string category to the int values. First we declare the label encoder and then apply the function le.fit_transform() method on the each of the string column that we got before.With this step we will encode all the string features to int values so that we can pass it to the model.

As the values of each columns are in different ranges, the model will be biased towards the high values , so we need to scale them to one range. We can do this by standard scaling method, this scales all the values between same range so that the model will not be biased towards the high values.This process can be done by StandardScaler() method from the sklearn method.We fit the method fit_transform() on the data to convert all the values to same range.So that we will not face any issues while training the model.

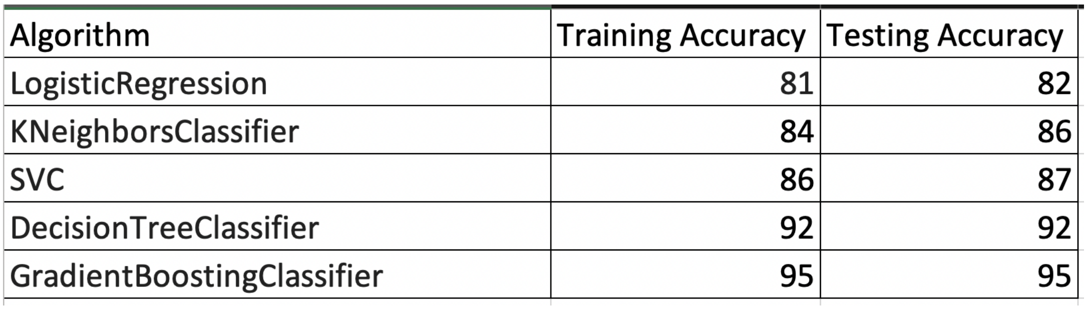
As the part of next step we need to split the data into training and testing so that we can check how our model is performing on the unseen data. we can do this by using sckit_learn train_test_split() method, we need to pass the features and the target column with specifying the test_size and also we can add random_state and stratify so that it can be stratified based on the target columns.so that there will be proper mix of both the classes.This method contains many other parameters like shuffle , train_size. The input for this should be arrays of the features from the dataframe.

7 Model Building

The most important part of the project is the modelling part, as our data is highly imbalanced in nature, we thought of using original data as it is for the initial step of training so that we can use this results as the benchmark for the comparison with the techniques we use for the model regarding the balance of data.Logistic regression will be very good for this type of classification. But for this data it was not that good with the accuracy part.We tried different algorithms as part of machine learning.

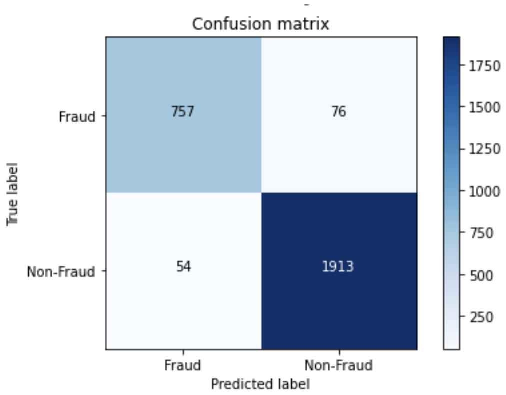
7.1 Machine Learning

As the data is very imbalanced the machine learning techniques were not that good against the data.But xgboost algorithm was much better compared to all the machine learning algorithms.As there was imbalance in the data we had many wrong predictions in the data.The following figures shows the training accuracy and testing accuracy



Algorithm	Training Accuracy	Testing Accuracy
LogisticRegression	81	82
KNeighborsClassifier	84	86
SVC	86	87
DecisionTreeClassifier	92	92
GradientBoostingClassifier	95	95

This Accuracy is acheived by the cross validation method cross_val_score(classifier, X_train, y_train, cv=5). This method chooses the optimum values for the training method . It chooses one subset use that for training and create altogether new set to train again. This happens for 5 times as mentioned in the function.From the given picture we can say that xgboost performs well compared to all other algorithms from the accuracy.



We can say that there are good number of miss classifications happening still, we will check with any other method if we can overcome these errors.

7.1.1 Under-Sampling

This is a technique where you take the under sampled class and count the number of values present in it and count the equal number of classes from the over sampled classes.So that both the types will be in equal amount.So that the model will not be biased towards the one class.But as our class are equal in number and the number of non fraud cases are more in number in original data set than the equal data set.Because of this we can say that our accuracy dropped much when we check it on the original data set.

7.2 Deep Learning

To check how deep learning works on our data we used basic neural network to do the classification task of both the category.First we did scaling to bring all the features to equal range as deep learning is highly prone to the high values. Then we did pass them to the basic neural network to check the accuracy of it. The training accuracy was around 82.23% and testing accuracy around 83.2%.

7.2.1 Under-Sampling

In this as discussed before we did take the equal amount of both the categories and merged into a single data frame, we will scale all the category values to one range and pass it on to the neural network model so that predictions can work properly without no bias.The training and testing accuracy with this type were 87.12% and 83.97% respectively.

8 Post-processing

Once we are done with this we used Flask API to host our application so we that can normally pass the values to test the model, We used two end points to train and test. When the hit will come to end point i.e training the training will start and the hit comes to another endpoint the testing happens and we will return back the respective category of the predicted column.We also used ngrok so that the testing can be done from any where in the world.

9 Applications

The major application of this project is the whole Banking sector.They can use this as a base model to check for the fraudulent transactions.

10 Future Work

In the Future work we can improve the accuracy of the model and also we can use the other methods that deals with the im balance of data. And we can add some of the pre check conditions in testing endpoint so that we can check the inputs given in our training and testing file so that we will not waste the time.

11 References

1.https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.train_test_split.html

2.https://xgboost.readthedocs.io/en/stable/

3.https://machinelearningmastery.com/undersampling-algorithms-for-imbalanced-classi

4.https://www.kaggle.com/joparga3/in-depth-skewed-data-classif-93-recall-acc-now

5.https://github.com/SimarjotKaur/Credit-Card-Fraud-Detection/blob/master/credit-card-fraud-detection-using-neural-networks.ipynb

6.https://keras.io/

7.https://www.kaggle.com/janiobachmann/credit-fraud-dealing-with-imbalanced-dataset