CIS 5570 BIG DATA

FRAUD DETECTION ANALYSIS ON MOBILE MONEY TRANSACTIONS

Abstract:

Financial frauds have been growing with the advent of digital technology and the surge in online transactions. These fraudulent activities pose a significant threat to businesses, organizations, and individuals, leading to financial losses and undermining their trust in the financial institutions. With this project we look at the mechanisms of these fraud transactions and the challenges in detecting them. We explore the role of big data analytics and machine learning in finding financial fraud. These technologies help us to analyze large amounts of transaction data to identify patterns and anomalies that may indicate fraudulent transactions.

Introduction:

With the rapid increase in mobile transactions, financial fraud has emerged as a significant concern for businesses, organizations, and individuals. This project aims to address this issue by performing an exploratory analysis to detect and prevent fraudulent activities based mobile transaction data. We use the power of PySpark to process large volumes of data and engineer relevant features. We use Spark ML algorithms such as Logistic Regression and Principal Component Analysis to predict fraudulent transactions. The performance of these models is evaluated using metrics like recall and precision. This project highlights the importance of data analysis, feature engineering and model selection in achieving high accuracy in identifying fraudulent transactions.

Dataset:

The Dataset used in this project is a representative sample of real transactions logs collected over a month from PaySim, a mobile money service operating in a

African country. This dataset contains binary flags indicating whether a transaction is flagged as fraudulent or not. This Dataset includes following attributes:

Step: Maps a unit of time in the real world. In this case 1 step is 1 hour of time.

Type: CASH-IN, CASH-OUT, DEBIT, PAYMENT and TRANSFER.

Amount: Amount of the transaction in local currency.

NameOrig: Customer who started the transaction

OldbalanceOrg: Initial balance before the transaction

NewbalanceOrig: New balance after the transaction.

NameDest: Customer who is the recipient of the transaction

OldbalanceDest: Initial balance recipient before the transaction

NewbalanceDest: New balance recipient after the transaction.

IsFraud: This is the transactions made by the fraudulent agents inside the simulation. In this specific dataset the fraudulent behavior of the agents aims to profit by taking control or customers' accounts and trying to empty the funds by transferring to another account and then cashing out of the system.

IsFlaggedFraud: The business model aims to control massive transfers from one account to another and flags illegal attempts. An illegal attempt in this dataset is an attempt to transfer more than 200.000 in a single transaction.

- The size of the dataset is approximately around 470MB.

step	type	amount	nameOrig	oldbalanceOrg	newbalanceOrig	nameDest	oldbalanceDest	newbalanceDest	isFraud	isFlaggedFraud
1	PAYMENT	9839.64	C1231006815	170136.0	160296.36	M1979787155	0.0	0.0	0	0
1	PAYMENT	1864.28	C1666544295	21249.0	19384.72	M2044282225	0.0	0.0	0	0
1	TRANSFER	181.0	C1305486145	181.0	0.0	C553264065	0.0	0.0	1	0
1	CASH_OUT	181.0	C840083671	181.0	0.0	C38997010	21182.0	0.0	1	0
1	PAYMENT	11668.14	C2048537720	41554.0	29885.86	M1230701703	0.0	0.0	0	0

Methodology:

This project is executed in Google Colab

1. Installed Spark and configured the session on Google Colab.
2. We have uploaded the transaction dataset(.csv) in google collab's drive. This (.csv) file is loaded into a data frame in spark.
3. Understanding the data: We have conducted data analysis on the dataset to understand the dataset's characteristics, distributions, and potential data quality issues. Here we have encountered a outlier.

4. The number of actual fraud transactions in the “isFraud” columns is not equal to the number of fraud transactions flagged by the business model “isFlaggedFraud”
5. Feature Engineering: We have performed feature engineering to extract relevant information like:
6. Created a new attribute by adding conditional statements.
7. Splitting the dataset into train and test sets using randomsplit.
8. Performed StringIndexer to convert the categorical column into variable column and also converted the type column into numerical column.
9. Performed one-hot encoding to transform the categorical variables into a format that can be provided to PySpark ML algorithms more effectively. In the resulting encoding vectors, each category is assigned to a binary value (0 or 1), indicating 1 as the presence of the category and 0 indicating the absence of the category.
10. Used feature transformer called ‘vector assembler’ to combine multiple feature columns into a single vector column.
11. **Created a pipeline to chain stringindexer,one-hot encoder and vectorAssembler stages.Fit pipeline on the training data to preprocess and created feature vectors.**
12. Trained a Logistic regression model on preprocessed training data with feature vectors and target variable.Evaluated the trained model on the test data to preprocess and calculated area under the ROC.
13. Principal Component Analysis (PCA) is done on the dataset to reduce the number of features while preserving most of the variance.

Contributions and Responsibilities:

1.Ravi Kiran Puramsetti:

- Responsible for Data collection and data analysis
- Involved in final presentation for the project

2. Teja Sai Pranav Kidambi:

- Responsible for Data preprocessing and PCA
- Involved in documentation for the project

3.Hemanth Sai Prasanth Vaddadi:

- Responsible for implementing data pipeline and Logistic regression model

- Involved in final presentation for the project

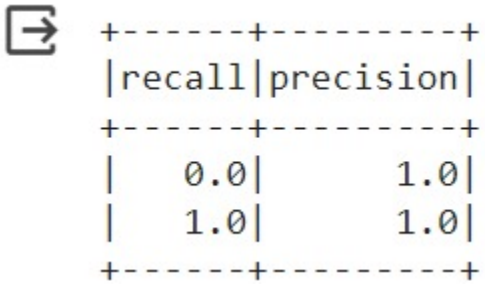
4. Siva Nishwanth Musamalla:

- Responsible for model evaluation
- Involved in documentation for the project

Results:

Model performance for the Logistic regression is measured by recall and precision.We observed recall 1 and precision 1 indicates that all positive predictions made by the model are correct.

Model performance for PCA is measured using Area under ROC the principal components derived from the data were highly effective at separating fraudulent from legitimate transactions. We can see that ROC score of 1 indicates that the model perfectly separates positive and negative instances.



recall	precision
0.0	1.0
1.0	1.0

```
Area Under ROC for PCA-Transformed Data: 1.0
```

References:

1. https://journalofbigdata.springeropen.com/articles/10.1186/s40537-022-00573-8
2. https://onlinelibrary.wiley.com/doi/full/10.1111/joes.12372
3. https://link.springer.com/article/10.1007/s11042-023-16068-4
4. https://www.researchgate.net/publication/313138956_PAYSIM_A_FINANCIAL_MOBILE_MONEY_SIMULATOR_FOR_FRAUD_DETECTION
5. **Dataset**: https://www.kaggle.com/datasets/ealaxi/paysim1