Fraud Detection

Introduction

The fraud detection dataset obtained from a sandbox provides a comprehensive collection of data related to financial transactions and associated entities, offering valuable insights into fraudulent activities within the financial domain. With an extensive array of node labels representing various entities and actions involved in transactions, this dataset serves as a rich source of information for researchers, data scientists, and organizations aiming to develop robust fraud detection systems.

Objective:

The primary objective of utilizing the fraud detection dataset is to develop and evaluate effective fraud detection algorithms and systems within the financial domain. Develop machine learning models, anomaly detection algorithms, and statistical techniques to accurately identify and flag fraudulent transactions based on patterns, anomalies, and risk indicators within the dataset. Extract meaningful features from the transactional data, client information, and merchant details to enhance the predictive power and performance of fraud detection models. Evaluate the performance of fraud detection algorithms against ground truth labels within the dataset, benchmarking accuracy, precision, recall, and false positive rates to assess effectiveness and reliability.

Analysis 1

```
MATCH (c:Client)-[:PERFORMED]→(t:Transaction)
RETURN c.name, COUNT(t) AS num_transactions
ORDER BY num_transactions DESC
LIMIT 10;
```

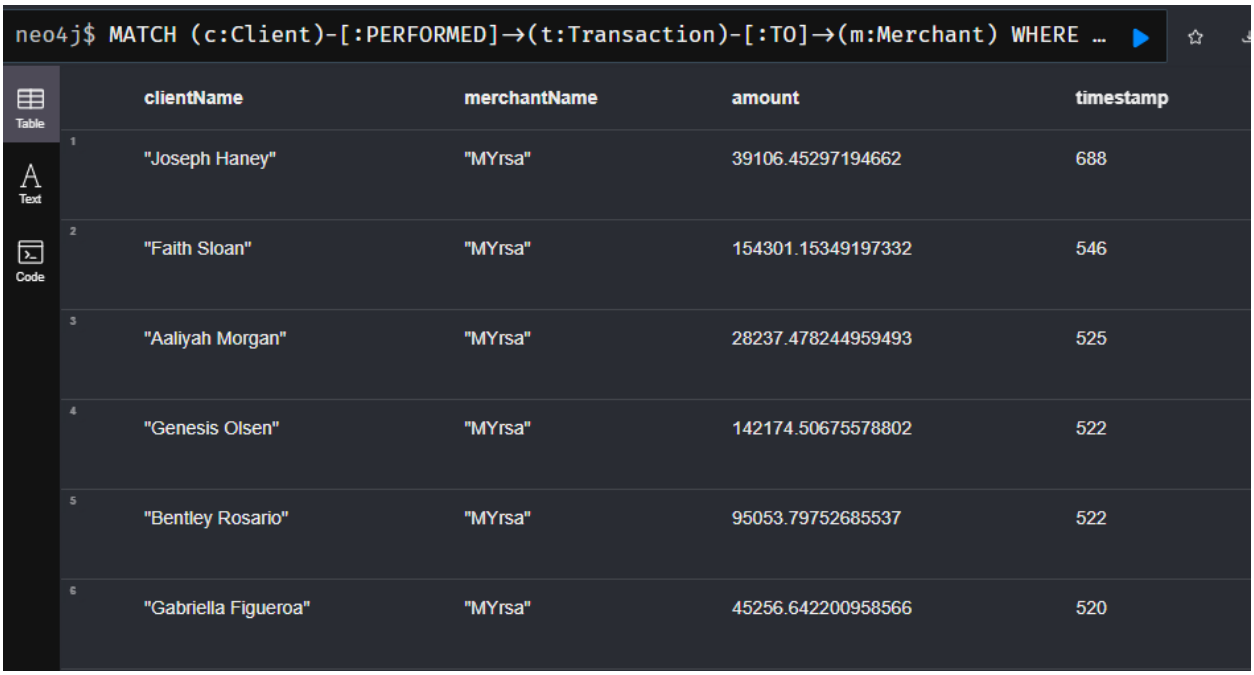
Table | Text | Code

	c.name	num_transactions
1	"Daniel Hendrix"	1933
2	"Isabella Grant"	1547
3	"Aubree David"	1365
4	"Evelyn Craig"	1171
5	"Andrea Sweet"	1057
6	"Michael Cooper"	1039
7		

Started streaming 10 records after 11 ms and completed after 194 ms.

This Analysis shows the names of clients along with the count of transactions each client has performed. This is useful for analyzing which clients are the most active or have the highest transaction volumes in the database.

Analysis 2



	clientName	merchantName	amount	timestamp
1	"Joseph Haney"	"MYrsa"	39106.45297194662	688
2	"Faith Sloan"	"MYrsa"	154301.15349197332	546
3	"Aaliyah Morgan"	"MYrsa"	28237.478244959493	525
4	"Genesis Olsen"	"MYrsa"	142174.50675578802	522
5	"Bentley Rosario"	"MYrsa"	95053.79752685537	522
6	"Gabriella Figueroa"	"MYrsa"	45256.642200958566	520

This analysis shows the names of Merchants along with the count of unique clients associated through their performed transactions and shows the total amount. This information can be valuable for analyzing the distribution of clients with identifying the number of occurrences with the largest customer bases.

Analysis 3

```
1 MATCH (c:Client)-[:PERFORMED]→(t:Transaction {fraud: true})
2 RETURN DISTINCT c.name AS clientName, c.id AS clientId
3 LIMIT 10
4
5 // Clients Involved in Fraudulent Activities:
```

Table | Text | Code

	clientName	clientId
1	"Landon Adams"	"4287186486553145"
2	"Landon Sandoval"	"4234798486577769"
3	"Aaliyah Sharpe"	"4361287590543243"
4	"Ashley Chapman"	"4833833649287561"
5	"Scarlett Coffey"	"4599940846994966"
6	"Autumn Petty"	"4890138988127380"
7	"Melanie Kent"	"4281501376120050"

Started streaming 10 records after 9 ms and completed after 23 ms.

This analysis shows List of clients with their ClientID which performed the Fraud Transaction.

Analysis 4

```
MATCH (c:Client)-[:PERFORMED]→(:Transaction)-[:TO]→(b:Bank)
RETURN b.name, COUNT(DISTINCT c) AS num_clients
ORDER BY num_clients DESC
LIMIT 5;
```

Table | Text | Code

	b.name	num_clients
1	"Bank of Burt"	483
2	"Bank of Mccoy"	446
3	"Bank of Mckinney"	410

Started streaming 3 records after 12 ms and completed after 26 ms.

This query retrieves all transactions flagged as fraudulent (where the "fraud" property is set to true) and shows the bank names with the highest number of fraud occurrences first. This query helps identify and analyze patterns of fraudulent behavior within the transaction data.

Analysis 5

```
MATCH (c:Client)-[:PERFORMED]→(t:Transaction)
WITH c, count(t) AS totalTransactions, sum(t.amount) AS totalAmount
WHERE totalTransactions > 100 AND totalAmount > 10000
RETURN c.name AS clientName, c.id AS clientId, totalTransactions, totalAmount
LIMIT 10

// Potential Fraudulent Clients:
```

Table | Text | Code

	clientName	clientId	totalTransactions	totalAmount
1	"Bentley Peck"	"4997933060327094"	825	119620267.9859122
2	"Dominic Boyer"	"4776276949898423"	355	55251727.086229675
3	"Faith Dotson"	"4858607188760216"	238	30829964.756352518
4	"Landon Adams"	"4287186486553145"	294	44373796.48252735
5	"Lauren Mack"	"4661202154682409"	252	40010557.763853095
6	"Kylie Barron"	"4649268238636650"	287	42067421.26231002

Started streaming 10 records after 18 ms and completed after 56 ms.

This analysis shows the name of client along with the ClientID and it shows the number of transactions which flagged as Fraud and Total amount of transactions they had done.

Machine Learning Algorithm

1. Linear Regression

A simple predictive modelling method called linear regression is used to ascertain the correlation between independent variables, or features, and dependent variables, or outcomes. It aims to predict numerical outcomes by fitting a line to the data points in a way that minimizes the gap between the data points and the line.

Use in Project: Using categorical data converted into a numeric format, we used it to forecast crime rates depending on the kind of locale.

2. Regression with Random Forest

A more sophisticated model called Random Forest creates several decision trees to improve forecast accuracy. To increase accuracy and reduce over-fitting—the phenomenon where a model is too closely fitted to a small number of data points—it integrates the predictions of many trees.

● Application in Project: It is recommended as an enhancement to better manage complicated data and forecasts crime rates by more thoroughly evaluating the effects of different features.

Model Training using Python.

```python
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
import matplotlib.pyplot as plt

# Load the data
data = pd.read_csv('/content/export.csv')

# Assuming the rows in your dataset are in a sequential order
data['sequence'] = range(len(data))

# Feature and target variable
X = data[['sequence']]  # Features (here, just the sequence number)
y = data['num_clients']  # Target variable

# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=0)

# Create a linear regression model
model = LinearRegression()

# Fit the model
model.fit(X_train, y_train)
```

```python
# Fit the model
model.fit(X_train, y_train)

# Make predictions
predictions = model.predict(X_test)

# To predict future points, increase the sequence number
future_sequence = np.array([[len(data) + i] for i in range(1, 6)])  # Predict the next 5 points
future_predictions = model.predict(future_sequence)

# Plotting
plt.figure(figsize=(10, 6))
plt.scatter(X_train, y_train, color='blue', label='Training data')
plt.scatter(X_test, y_test, color='green', label='Testing data')
plt.plot(X_train, model.predict(X_train), color='red', label='Model')
plt.scatter(future_sequence, future_predictions, color='orange', label='Future predictions')
plt.title('Linear Regression Model for Transaction Prediction')
plt.xlabel('Sequence')
plt.ylabel('Number of Transaction')
plt.legend()
plt.show()
```

Model Explanation

Loading the Data: The script starts by loading a CSV file into a Pandas Data Frame. This file contains the data that will be used for both training and predicting.

Adding Sequence Numbers: A new column, 'sequence', is added to the DataFrame. This sequence number serves as the feature (independent variable) for the linear regression, representing the order of the data points.

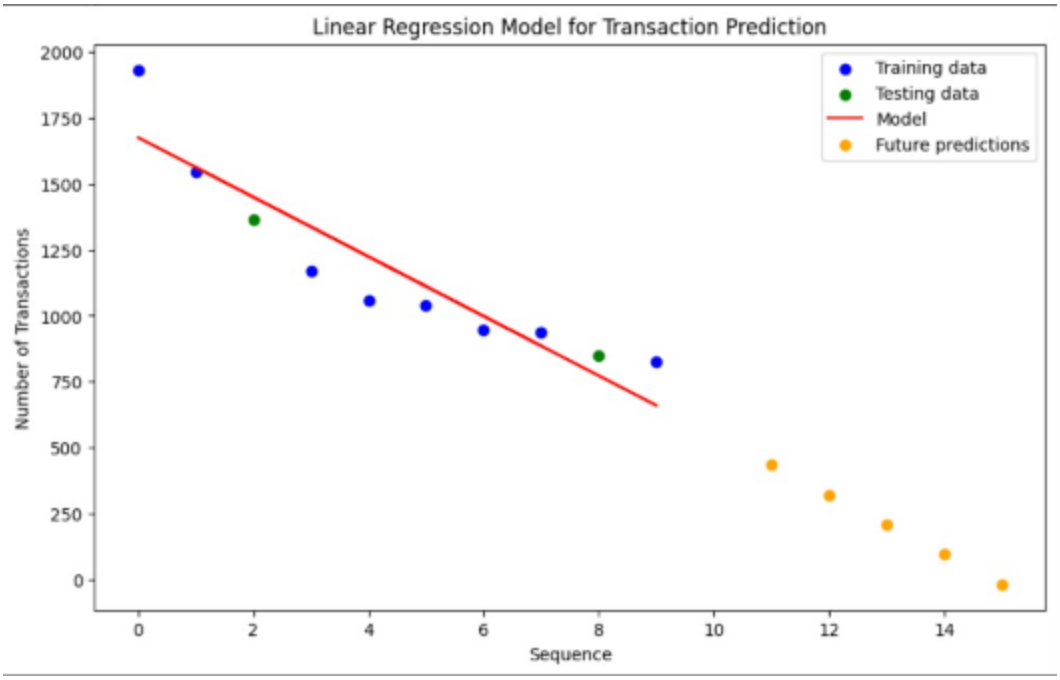
Setting Up Features and Target: The 'sequence' column is used as the feature (X), while another column, presumably 'num_clients', is used as the target variable (y). This target variable represents the count or number of clients, which the model aims to predict.

Splitting the Data: The data is split into training and testing sets. The training set is used to fit the model, and the testing set is used to evaluate its performance. Here, 80% of the data is used for training, and 20% is for testing, controlled by the test_size=0.2 parameter.

Creating and Training the Model: A linear regression model is instantiated and trained (fitted) using the training data. This involves finding the best-fit line that minimizes the differences (errors) between predicted and actual values in the training data.

Making Predictions: The model makes predictions on the testing set to evaluate its accuracy. It's also used to predict future values based on extending the sequence number beyond the current data.

Graph



Plotting: The graph is plotted with the following elements:

Blue dots represent the training data, showing the relationship between the sequence numbers and the actual number of clients in the training set.

Green dots show the testing data, indicating how the model performs against unseen data.

Red line is the best-fit line produced by the linear regression model over the training dataset, showing the trend that the model has learned.

Orange dots represent the model's predictions for future data points, extending five steps beyond the existing data based on the learned trend.

Graph Labels and Legend: The graph includes labels for the axes—'Sequence' for the x-axis and 'Number of Clients' for the y-axis. A legend is also provided to help identify each component of the plot.

Graph Interpretation

The red line (model) indicates the trend that the linear regression has identified. If the line fits well with both blue and green dots, the model can be considered effective for this data under linear assumptions.

The position and spread of the green dots relative to the red line show how well the model predicts new, unseen data. Close alignment indicates good model performance.

The orange dots show the model's extension into the future. They represent the model’s expectation of client numbers if the current trend continues.

This kind of model and graph is useful for forecasting future trends based on historical data, assuming the relationship between the sequence number and the target variable remains consistent and linear.