CL-II 4 IR

July 22, 2025

```
[ ]: # Implement Agglomerative hierarchical clustering algorithm using
     # appropriate dataset.
```

```
[ ]: '''NAME:Aher Swami Sandip
     ROLL NO.01
     COURSE: AI&DS
     CLASS: BE
     SUB:Computer Laboratory-II (Information Retrival)'''
```

```
[18]: import pandas as pd
      import numpy as np
      import matplotlib.pyplot as plt
      from sklearn.decomposition import PCA
      from sklearn.cluster import AgglomerativeClustering
      from sklearn.preprocessing import StandardScaler, normalize
      from sklearn.metrics import silhouette_score
      import scipy.cluster.hierarchy as shc
```

```
[34]: #Step 2: Loading and Cleaning the data
      df=pd.read_csv("Customer_Data.csv")
```

```
[36]: X.head()
```

```
[36]:      ID  Year_Birth   Education Marital_Status    Income  Kidhome  Teenhome  \
      0  1000        1978         PhD       Together  24342.06        1         0
      1  1001        1991      Master          Widow  51784.68        0         0
      2  1002        1968       Basic       Divorced  65007.28        2         1
      3  1003        1954  Graduation          Widow  52286.31        1         0
      4  1004        1982         PhD       Together  40979.04        1         0

        Dt_Customer  Recency  MntWines  …  NumWebVisitsMonth  AcceptedCmp3  \
      0  01-01-2012       38       373  …                  1             1
      1  12-01-2012       90        64  …                  8             1
      2  23-01-2012       73       145  …                  3             0
      3  03-02-2012       89       223  …                  2             1
      4  14-02-2012       18       238  …                  1             1

         AcceptedCmp4  AcceptedCmp5  AcceptedCmp1  AcceptedCmp2  Complain  \
      0             1             0             0             1         1
      1             0             0             0             0         1
      2             0             1             1             1         0
      3             0             0             1             1         1
      4             0             0             0             0         0

         Z_CostContact  Z_Revenue  Response
      0              3         11         1
      1              3         11         0
      2              3         11         1
      3              3         11         1
      4              3         11         1

      [5 rows x 25 columns]
```

[38]:

```
# Handling the missing values
X.ffill(inplace=True)
```

[40]:

```
X = df.select_dtypes(include=[float, int])
```

[42]:

```
# Scaling the data so that all the features become comparable
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)
```

[44]:

```
X.dropna(inplace=True)
```

[46]:

```
# Normalizing the data so that the data approximately
# follows a Gaussian distribution
X_normalized = normalize(X_scaled)
```

[48]:

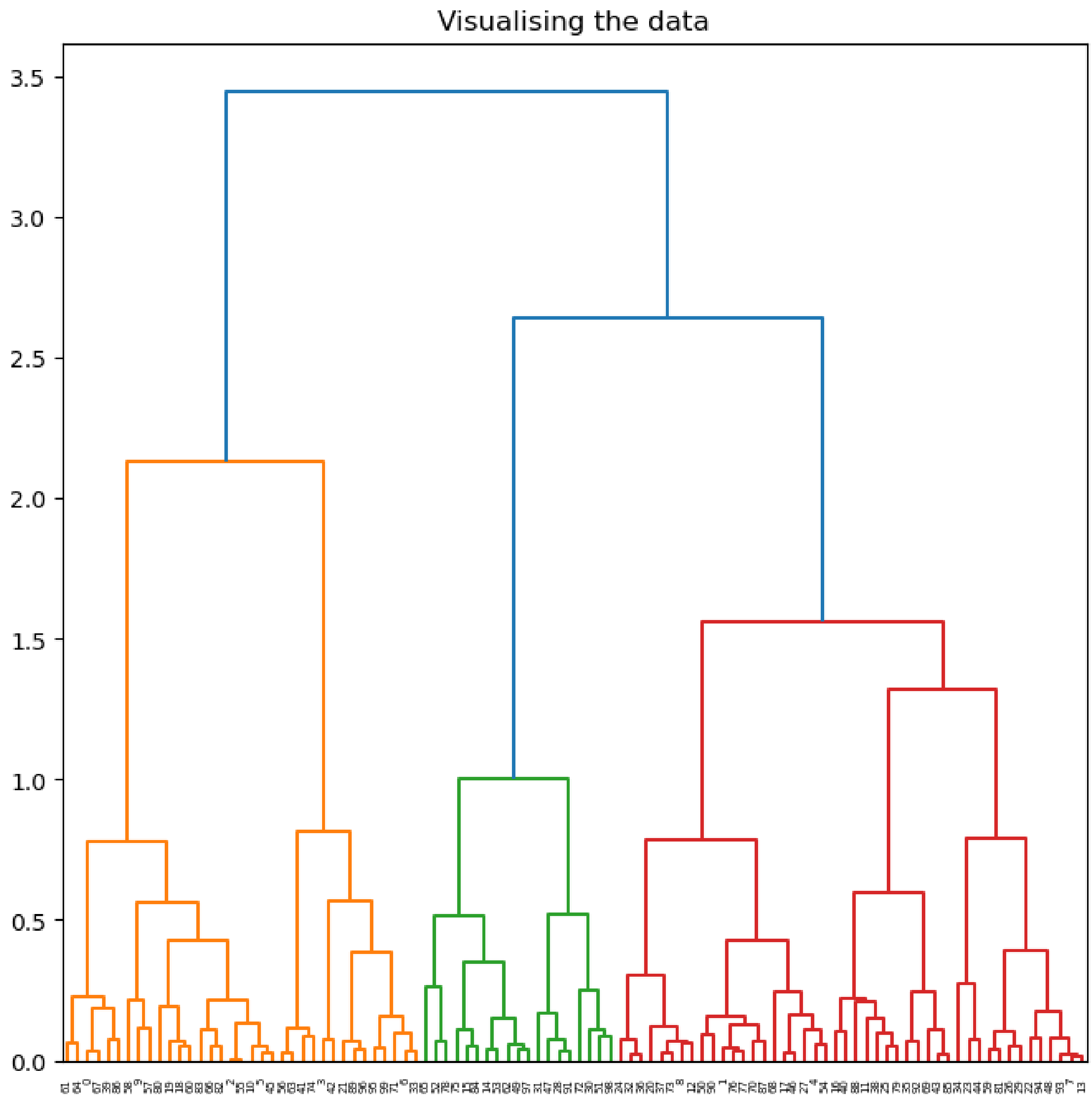
```
# Converting the numpy array into a pandas DataFrame
X_normalized = pd.DataFrame(X_normalized)
```

[50]:

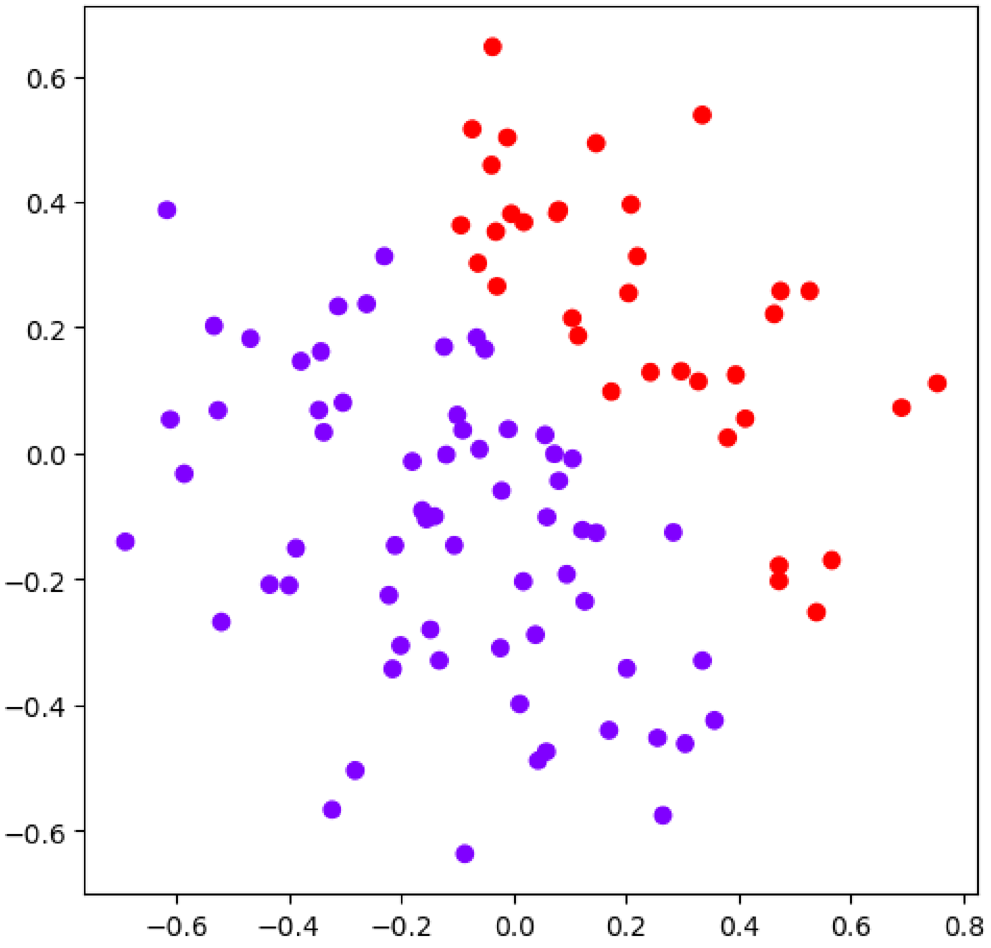
```
#Step 4: Reducing the dimensionality of the Data
pca = PCA(n_components = 2)
X_principal = pca.fit_transform(X_normalized)
X_principal = pd.DataFrame(X_principal)
X_principal.columns = ['P1', 'P2']
```

[52]:

```
 #Step 5: Visualizing the working of the Dendrograms
plt.figure(figsize =(8, 8))
plt.title('Visualising the data')
Dendrogram = shc.dendrogram((shc.linkage(X_principal, method ='ward')))
```

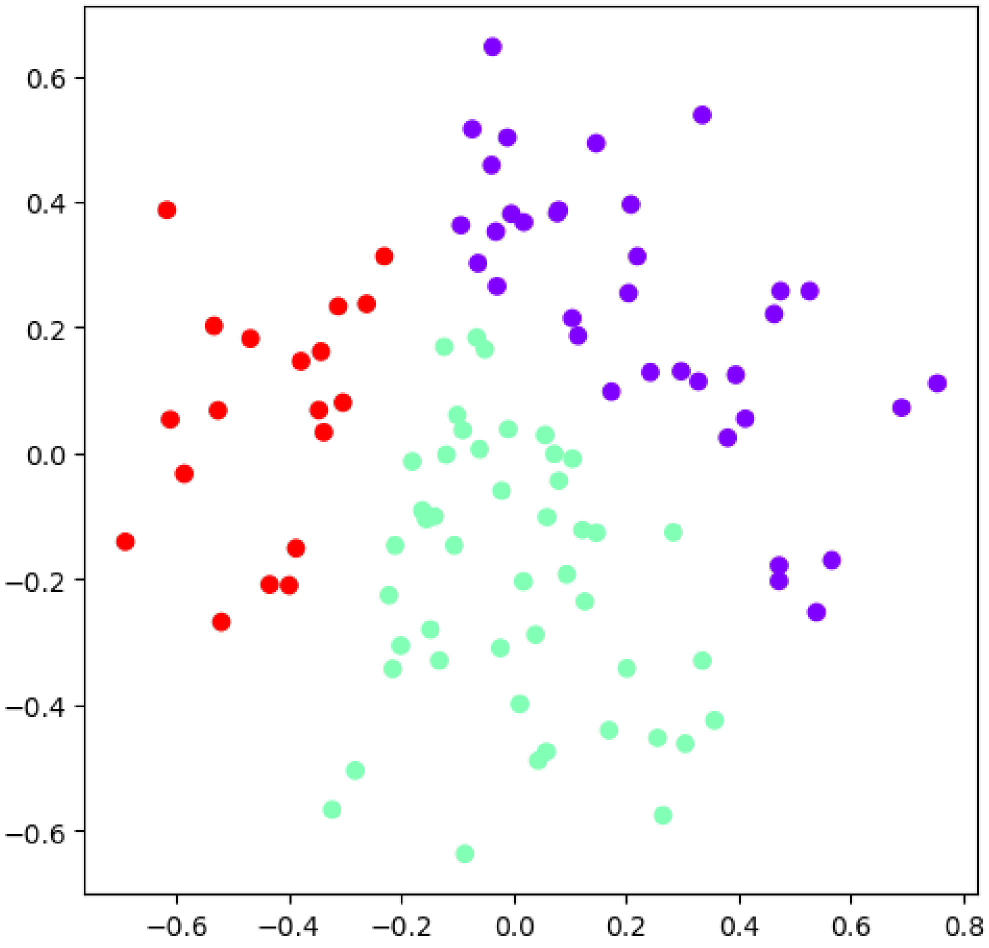


[53]:
```
#Step 6: Building and Visualizing the different clustering models for different␣
↪values of k a) k = 2
ac2 = AgglomerativeClustering(n_clusters = 2)
# Visualizing the clustering
plt.figure(figsize =(6, 6))
plt.scatter(X_principal['P1'], X_principal['P2'],
c = ac2.fit_predict(X_principal), cmap ='rainbow')
plt.show()
```



[56]:

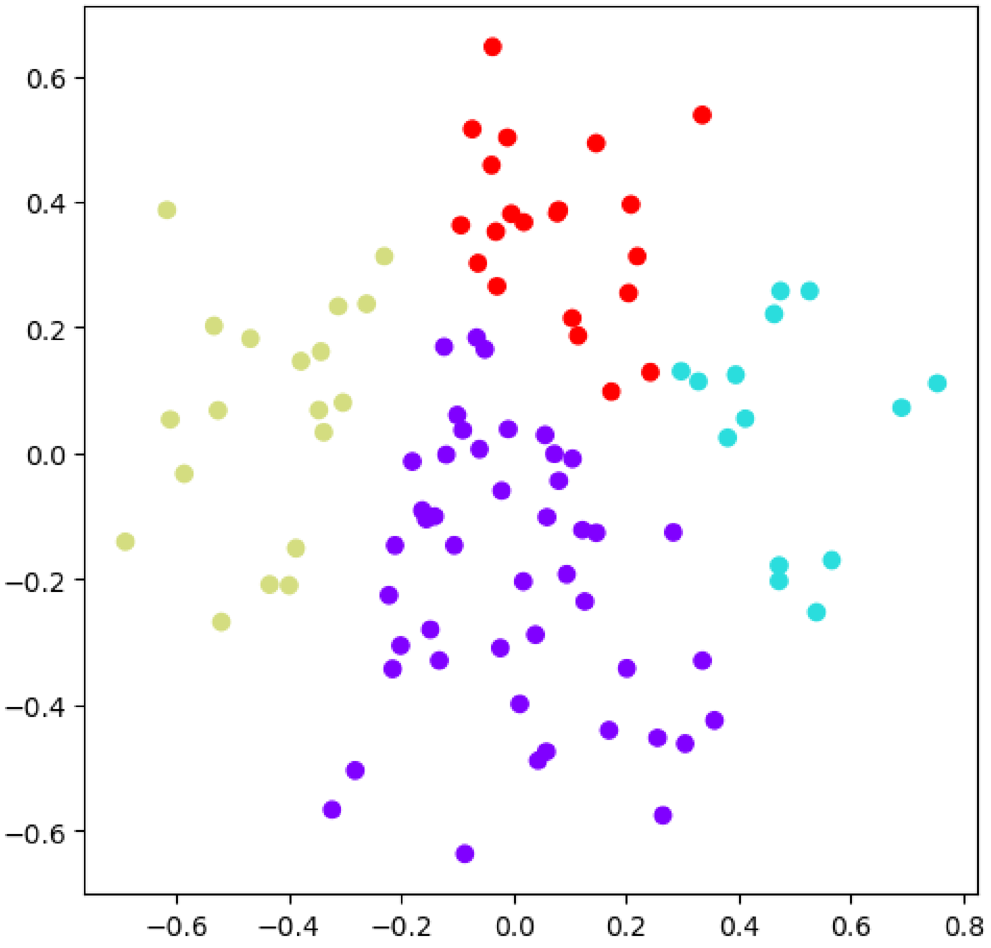
```
ac3 = AgglomerativeClustering(n_clusters = 3)
plt.figure(figsize =(6, 6))
plt.scatter(X_principal['P1'], X_principal['P2'],
c = ac3.fit_predict(X_principal), cmap ='rainbow')
plt.show()
```



[58]:

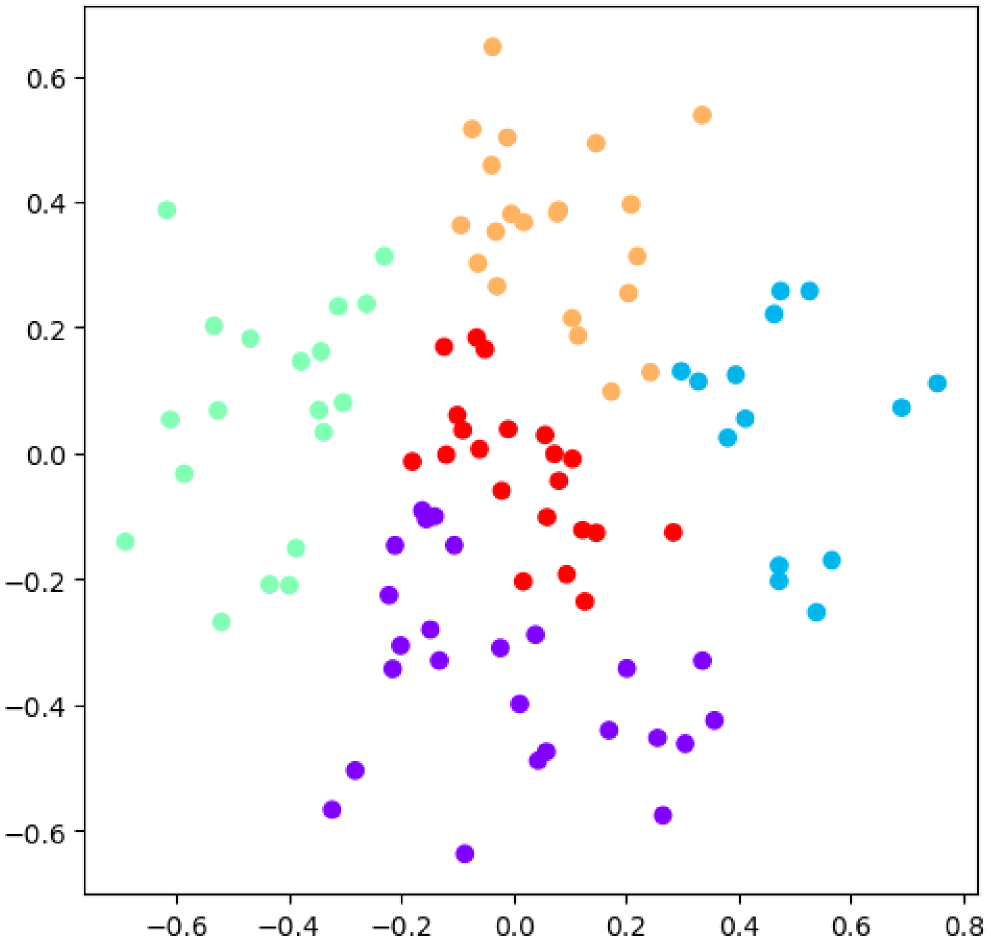
```
ac4 = AgglomerativeClustering(n_clusters = 4)

plt.figure(figsize =(6, 6))
plt.scatter(X_principal['P1'], X_principal['P2'],
c = ac4.fit_predict(X_principal), cmap ='rainbow')
plt.show()
```



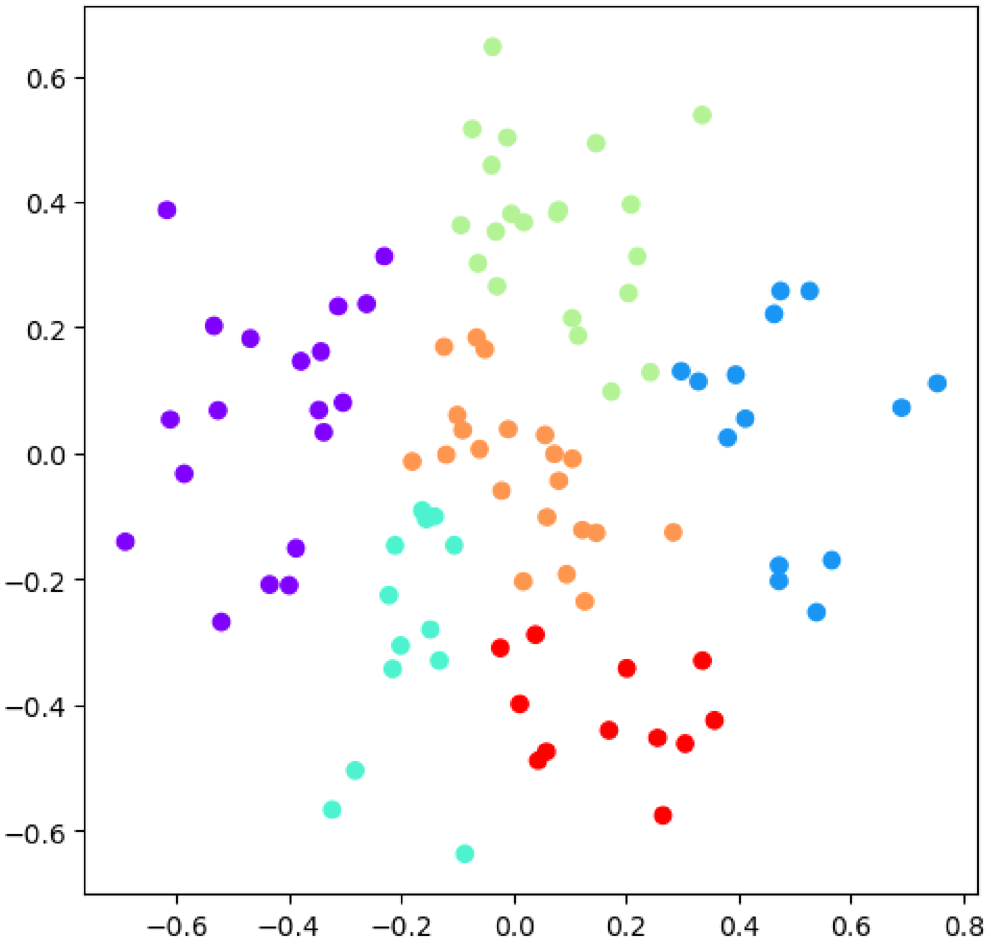
[60]:

```
ac5 = AgglomerativeClustering(n_clusters = 5)
plt.figure(figsize =(6, 6))
plt.scatter(X_principal['P1'], X_principal['P2'], c = ac5.
  ↪fit_predict(X_principal), cmap ='rainbow')
plt.show()
```



[62]:

```
ac6 = AgglomerativeClustering(n_clusters = 6)
plt.figure(figsize =(6, 6))
plt.scatter(X_principal['P1'], X_principal['P2'],
c = ac6.fit_predict(X_principal), cmap ='rainbow')
plt.show()
```

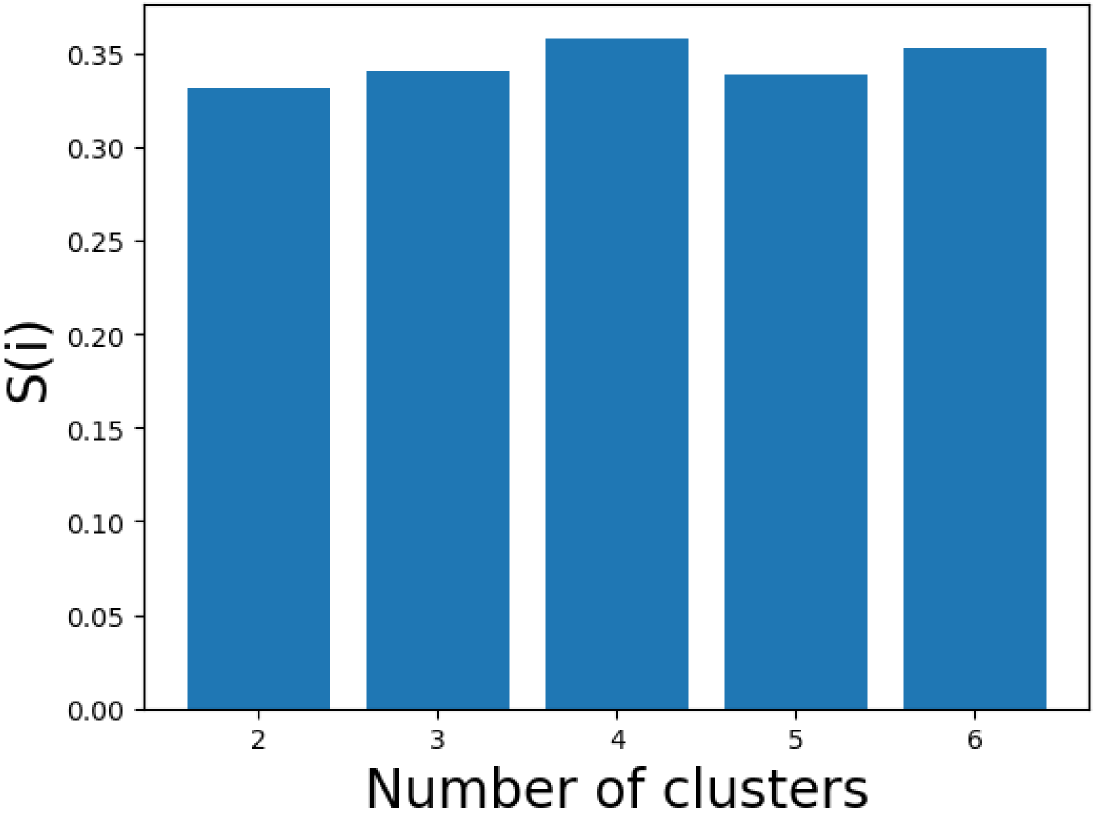


[64]:
```
#Step 7: Evaluating the different models and Visualizing the
#results.
k = [2, 3, 4, 5, 6]

# Appending the silhouette scores of the different models to the list
silhouette_scores = []
silhouette_scores.append(
silhouette_score(X_principal, ac2.fit_predict(X_principal)))
silhouette_scores.append(
silhouette_score(X_principal, ac3.fit_predict(X_principal)))
silhouette_scores.append(
silhouette_score(X_principal, ac4.fit_predict(X_principal)))
silhouette_scores.append(
silhouette_score(X_principal, ac5.fit_predict(X_principal)))
silhouette_scores.append(
silhouette_score(X_principal, ac6.fit_predict(X_principal)))
```

[66]:

```
# Plotting a bar graph to compare the results
plt.bar(k, silhouette_scores)
plt.xlabel('Number of clusters', fontsize = 20)
plt.ylabel('S(i)', fontsize = 20)
plt.show()
```



[ ]: