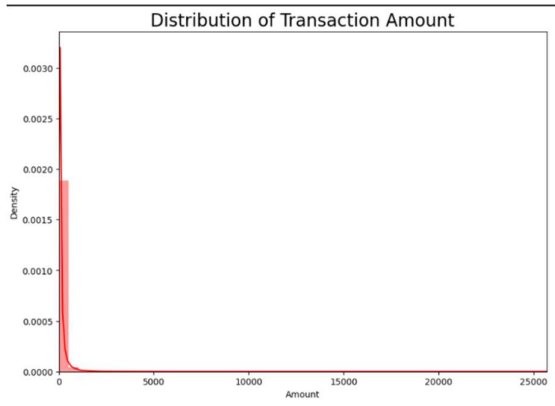
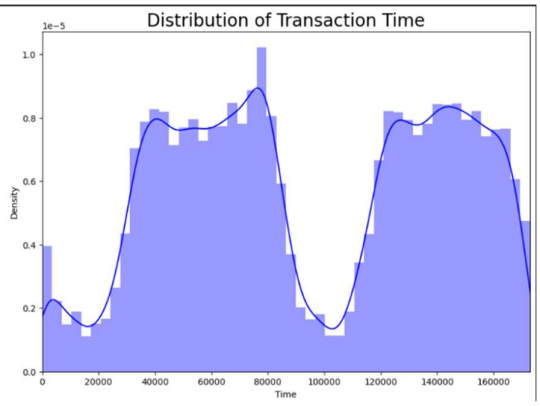
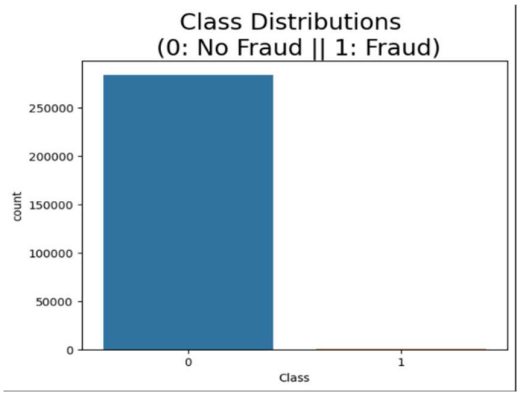
# Report on Peer-graded Assignment:
# Course Final Project

## About the dataset:

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.It contains only numerical input variables which are the result of a PCA transformation. Unfortunately, due to confidentiality issues, we cannot provide the original features and more background information about the data. Features V1, V2, ... V28 are the principal components obtained with PCA, the only features which have not been transformed with PCA are 'Time' and 'Amount'. Feature 'Time' contains the seconds elapsed between each transaction and the first transaction in the dataset. The feature 'Amount' is the transaction Amount, this feature can be used for example-dependant cost-sensitive learning. Feature 'Class' is the response variable and it takes value 1 in case of fraud and 0 otherwise.

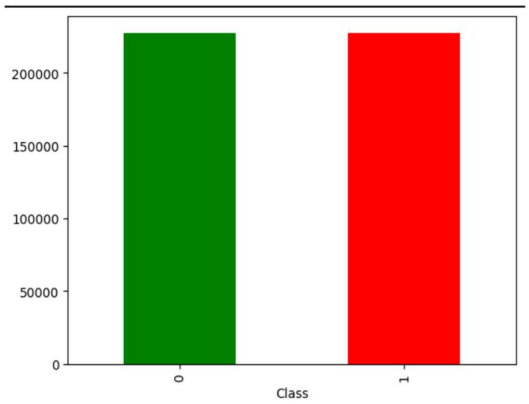



**( SMOTE)**

**---------->**




## Data Preprocessing :

The given dataset has no null values so no need of imputers , etc . All the columns(features) other than Amount and Time are scaled after PCA . (Amount and Time distribution is showed the picture above) Hence , we fit and transform using the Standard Scaler . The dataset is an example of an unbalanced class . So as to solve this issue , we use the ,method of SMOTE(Synthetic Minority Over-sampling Technique) . We then proceed to split the data into Train and Test splits using StratifiedKFold . We also check the proportion of the classes before and after splitting

```
Class
0    0.998273
1    0.001727
Name: proportion, dtype: float64
Class
0    0.998271
1    0.001729
Name: proportion, dtype: float64
Class
0    0.99828
1    0.00172
Name: proportion, dtype: float64
```

## Objective of the Analysis:

It is important that credit card companies are able to recognize fraudulent credit card transactions so that customers are not charged for items that they did not purchase. The main objective of the analysis is to classify the heavily unbalanced dataset using many different classification methods. Through the analysis , we understand which classification method is the best for the current dataset and can be used for future classification on test-data . Since , we have a heavily unbalanced data , we choose the metric for scoring as the "AUC" .

## About the Classification Models:

1. Logistic Regression
2. Logistic Regression (where weights are assigned to classes)
3. K- Nearest Neighbours
4. Decision Tree Classifier ( with GridSearchCV)
5. Bagging Classifier ( with base_estimator as DecisionTreeClassifier )
6. Adaboost Classifier ( with GridSearchCV)
7. Stacking Classifier ( with GradientBoostingClassifier & LogisticRegressor as estimators and XGB as the final estimator)

   **Best AUC Score : Logistic Regressor**

## Key Findings :

| | Model | Accuracy | Precision | Recall | Fscore | AUC |
|---|---|---|---|---|---|---|
| 0 | [LogisticRegression(BeforeClassWeight)] | [0.9885535717420691] | [0.11740331491712708] | [0.8673469387755102] | [0.6962822936357907] | [0.9280547014718871] |
| 1 | [LogisticRegression] | [0.9992450975228665] | [0.8089887640449438] | [0.7346938775510204] | [0.7372981488775109] | [0.8671974566869817] |
| 2 | [KNNeighbours] | [0.9994031003669177] | [0.9102564102564102] | [0.7244897959183674] | [0.7302215189873419] | [0.8621833465109661] |
| 3 | [DecisionTreeClassifier] | [0.9993328768806727] | [0.9166666666666666] | [0.673469387755102] | [0.6804123711340206] | [0.8366819354933646] |
| 4 | [BaggingClassifier] | [0.9993328768806727] | [0.8846153846153846] | [0.7040816326530612] | [0.7096518987341772] | [0.8519616787502509] |
| 5 | [AdaBoostClassifier] | [0.9987359772475904] | [0.9642857142857143] | [0.2755102040816326] | [0.2832929782082324] | [0.6377463089767853] |
| 6 | [StackingClassifier] | [0.9984550833026106] | [0.6923076923076923] | [0.1836734693877551] | [0.1890145395799677] | [0.5917663901816288] |

1. For an unbalanced dataset , a very high accuracy is very easily achieved but that should be a scoring metric due to possibility of false positives .
2. In such cases , Precision-Recall must be optimized .
3. For the dataset of my choice, after oversampling , we have around 2.84 hundred thousands data rows so GridSearchCV is extremely time consuming .
4. Logistic Regressor with and without assigning weights to the classes shows a very big difference in the results .
5. Logistic Regressor is very effective as it is not only time saving but also gives good result in terms of AUC score and Precision-Recall optimization .

## Further Suggestions:

The notebook I have used is attached below . It can be implemented on various similar datasets with same feature engineering on the data .

1. ipynb Notebook :
   https://github.com/darKKnight14110/Credit-Card-Fraud-Detection/blob/main/credit-card-fraud-detection%20(2).ipynb
2. Dataset :
   https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud/data