CS 5710
MACHINE LEARNING

Name: Shiva Kandagatla
Student Id: 700745245
E-mail: SXK54250@ucmo.edu
Course: CS 5710
Assignment: Assignment #5

GitHub Link:
https://github.com/Shiva-Kandagatla98/In-Class-Assignment-5.git

Video Demo Link:

1. Principal Component Analysis
   a. Apply PCA on CC dataset.
   b. Apply k-means algorithm on the PCA result and report your observation if the silhouette score has improved or not?
   c. Perform Scaling+PCA+K-Means and report performance.

Source Code:

```python

import pandas as pd
from sklearn.decomposition import PCA
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split
from sklearn.cluster import KMeans
from sklearn import metrics
import warnings
warnings.filterwarnings("ignore")

# Reading dataset CC.csv
import pandas as pd
from sklearn.decomposition import PCA
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split
from sklearn.cluster import KMeans
from sklearn import metrics
from sklearn.linear_model import LogisticRegression

# Load dataset
df = pd.read_csv('CC GENERAL.csv')

# Replace null values with mean
df['MINIMUM_PAYMENTS'].fillna(df['MINIMUM_PAYMENTS'].mean(), inplace=True)
df['CREDIT_LIMIT'].fillna(df['CREDIT_LIMIT'].mean(), inplace=True)

# Apply PCA on dataset
pca = PCA(n_components=2)
X_pca = pca.fit_transform(df.iloc[:, 1:18])
pca_df = pd.DataFrame(data=X_pca, columns=['PC1', 'PC2'])
final_df = pd.concat([pca_df, df[['TENURE']]], axis=1)
print(final_df)

# Split into train and test sets
X = final_df.drop('TENURE', axis=1).values
y = final_df['TENURE'].values
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=0)

```

question1.py

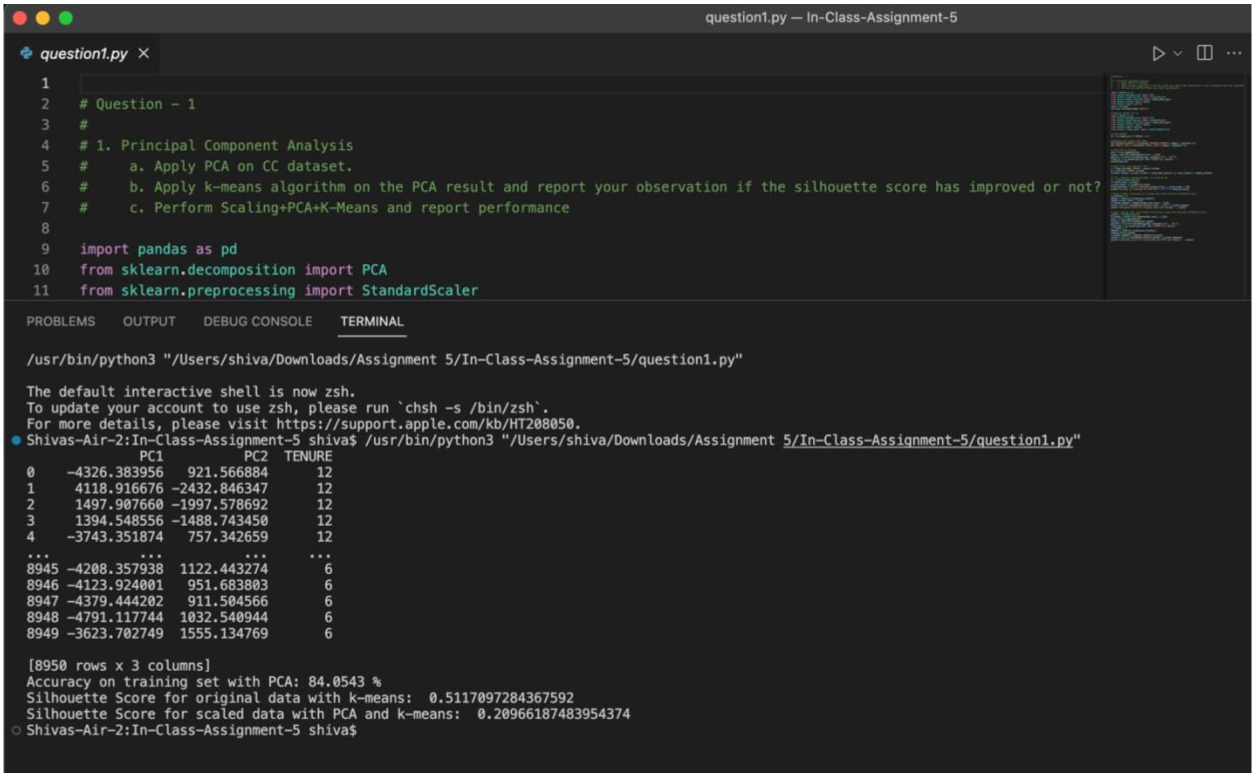
```python
# Split into train and test sets
X = final_df.drop('TENURE', axis=1).values
y = final_df['TENURE'].values
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=0)

# Train logistic regression model on training set
lr = LogisticRegression()
lr.fit(X_train, y_train)
y_train_pred = lr.predict(X_train)
train_accuracy = metrics.accuracy_score(y_train, y_train_pred) * 100
print('Accuracy on training set with PCA: %.4f %%' % train_accuracy)

# Apply k-means clustering on original data and calculate Silhouette score
n_clusters = 2
kmeans = KMeans(n_clusters=n_clusters)
kmeans.fit(df.iloc[:, 1:18])
y_cluster_kmeans = kmeans.predict(df.iloc[:, 1:18])
score = metrics.silhouette_score(df.iloc[:, 1:18], y_cluster_kmeans)
print('Silhouette Score for original data with k-means: ', score)

# Apply scaling, PCA, and k-means clustering on data and calculate Silhouette score
scaler = StandardScaler()
X_scaled = scaler.fit_transform(df.iloc[:, 1:18])
pca2 = PCA(n_components=2)
X_pca2 = pca2.fit_transform(X_scaled)
pca_df2 = pd.DataFrame(data=X_pca2, columns=['PC1', 'PC2'])
final_df2 = pd.concat([pca_df2, df[['TENURE']]], axis=1)
n_clusters = 2
kmeans2 = KMeans(n_clusters=n_clusters)
kmeans2.fit(X_scaled)
y_cluster_kmeans2 = kmeans2.predict(X_scaled)
score2 = metrics.silhouette_score(X_scaled, y_cluster_kmeans2)
print('Silhouette Score for scaled data with PCA and k-means: ', score2)
```

Output:



question1.py — In-Class-Assignment-5

question1.py

```python

# Question - 1
#
# 1. Principal Component Analysis
#     a. Apply PCA on CC dataset.
#     b. Apply k-means algorithm on the PCA result and report your observation if the silhouette score has improved or not?
#     c. Perform Scaling+PCA+K-Means and report performance

import pandas as pd
from sklearn.decomposition import PCA
from sklearn.preprocessing import StandardScaler
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
/usr/bin/python3 "/Users/shiva/Downloads/Assignment 5/In-Class-Assignment-5/question1.py"

The default interactive shell is now zsh.
To update your account to use zsh, please run `chsh -s /bin/zsh`.
For more details, please visit https://support.apple.com/kb/HT208050.
Shivas-Air-2:In-Class-Assignment-5 shiva$ /usr/bin/python3 "/Users/shiva/Downloads/Assignment 5/In-Class-Assignment-5/question1.py"
              PC1          PC2  TENURE
0    -4326.383956   921.566884      12
1     4118.916676 -2432.846347      12
2     1497.907660 -1997.578692      12
3     1394.548556 -1488.743450      12
4    -3743.351874   757.342659      12
...           ...          ...     ...
8945 -4208.357938  1122.443274       6
8946 -4123.924001   951.683803       6
8947 -4379.444202   911.504566       6
8948 -4791.117744  1032.540944       6
8949 -3623.702749  1555.134769       6

[8950 rows x 3 columns]
Accuracy on training set with PCA: 84.0543 %
Silhouette Score for original data with k-means:  0.5117097284367592
Silhouette Score for scaled data with PCA and k-means:  0.20966187483954374
Shivas-Air-2:In-Class-Assignment-5 shiva$
```

---------------*******---------------

2. Use pd_speech_features.csv

- a. Perform Scaling
- b. Apply PCA (k=3)
- c. Use SVM to report performance

Source Code:

question2.py

```python
# 2. Use pd_speech_features.csv
#     a. Perform Scaling
#     b. Apply PCA (k=3)
#     c. Use SVM to report performance

import pandas as pd
from sklearn.preprocessing import StandardScaler
from sklearn.decomposition import PCA
from sklearn.model_selection import train_test_split
from sklearn.svm import SVC

# Load dataset
data = pd.read_csv("pd_speech_features.csv")
features = data.drop('class', axis=1).values
labels = data['class'].values

# Apply scaling
scaler = StandardScaler()
scaled_features = scaler.fit_transform(features)

# PCA
pca = PCA(n_components=3)
pca_components = pca.fit_transform(scaled_features)
pca_df = pd.DataFrame(data=pca_components,
                      columns=['PC 1', 'PC 2', 'PC 3'])
final_df = pd.concat([pca_df, data[['class']]], axis=1)
print('Final dataset after PCA:')
print(final_df)

# Train and evaluate SVM model
X_train, X_test, y_train, y_test = train_test_split(scaled_features, labels, test_size=0.3, random_state=0)
svm_model = SVC(max_iter=1000)
svm_model.fit(X_train, y_train)
y_pred = svm_model.predict(X_test)
svm_acc = round(svm_model.score(X_train, y_train) * 100, 2)
print("SVM accuracy =", svm_acc)
```

<u>Output</u>:

question2.py — In-Class-Assignment-5

question2.py

```python
# 2. Use pd_speech_features.csv
#     a. Perform Scaling
#     b. Apply PCA (k=3)
#     c. Use SVM to report performance

import pandas as pd
from sklearn.preprocessing import StandardScaler
from sklearn.decomposition import PCA
from sklearn.model_selection import train_test_split
from sklearn.svm import SVC

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL

```
The default interactive shell is now zsh.
To update your account to use zsh, please run `chsh -s /bin/zsh`.
For more details, please visit https://support.apple.com/kb/HT208050.
Shivas-Air-2:In-Class-Assignment-5 shiva$ /usr/bin/python3 "/Users/shiva/Downloads/Assignment 5/In-Class-Assignment-5/question1.py"
              PC1          PC2  TENURE
0    -4326.383956   921.566884      12
1     4118.916676 -2432.846347      12
2     1497.907660 -1997.578692      12
3     1394.548556 -1488.743450      12
4    -3743.351874   757.342659      12
...           ...          ...     ...
8945 -4208.357938  1122.443274       6
8946 -4123.924001   951.683803       6
8947 -4379.444202   911.504566       6
8948 -4791.117744  1032.540944       6
8949 -3623.702749  1555.134769       6

[8950 rows x 3 columns]
Accuracy on training set with PCA: 84.0543 %
Silhouette Score for original data with k-means:  0.5117097284367592
Silhouette Score for scaled data with PCA and k-means:  0.20966187483954374
Shivas-Air-2:In-Class-Assignment-5 shiva$ /usr/bin/python3 "/Users/shiva/Downloads/Assignment 5/In-Class-Assignment-5/question2.py"
Final dataset after PCA:
          PC 1      PC 2       PC 3  class
0   -10.047372  1.471076  -6.846408      1
1   -10.637725  1.583749  -6.830979      1
2   -13.516185 -1.253542  -6.818700      1
3    -9.155084  8.833601  15.290900      1
4    -6.764471  4.611467  15.637122      1
..         ...       ...        ...    ...
751  22.322682  6.481912   1.458757      0
752  13.442877  1.449413   9.352299      0
753   8.270263  2.391285  -0.908676      0
754   4.011760  5.412258  -0.847130      0
755   3.993112  6.072418  -2.020722      0

[756 rows x 4 columns]
SVM accuracy = 91.68
Shivas-Air-2:In-Class-Assignment-5 shiva$
```

---------------*******---------------

3. Apply Linear Discriminant Analysis (LDA) on Iris.csv dataset to reduce dimensionality of data to k=2.

Source Code:

question3.py U

```python
# 3. Apply Linear Discriminant Analysis (LDA) on Iris.csv dataset to reduce dimensionality of data tok=2.

import pandas as pd
from sklearn.preprocessing import StandardScaler, LabelEncoder
from matplotlib import pyplot as plt
from sklearn.discriminant_analysis import LinearDiscriminantAnalysis
import warnings
warnings.filterwarnings("ignore")
import seaborn as sns
sns.set(style="white", color_codes=True)


import pandas as pd
from sklearn.preprocessing import StandardScaler, LabelEncoder
from sklearn.discriminant_analysis import LinearDiscriminantAnalysis
import seaborn as sns
import matplotlib.pyplot as plt

# Load iris dataset
df = pd.read_csv("iris.csv")

# Standardize features
scaler = StandardScaler()
X = scaler.fit_transform(df.iloc[:, :4].values)

# Encode class labels
label_encoder = LabelEncoder()
y = label_encoder.fit_transform(df['Species'].values)

# Apply LDA
lda = LinearDiscriminantAnalysis(n_components=2)
X_lda = lda.fit_transform(X, y)

# Create new dataframe with LDA components and class labels
data = pd.DataFrame(X_lda, columns=['LD1', 'LD2'])
data['class'] = y

print(data)
```

Output:

```
Shivas-Air-2:In-Class-Assignment-5 shiva$ /usr/bin/python3 "/Users/shiva/Downloads/Assignment 5/In-Class-Assignment-5/question3.py"
           LD1       LD2  class
0     9.423452 -0.513976      0
1     8.751900 -1.591678      0
2     8.973004 -1.068204      0
3     8.170186 -1.435135      0
4     9.249789 -0.136869      0
..         ...       ...    ...
145  -6.167264  2.006912      2
146  -6.462099  1.062334      2
147  -6.447135  2.055824      2
148  -7.018075  2.727998      2
149  -6.838386  2.108245      2

[150 rows x 3 columns]
Shivas-Air-2:In-Class-Assignment-5 shiva$
```

---------------*****---------------**

4. Briefly identify the difference between PCA and LDA

PCA is an unsupervised technique that seeks to reduce the dimensionality of a dataset while preserving most of its variance. It does so by transforming the data into a new coordinate system where the new axes (principal components) are orthogonal and ordered by the amount of variance they capture. PCA is often used for data visualization, data compression, and feature extraction.

LDA, on the other hand, is a supervised technique that seeks to find a linear combination of features that maximizes the separation between different classes of data. It does so by finding a projection that maximizes the between-class scatter and minimizes the within-class scatter. LDA is often used for classification and feature extraction.

In summary, PCA is an unsupervised technique that preserves the maximum amount of variance in the data while reducing dimensionality, while LDA is a supervised technique that seeks to maximize the separation between classes by finding a projection that preserves the class structure of the data.

---------------*****---------------**