Name : Akshit Kothari
Section : BCA A1
Roll no : 07 (2221118)

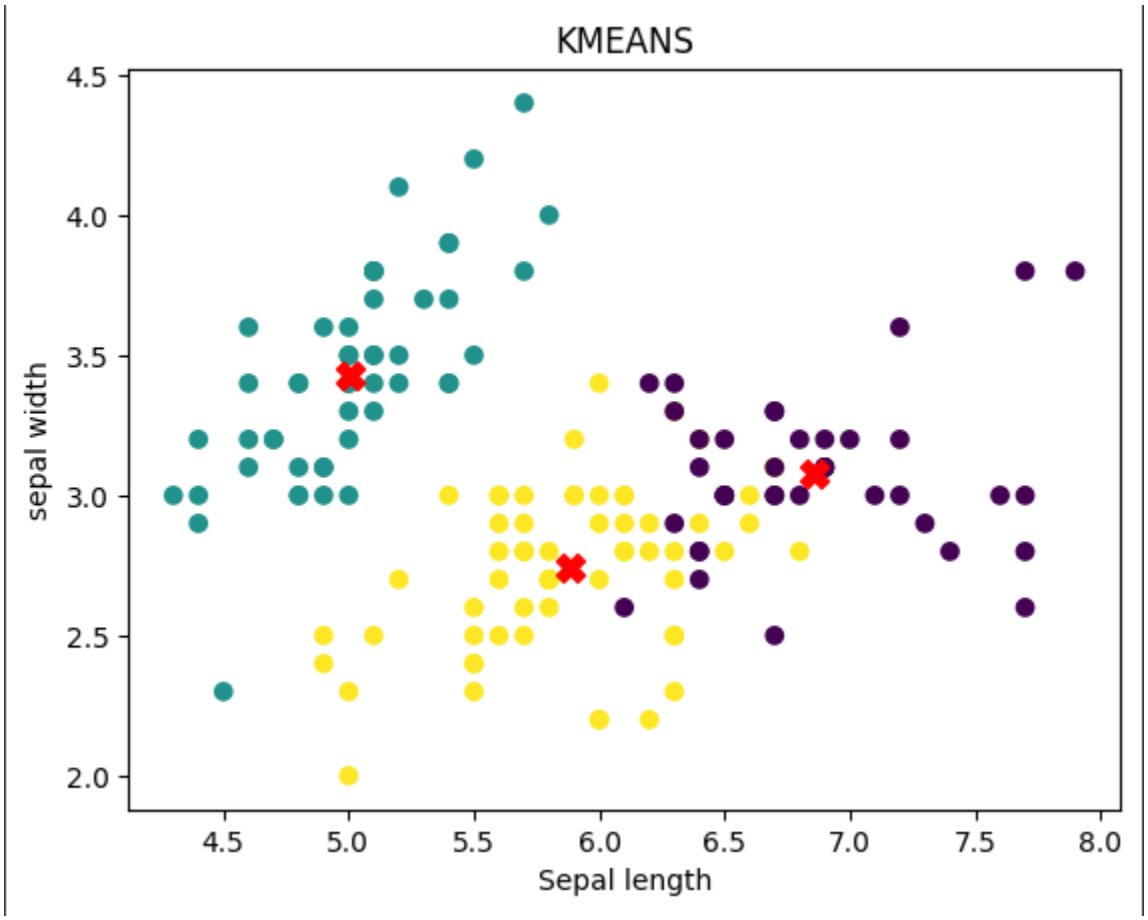
PROBLEM STATEMENT (9):- Write a program to implement k-means clustering algorithm by taking suitable dataset.

PROGRAM:-

```
from sklearn.datasets import load_iris
iris=load_iris()
x=iris.data

from sklearn.cluster import KMeans
kmeans=KMeans(n_clusters=3,random_state=42)
kmeans.fit(x)

import matplotlib.pyplot as plt

labels=kmeans.labels_

plt.scatter(x[:,0],x[:,1],c=labels,cmap='viridis')
plt.scatter(kmeans.cluster_centers_[:,0],kmeans.cluster_centers_[:,1],s=100,c='red',marker='X',la
bel='Centroids')
plt.xlabel('Sepal length')
plt.ylabel('sepal width')
plt.title('KMEANS')
plt.show()
```

Name : Akshit Kothari
Section : BCA A1
Roll no : 07 (2221118)

Output:



Name : Akshit Kothari
Section : BCA A1
Roll no : 07 (2221118)

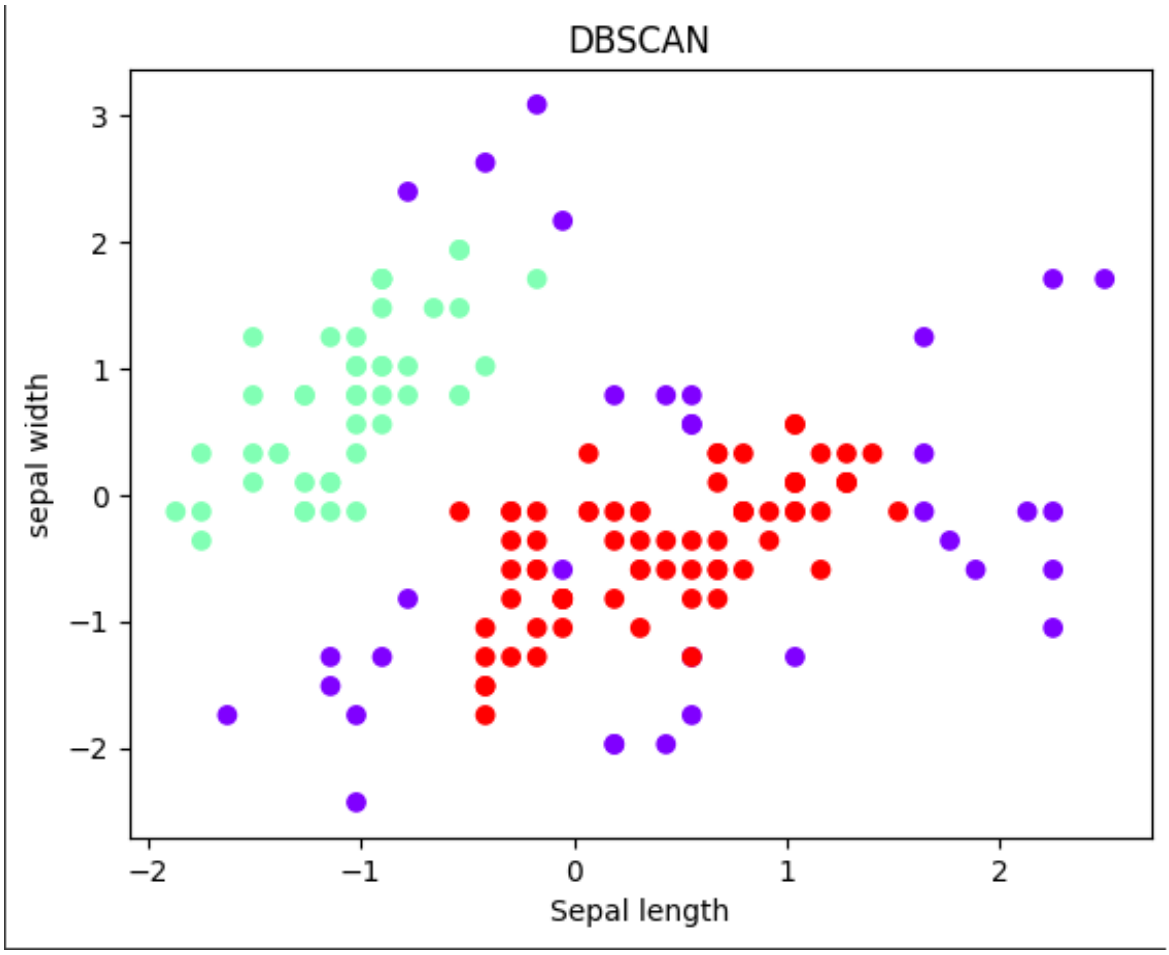
PROBLEM STATEMENT (10):- Write a program to implement DBScan.

PROGRAM:-

```
from sklearn.datasets import load_iris

iris=load_iris()
x=iris.data

from sklearn.preprocessing import StandardScaler

scaler = StandardScaler()
X_scaled = scaler.fit_transform(x)

from sklearn.cluster import DBSCAN

dbscan=DBSCAN(eps=0.5,min_samples=5)
labels=dbscan.fit_predict(X_scaled)

import matplotlib.pyplot as plt

plt.scatter(X_scaled[:,0],X_scaled[:,1],c=labels,cmap='rainbow')
plt.xlabel('Sepal length')
plt.ylabel('sepal width')
plt.title('DBSCAN')
plt.show();
```

Name : Akshit Kothari
Section : BCA A1
Roll no : 07 (2221118)

Output:



Name : Akshit Kothari
Section : BCA A1
Roll no : 07 (2221118)

PROBLEM STATEMENT (11):- Write a program to implement low variance filter.

PROGRAM:-

```
import pandas as pd
import numpy as np
from sklearn.datasets import load_iris

iris=load_iris()
x=iris.data


print("dimensons before low variance :  ",x.shape)

variance=np.var(x,axis=0)
print("Variances:  ",variance)

threshold=0.8

low_var=np.where(variance<threshold)

x_filter=np.delete(x,low_var,axis=1)

print("dimensons after low variance: ",x_filter.shape)
```

Output:

```
dimensons before low variance :  (150, 4)
Variances:  [0.68112222 0.18871289 3.09550267 0.57713289]
dimensons after low variance :  (150, 1)
```

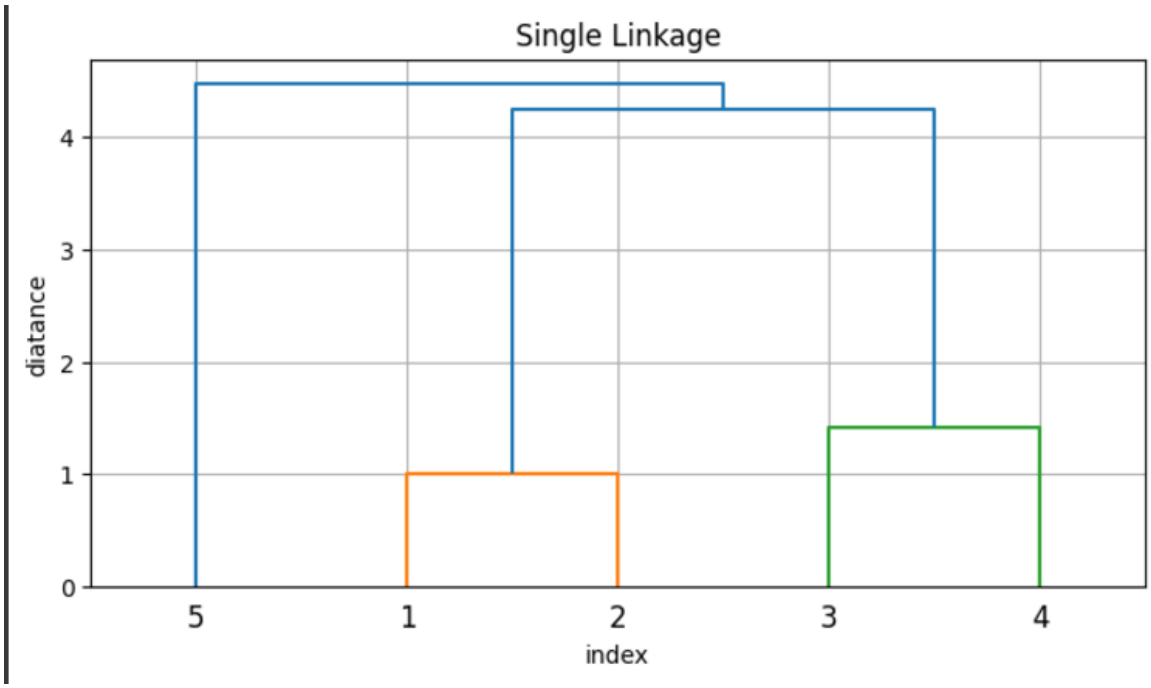
Name : Akshit Kothari
Section : BCA A1
Roll no : 07 (2221118)

PROBLEM STATEMENT (12):- Write a program to implement single linkage hierarchical clustering.

PROGRAM:-

```
import numpy as np
from scipy.cluster.hierarchy import linkage,dendrogram
import matplotlib.pyplot as plt

X=np.array([
   [1,2],[2,2],[5,5],[6,6],[10,8]
])

linked=linkage(X,method='single')
plt.figure(figsize=(8,4))
dendrogram(linked,labels=np.arange(1,len(X)+1),distance_sort='ascending',show_leaf_counts=T
rue)
plt.title("Single Linkage")
plt.xlabel('index')
plt.ylabel('diatance')
plt.grid(True)
plt.show()
```

Output:



Name : Akshit Kothari
Section : BCA A1
Roll no : 07 (2221118)

PROBLEM STATEMENT (13):- Write a program to implement linear regression.

PROGRAM:-

```
import numpy as np
import matplotlib.pyplot as plt
x=np.array([10,20,30,40,50])
y=np.array([11,20,9,16,39])
n=len(x)

numerator=n*np.sum(x*y)-(np.sum(x)*np.sum(y))
denominator=n*np.sum(x**2)-(np.sum(x)**2)

m=numerator/denominator
b=(np.sum(y)-m*np.sum(x))/n

y_pred=m*x+b
print(m)
print(b)
print(y_pred)

plt.figure(figsize=(8,5))
plt.scatter(x,y,label='data point')
plt.plot(x,y_pred,color='red',label='reg line')
plt.grid(True)
plt.xlabel('X')
plt.ylabel('y')
plt.title("Simple linear regression")
plt.legend()
plt.show()
```

Name : Akshit Kothari
Section : BCA A1
Roll no : 07 (2221118)

Output: