Credit card Fraud detection: A Machine learning Approach

Abstract:

The report's primary objective is to comprehensively overview the process, academic research, and impact of the machine learning model that detects fraudulent transactions. It discusses the role and impact of the data scientist in creating the machine learning model and clarifies the model and techniques used and ignored while building it.

In conclusion, the report addresses why the random forest algorithm is the best for detecting fraudulent transactions compared to XGboost and Logistic regression. It highlights the challenges faced during the model creation and explains the strategic recommendations to improve the model and overcome challenges with relevant academic literature.

Introduction:

Billions of dollars are lost every year due to fraudulent credit card transactions. The use of credit cards is more common in the modern era, and the development of new technology and payment methods provides additional ways for criminals to commit fraud. The financial loss due to fraud affects merchants, banks, and individual customers.

As data scientists, we analyse the data and algorithms to develop a machine-learning model that detects fraudulent transactions. We used a dataset containing European cardholders' credit card transactions in 2023 to perform exploratory data analysis and build and train the machine learning model. The project intends to detect fraudulent transactions by incorporating data science and machine learning to improve fraud detection systems further when online transactions are more prevalent.

Role exploration:

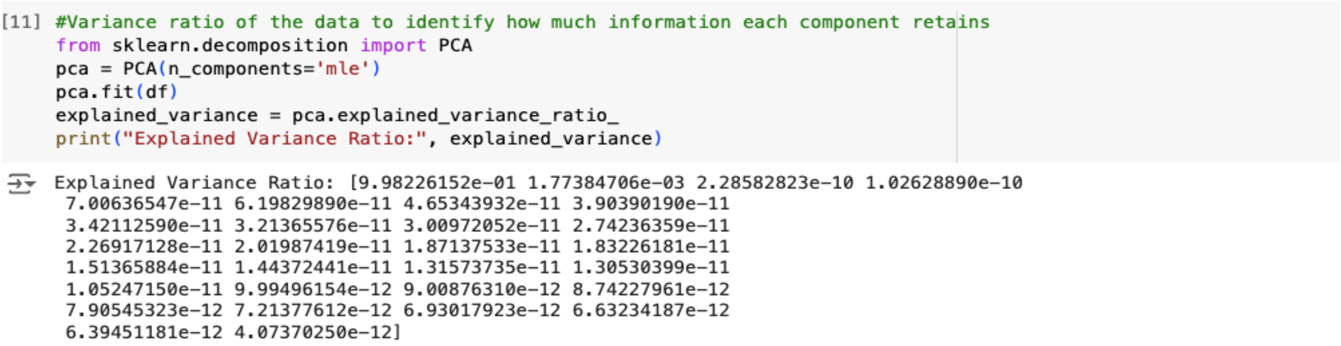
The data scientist role is crucial because it provides technical and analytical insights for building effective fraud-detecting systems. As data scientists, I worked on the random forest classifier for our machine learning model since the random forest algorithm is robust to outliers and identifies complex and hidden patterns that are usually common in fraudulent transactions. It is necessary that the data is clean and aligns with the business context as the model's performance relies on the data it is trained upon, so I performed exploratory data analysis on the dataset, which was obtained from Kaggle to identify outliers, trends, spending patterns and correlations among the features. The "**challenges and methodologies"** explain the approach and shows the significance of the data scientist role in building and improvising the model. The insights provided by our role help data engineers identify the threshold limit to remove outliers, find the class count to determine if the dataset is balanced or imbalanced and scale the "Amount" column to the other columns by standardisation. The insights also aided the data visualisation team in creating a correlation chart to identify highly correlative features to reduce dimensionality, bar charts to identify class distribution, and histograms to identify the skewness of the data in respective features to avoid any bias towards a feature. The technical insights from data scientists ensure the reliability of the analysis that aligns to identifying fraud in real life scenarios and help create visualisations that translate model's output to understandable insights for both technical and non-technical stakeholders. It also helps business strategists select random forest as the optimal model and provides strategic recommendations aligned with the objective of fraud detection.

Challenges and Methodologies:

We have used Logistic Regression, Random Forest, and XG Boost classification algorithms for our Machine-learning models and compared their effectiveness in identifying fraudulent transactions in the dataset. Below are the details of the challenges we encountered and the methods we used to counter them.

1. The credit card dataset was anonymised using principal component analysis (PCA) to protect sensitive information and ensure privacy.

- **Impact**: The anonymisation made it difficult to identify important information such as gender, age, spending pattern, location, and more. It risks the model's focus on specific data or features, eventually leading to poor generalisation and analysis of the new data in real-world applications.
- **Approach**: The PCA is a linear transformation technique that transforms data into sets of principal components, which are original feature's linear combination. As shown in the *figure 1*, we identified the variance ratio of every transformed feature to understand how much information each component retains.



```
[11] #Variance ratio of the data to identify how much information each component retains
     from sklearn.decomposition import PCA
     pca = PCA(n_components='mle')
     pca.fit(df)
     explained_variance = pca.explained_variance_ratio_
     print("Explained Variance Ratio:", explained_variance)
```

```
Explained Variance Ratio: [9.98226152e-01 1.77384706e-03 2.28582823e-10 1.02628890e-10
 7.00636547e-11 6.19829890e-11 4.65343932e-11 3.90390190e-11
 3.42112590e-11 3.21365576e-11 3.00972052e-11 2.74236359e-11
 2.26917128e-11 2.01987419e-11 1.87137533e-11 1.83226181e-11
 1.51365884e-11 1.44372441e-11 1.31573735e-11 1.30530399e-11
 1.05247150e-11 9.99496154e-12 9.00876310e-12 8.74227961e-12
 7.90545323e-12 7.21377612e-12 6.93017923e-12 6.63234187e-12
 6.39451181e-12 4.07370250e-12]
```

***Figure 1**: Code snippet of variance ratio*

2. We have created Pearson correlation heatmap that identifies multiple correlative coefficients among the components in the dataset, as shown in the *figure 2* below.

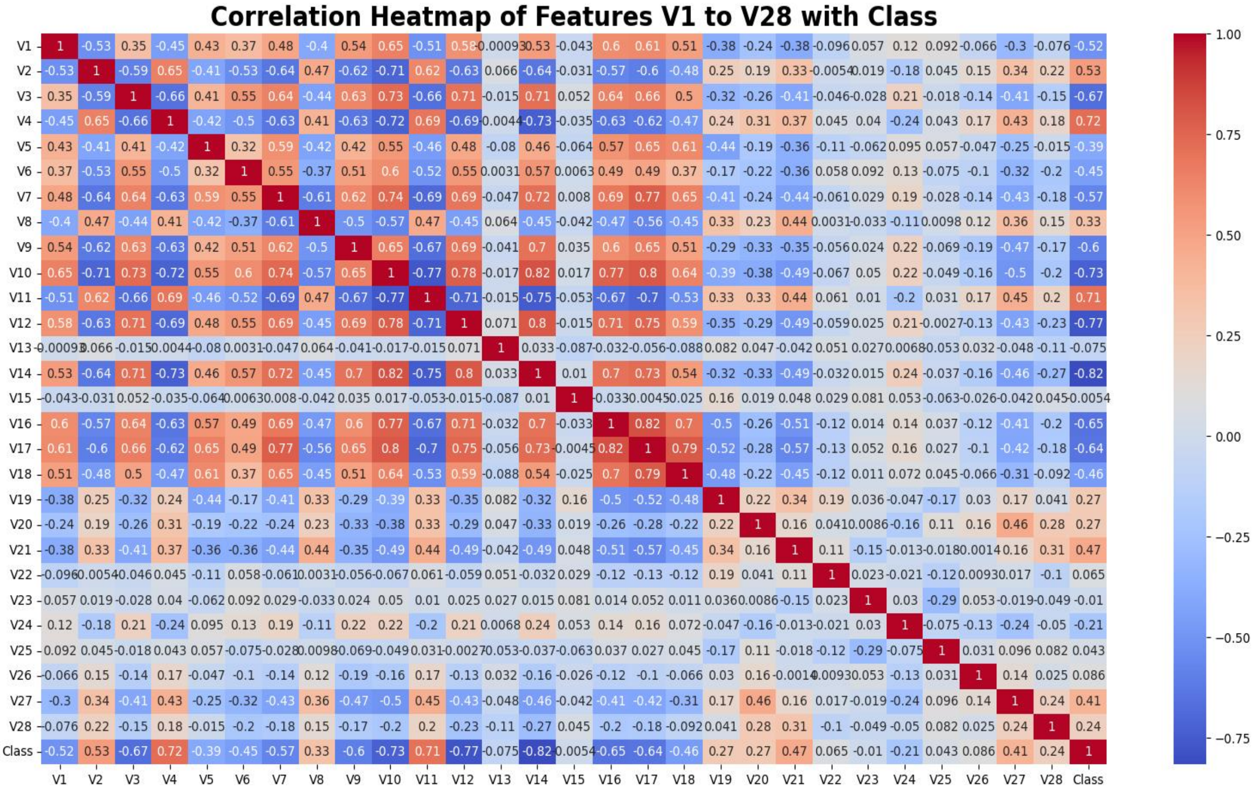


Figure 2: *Diagram of correlation heatmap*

- **Impact**: Some correlative features are often redundant and irrelevant to the factors contributing to fraudulent transactions. They risk overfitting, which causes the model to fail to generalise with new data.

- **Approach**: Since we cannot de-anonymise the dataset or reduce the dimensionality without considering the relevance and importance of the features, we have selected algorithms such as random forest and XG boost that can handle collinearity and multi-dimensionality.

3. The class distribution (legit or fraud) in our dataset was highly balanced as shown in the *figure 3*, but this will not be the case in real-life scenarios.

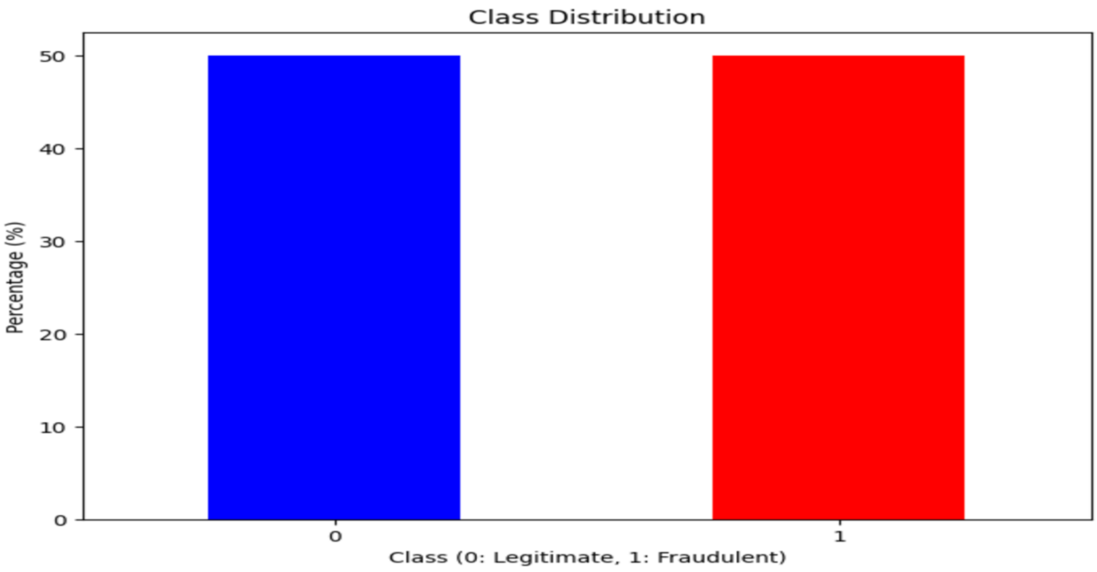
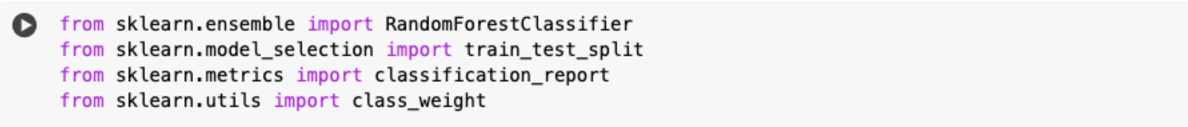


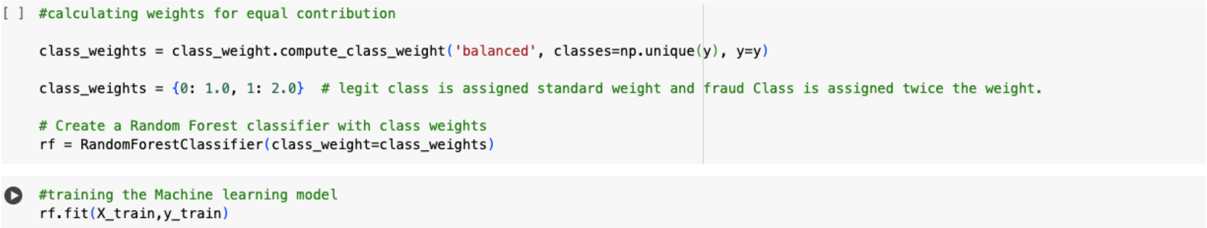
Figure 3: *Bar graph of the class distribution*

- **Impact:** Models trained on highly balanced datasets will allocate importance or focus on learning patterns of respective classes (o and 1). In real-life scenarios, we will have highly unbalanced data with <1% of fraudulent transactions; since the model trains on balanced data, it will fail to identify relatively more minor datasets, i.e. fraudulent transactions.

- **Approach:** Alternatively, we cannot train the model with an unbalanced dataset, as the model will be biased towards the more significant sample (legitimate transactions). So, we have employed the Class weighting method and assigned standard importance to legitimate transactions and double the importance to fraudulent transactions as shown in the *figure 4*. We can also use the synthetic minority over-sampling technique (SMOTE) to overcome the class imbalance. It creates synthetic instances of fraudulent transactions and a balanced dataset to train the model.



```
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import train_test_split
from sklearn.metrics import classification_report
from sklearn.utils import class_weight
```



```
#calculating weights for equal contribution

class_weights = class_weight.compute_class_weight('balanced', classes=np.unique(y), y=y)

class_weights = {0: 1.0, 1: 2.0}  # legit class is assigned standard weight and fraud Class is assigned twice the weight.

# Create a Random Forest classifier with class weights
rf = RandomForestClassifier(class_weight=class_weights)
```

```
#training the Machine learning model
rf.fit(X_train,y_train)
```

Figure 4*: code snippet of class weights*

4. We can use multiple algorithms to create a machine-learning model to predict fraud. It was a challenge to identify which model would perform the best and be the most efficient since the random forest and XG boost deliver similar results.

- **Impact**: All the selected algorithms handle classification tasks. Suppose we neglect the swift execution and computational requirement ability of the respective algorithms to predict fraud much quicker. In that case, it will set us back financially and delay counter-fraudulent measures in time.

- **Approach:** We have studied research papers on all the respective algorithms to understand their ability to handle the task quickly and more efficiently. We have opted out of logistic regression since the random forest and XGboost algorithms are more robust to outliers and collinearity. We have selected the random forest algorithm for our final model as it gives similar results with fewer computational requirements**.**

Integration of Academic research:

I. According to **(Frost, 2023)**, Principal component analysis guide and example, PCA takes a large dataset with multiple variables and reduces it to a combined set of indices, which retain most of the information and simplify the data to analyse more easily. The findings of *Jim Frost MS* assisted us in understanding the modifications of the transformed data by identifying the Variance ratio to determine which component retains most of the information during our data exploration phase.

II. **(Thinesh M.A et al., 2023)** paper on Detection of Credit Card Fraud Using Random Forest Classification Model explains the architecture of the and performance metrics of the random forest classifier model. As shown in *figure 5*, we have created a similar architecture to understand the sequence of analysing the dataset and building a machine learning model that uses random forest classifier.

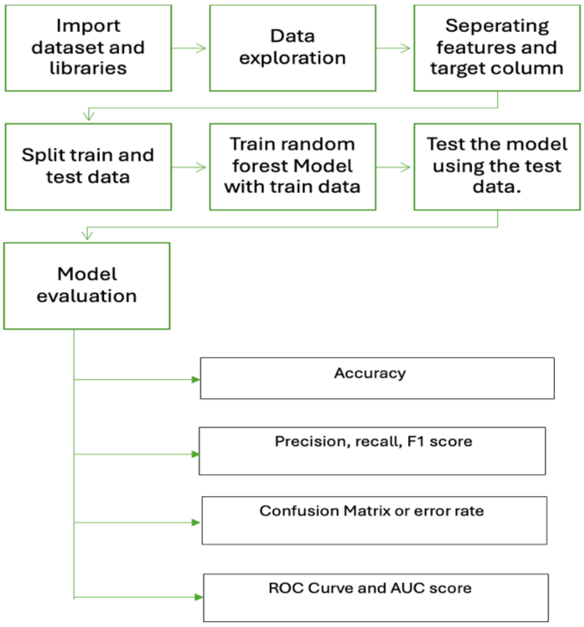

Import dataset and libraries
Data exploration
Seperating features and target column
Split train and test data
Train random forest Model with train data
Test the model using the test data.
Model evaluation
Accuracy
Precision, recall, F1 score
Confusion Matrix or error rate
ROC Curve and AUC score

Figure 5: *Architecture Diagram of the Model*

III. The paper on Overview of machine learning classification techniques by **(Alnuaimi and Albaldawi, 2024)** explained the types of machine learning and the classification. It helped us choose relevant model to predict fraudulent transaction and the type of classification based on the dataset we collected from Kaggle.

IV. **(Alsufyani et al., 2022)** Credit card fraud detection via machine learning, explains the need for data processing and the techniques which can be used to improve the model's efficiency. We have used his paper to understand how it can improve the model by using Pearson correlation to reduce the number of features that have high collinearity.

V. **(Kumar et al., 2019)** paper on credit card fraud detection using random forest algorithm focuses on how and why random forest can be used to predict fraudulent transactions. It helps us understand the techniques such as multiple decision trees, ensemble and bagging techniques that the algorithm uses for classification.

VI. The paper on Classification Model evaluation metrics by **(Vujovic, 2021)** describes confusion matrix and other metrics such as Accuracy, precision and ROC AUC. It assisted us to evaluate our models' performance and its ability to differentiate legitimate and fraudulent transactions.

VII. The article by **(Sruthi, 2021)** explains the working of random forest algorithm, how to set thresholds, extract information and customise the model based on the objective. It contributes to our project by helping us reduce the processing time by setting a limit on the number of features it considers splitting.

VIII. **(Abhinav, 2024)** Blog on Class weights in machine learning explains why class imbalance matters and how to overcome it by class weights. We have used this method in our model for accurately predicting fraud since the real-life data has a high class imbalance where fraudulent data is the minorly class.

IX. **(Ghattikar, 2023)** article explains how to extract the information of the impact of the feature on the model's prediction. We evaluated the feature's importance and the impact of that component and derived recommendations to prevent fraudulent transactions.

X. The standardisation of machine learning **(Sachin Vinay, 2021)** explains various methods, such as Z-score, min-max and standard deviation, to scale data and their respective advantages. It helped us identify that we should use Z-score method to scale the Amount data in our dataset since the credit card dataset contains a diverse range of data.

Critical reflection:

The data scientist role has considerably improved my overall analytical and technical skills. I gained a keen understanding of why analysing collinearity and dimensionality of the component and why identification and reduction methods such as Pearson correlation and PCA are crucial for classification machine learning models. Applying theoretical methods such as class weights and SMOTE in a real-world dataset helped me learn the necessity of using these techniques on the model to improve its fraud detection. The experience gained to overcome project challenges, specifically in analysing anonymised data to differentiate between relevant and irrelevant components using variance ratio score and Inter quartile methods. Additionally, being in a team improved my communication and collaborative skills since data scientists rely on clean data provided by the data engineers based on the objective set by the business strategists. Writing consultancy and individual reports provided me with an understanding of how to communicate findings and research to technical and non-technical stakeholders.

We understand that the random forest algorithm performs better than others in identifying fraudulent transactions, as shown in the table below. We can also identify that the model's error rate is considerably low, as shown in *figure 6* below.

	Logistic regression	Random Forest	XG boost
True positive (TP)	61114	72040	72046
True negatives (TN)	71984	61212	61197
False positive (FP)	56	6	9
False negative (FN)	119	15	21

Table 1: Error rate comparison of algorithms

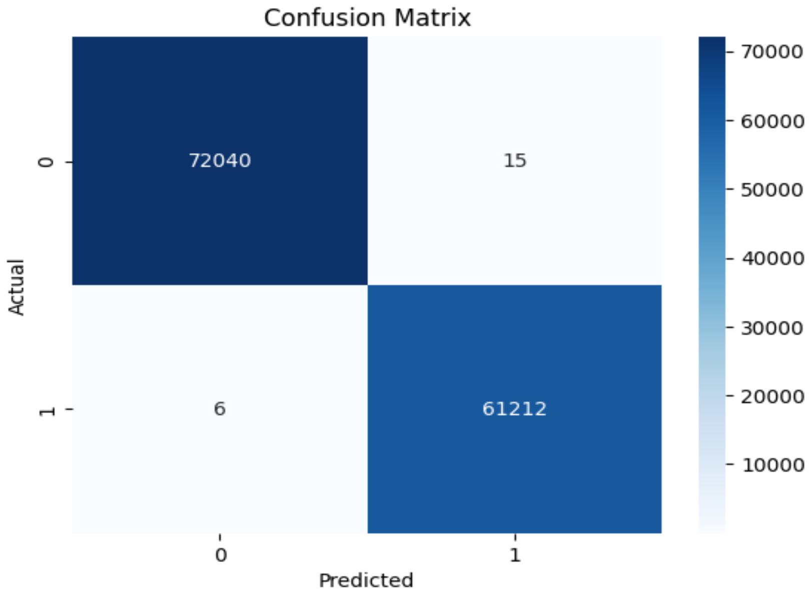


Figure 6: *Confusion matrix of random forest ML model*

True positive (TP) - correctly identifies 72,035 legitimate transactions.
True negative (TN) - correctly identifies 61,212 fraudulent transactions.
False negative (FN) - Identifies 15 legitimate transactions as fraud.
False positive (FP) - Identifies 6 fraudulent transactions as legitimate.

Recommendation for future project:

- An isolation forest randomly partitions the features, producing shorter tree paths for anomalous data points. In turn, it requires less memory than other algorithms, such as Z-score and IQR which saves computational requirements and execution time.

- Since the credit card transactions dataset often contains outliers and a wide range of data, using a robust scaler instead of standard scaling minimises the influence of outliers and skewness of the data while scaling, ensuring the importance of all features.

- Identifying the key principal components by mapping back to their original features and creating biplots and heatmaps showing how the components are related may provide insights into features that contribute the most to improving the model's prediction further.

Conclusion:

In summary, this report explains why employing data analysis methods such as Pearson correlation, Class weights, SMOTE and variance ratio on the dataset is essential before training the machine learning model. It also explains the challenges of understanding anonymised data and methodologies used to get further insight. Additionally, the suggested random forest classifier got an excellent result of 1 ROC AUC score and the least error (0.015%) compared to other classifiers such as logistic regression and XG boost.

The role of data scientists is crucial because they transform raw data into usable insights by feature engineering and developing machine learning model. Data scientists understand the patterns and thought processes behind fraudulent transactions from the dataset and create fraud detection models accordingly. Their role is imperative in maintaining a secure and safe financial ecosystem through data analytics.

References:

1. Frost, J. (2023). Principal Component Analysis Guide & Example. [online] Statistics By Jim. Available at: https://statisticsbyjim.com/basics/principal-component-analysis/.
2. Thinesh M.A, S S Mukhil Varmann, Leoni Sharmila and Das, S.R. (2023). Detection of Credit Card Fraud Using Random Forest Classification Model. ResearchGate, [online] 1(4), pp.184–187.
   Available at
   https://www.researchgate.net/publication/379957258_Detection_of_Credit_Card_Fraud_Using_Random_Forest_Classification_Model[Accessed 18 Nov. 2024].

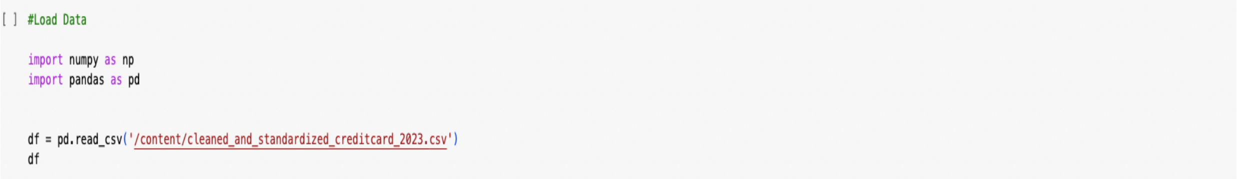
3. Alnuaimi, A.F.A.H. and Albaldawi, T.H.K. (2024). An overview of machine learning classification techniques. *Bio web of conferences/BIO web of conferences*, 97, pp.08–12. doi:https://doi.org/10.1051/bioconf/20249700133
4. Alsufyani, K., AlMuallim, A., AlShahrani, M., Alsufyani, A., Alhanaya, O. and Zerguine, A. (2022). *Credit Card Fraud Detection via Machine Learning*. [online] IEEE Xplore. doi:https://doi.org/10.1109/SSD54932.2022.9955815
5. Kumar, M.S., Soundarya, V., Kavitha, S., Keerthika, E.S. and Aswini, E. (2019). *Credit Card Fraud Detection Using Random Forest Algorithm*. [online] IEEE Xplore. doi:https://doi.org/10.1109/ICCCT2.2019.8824930
6. Vujovic, Ž.Đ. (2021). Classification Model Evaluation Metrics. *International Journal of Advanced Computer Science and Applications*, 12(6). Doi:https://doi.org/10.14569/ijacsa.2021.0120670
7. Sruthi, E.R. (2021). Random Forest | Introduction to Random Forest Algorithm. [online] Analytics Vidhya. Available at: https://www.analyticsvidhya.com/blog/2021/06/understanding-random-forest/
8. Abhinav, R. (2023). Improving Class Imbalance with Class Weights in Machine Learning. [online] Medium. Available at: https://medium.com/@ravi.abhinav4/improving-class-imbalance-with-class-weights-in-machine-learning-af072fdd4aa4.
9. Prasanna Ghattikar (2023). Using Random Forest For Feature Importance And Feature Selection. [online] Medium. Available at: https://medium.com/@prasannarghattikar/using-random-forest-for-feature-importance-118462c40189.
10. Sachin Vinay (2021). STANDARDIZATION IN MACHINE LEARNING. [online] Available at: https://www.researchgate.net/publication/349869617_STANDARDIZATION_IN_MACHINE_LEARNING#fullTextFileContent.

11. Bhatla, T., Prabhu, V. and Dua, A. (2003). *Understanding Credit Card Frauds*. [online] Available at: https://popcenter.asu.edu/sites/default/files/problems/credit_card_fraud/PDFs/Bhatla.pdf.

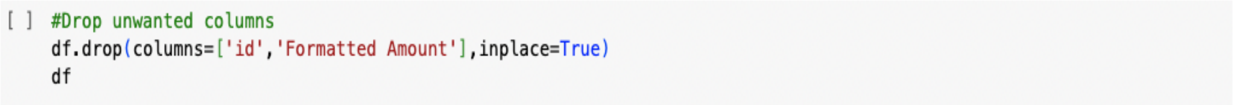
12. Thennakoon, A., Bhagyani, C., Premadasa, S., Mihiranga, S. and Kuruwitaarachchi, N. (2019). Real-time Credit Card Fraud Detection Using Machine Learning. [online] IEEE Xplore. doi:https://doi.org/10.1109/CONFLUENCE.2019.8776942

Appendix:

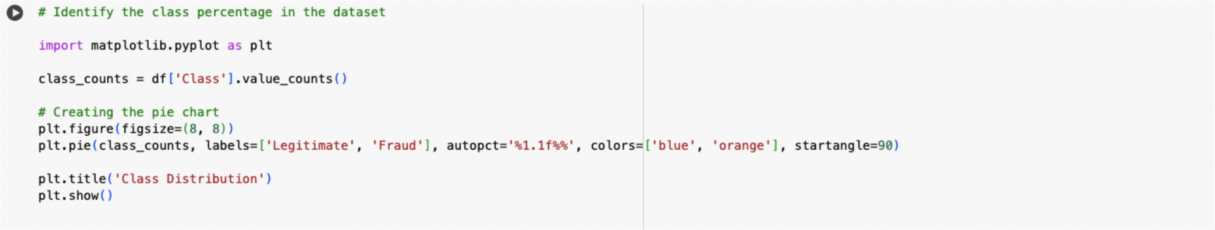
Google collab Notebook: https://colab.research.google.com/drive/1YwSFj9kq-QmMFZfAglj9hZx32xyUN_Cf?usp=sharing



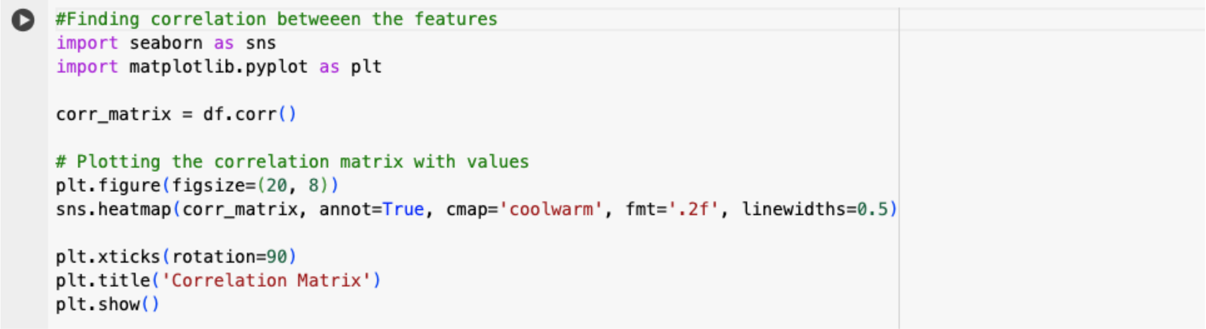
```
[ ] #Load Data

    import numpy as np
    import pandas as pd


    df = pd.read_csv('/content/cleaned_and_standardized_creditcard_2023.csv')
    df
```



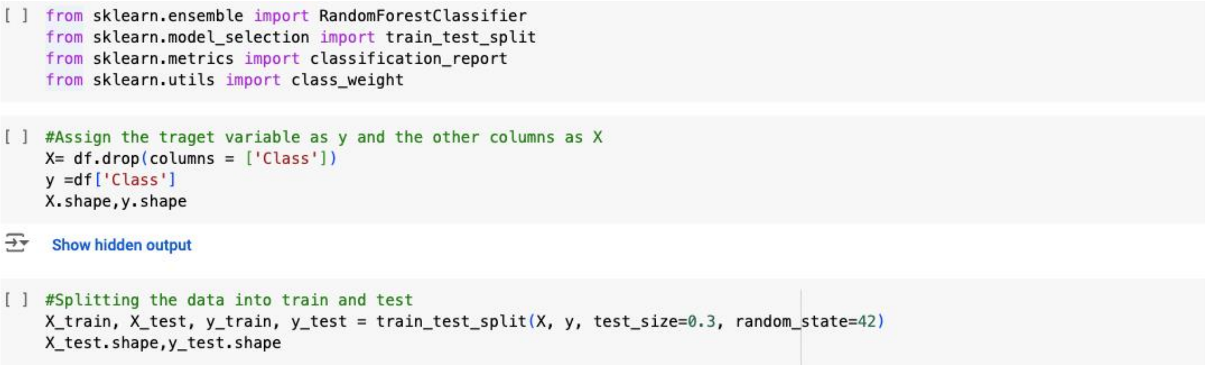
```
[ ] #Drop unwanted columns
    df.drop(columns=['id','Formatted Amount'],inplace=True)
    df
```



```
# Identify the class percentage in the dataset

import matplotlib.pyplot as plt

class_counts = df['Class'].value_counts()

# Creating the pie chart
plt.figure(figsize=(8, 8))
plt.pie(class_counts, labels=['Legitimate', 'Fraud'], autopct='%1.1f%%', colors=['blue', 'orange'], startangle=90)

plt.title('Class Distribution')
plt.show()
```



```
#Finding correlation betweeen the features
import seaborn as sns
import matplotlib.pyplot as plt

corr_matrix = df.corr()

# Plotting the correlation matrix with values
plt.figure(figsize=(20, 8))
sns.heatmap(corr_matrix, annot=True, cmap='coolwarm', fmt='.2f', linewidths=0.5)

plt.xticks(rotation=90)
plt.title('Correlation Matrix')
plt.show()
```

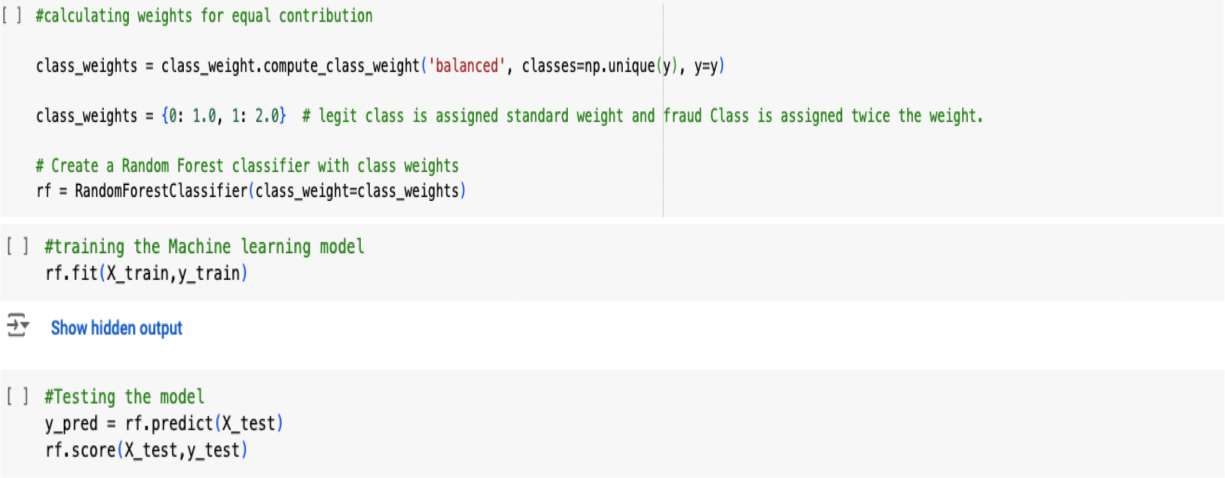


```
[ ] from sklearn.ensemble import RandomForestClassifier
    from sklearn.model_selection import train_test_split
    from sklearn.metrics import classification_report
    from sklearn.utils import class_weight
```

```
[ ] #Assign the traget variable as y and the other columns as X
    X= df.drop(columns = ['Class'])
    y =df['Class']
    X.shape,y.shape
```

Show hidden output

```
[ ] #Splitting the data into train and test
    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)
    X_test.shape,y_test.shape
```

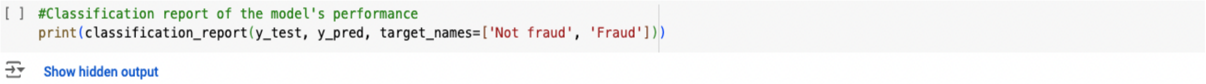


```
[ ] #calculating weights for equal contribution

    class_weights = class_weight.compute_class_weight('balanced', classes=np.unique(y), y=y)

    class_weights = {0: 1.0, 1: 2.0}  # legit class is assigned standard weight and fraud Class is assigned twice the weight.

    # Create a Random Forest classifier with class weights
    rf = RandomForestClassifier(class_weight=class_weights)
```

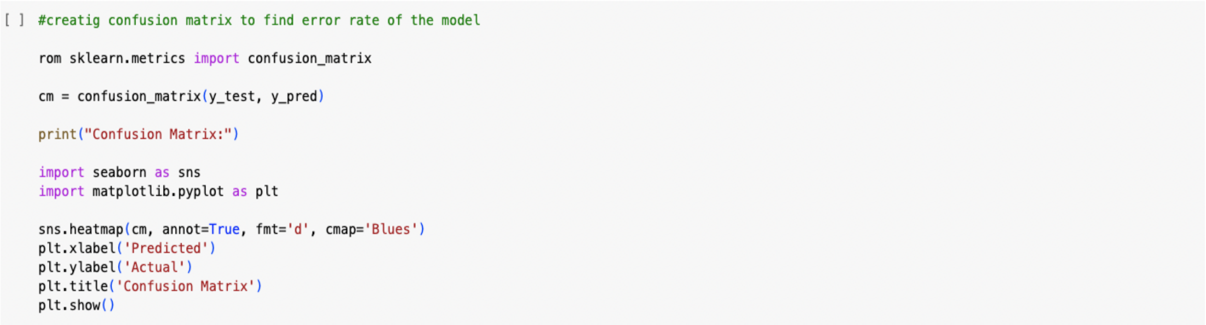
```
[ ] #training the Machine learning model
    rf.fit(X_train,y_train)
```

Show hidden output

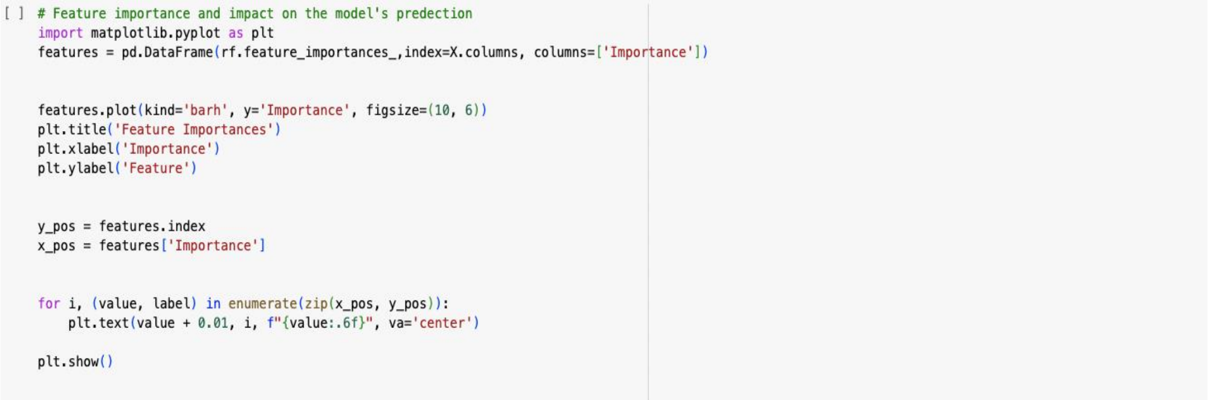
```
[ ] #Testing the model
    y_pred = rf.predict(X_test)
    rf.score(X_test,y_test)
```



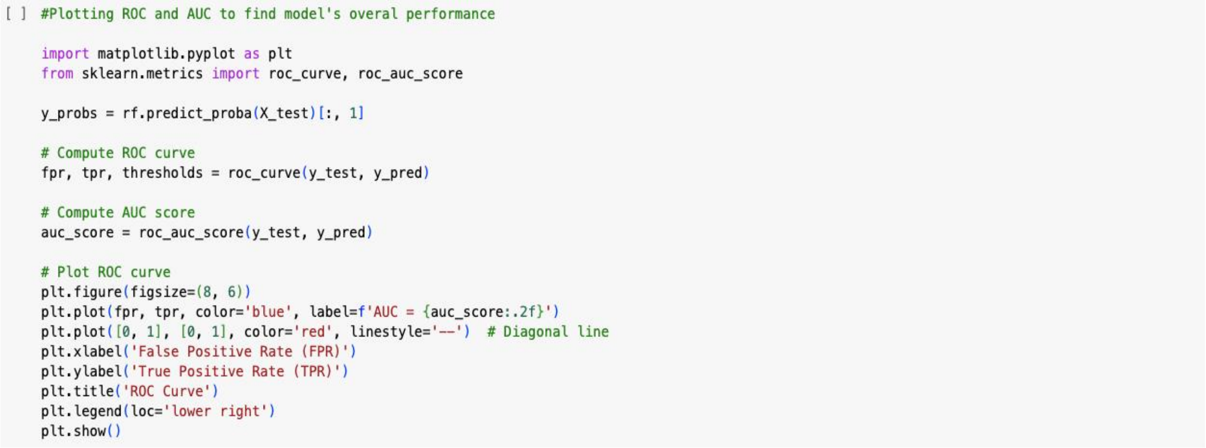
```
#Classification report of the model's performance
print(classification_report(y_test, y_pred, target_names=['Not fraud', 'Fraud']))
```



```
#creatig confusion matrix to find error rate of the model

rom sklearn.metrics import confusion_matrix

cm = confusion_matrix(y_test, y_pred)

print("Confusion Matrix:")

import seaborn as sns
import matplotlib.pyplot as plt

sns.heatmap(cm, annot=True, fmt='d', cmap='Blues')
plt.xlabel('Predicted')
plt.ylabel('Actual')
plt.title('Confusion Matrix')
plt.show()
```

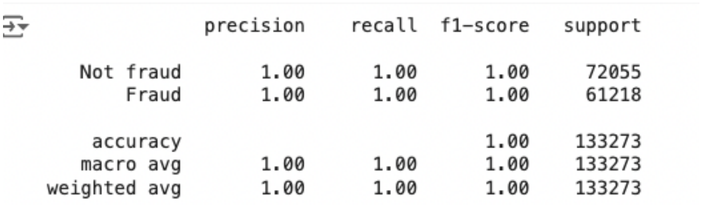


```
# Feature importance and impact on the model's predection
import matplotlib.pyplot as plt
features = pd.DataFrame(rf.feature_importances_,index=X.columns, columns=['Importance'])


features.plot(kind='barh', y='Importance', figsize=(10, 6))
plt.title('Feature Importances')
plt.xlabel('Importance')
plt.ylabel('Feature')


y_pos = features.index
x_pos = features['Importance']


for i, (value, label) in enumerate(zip(x_pos, y_pos)):
    plt.text(value + 0.01, i, f"{value:.6f}", va='center')

plt.show()
```

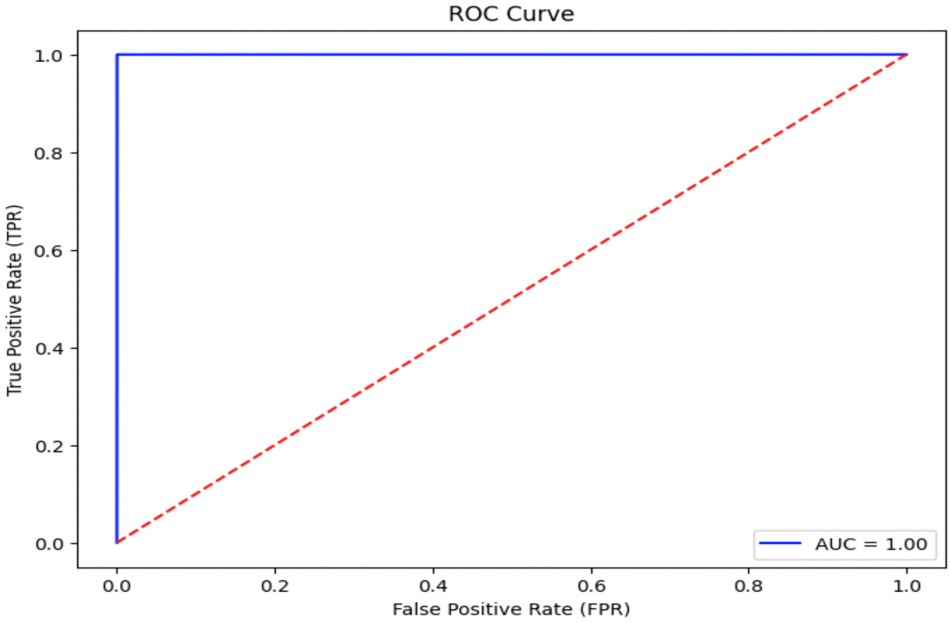


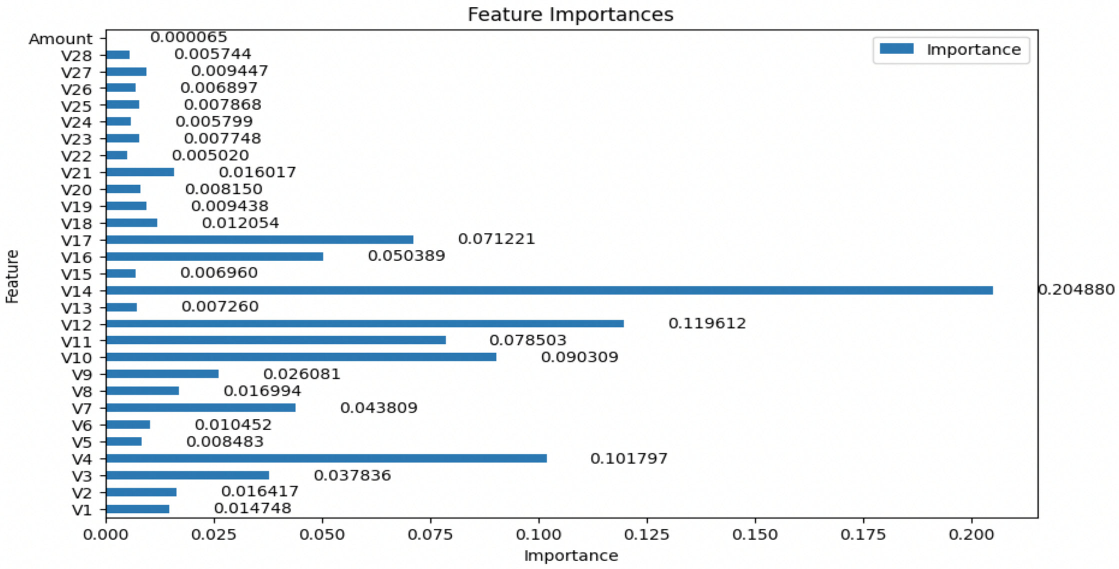
```
#Plotting ROC and AUC to find model's overal performance

import matplotlib.pyplot as plt
from sklearn.metrics import roc_curve, roc_auc_score

y_probs = rf.predict_proba(X_test)[:, 1]

# Compute ROC curve
fpr, tpr, thresholds = roc_curve(y_test, y_pred)

# Compute AUC score
auc_score = roc_auc_score(y_test, y_pred)

# Plot ROC curve
plt.figure(figsize=(8, 6))
plt.plot(fpr, tpr, color='blue', label=f'AUC = {auc_score:.2f}')
plt.plot([0, 1], [0, 1], color='red', linestyle='--')  # Diagonal line
plt.xlabel('False Positive Rate (FPR)')
plt.ylabel('True Positive Rate (TPR)')
plt.title('ROC Curve')
plt.legend(loc='lower right')
plt.show()
```

```
# Feature importance and impact on the model's predection
import matplotlib.pyplot as plt
features = pd.DataFrame(rf.feature_importances_,index=X.columns, columns=['Importance'])


features.plot(kind='barh', y='Importance', figsize=(10, 6))
plt.title('Feature Importances')
plt.xlabel('Importance')
plt.ylabel('Feature')


y_pos = features.index
x_pos = features['Importance']


for i, (value, label) in enumerate(zip(x_pos, y_pos)):
    plt.text(value + 0.01, i, f"{value:.6f}", va='center')

plt.show()
```



```
              precision    recall  f1-score   support

   Not fraud       1.00      1.00      1.00     72055
       Fraud       1.00      1.00      1.00     61218

    accuracy                           1.00    133273
   macro avg       1.00      1.00      1.00    133273
weighted avg       1.00      1.00      1.00    133273
```





End of Report