Assigment 3 Pattern Recognition Lab 2021-2022

In [1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
%matplotlib inline
from sklearn.cluster import KMeans
import numpy as np
import warnings
warnings.filterwarnings('ignore') # σε περιπτωση που θα μας συστησει μια μεθοδο καλυτερα απο αυτην που μας βολε
#και δεν θελουμε να εχει αυτο το κοκκινο warning απο κατω!
```

Load the dataset

In [2]:
```
gl = pd.read_csv('glass.csv')# Ανεβαζουμε τα δεδομενα
```

In [3]:
```
from sklearn import preprocessing
min_max_scaler = preprocessing.MinMaxScaler()
gl_s = min_max_scaler.fit_transform(gl[['Rl','Na','Mg','Al','Si','K','Ca','Ba','Fe']]) # Κανουμε Scale 0-1
```

In [4]:
```
X = gl[['Rl','Na']] #θελουμε να δουμε αποτελεσματα σχετικα με τα features αναλογια Na προς Rl
```

In [5]:
```
kmeans = KMeans(n_clusters = 6,random_state=0).fit(X) #Χρησιμοποιουμε τον αλγοριθμο k-means
centers = kmeans.cluster_centers_ # βρισκουμε το κεντρο απο καθε feature
centers
```

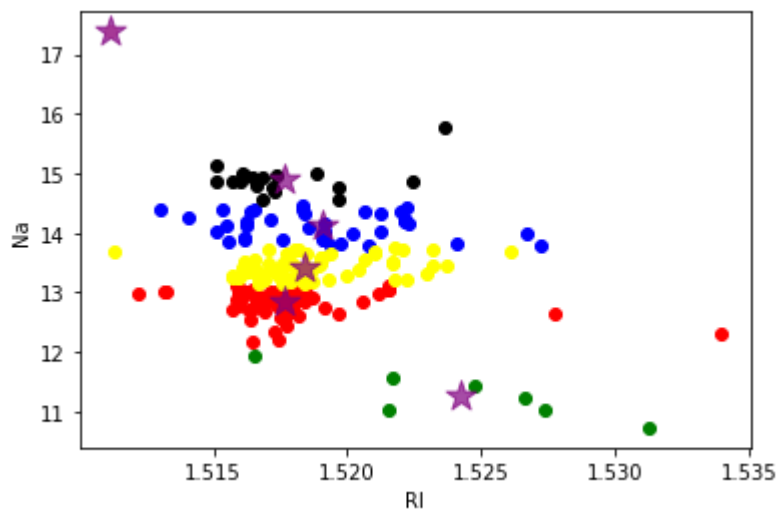
Out[5]:
```
array([[ 1.51766487, 12.82986842],
       [ 1.51907949, 14.11692308],
       [ 1.52425429, 11.28142857],
       [ 1.51843233, 13.41150685],
       [ 1.51115   , 17.38      ],
       [ 1.51761556, 14.90333333]])
```

In [6]:
```
#y_predicted = kmeans.fit_predict(X)
y_predicted = kmeans.fit_predict(X) # κανουμε τα prediction και βλεπουμε σε ποιες ομαδες μπαινουν μαζι τα δεδομ
y_predicted
```

Out[6]:
```
array([3, 1, 3, 3, 3, 0, 3, 3, 1, 0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 5,
       0, 0, 3, 0, 3, 0, 0, 0, 0, 0, 0, 0, 0, 3, 1, 0, 1, 1, 0, 0, 3, 3,
       0, 3, 3, 1, 3, 3, 3, 3, 3, 3, 3, 0, 0, 0, 3, 3, 3, 1, 3, 1, 3, 3,
       0, 0, 0, 3, 5, 3, 0, 3, 0, 0, 3, 0, 1, 0, 0, 3, 3, 0, 1, 3, 3, 3,
       0, 0, 0, 0, 0, 3, 0, 3, 0, 0, 0, 0, 0, 0, 0, 1, 1, 2, 2, 0, 1, 3,
       2, 2, 0, 3, 0, 3, 3, 3, 3, 3, 3, 0, 3, 3, 3, 0, 0, 1, 3, 1, 3, 3,
       3, 3, 3, 3, 0, 0, 0, 0, 3, 3, 0, 0, 0, 0, 3, 3, 3, 0, 3, 1, 3, 3,
       0, 0, 3, 1, 3, 3, 3, 3, 1, 1, 0, 2, 2, 0, 0, 3, 3, 0, 0, 3, 0, 0,
       1, 1, 1, 1, 1, 5, 1, 5, 4, 3, 1, 3, 5, 5, 1, 5, 1, 5, 5, 1, 1, 5,
       1, 5, 5, 2, 5, 5, 5, 5, 5, 1, 1, 1, 5, 1, 1, 1])
```

In [7]:
```
X['cluster']=y_predicted# προσθετουμε τα clusters διπλα απο τα values των Rl και Na
#Τα σπαμε ανα cluster
filtered_label1 = X[X.cluster == 0]
filtered_label2 = X[X.cluster == 1]
filtered_label3 = X[X.cluster == 2]
filtered_label4 = X[X.cluster == 3]
filtered_label5 = X[X.cluster == 4]
filtered_label6 = X[X.cluster == 5]
```

In [8]:
```
#Plotting τα αποτελεσματα
plt.scatter(filtered_label1.Rl , filtered_label1.Na , color = 'red')
plt.scatter(filtered_label2.Rl , filtered_label2.Na , color = 'blue')
plt.scatter(filtered_label3.Rl , filtered_label3.Na , color = 'green')
plt.scatter(filtered_label4.Rl , filtered_label4.Na , color = 'yellow')
plt.scatter(filtered_label5.Rl , filtered_label5.Na , color = 'pink')
plt.scatter(filtered_label6.Rl , filtered_label6.Na , color = 'black') #μαλλον ειναι Outlier
plt.scatter(centers[:,0], centers[:,1], c='purple', s=250, alpha=0.7,marker='*');#Plot τα κεντρα των ομαδων αν
plt.xlabel('Rl')
plt.ylabel('Na')
plt.show()
```



Ward Linkage

In [9]:
```
from scipy.cluster.hierarchy import ward, dendrogram,linkage
np.set_printoptions(precision=4, suppress=True)
distance = linkage(X, 'ward') #Χρησιμοποιουμε την μεθοδο ward για να κανουμε το Linkage και μετα να
#δημιουργηθει το δεντρογραμμα

# Dendrogram
plt.figure(figsize=(25,10))
plt.title("Hierachical Clustering Dendrogram")
plt.xlabel("Index")
plt.ylabel("Ward's distance")
dendrogram(distance,leaf_rotation=90.,leaf_font_size=9.,);
```


Αυτό που παρατηρείται στην μέθοδο ιεραρχίας είναι ότι χρησιμοποιεί μια πολύ καλά δομημένη μέθοδο στην αναπαράσταση και υπολογισμό ενός dataset μέσο της ιεραρχίας και αυτό το κάνει με το να παίρνει τα δικά του clusters ενώ με την μέθοδο K-Means πρέπει να αποφασίσουμε εμείς το ποσά clusters θα βάλουμε και αυτό γίνεται συνήθως με την καμπύλη του αγκώνα η του γονάτου που αυτό θα υπολογίσει το ποσά clusters θα ήταν η καλύτερη λύση στο dataset μας. Ουσιαστικά στο K-means αλλάζει το κέντρο των clusters ανά Κ και αν πάρουμε ένα παράδειγμα με πίτσες αρχικά θα δούμε αποτελέσματα όπως μερικά διαμερίσματα να παραγγέλνουν πίτσες από το πιο κοντινό (σχετικά με την απόσταση) εστιατόριο και μετα αν πάρουμε το Μ.Ο από της 1ες ομάδες και ξανακάνουμε clustering (ανάλογα και το dataset) μπορεί να αλλάξει το κέντρο και μερικά διαμερίσματα να αλλάξουν το μαγαζί που πρωτοέπαιρναν πίτσα.

Κωνσταντινος Παπη ΤΠ4720