Experiment 07 : Implement Feature Extraction in Cognitive Computing.

Learning Objective : Students should be able to apply Feature Extraction for Cognitive Computing and solving problems.

Tool : Python

Theory :

Feature extraction is a critical step in cognitive computing that involves selecting and extracting relevant features from raw data to improve the accuracy and efficiency of machine learning algorithms.

Some common feature extraction techniques used in cognitive computing include:

- Principal Component Analysis (PCA)
- Discrete Wavelet Transform (DWT)
- Fourier Transform
- Mel-Frequency Cepstral Coefficients (MFCC)
- Local Binary Patterns (LBP)

The choice of technique depends on the type of data and the problem to be solved. These techniques can be implemented using various programming languages such as Python, R, and Java. The choice of programming language depends on the specific requirements of the project and the availability of libraries for implementing the algorithms

Implementation :

1. **Data Collection and Preprocessing :**

[2]:
```
# Data Collection and Preprocessing
from sklearn.datasets import load_iris

# Load the Iris dataset
iris = load_iris()
X = iris.data
y = iris.target
```

[3]:
```
# Preprocessing: Standardize the features
from sklearn.preprocessing import StandardScaler

scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)
```

2. **Feature Selection and Extraction :**

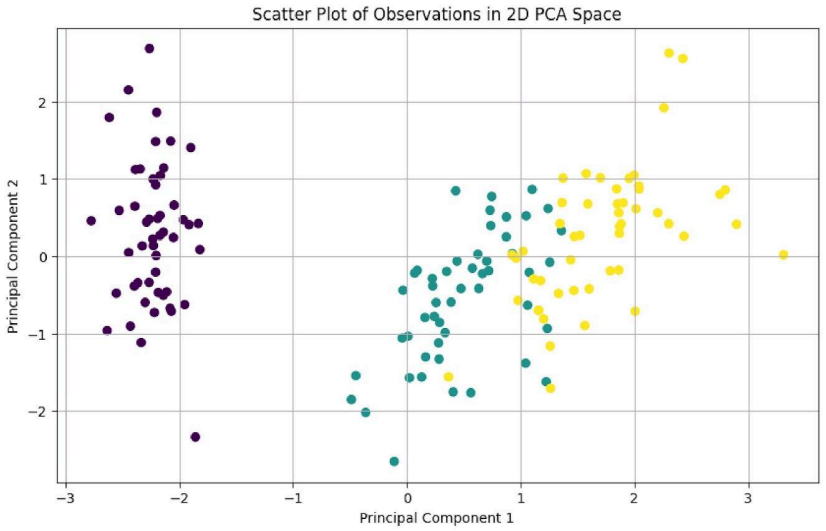
[4]:
```
# Feature Selection and Extraction: Apply PCA
from sklearn.decomposition import PCA

pca = PCA(n_components=2) # Reduce to 2 components
X_pca = pca.fit_transform(X_scaled)
```

3. Deployment :

[5]:
```
# Visualization
import matplotlib.pyplot as plt

plt.figure(figsize=(10, 6))
plt.scatter(X_pca[:, 0], X_pca[:, 1], c=y, cmap='viridis')
plt.xlabel('Principal Component 1')
plt.ylabel('Principal Component 2')
plt.title('Scatter Plot of Observations in 2D PCA Space')
plt.grid(True)
plt.show()
```



4. Testing :

[7]:
```
import numpy as np

explained_variance_ratio = pca.explained_variance_ratio_
print("Explained Variance Ratio:", explained_variance_ratio)

cumulative_variance_ratio = np.cumsum(explained_variance_ratio)
print("Cumulative Explained Variance Ratio:", cumulative_variance_ratio)
```

```
Explained Variance Ratio: [0.72962445 0.22850762]
Cumulative Explained Variance Ratio: [0.72962445 0.95813207]
```

Result and Discussion :

Learning Outcomes : Students should have the ability to

LO 1: Formulate the problem using AI and CC Approach.
LO 2: Solve the problem using Pattern Recognition Algorithm.

Course Outcomes :

CO : Apply cognitive computing in business implications.

Conclusion :

_____

_____

Viva Questions :

Q1. What is feature extraction, and why is it important in cognitive computing?
Q.2 How do you evaluate the effectiveness of feature extraction techniques?
Q.3 Can you explain how DWT works, and what are its benefits and drawbacks?

For Faculty Use

Correction Parameters	Formative Assessment [40%]	Timely completion of Practical [40%]	Attendance / Learning Attitude [20%]	Total
Marks Obtained				