

# CREDIT CARD FRAUD DETECTION

MINOR PROJECT

CHAITRAVI KANE (B21EE091) & PRIYAM GAURAV(B21EE092)

## PROBLEM:

Credit risk is associated with the possibility of a client failing to meet contractual obligations, such as mortgages, credit card debts, and other types of loans. 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. These datasets are hard to handle. You have to predict whether, given the details about the credit card, it is real or fake.

## SOLUTION:

It was an imbalanced dataset. There are few methods to handle an imbalanced dataset. We used two such techniques

### 1. Ensemble Technique:

- For using this technique, I first loaded the dataset. For preprocessing, firstly, I tried to find the missing values using 'isnull().sum()', but I found that there were no missing values. Then I displayed the number of frauds and non frauds. I then separated the features and target.
- Then I splitted my dataset into training and testing dataset. Then I chose 10 classifiers to train my model which included random forest classifier, decision tree classifier, SVC, KNN, Logistic regression, GaussianNB, Adaboost, Gradient Boosting Classifier, CatBoost Classifier, XGB Classifier. I recorded their accuracies, F1 scores, Kappa scores in a dictionary. After comparing all the scores I found that XgBoost classifier works as the best classifier in this case.
- I trained my model with the XgBoost classifier and recorded its accuracy. I also displayed its confusion matrix.
- To get the best model, I need to find optimal parameters, so I performed hyperparameter tuning using gridsearch.
- For performing gridsearch, I trained a GridSearchCV model over a range of parameters. Then I found the best estimator and the best score. Then I tuned and trained my model with these new

found optimal parameters. Then I fitted my model and found predicted target values and found its accuracy score. I also displayed its confusion matrix. We observe after hyperparameter tuning the accuracy has increased slightly.

- The best algorithm applied to this dataset was XgBoost (in ensemble technique)

## 2. Resampling Technique:

- In Resampling Technique, we have performed certain tasks on three different datasets. First on the raw dataset that we loaded, second dataset oversampled using SMOTE and third dataset Undersampled and Oversampled.
- Before applying the Resampling algorithm, we will preprocess the dataset. Our dataset does not contain any Null values, neither does it have any object so it does not require any Encoding. All the columns except Time and Amount have been Normalized.
- So, our first task was to Normalize the Time and Amount Column using the Robust Scalar. As Robust Scalar is less prone to Outliers.
- After that we apply the same sets of tasks to the 3 different datasets that we created.
- We first use the count function to count the number of datapoints in each class, to know how many Fraud and Non Fraud Cases are present in the dataset.
- In Case of the raw dataset, we observed that the data is highly imbalanced, as we only had 492 cases of Fraud which is very less compared to the Non Fraud cases.
- To counter this, we apply the Resampling algorithm.
- In case of SMOTE, it basically does is oversamples the minority class, by taking a random sample of the positive class and then KNN's are obtained for that instance. Using the distance metric, we multiply the it to random number between (0,1) and added the previous feature vector. Using this we create more no. of data points and stop when there is Binary Class Distribution 1:1.
- We observe the highly imbalanced data using the Count plot.
- Further after Normalizing the 'Time' and 'Amount' columns, we distplot them to visualize them. We can see that they lie b/w a single range of values , which makes it easier for us to work with.
- After that we Scatterplot all the 3 datasets, to check how the data points are distributed, how many outliers are present, the spread of the data and so on.
- After that we split the dataset using the train test split.
- Now for each of the 3 dataset, we will apply the dimensionality Reduction algorithm to check which algorithm is best suited for the dataset.
- We have applied PCA, LDA and ICA dimensionality reduction techniques. PCA is a dimensionality reduction algorithm used mostly for Unsupervised learning, while LDA and ICA are mostly for supervised learning.
- As our dataset is supervised, we still use PCA to see how much difference in the dataset we will observe. Will we get some random results or will the algorithm work better than the other two,
- After applying the Dimension Reduction, we plotted the scatter plot for all the datasets in each two of the Resampling Algorithms. So we got 9 scatter plots in which 3 are from the original dataset. We compare the other two resampled datasets, to see which one is working better in visualizing the datasets.
- In our code, we implemented PCA and LDA from scratch and used it for the dimensional reduction,
- After that to fully compare the workings of the algorithm, we each trained 10 classifiers on the 3 datasets. To find the F1 scores and Kappa Scores.
- In this case we are not relying on accuracy or precision or Recall as our dataset is highly imbalanced it will not provide accurate results.

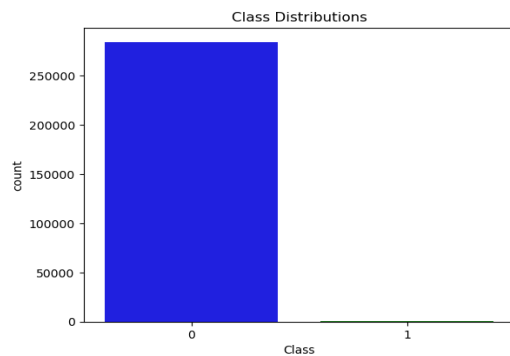
- Now we will get 10 scores, 4 from raw dataset and 3 from each resampled dataset. 1 is applied on the raw dataset and other 3 or 2 from the reduced dimension datasets.
- Our goal is to find the best algorithm for Resampling and also the best Dimensionality Reduction algorithm applied best to the dataset.
- We also used the dataset from the Non reduced dimensionality as a parameter to check whether dimension reduced was required in this case or not.

## Observation:

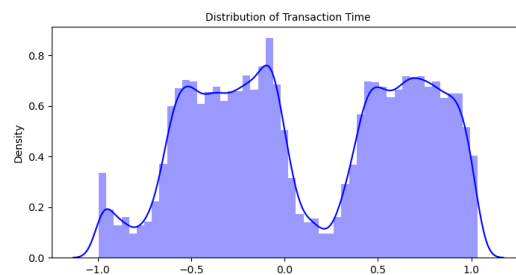
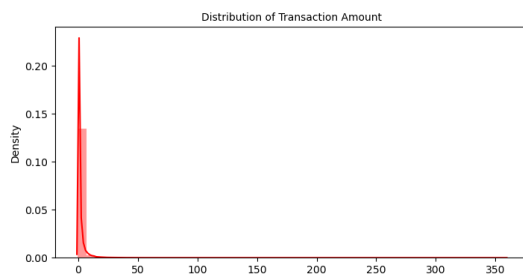
- The best algorithm applied to this dataset was XgBoost (in ensemble technique).

Graphs and Plots are attached below:

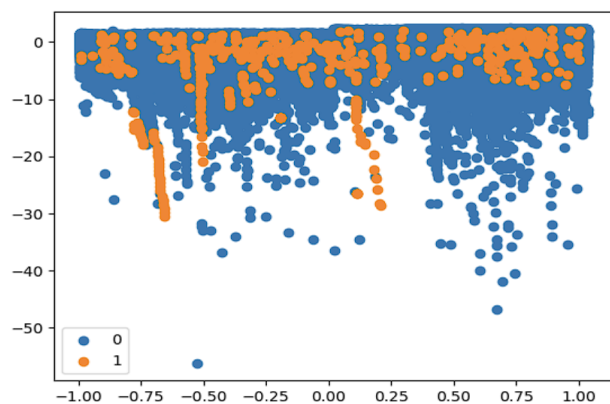
### Count Plot for Class:



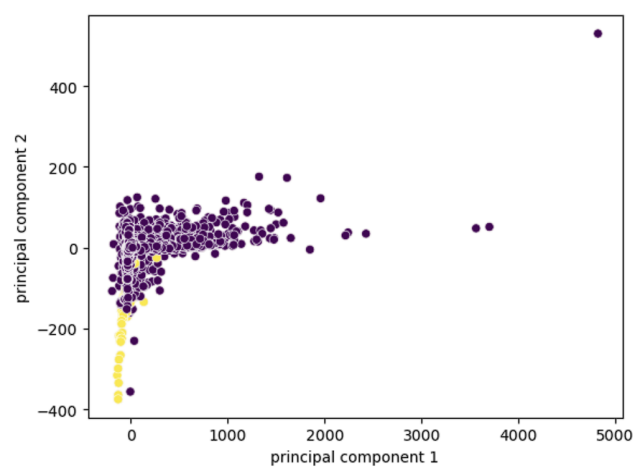
### Distplot For Time and Amount:



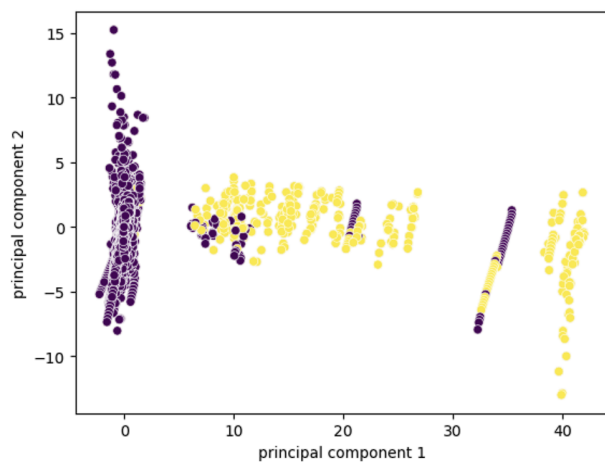
### Scatter Plot for Raw Data:



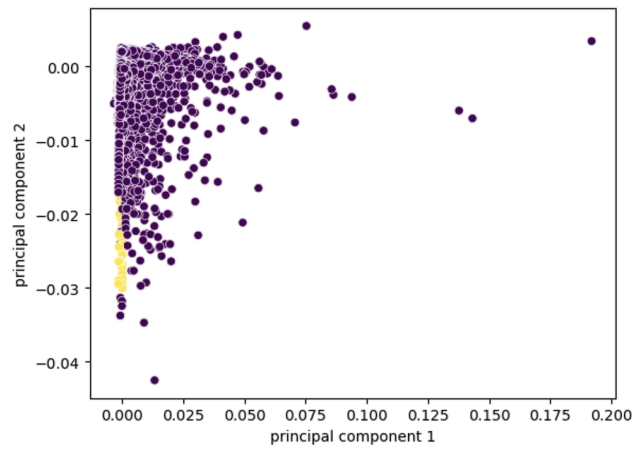
**Scatter Plot For PCA on Raw Data:**



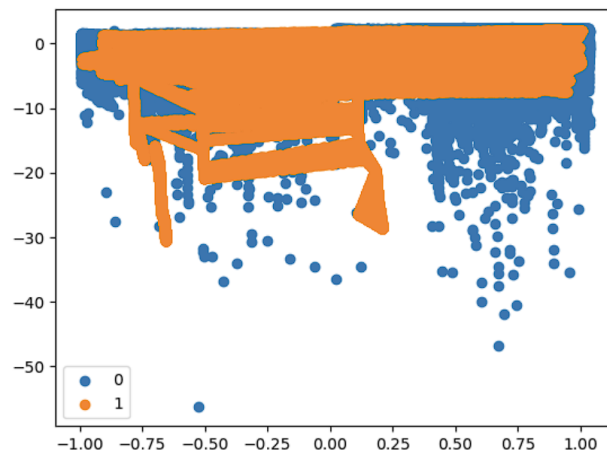
**Scatter Plot for LDA on Raw Data:**



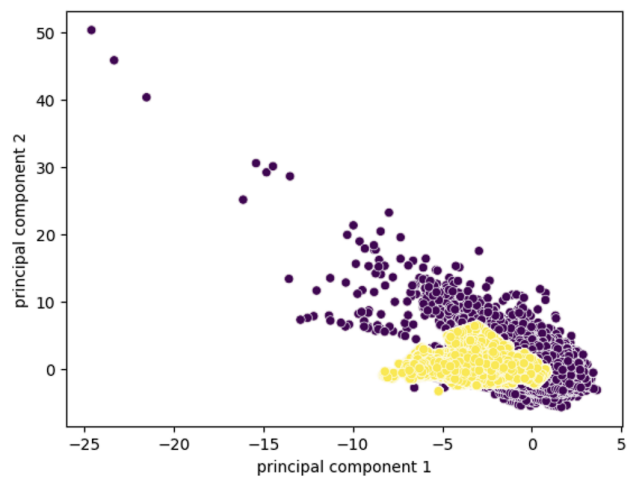
**Scatter Plot for ICA on Raw Data:**



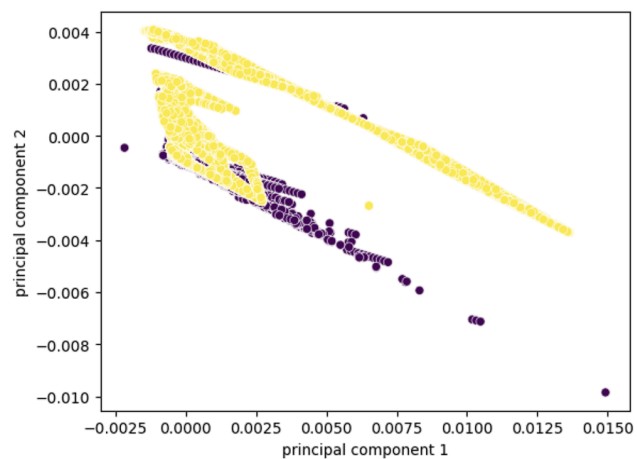
**Scatter Plot for Data Using SMOTE:**



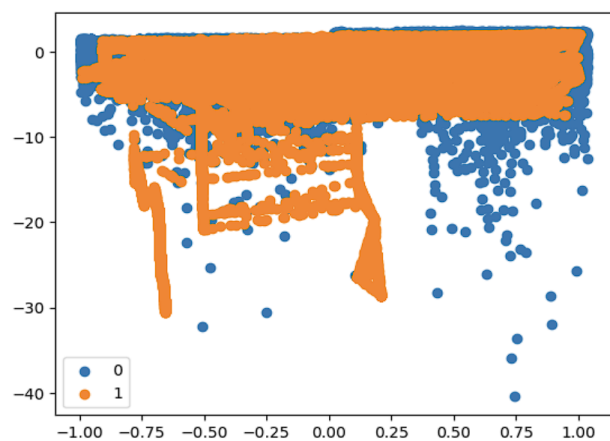
**Scatter Plot for LDA Data resampled Using SMOTE:**



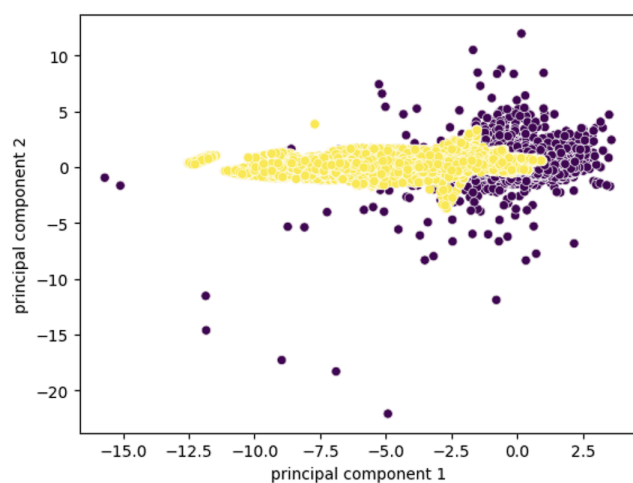
**Scatter Plot for ICA Data resampled Using SMOTE:**



**Scatter Plot for Data Using SMOTE and RandomUnderSampler combined:**



**Scatter Plot for LDA Data resampled Using SMOTE and RandomUnderSampler:**



**Scatter Plot for ICA Data resampled Using SMOTE and RandomUnderSampler:**

