Lab Assignment-5

import pandas as pd

from sklearn.model\_selection import train\_test\_split

from sklearn.linear\_model import LinearRegression

from sklearn.metrics import mean\_squared\_error, r2\_score

from sklearn.cluster import KMeans

from sklearn.metrics import silhouette\_score, calinski\_harabasz\_score, davies\_bouldin\_score

import matplotlib.pyplot as plt

# Load the dataset

df = pd.read\_csv("/mnt/data/diabetes.csv")

### A1: Linear Regression with One Feature ###

X = df[['Glucose']]

y = df['Outcome']

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

reg = LinearRegression().fit(X\_train, y\_train)

y\_train\_pred = reg.predict(X\_train)

y\_test\_pred = reg.predict(X\_test)

### A2: Compute and Print Metrics for One Feature ###

def compute\_metrics(y\_true, y\_pred):

mse = mean\_squared\_error(y\_true, y\_pred)

rmse = mse \*\* 0.5

r2 = r2\_score(y\_true, y\_pred)

return mse, rmse, r2

train\_metrics = compute\_metrics(y\_train, y\_train\_pred)

test\_metrics = compute\_metrics(y\_test, y\_test\_pred)

print("A2 - Metrics for One Feature:")

print(f"Train - MSE: {train\_metrics[0]}, RMSE: {train\_metrics[1]}, R²: {train\_metrics[2]}")

print(f"Test - MSE: {test\_metrics[0]}, RMSE: {test\_metrics[1]}, R²: {test\_metrics[2]}")

### A3: Linear Regression with Multiple Features ###

X\_multi = df.drop(columns=['Outcome'])

X\_train\_m, X\_test\_m, y\_train\_m, y\_test\_m = train\_test\_split(X\_multi, y, test\_size=0.2, random\_state=42)

reg\_multi = LinearRegression().fit(X\_train\_m, y\_train\_m)

y\_train\_pred\_m = reg\_multi.predict(X\_train\_m)

y\_test\_pred\_m = reg\_multi.predict(X\_test\_m)

train\_metrics\_m = compute\_metrics(y\_train\_m, y\_train\_pred\_m)

test\_metrics\_m = compute\_metrics(y\_test\_m, y\_test\_pred\_m)

print("A3 - Metrics for Multiple Features:")

print(f"Train - MSE: {train\_metrics\_m[0]}, RMSE: {train\_metrics\_m[1]}, R²: {train\_metrics\_m[2]}")

print(f"Test - MSE: {test\_metrics\_m[0]}, RMSE: {test\_metrics\_m[1]}, R²: {test\_metrics\_m[2]}")

### A4: K-Means Clustering ###

X\_cluster = df.drop(columns=['Outcome'])

kmeans = KMeans(n\_clusters=2, random\_state=0, n\_init="auto").fit(X\_cluster)

labels = kmeans.labels\_

### A5: Compute Clustering Metrics ###

sil\_score = silhouette\_score(X\_cluster, labels)

ch\_score = calinski\_harabasz\_score(X\_cluster, labels)

db\_score = davies\_bouldin\_score(X\_cluster, labels)

print("A5 - Clustering Metrics:")

print(f"Silhouette Score: {sil\_score}")

print(f"Calinski-Harabasz Score: {ch\_score}")

print(f"Davies-Bouldin Score: {db\_score}")

### A6: K-Means with Different k Values ###

k\_values = range(2, 10)

sil\_scores, ch\_scores, db\_scores = [], [], []

for k in k\_values:

kmeans = KMeans(n\_clusters=k, random\_state=42).fit(X\_cluster)

sil\_scores.append(silhouette\_score(X\_cluster, kmeans.labels\_))

ch\_scores.append(calinski\_harabasz\_score(X\_cluster, kmeans.labels\_))

db\_scores.append(davies\_bouldin\_score(X\_cluster, kmeans.labels\_))

plt.figure(figsize=(10, 5))

plt.plot(k\_values, sil\_scores, label='Silhouette Score')

plt.plot(k\_values, ch\_scores, label='Calinski-Harabasz Score')

plt.plot(k\_values, db\_scores, label='Davies-Bouldin Index')

plt.xlabel('Number of Clusters (k)')

plt.ylabel('Score')

plt.legend()

plt.title('Clustering Evaluation Metrics')

plt.show()

### A7: Elbow Method for Optimal k ###

distortions = []

k\_range = range(2, 20)

for k in k\_range:

kmeans = KMeans(n\_clusters=k).fit(X\_cluster)

distortions.append(kmeans.inertia\_)

plt.figure(figsize=(10, 5))

plt.plot(k\_range, distortions, marker='o')

plt.xlabel('Number of Clusters (k)')

plt.ylabel('Inertia')

plt.title('Elbow Method for Optimal k')

plt.show()

Output:

A2 - Metrics for One Feature:

Train - MSE: 0.17942747567178632, RMSE: 0.42358880494152146, R²: 0.20804279654899438

Test - MSE: 0.17113033279525355, RMSE: 0.4136790214589731, R²: 0.25463232826956206

A3 - Metrics for Multiple Features:

Train - MSE: 0.15744485172625472, RMSE: 0.3967932102824527, R²: 0.30506972801106247

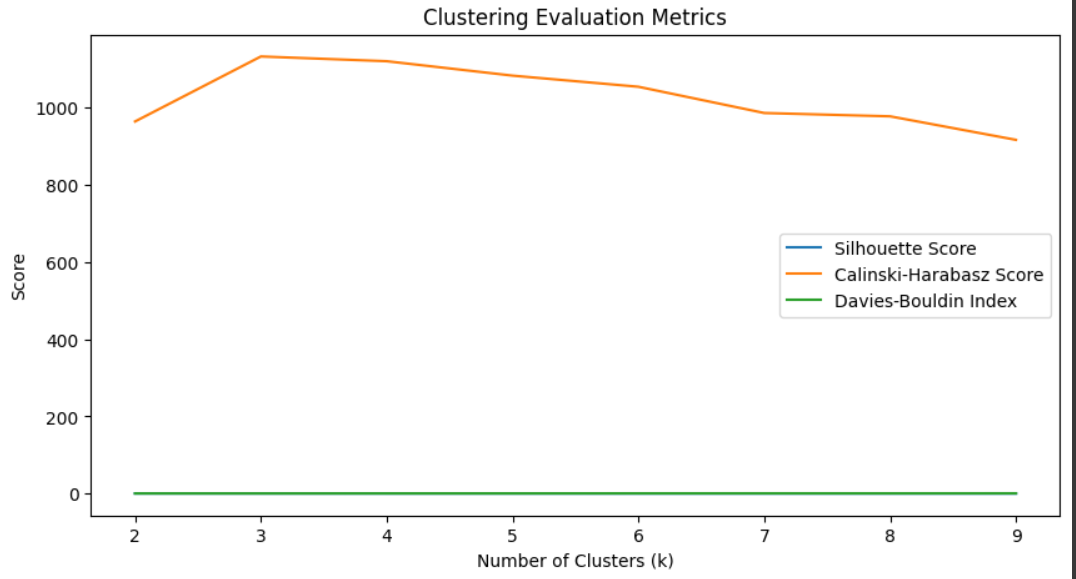
Test - MSE: 0.17104527280850104, RMSE: 0.4135761995189049, R²: 0.25500281176741757

A5 - Clustering Metrics:

Silhouette Score: 0.5687897205830247

Calinski-Harabasz Score: 964.2725250859544

Davies-Bouldin Score: 0.7133822795826191

A graph with a line

AI-generated content may be incorrect.