# CREDIT CARD FRAUD DETECTION

Project Report

2024

presented by: Team Epsilon

# Table of **Content**

# INTRODUCTION

In today's digital age, credit card transactions have become a cornerstone of modern commerce, facilitating millions of transactions globally every day. However, this widespread use has also given rise to a significant increase in fraudulent activities, posing severe financial risks to consumers and financial institutions alike. Credit card fraud not only leads to substantial financial losses but also erodes trust in digital payment systems, making it imperative to develop robust mechanisms for detecting and preventing fraud.

The objective of this project is to develop a sophisticated machine learning model capable of identifying and preventing credit card fraud in real-time. Leveraging a comprehensive dataset containing transaction details and associated labels indicating fraud, this project aims to implement advanced algorithms to accurately classify and predict fraudulent transactions. The dataset used comprises various features, including transaction ID, date, time, type of card, entry mode, amount, type of transaction, merchant group, country of transaction, shipping address, country of residence, gender, age, bank, and fraud status.

By employing state-of-the-art machine learning techniques, including data preprocessing, feature engineering, model training, and evaluation, this project seeks to enhance the accuracy and reliability of fraud detection systems. The ultimate goal is to provide financial institutions with a powerful tool to mitigate the risks associated with credit card fraud, ensuring secure and trustworthy transactions for all users.

This report will outline the methodology employed, including data exploration, model selection, and performance metrics, followed by a discussion of the results and potential future enhancements. Through this project, we aim to contribute to the ongoing efforts in combating credit card fraud and safeguarding the financial ecosystem.

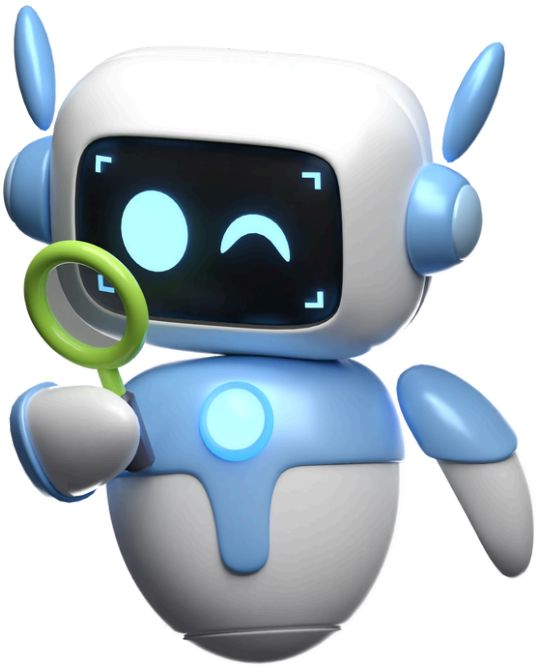
# Data Acquisition

| | |
|---|---|
| 1 | Transaction ID |
| 2 | Date |
| 3 | Day of Week |
| 4 | Time |
| 5 | Type of Card |
| 6 | Entry Mode |
| 7 | Amount |
| 8 | Type of Transaction |
| 9 | Merchant Group |
| 10 | Country of Transaction |
| 11 | Country of Residence |
| 12 | Shipping Address |
| 13 | Gender |
| 14 | Age |
| 15 | Bank |

For this project, we sourced the dataset from Kaggle, a renowned platform for data science and machine learning datasets. The dataset comprises 1 lakh rows and includes a variety of features pertinent to credit card transactions. These features are: Transaction ID, Date, Time, Type of Card (Visa, MasterCard), Entry Mode (CVC, Tap, PIN), Amount, Type of Transaction (Online, POS, ATM), Merchant Group, Transaction Country, Shipping Address, Billing Address, Gender of Cardholder, Age of Cardholder, and Issuing Bank. By leveraging this rich and diverse dataset, we aimed to build a robust model capable of accurately detecting fraudulent transactions.

# DATA PREPROCESSING

Effective data preprocessing is a crucial step in preparing a dataset for machine learning tasks. It involves transforming raw data into a clean and usable format, ensuring that the machine learning models can perform optimally. In this project, we have undertaken several preprocessing steps to enhance the quality of our dataset and facilitate accurate fraud detection.

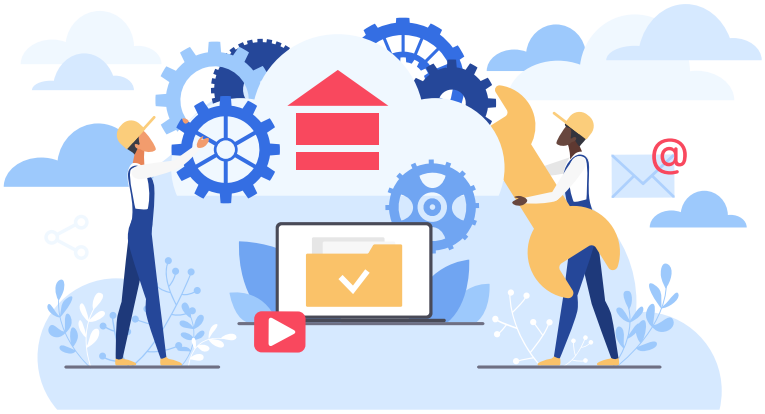


- Removal of Null Value Columns
- One-Hot Encoding and label encoding for Categorical Data
- Removal of Currency Signs and hash
- Conversion of Strings to Integer
- Balancing Class using SMOTE

## Description

To begin with, we addressed missing data by removing columns with null values, as they can introduce biases and inaccuracies in the model. Next, categorical variables were converted into a numerical format using one-hot encoding, allowing the model to interpret and process these features effectively. Additionally, currency signs were stripped from the amount fields, and string data types were converted to integers to standardize the numerical values. Furthermore, we employed the Synthetic Minority Over-sampling Technique (SMOTE) to address class imbalance by generating synthetic samples for the minority class. These preprocessing steps are essential to ensure that the data is in a suitable format for machine learning algorithms, ultimately improving the model's performance

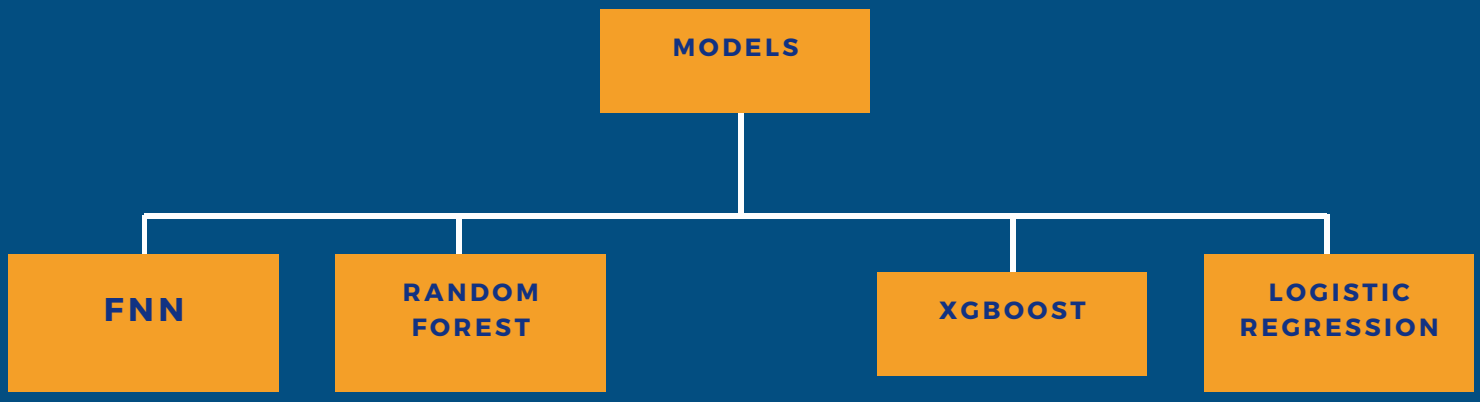
# Data Engineering



## Description

- **Creation of Day, Month, and Year from Date:** The date feature was decomposed into separate day, month, and year components, allowing the model to utilize temporal patterns more effectively.
- **Calculation of Distance from Home:** The distance from the cardholder's home to the transaction location was calculated, providing a critical feature for identifying unusual spending behavior and potential fraud.

# Model developement



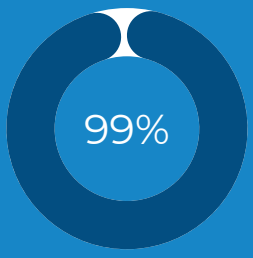

for our project we have evaluated results of FNN, Random Forest, XGBoost and Logistic Regression

# Model Evaluation

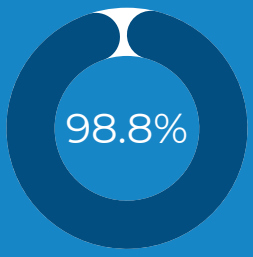
## Description :

Model evaluation is a critical phase in the machine learning pipeline that assesses the performance and reliability of the trained model. This step involves using various metrics and techniques to measure how well the model predicts outcomes on unseen data. We concluded that FNN is performing best amongst all Models

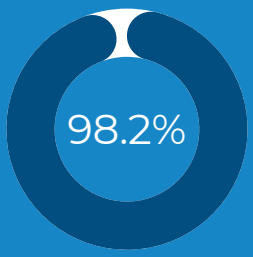
### Models Accuracy



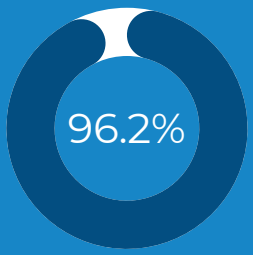

**FNN**
FNN gave us best results among all models yielding 99% Accuracy




**Random Forest**
Random forest gave us accuracy of 98.8%

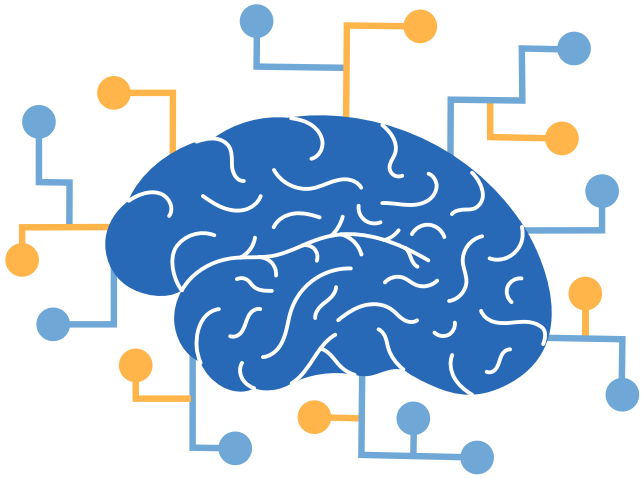



**Logistic Regression**
Logistic Regression gave us accuracy of 98.2%




**XGBoost**
XGBoost model gave us accuracy of 96.2%

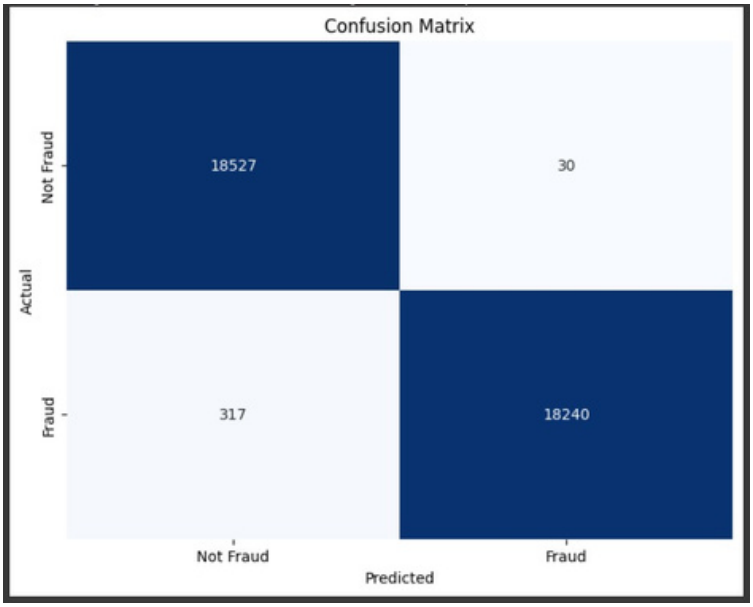
# What Model we used?
# Forward Neural Network

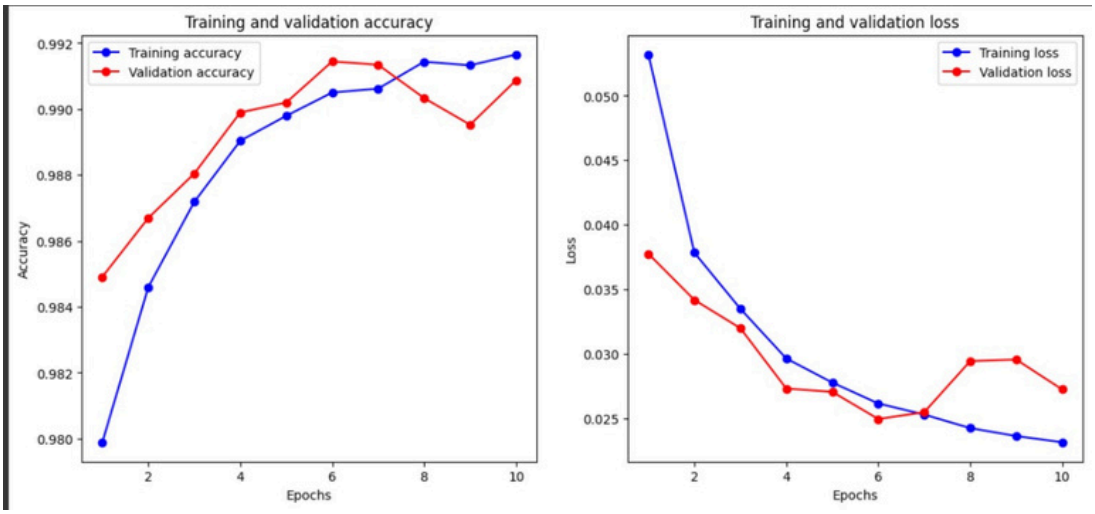


In this project, we used a Feedforward Neural Network (FNN) to classify transactions as fraudulent or non-fraudulent. The FNN processes each transaction independently, utilizing a structured approach. The input layer accepts the features of each transaction, such as amount, type of transaction, and entry mode. These features are then passed through hidden layers composed of neurons that apply weights and biases to the inputs. An activation function introduces non-linearity, enabling the model to learn complex patterns in the data. Finally, the output layer produces the final prediction, indicating whether a transaction is likely to be fraudulent.

## Confusion Matrix

- True Negatives (TN): 18,475
- These are the instances where the model correctly predicted that the transaction is not fraudulent.
- False Positives (FP): 82
- These are the instances where the model incorrectly predicted that a non-fraudulent transaction is fraudulent.
- False Negatives (FN): 232
- These are the instances where the model incorrectly predicted that a fraudulent transaction is not fraudulent.
- True Positives (TP): 18,325
- These are the instances where the model correctly predicted that the transaction is fraudulent.




## Accuracy and loss

On the test dataset, the model achieved an accuracy of 99.15% and a loss of 0.0243, confirming its high accuracy and low error rate in predicting credit card fraud.

# USER-INTERFACE

The system is built using Flask, a lightweight web framework for Python, and includes several HTML pages to provide a comprehensive user interface

**The application consists of the following seven HTML pages**

1. Home Page
2. Login Page
3. About Us Page
4. Contact Us Page
5. Vision Page
6. Fraud DetectionPage
7. Results Page

# FUTURE WORK

1. To implement real-time fraud detection, we plan to integrate Apache Kafka into our deployment pipeline. Kafka's robust and scalable messaging platform will allow us to process and analyze streaming transaction data in real-time, ensuring immediate detection and prevention of fraudulent activities.

2. We will experiment with ensemble learning methods, combining multiple models to improve prediction accuracy and robustness. Techniques such as stacking, boosting, and bagging can be employed to enhance the model's performance.

3. As the volume of transaction data grows, it is crucial to ensure that our system can scale efficiently. We will focus on optimizing the performance of our machine learning models and the deployment infrastructure to handle high throughput and low latency requirements.