FINANCIAL FRAUD DETECTION USING MACHINE LEARNING MODELS

BY

Vinay Chowdari Mandava, F233Z974

Vijaya Ramya CH, K968X456

Ravikiran Nallamothu, Q684A655

Lokesh Muppalla, J639H476

Sai Manish Koganti, H664V993

Sanku Venkata Kiran,S867B497

ABSTRACT:

Payments-related fraud is a major concern for cyber-crime organizations, and recent research has shown that machine learning approaches may successfully detect fraudulent transactions in vast amounts of data. Such algorithms can detect fraudulent transactions that human auditors may not identify, and they can do so in real-time. Using publicly available simulated payment transaction data, we apply different supervised machine learning algorithms to the problem of fraud detection in this research. We want to show we may use how supervised machine learning techniques to accurately categorize data with substantial class imbalance. We show how to use exploratory analysis to distinguish between fraudulent and non-fraudulent transactions. In addition, we also show that tree-based techniques like Random Forest outperform Logistic Regression given a well-separated dataset.

INTRODUCTION:

Digital payments in various forms are on the rise all around the world. The volume of transactions handled by payment companies is rapidly increasing. In 2018, PayPal, for example, processed $578 billion in total payments. Along with this change, there has been a significant surge in financial fraud in various payment systems. The role of cybersecurity and cyber-crime teams includes preventing online financial fraud. Most banks and financial organizations have specialized teams of dozens of analysts working on automated systems to monitor transactions made through their products and flag those that are possibly fraudulent. As a result, in order to be better prepared to address the challenge of identifying fraudulent entries/transactions in vast volumes of data, it is critical to investigate the strategy to tackle the problem.

DATA SOURCES:

Because financial data is private, there are few publicly available datasets that can be used for analysis. A synthetic dataset built using a simulator called PaySim is used in this research, which is public on Kaggle. I introduced malicious entries into the dataset after they constructed it using aggregated metrics from a multinational mobile financial services company's proprietary dataset.

The dataset contains 11 columns of information for ~6 million rows of data. The key columns available are

- Type of transactions
- Amount transacted
- Customer ID and Recipient ID
- Old and New balance of Customer and Recipient
- Time step of the transaction
- Whether or not the transaction was fraudulent

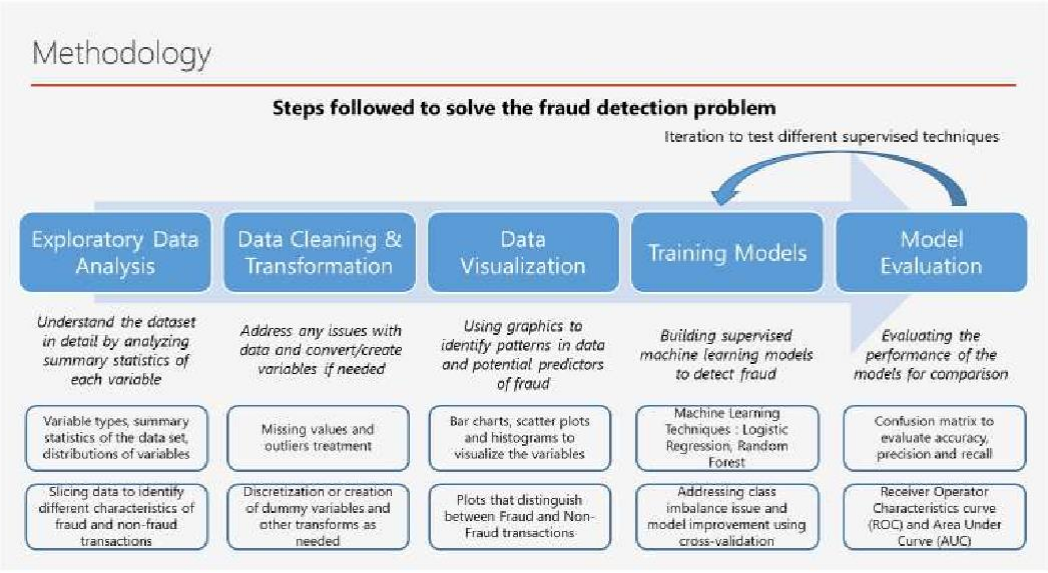
In the following figure, a snapshot of the first few lines of the data set is presented.

Figure 1: Snapshot of the raw dataset

step	type	amount	nameOrig	oldbalanceOrg	newbalanceOrig	nameDest	oldbalanceDest	newbalanceDest	isFraud	isFlaggedFraud
1	PAYMENT	9839.64	C1231006815	170136	160296.36	M1979787155	0	0	0	0
1	PAYMENT	1864.28	C1666544295	21249	19384.72	M2044282225	0	0	0	0
1	TRANSFER	181	C1305486145	181	0	C553264065	0	0	1	0
1	CASH_OUT	181	C840083671	181	0	C38997010	21182	0	1	0
1	PAYMENT	11668.14	C2048537720	41554	29885.86	M1230701703	0	0	0	0
1	PAYMENT	7817.71	C90045638	53860	46042.29	M573487274	0	0	0	0
1	PAYMENT	7107.77	C154988899	183195	176087.23	M408069119	0	0	0	0
1	PAYMENT	7861.64	C1912850431	176087.23	168225.59	M633326333	0	0	0	0
1	PAYMENT	4024.36	C1265012928	2671	0	M1176932104	0	0	0	0
1	DEBIT	5337.77	C712410124	41720	36382.23	C195600860	41898	40348.79	0	0
1	DEBIT	9644.94	C1900366749	4465	0	C997608398	10845	157982.12	0	0
1	PAYMENT	3099.97	C249177573	20771	17671.03	M2096539129	0	0	0	0
1	PAYMENT	2560.74	C1648232591	5070	2509.26	M972865270	0	0	0	0
1	PAYMENT	11633.76	C1716932897	10127	0	M801569151	0	0	0	0

METHODOLOGY:

In this project, the typical machine learning approach was used. I used the tagged class variable in the discovered dataset as the prediction variable in machine learning models.



The project's deliverables were based on this method. It discusses the outcomes of each phase that was tested and compares them to determine which strategy is the best for addressing the fraud detection challenge.

We describe the findings from each step of the project in the output for that phase. The following are the deliverables that we used for this project.

Table 1: Project Deliverables

Methodology Phases	Project Deliverables
Understanding the data set	● Report on the summary of the data set and each variable it contains along with necessary visualizations
Exploratory Data Analysis	● Report on analysis conducted and critical findings with a full description of data slices considered ● Hypothesis about the separation between fraud and non- fraud transactions ● Visualizations and charts that show the differences between fraud and non-fraud transactions ● Python code of the analysis performed
Modeling	● Report on the results of the different techniques tried out, iterations that were experimented with, data transformations and the detailed modeling approach ● Python code used to build machine learning models
Final Project Report	● Final report summarizing the work done over the course of the project, highlighting the key findings, comparing different models and identifying best model for financial fraud detection

TOOLS USED:

We entirely did this project using Python, and we documented the analysis in a Jupyter notebook. Standard python libraries were used to conduct different analyses. I described these libraries below –

✓ ***sklearn*** – used for machine learning tasks
✓ ***seaborn*** – used to generate charts and visualizations
✓ ***pandas*** – used for reading and transforming the data

DATA DESCRIPTION:

We generated the data for this investigation using a PaySim simulator to create a synthetic dataset of digital transactions. PaySim replicates mobile money transactions using a sample of genuine transactions collected from a month's worth of financial logs from an African country's mobile money service. It creates a synthetic dataset from anonymized data from the private dataset and then injects fraudulent transactions.

There are over 6 million transactions in the dataset, including 11 variables. The variable 'isFraud' specifies whether the transaction is actually fraudulent

We describe the columns in the dataset as follows:

Table 2: Variables in the Dataset

Name of the variable	Description
step	Maps a unit of time in the real world. 1 step is 1 hour of time.
type	Indicates the type of transaction. This can be CASH-IN, CASH-OUT, DEBIT, PAYMENT or TRANSFER
amount	amount of the transaction in local currency
nameOrig	identifier of the customer who started the transaction
oldbalanceOrg	initial balance of the originator before the transaction
newbalanceOrg	originator's balance after the transaction
nameDest	identifier of the recipient who received the transaction
oldbalanceDest	initial balance of the recipient before the transaction
newbalanceDest	recipient's balance after the transaction
isFraud	indicates whether the transaction is actually fraudulent or not. The value 1 indicates fraud and 0 indicates non-fraud

>First, we removed Unwanted Columns

```
In [5]: #Removing the unwanted columns
        df = df.drop(['nameOrig', 'nameDest'], axis=1)
```

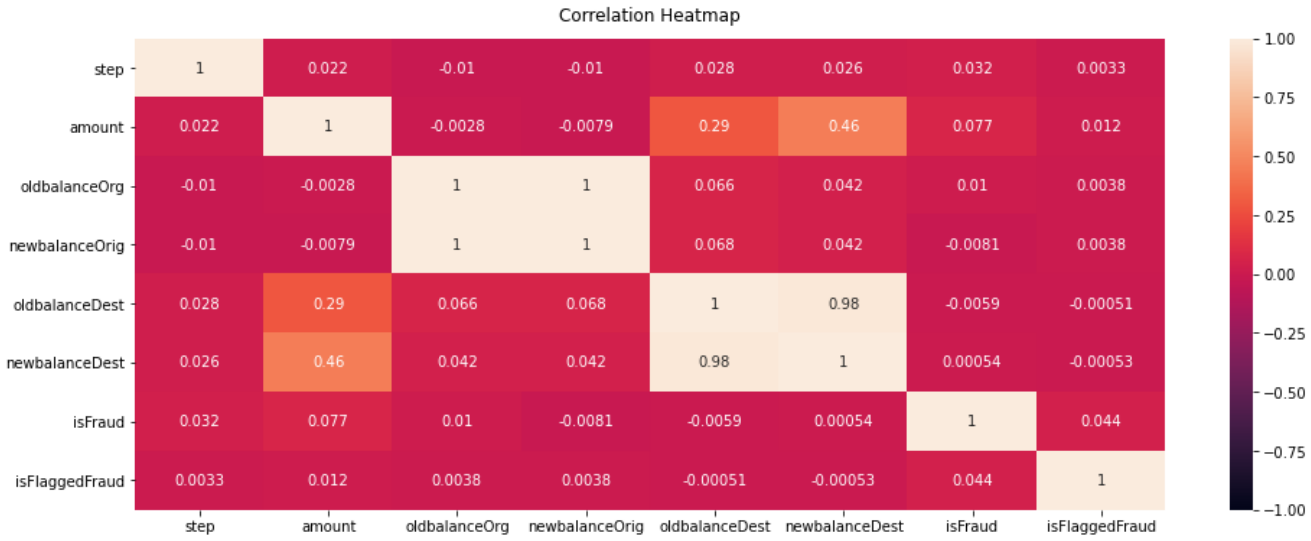
>Checked for Datatypes

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 6362620 entries, 0 to 6362619
Data columns (total 9 columns):
 #   Column          Dtype
---  ------          -----
 0   step            int64
 1   type            object
 2   amount          float64
 3   oldbalanceOrg   float64
 4   newbalanceOrig  float64
 5   oldbalanceDest  float64
 6   newbalanceDest  float64
 7   isFraud         int64
 8   isFlaggedFraud  int64
dtypes: float64(5), int64(3), object(1)
memory usage: 436.9+ MB
```

DATA VISUALIZATION:

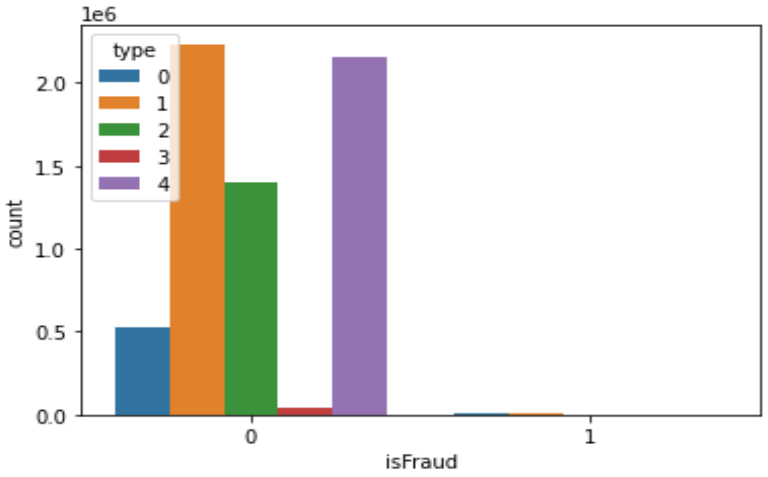
Data visualization is the graphical representation of information and data. By using visual elements like charts, graphs, and maps, data visualization tools provide an accessible way to see and understand trends, outliers, and patterns in data.

>Correlation Plot



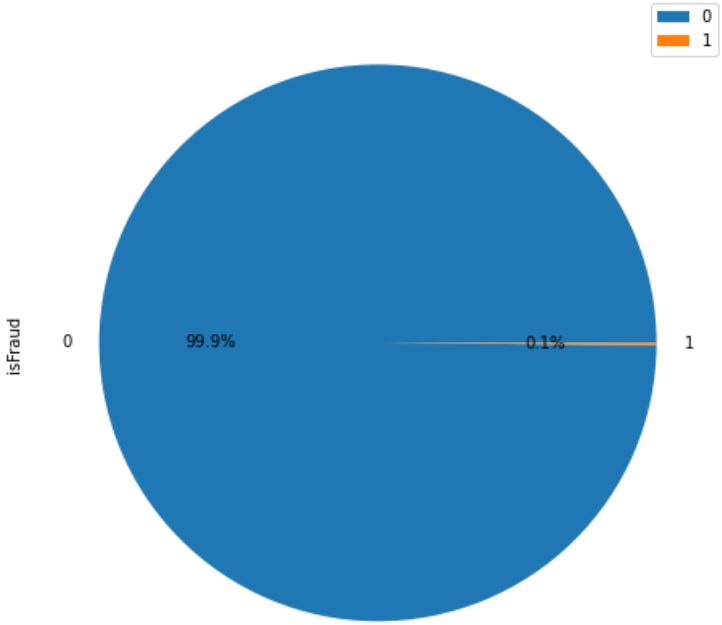
As we can see from above that there is no alarming correlation between the features.

>Function Count Plot



From the above visualization we see that the fraud happens only in cash_out and transfer transactional types.

>Pie Chart



From the above pie chart, we see only 1% of the total transactions are fraud.

FEATURE ENGINEERING:

>Creating an Errors column to check for errors in balance calculation in both Origin and Destination.

>Checked for null values.

DATA PRE-PROCESSING:

> Making a dataset with only CASH_OUT and TRANSFER medium of transaction since fraud happens with only these two mediums of transfer.

> Defining X and Y to generate the model.

>Installed imblearn to import SMOTE. SMOTE(Synthetic Minority Over-sampling Technique) is used to balance the dataset.

A. Random Forest: Random Forest is an ensemble learning method for classification and regression by constructing multiple decision trees for training and outputting the classification or prediction(regression). The goal of Random Forest is to combine weak learning models into strong learning models. We can summarize the algorithm of Random Forest in 4 steps:

Step 1: Randomly draw M bootstrap samples from the training set with a replacement.

Step 2: Grow a decision tree from the bootstrap samples. At each node: Randomly select K features without replacement and split the node by finding the best cut among the selected features that maximize the information gain.

Step 3: Repeat steps 1 and 2 T times to get T trees.

Step 4: Aggregate the predictions made by different trees via the majority vote

We used the Random Forest Model because it is a meta estimator that fits several decision tree classifiers on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and control over-fitting.

>In the first step, we split the data into test and train data.

>We defined the model using the RandomForestClassifier method.

>Scaled our train and test data in independent variables.

>Performed fitting and prediction of test data

>Our Precision-Recall Curve Score is 0.9979519234349783. This score of the model tells us how efficiently the values are predicted.

>Area Under the Curve score is 0.9994657197133419. This score of the model is the measure of the ability of a classifier to distinguish between classes and is used as a summary of the ROC curve. The higher the AUC, the better the performance of the model at distinguishing between the positive and negative classes.

>F1_score is 0.9979514629167612. The F1-score combines the precision and recall of a classifier into a single metric by taking their harmonic mean. It is primarily used to compare the performance of two classifiers. Suppose that classifier A has a higher recall, and classifier B has higher precision.

>Confusion Matrix of Random Forest Classifier

```
[[828937      5]
[  3382 824994]]
```

828937 true negative predictions: Ones predicted correctly as non-fraud.
3382 false-negative predictions: These are the ones wrongly predicted.
5 false-positive predictions: These are ones predicted incorrectly as fraud
824994 true positive predictions: These are ones predicted correctly as fraud.

B. Logistic Regression: Logistic Regression is supervised learning that computes the probabilities for classification problems with two outcomes. We can also extend it to predict several classes. In the Logistic Regression model, we apply the sigmoid function, which is

$$\sigma(z) = \frac{1}{1 + e^{-z}}$$

We used logistic for the second model because it belongs to the group of linear classifiers, while Random Forest is a multiclass classifier. Logistic regression is fast and relatively uncomplicated. It's essentially a method for binary classification but could be used for multiclass problems as well. It uses a sigmoid function to convert the outcome into categorical values from the logit function.

>First, we define the model using the LogisticRegression() function.

>Then we predict the test set results and calculate the accuracy.

>Our Precision-Recall Curve Score is 0.9478365105757705. This score of the model tells us how efficiently the values are predicted.

>F1_score is 0.9473648612741616. The F1-score combines the precision and recall of a classifier into a single metric by taking their harmonic mean.

>Confusion Matrix of Logistic Regression
[[785774 43168]
[43992 784384]]

785774 true negative predictions: Ones predicted correctly as non-fraud.

43992 false-negative predictions: These are the ones wrongly predicted.

43168 false-positive prediction: These are ones predicted incorrectly as fraud

784384 true positive predictions: These are ones predicted correctly as fraud.

RESULTS:

Since our project is a classification problem, we use accuracy, precision, recall, and F1 score to evaluate the models. We would like to introduce the meaning of TP, FP, TN, and FN. A true positive (TP) is a positive outcome predicted by the model correctly, while a false positive (FP) is a positive outcome predicted by the model incorrectly. Here is the table of results of different methods and we will talk about each evaluation of methods.

METHODS	PRECISION	RECALL	F1 SCORE
RANDOM FOREST	0.9999939393865932	0.9959173129110452	0.9979514629167612
LOGISTIC REGRESSION	0.9478365105757705	0.946893681130308	0.9473648612741616

CONCLUSION:

Finally, we developed a framework for detecting fraudulent transactions in financial data that worked well. This framework will help comprehend the complexities of fraud detection, such as the construction of derived variables that may assist in class separation, addressing the class imbalance, and selecting the machine learning method.

We tried out Logistic Regression and Random Forest as machine learning techniques. Considering the F1 Score of both the models, the Random Forest approach outperformed Logistic Regression with an F1 score very close to 1, suggesting that tree-based algorithms are effective for transaction data with well-defined classes. This highlights the importance of undertaking a thorough exploratory analysis to fully comprehend the data before constructing machine learning models. We discovered a few factors that distinguished the classes better than the raw data through our exploratory research.