## مقدمه ای بر خوشه بندی سلسله مراتبی (Hierarchical clustering):

خوشه بندی سلسله مراتبی یکی دیگر از الگوریتم های یادگیری بدون نظارت است که از آن برای گروه بندی نقاط داده بدون برچسب که خصوصیات مشابه دارند استفاده می شود. الگوریتم های خوشه بندی سلسله مراتبی به دو دسته زیر تقسیم می شوند:

### الگوریتم های سلسله مراتبی جمع کننده (Agglomerative) :

در الگوریتم های سلسله مراتبی جمع کننده، هر نقطه داده به عنوان یک خوشه تکی در نظر گرفته می شود، سپس به طور پی در پی با جفتی از خوشه ها ادغام یا جمع (روش پایین به بالا)می شود. سلسله مراتب خوشه ها به صورت dendrogram یا ساختار درختی نمایش داده می شود.

### الگوریتم های سلسله مراتبی تقسیم کننده (Divisive) :

از طرف دیگر، در الگوریتم های سلسله مراتبی تقسیم کننده، تمامی نقاط داده به عنوان یک خوشه بزرگ در نظر گفته می شوند و روند خوشه بندی شامل تقسیم (روش بالا به پایین) یک خوشه بزرگ به چندین خوشه کوچک است.

### مراحل اعمال خوشه بندی سلسله مراتبی جمع کننده:

در اینجا به توضیح پر کاربرد ترین و مهم ترین خوشه بندی سلسله مراتبی یعنی جمع کننده می پردازیم. مراحل انجام آن به شرح زیر است:

**مرحله ۱-** هر نقطه داده را یک خوشه تکی در نظر بگیرید. در این صورت در ابتدا k تا خوشه خواهیم داشت. تعداد نقاط داده نیز در ابتدا k خواهد بود.

**مرحله ۲-** حال، در این مرحله باید با ادغام دو نقطه داده closet ، یک خوشه بزرگ ایجاد کنیم. در نتیجه کلا k-1 خوشه خواهیم داشت.

**مرحله ۳-** حال، برای ایجاد خوشه های بیشتر باید دو خوشه closet را ادغام کنیم. در نتیجه کلا k-2 خوشه خواهیم داشت.

**مرحله ۴-** حال، برای ایجاد یک خوشه بزرگ سه مرحله فوق را تکرار کنید تا k تبدیل به 0 شود (هیچ نقاط داده ای برای ادغام باقی نمانده باشد).

**مرحله ۵-** در انتها، پس از ایجاد تنها یک خوشه بزرگ، از dendrograms برای تقسیم به چندین خوشه بر اساس مساله استفاده می شود.

## نقش dendrograms در خوشه بندی سلسله مراتبی جمع کننده:

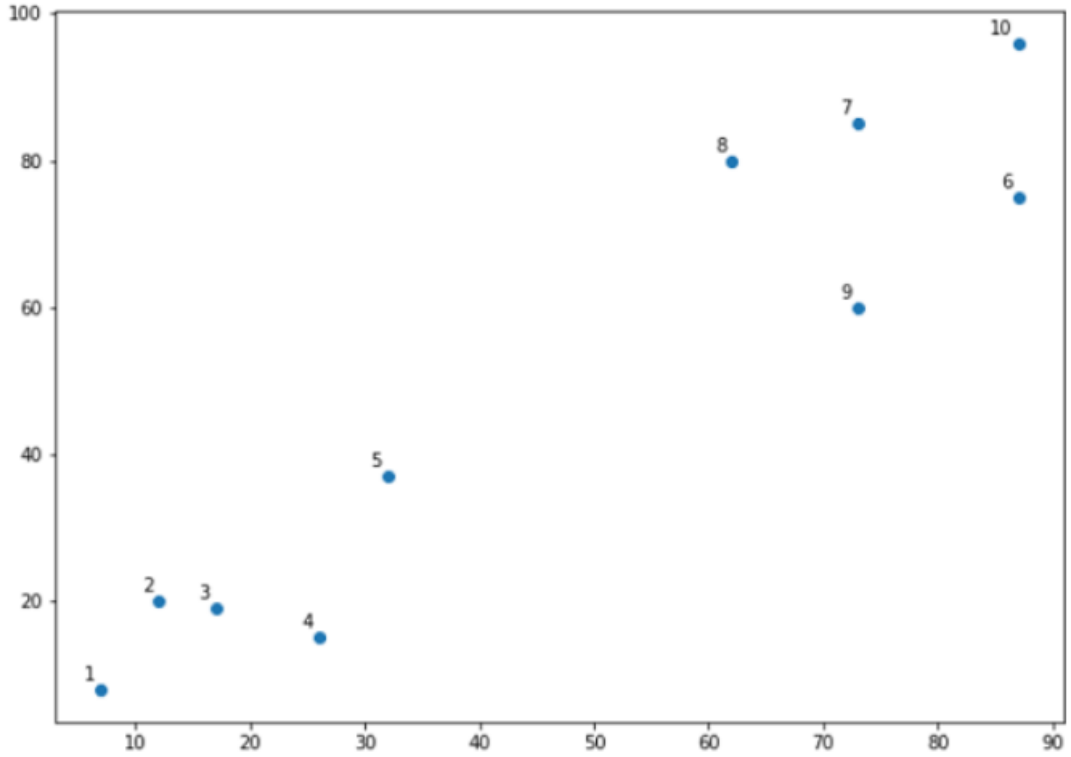
همان طور که در مرحله قبل توضیح داده شد، نقش dendrograms زمانی شروع می شود که خوشه بزرگ شکل گرفته باشد. از dendrograms برای تفکیک خوشه ها به چندین خوشه از نقاط داده مرتبط، براساس مساله خود استفاده می شود. با کمک مثال زیر می توان این مفهوم را درک کرد.

**مثال ۱:** با وارد کردن کتابخانه های مورد نیاز به صورت زیر شروع می کنیم.

```
%matplotlib inline
import matplotlib.pyplot as plt
import numpy as np
```

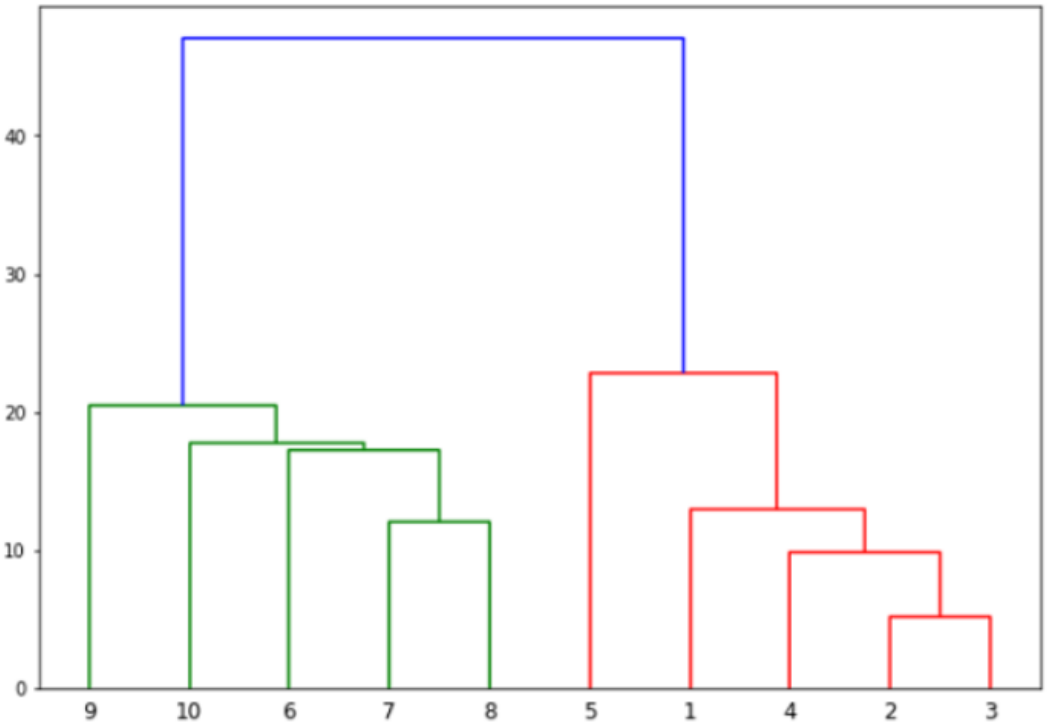
سپس، نقاط داده دریافت شده از این مثال را رسم می کنیم.

```
X =
np.array([[7,8],[12,20],[17,19],[26,15],[32,37],[87,7
5],[73,85], [62,80],[73,60],[87,96],])
labels = range(1, 11)
plt.figure(figsize=(10, 7))
plt.subplots_adjust(bottom=0.1)
plt.scatter(X[:,0],X[:,1], label='True Position')
for label, x, y in zip(labels, X[:, 0], X[:, 1]):
   plt.annotate(label,xy=(x, y), xytext=(-3,
3),textcoords='offset points', ha='right',
va='bottom')
plt.show()
```

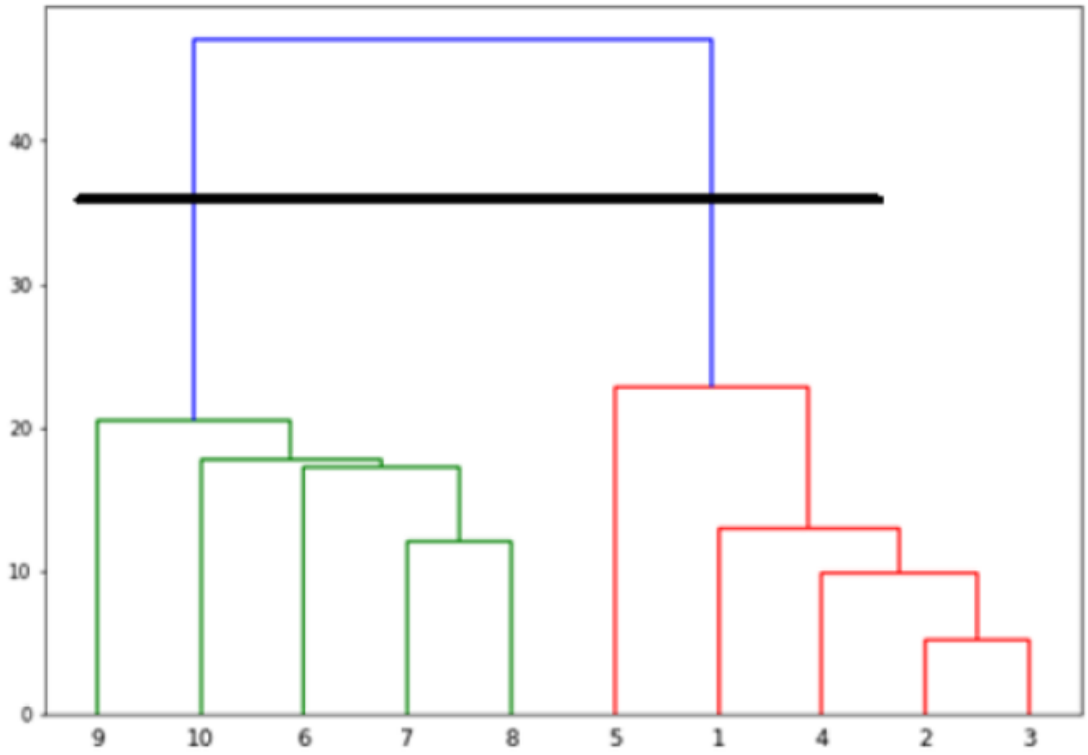



با توجه به نمودار فوق، به سادگی می توان مشاهده کرد که دو خوشه در نقاط داده داریم، اما در داده دنیای واقعی، می تواند هزاران خوشه وجود داشته باشد. سپس، با استفاده از کتابخانه Scipy ، dendrograms مربوط به نقاط داده خود را رسم خواهیم کرد.

```
from scipy.cluster.hierarchy import dendrogram, linkage
from matplotlib import pyplot as plt
linked = linkage(X, 'single')
labelList = range(1, 11)
plt.figure(figsize=(10, 7))
dendrogram(linked,
orientation='top',labels=labelList,
distance_sort='descending',show_leaf_counts=True)
plt.show()
```




حال، زمانی که خوشه بزرگ شکل گرفت، طولانی ترین فاصله عمودی انتخاب می شود. سپس مطابق شکل زیر یک خط عمودی از میان آن عبور داده می شود. از آنجایی که خط افقی خط آبی را در دو نقطه قطع می کند، تعداد خوشه ها ۲ خواهد بود.

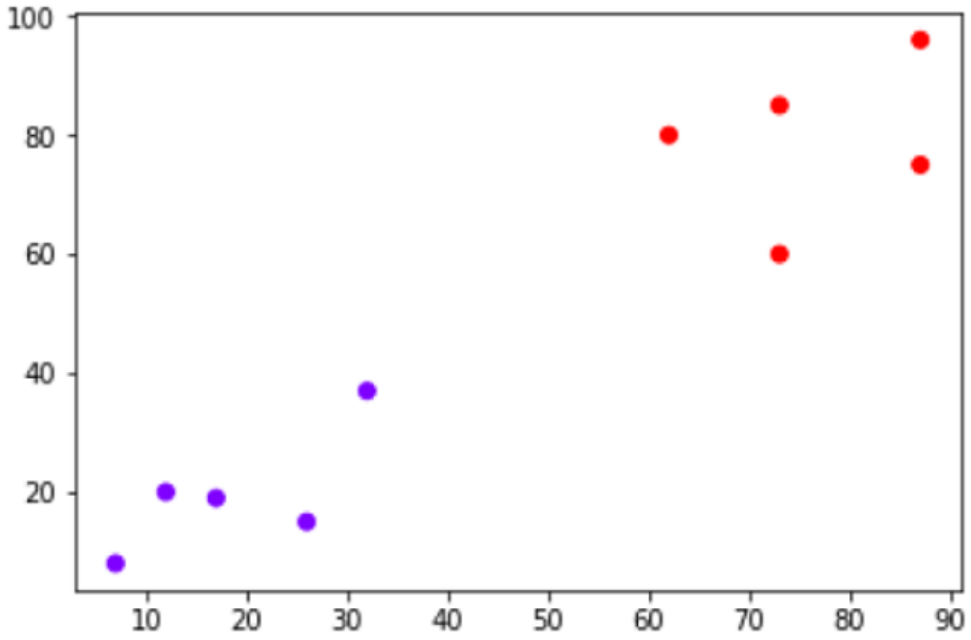



سپس، باید برای خوشه بندی، کلاس را وارد کنیم و متد fit_predict مربوط به آن را برای پیش بینی خوشه فراخوانی کنیم. کلاس AgglomerativeClustering از کتابخانه sklearn.cluster را وارد می کنیم.

```
from sklearn.cluster import AgglomerativeClustering
cluster = AgglomerativeClustering(n_clusters=2,
affinity='euclidean', linkage='ward')
cluster.fit_predict(X)
```

سپس، با کمک کد زیر خوشه را رسم کنید.

```
plt.scatter(X[:,0],X[:,1], c=cluster.labels_,
cmap='rainbow')
```



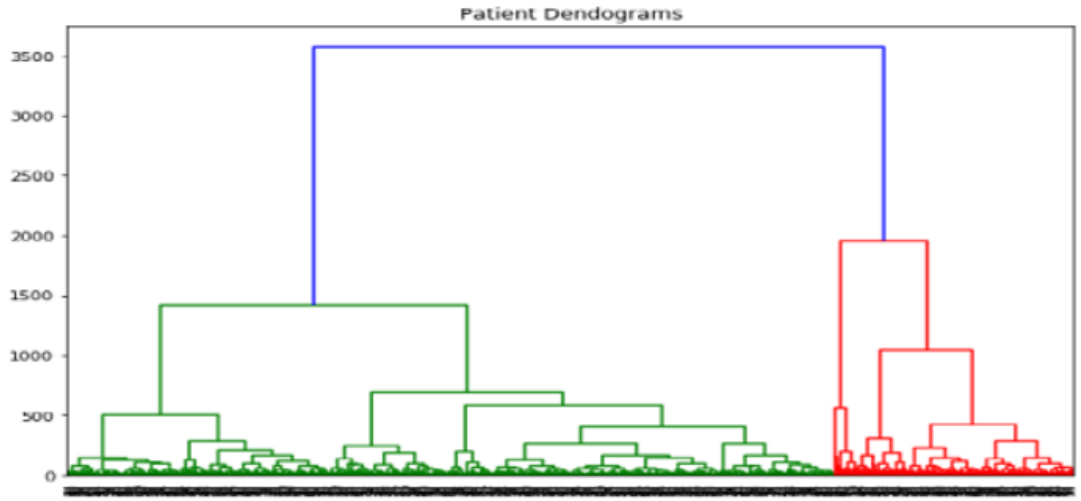

نمودار فوق دو خوشه مربوط به نقاط داده ما را نشان می دهد.

**مثال ۲ :** از آنجایی که توسط مثال ساده توضیح داده شده در بالا، مفهوم dendrograms را فهمیدیم، حال به یک مثال دیگر می پردازیم که در آن با استفاده از خوشه بندی سلسله مراتبی به ساخت خوشه هایی از نقاط داده از مجموعه داده Pima Indian Diabetes خواهیم پرداخت.

```
import matplotlib.pyplot as plt
import pandas as pd
%matplotlib inline
import numpy as np
from pandas import read_csv
path = r"C:\pima-indians-diabetes.csv"
headernames = ['preg', 'plas', 'pres', 'skin',
'test', 'mass', 'pedi', 'age', 'class']
data = read_csv(path, names=headernames)
array = data.values
X = array[:,0:8]
Y = array[:,8]
data.shape
(768, 9)
data.head()
```

| slno. | preg | Plas | Pres | skin | test | mass | pedi | age | class |
|---|---|---|---|---|---|---|---|---|---|
| 0 | 6 | 148 | 72 | 35 | 0 | 33.6 | 0.627 | 50 | 1 |
| 1 | 1 | 85 | 66 | 29 | 0 | 26.6 | 0.351 | 31 | 0 |
| 2 | 8 | 183 | 64 | 0 | 0 | 23.3 | 0.672 | 32 | 1 |
| 3 | 1 | 89 | 66 | 23 | 94 | 28.1 | 0.167 | 21 | 0 |
| 4 | 0 | 137 | 40 | 35 | 168 | 43.1 | 2.288 | 33 | 1 |

```
patient_data = data.iloc[:, 3:5].values
import scipy.cluster.hierarchy as shc
plt.figure(figsize=(10, 7))
plt.title("Patient Dendograms")
dend = shc.dendrogram(shc.linkage(data,
method='ward'))
```




```
from sklearn.cluster import AgglomerativeClustering
cluster = AgglomerativeClustering(n_clusters=4,
affinity='euclidean', linkage='ward')
cluster.fit_predict(patient_data)
plt.figure(figsize=(10, 7))
plt.scatter(patient_data[:,0], patient_data[:,1],
c=cluster.labels_, cmap='rainbow')
```