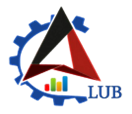
 **WiDS ‘22 - ‘23 Final Documentation** 

54-Fraud Detection Algorithms in Credit Card Transaction

Mentor- Ankit Yadav

|  |  |  |
| --- | --- | --- |
| Team Member Name | Roll Number | Email-Id |
| Sudipta Das | 22N0075 | 22n0075@iib.ac.in |

1. **Introduction**

The term algorithm derives from the name of Muhammad ibn Mūsā al’Khwārizmī, a ninth-century Persian mathematician. His latinized name, Algoritmi, meant “the decimal number system” and was used in this meaning for centuries.

According to historical documents and archaeological evidence, the Babylonians may have constructed the first identifiable algorithm about 1600 BC. In approximately 300 BC, the great Euclid created his renowned “Euclidean algorithm,” which was followed by Eratosthenes in 200 BC with his so-called “Sieve of Eratosthenes.” Lui Hui described Gaussian elimination in 263 AD, while Brahmagupta created Chakravala in 628 AD.

Donald Ervin Knuth is an American author. is an American computer scientist, and mathematician, and was a professor at Stanford University. He received the ACM Turing Award in 1974, which is regarded as the “Nobel Prize” in computer science. Knuth has been named the “**Father of Algorithm Analysis**.”

The following machine learning algorithms we will use to treat our problem:

* Logistic Regression
* Decision Tree Classifier
* Random Forest Classifier
* XGBoost
* AdaBoost

1. **Statement of the problem**

Credit card fraud is an inclusive term for **fraud committed using a payment card**, such as a credit card or debit card. The purpose may be to obtain goods or services or to make payment to another account, which is controlled by a criminal. The thief either physically steals your credit card, or steals your card information via phishing or credit card skimming. Credit Cards were made available for the people to increase their buying power, it is an agreement with your bank that lets the user use the money lent by the bank in exchange for the repayment of this money on the due date or incur interest charges. With the rise in e-commerce and the recent boom of OTT platforms during the Coronavirus Pandemic, use of credit cards has risen exponentially along with other payment processes. As all the things in nature are binary, cases of credit card frauds were also achieved in high numbers. Global economy pays the price of more than $ 24 billion per year due to these frauds. Thus, building automated models for such a rising problem statement is necessary and ML Algorithm is the key for it!

1. **Objective**

Our primary objective is to extract maximum knowledge about the data i.e. the story behind the data and find some ML algorithms which are best for this data to detect fraudulent transactions.

At first, we will build some machine learning models on this dataset and then choose a model which is best in some sense.

1. **Description of the dataset**

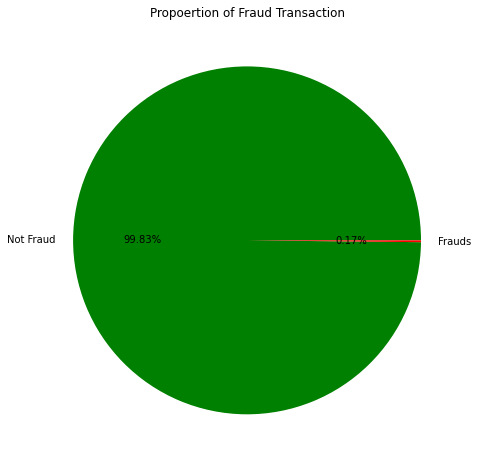
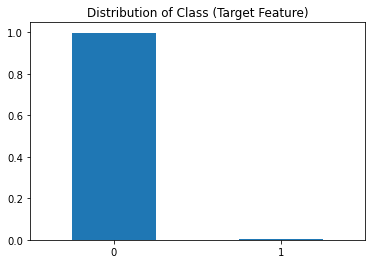
The dataset contains transactions made by credit cards in September 2013 by European cardholders.

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 numeric input variables which are the result of a PCA transformation, due to confidentiality issues. 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. Feature 'Class' is the response variable and it takes value 1 in case of fraud and 0 otherwise. It has a total 284807 observations in which 1081 observations are duplicate.

**WEEK 1: Exploratory Data Analysis**

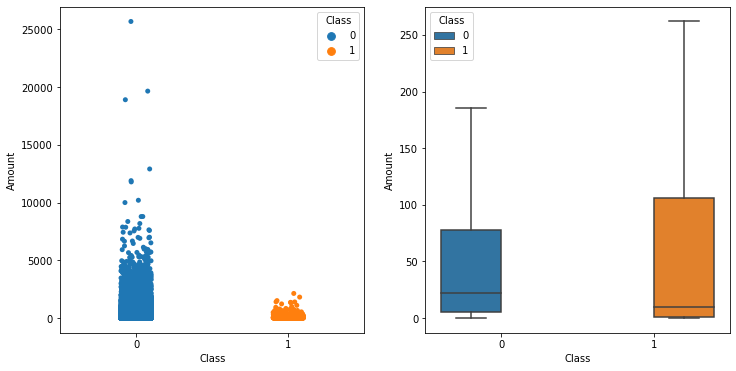
1. **Analysis of Response Variable**



**Inference:**

From the above visuals we can observe that most of the transactions are non-fraud. Approx 0.17% of all transactions are frauds. Also, those graphs indicate that our data is **highly imbalanced.**

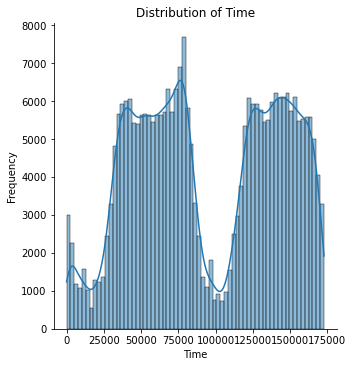
**6 Analysis of Predictor features**

**A. Analysis of ‘Amount’ feature**: 

**Inference:**

Most of the transaction amount falls between 0 and about 5000 and we have some outliers for really big amount transactions. It is observed that the data contains outliers. The highest amount is over 25000.

**B. Analysis of ‘Time’ feature:**



**Inference:**

From the graph, we can see there are two peaks and even there are some local peaks. We can think of these as the time of the day like the peak is the day time when most people do the transactions and the depth is the night time when most people are just sleeping.

**WEEK 2: Data Preprocessing**

Most of the transactions are non-fraud. If we use this dataframe as the base for our predictive models and analysis, our algorithms will probably overfit since it will "assume" that most transactions are not a fraud. But we don't want our model to assume, we want our model to detect patterns that give signs of **fraud!**

So, first we have to make our dataset balanced before building a model using some statistical technique.

We will use Undersampling and Oversampling technique to balance our dataset

**Undersampling Technique:**

Undersampling is a statistical technique used to reduce the number of observations of the class (Target feature) which has higher variables. We will use Random Under sampling technique to reduce the number of observations which belong to non-frauds transactions.

**Oversampling Technique:**

Oversampling is a statistical technique used to create new sample data with minority class. We will use SMOTE (Synthetic Minority Oversampling Technique).

**SMOTE:** SMOTE starts by picking a minority class instance at random and then looking for its k closest minority class neighbours. The synthetic instance is then constructed by randomly selecting one of the k nearest neighbours b and connecting a and b in the feature space to form a line segment. The synthetic instances are created by combining the two chosen examples a and b in a convex way.



**Inference:**

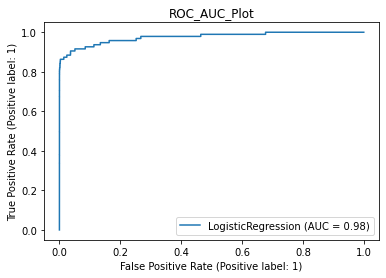
After using undersampling and oversampling techniques we get a nearly balanced data.

**WEEK 3: Model Building**

**A. Logistic Regression:**

Cross Validation Score : 99.1522%

ROC\_AUC Score : 92.4518%



Classification Report:

precision recall f1-score support

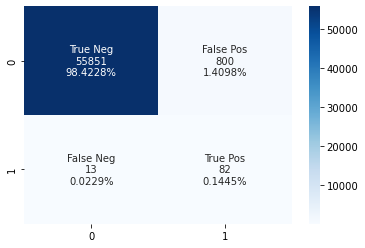
0 1.00 0.99 0.99 56651

1 0.09 0.86 0.17 95

accuracy 0.99 56746

macro avg 0.55 0.92 0.58 56746

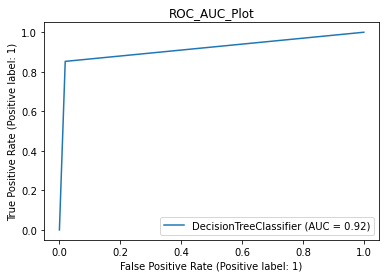
weighted avg 1.00 0.99 0.99 56746



**B. Decision Tree Classifier.**

Cross Validation Score : 98.19%

ROC\_AUC Score : 91.68%



Classification Report:

precision recall f1-score support

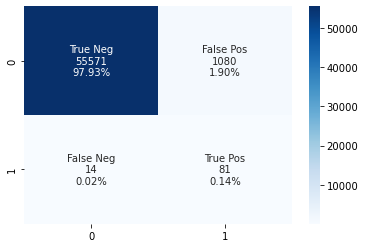
0 1.00 0.98 0.99 56651

1 0.07 0.85 0.13 95

accuracy 0.98 56746

macro avg 0.53 0.92 0.56 56746

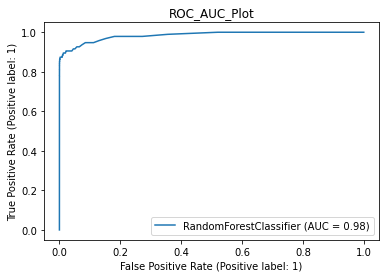
weighted avg 1.00 0.98 0.99 56746



**C. Random Forest Classifier:**

Cross Validation Score : 99.97%

ROC\_AUC Score : 93.56%

****

Classification Report:

precision recall f1-score support

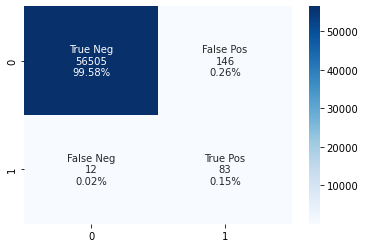
0 1.00 1.00 1.00 56651

1 0.36 0.87 0.51 95

accuracy 1.00 56746

macro avg 0.68 0.94 0.76 56746

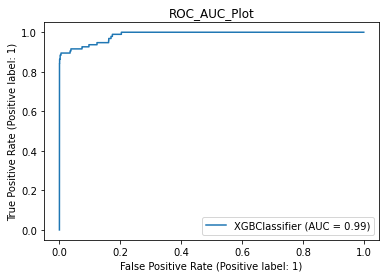
weighted avg 1.00 1.00 1.00 56746

****

**D. XGBoost Classifier:**

Cross Validation Score : 99.99%

ROC\_AUC Score : 94.04%



Classification Report:

precision recall f1-score support

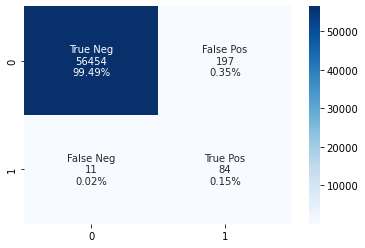
0 1.00 1.00 1.00 56651

1 0.30 0.88 0.45 95

accuracy 1.00 56746

macro avg 0.65 0.94 0.72 56746

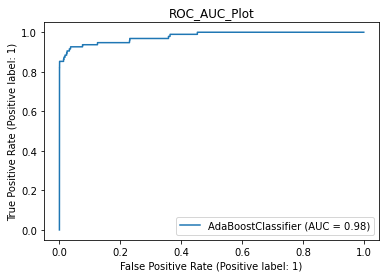
weighted avg 1.00 1.00 1.00 56746



**E. AdaBoost Classifier:**

Cross Validation Score : 99.78%

ROC\_AUC Score : 92.01%



Classification Report:

precision recall f1-score support

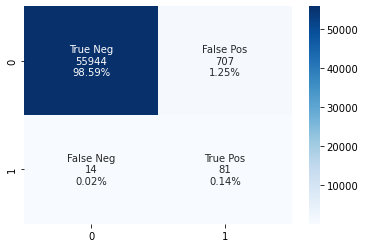
0 1.00 0.99 0.99 56651

1 0.10 0.85 0.18 95

accuracy 0.99 56746

macro avg 0.55 0.92 0.59 56746

weighted avg 1.00 0.99 0.99 56746



**WEEK 4: Model Selection**

***ML Algorithm Results Table :***

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Sl. No**. | **ML Algorithms** | **Cross Validation Score** | **ROC AUC Score** | **Weighted average F1 Score** |
| 1 | Logistic Regression | 98.83% | 91.74% | 0.99 |
| 2 | Decision Tree Classifier | 98.19% | 93.74% | 0.99 |
| 3 | Random Forest Classifier | 99.96% | 89.37% | 1 |
| 4 | XGBoost Classifier | 99.98% | 89.87% | 1 |
| 5 | AdaBoost Classifier | 99.75 | 90.99% | 0.99 |

So, on the basis of weighted average f1-score we can conclude that **Random Forest** and **XGBOOST** are the best models for this dataset

**Results:**

1. We find that for this data **Random Forest** and **XGBoost** are best models

2. We also find that out of all predictor features only **V17,V16,V14,V12,V10,V7,V4,V3** are relatively highly correlated with the response feature **Class**.

Link:> <https://github.com/Sudipta570/Credit-card-fraud/tree/main>

**References:**

1. Source dataset:<https://www.kaggle.com/datasets/mlgulb/creditcardfraud?q=credit+card+fraud+detection+in%3Adatasets>

2.Websites:

<https://medium.com/analytics-vidhya/undersampling-and-oversampling-an-old-and-a-new-approach-4f984a0e8392#:~:text=Undersampling%20and%20oversampling%20are%20techniques%20used%20to%20combat,of%20other%20classes%20whether%20it%E2%80%99s%20one%20or%20multiple>.

<https://www.geeksforgeeks.org/regression-classification-supervised-machine-learning/>