**Software Development for Artificial Intelligence**

**Project 1**

**Credit Card Fraud Detection**

I choose Credit Card Fraud Detection dataset, there are 8000 transactions in this dataset, all of which have been painstakingly documented with full cardholder and merchant details, transaction amount, date and time, and other relevant facts. From encrypted card numbers to merchant category codes and transaction locations, it contains both numerical and categorical characteristics. Most importantly, it has a "Fraud Flag or Label" that indicates whether a transaction is fraudulent. It also has other metadata that includes information on the device and transaction source, giving analysts and investigators a thorough foundation for identifying and studying credit card fraud patterns.

**Pandas** helped us prepare and arrange our data for our project so that it would be more manageable. To make graphs that will improve our ability to observe and comprehend the data, we used **Seaborn** and **Matplotlib**. Specifically, **PyTorch** allowed us to construct and train a smart model to identify fraudulent transactions, which helped us with the labor-intensive task of fraud detection.

Importing seaborn and matplotlib to create charts, and pandas for data analysis, among other necessary packages for data management and visualization. A dataset called "credit\_card\_fraud.csv" is then loaded into a DataFrame **df** for additional processing and analysis.

A screenshot of a computer

Description automatically generated

Pandas in DataFrame that lists the number and percentage of missing (null) values in each of the df DataFrame's columns. It determines the total nulls in each column as well as their percentage in relation to the size of the dataset.

A screenshot of a computer

Description automatically generated

The graph shows the variation of transaction amounts in a dataset as a histogram layered with a kernel density estimate (KDE) curve. The frequency of transactions within different amount ranges is displayed on the y-axis, while the x-axis displays the transaction value. Many peaks where the amounts of transactions are more frequent are shown in the distribution, which seems to be multimodal. Observing trends and outliers in the transaction data is made possible by the utilization of 50 bins, which offers a comprehensive perspective of the distribution.

A graph of a diagram

Description automatically generated with medium confidence

The contrast of fraudulent and non-fraudulent transactions in a dataset is shown using a bar chart. The number of non-fraudulent transactions (label 0) and fraudulent transactions (label 1) are shown by two bars, each colored green and orange. The noteworthy variation in height implies a discrepancy in the occurrence of each class, signifying that non-fraudulent activities transpire considerably more frequently in the dataset than fraudulent ones.

A green and orange rectangular bars

Description automatically generated

Its transaction currency distribution is shown in a pie chart. It is separated into three primary parts, each of which represents a different currency (INR, USD, and EUR), accounting for around one-third of the transactions. The three groups are distributed rather evenly thanks to the chart's use of various hues to assist distinguish between the currencies.

A pie chart with numbers and a few percentages

Description automatically generated

The graph, which is a bar chart, shows how different transaction sources are distributed over a dataset. Two categories, "In-Person" and "Online," together with the number of transactions in each, are displayed. The counts of the two groups are similar, with "In-Person" transactions being slightly less than "Online" transactions, suggesting that the frequency of the various transaction types in this dataset is similar. The diagram provides a clear visual comparison of the two transaction source modalities.

A screenshot of a computer

Description automatically generated

**Preprocessing:**

By eliminating unnecessary columns that either contain sensitive data or aren't important for predictive modeling, this preprocessing script cleans the credit card fraud dataset. Rows with missing values are dropped while it searches for and addresses any missing data. One-hot encoding is used to convert categorical variables into numerical values and standardize the 'Transaction Amount' in order to improve model compatibility. The initial timestamp column is then removed when additional characteristics are created from the transaction timestamps to offer more context for the analysis, such as the day of the week and the hour of the day. This extensive preparation guarantees that the data is clear, instructive, and prepared for the next steps in the construction of the machine learning model.

Here is the data after completing the preprocessing steps.

A screenshot of a black screen

Description automatically generated

A screenshot of a computer

Description automatically generated

A dataset is ready for machine learning by running the code below. 'Fraud Flag or Label' is the target variable (y), and it first extracts the characteristics (X) from it. Next, with the random\_state parameter set to 42, it creates a consistent collection of random samples by splitting the dataset into a training set (80%) and a test set (20%) using the scikit-learn **train\_test\_split** function.

A computer code with colorful text

Description automatically generated with medium confidence

Ensures that the feature and label data from the train and test sets are of the right data type—float for features and long for labels—for model processing by converting them into PyTorch tensors. The final tensor forms are produced, verifying the dimensions: 6400 training samples with 19 characteristics apiece and 1600 testing samples with matching labels for each.

A screenshot of a computer program

Description automatically generated

The neural network class **FraudDetectionNetwork**, with two hidden layers and **ReLU** activation functions, is defined in the code below for PyTorch-based fraud detection. The model starts with only one output neuron that is appropriate for binary classification, a hidden layer with a predetermined size, and a source of input size. It utilizes the Adam optimizer to train the network at a learning rate of 0.001 and configures the binary cross-entropy with logits as the loss function.

A computer screen with text and images

Description automatically generated

Evaluates how well a neural network model that has been trained performs using test data. It makes predictions using the model, which are then transformed into binary forecasts by rounding and using the sigmoid function. A classification report and confusion matrix are generated by comparing the outcomes with the genuine labels. A heatmap is then used to illustrate how well the model predicts both fraudulent and non-fraudulent transactions.

For this project, we preprocessed a dataset of credit card transactions using Python. We did this by encoding categorical variables, addressing missing values, and eliminating extraneous columns. To anticipate fraudulent transactions, a neural network model was constructed and trained with PyTorch. Test data was then used to assess the model's performance, and metrics represented through a confusion matrix were used to gauge how well the model could predict outcomes.

**References:**

1. J. O. Awoyemi, A. O. Adetunmbi and S. A. Oluwadare, "Credit card fraud detection using machine learning techniques: A comparative analysis"
2. . Yan, Y. Li and J. He, "Comparison of Machine Learning and Neural Network Models on Fraud Detection"
3. Credit Card Fraud Detection Using Deep Learning by Vimal Kansal

<https://medium.com/@vkansal/credit-card-fraud-detection-using-deep-learning-pytorch-47f680a8c5be>