Assignment 6

Set A

1) **Write a python program to implement k-means algorithm to build prediction model (Use Credit Card Dataset CC GENERAL.csv Download from kaggle.com)**

```
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.cluster import KMeans
from sklearn.preprocessing import StandardScaler

# Step 1: Load the dataset
df = pd.read_csv("CC GENERAL.csv")

# Step 2: Preprocess the data
# Drop the 'CUST_ID' column as it's irrelevant for clustering
df = df.drop(columns=['CUST_ID'], axis=1)

# Fill missing values with the column mean
df.fillna(df.mean(), inplace=True)

# Standardize the features for better clustering performance
scaler = StandardScaler()
data_scaled = scaler.fit_transform(df)

# Step 3: Apply K-Means Clustering
kmeans = KMeans(n_clusters=3, random_state=42)  # Choose 3 clusters arbitrarily
df['Cluster'] = kmeans.fit_predict(data_scaled)

# Step 4: Display the results
print("Cluster assignments for the data:")
print(df['Cluster'].value_counts())

# Show first 5 rows with cluster labels
print("\nSample data with cluster assignments:")
print(df.head())

# Step 5: Visualize the Clusters
```

```
plt.figure(figsize=(8, 6))

# Select two features for plotting (e.g., 'BALANCE' vs 'PURCHASES')
plt.scatter(df['BALANCE'], df['PURCHASES'], c=df['Cluster'], cmap='viridis', alpha=0.5)

plt.xlabel("Balance")
plt.ylabel("Purchases")
plt.title("K-Means Clustering (3 Clusters)")
plt.colorbar(label="Cluster")
plt.show()
```

2) **Write a python program to implement hierarchical Agglomerative clustering algorithm. (Download Customer.csv dataset from github.com).**

```
import pandas as pd
from scipy.cluster.hierarchy import dendrogram, linkage
import matplotlib.pyplot as plt

# Step 1: Load the dataset
df = pd.read_csv("Mall_Customers.csv")

# Step 2: Select features for clustering
# Assuming 'Annual Income (k$)' and 'Spending Score (1-100)' are relevant features
X = df[['Annual Income (k$)', 'Spending Score (1-100)']]

# Step 3: Plot the dendrogram
plt.figure(figsize=(10, 7))
linkage_matrix = linkage(X, method='ward')  # Create linkage matrix
dendrogram(linkage_matrix)
plt.title("Dendrogram")
plt.xlabel("Customers")
plt.ylabel("Distance")
plt.show()
```

Set B

1) **Write a python program to implement k-means algorithms on a synthetic (Artificial Generated Data) dataset.**

```
import numpy as np
import matplotlib.pyplot as plt
from sklearn.datasets import make_blobs
from sklearn.cluster import KMeans

# Step 1: Generate synthetic data with make_blobs
X, _ = make_blobs(n_samples=300, centers=4, random_state=42)

# Step 2: Apply K-Means clustering algorithm
kmeans = KMeans(n_clusters=4)
kmeans.fit(X)

# Step 3: Plot the results
plt.scatter(X[:, 0], X[:, 1], c=kmeans.labels_, cmap='viridis')
plt.scatter(kmeans.cluster_centers_[:, 0], kmeans.cluster_centers_[:, 1], s=200, c='red',
marker='X')
plt.title("K-Means Clustering")
plt.show()

# Step 4: Print the cluster centers (centroids)
print('Cluster Centers (Centroids):')
print(kmeans.cluster_centers_)

# Step 5: Print the number of data points in each cluster
print("\nData points in each cluster:")
for i in range(4):  # We know there are 4 clusters
   print(f"Cluster {i}: {np.sum(kmeans.labels_ == i)} points")
```

2) Write a python program to implement hierarchical clustering algorithm. (Download Wholesale customers data dataset from github.com)

```
import pandas as pd

import matplotlib.pyplot as plt

from scipy.cluster.hierarchy import dendrogram, linkage


data = pd.read_csv('wholesale.csv')


# Use only numerical data (skip 'Channel' and 'Region')

features = data.iloc[:, 2:]  # Assuming 'Channel' and 'Region' are the first two columns


# Step 2: Perform Hierarchical Clustering

# Compute linkage matrix for dendrogram

linkage_matrix = linkage(features, method='ward')


# Plot the dendrogram

plt.figure(figsize=(10, 6))
```

```
plt.title("Dendrogram")
plt.xlabel("Data points")
plt.ylabel("Euclidean distances")
dendrogram(linkage_matrix)
plt.show()
```