ML Lab-4

Praneesh R V

CB.SC.U4CYS23036

1. Use StandardScaler to standardize the features of a Credit card fraud dataset. Include code, description and screenshots of outputs.

! pip install Kaggle

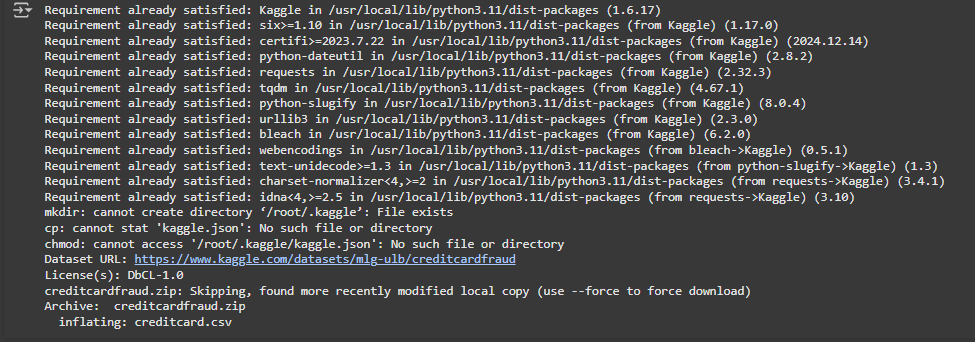
! mkdir ~/.kaggle

! cp kaggle.json ~/.kaggle/

! chmod 600 ~/.kaggle/kaggle.json

!kaggle datasets download -d mlg-ulb/creditcardfraud

!unzip creditcardfraud.zip



Uploading the file first.

import pandas as pd

from sklearn.preprocessing import StandardScaler

data = pd.read\_csv('creditcard.csv')

X = data.drop('Class', axis=1)

y = data['Class']

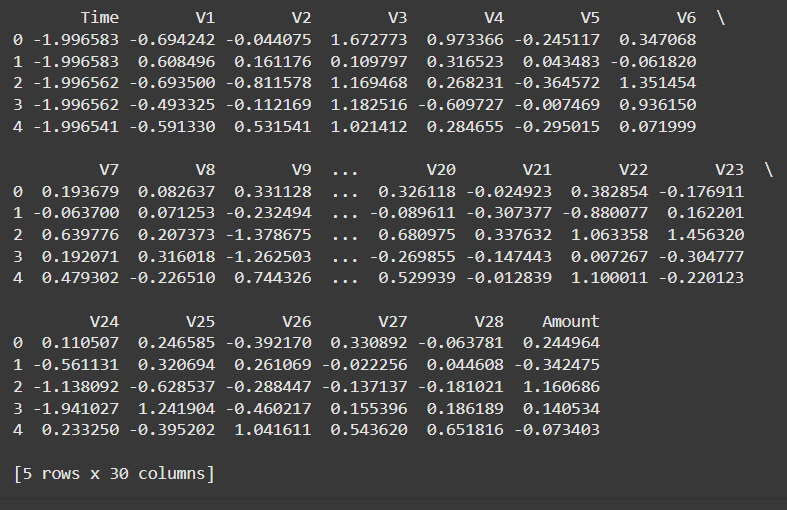
scaler = StandardScaler()

X\_scale = scaler.fit\_transform(X)

X\_scale\_df = pd.DataFrame(X\_scale, columns=X.columns)

print(X\_scale\_df.head())

reading the dataset and scaling it. Displaying the head values.



from imblearn.over\_sampling import SMOTE

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

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X\_scaled, y, test\_size=0.2, random\_state=42)

smote = SMOTE(random\_state=42)

X\_train\_resampled, y\_train\_resampled = smote.fit\_resample(X\_train, y\_train)

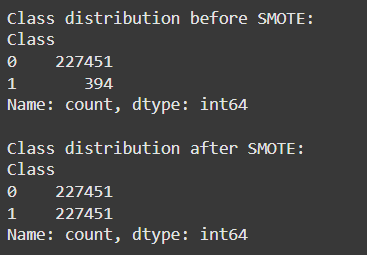
print("Class distribution before SMOTE:")

print(y\_train.value\_counts())

print("\nClass distribution after SMOTE:")

print(pd.Series(y\_train\_resampled).value\_counts())

using SMOTE to equalize the number of class(0) and class(1) samples.



from sklearn.tree import DecisionTreeClassifier

from sklearn.metrics import confusion\_matrix, accuracy\_score, precision\_score, recall\_score, f1\_score

model = DecisionTreeClassifier(random\_state=42)

model.fit(X\_train\_resampled, y\_train\_resampled)

y\_pred = model.predict(X\_test)

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

accuracy = accuracy\_score(y\_test, y\_pred)

precision = precision\_score(y\_test, y\_pred)

recall = recall\_score(y\_test, y\_pred)

f1 = f1\_score(y\_test, y\_pred)

print("Confusion Matrix:")

print(conf\_matrix)

print("\nAccuracy:", accuracy)

print("Precision:", precision)

print("Recall:", recall)

print("F1 Score:", f1)

creating a decision tree classifier and other performance measures to show the quality of the model.

