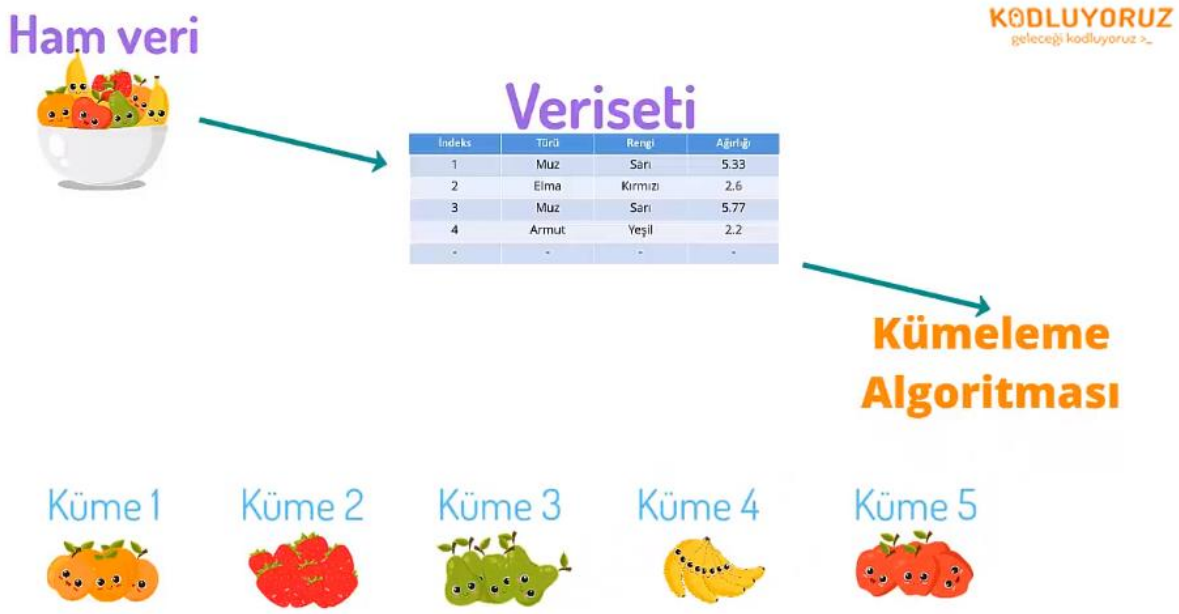
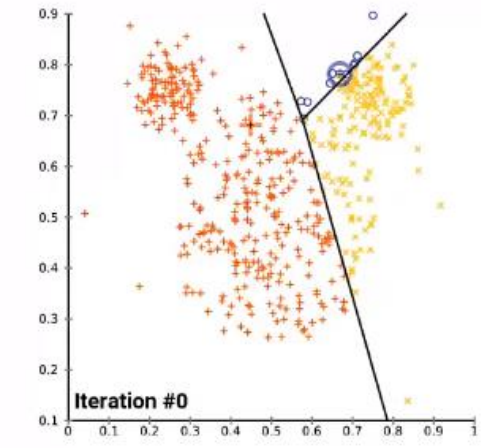
KÜMELEME ALGORİTMASI

Bir makine öğrenimi sistemindeki bir konuyu (veri setini) anlamak için ilk adım olarak örnekleri gruplandırıyoruz.

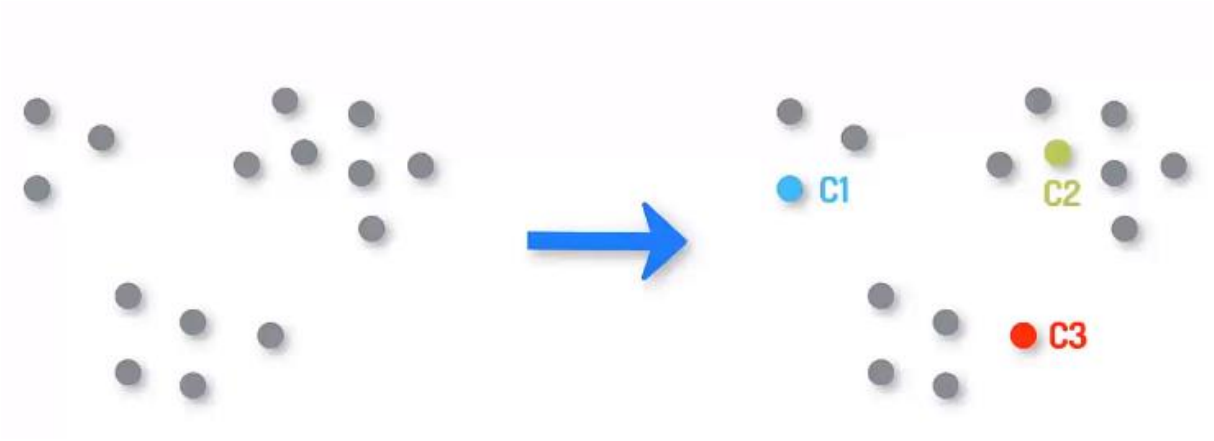


Buradaki örneklerimiz sadece girdi verilerinden oluşuyor. Yani herhangi bir çıktı verisi bulunmamakta. Bu etiketsiz örneklerin gruplandırılmasına ise kümeleme diyoruz.

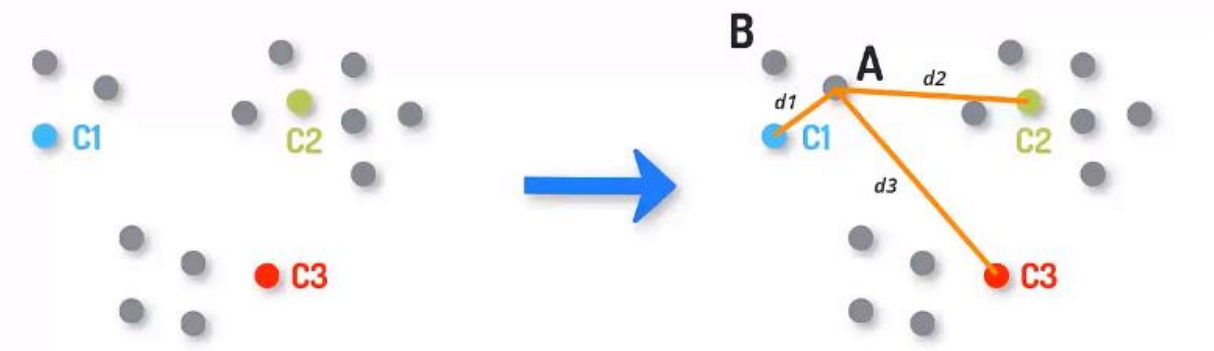
Kümeleme, basit bir deyişle, amaç benzer özelliklere sahip grupları ayırmak ve onları kümelemektir. Kümeleme denilince akla ilk gelen algoritma, K-Means algoritmasıdır. K-Means algoritmasında k değeri -> küme sayısını belirler ve bu değeri parametre olarak alması gerekir.



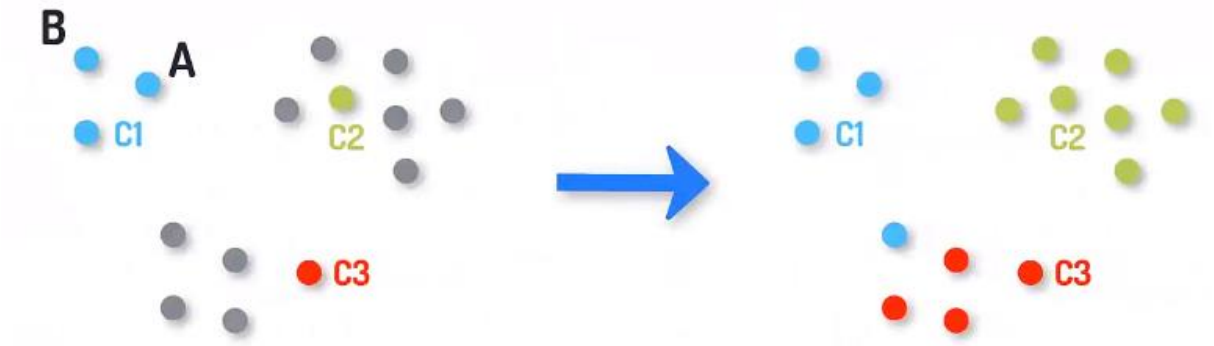
Örnek: İlk görselde bu gri noktalara sahibiz ve bunları üç kümeye ayırmak istiyoruz. Bunun için ilk olarak rastgele üç noktası C1, C2 ve C3 seçip, küme merkezlerini temsil etmek için ayrı ayrı mavi, yeşil ve kırmızı renklerle etiketliyoruz.



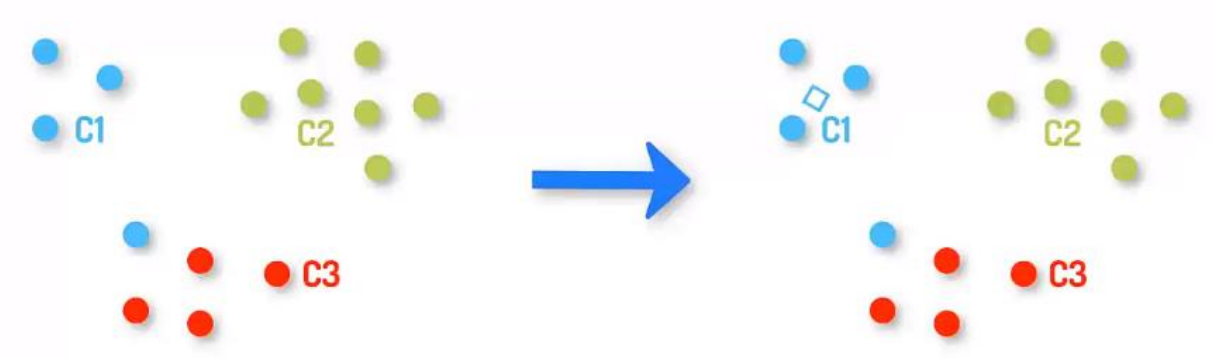
Her noktayı küme merkezine olan minimum mesafeye göre kümelere atamalıyız. Burada gördüğümüz A verisi için sırasıyla C1, C2 ve C3'e olan mesafesini hesaplarız. Ve d1, d2 ve d3 uzunluklarını karşılaştırdıktan sonra d1'in en küçük olduğunu anlıyoruz.



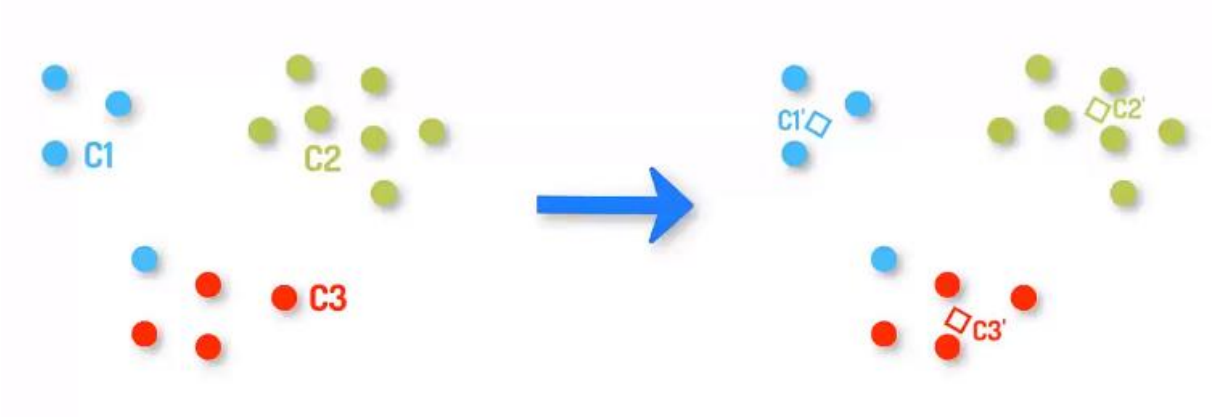
Bu nedenle mavi kümeye A noktasını atayıp mavi ile etiketliyoruz. Daha sonra aynı aşamaları B için yapıyoruz. Bu işlemi tüm noktalar için gerçekleştiriyoruz. Ve 3 farklı renkte gördüğümüz bu kümelemeyi elde ediyoruz.



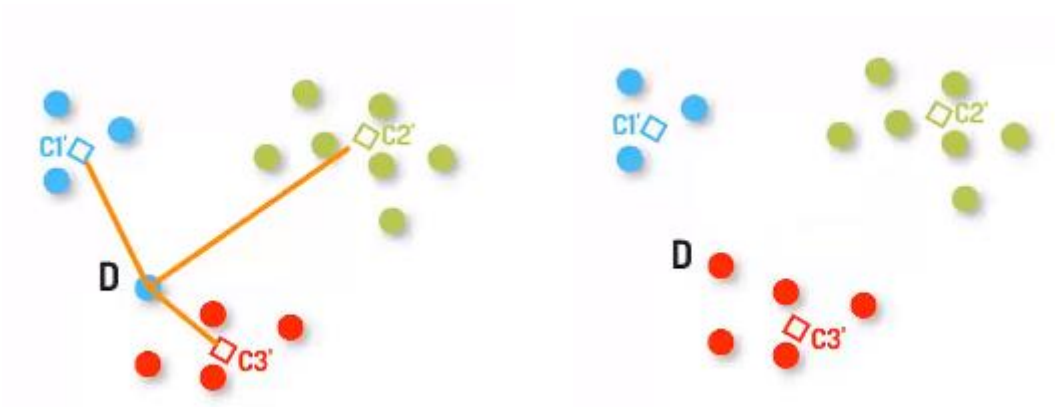
Tüm noktaları en yakın oldukları küme merkezine göre atadık. Ardından küme merkezlerini kendilerine atanan noktalara göre güncellememiz gerekiyor.



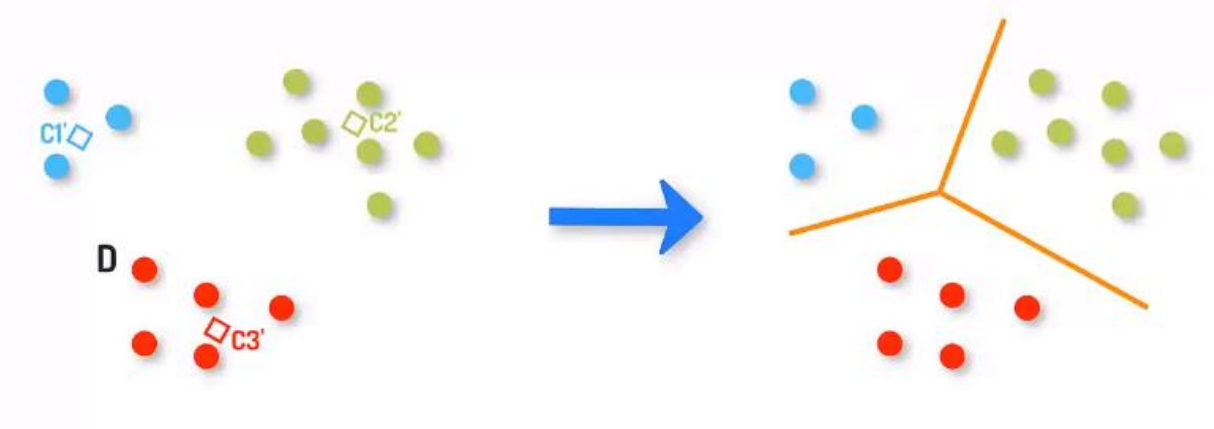
Örneğin; mavi kümenin merkez kütlesini, tüm mavi noktaları toplayarak ve burada 4 olan toplam nokta sayısına bölerek bulabiliriz. Ve mavi bir dörtgenle temsil edilen sonuçta ortaya çıkan merkez kütlesi C1', mavi küme için yeni merkezimizdir. Benzer şekilde yeşil ve kırmızı kümeler için yeni C2' ve C3' merkezlerini bulabiliriz.



Son adımımız ise sadece yukarıdaki iki adımı tekrar etmektir. D noktası C3' ne yaklaşır ve bu nedenle kırmızı kümeye atanabilir.



Noktaları küme merkezlerine atama ve yakınsamaya kadar küme merkezlerini güncelleme arasında yinelemeye devam ediyoruz ve son şeklimizi elde etmiş oluyoruz.



Kümeleme Algoritma Uygulaması

Kütüphane ekleyip, verimizi okuyalım:

```
[1] import pandas as pd

[2] df = pd.read_csv('/content/Mall_Customers.csv')

    df.head(10)
```

	CustomerID	Gender	Age	Annual Income (k$)	Spending Score (1-100)
0	1	Male	19	15	39
1	2	Male	21	15	81
2	3	Female	20	16	6
3	4	Female	23	16	77
4	5	Female	31	17	40
5	6	Female	22	17	76
6	7	Female	35	18	6
7	8	Female	23	18	94
8	9	Male	64	19	3
9	10	Female	30	19	72

Boyutumuzu öğrenelim:

```
df.shape
```

```
(200, 5)
```

Genel bilgilere bakalım:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 200 entries, 0 to 199
Data columns (total 5 columns):
 #   Column                  Non-Null Count  Dtype
---  ------                  --------------  -----
 0   CustomerID              200 non-null    int64
 1   Gender                  200 non-null    object
 2   Age                     200 non-null    int64
 3   Annual Income (k$)      200 non-null    int64
 4   Spending Score (1-100)  200 non-null    int64
dtypes: int64(4), object(1)
memory usage: 7.9+ KB
```

Eksik değerlerimizi kontrol edelim:

```
df.isnull().sum()
```

```
CustomerID                0
Gender                    0
Age                       0
Annual Income (k$)        0
Spending Score (1-100)    0
dtype: int64
```

Boş değerimiz yok. CustomerId kolonunu silelim, kullanmayacağız.

```
[7] df.drop('CustomerID', axis=1, inplace=True )
```

```
df.columns = ['Gender', 'Age', 'Annual Income','Spending Score']
df.head()
```

	Gender	Age	Annual Income	Spending Score
0	Male	19	15	39
1	Male	21	15	81
2	Female	20	16	6
3	Female	23	16	77
4	Female	31	17	40

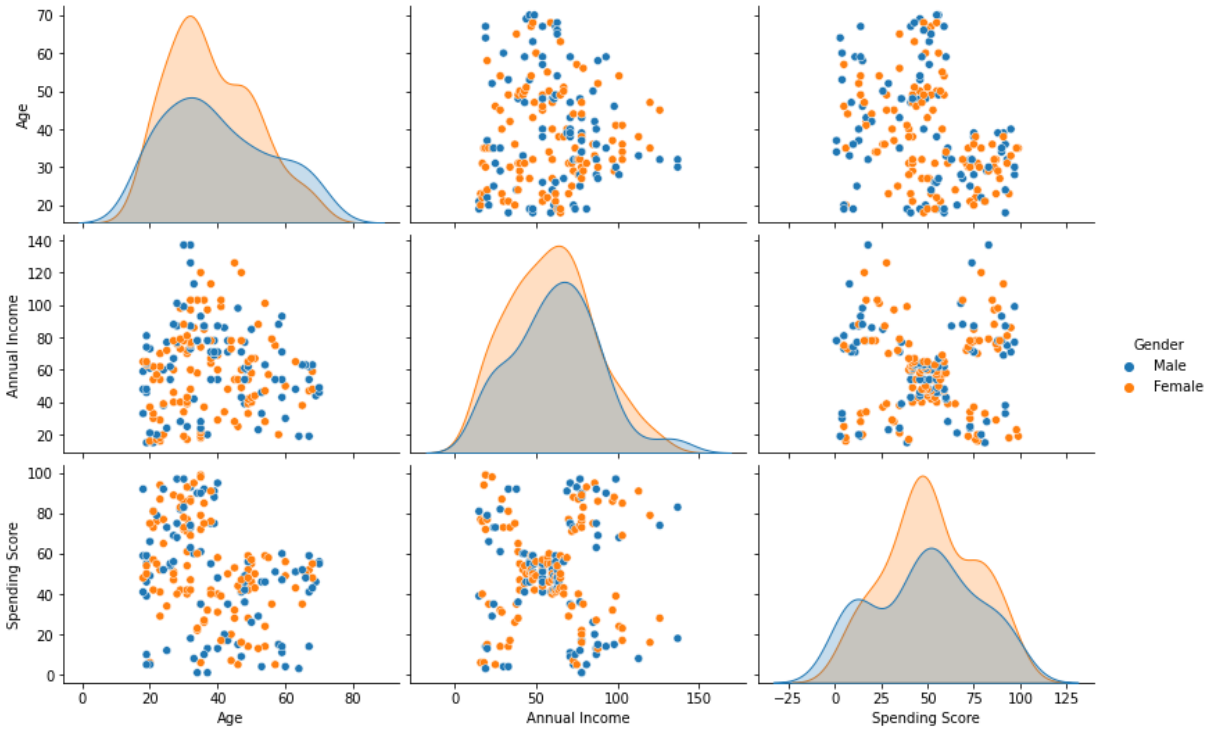
Nümerik verilerimizin ortalamalarına, min ve max değerlerine bakalım:

```
df.describe().T
```

	count	mean	std	min	25%	50%	75%	max
Age	200.0	38.85	13.969007	18.0	28.75	36.0	49.0	70.0
Annual Income	200.0	60.56	26.264721	15.0	41.50	61.5	78.0	137.0
Spending Score	200.0	50.20	25.823522	1.0	34.75	50.0	73.0	99.0

Değişkenlerimizi grafik üzerinde inceleme:

```
import seaborn as sns
sns.pairplot(df, hue = 'Gender', aspect = 1.5)
```



K-Means algoritması ile verileri gruplara ayırma: Gelir ve skor değişkenlerini kullanarak bir kümeleme yapma.

```
x = df[['Annual Income', 'Spending Score']]
```

K-Means kümelemesindeki K'nın uygun değerini bulup uygulamak için Elbow Method'una bakmalıyız. Bu metot bir veri kümesindeki uygun sayıda küme bulmaya yardımcı olmak için tasarlanmış küme içi tutarlılık analizinin yorumlanması ve doğrulanması için bir yöntemdir.

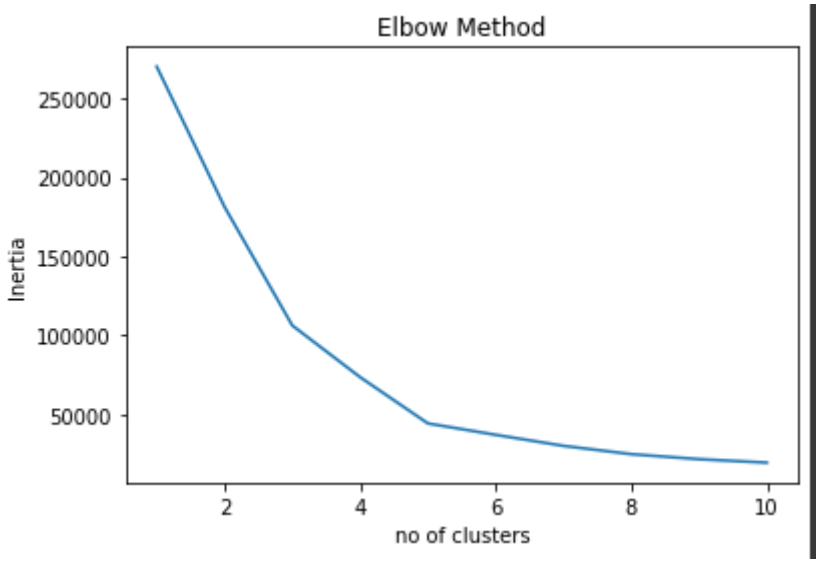
```python
from sklearn.cluster import KMeans
import matplotlib.pyplot as plt

clusters = []


for i in range(1,11):
    kmeans = KMeans(n_clusters= i, init='k-means++', random_state=0)
    kmeans.fit(X)
    clusters.append(kmeans.inertia_)


plt.plot(range(1,11),clusters)
plt.title('Elbow Method')
plt.xlabel('no of clusters')
plt.ylabel('Inertia')
plt.show()
```

Burada kullandığımız inertia da veri noktalarını kümelere ayırmak için kullanılan formüldür.



Küme sayısı 5'ten küçükse inertia'nın yüksek bir değere sahip olduğunu ancak küme sayısı 5'ten büyükse nispeten sabit olduğunu görebiliyoruz. Bu yüzden optimum küme sayısı olarak 5'i alıyoruz.

Optimum küme sayımızla kümelerimizi ayırma:

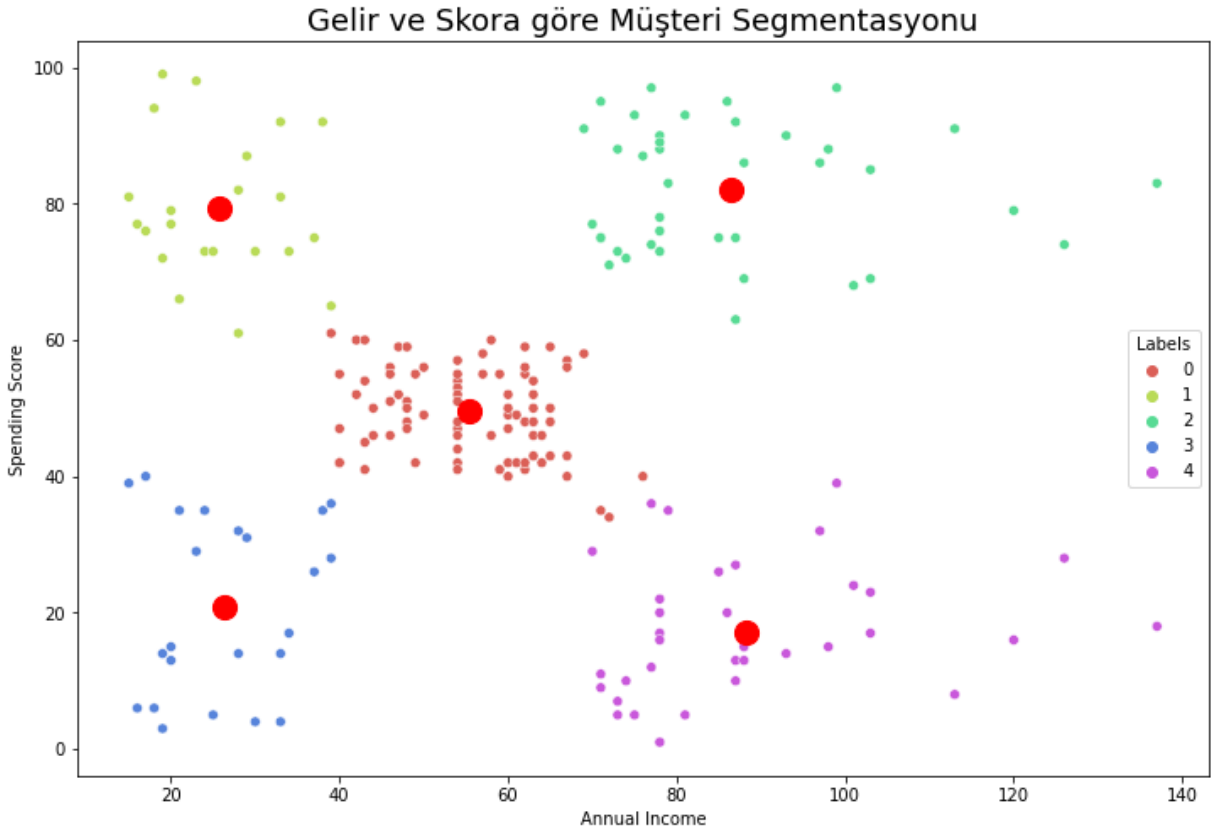
```
km_5 = KMeans(n_clusters=5, init='k-means++', random_state=0)
km_5.fit(x)
centroids = km_5.cluster_centers_
x['Labels'] = km_5.labels_


plt.figure(figsize=(12, 8))

sns.scatterplot(x['Annual Income'], x['Spending Score'], hue=x['Labels'],
                palette=sns.color_palette('hls', 5))

plt.scatter(centroids[:,0], centroids[:,1], c='red',s=200)

plt.title('Gelir ve Skora göre Müşteri Segmentasyonu',fontsize=18)
plt.show()
```



Yapılan bu uygulama sonucunda gruplara ayrılan müşterilere farklı yaklaşımlar sergileyerek davranışlarında değişikliğe yol açabiliriz. Bir grup müşteriye ihtiyaçları doğrultusunda tasarlanmış bir pazarlamanın parçası olarak kişiselleştirilmiş mesajlar gönderildiğinde, şirketlerin bu müşterilere onları daha fazla ürün satın almaya teşvik edecek özel teklifler göndermesi daha kolaydır.

KAYNAKÇA

Bilgeiş “Herkes için Yapay Zekâ II” eğitimi.

KODLUYORUZ
geleceği kodluyoruz >_

EMpower
Enriching young lives in emerging markets