MACHINE LEARNING ENGINEER NANODEGREE

CAPSTONE PROJECT

TITLE: CREDIT CARD FRAUD DETECTION

NUWAYIR ALSUBAIE

DEFINITION

Project Overview

Credit card is a small thin plastic or fiber card that contains information about the person such as a picture or signature and person named on it to charge purchases and service to his linked account charges for which will be debited regularly. Nowadays card information is read by ATM's, swiping machines, store readers, bank and online transactions. Each card has a unique card number which is very important, its security mainly relies on physical security of the card and also privacy of the credit card number . Recently, with the increase in websites that support payment on the Internet, it was required to find ways to detect fraud in cards on the Internet, and in this project, I will address that. The Credit Card Fraud Detection Problem includes modeling past credit card transactions with the knowledge of the ones that turned out to be a fraud. This model is used to identify whether a new transaction is fraudulent or not. Our aim here is to detect 100% of the fraudulent transactions while minimizing the incorrect fraud classifications.

Problem Statement

There is a rapid increase in the credit card transaction which has led to substantial growth in fraudulent cases. Many data mining and statistical methods are used to detect fraud. Many fraud detection techniques are implemented using artificial intelligence, pattern matching. Detection of fraud using efficient and secure methods are very important. Credit card frauds are increasing heavily because of fraud financial loss is increasing drastically. Nowadays Internet or online transactions are growing as new technology is coming day by day. In these transactions the Credit card holds the maximum share. In 2018 Credit card fraud losses in London estimated US dollar 844.8 million. To reduce these losses prevention or detection of fraud must be done. There are different types of frauds occurring as technology is growing rapidly. So there are many machine algorithms used to detect fraud, nowadays and in this project I used two of them, which is logistic regression algorithm and Random forest algorithm .

Metrics

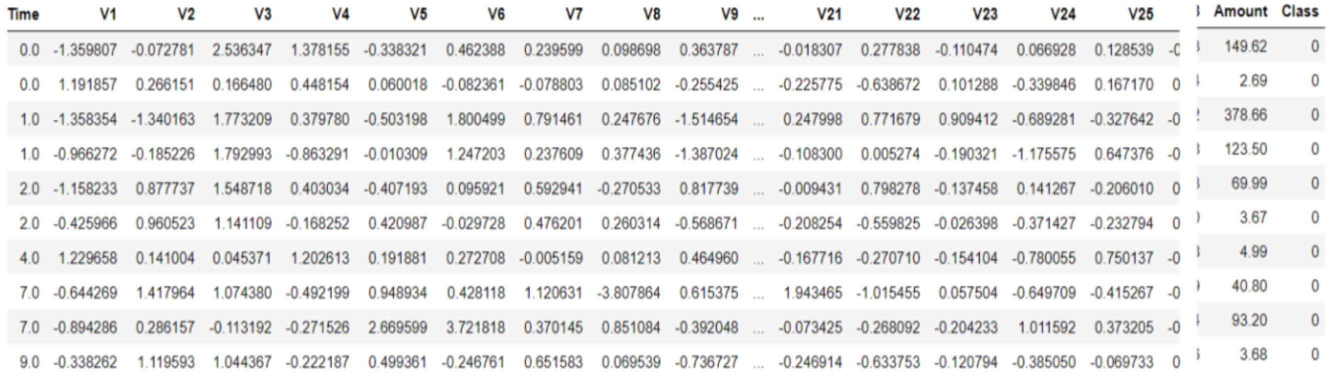
It is important for any model or project to have evaluation metrics so to measure the strength of the models, we used many metrics of the performance for the Classification Problems such as accuracy, f1-score, recall, precision, and confusion metrics for both the training and testing to make sure that we didn't have overfitting nor underfitting. As our goal is to determine whether the transactions are fraudulent or genuine, and also the benchmark was a basic comparison in our models, as it is considered a good reference for learning from the negatives of old models and improving them in our Model.

.

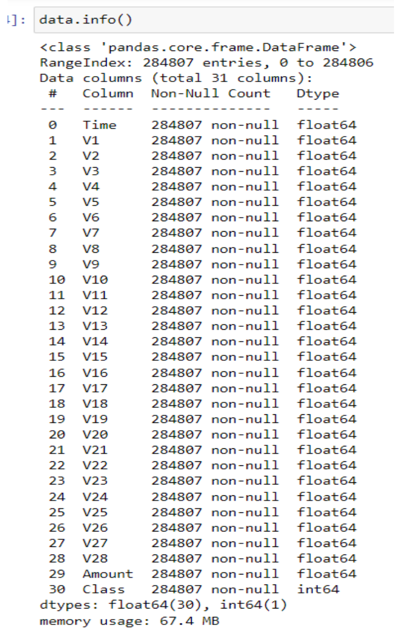
ANALYSIS

Data Exploration

The dataset obtained from Kaggle and its data for two days transactions so as we can observe the number of transactions is 280987 (rows) and for the feature, we ended up knowing that many of features are not always the same and its depend on the different aspects related to the region and the banks. For our dataset, we have 30features (columns) for each transaction made during these two days. Data is not balanced because there are fewer fraud cases as compared to huge transaction data. Dataset has converted Principal Component Analysis (PCA) transformation and contains only numeric values. Due to privacy and confidentiality, much background information is not provided, only PCA transformed data is given. Only time and amount are not transformed to PCA all other given values v1, v2, v3 . . . .. v28 are PCA transformed numeric values. Feature class contains 1 for fraud and 0 for normal transactions.



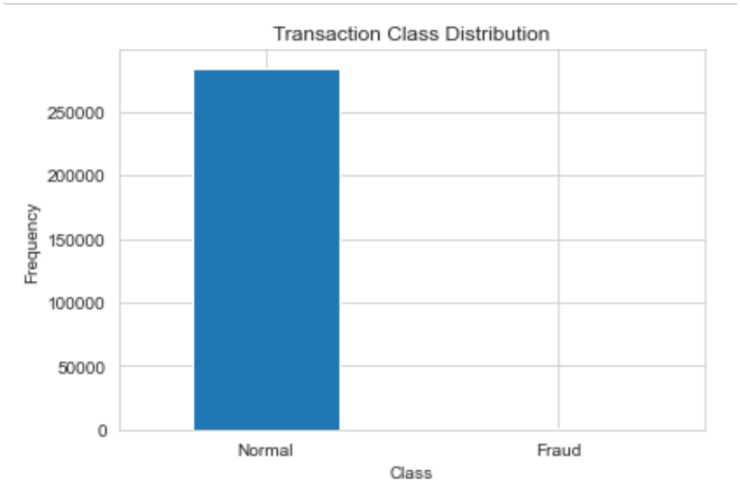
Time	V1	V2	V3	V4	V5	V6	V7	V8	V9	...	V21	V22	V23	V24	V25		Amount	Class
0.0	-1.359807	-0.072781	2.536347	1.378155	-0.338321	0.462388	0.239599	0.098698	0.363787	...	-0.018307	0.277838	-0.110474	0.066928	0.128539	-0	149.62	0
0.0	1.191857	0.266151	0.166480	0.448154	0.060018	-0.082361	-0.078803	0.085102	-0.255425	...	-0.225775	-0.638672	0.101288	-0.339846	0.167170	0	2.69	0
1.0	-1.358354	-1.340163	1.773209	0.379780	-0.503198	1.800499	0.791461	0.247676	-1.514654	...	0.247998	0.771679	0.909412	-0.689281	-0.327642	-0	378.66	0
1.0	-0.966272	-0.185226	1.792993	-0.863291	-0.010309	1.247203	0.237609	0.377436	-1.387024	...	-0.108300	0.005274	-0.190321	-1.175575	0.647376	-0	123.50	0
2.0	-1.158233	0.877737	1.548718	0.403034	-0.407193	0.095921	0.592941	-0.270533	0.817739	...	-0.009431	0.798278	-0.137458	0.141267	-0.206010	0	69.99	0
2.0	-0.425966	0.960523	1.141109	-0.168252	0.420987	-0.029728	0.476201	0.260314	-0.568671	...	-0.208254	-0.559825	-0.026398	-0.371427	-0.232794	0	3.67	0
4.0	1.229658	0.141004	0.045371	1.202613	0.191881	0.272708	-0.005159	0.081213	0.464960	...	-0.167716	-0.270710	-0.154104	-0.780055	0.750137	-0	4.99	0
7.0	-0.644269	1.417964	1.074380	-0.492199	0.948934	0.428118	1.120631	-3.807864	0.615375	...	1.943465	-1.015455	0.057504	-0.649709	-0.415267	-0	40.80	0
7.0	-0.894286	0.286157	-0.113192	-0.271526	2.669599	3.721818	0.370145	0.851084	-0.392048	...	-0.073425	-0.268092	-0.204233	1.011592	0.373205	-0	93.20	0
9.0	-0.338262	1.119593	1.044367	-0.222187	0.499361	-0.246761	0.651583	0.069539	-0.736727	...	-0.246914	-0.633753	-0.120794	-0.385050	-0.069733	0	3.68	0



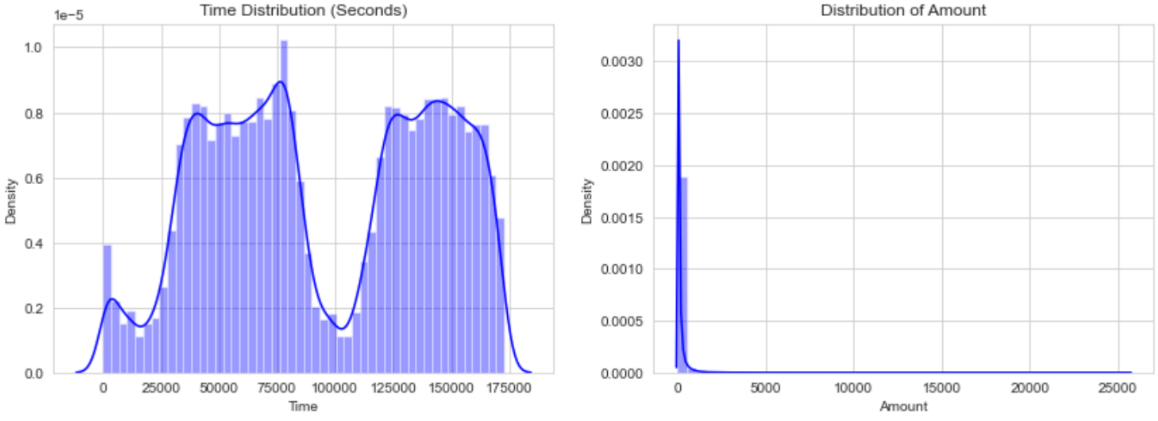
```
]: data.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 284807 entries, 0 to 284806
Data columns (total 31 columns):
 #   Column  Non-Null Count   Dtype
---  ------  --------------   -----
 0   Time    284807 non-null  float64
 1   V1      284807 non-null  float64
 2   V2      284807 non-null  float64
 3   V3      284807 non-null  float64
 4   V4      284807 non-null  float64
 5   V5      284807 non-null  float64
 6   V6      284807 non-null  float64
 7   V7      284807 non-null  float64
 8   V8      284807 non-null  float64
 9   V9      284807 non-null  float64
 10  V10     284807 non-null  float64
 11  V11     284807 non-null  float64
 12  V12     284807 non-null  float64
 13  V13     284807 non-null  float64
 14  V14     284807 non-null  float64
 15  V15     284807 non-null  float64
 16  V16     284807 non-null  float64
 17  V17     284807 non-null  float64
 18  V18     284807 non-null  float64
 19  V19     284807 non-null  float64
 20  V20     284807 non-null  float64
 21  V21     284807 non-null  float64
 22  V22     284807 non-null  float64
 23  V23     284807 non-null  float64
 24  V24     284807 non-null  float64
 25  V25     284807 non-null  float64
 26  V26     284807 non-null  float64
 27  V27     284807 non-null  float64
 28  V28     284807 non-null  float64
 29  Amount  284807 non-null  float64
 30  Class   284807 non-null  int64
dtypes: float64(30), int64(1)
memory usage: 67.4 MB
```

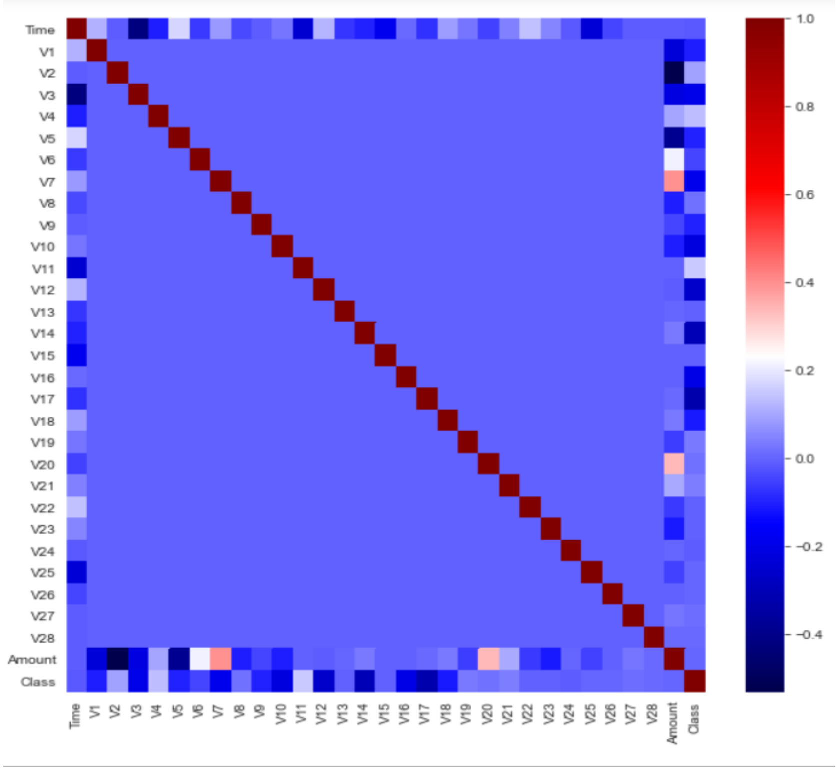
Exploratory Visualization



Notice how imbalanced is our original dataset. Most of the transactions are non-fraud. If we use this dataframe as the base for our predictive models and analysis we might get a lot of errors and our algorithms will probably overfit since it will "assume" that most transactions are not fraud. But we don't want our model to assume, we want our model to detect patterns that give signs of fraud.



We can observe that the most fraud detections happen with a small amount of transactions. However, the time of fraud transactions are not on a specific time frame. It can happen any time depending on this dataset. Also the time here is measured by seconds so it's quite complex to focus on a specific time frame .



A heat map is a two-dimensional representation of data in which values are represented by colors. Correlation Heat map is a two dimensional plot of the amount of correlation (measure of dependence) between variables represented by colors. The varying intensity of color represents the measure of correlation. Correlation is a measure of linear relationship between two variables. Correlation between two variables.Correlation values ranges from -1 to +1. Highest correlations come from:Time & V3 (-0.42),Amount & V2 (-0.53),Amount & V4 (0.4).While these correlations are high, I don't expect it to run the risk of multicollinearity.The correlation matrix shows also that none of the V1 to V28 PCA components have any correlation to each other however if we observe Class has some form positive and negative correlations with the V components but has no correlation with Time and Amount.

Algorithms and Techniques

There's many classification algorithms that can be used in classification problems but in this project we focus on two of them : logistic regression and random forest . Before we go any step forward we must solve our unbalanced dataset , let’s understand why the unbalanced dataset affects model performance . One way the imbalance may affect our Machine Learning algorithm is when our algorithm completely ignores the minority class. The reason this is an issue is because the minority class is often the class that we are most interested in.When building a classifier to classify fraudulent and non-fraudulent transactions from various observations, the data is likely to have more non-fraudulent transactions than that of fraud. It would be very worrying if we had an equal amount of fraudulent transactions as non-fraud.

Why does class imbalance affect model performance?

- In general, we want to maximize the recall while capping FPR (False Positive Rate), but you can classify a lot of charges wrong and still maintain a low FPR because you have a large number of true negatives.
- This is conducive to picking a relatively low threshold, which results in the high recall but extremely low precision

The technique we used to handle this imbalance class is an oversampling technique is a bias to select more samples from one class than from another, to compensate for an imbalance that is either already present in the data.

I was thinking about using SVM but after we worked we realize that we didn't need it in this project

Logistic regression

Logistic regression works with sigmoid function because the sigmoid function can be used to classify the output of a dependent feature and it uses the probability for classification of the dependent feature. This algorithm works well with less dataset because of the use of sigmoid function. If the value of the sigmoid function is greater than 0.5 the output will be 1 if the output of the sigmoid function is less than 0.5 then the output is considered as the 0. But this sigmoid function is not suitable for deep learning because of deep learning, when we backtracking from the output to input we have to update the weights to minimize the error in weight update. We have to differentiate the sigmoid activation function in the middle layer neuron then results in the value of 0.25 this will affect the accuracy of the module in deep learning.

Random Forest

The Random Forest works by using a lot of decision trees and taking the majority predictions of the ensemble to get a final result. The random forest randomly selects the features that are independent variables and also randomly selects the rows by row sampling and the number of the decision tree can be determined by using hyperparameter optimization. in this project, I tried to use the tuning technique to choose the number of estimators in a more make sense way but unfortunately, my laptop didn't help me because it takes time and needs more RAM storage than is mine. For classification problem statements the output is the maximum occurrence output from each decision tree model inside the random forest. This is one of the widely used machine learning algorithms in real-world scenarios and deployed models. And in most of the Kaggle computation challenges, this algorithm is used to solve the problem statement. In this project, I picked 100 as the number of estimators and its work is really good.

After building models , the F-score , accuracy ,recall, precision, and confusion metrics have been calculated for each model and the Random Forest has registered the highest score .Then the logistics regression

Benchmark

Building a support vector machine model with a linear kernel has register accuracy 97.5% . Our goal is to build a supervised model with a better result than SVM [6]. We pass this result by 100% using Random Forest .

METHODOLOGY

Data Preprocessing

The procedure which we followed to predict the result is understanding problem statement and data by performing statistical analysis and visualization then checking whether the data is balance or not, In this data set the data is imbalanced, balanced by using oversampling, then we use cross-validation technique to make sure that our model will not have overfitting nor underfitting, by taking the accuracy of training then compared with testing accuracy apply two different machine learning algorithm both of them give us different results.

Implementation

The following steps summarize the implementation of the model :

1. **Load data and import all necessary libraries**
2. **Explore the data**
    a. Explore data type for each column
    b. Show the statistics for each column such as std , mean
    c. Check if there is null values or not
    d. Discover the class using histogram
    e. Labeled Class column as 0 (non fraud) and 1(fraud)
    f. Our dataset are unbalanced so we need to fix that in next step
    g. Visualize our dataset based on time distribution and amount distribution to find any patterns
    h. Apply heatmap to find correlation between the features
3. **Data Preprocessing**

a. Unbalanced data affect the accuracy of any model so we need to handle the unbalanced data to make sure about the final model accuracy , there are many techniques to handle the unbalanced data in this project we use oversampling technique

b. Standardized , The main idea is to normalize/standardize, will transform your data such that its distribution will have a mean value 0 and standard deviation of 1. i.e. $\mu = 0$ and $\sigma = 1$ your features/variables/columns of X, individually, before applying any machine learning model.Variables that are measured at different scales do not contribute equally to the model fitting & model learned function and might end up creating a bias. Thus, to deal with this potential problem, feature-wise standardized (μ=0, σ=1) is usually used prior to model fitting.

c. Split data, we split our data into training and testing sets to train and evaluate our model

4. **Modeling**

Our task in this project is to classify the transactions so we will choose some supervised learning algorithms

a. Develop Logistic regression
b. Develop Random Forest (RF).

5. **Evaluation**

a. Calculate training accuracy for each model .
b. Calculate F1-score, precision and recall for result of each model .
c. Calculate Confusion Matrix for each model

6. **Visualize and calculate feature importance for each model .**

One of the issues we couldn't make the auto fine tuning of parameters for each model because the limitation of our hardware resources

REFINEMENT

In the logistic regression model we play with parameter C to improve accuracy of the model where C refers to Inverse of regularization strength ,smaller values specify stronger regularization. while the model for RF we play with the number of estimator (n_estimators), n_estimators refer to the number of trees in the forest

Algorithm	Parameter(C)	F1-score	Recall	Precision
Logistic regression	Not specified	0.98	0.98	0.98
	10000	0.98	0.98	0.98

Algorithm	n_estimators	F1-score	Recall	Precision
Random Forest	10	1.0	1.0	1.0
	100	1.0	1.0	1.0

After applying two different classification algorithms and observing the model performance , the random forest algorithm obtained the highest scores in all of the f1-score, precision , and recall measures .Then the random forest parameters (n_estimators) have been used to tune over the model and the final scores was

RESULTS

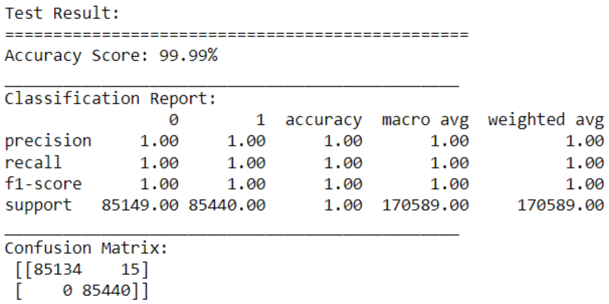
Model evaluation and Validation

The final random forest model was selected according to the evaluation metrics, it yielded the highest score after many trials. The number of estimators was 100 , the model score was 1.0 , 1.0, 1.0 and1.0 for F1-score, precision and recall.

To overcome the unbalanced class the oversampling technique and cross-validation has been used then the training accuracy has been calculated . Also the F1-score, precision and recall has been used as evaluation metrics and calculate the final score average. Hence, the model is robust and can be trusted to make accurate predictions. However, if still can be further improved and train the model more time and deploy it on segmaker .

Justification

The final solution provides better results than the benchmark. The benchmark resulted in a recall 0.6 and precision 0.7 while the final model was 1.0, 1.0, 1.0 for F1-score , precision, and recall. respectively. Hence, the final model and the implemented solution is significant enough to have adequately solved the problem.

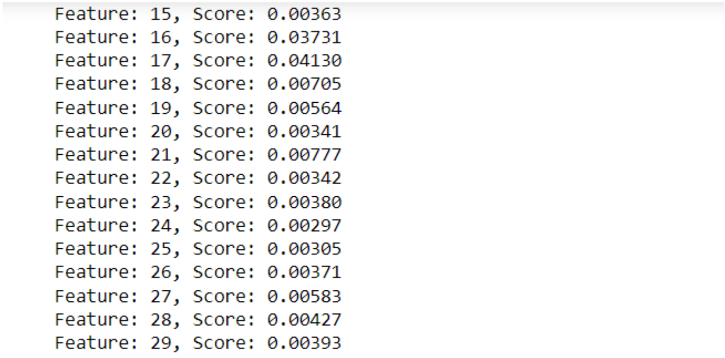


```
Test Result:
================================================
Accuracy Score: 99.99%
_______________________________________________
Classification Report:
                 0        1  accuracy  macro avg  weighted avg
precision     1.00     1.00      1.00       1.00          1.00
recall        1.00     1.00      1.00       1.00          1.00
f1-score      1.00     1.00      1.00       1.00          1.00
support   85149.00 85440.00      1.00  170589.00     170589.00
_______________________________________________
Confusion Matrix:
 [[85134    15]
 [    0 85440]]
```

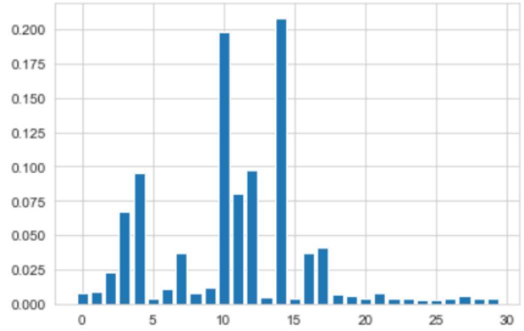
CONCLUSION

Credit card fraud is the most common problem resulting in loss of a lot of money for people and loss for some banks and credit card companies. This project want to help the peoples from their wealth loss and also for the banked company and trying to develop the model which more efficiently separate the fraud and fraud less transaction by using the time and amount feature in data set given in the Kegel. First we build the model using some machine learning algorithms such as logistic Regression, Random Forest , These all are supervised machine learning algorithms in machine learning.

Additional -Form Visualization



```
Feature: 15, Score: 0.00363
Feature: 16, Score: 0.03731
Feature: 17, Score: 0.04130
Feature: 18, Score: 0.00705
Feature: 19, Score: 0.00564
Feature: 20, Score: 0.00341
Feature: 21, Score: 0.00777
Feature: 22, Score: 0.00342
Feature: 23, Score: 0.00380
Feature: 24, Score: 0.00297
Feature: 25, Score: 0.00305
Feature: 26, Score: 0.00371
Feature: 27, Score: 0.00583
Feature: 28, Score: 0.00427
Feature: 29, Score: 0.00393
```



This figure Determines the features that had the greatest impact on the prediction after the final model was created , we do this to have more information for further improvement .

References

1. https://machinelearningmastery.com/calculate-feature-importance-with-python/

2. https://stackoverflow.com/questions/20107570/removing-index-column-in-pandas-when-reading-a-csv
3. https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html

4. https://projecteuclid.org/journals/statistical-science/volume-17/issue-3/Statistical-Fraud-Detection-A-Review/10.1214/ss/1042727940.full

5. https://www.geeksforgeeks.org/pandas-how-to-shuffle-a-dataframe-rows/

6. https://projecteuclid.org/journals/statistical-science/volume-17/issue-3/Statistical-Fraud-Detection-A-Review/10.1214/ss/1042727940.full

7. https://scikit-learn.org/stable/modules/model_evaluation.html

8. https://www.researchgate.net/publication/343632766_Credit_Card_Fraud_Detection_using_Machine_Learning_Algorithms