

## 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
```

```
Requirement already satisfied: Kaggle in /usr/local/lib/python3.11/dist-packages (1.6.17)
Requirement already satisfied: six>=1.10 in /usr/local/lib/python3.11/dist-packages (from Kaggle) (1.17.0)
Requirement already satisfied: certifi>=2023.7.22 in /usr/local/lib/python3.11/dist-packages (from Kaggle) (2024.12.14)
Requirement already satisfied: python-dateutil in /usr/local/lib/python3.11/dist-packages (from Kaggle) (2.8.2)
Requirement already satisfied: requests in /usr/local/lib/python3.11/dist-packages (from Kaggle) (2.32.3)
Requirement already satisfied: tqdm in /usr/local/lib/python3.11/dist-packages (from Kaggle) (4.67.1)
Requirement already satisfied: python-slugify in /usr/local/lib/python3.11/dist-packages (from Kaggle) (8.0.4)
Requirement already satisfied: urllib3 in /usr/local/lib/python3.11/dist-packages (from Kaggle) (2.3.0)
Requirement already satisfied: bleach in /usr/local/lib/python3.11/dist-packages (from Kaggle) (6.2.0)
Requirement already satisfied: webencodings in /usr/local/lib/python3.11/dist-packages (from bleach->Kaggle) (0.5.1)
Requirement already satisfied: text-unidecode>=1.3 in /usr/local/lib/python3.11/dist-packages (from python-slugify->Kaggle) (1.3)
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.11/dist-packages (from requests->Kaggle) (3.4.1)
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.11/dist-packages (from requests->Kaggle) (3.10)
mkdir: cannot create directory '/root/.kaggle': File exists
cp: cannot stat 'kaggle.json': No such file or directory
chmod: cannot access '/root/.kaggle/kaggle.json': No such file or directory
Dataset URL: https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud
License(s): DbCL-1.0
creditcardfraud.zip: Skipping, found more recently modified local copy (use --force to force download)
Archive: creditcardfraud.zip
  inflating: creditcard.csv
```

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.

```

      Time      V1      V2      V3      V4      V5      V6 \
0 -1.996583 -0.694242 -0.044075  1.672773  0.973366 -0.245117  0.347068
1 -1.996583  0.608496  0.161176  0.109797  0.316523  0.043483 -0.061820
2 -1.996562 -0.693500 -0.811578  1.169468  0.268231 -0.364572  1.351454
3 -1.996562 -0.493325 -0.112169  1.182516 -0.609727 -0.007469  0.936150
4 -1.996541 -0.591330  0.531541  1.021412  0.284655 -0.295015  0.071999

      V7      V8      V9      ...      V20      V21      V22      V23 \
0  0.193679  0.082637  0.331128  ...  0.326118 -0.024923  0.382854 -0.176911
1 -0.063700  0.071253 -0.232494  ... -0.089611 -0.307377 -0.880077  0.162201
2  0.639776  0.207373 -1.378675  ...  0.680975  0.337632  1.063358  1.456320
3  0.192071  0.316018 -1.262503  ... -0.269855 -0.147443  0.007267 -0.304777
4  0.479302 -0.226510  0.744326  ...  0.529939 -0.012839  1.100011 -0.220123

      V24      V25      V26      V27      V28      Amount
0  0.110507  0.246585 -0.392170  0.330892 -0.063781  0.244964
1 -0.561131  0.320694  0.261069 -0.022256  0.044608 -0.342475
2 -1.138092 -0.628537 -0.288447 -0.137137 -0.181021  1.160686
3 -1.941027  1.241904 -0.460217  0.155396  0.186189  0.140534
4  0.233250 -0.395202  1.041611  0.543620  0.651816 -0.073403

[5 rows x 30 columns]
```

```
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.

```
Class distribution before SMOTE:
Class
0      227451
1         394
Name: count, dtype: int64

Class distribution after SMOTE:
Class
0      227451
1      227451
Name: count, dtype: int64
```

```
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.

Confusion Matrix:

```
[[56745  119]
 [   21   77]]
```

Accuracy: 0.997542221129876

Precision: 0.39285714285714285

Recall: 0.7857142857142857

F1 Score: 0.5238095238095238