Machine Learning Lab (AI332L)

Class:	BS Artificial Intelligence	**Semester:**	5th (Fall 2023)
Instructor:	Dr. M. Ehatisham-ul-Haq	**Email:**	ehtisham@mail.au.edu.pk

Unsupervised Learning Algorithms in Python

Lab Objective:

In this lab tutorial, we will apply different unsupervised learning algorithms (s) to the datasets using Python. This tutorial will guide you through the entire process for unsupervised classification tasks.

Building Unsupervised Machine Learning Classification Model

Importing Libraries:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.cluster import KMeans
from sklearn.decomposition import PCA
from sklearn.datasets import load_iris
```

Load a Sample Dataset: Iris Dataset

```
# Load Iris dataset
iris = load_iris()
X_iris = iris.data
y_iris = iris.target
```

Applying K-Means Clustering for Classification:

```
from sklearn.cluster import KMeans

# Create a K-means clustering model with 3 clusters (as there are 3 species in
the Iris dataset)
kmeans = KMeans(n_clusters=3, random_state=42)
kmeans.fit(X)
# Get the cluster centers and labels
cluster_centers = kmeans.cluster_centers_
labels = kmeans.labels_


# Visualize the clusters
plt.figure(figsize=(8, 6))
plt.scatter(X[:, 0], X[:, 1], c=labels, s=50, cmap='viridis')
plt.scatter(cluster_centers[:, 0], cluster_centers[:, 1], s=200, c='red',
marker='X')
plt.title("K-means Clustering")
```

```
plt.xlabel(iris.feature_names[0])
plt.ylabel(iris.feature_names[1])
plt.show()


# Evaluating results
inertia = kmeans.inertia_
print(f"Inertia: {inertia}")
```

Applying Hierarchical Clustering for Classification:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.datasets import load_iris
from sklearn.cluster import AgglomerativeClustering
from scipy.cluster.hierarchy import dendrogram, linkage

# Load Iris dataset
iris = load_iris()
X_iris = iris.data
y_iris = iris.target

# Apply Agglomerative Hierarchical Clustering
agg_clustering = AgglomerativeClustering(n_clusters=3)
labels_agg = agg_clustering.fit_predict(X_iris)

# Visualize the clusters
plt.scatter(X_iris[:, 0], X_iris[:, 1], c=labels_agg, s=50, cmap='viridis')
plt.title("Agglomerative Hierarchical Clustering (Iris Dataset)")
plt.show()

# Create a dendrogram for hierarchical clustering
linkage_matrix = linkage(X_iris, method='ward')
dendrogram(linkage_matrix,   truncate_mode='level',   p=3,   leaf_rotation=90.,
leaf_font_size=8., show_contracted=True)
plt.title("Hierarchical Clustering Dendrogram (Iris Dataset)")
plt.show()
```

Lab Task:

1. **Explore K-means++ and K-Mediods clustering techniques. Choose a publically available dataset of your own choice and apply K-means++, K-Mediods, K-Means clustering, and Hierarchical Clustering, and compare the clustering performance.**
2. **Use unsupervised learning for anomaly detection in a dataset of your own choice. Select a dataset with clear anomalies or outliers.**
3. **Use clustering as a feature engineering technique. Choose a dataset where clustering might reveal hidden patterns. Apply a clustering algorithm and use the cluster labels as new features.**
   Train a supervised model with and without the cluster labels. Compare the performance of the models and discuss the impact of clustering on feature engineering.
