Credit card fraud detection

# Program:

Program for Data visualization :

Import numpy as np

Import pandas as pd

Import seaborn as sns

Import matplotlib.pyplot as plt

From sklearn.model\_selection import train\_test\_split

From imblearn.over\_sampling import SMOTE

From sklearn.preprocessing import StandardScaler

From sklearn.ensemble import RandomForestClassifier

From sklearn.metrics import confusion\_matrix, classification\_report

Df=pd.read\_csv(‘/content/drive/MyDrive/creditcard.csv’)

Df

Df.info()

Df.describe()

Df[‘Class’].unique()

Df[‘Class’].value\_counts()

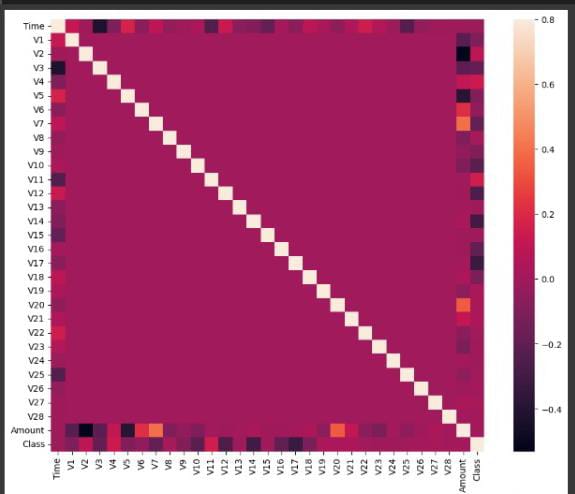
Corr = df.corr()

Fig = plt.figure(figsize = (12, 9))

Sns.heatmap(corr, vmax = .8, square = True)

Plt.show()

**Output:**



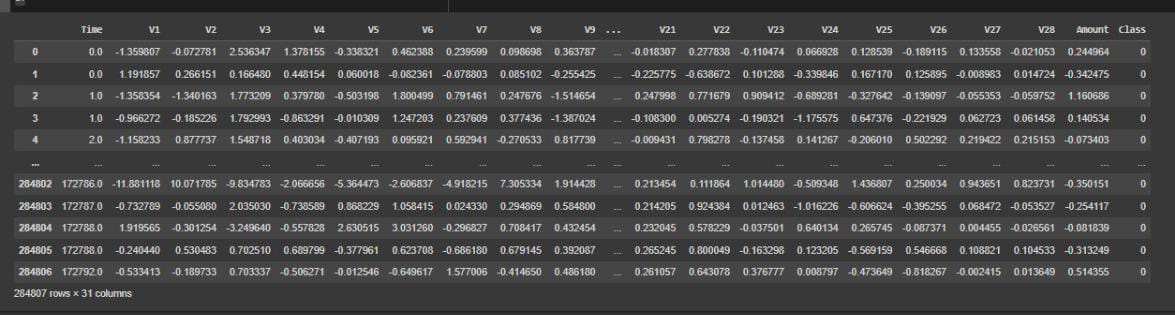
**Program 1:**

Df[‘Amount’] = StandardScaler().fit\_transform(np.array(df[‘Amount’]).reshape(-1,1))

Df

Df[‘Amount’] = StandardScaler().fit\_transform(np.array(df[‘Amount’]).reshape(-1, 1))

D

**Output 1:**

**PROGRAM 2:**

X=df.drop([‘Class’], axis=1)

Y=df[“Class”]

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size = 0.3, random\_state = 0)

Sm = SMOTE() #solving class imbalance problem

X\_train\_res, y\_train\_res = sm.fit\_resample(X\_train, y\_train.ravel())

Rfc=RandomForestClassifier()

Rfc.fit(X\_train\_res,y\_train\_res)

Y\_pred = rfc.predict(X\_test)

Print(classification\_report(y\_test, y\_pred))

**OUTPUT 2:**

Precision Recall f1-score Support

0 1.00 1.00 1.00 85296

1 0.91 0.83 0.87 147

Accuracy

Macro avg 0.96 0.91 0.93 85443

Weighted avg 1.00 1.00 1.00 85443

**PROGRAM 3:**

LABELS = [‘Normal’, ‘Fraud’]

Conf\_matrix = confusion\_matrix(y\_test, y\_pred)

Plt.figure(figsize=(12, 12))

Sns.heatmap(conf\_matrix, xticklabels=LABELS, yticklabels=LABELS, annot=True, fmt=”d”);

Plt.title(“Confusion matrix”)

Plt.ylabel(‘True class’)

Plt.xlabel(‘Predicted class’)

Plt.show()

**Output 3:**

