CREDIT CARD FRAUD DETECTION

The Complete steps involved in training the model are as follows:

1) **IMPORT NECESSARY LIBRARIES:**

In this step, essential Python libraries such as pandas, numpy, matplotlib, seaborn, and scikit-learn are imported. These libraries enable data manipulation, visualization, and machine learning functionalities, providing the necessary tools for exploring and modelling the credit card fraud detection dataset.

2) **LOAD THE DATASET:**

In this step, the Kaggle Credit Card Fraud Detection dataset is loaded into a pandas data frame using the `read_csv` function. This step prepares the dataset for further exploration and analysis in subsequent steps of the machine learning pipeline.

3) **EXPLORE THE DATA:**

In step 3, the dataset is explored to understand its structure, features, and class distribution. Information such as data types, summary statistics, and basic characteristics of the dataset is examined. This exploration helps in gaining insights into the nature of the data before proceeding with pre-processing and modelling steps.

4) **DATA PRE-PROCESSING:**

In this step, data preprocessing begins by separating the features (X) from the target variable (y), assuming 'Class' is the target representing fraud or non-fraud transactions. The features are stored in variable `X`, and the target variable is stored in `y`. This separation is a fundamental step for supervised machine learning. The subsequent steps will involve further preprocessing, handling missing values, outliers, and scaling/normalizing numerical features to ensure the data is ready for training a machine learning model. The ultimate goal is to have a well-structured dataset that can be used to train and evaluate a model for credit card fraud detection.

5) **HANDLING IMBALANCED DATA:**

In this step, the class imbalance in the dataset is addressed using the Synthetic Minority Over-sampling Technique (SMOTE) from the imbalanced-learn library. SMOTE generates synthetic examples of the minority class (fraudulent transactions) to balance the class distribution. This technique helps prevent the model from being biased towards the majority class and improves its ability to detect fraud. The resampled features (`X_resampled`) and target labels (`y_resampled`) are obtained and will be used for subsequent model training to ensure the model is exposed to a more balanced representation of both classes.

6) SPLIT DATA INTO TRAINING AND TESTING DATA:

In this step, the preprocessed data is split into training and testing sets using the `train_test_split` function from scikit-learn. This separation is crucial for assessing the model's performance on unseen data. The features (`X_resampled`) and target labels (`y_resampled`) are divided into training (`X_train`, `y_train`) and testing (`X_test`, `y_test`) sets with an 80-20 split, respectively. The random_state parameter ensures reproducibility. The training set is utilized for training the machine learning model, while the testing set is kept aside to evaluate its performance in identifying credit card fraud.

7) MODEL TRAINING:

In this step, a Logistic Regression model is chosen for its suitability in binary classification tasks. The `Logistic Regression` class from scikit-learn is instantiated, and the model is trained using the training data (`X_train` and `y_train`) through the `fit` method. Logistic Regression is particularly well-suited for fraud detection due to its simplicity and effectiveness in modelling the probability of an event occurring. The model will learn the relationships between features and the likelihood of a transaction being fraudulent. This step sets the foundation for subsequent evaluation and analysis of the model's performance.

8) MODEL EVALUATION:

In this step , the trained Logistic Regression model is evaluated on the testing set (`X_test` and `y_test`). Predictions are made using the testing features, and performance metrics are calculated to assess the model's effectiveness in credit card fraud detection. The confusion matrix, classification report, and AUC-ROC score provide insights into the model's ability to classify transactions as either fraudulent or non-fraudulent. These metrics, such as **precision, recall, and the AUC-ROC curve**, offer a comprehensive view of the model's strengths and weaknesses, guiding further adjustments or improvements as necessary.

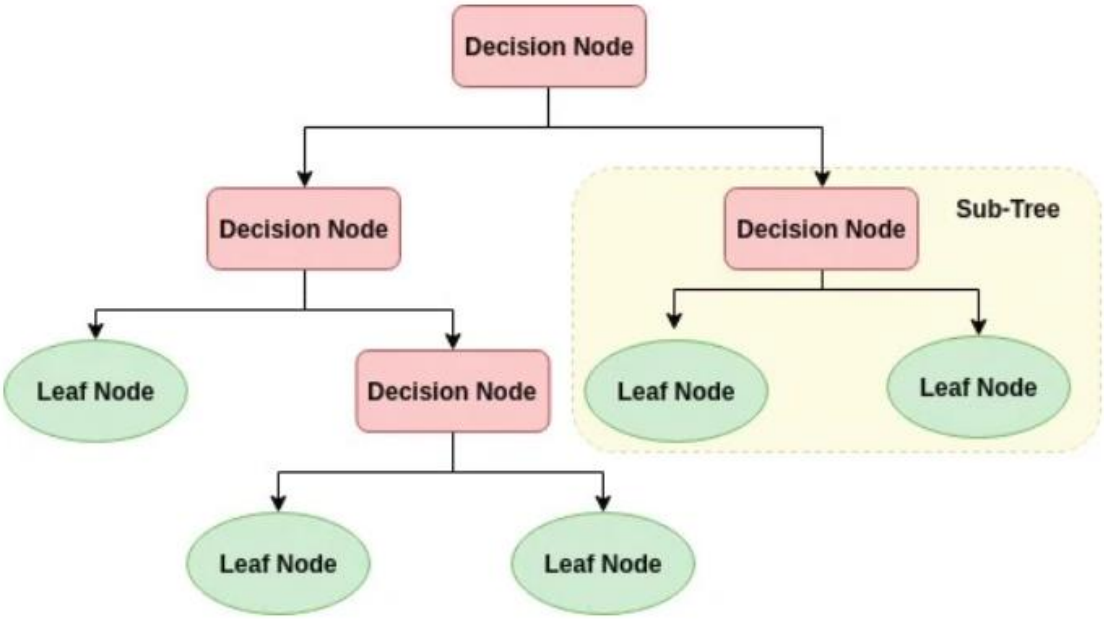
9) ADJUST THRESHOLD:

This step involves an optional adjustment of the decision threshold for the Logistic Regression model. By modifying the threshold for classification probability, you can fine-tune the model's sensitivity and specificity, depending on the desired trade-off between false positives and false negatives.

10) FEATURE IMPORTANCE:

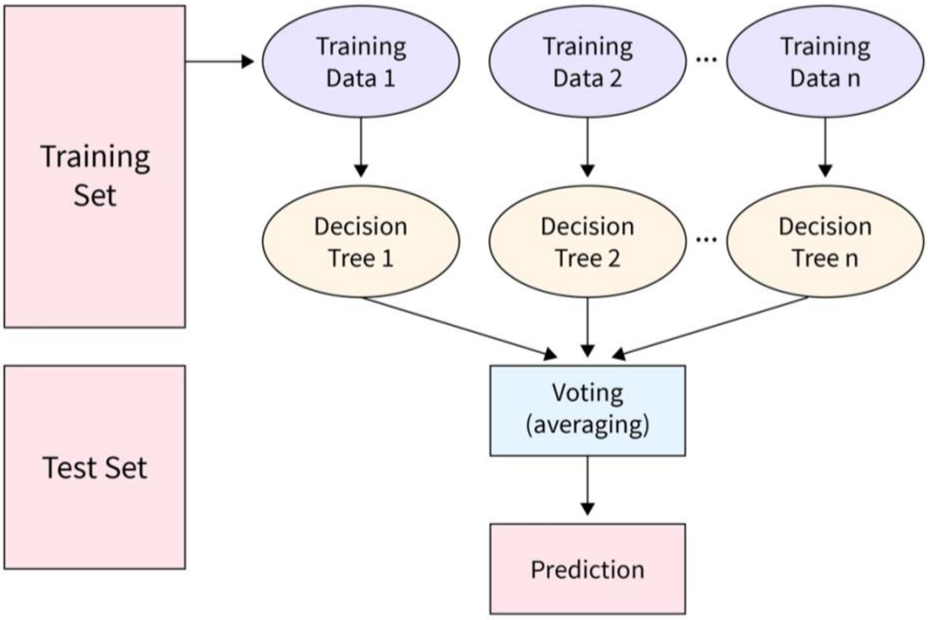
In this step, an optional analysis of feature importance is performed. The coefficients of the Logistic Regression model are examined to understand the impact of each feature on predicting credit card fraud. This insight can aid in identifying crucial factors contributing to fraudulent transactions.

MODELS USED :



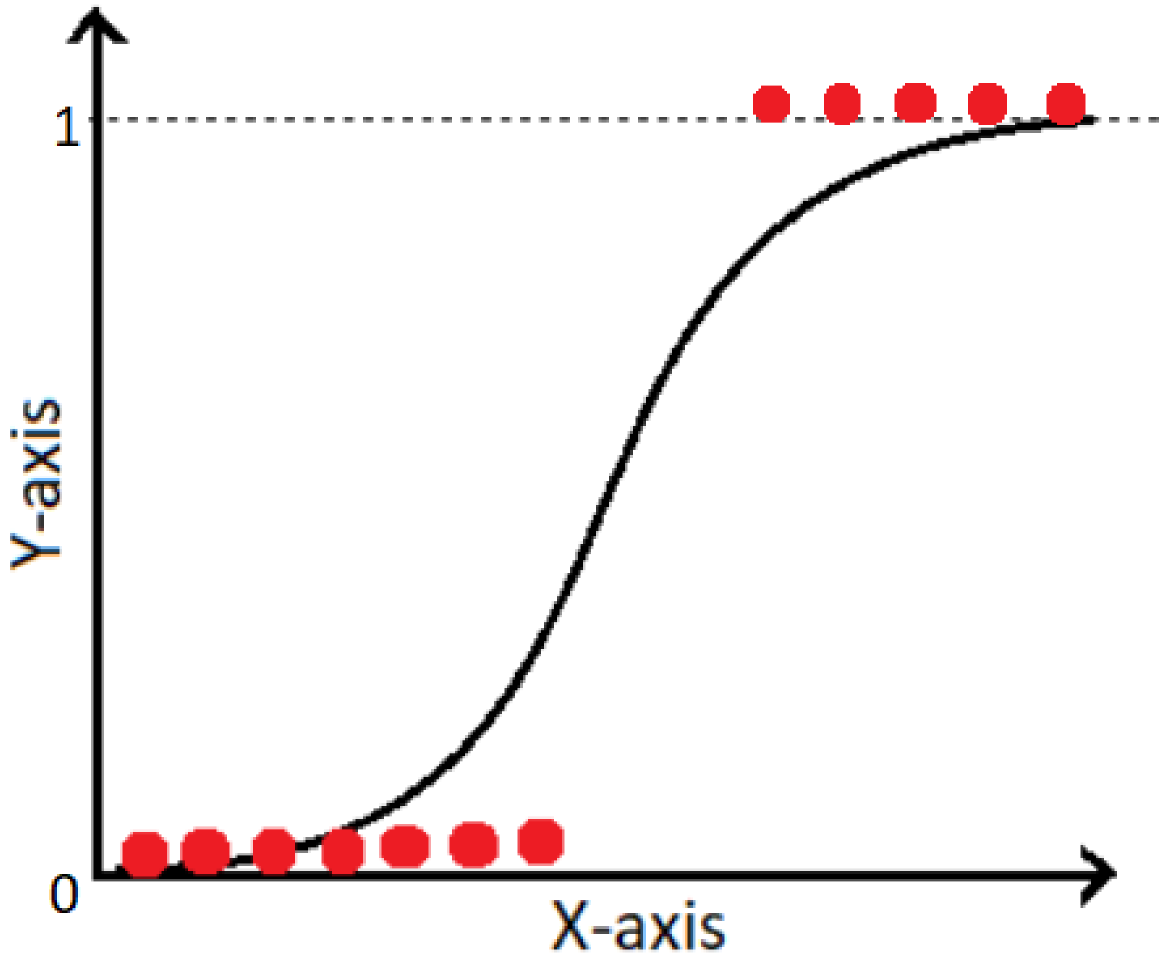
DECISION TREE

A decision tree is a supervised machine learning model that makes decisions or predictions by recursively splitting a dataset into subsets based on features. It starts with a root node representing the entire dataset, selects the best feature to split the data, and continues this process to create branches and leaf nodes. Leaf nodes provide the final predictions. Decision trees are interpretable but can overfit; pruning is used to prevent this. They are commonly used for both classification and regression tasks, and techniques like Random Forest and Gradient Boosting extend their capabilities by combining multiple trees for improved accuracy and robustness.



RANDOM FOREST

Random Forest is an ensemble learning method that combines multiple decision trees for enhanced predictive accuracy and robustness. It constructs individual decision trees by bootstrapping the training data and selecting random subsets of features at each split, ensuring diversity among the trees. During predictions, it combines the results through voting (for classification) or averaging (for regression), reducing overfitting and improving generalization. Random Forest also offers feature importance insights and an estimate of model performance through out-of-bag error. It excels in various tasks, making it a popular choice in machine learning for its versatility and effectiveness.



LOGISTIC REGRESSION

Logistic Regression is a binary classification model that predicts the probability of an instance belonging to one of two classes using a sigmoid function. It calculates a linear combination of input features and learns optimal weights and a bias term during training. With a decision boundary at 0.5 probability, it classifies instances into one of the two classes. It's simple, interpretable, and efficient for small to medium datasets but assumes linear relationships between features and the log-odds of the target variable, making it less suitable for highly nonlinear data. Regularization can be applied to prevent overfitting, and it can be extended to multiclass classification tasks.

ADVANCED TECHNIQUES FOR IMPROVING FRAUD DETECTION

ANOMALY DETECTION:

Anomaly detection in credit card fraud detection involves collecting transaction data, pre-processing, and selecting or engineering relevant features. A chosen algorithm is trained on historical data to learn normal behaviour, enabling it to identify unusual or suspicious transactions in real-time. When deviations from normal patterns are detected, the system generates alerts, triggering responses such as notifying cardholders or further investigation. Regular model updates are performed to adapt to evolving fraud tactics. Striking a balance between minimizing false positives and false negatives is crucial for effective fraud prevention, protecting both cardholders and financial institutions from fraudulent activities.

ISOLATION FOREST:

Isolation Forest is a highly suitable algorithm for credit card fraud detection due to its scalability and ability to efficiently identify anomalies in large datasets, making it ideal for high-dimensional transaction data. Being unsupervised, it doesn't require labelled fraud data for training, making it adaptable to evolving fraud patterns. However, parameter tuning is necessary, and regular model updates are crucial as fraud tactics change over time. Isolation Forest provides interpretable anomaly scores for ranking anomalies, and it can be applied in real-time for timely fraud detection. Combining it with other techniques and addressing data imbalance are common practices. Evaluating its performance with appropriate metrics is essential to ensure it meets fraud detection goals effectively.