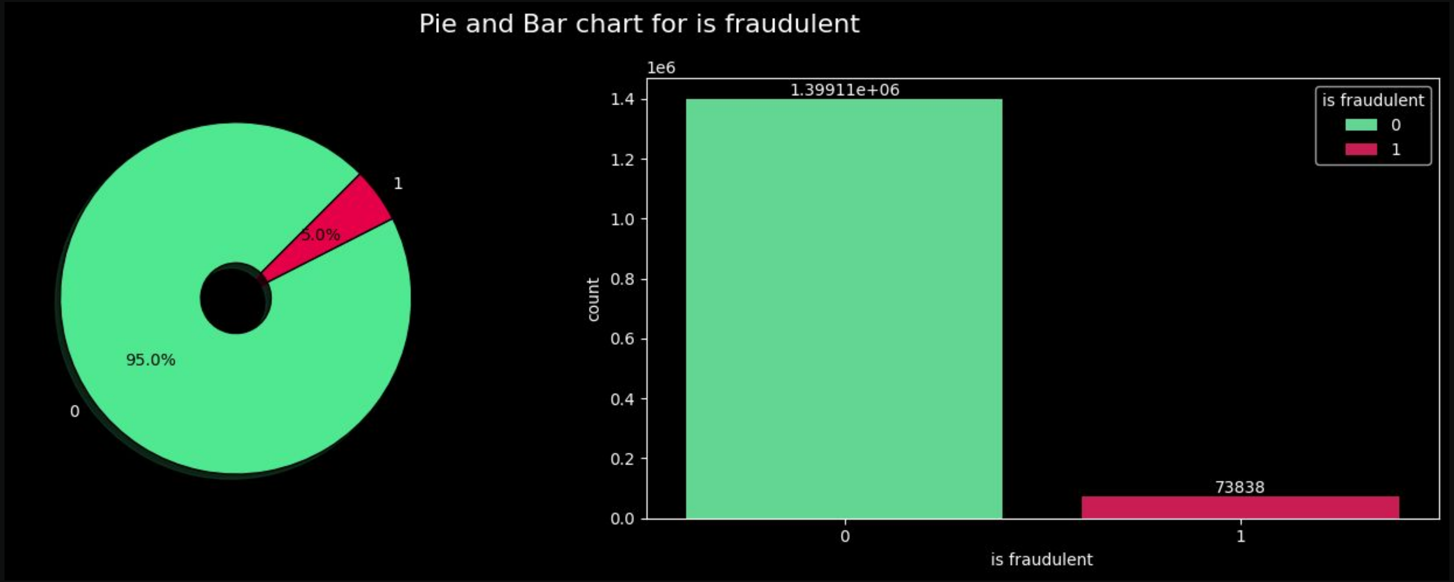
E-commerce Fraud Transactions

Presented by: Hugo Milesi
Last Updated: June 19th, 2024

Overview

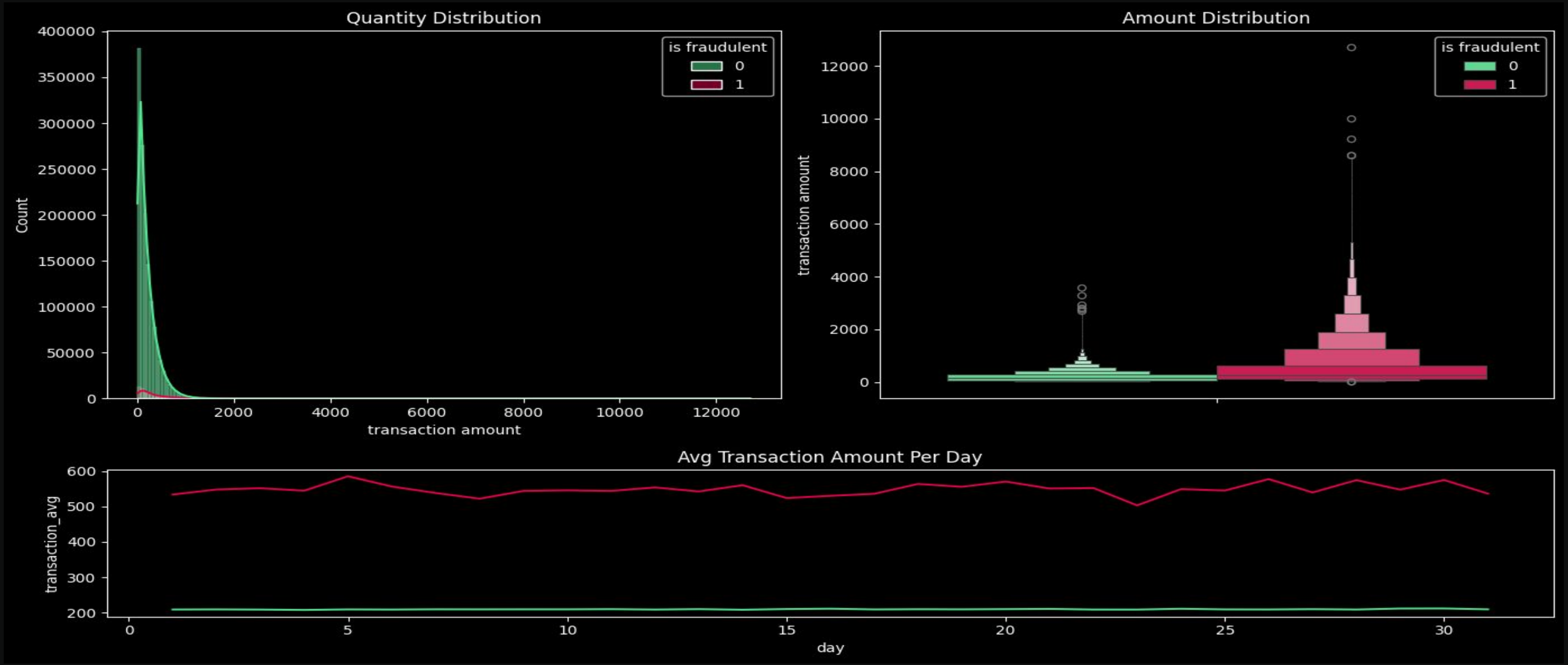
- Analyze E-commerce company data to predict whether or not a transaction is fraudulent or not.
- The data has 1472952 unique transactions (16 features).
  - Is fraudulent (0 or 1)
  - Transaction information (transaction amount, date, quantity, etc)
  - Demographic information (customer age, location)
- Source: Kaggle

Target Variable - Is Fraudulent



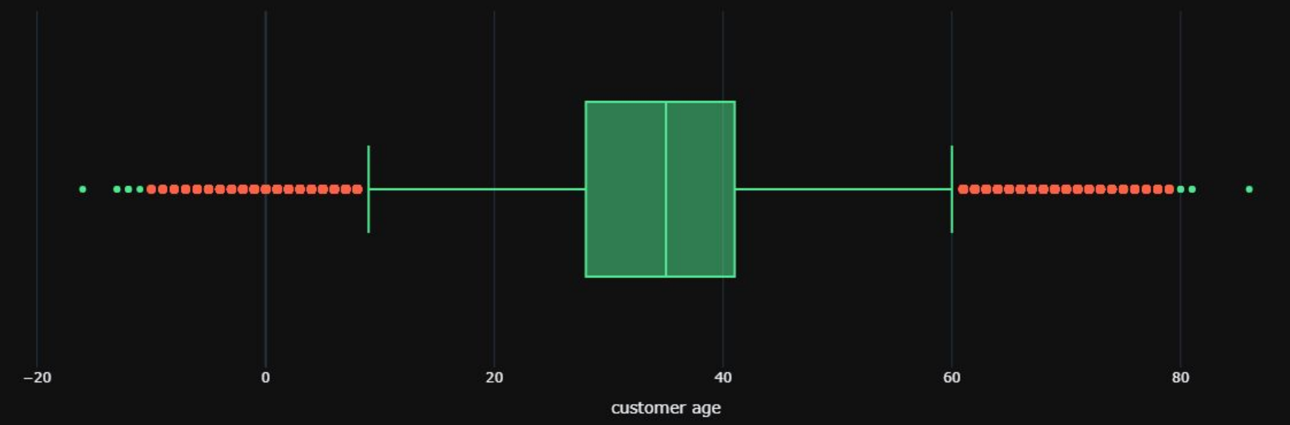
- **Imbalance target**: The dataset is highly imbalanced with only 5% of the transactions being fraudulent.
- To approach this problem, i will use an oversampling technique callet SMOTE.

Distribution of Transaction Values



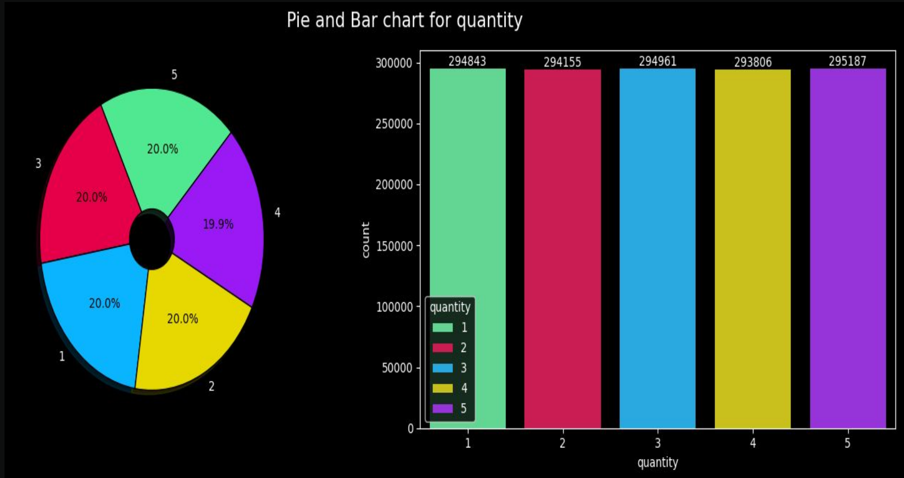
- Transaction amount exceeding 4000 are predominantly fraudulent.
- Fraudulent Transactions tend to have higher amounts.

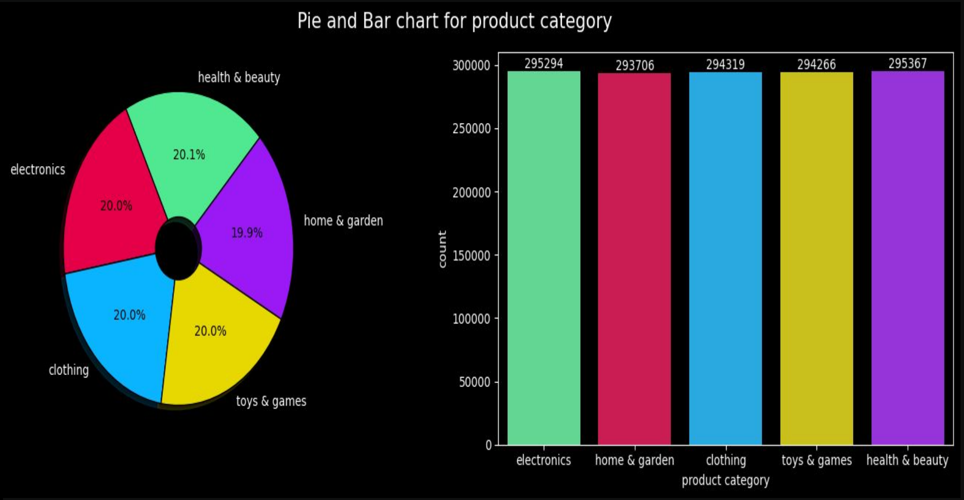
Customer Age

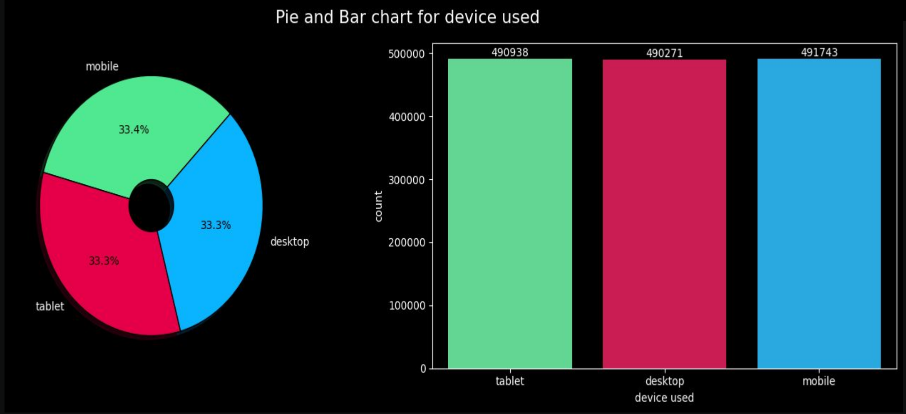


- Given the summary of the boxplot for customer age, it is clear that the data contains erroneous values.
- The lower fence outliers indicate ages smaller than 9 and impossible values like negative ages.

Quantity, Product Category and Device Used

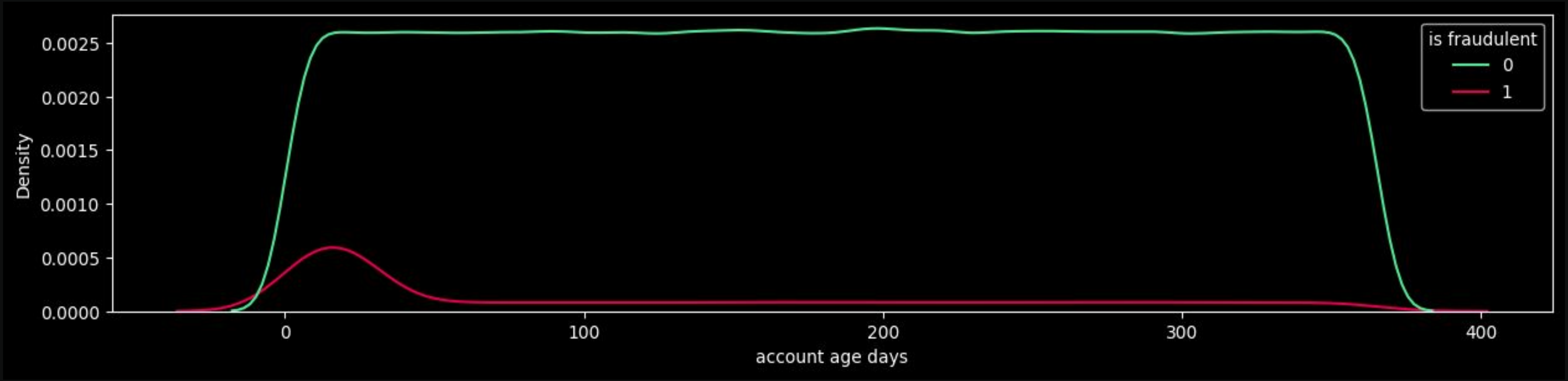






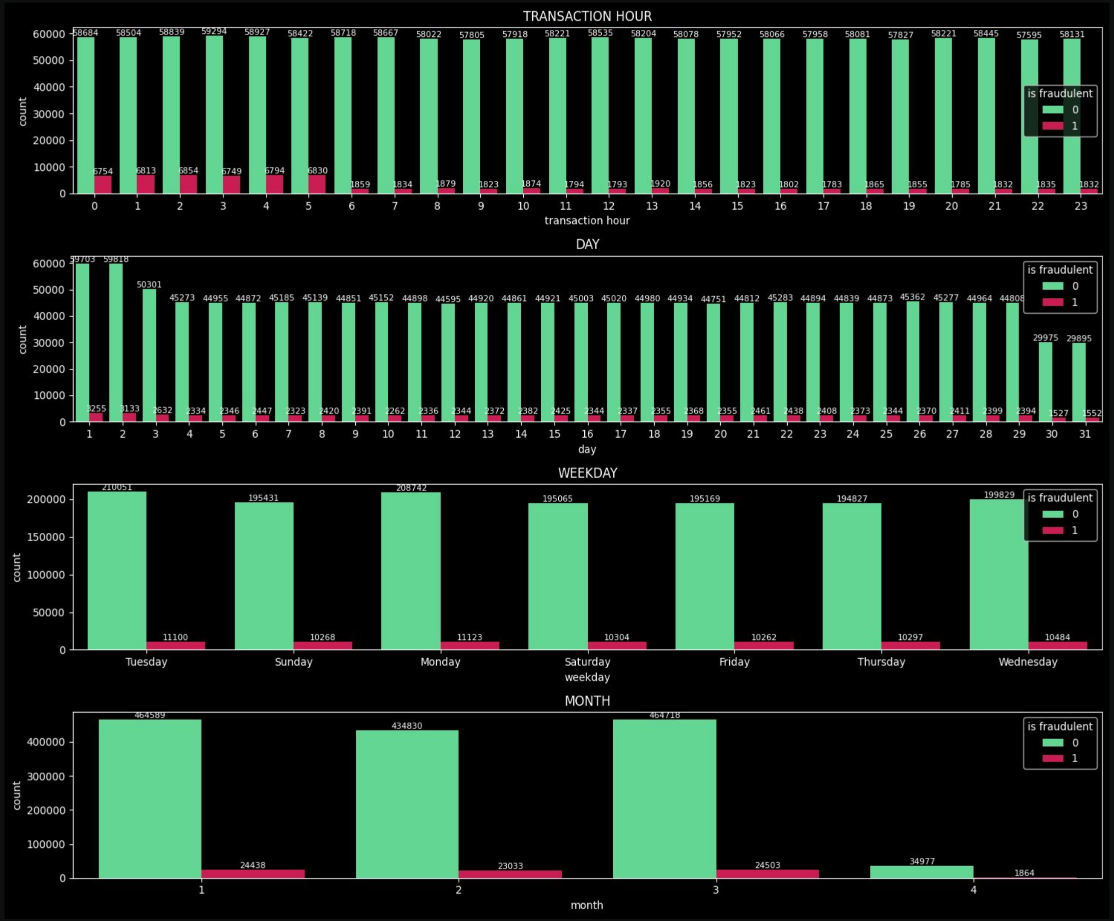
- Data Categories are evenly distributed.

Distribution of Account Age (days)



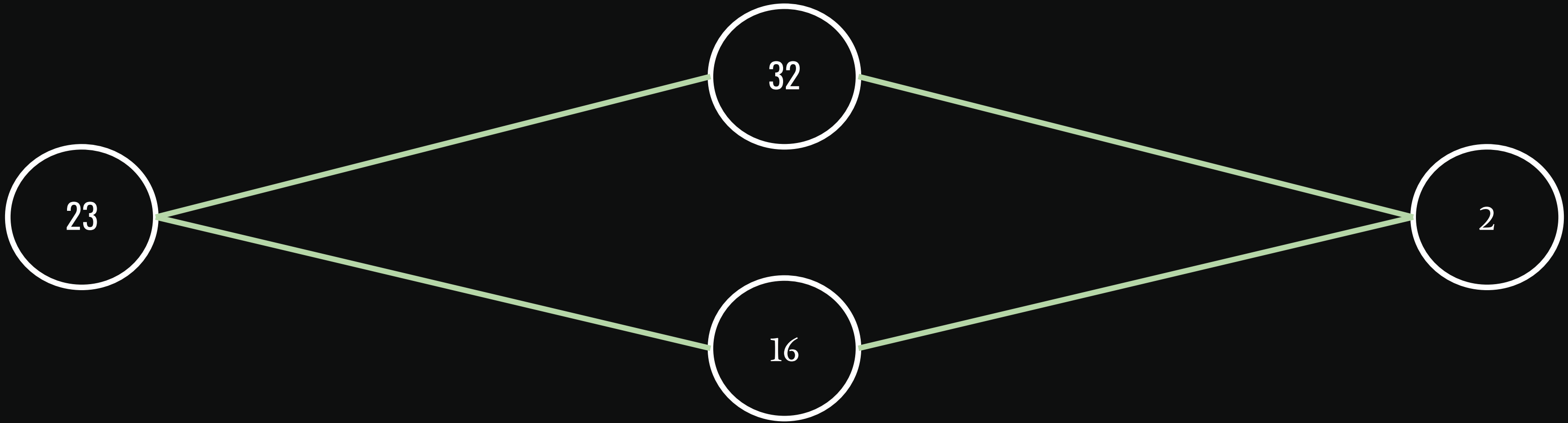
- Accounts created recently exhibit a higher tendency for fraudulent activity.

Time Frames



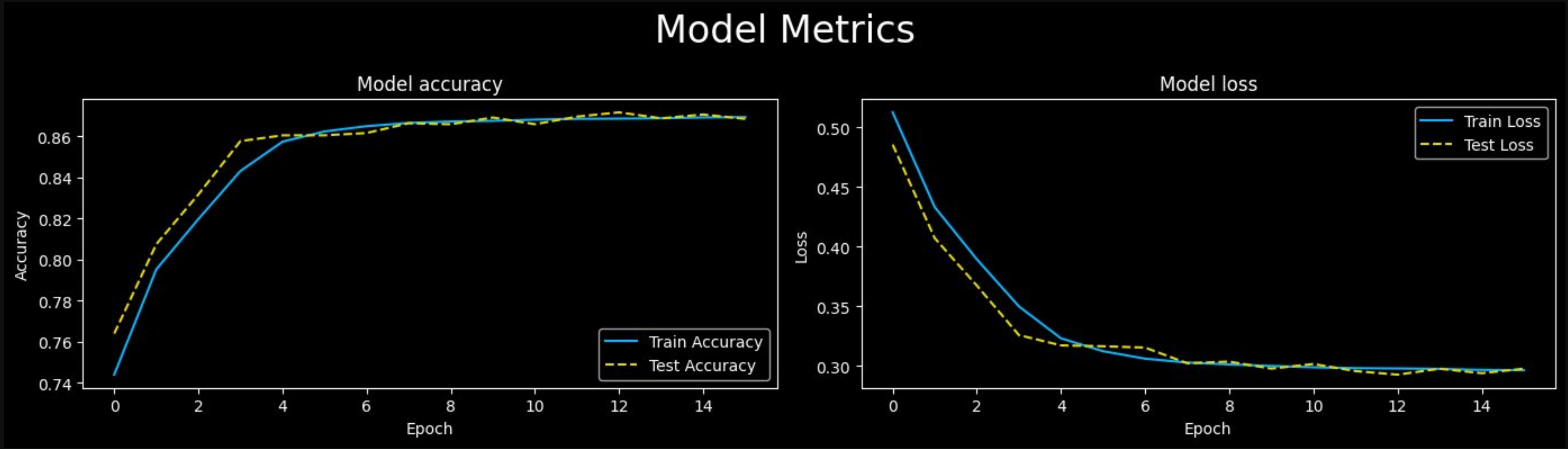
- Hours 0 to 4 exhibit a higher frequency of fraudulent transactions.
- Day 1 and 2 have higher number of transactions.
- Even distribution for weekday.
- Low register for month 4.

Model Explanation



- Fully connected neural network with 2 hidden layers to classify the transactions.
- The neural network has 23 input nodes (the number of all the features of the dataset), 32 nodes in the first hidden layer, 16 in the second hidden layer and 2 output nodes for each class (fraudulent and non-fraudulent).

Loss and Accuracy Results



- The training loss and test loss shows that the model was able to converge and learn from the training data without overfitting.
- Similarly, the training and test accuracy shows that the model was able to achieve good accuracy on both data.
- Accuracy plateaued around 87% with a loss score of 2977.

Test and Validation Results

Test Results

	precision	recall	f1-score	support
non-fraudulent	0.83	0.93	0.88	279823
Fraudulent	0.93	0.81	0.86	279823
accuracy			0.87	559646
macro avg	0.88	0.87	0.87	559646
weighted avg	0.88	0.87	0.87	559646

Validation Results

	precision	recall	f1-score	support
non-fraudulent	0.92	0.94	0.93	22412
Fraudulent	0.93	0.92	0.93	22412
accuracy			0.93	44824
macro avg	0.93	0.93	0.93	44824
weighted avg	0.93	0.93	0.93	44824

- **Perform comparison:** The model performs slightly better on the validation set compared to the test set, with higher precision, recall and F1-scores.
- **Precision and Recall Trade-off:** For fraudulent transactions, precision is high, meaning some fraudulent transactions might not be detected. For non-fraudulent transactions, the opposite is true, with slightly lower precision and higher recall.

Conclusions

- **Imbalance dataset**: The target variable is highly imbalanced with only 5% of the transactions being fraudulent.
- Fraudulent Transactions tend to have higher amounts.
- Accounts created recently exhibit a higher tendency for fraudulent activity.
- Hours 0 to 4 exhibit a higher frequency of fraudulent transactions.
- The model is composed of four layers. The input layer contains 23 nodes, corresponding to each feature. It includes two hidden layers, and the output layer consists of 2 nodes.
- Model Accuracy plateaued around 87% with a loss score of 2977.
- The slight differences between the test and validation results suggest that the model is well-tuned but should still be monitored for any signs of overfitting or underfitting in future data.