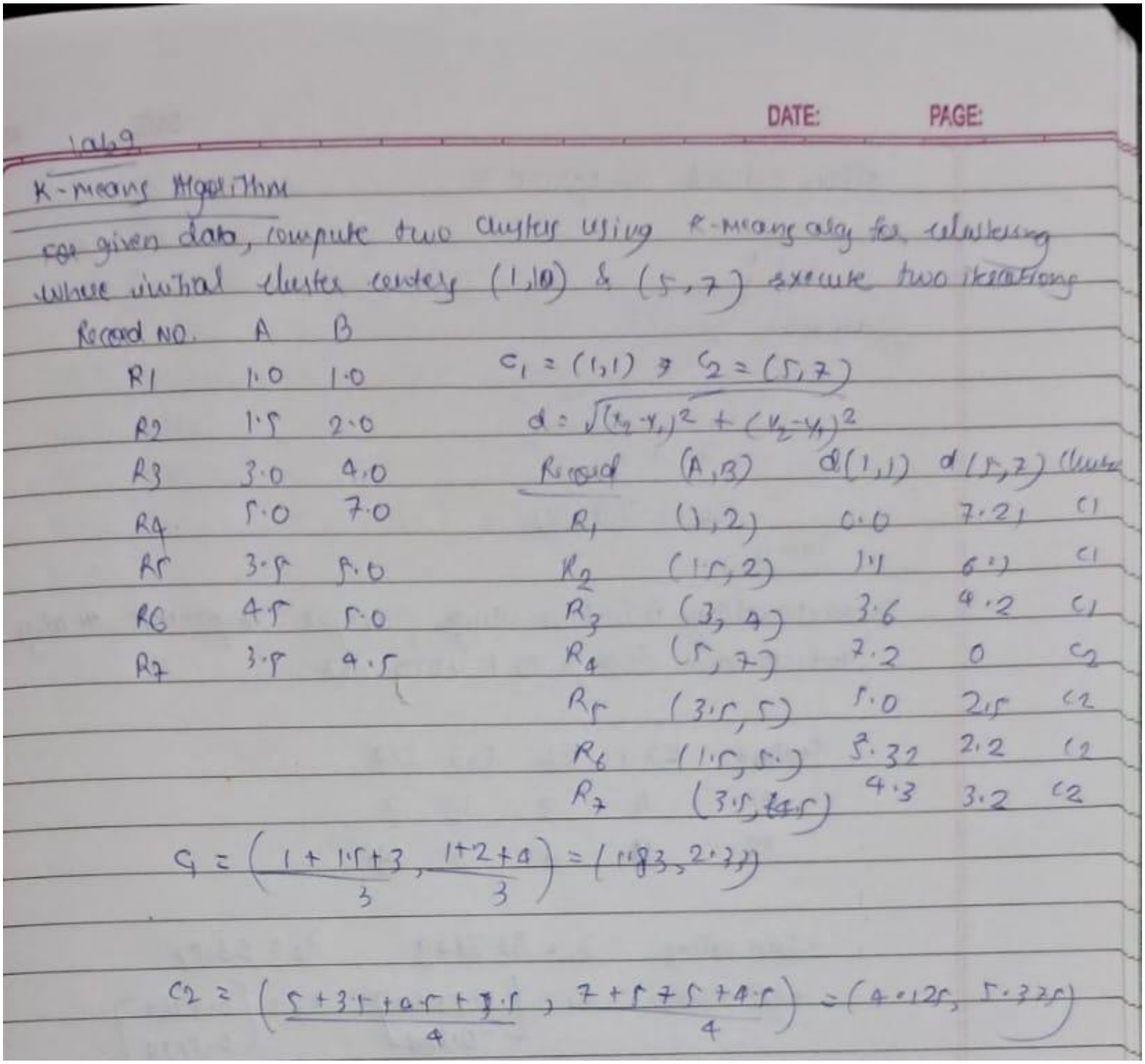
Program 10

Build k-Means algorithm to cluster a set of data stored in a .CSV file

Screenshot:



DATE: PAGE:

Lab 9

K-means Algorithm

for given data, compute two clusters using K-means alg for clustering where initial cluster centers (1,10) & (5,7) execute two iterations

Record NO.	A	B
R1	1.0	1.0
R2	1.5	2.0
R3	3.0	4.0
R4	5.0	7.0
R5	3.5	5.0
R6	4.5	5.0
R7	3.5	4.5

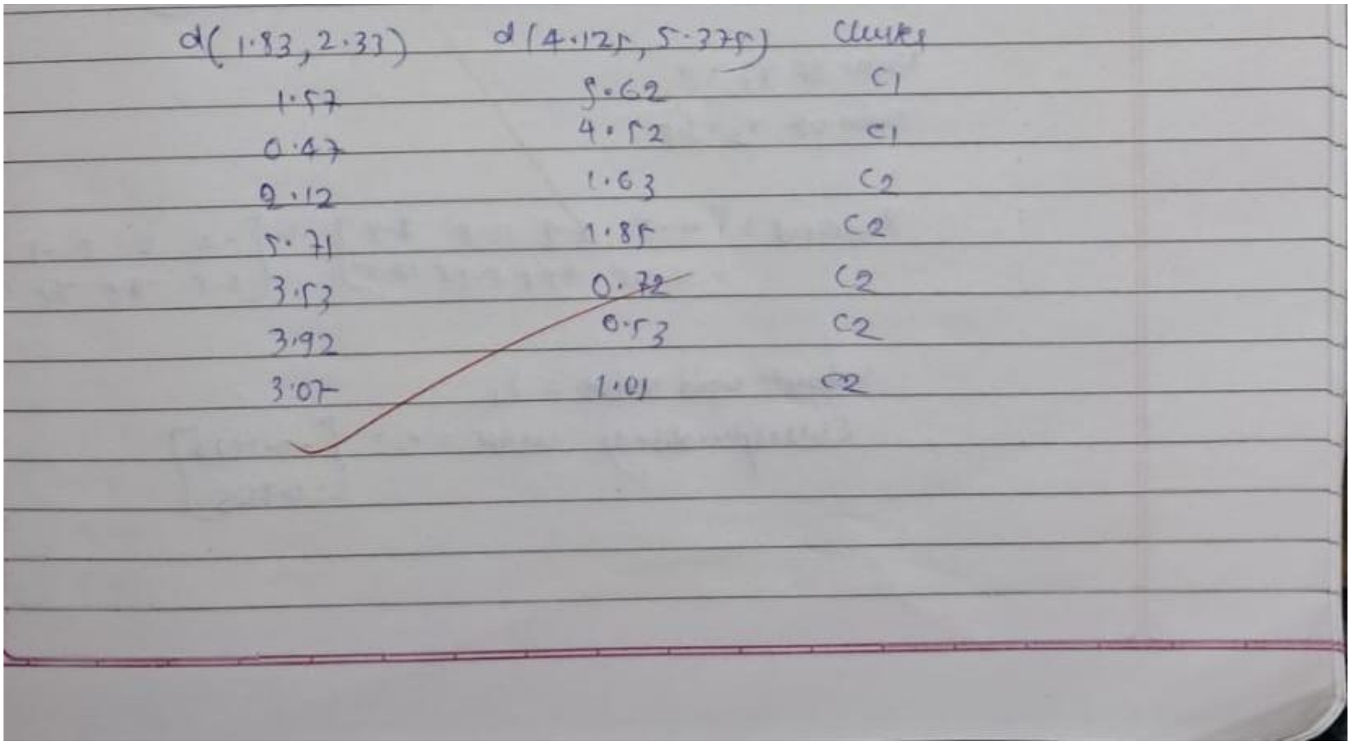
$c_1 = (1,1)$ & $c_2 = (5,7)$

$d = \sqrt{(x_2-x_1)^2 + (y_2-y_1)^2}$

Record	(A,B)	d(1,1)	d(5,7)	Cluster
R_1	(1,2)	0.0	7.21	C1
R_2	(1.5,2)	1.1	6.1	C1
R_3	(3,4)	3.6	4.2	C1
R_4	(5,7)	7.2	0	C2
R_5	(3.5,5)	5.0	2.5	C2
R_6	(1.5,5.)	5.32	2.2	C2
R_7	(3.5,4.5)	4.3	3.2	C2

$$C_1 = \left(\frac{1+1.5+3}{3}, \frac{1+2+4}{3}\right) = (1.83, 2.33)$$

$$C_2 = \left(\frac{5+3.5+4.5+3.5}{4}, \frac{7+5+5+4.5}{4}\right) = (4.125, 5.375)$$



d(1·83, 2·33)	d(4·125, 5·375)	Cluster
1·57	5·62	C1
0·47	4·52	C1
2·12	1·63	C2
5·71	1·85	C2
3·53	0·72	C2
3·92	0·53	C2
3·07	1·01	C2

Code:

```
import pandas as pd

import matplotlib.pyplot as plt

from sklearn.cluster import KMeans

from sklearn.preprocessing import StandardScaler


# 1. Load dataset

df = pd.read_csv("/content/iris.csv")


# 2. Drop Sepal features, keep only Petal length and width

X = df[['petal_length', 'petal_width']]


# 3. Scaling (K-Means is distance-based, scaling helps!)
```

```
scaler = StandardScaler()

X_scaled = scaler.fit_transform(X)


# 4. Elbow Method to find optimal k

inertia = []

k_range = range(1, 11)


for k in k_range:

    kmeans = KMeans(n_clusters=k, random_state=42)

    kmeans.fit(X_scaled)

    inertia.append(kmeans.inertia_)


# Plot elbow curve

plt.figure(figsize=(8,5))

plt.plot(k_range, inertia, 'bo-')

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

plt.ylabel('Inertia (Within-cluster sum of squares)')

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

plt.grid(True)

plt.show()


# 5. Choose optimal k (usually at 'elbow' point, say k=3)

optimal_k = 3

kmeans = KMeans(n_clusters=optimal_k, random_state=42)
```

```
clusters = kmeans.fit_predict(X_scaled)

# 6. Add clusters to original dataframe for visualization
df['Cluster'] = clusters

# Plot clusters
plt.figure(figsize=(8,5))
for i in range(optimal_k):
    cluster_data = df[df['Cluster'] == i]
    plt.scatter(cluster_data['petal_length'], cluster_data['petal_width'], label=f'Cluster {i}')

plt.xlabel('Petal Length')
plt.ylabel('Petal Width')
plt.title('K-Means Clustering of Iris (Petal Features)')
plt.legend()
plt.grid(True)
plt.show()
```