**PHASE -4**

**CREDIT CARD FRAUD DETECTION**

|  |  |
| --- | --- |
| **NAME:** | **NM ID:** |
| R.ABINAYA | D7A51E6301AAC2898AF49A824DCD7A90 |
| S.ANBARASI | AC6081E93C2455699F14343B41655F81 |
| K.BHUVANASURUTHI | A5D1B0C0ADB088EB0914A8650735635B |
| M.GAYATHRI | F761865DB95909598C14D1AF5B1658B8 |

**FEATURE ENGINEERING:**

Feature engineering for credit card fraud detection involves selecting, creating, or transforming data attributes (features) to improve the accuracy of fraud detection models. This includes things like analyzing transaction patterns, extracting relevant information, and designing features that help algorithms identify fraudulent activity.

**MODEL TRAINING:**

Model training for credit card fraud detection is the process of using historical transaction data to teach a machine learning or AI model to recognize patterns and anomalies indicative of fraudulent transactions. It involves feeding the model with labeled examples (fraudulent and non-fraudulent transactions) to enable it to learn and make predictions about new, unseen transactions.

**EVALUATION:**

Evaluation for credit card fraud detection involves assessing the performance of a fraud detection model. It includes metrics like precision, recall, F1 score, and accuracy to measure how well the model identifies fraudulent transactions while minimizing false positives. A higher evaluation score indicates a more effective fraud detection system.

**PROGRAM 1:**

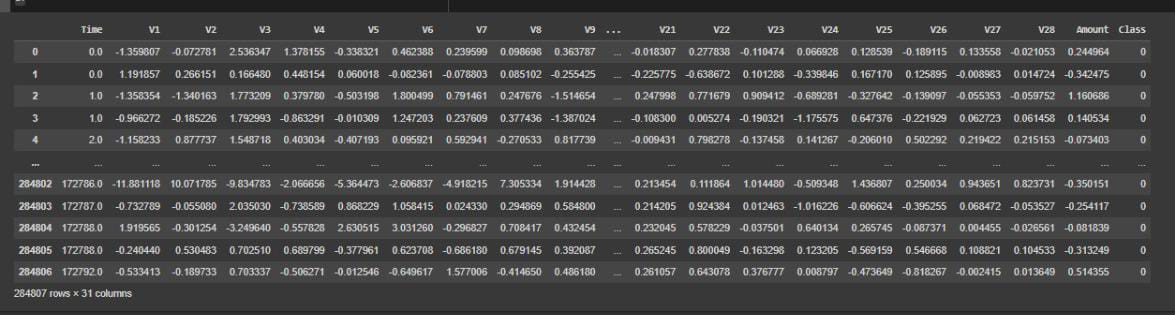
df['Amount'] = StandardScaler().fit\_transform(np.array(df['Amount']).reshape(-1,1))

df

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

df

**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:**

