BIG DATA ALGORITHMS AND STATISTIC

TERM PROJECT

GROUP G

ALWIN SCARIA

MOHAMED AFTHAB

DONA BIJU

JERIN THOMAS

SNEHA SUJATHA

TITLLE: ONLINE PAYMENTS FRAUD DETECTION

Dataset : https://www.kaggle.com/datasets/rupakroy/online-payments-fraud-detection-dataset

Git Repository: https://github.com/dona516/Fraud_Detection

SUMMARY

OBJECTIVES:

Fraud is one of the main issues that companies and organizations are dealing with in this increasingly digital world. Using creative and efficient methods to halt fraudulent behaviour is essential since scammers are employing advanced tactics to exploit financial systems and personal data. Fraud data analytics has grown to be a vital tool in the fight against fraud, providing organizations with new insights into potential risks and enabling them to recognize, detect, and thwart fraudulent activity.

The purpose of selecting the "Online payments Fraud Detection Dataset" was to develop a trustworthy system for detecting fraudulent activities associated with online payments.

Project Outcome: The goal is to create a fraud detection model that uses label encoding for categorical variables to offer a dataset suitable for machine learning techniques. The model needs to have enhanced accuracy and efficacy in identifying fraudulent transactions.

Data Source Description: The data for this project was sourced from Kaggle. The simulation spans 30 days. The dataset denotes different types of transaction, amount, old balance ,new balance, financial status.

The dataset includes details on online transactions and associated attributes. In the dataset, each transaction is represented by a row that has the following properties:

Features
step:
Definition: Depicts a real-world time unit in which one hour is represented by each step.
Context: There are 744 steps in all during the 30-day simulation.
kind:

type:
Definition: Indicates the kind of transaction and accepts the following values: TRANSFER, PAYMENT, DEBIT, CASH-IN, and CASH-OUT.
Context: Explains the type of financial activity that is involved in every transaction.

amount:
Definition: Denotes the transaction's monetary worth.
Context: Indicates the total amount of money used in every financial transaction.

nameOrg:
Defined: Ascertains which client started the transaction.
Context: Assigns the account holder who initiated each transaction to that transaction.

oldbalanceOrg:
Definition: Represents the customer's starting account balance prior to the transaction.
Context: Gives the account holder's initial financial situation.

newbalanceOrg:
Definition: Shows the customer's updated account balance following the transaction.
Context: After the transaction, records the account holder's revised financial situation

nameDest:
Definition: Identifies the transaction's recipient customer.
Context: Connects transactions to the account that is receiving the money transfer.

previousbalanceDest:
Definition: Shows the recipient's starting account amount prior to the transaction.
Information for clients beginning with M (Merchants) is absent.
Provides information on the recipient account's initial financial situation.

newbalanceDest:
Definition: Denotes the recipient's updated account balance following the transaction.
Information for clients beginning with M (Merchants) is absent.
Context: After the transaction, records the recipient account's modified financial state.

isFraud:
Definition: A binary indicator (0 or 1) indicating the presence or absence of fraud (1) or (0) in the transaction.
Context: Recognizes simulation transactions that involve dishonest activity.

METHODS USED:

- Data Preprocessing
- Data Visualization
- Feature Engineering
- Feature Scaling and Transformation
- Model Selection and Creation
- Model Evaluation

INTRODUCTION

Data Preprocessing

In the data preprocessing step, basic data overview is displayed like the number of rows and columns, statistical summary of the dataset, datatypes, unique value, also performed aggregation functions such as to find the total value and sum of features.

Data Cleaning:

The data consist of null values in the columns oldbalanceOrg,nameDest,oldbalanceDest.

```
In [62]: data.isna().sum() # print number of missing values in each columns

Out[62]: step              0
         type              0
         amount            0
         nameOrig          0
         oldbalanceOrg     5
         newbalanceOrig    0
         nameDest          4
         oldbalanceDest    5
         newbalanceDest    0
         isFraud           0
         isFlaggedFraud    0
         dtype: int64

In [63]: # filling missing values in numerical columns 'oldbalanceOrg'and 'newbalanceOrig' with mean values
         numerical_cols = ['oldbalanceOrg','oldbalanceDest']
         for col in numerical_cols:
             mean_val = data[col].mean()
             data[col].fillna(mean_val, inplace=True)

In [64]: # filling missing values in categorical column 'nameDest' with mode
         data['nameDest'].fillna(data['nameDest'].mode()[0], inplace=True)
```

The missing values in the numerical column like oldbalanceOrg and oldbalanceDestis filled with the imputation method using mean.

The missing value in the categorical column like nameDest is filled with the mode method.

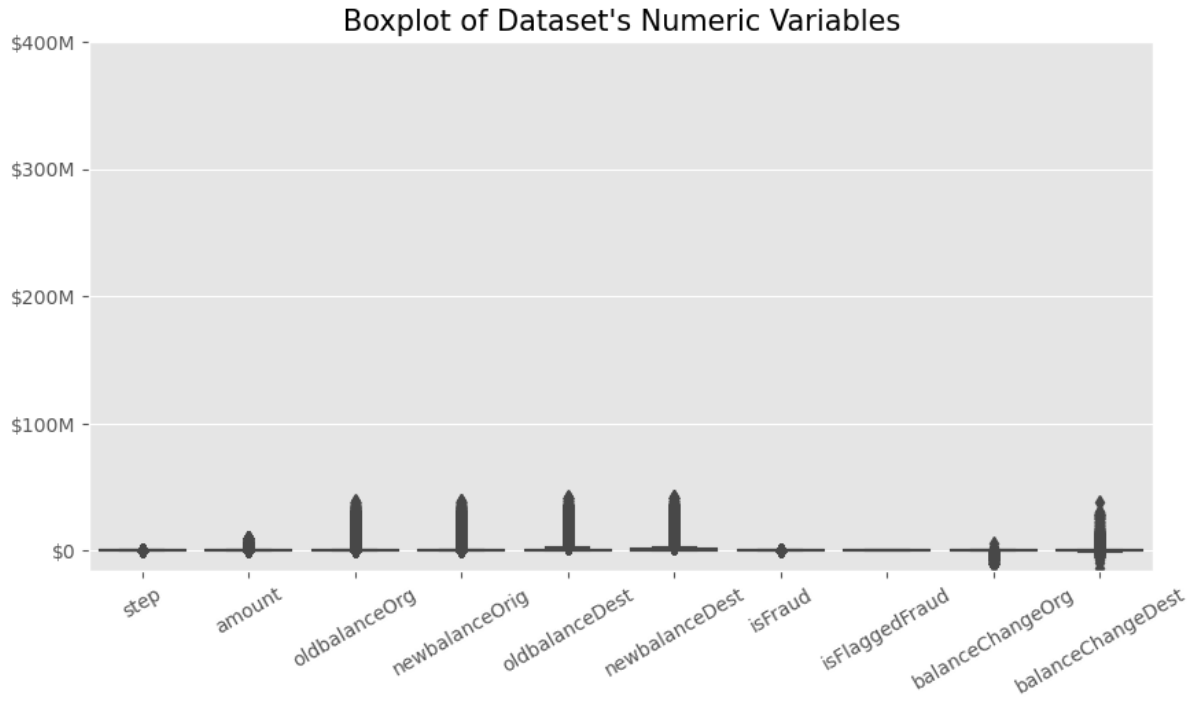
In the further steps, all missing values were filled appropriately.

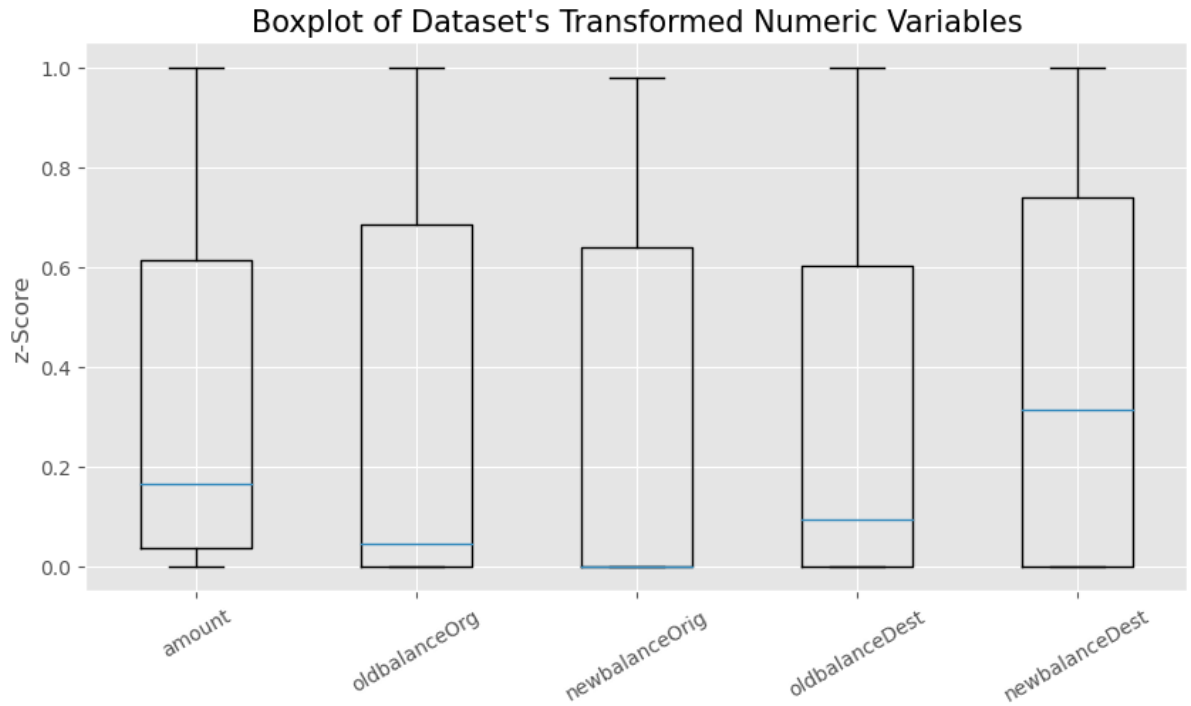
```
print(data.isnull().sum())

step              0
type              0
amount            0
nameOrig          0
oldbalanceOrg     0
newbalanceOrig    0
nameDest          0
oldbalanceDest    0
newbalanceDest    0
isFraud           0
isFlaggedFraud    0
dtype: int64
```

Also, there was no duplicate values in the dataset.

Outlier Detection: Outlier in the dataset is detected with the help of boxplot which helps to show the extreme high and low values in each feature.

Here we have used the SimpleImputer and Normalizer method to handle outliers.IQR method is also implemented to indicate how data points are distributed around the median.





DATA ENCODING

Data encoding is a very important aspect in machine learning, to handle the categorical data and convert into numerical data. Here the 'type' feature consisted of values ('PAYMENT','TRANSFER','CASH_OUT','DEBIT','CASH_IN') which is further converted to numerical values.

	step	type	amount	nameOrig	oldbalanceOrg	newbalanceOrig	nameDest	oldbalanceDest	newbalanceDest	isFraud	isFlaggedFraud
0	1	PAYMENT	9839.64	C1231006815	170136.00	160296.36	M1979787155	0.00	0.00	0	0
1	1	PAYMENT	1864.28	C1666544295	21249.00	19384.72	M2044282225	0.00	0.00	0	0
2	1	TRANSFER	181.00	C1305486145	181.00	0.00	C553264065	0.00	0.00	1	0
3	1	CASH_OUT	181.00	C840083671	181.00	0.00	C38997010	21182.00	0.00	1	0
4	1	PAYMENT	11668.14	C2048537720	0.00	29885.86	M1230701703	0.00	0.00	0	0
...	...	...	...	...	...	...	...	...	...	...	...
048570	95	CASH_OUT	132557.35	C1179511630	479803.00	347245.65	C435674507	484329.37	616886.72	0	0
048571	95	PAYMENT	9917.36	C1956161225	90545.00	80627.64	M668364942	0.00	0.00	0	0
048572	95	PAYMENT	14140.05	C2037964975	20545.00	6404.95	M1355182933	0.00	0.00	0	0
048573	95	PAYMENT	10020.05	C1633237354	90605.00	80584.95	M1964992463	0.00	0.00	0	0
048574	95	PAYMENT	11450.03	C1264356443	80584.95	69134.92	M677577406	0.00	0.00	0	0

```
labelencoder = LabelEncoder()
data['type'] = labelencoder.fit_transform(data['type'])
print(data['type'])
```

```
0          3
1          3
2          4
3          1
4          3
          ..
1048570    1
1048571    3
1048572    3
1048573    3
1048574    3
Name: type, Length: 1048575, dtype: int32
```

```
40]:  data.head() # display first 5 rows of dataframe for checking the encoding
```

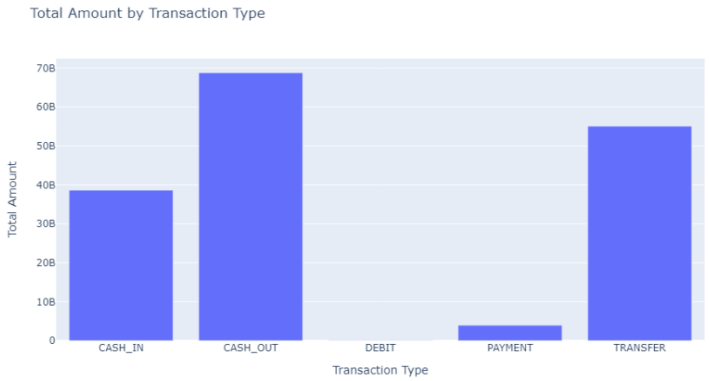
Out[40]:

	step	type	amount	nameOrig	oldbalanceOrg	newbalanceOrig	nameDest	oldbalanceDest	newbalanceDest	isFraud	isFlaggedFraud	balanceCha
0	1	3	9839.64	C1231006815	170136.0	160296.36	M1979787155	0.0	0.0	0	0	-
1	1	3	1864.28	C1666544295	21249.0	19384.72	M2044282225	0.0	0.0	0	0	-
2	1	4	181.00	C1305486145	181.0	0.00	C553264065	0.0	0.0	1	0	
3	1	1	181.00	C840083671	181.0	0.00	C38997010	21182.0	0.0	1	0	
4	1	3	11668.14	C2048537720	0.0	29885.86	M1230701703	0.0	0.0	0	0	2

Data Visualization

Various visualizations are performed in order to have a better understanding of the data and it's outcome. Graphs like Barchart, Histogram, Pie chart, Scatter plot, Pair plot, Countplot, Heatmap and Box Plot are represented with the help of libraries such as Matplotlib, Plotly, Seaborn.

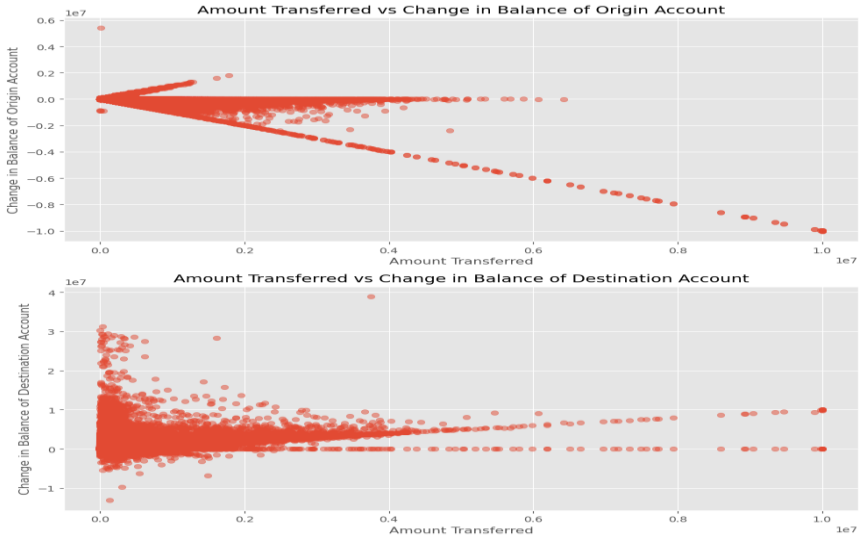
1) Bar Chart



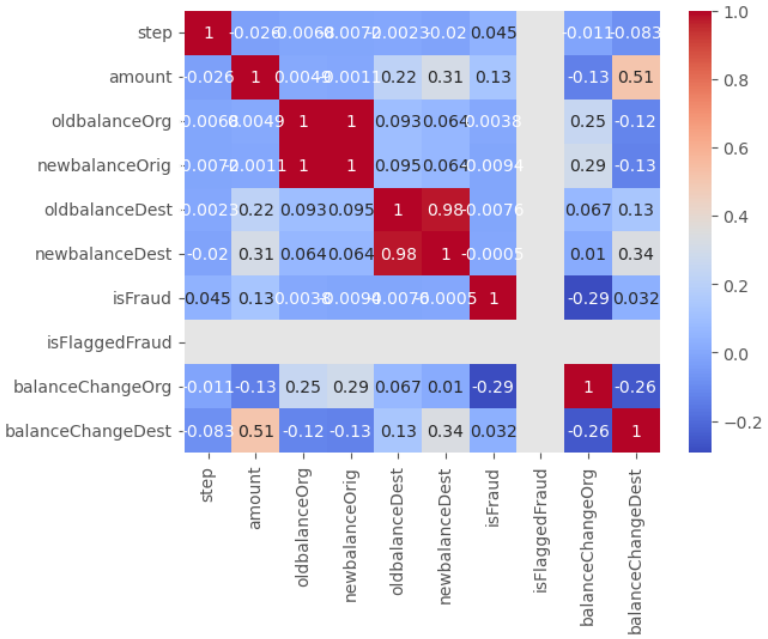
Bar chart showing that CASH_OUT has the highest count followed by direct transfer,CASH_IN and the lowest count is for the Payment value.The Debit transaction type value is not present in the dataset.

2) Scatter Plot

The scatter plots make it possible to visually examine possible correlations or trends between the amount transferred and the variations in the balances of the source and destination accounts. The transaction behaviour can be better understood, and any trends or abnormalities can be found with the use of this display.



3) Heatmap
The heatmap visualization is represented to show the correlation between the features to understand the multicollinearity and the correlation with the target variable.



Data Transformation

In this project we performed data transformation using 3 Techniques:

Normalization:

By using the 'Normalizer' class from scikit learn, we have performed transformation on the numerical columns. Normalization scales each column to have a unit such that to have a mean of 0 and a standard deviation of 1.

Standardization:

'StandardScaler' class is used to standardize our feature and to assume the data is normally distributed. The training and test data (X_train and X_test) were changed using the fit_transform() and transform() methods, respectively, after the scaler was fitted to the training data (X_train).

Min-Max Scaling:

As with standardization, we first fitted the scaler to the training set, and then we used the fit_transform() and transform() methods to transform the training set as well as the test set.

These step helps to improve the performance and convergence of certain machine learning algorithms.

Challenges Faced in the Data Preprocessing Steps:

- Some of the challenges faced in the Data Preprocessing step is to remove the outlier with the right technique.
- Also, to identify the collinearity among the features was difficult since most of them showed a minimal correlation.

MODEL CREATION

MODEL 1 : DECISION TREE CLASSIFIER

SimpleImputer class is used to impute missing feature value with the mean of each feature.

Split data into training and testing sets using the train_ test_split function from scikit-learn

The Decision Tree Classifier with default hyperparameters and fixed random state is created .

Using the fit() function, we trained the decision tree classifier using the training data (X_train and y_train).

Following training, we used the predict() method to make predictions on the test data (X_test).

Accuracy
The accuracy using this model is 99%

MODEL INSIGHTS

The accuracy of the model was at about 99.98%. But when it comes to imbalanced datasets, accuracy might not be the most useful metric—especially when it comes to fraud detection.

MODEL EVALUATION METRICS:

Precision: 0.880

Recall: 0.849

F1 Score : 0.8649

ROC-AUC Score : 0.927

Confusion Matrix:

Confusion Matrix:
[[816769 75]
[98 554]]

INSIGHTS OF MODEL EVALUATION:

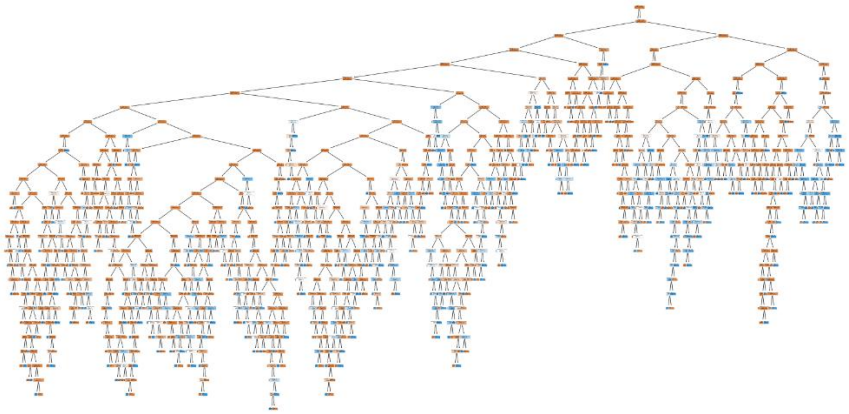
- Precision is about 88.08%, which is the percentage of true positive forecasts among all positive predictions. This means that 88.08% of the time, the model is right when it p
- redicts a transaction to be fraudulent. The percentage of accurate positive predictions among all real positive cases is known as recall, and it is roughly 84.97%.

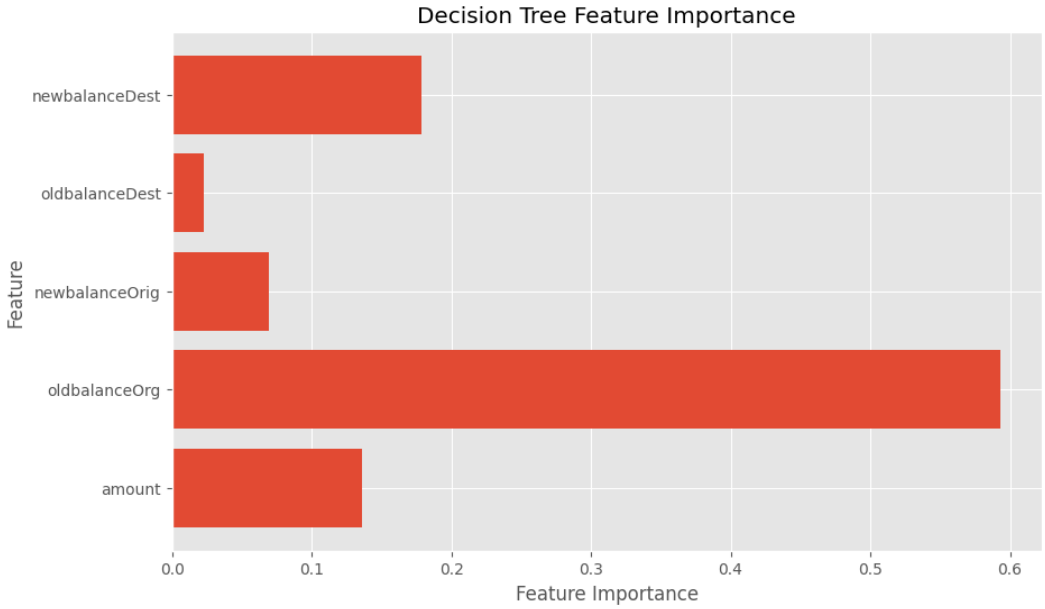
This implies that around 84.97% of all real fraudulent transactions can be detected by the program.

- F1 Score and ROC-AUC Score: Approximately 86.50% is the F1 score, which strikes a balance between recall and precision. This score offers a combined assessment of the model's ability to distinguish between transactions that are fraudulent and those that are not.
  The model's ability to differentiate between classes is evaluated by the ROC-AUC score, which comes out to be roughly 92.71%. Better discriminating between positive and negative cases is indicated by a higher ROC-AUC score.
-

  The model's predictions are broken down in great depth in the confusion matrix. It demonstrates that the model properly identified some fraudulent transactions (true positives: 554) and correctly classified a sizable number of transactions as non-fraudulent (816,769). There were, nevertheless, also cases of incorrect categorization, including 98 false negative and 75 false positive results.

MODEL INTERPRETATION





INSIGHTS OF MODEL INTERPRETATION:

OldbalanceOrg is the most significant feature, followed by newbalanceDest; oldbalanceDest is considered the least significant feature. This implies that a major factor in predicting fraudulent transactions is the destination account's balance (oldbalanceOrg and newbalanceDest).

ENSEMBLE METHOD

GRID SEARCH FOR HYPERPARAMTER TUNING

We have used " GridSearchCV" to search for best hyperparameters for a Decision Tree Classifier using cross-validation.
The 'param_grid' variable contains the hyperparamter grid

The best model is extracted using ‘best_estimator_’

Ensemble Method-Random Forest
We have installed a Random Forest Classifier with 100 trees and fitted to the training data.

Cross Validation
We have performed Cross Validation to assess the Performance of Random Forest Model

Deployment
The trained Random Forest model is saved using ‘joblib.dump’ for future deployment.

Evaluation of Model's Advantages and Drawbacks:
Advantages:
High performance metrics and accuracy show how well the model separates fraudulent from non-fraudulent transactions.

Decision trees' interpretability makes it simple to comprehend the underlying decision-making process.
Weaknesses: Overfitting of decision trees to the training set may result in poorer generalization performance on untrained data.
A model may be biased towards the majority class and performance measures may be impacted by imbalanced datasets, which are common in fraud detection.

Conclusion:

Major findings:
The decision tree model successfully detects fraudulent transactions, and destination account balances are an important factor in this process.
A fraud detection model was successfully constructed, and the project also yielded insights into feature relevance and model performance. These are the accomplishments of the project objectives.

Suggestions for Further Research:
To increase the model's performance, more research might investigate sophisticated anomaly detection techniques. It could also look at features other than account balances for improved fraud detection capabilities.

REFERENCE:

https://www.analyticsvidhya.com/blog/2021/10/everything-you-need-to-know-about-linear-regression/

http://www.stat.yale.edu/Courses/1997-98/101/linreg.htm

https://towardsdatascience.com/what-is-stratified-cross-validation-in-machine-learning-8844f3e7ae8e

https://www.simplilearn.com/what-is-xgboost-algorithm-in-machine-learning-article

https://www.kaggle.com/datasets/rupakroy/online-payments-fraud-detection-dataset