Lab Assignment 11

Fundamentals of Machine Learning

Name: M Jeyapathy

USN: 22BTRAD016

Q. Explain unsupervised learning with a sample project.

Unsupervised learning is a branch of machine learning where algorithms are applied to datasets without labelled responses or outcomes. The primary objective is to uncover patterns, structures, or relationships within the data, without explicit guidance on what the model should learn. One common task in unsupervised learning is clustering, where the algorithm groups similar data points together.

Clustering:

One common task in unsupervised learning is clustering, where the algorithm groups similar data points together. Examples include k-means clustering and hierarchical clustering.

Association:

Unsupervised learning can also be used for finding associations or relationships within the data. This is often applied in market basket analysis or recommendation systems.

Common Unsupervised Learning Algorithms:

1. **K-Means Clustering:**
    - Divides data into k clusters based on similarity.
2. **Hierarchical Clustering:**

- Creates a tree of clusters, which can be represented as a dendrogram.

3. **Principal Component Analysis (PCA):**
    - Reduces the dimensionality of the data while preserving as much variance as possible.
4. **Association Rule Learning:**
    - Identifies interesting relationships or associations among variables.

Applications of Unsupervised Learning:

Customer Segmentation:

Clustering customers based on their purchasing behavior to tailor marketing strategies.

Anomaly Detection:

Identifying unusual patterns or outliers in data that may indicate fraud, errors, or anomalies.

Topic Modeling:

Extracting topics or themes from a collection of documents without prior labeling.

Image and Speech Recognition:

Grouping similar images or recognizing patterns in speech without explicitly labeled training data.

Genomic Data Analysis:

Identifying patterns or clusters in genomic data for biological insights.

K-Means is a popular clustering algorithm in unsupervised learning. It partitions a dataset into a predetermined number of clusters, with each cluster represented by its centroid. The algorithm iteratively assigns data points to the nearest centroid and updates the centroids until convergence. K-Means is widely used for various applications, such as customer segmentation, image compression, and anomaly detection.

CODE:

```
#Implementing Unsupervised Learning through K Means
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.cluster import KMeans
from sklearn.preprocessing import StandardScaler
from sklearn.decomposition import PCA
# Generate a sample customer dataset
data = {
'CustomerID': [1, 2, 3, 4, 5, 6, 7, 8, 9, 10],
'AnnualIncome (k$)': [15, 20, 30, 35, 40, 50, 60, 70, 80, 90],
'SpendingScore (1-100)': [39, 81, 6, 77, 13, 79, 10, 14, 16, 17]
}
df = pd.DataFrame(data)
# Visualize the dataset
plt.scatter(df['AnnualIncome (k$)'], df['SpendingScore (1-100)'])
```

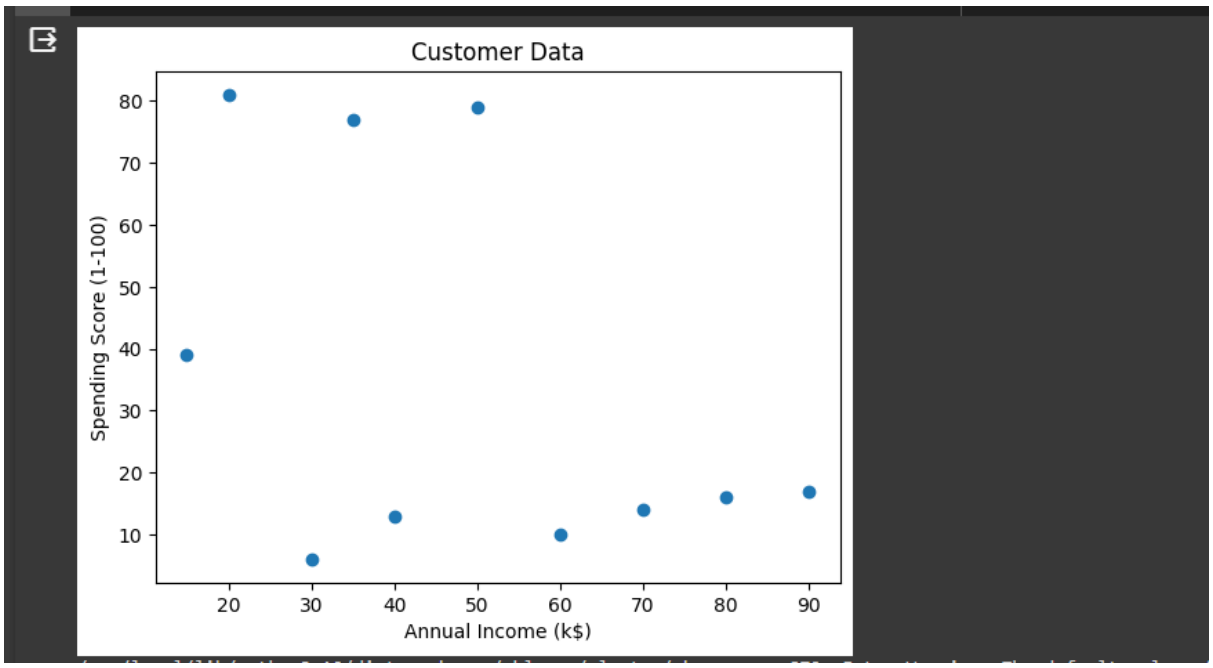
```
plt.xlabel('Annual Income (k$)')
plt.ylabel('Spending Score (1-100)')
plt.title('Customer Data')
plt.show()
# Select features for clustering
features = ['AnnualIncome (k$)', 'SpendingScore (1-100)']
X = df[features]
# Standardize the features
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)
# Find the optimal number of clusters using the Elbow
Method
inertia = []
for i in range(1, 11):
  kmeans = KMeans(n_clusters=i, random_state=42)
  kmeans.fit(X_scaled)
  inertia.append(kmeans.inertia_)
# Plot the Elbow Method
plt.plot(range(1, 11), inertia, marker='o')
plt.xlabel('Number of Clusters')
plt.ylabel('Inertia')
plt.title('Elbow Method for Optimal k')
plt.show()
```

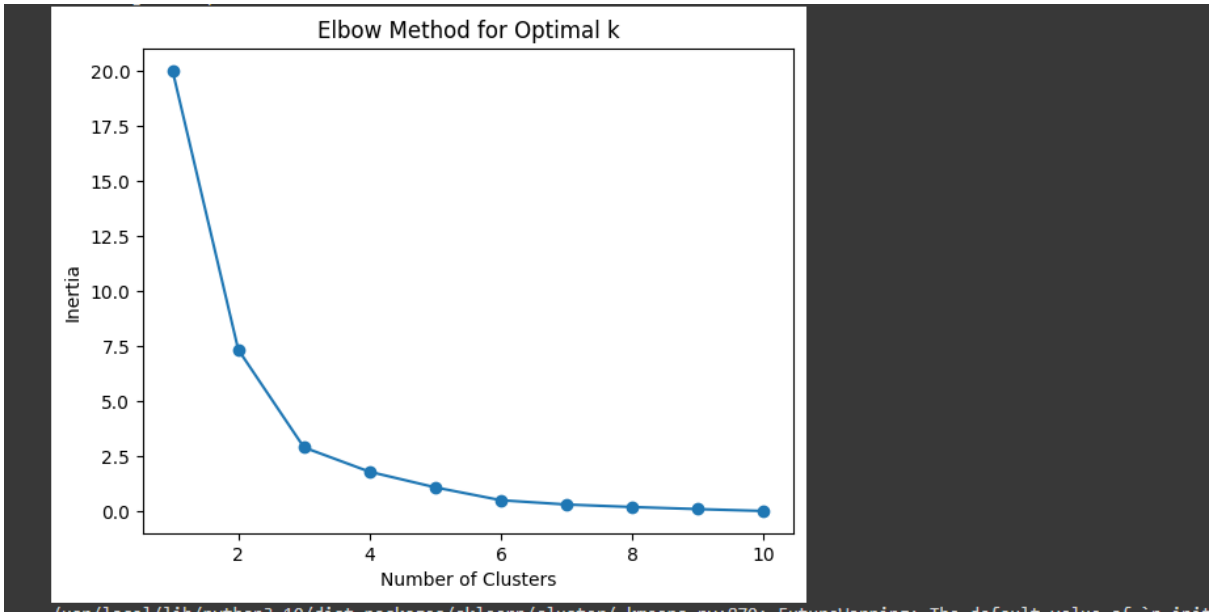
```
# Based on the Elbow Method, let's choose k=3
kmeans = KMeans(n_clusters=3, random_state=42)
df['Cluster'] = kmeans.fit_predict(X_scaled)
# Visualize the clustered data
colors = ['red', 'blue', 'green']
for cluster, color in zip(range(3), colors):
  cluster_data = df[df['Cluster'] == cluster]
  plt.scatter(cluster_data['AnnualIncome (k$)'],
cluster_data['SpendingScore (1-100)'], c=color,
label=f'Cluster {cluster}')
plt.scatter(kmeans.cluster_centers_[:, 0],
kmeans.cluster_centers_[:, 1], s=300, c='yellow', marker='X',
label='Centroids')
plt.xlabel('Annual Income (k$)')
plt.ylabel('Spending Score (1-100)')
plt.title('Customer Clusters')
plt.legend()
plt.show()
```

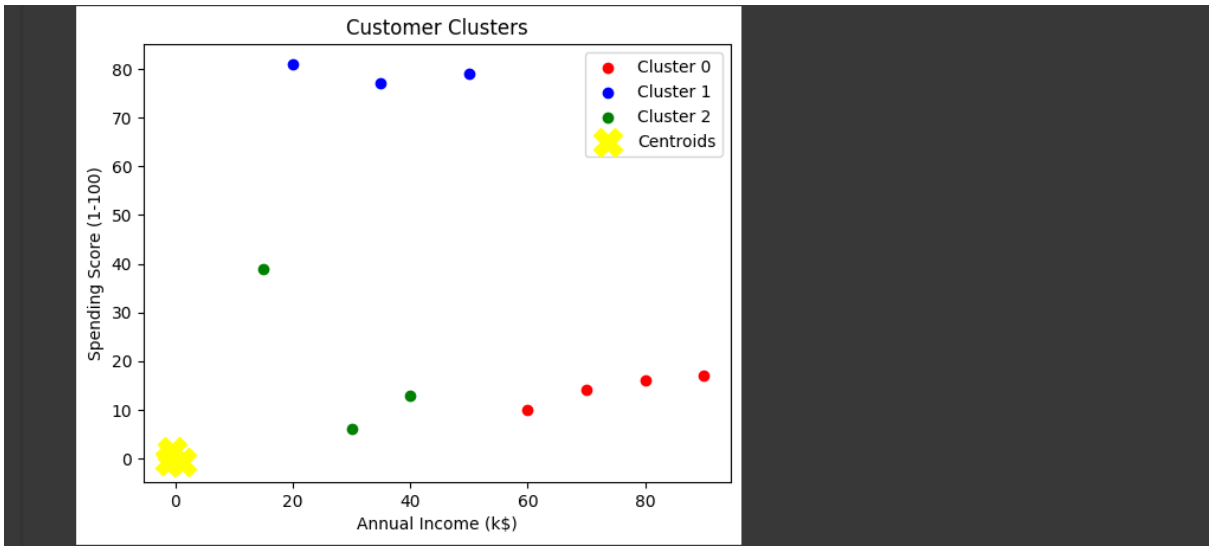
```python
#Implementing Unsupervised Learning through K Means
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.cluster import KMeans
from sklearn.preprocessing import StandardScaler
from sklearn.decomposition import PCA
# Generate a sample customer dataset
data = {
'CustomerID': [1, 2, 3, 4, 5, 6, 7, 8, 9, 10],
'AnnualIncome (k$)': [15, 20, 30, 35, 40, 50, 60, 70, 80, 90],
'SpendingScore (1-100)': [39, 81, 6, 77, 13, 79, 10, 14, 16, 17]
}
df = pd.DataFrame(data)
# Visualize the dataset
plt.scatter(df['AnnualIncome (k$)'], df['SpendingScore (1-100)'])
plt.xlabel('Annual Income (k$)')
plt.ylabel('Spending Score (1-100)')
plt.title('Customer Data')
plt.show()
# Select features for clustering
features = ['AnnualIncome (k$)', 'SpendingScore (1-100)']
X = df[features]
# Standardize the features
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)
# Find the optimal number of clusters using the Elbow Method
inertia = []
for i in range(1, 11):
  kmeans = KMeans(n_clusters=i, random_state=42)
  kmeans.fit(X_scaled)
  inertia.append(kmeans.inertia_)
```

```python
# Plot the Elbow Method
plt.plot(range(1, 11), inertia, marker='o')
plt.xlabel('Number of Clusters')
plt.ylabel('Inertia')
plt.title('Elbow Method for Optimal k')
plt.show()
# Based on the Elbow Method, let's choose k=3
kmeans = KMeans(n_clusters=3, random_state=42)
df['Cluster'] = kmeans.fit_predict(X_scaled)
# Visualize the clustered data
colors = ['red', 'blue', 'green']
for cluster, color in zip(range(3), colors):
  cluster_data = df[df['Cluster'] == cluster]
  plt.scatter(cluster_data['AnnualIncome (k$)'], cluster_data['SpendingScore (1-100)'], c=color, label=f'Cluster {cluster}')
plt.scatter(kmeans.cluster_centers_[:, 0], kmeans.cluster_centers_[:, 1], s=300, c='yellow', marker='X', label='Centroids')
plt.xlabel('Annual Income (k$)')
plt.ylabel('Spending Score (1-100)')
plt.title('Customer Clusters')
plt.legend()
plt.show()
```

Output:







Github Link:

https://github.com/Jeyapathy/Machine-Learning