INTRODUCTION:

Credit card fraud occurs when someone uses another person's credit card information without authorization to make unauthorized purchases or transactions. This deceptive activity can involve stolen physical cards, compromised card details through data breaches, or various online scams. Credit card fraud poses financial risks for individuals and businesses, leading to unauthorized charges, compromised personal data, and potential damage to credit scores. To combat this, financial institutions employ security measures such as fraud detection algorithms and two-factor authentication to enhance cardholder protection.

Credit card fraud detection using machine learning involves leveraging algorithms to analyze patterns and detect unusual activities in credit card transactions. Various machine learning techniques, such as supervised learning, unsupervised learning, and anomaly detection, can be employed for this purpose. In supervised learning, models are trained on labeled data to distinguish between legitimate and fraudulent transactions. Unsupervised learning, on the other hand, can identify anomalies without prior labeling, making it effective for detecting novel fraud patterns. Feature engineering plays a crucial role, as relevant transaction features need to be extracted for the algorithms to identify suspicious behavior.

Logistic regression

Logistic regression is a supervised machine learning algorithm mainly used for classification tasks where the goal is to predict the probability that an instance belongs to a given class or not. It is a kind of statistical algorithm, which analyze the relationship between a set of independent variables and the dependent binary variables. It is a powerful tool for decision-making. For example email spam or not. sigmoid function that takes input as independent variables and produces a probability value between 0 and 1. For example, we have two classes Class 0 and Class 1 if the value of the logistic function for an input is greater than 0.5 (threshold value) then it belongs to Class 1 it belongs to Class 0. It’s referred to as regression because it is the extension of linear regression but is mainly used for classification problems. The difference between linear regression and logistic regression is that linear regression output is the continuous value that can be anything while logistic regression predicts the probability that an instance belongs to a given class or not.

Basic Libraries import

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

import math

import sklearn

import numpy as np

import warnings

warnings.filterwarnings('ignore')

%matplotlib inline

Reading the dataset

df = pd.read_csv("creditcardd.csv")

Number of rows and columns

Print ('Total de linhas e colunas\n\n',df.shape,'\n')

Total de linhas e colunas

(284807, 31)

Verification of the existence of null or missing values

df.isnull().sum()

Time 0

V1 0
V2 0
V3 0
V4 0
V5 0
V6 0
V7 0
V8 0
V9
V10 0
V11 0
V12 0
V13 0
V14 0
V15 0
V16 0
V17 0
V18 0

V19 0
V20 0
V21 0
V22 0
V23 0
V24 0
V25 0
V26 0
V27 0
V28 0
Amount 0
Class 0
dtype: int64

Variable type in each column

df.info()

<class 'pandas.core.frame.DataFrame'>

RangeIndex: 284807 entries, 0 to 284806

Data columns (total 31 columns):

#Column		Non-Null Count	Dtype
0	**Time**	**284807 non-null**	**float64**
1	**V1**	**284807 non-null**	**float64**
2	**V2**	**284807 non-null**	**float64**
3	**V3**	**284807 non-null**	**float64**
4	**V4**	**284807 non-null**	**float64**
5	**V5**	**284807 non-null**	**float64**
6	**V6**	**284807 non-null**	**float64**
7	**V7**	**284807 non-null**	**float64**
8	**V8**	**284807 non-null**	**float64**
9	**V9**	**284807 non-null**	**float64**
10	**V10**	**284807 non-null**	**float64**
11	**V11**	**284807 non-null**	**float64**
12	**V12**	**284807 non-null**	**float64**
13	**V13**	**284807 non-null**	**float64**
14	**V14**	**284807 non-null**	**float64**
15	**V15**	**284807 non-null**	**float64**
16	**V16**	**284807 non-null**	**float64**
17	**V17**	**284807 non-null**	**float64**
18	**V18**	**284807 non-null**	**float64**
19	**V19**	**284807 non-null**	**float64**
20	**V20**	**284807 non-null**	**float64**
21	**V21**	**284807 non-null**	**float64**

```
 22  V22     284807 non-null  float64
 23  V23     284807 non-null  float64
 24  V24     284807 non-null  float64
 25  V25     284807 non-null  float64
 26  V26     284807 non-null  float64
 27  V27     284807 non-null  float64
 28  V28     284807 non-null  float64
 29 Amount  284807 non-null  float64
 30  Class    284807 non-null  int64
dtypes: float64(30), int64(1)
memory usage: 67.4 MB
```

Statistical information in each class

```
print ('Not Fraud % ',round(df['Class'].value_counts()[0]/len(df)*100)
print ()
print (round(df.Amount[df.Class == 0].describe(),2))
print ()
print ()
print ('Fraud %',round(df['Class'].value_counts()[1]/len(df)*100,2)
print ()
print (round(df.Amount[df.Class == 1].describe(),2))
```

```
Not Fraud %  99.83
count    284315.00
mean         88.29
std         250.11
min           0.00
25%           5.65
50%          22.00
75%          77.05
max       25691.16
Name: Amount, dtype: float64
Fraud %     0.17
count     492.00
mean      122.21
std       256.68
min         0.00
25%         1.00
50%         9.25
75%       105.89
```

max 2125.87

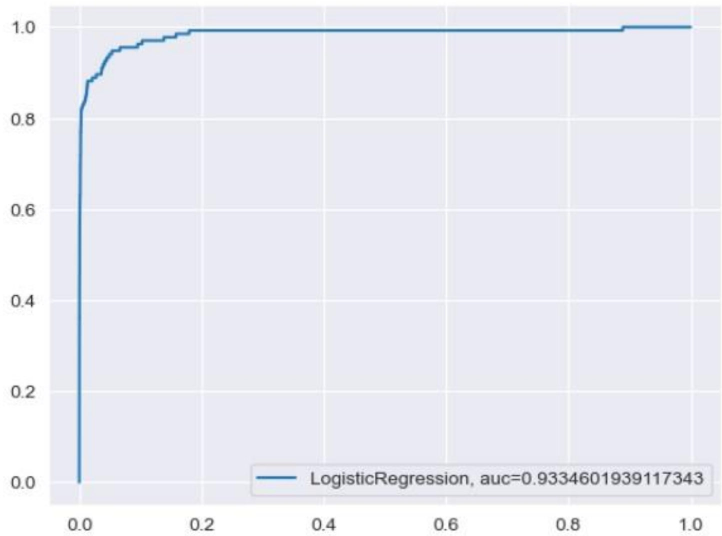
Name: Amount, dtype: float64

Comparing the amount value of normal transactions versus fraud

```
plt.figure(figsize=(10,8))
sns.set_style('darkgrid')
sns.barplot(x=df['Class'].value_counts().index,y=df['Class'].value_counts(), palette=["C1", "C
8"])
plt.title('Non Fraud X Fraud')
plt.ylabel('Count')
plt.xlabel('0: Non Fraud,  1: Fraud')
print ('Non Fraud % ',round(df['Class'].value_counts()[0]/len(df)*100,2))
print ('Fraud %    ',round(df['Class'].value_counts()[1]/len(df)*100,2));
Non Fraud %  99.83
Fraud %     0.17
```

Measurement of classifier performance through the ROC and AUC curve

```
from sklearn import metrics
 clf=LogisticRegression(C=1, penalty='l2')
clf.fit(X_undersampled_train, Y_undersampled_train)
y_pred = clf.predict(X_test)
y_pred_probability = clf.predict_proba(X_test)[::,1]
fpr, tpr, _ = metrics.roc_curve(y_test, y_pred_probability)
auc = metrics.roc_auc_score(y_test, pred)
plt.plot(fpr,tpr,label="LogisticRegression, auc="+str(auc))
plt.legend(loc=4)
plt.show()
```



print(list(y_pred))

[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 , 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0]

import pandas as pd

value1=X_test[8:9]

print('---')

print(X_test.iloc[7,:])

V1	**-0.992899**
V2	**1.430204**
V3	**1.071256**
V4	**1.363127**
V5	**0.116315**
V6	**0.217868**
V7	**0.208391**
V8	**0.319128**
V9	**1.483134**
V10	**-0.014515**
V11	**-0.277216**
V12	**-3.182761**
V13	**0.093851**
V14	**1.724100**

V15 -0.314430

V16 -1.069052

V17 1.364881

V18 -0.121760

V19 0.645493

V20 0.048699

V21 -0.258903

V22 -0.104189

V23 -0.100144

V24 -0.369103

V25 -0.068048

V26 -0.266731

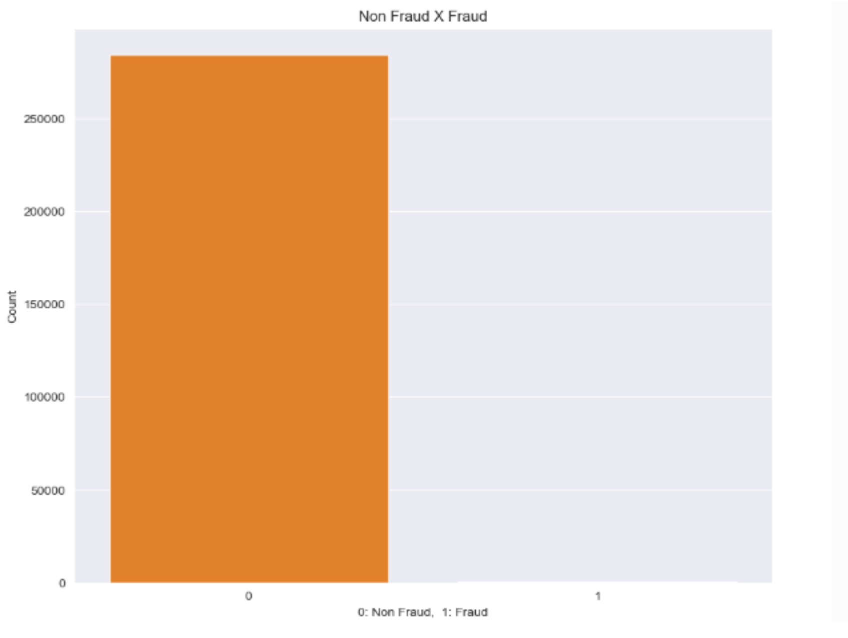
V27 0.080402

V28 -0.034571

Amount 1.000000

Name: 13629, dtype: float64

OUTPUT:



predicted_value=clf.predict(value1)

print(predicted_value)

[0]