**Project Report**

**Find Default (Prediction of Credit Card fraud)**

**A fishing hook on a keyboard

Description automatically generated**

**Overview**

In response to the increasing number of credit card frauds, our project focuses on developing an effective solution for real-world fraud detection scenarios. With criminals employing fake identities and advanced technologies to exploit users, it has become imperative to formulate efficient strategies. Our proposed model aims to tackle this challenge by leveraging machine learning and other relevant IT fields.

To address the evolving nature of fraudulent activities, we employ a logistic regression algorithm, decision tree algorithm and XGBoost algorithm to classify credit card transactions as either fraud or genuine. By analysing extensive datasets and including real-time user data, our model enhances accuracy in detecting fraud transactions. Furthermore, we utilize data visualization techniques to process key attributes, enabling the identification of fraudulent patterns.

Evaluation of our techniques is based on metrics such as accuracy, precision, ROC curves and recall. Our results demonstrate the superior performance of the XGBoost algorithm, achieving significant accuracy levels.

* **INTRODUCTION: -**

Detecting credit card fraud is developing challenges in the financial sector, with billions lost annually due to fraudulent activities. Traditional methods involve analyzing spending patterns and implementing government and bank safeguards, yet fraudsters continually adapt to evade detection.

Although incidences of credit card fraud are limited to about 0.1% of all card transactions, they have resulted in huge financial losses as the fraudulent transactions have been large value transactions. It is important that credit card companies can recognize fraudulent credit card transactions so that customers are not charged for items that they did not purchase. What we need is an algorithm, which could classify a transaction as fraudulent or non-fraudulent. Doing so will benefit both the credit card companies and the customers who must go through the ordeal.

To address this, our project focuses on predictive analytics, employing three classification algorithms: Logistic Regression, Decision Trees, and XGBoost. By analyzing a credit card dataset, we aim to identify the most effective model for distinguishing between genuine and fraudulent transactions.

* **PROBLEM STATEMENT: -**

A credit card is one of the most used financial products to make online purchases and payments. Though the Credit cards can be a convenient way to manage your finances, they can also be risky. Credit card fraud is the unauthorized use of someone else's credit card or credit card information to make purchases or withdraw cash. It is important that credit card companies can recognize fraudulent credit card transactions so that customers are not charged for items that they did not purchase. The dataset contains transactions made by credit cards in September 2013 by European cardholders. This dataset presents transactions that occurred in two days, where we have 492 frauds out of 284,807 transactions. The dataset is highly unbalanced, the positive class (frauds) account for 0.172% of all transactions.

We have to build a classification model to predict whether a transaction is fraudulent or not.

* **Focus on this project: -**
* **Exploratory Data Analysis:** Analyze and understand the data to identify patterns, relationships, and trends in the data by using Descriptive Statistics and Visualizations.
* **Data Cleaning:** This might include standardization, handling the missing values and outliers in the data.
* **Dealing with Imbalanced data:** This data set is highly imbalanced. The data should be balanced using the appropriate methods before moving onto model building.
* **Feature Engineering:** Create new features or transform the existing features for better performance of the ML Models.
* **Model Selection:** Choose the most appropriate model that can be used for this project.
* **Model Training:** Split the data into train & test sets and use the train set to estimate the best model parameters.
* **Model Validation:** Evaluate the performance of the model on data that was not used during the training process. The goal is to estimate the model's ability to generalize to new, unseen data and to identify any issues with the model, such as overfitting.
* **DATA: -**

The dataset used in this project contains transactions made by credit cards in September 2013 by European cardholders. It includes a total of 284,807 transactions, out of which 492 are labelled as fraudulent. The dataset is highly imbalanced, with fraudulent transactions accounting for only 0.172% of all transactions.

* **DATA CLEANING: -**
* **Missing Value: -** There are no missing values in this dataset.
* **Feature Scaling: -** Through the above dataset analysis, it is seen that all the columns are scaled except the Amount & Time features.

These are the only features which have not been transformed. Feature **Time** contains the seconds elapsed between each transaction and the first transaction in the dataset. The feature **Amount** is the transaction Amount. We have scaled these features using the **Standardization** method.

* **Encoding: -** Not performing any encoding for this dataset. Because all the columns are in numerical format.
* **Imbalance data: -** As there are fewer number of transactions for a particular class, the data can be said to be unbalanced. The unbalanced class distribution can be visualized in a diagram given below.

**A close-up of a graph

Description automatically generated**

**Bar distribution of classes: -**

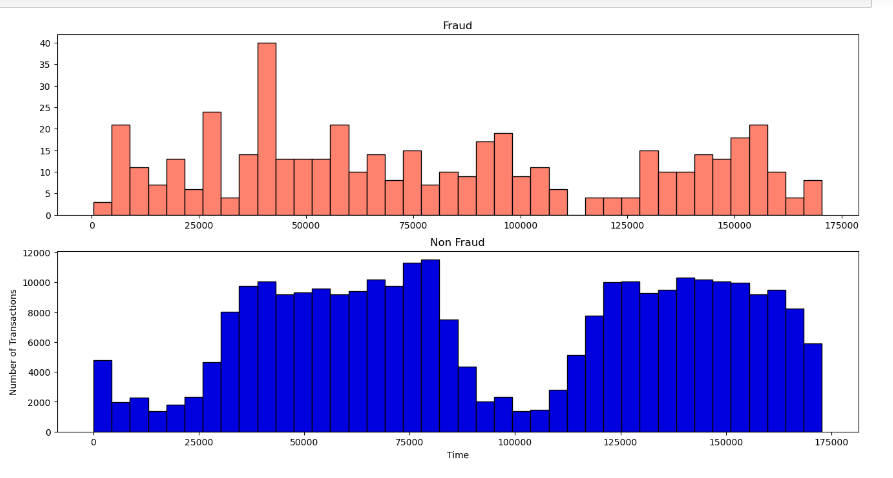
**A graph with a green rectangular bar

Description automatically generated with medium confidence**

* Here, 283253 transactions are genuine (class 0), and 473 transactions are fraud (class 1) which clearly depicts the high imbalance in our dataset.
* Handling imbalance data: - To make the data balancing Oversampling technique is used. We will use SMOTE (Synthetic Minority Over-sampling Technique). SMOTE is a technique used to generate synthetic samples for the minority class to balance the class distribution in the dataset. By creating synthetic samples, SMOTE helps mitigate the impact of class imbalance and improves the performance of machine learning models in predicting the minority class.

Using Oversampling (SMOTE) instead of undersampling because the difference between the two classes (0 and 1) is huge, so if undersampling technique is used, it will lead to losing of most of the sensitive information from the data. Therefore, to overcome that oversampling is used.

* Exploratory Data Analysis: - We will use Subplots, Bar plot, Box plots, Histogram, Heatmap for basic Exploratory Data Analysis.



A green and blue rectangular shapes

Description automatically generated **A diagram of a graph

Description automatically generated with medium confidence**

A graph of a blue line

Description automatically generated with medium confidence

* **Model Selection, Building and Evaluation: -**

We build the model with train-test split in the ratio 80-20% (80% training data and 20% test data). Then we have found which Machine Learning model works well with the balanced dataset.

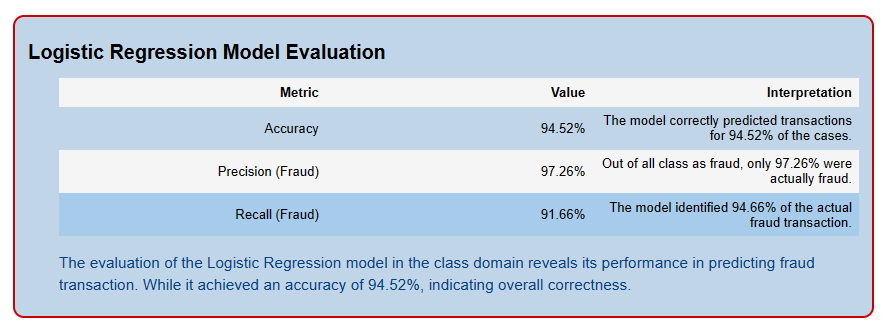
The algorithms used are: -

1. Logistic Regression
2. Decision Tree
3. XGBoost

* **Logistic regression: -**

Logistic regression serves as a crucial tool for classification tasks, primarily applied when the target variable is categorical. It operates by utilizing a logistic function to model a binary response variable. While there exist various types of logistic regression, for the current project, binary logistic regression suffices, accommodating two distinct classes (0 and 1). Within this framework, predictions manifest as the likelihood of attaining outcomes within each class.

Model Evaluation of logistic regression shown below: -



* **Decision Tree: -**

Decision Trees (DTs) are a non-parametric supervised learning method used for classification and regression. The goal is to create a model that predicts the value of a target variable by learning simple decision rules inferred from the data features. A tree can be seen as a piecewise constant approximation.

Model Evaluation of decision tree shown below: -

A screenshot of a computer

Description automatically generated

* **XGBoost: -**

XGBoosting is an extended version of gradient boosting, with additional features like parallel tree learning algorithm and regularization for finding the best split. XGBoost follows the principle of gradient boosting. There are, however, differences in modeling details. Specifically, xgboost used a more regularized model formalization to control over-fitting, which gives it better performance.

The name xgboost, though, refers to the engineering goal to push the limit of computations resources for boosted tree algorithms. Which is the reason why many people use xgboost. For model, it might be more suitable to be called as regularized gradient boosting.

Model Evaluation of XGBoost shown below: -

A screenshot of a computer

Description automatically generated

* **Hyperparameter tuning: -**

Hyperparameter tuning is the process of selecting the optimal values for a [machine learning](https://www.geeksforgeeks.org/machine-learning/) model’s hyperparameters. Hyperparameters are settings that control the learning process of the model, such as the learning rate, the number of neurons in a neural network, or the kernel size in a support vector machine. The goal of hyperparameter tuning is to find the values that lead to the best performance on a given task.

Using the XGBoost Classifier with optimal hyperparameters model evaluation after tuning is shown below:

A screenshot of a computer

Description automatically generated

* **Conclusion: -**

Project aimed to develop algorithms to detect fraudulent credit card transactions. Various models were tested after balancing the data to find the best performer across balanced class distributions. After analyzing the outcomes, it's evident that leveraging XGBoost with oversampled data and fine-tuned hyperparameters yields superior results in identifying fraudulent transactions. This method consistently demonstrates high precision and recall rates across various folds, underscoring its resilience and accuracy in managing imbalanced datasets and delivering precise predictions.