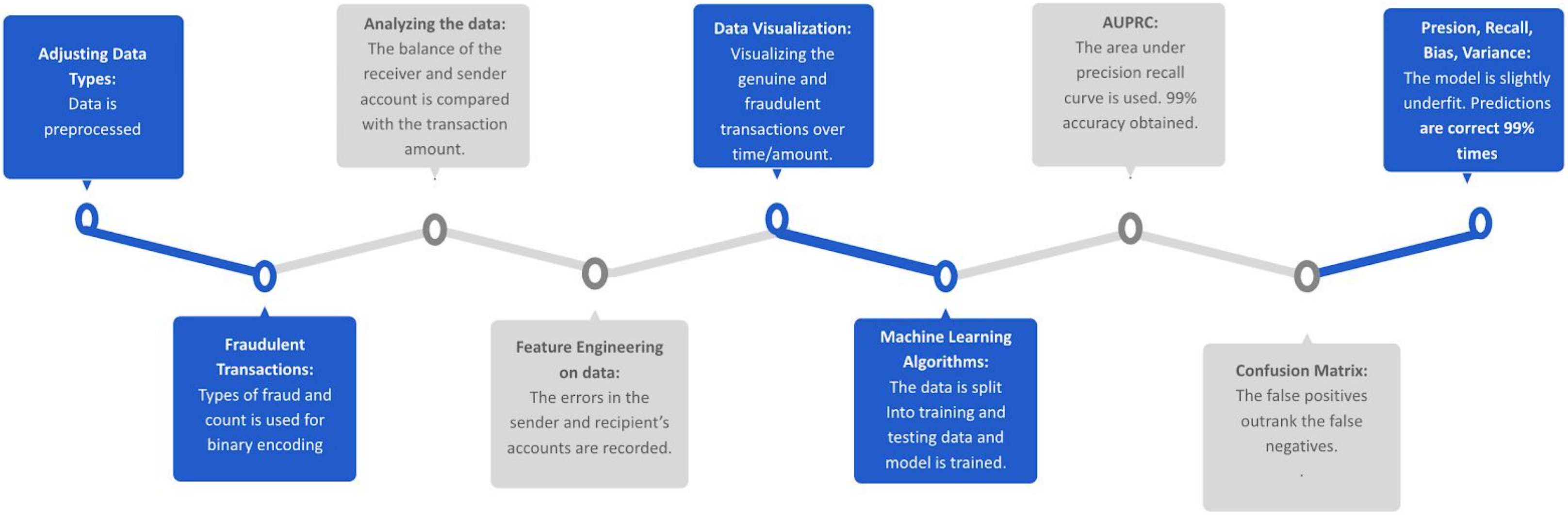
# Detailed Solution Design and Architecture

## ONLINE PAYMENT TRANSACTION FRAUD



**Adjusting Data Types:** Data is preprocessed

**Fraudulent Transactions:** Types of fraud and count is used for binary encoding

**Analyzing the data:** The balance of the receiver and sender account is compared with the transaction amount.

**Feature Engineering on data:** The errors in the sender and recipient's accounts are recorded.

**Data Visualization:** Visualizing the genuine and fraudulent transactions over time/amount.

**Machine Learning Algorithms:** The data is split Into training and testing data and model is trained.

**AUPRC:** The area under precision recall curve is used. 99% accuracy obtained.

**Confusion Matrix:** The false positives outrank the false negatives.

.

**Presion, Recall, Bias, Variance:** The model is slightly underfit. **Predictions are correct 99% times**

Let's Look at Fig 1. In detail.

1. **Adjusting Data Types:** The data is preprocessed according to the algorithm.
2. **Fraudulent transactions:** The different types of frauds and their respective count is found out.These become the explanatory and dependent variables for analysis. Binary encoding is then used by labelling the two cases.
3. **Analysing the Data:**

   - Both the old and new balance in the recipient's account were zero, but transferred amount was not zero
   - Both the old and new balance in the sender's account were zero, but transferred amount was not zero

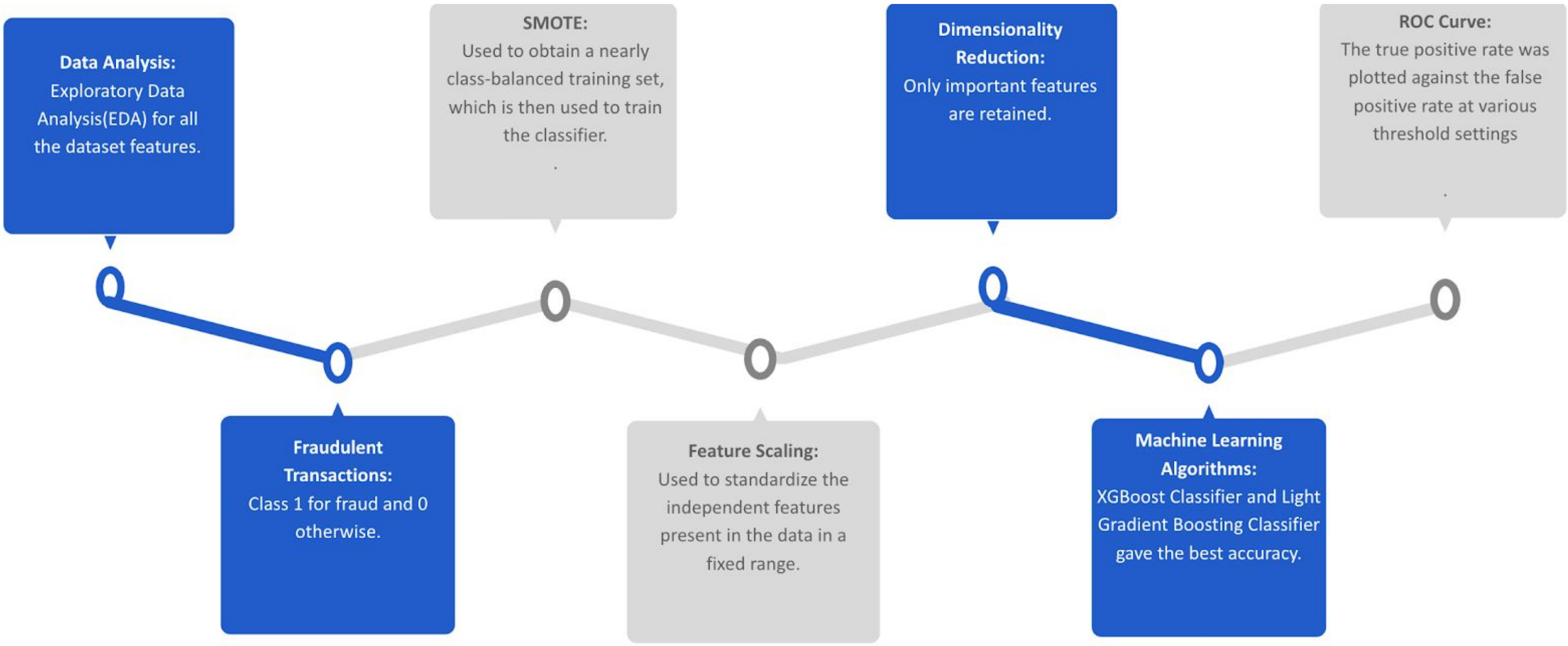
   We observed that zero balances in both sender and recipient's accounts are strong indicators of fraud, when the transaction is non-zero.
4. **Feature Engineering on data:**

   - Two new features (columns) are created to record errors in the senders' and receivers' accounts for each transaction for better analysis.
5. **Data Visualization:**

- Visualizing the fraudulent and genuine transactions over time.
- Visualizing the fraudulent and genuine transactions over amount.
- Dispersion over error in balance in destination accounts
- Separating out genuine transactions from fraudulent transactions

6. **Machine Learning to Detect Fraud in Skewed Data:**
    - The data is split into 80:20 train : test data and the model is trained.
7. **Investigating the AUPRC(Area under Precision-Recall Curve):**
    - Since the data is highly skewed, the area under the precision-recall curve (AUPRC) is used. The model gave a 99% accuracy for predicting fraudulent transactions.
8. **Confusion Matrix:**
    - The number of both False Negatives (21) and False Positives (649) were low. It can be deduced from the confusion matrix that the FPs outrank the FNs. This is beneficial as we rather get more false alarms than letting actual frauds slip through undetected.
9. **Precision and Recall:**
    - The prediction of a fraudulent transaction is correct 99% of the time but it also manages to detect 99.7% of the fraudulent transactions. We also found the harmonic mean of Precision and Recall.
10. **Bias-variance tradeoff:**
    - The model that we trained had a degree of bias and was slightly underfit. The easiest way to improve the performance of the model we found was to increase the max_depth parameter of the XGBClassifier at the expense of the longer time spent learning the model.

# CREDIT CARD TRANSACTION FRAUD



**Data Analysis:** Exploratory Data Analysis(EDA) for all the dataset features.

**Fraudulent Transactions:** Class 1 for fraud and 0 otherwise.

**SMOTE:** Used to obtain a nearly class-balanced training set, which is then used to train the classifier.

**Feature Scaling:** Used to standardize the independent features present in the data in a fixed range.

**Dimensionality Reduction:** Only important features are retained.

**Machine Learning Algorithms:** XGBoost Classifier and Light Gradient Boosting Classifier gave the best accuracy.

**ROC Curve:** The true positive rate was plotted against the false positive rate at various threshold settings

Let's Look at Fig 2. In detail.

1. **Data Analysis:**
   - **Downloading and combining the data**

     The data consists of over 284,807 rows in csv. Next, data is imported and concatenated into one dataframe.

   - **Exploratory Data Analysis(EDA) for all the dataset features**

     A distance plot for every column of the dataset is plotted.

2. **Fraudulent transactions:**
   - Feature 'Class' is the target variable and it takes value 1 in case of fraud and 0 otherwise. So pointing out such fraudulent and genuine transactions over Time plotting the Amount of money in such transactions.
   - We are creating a dataframe of the shuffled class 0 and 1 transactions and are then putting them all together into a new csv file. Now using these new dataframe we are able to separate out the Class 0 and 1 transactions widely and visualize them.

3. **SMOTE**

SMOTE, an oversampling technique is used here to generate synthetic samples. It is used to obtain a synthetically class-balanced or nearly class-balanced training set, which is then used to train the classifier.

4. **Feature Scaling**

   This technique is used here to standardize the independent features present in the data in a fixed range.

5. **Dimensionality Reduction**

   We are only limiting the data to the most important features only and adding the relevant features as the data is highly unbalanced.

   Thus the new dataframe now contains only the relevant 4000 rows and 8 columns. The new data ready for training the model is now placed in a different csv file to be worked on for further analysis.

6. **Machine Learning Algorithms**
   - **Logistic Regression**

     . A statistical model that uses a logistic function to model a binary dependent variable Class here in the data.For this regression model, we are getting a predictive accuracy of 97.33%.

   - **Hyperparameter tuning**

     Parameters which define the model architecture are referred to as hyperparameters. Here we are now going to use different classification algorithms and build models, thus there is a need of a process of searching for the ideal model architecture which can be done with the help of hyperparameter tuning.

   - **Support Vector Classifier**

     We are using Support Vector classifier to fit the model to the data, returning a "best fit" hyperplane that divides, or categorizes the data. Here it is observed that SVC gives an accuracy of around 96.5% for classification.

   - **Decision Tree Classifier**

     This algorithm of classification is used to create the classification model by building a decision tree. Each node in the tree specifies a test on the attributes, each branch descending from that node corresponds to one of the possible values for that attribute. This is found to be giving an accuracy of 96.41%.

   - **Random Forest Classifier**

Multiple decision trees are used. It aggregates the votes from different decision trees to decide the final type of the transaction as fraudulent or genuine of the "Class" test object from the data. As expected, Random Forest appears to give an improved accuracy of 97.41% than the Decision Tree classifier.

- **K Nearest Neighbor Classifier**

  An object is classified by a plurality vote of its neighbors in KNN. Here we have tried giving two different values for k ie. 5 and 2. When k=5, we are getting an accuracy score of 96.75% whereas for k=2 which is a much smaller value for k, we are getting slightly less accurate predictions as the accuracy comes out to be 96.58%.

- **XGBoost Classifier**

  XGBClassifer is an implementation of gradient boosted decision trees designed for speed and performance that is dominative competitive machine learning. We need our model to be the best fit for detection and prediction of frauds. The accuracy for prediction using XGBClassifier is 97.33% almost similar to the random forest approach.

- **LGB Classifier**

  LGB Classifier is a gradient boosting framework that uses tree based learning algorithm as like decision trees, Random forest and XGBClassifer. Light GBM grows tree vertically while other algorithm grows trees horizontally meaning that Light GBM grows tree leaf-wise while other algorithm grows level-wise. When growing the same leaf, Leaf-wise algorithm like LGB can reduce more loss than a level-wise algorithm and hence we have created a model using this algorithm. It is giving us a prediction accuracy of 97.33%.

### 7. ROC Curve

A ROC (receiver operating characteristic) curve is a graphical plot that illustrates the diagnostic ability of a binary classifier system as its discrimination threshold is varied for various algorithms. The ROC curve is created by plotting the true positive rate against the false positive rate at various threshold settings for every classifier we have used here to build the prediction models for detection of Frauds in Credit card payments transactions.