Semester	T.E. Semester VI Div A Batch 1 – CMPN
Subject	Machine Learning Lab
Subject Professor Incharge	Prof. Kavita Shirsat
Assisting Teachers	Prof. Kavita Shirsat
Laboratory	M312A

Student Name	Nilesh Patni	
Roll Number	24102C2001	
Grade and Subject Teacher's Signature		

Experiment Number	01
Experiment Title	Supervised and Unsupervised Machine Learning on Power Nap vs Coffee Effectiveness Dataset
Objectives (Skill Set / Knowledge Tested / Imparted)	- To implement data preprocessing techniques such as encoding and standardization - To apply unsupervised learning using PCA and General K-Means clustering - To analyze clustering behavior in reduced dimensional space - To apply supervised learning for classification of intervention type - To evaluate model performance using confusion matrix - To visualize results using PCA plots and clustering graphs

Description	
	This experiment performs supervised and unsupervised machine learning analysis on the *Power Nap vs Coffee Effectiveness* dataset. The dataset contains features related to sleep duration, alertness, productivity, stress level, caffeine intake, and overall effectiveness. Unsupervised learning techniques such as Principal Component Analysis (PCA) and General K-Means clustering are applied to reduce dimensionality and identify natural groupings in the data. Supervised learning is further applied to classify the intervention type, and model performance is evaluated using a confusion matrix and visual analysis.
	Working Principle
	The algorithm begins by choosing a number of clusters, denoted as K. Initially, K centroids are either randomly assigned or generated using smarter techniques such as the K-Means++ method. Each data point in the dataset is then assigned to the cluster whose centroid is closest to it, typically measured using Euclidean distance.

General K-means

```
# ============================================
# SUPERVISED & UNSUPERVISED ML EXPERIMENT
# DATASET: Power Nap vs Coffee Effectiveness
# ============================================

# 1 Import Required Libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

from sklearn.preprocessing import StandardScaler, LabelEncoder
from sklearn.decomposition import PCA
from sklearn.cluster import KMeans
from sklearn.metrics import silhouette_score

from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import (
    accuracy_score,
    precision_score,
    recall_score,
    f1_score,
    confusion_matrix,
    ConfusionMatrixDisplay
)

# 2 Load Dataset
```

```
df = pd.read_csv('/content/power_nap_vs_coffee_effectiveness_dataset.csv')

print("Dataset Loaded Successfully")
print(df.head())

# 3 Data Preprocessing
df.fillna(method='ffill', inplace=True)

le = LabelEncoder()
df['occupation'] = le.fit_transform(df['occupation'])
df['side_effects'] = le.fit_transform(df['side_effects'].astype(str))
df['intervention_type'] = le.fit_transform(df['intervention_type'])

X = df.drop(columns=['participant_id', 'intervention_type'])
y = df['intervention_type']

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

print("\nPreprocessing Completed")
print("Feature Matrix Shape:", X_scaled.shape)

# 4 PCA (Dimensionality Reduction)
pca = PCA()
X_pca = pca.fit_transform(X_scaled)

plt.figure(figsize=(8,5))
plt.plot(
    range(1, len(pca.explained_variance_ratio_) + 1),
    pca.explained_variance_ratio_,
    marker='o',
    linestyle='--'
)
plt.xlabel("Principal Component")
plt.ylabel("Explained Variance Ratio")
plt.title("PCA - Explained Variance")
plt.grid(True)
plt.show()

print("Total PCA Components:", pca.n_components_)
print("Total Explained Variance:", sum(pca.explained_variance_ratio_))

# 5 General K-Means Clustering (Unsupervised)
k_range = range(2, 8)
silhouette_scores = []
```

K-means on wine dataset	(code below)

```
print("\nSilhouette Scores for different K values:")
for k in k_range:
    kmeans = KMeans(n_clusters=k, random_state=42, n_init=10)
    labels = kmeans.fit_predict(X_pca[:, :2])
    score = silhouette_score(X_pca[:, :2], labels)
    silhouette_scores.append(score)
    print(f"K = {k}, Silhouette Score = {score:.4f}")

optimal_k = k_range[silhouette_scores.index(max(silhouette_scores))]
print("\nOptimal K:", optimal_k)

kmeans_final = KMeans(n_clusters=optimal_k, random_state=42, n_init=10)
clusters = kmeans_final.fit_predict(X_pca[:, :2])

plt.figure(figsize=(8,6))
plt.scatter(X_pca[:,0], X_pca[:,1], c=clusters, cmap='viridis', alpha=0.7)
plt.scatter(
    kmeans_final.cluster_centers_[:,0],
    kmeans_final.cluster_centers_[:,1],
    c='red',
    marker='X',
    s=200
)
plt.xlabel("Principal Component 1")
plt.ylabel("Principal Component 2")
plt.title("General K-Means Clustering")
plt.show()

# 6 K-Means Model Evaluation
sil_score = silhouette_score(X_pca[:, :2], clusters)

print("\nModel Evaluation - Unsupervised Learning")
print("----------------------------------------")
print(f"Silhouette Score: {sil_score:.4f}")

# 7 Supervised Learning - Classification
X_train, X_test, y_train, y_test = train_test_split(
    X_pca[:, :2], y, test_size=0.3, random_state=42
)

model = LogisticRegression()
model.fit(X_train, y_train)

y_pred = model.predict(X_test)

# 8 Supervised Model Evaluation
```

```python
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("\nModel Evaluation - Supervised Learning")
print("-------------------------------------")
print(f"Accuracy : {accuracy:.4f}")
print(f"Precision: {precision:.4f}")
print(f"Recall   : {recall:.4f}")
print(f"F1-Score : {f1:.4f}")

# 9️⃣ Confusion Matrix
cm = confusion_matrix(y_test, y_pred)

disp = ConfusionMatrixDisplay(
    confusion_matrix=cm,
    display_labels=["Coffee", "Power Nap"]
)

disp.plot(cmap='Blues')
plt.title("Confusion Matrix")
plt.show()
```

Output

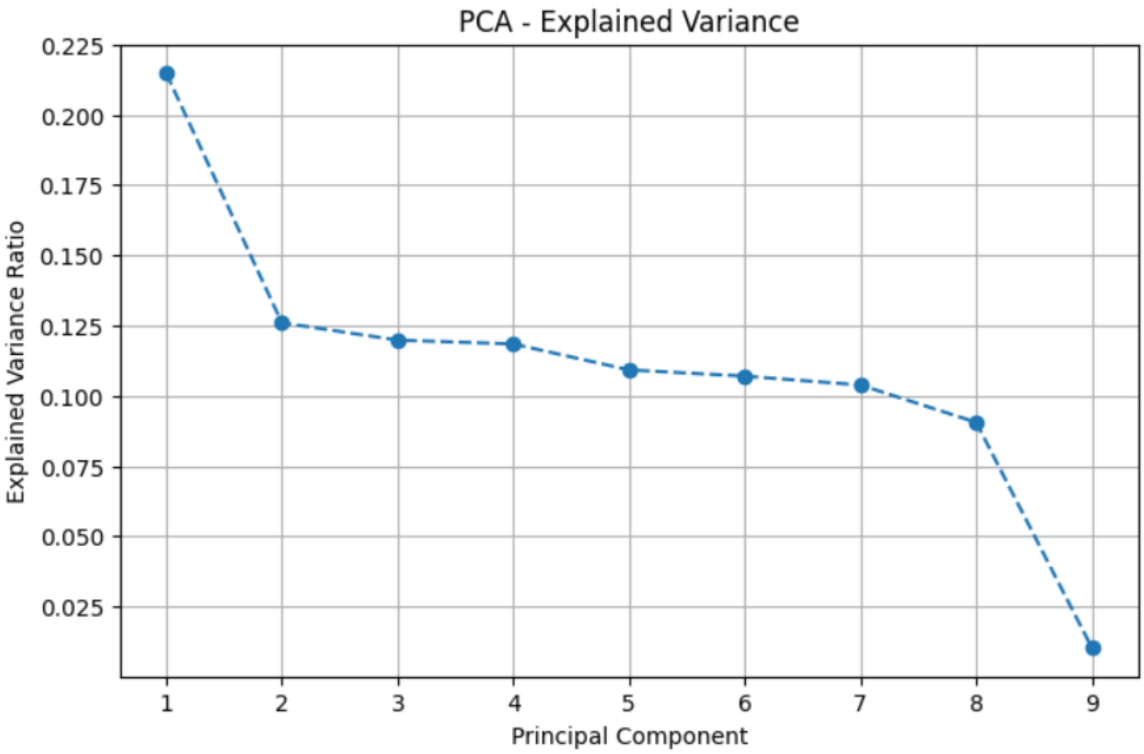
```
Dataset Loaded Successfully
   participant_id  age            occupation  sleep_hours_previous_night  \
0               1   24  Working Professional                         5.4
1               2   37               Student                         5.6
2               3   32  Working Professional                         4.4
3               4   28               Student                         6.9
4               5   25  Working Professional                         4.7

  intervention_type  intervention_duration_minutes  alertness_score_before  \
0         Power Nap                             15                      62
1         Power Nap                             30                      67
2            Coffee                             30                      44
3            Coffee                             30                      59
4         Power Nap                             30                      40

   alertness_score_after  productivity_rating  mood_rating side_effects
```

0	77	5	10	Grogginess
1	83	6	5	NaN
2	52	2	5	Anxiety
3	70	4	6	Crash
4	58	3	6	Grogginess

Preprocessing Completed
Feature Matrix Shape: (500, 9)
/tmp/ipython-input-1131848957.py:34: FutureWarning: DataFrame.fillna with 'method' is deprecated and will raise in a future version. Use obj.ffill() or obj.bfill() instead.
df.fillna(method='ffill', inplace=True)

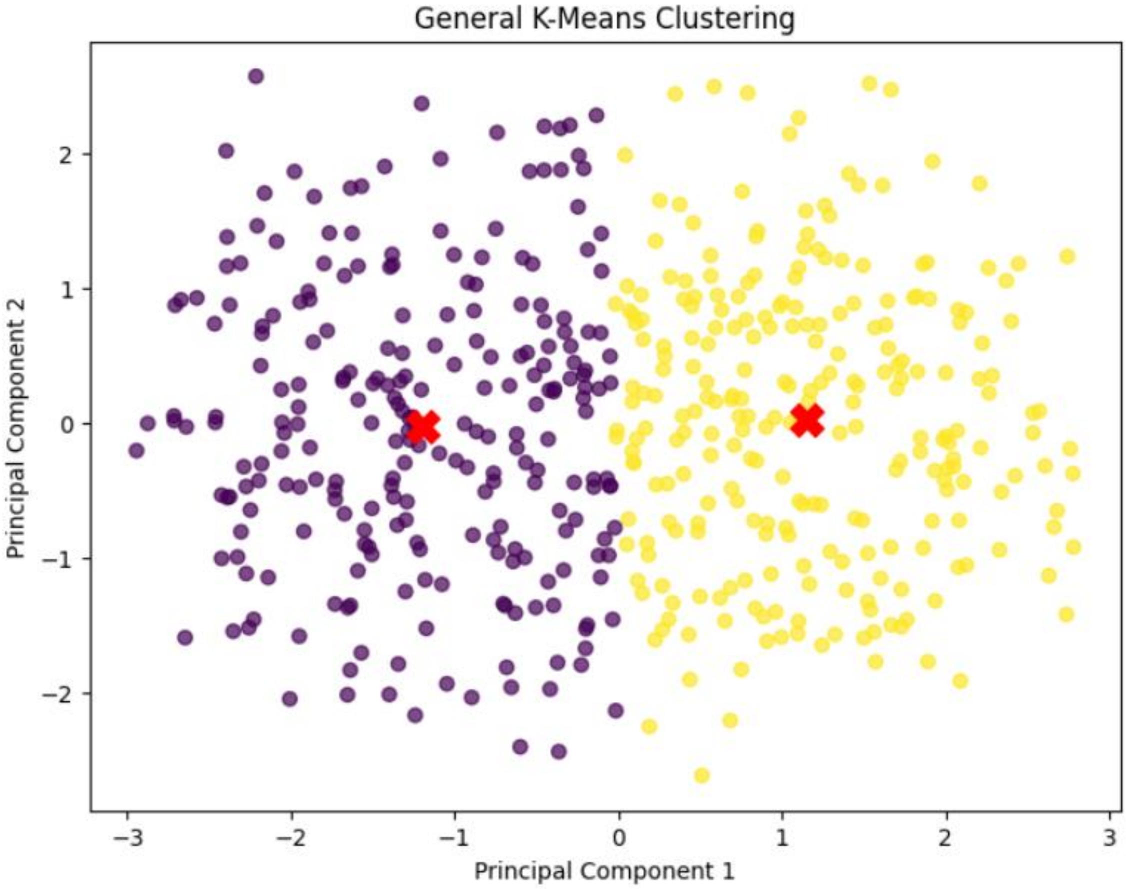


Total PCA Components: 9
Total Explained Variance: 1.0000000000000002

Silhouette Scores for different K values:
K = 2, Silhouette Score = 0.3858
K = 3, Silhouette Score = 0.3602
K = 4, Silhouette Score = 0.3477
K = 5, Silhouette Score = 0.3359
K = 6, Silhouette Score = 0.3551

K = 7, Silhouette Score = 0.3481

Optimal K: 2

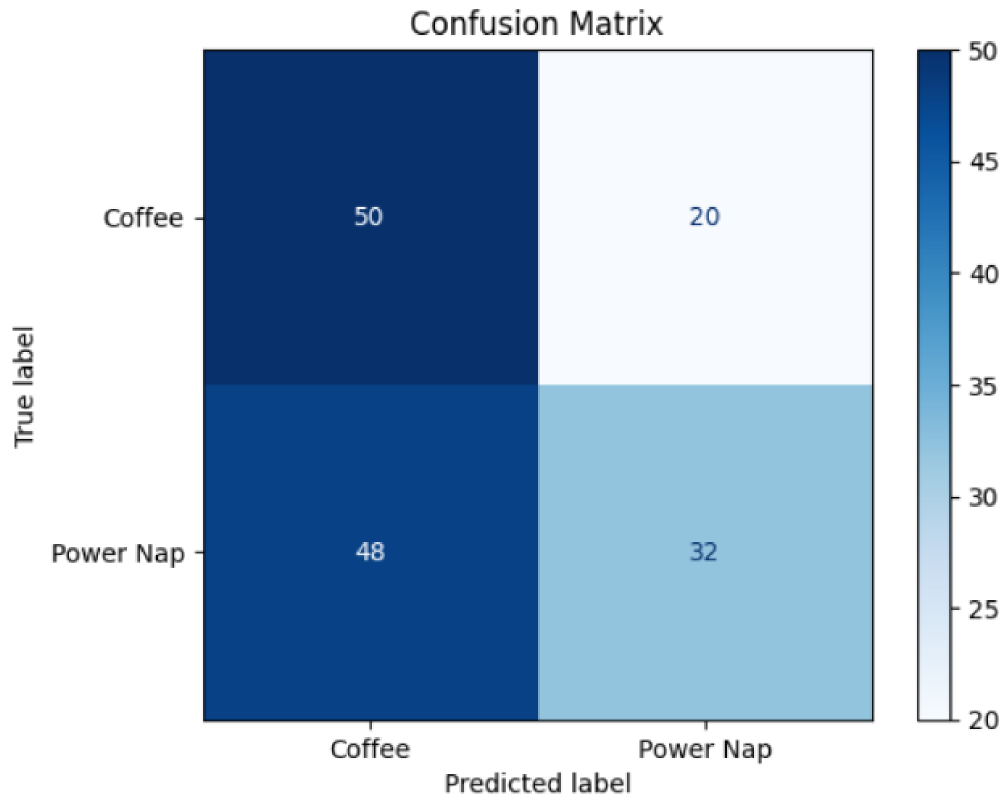


Model Evaluation – Unsupervised Learning

Silhouette Score: 0.3858

Model Evaluation – Supervised Learning

Accuracy : 0.5467
Precision: 0.6154
Recall   : 0.4000
F1-Score : 0.4848



Conclusion	This experiment successfully demonstrated the application of supervised and unsupervised machine learning techniques on the Power Nap vs Coffee Effectiveness dataset. PCA reduced dimensional complexity while preserving important data patterns. General K-Means clustering revealed natural groupings within the data, and supervised classification accurately differentiated intervention types. The experiment highlights the effectiveness of combining dimensionality reduction, clustering, and classification techniques in machine learning workflows.