

✦ **Jump-start your best year yet:** Become a member and get 25% off the first year



K-Means Algoritması



Şevket Ay · Follow

Published in Deep Learning Türkiye

7 min read · Oct 28, 2019

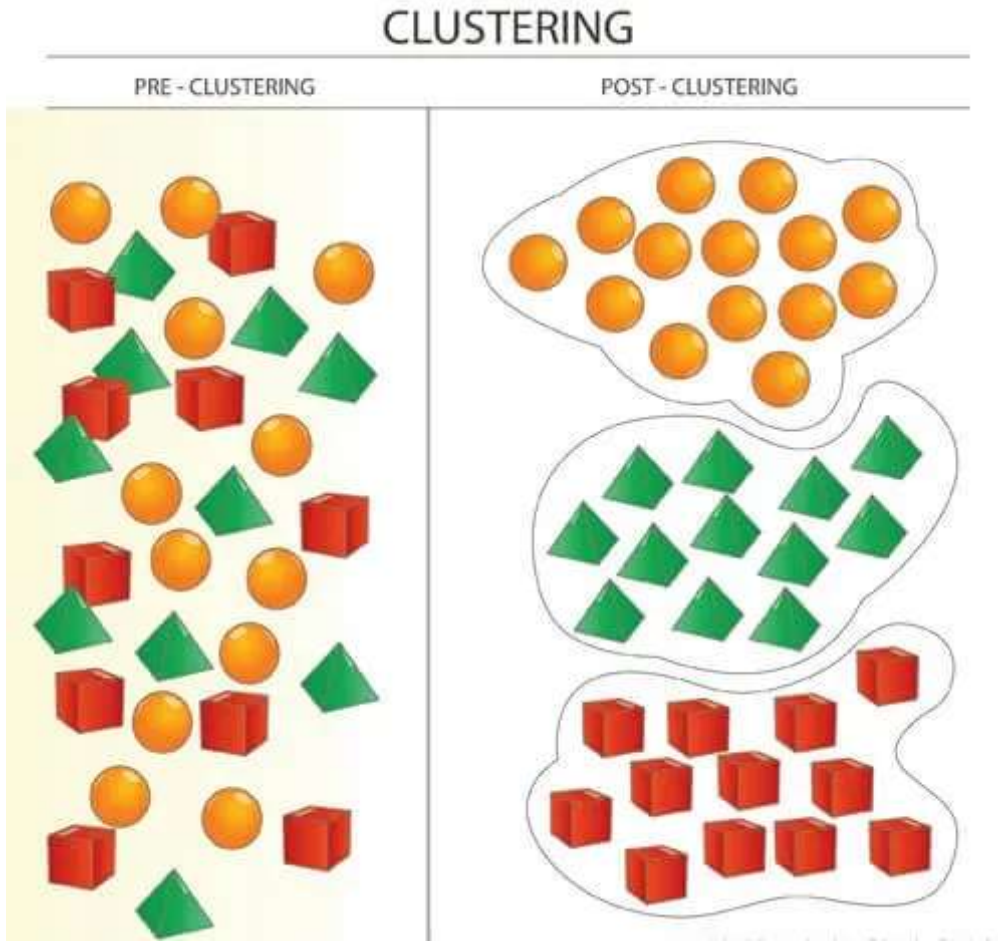


Share



More

Hiç kitaplığınızı, aynı konulara ait kitapların aynı raflarda veya aynı blokta olacak şekilde organize ettiniz mi? Muhtemelen ettiniz. Benzer nesneleri birlikte gruplamayı zaten biliyorsunuz. Fikir çok basit olsa da, bu fikirden etkilenen kullanım durumlarının miktarı muazzamdır. Makine öğrenmesi literatüründe buna genellikle kümeleme denir — *benzer nesneleri otomatik olarak aynı gruplara gruplandırır*.



Kümeleme, verilerin yapısı hakkında bir sezgiyi elde etmek için kullanılan en yaygın keşifsel veri analizi tekniklerinden biridir. Farklı kümelerdeki veri noktaları çok farklıken, aynı alt gruptaki (küme) veri noktalarının çok benzer olması nedeniyle verilerdeki alt grupların belirlenmesi görevi olarak tanımlanabilir. Bu yazıda, sadeliği nedeniyle en çok kullanılan kümeleme algoritmalarından biri olarak kabul edilen *K-means algoritmasını* ele alacağız.

K-Means'ı Nerede Uygulayabilirim?

Öncelikle bu algoritmayı nerelerde, hangi amaçlarda kullanabiliriz bunlara göz atalım. Bana göre bir algoritmayı incelemeye başlamadan önce onun neye hizmet ettiğini bilmek bizim çalışma verimimizi arttırır.

→Belge Sınıflandırması

Belgeleri etiketlere, konulara ve belgenin içeriğine göre birden fazla kategoride kümeleyin. Bu çok standart bir sınıflandırma problemidir ve k-means aracı bu amaç için oldukça uygun bir algoritmadır.

→Suç Yerlerinin Belirlenmesi

Bir şehirdeki belirli bölgelerde mevcut olan suçlarla ilgili veriler, suç kategorisi, suç

alanı ve ikisi arasındaki ilişki, bir şehirdeki ya da bölgedeki suç eğilimli alanlara ilişkin kaliteli bilgiler verebilir. [İşte bununla ilgili güzel bir makale..](#)

→Müşteri Segmentasyonu

Kümeleme, pazarlamacıların müşteri tabanını geliştirmelerine, hedef alanlarda çalışmasına ve müşterileri satın alma geçmişine, ilgi alanlarına veya etkinlik izlemeye göre segmentlere ayırmasına yardımcı olur. Sınıflandırma, şirketin belirli kampanyalar için belirli müşteri kümelerini hedeflemesine yardımcı olur.

→Oyuncu Analizi

Oyuncu istatistiklerini analiz etmek, spor dünyasının her zaman kritik bir unsuru olmuştur ve artan rekabetle birlikte, makine öğrenmenin burada oynayacağı kritik bir rol vardır. [Ayrıntılar ve örnek bir uygulama için bu makaleye göz atın.](#)

→Dolandırıcılık Tespiti

Makine öğrenimi sahtekarlık tespitinde önemli bir rol oynar ve otomobil, sağlık ve sigorta sahtekarlığı tespitinde sayısız uygulamaya sahiptir. Sahte iddialarla ilgili geçmiş verileri kullanarak, yeni iddiaları , sahte kalıpları belirten kümelere yakınlığına dayanarak izole etmek mümkündür.

→Çağrı Kaydı Detay Analizi

Bir çağrı detay kaydı (CDR), telekom şirketleri tarafından bir müşterinin araması, SMS ve internet etkinliği sırasında elde edilen bilgilerdir. Bu bilgiler, müşteri demografisiyle birlikte kullanıldığında, müşterinin ihtiyaçları hakkında daha fazla bilgi sağlar. [Bu makalede, denetlenmeyen k-means kümeleme algoritmasını kullanarak müşteri faaliyetlerini 24 saat boyunca nasıl kümelendirebileceğinizi anlayacaksınız.](#) Müşteri segmentlerini saatlerce kullanımlarına göre anlamak için kullanılır.

Open in app ↗



Search



A

sorunlara işaret ettiğinden, sonraki işlemler için önceliklendirme için manuel olarak taranmaları gerekir. Verilerin kümelenmesi, uyarı kategorileri hakkında bilgi verebilir ve ortalama onarım süresi ve arıza tahminlerinde yardımcı olabilir.

→Burada yazmayan daha birçok uygulamada (Görüntü Tanıma gibi) K-Means algoritması kullanılabilir.

Denetimsiz Öğrenmenin(Unsupervised Learning) Anlamı

K-Means algoritması bir unsupervised learning(gözetimsiz öğrenme) ve kümeleme algoritmasıdır.

Denetimsiz öğrenme, *modeli denetlemeniz gerekmeyen bir makine öğrenme tekniğidir. Bunun yerine, modelin bilgi keşfetmek için kendi başına çalışmasına izin vermeniz gerekir. Temel olarak etiketlenmemiş verilerle ilgilenir. Denetimsiz öğrenme algoritmaları, denetimli öğrenmeye kıyasla daha karmaşık işlem görevlerini gerçekleştirmenizi sağlar.*

Gelin,günlük hayatımızdan bir örnek ile Denetimsiz Öğrenmeyi pekiştirelim.

Bebek,bu köpeği tanıyor ve biliyor,onlar çok yakın iki arkadaş.



Bir gün,bir aile arkadaşı ziyarete gelirken kendi köpeğini de getirir ve bebekle oynamaya çalışır.



Bebek bu köpeği daha önce görmedi. Ancak birçok özelliğini tanıdı. (2 kulak, göz, 4 ayak üzerinde yürümek).Evcil köpeğine benziyor.Ve yeni hayvanı köpek olarak tanımlar. Bu, sizin ona öğretmediğiniz, ancak onun kendi verilerden öğrendiği (bu durumda bir köpek hakkındaki veriler), denetlenmeyen bir öğrenmedir. Bu, denetlenen bir öğrenim olsaydı, aile dostu, bebeğe köpek olduğunu söylerdi.

Denetimsiz Öğrenimi kullanmanın başlıca nedenleri

- Denetimsiz makine öğrenmesi, verilerde her türlü bilinmeyen kalıpları bulur.
- Denetimsiz yöntemler, kategorizasyon için faydalı olabilecek özellikleri bulmanıza yardımcı olur.
- Gerçek zamanlı olarak gerçekleşir, bu nedenle tüm girdi verileri öğrenilenlerin varlığında analiz edilir ve etiketlenir.
- Etiketlenmemiş verileri bilgisayardan elde etmek, elle müdahale gerektiren etiketli verilerden daha kolaydır.

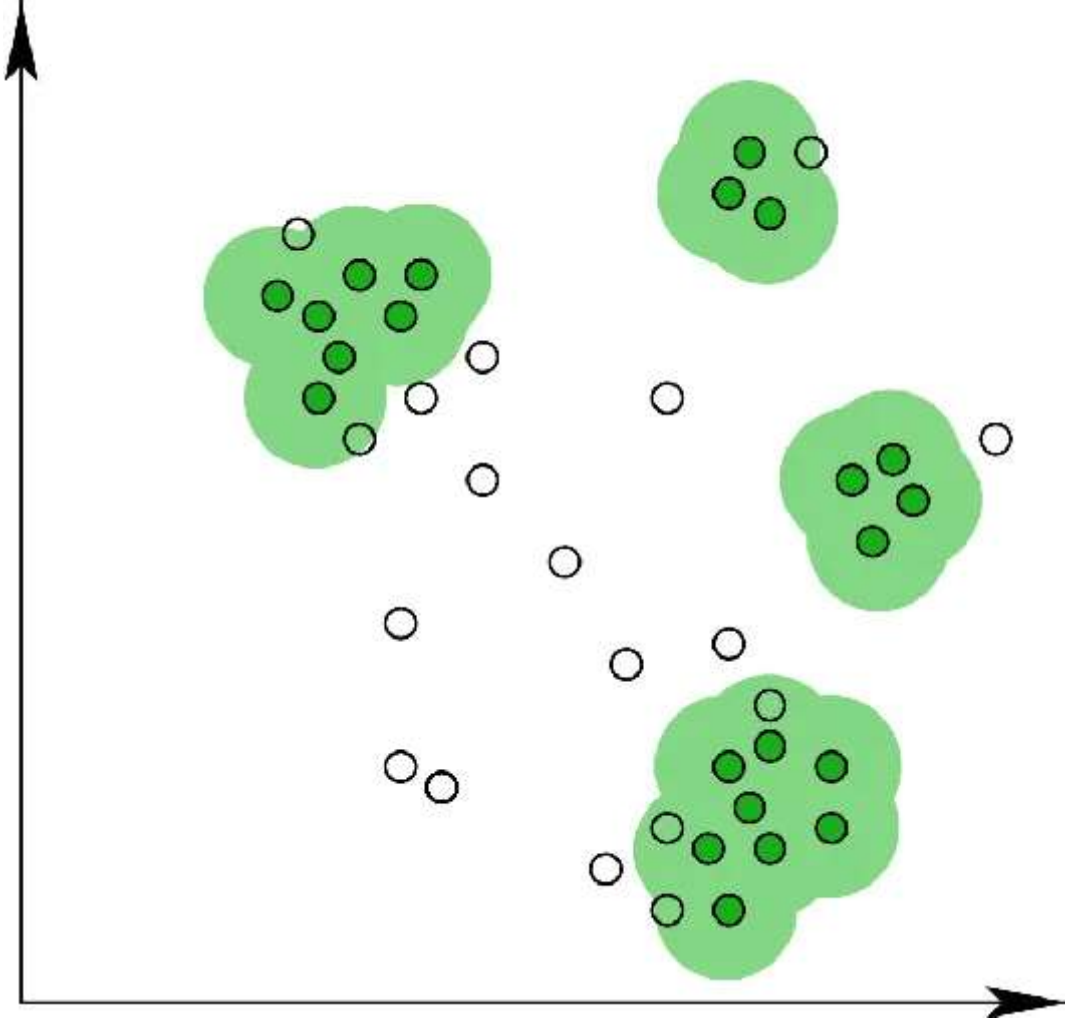
Kümeleme Yöntemleri

Kümelerin verilerden oluşturulma şekline bağlı olarak, farklı kümeleme yöntemleri olabilir. Kurumlar tarafından yoğun olarak kullanılan en popüler kümelendirme tekniklerine bir göz atalım. Bu türler:

- Bölümleme yöntemleri(Partitioning methods)

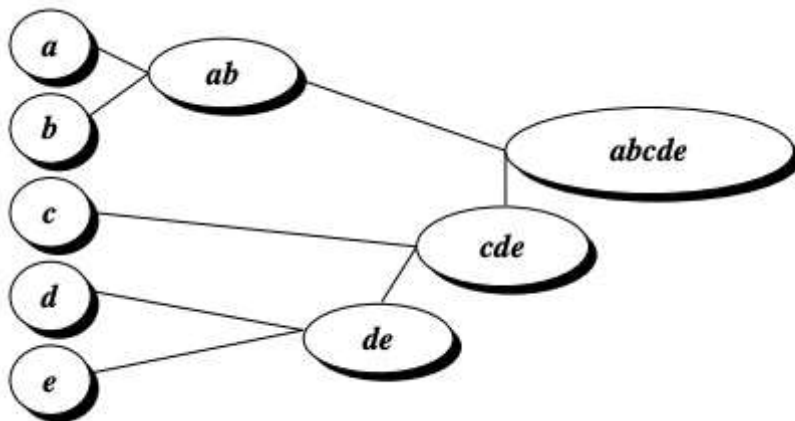
Bölüm tabanlı kümeleme yöntemleri, verilen nesneleri, n boyutlu bir düzlemde rastgele veya belirli bazı nesnelerden uzaklıklarını ölçerek kümeler. Bu nedenle, bu

yöntemler aynı zamanda mesafeye dayalı yöntemler olarak da bilinir.



- **Hiyerarşik yöntemler(Hierarchical methods)**

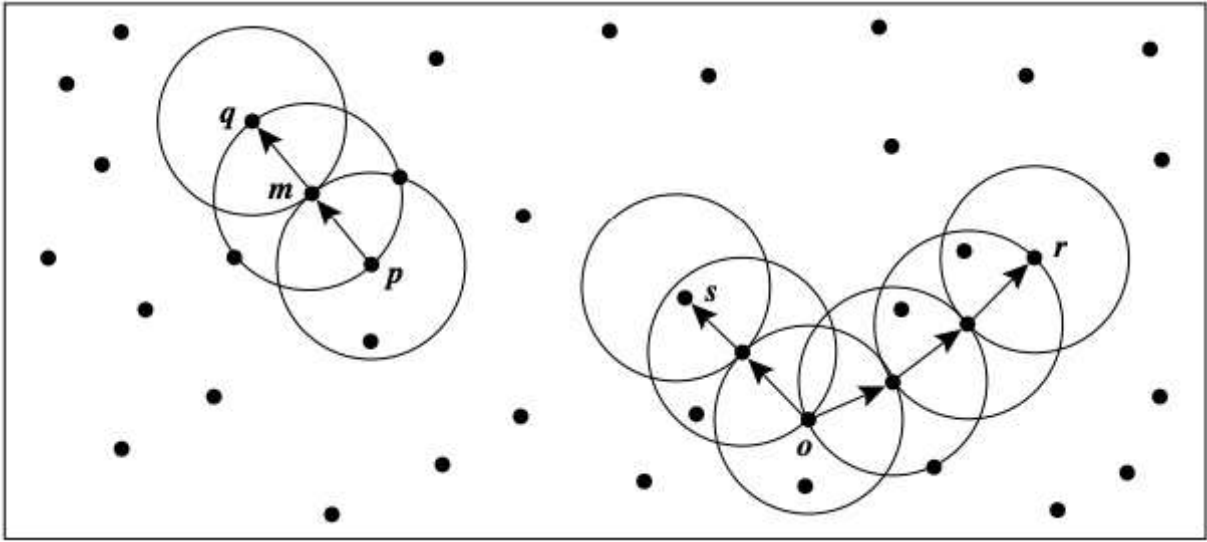
Hiyerarşik kümeleme yöntemleri, bölümlendirme yöntemlerinden farklıdır. Veri noktalarını benzerliklerine göre seviyelere / hiyerarşilere ayırırlar. Bu seviyeler birlikte ağaç benzeri bir yapı oluşturur (dendrogram). Bu bölünme iki şekilde gerçekleşebilir — yukarıdan aşağıya ve aşağıdan yukarıya.



Data Mining'den alınmıştır.

- Yoğunluk bazlı yöntemler(Density-based methods)

Veri noktalarının mesafesini göz önünde bulundurmamak yerine, yoğunluğa dayalı kümeleme yöntemlerinde, bir mahalle kümeleri oluşturduğu düşünülmektedir. Komşuluk, verilen verilerden bir küme oluşturmak için ilgilenilen bir bölgede (tipik olarak başka bir veri noktası) bulunması gereken buradaki veri noktalarının sayısını ifade eder.



Data Mining'den alınmıştır.

Artık sektörde yoğun olarak uygulanan yaygın kümelenme yöntemlerine genel bir bakış var. Şimdi Python'da K-Means kümeleme algoritmasını kullanacağımız bir problemi ele alalım.

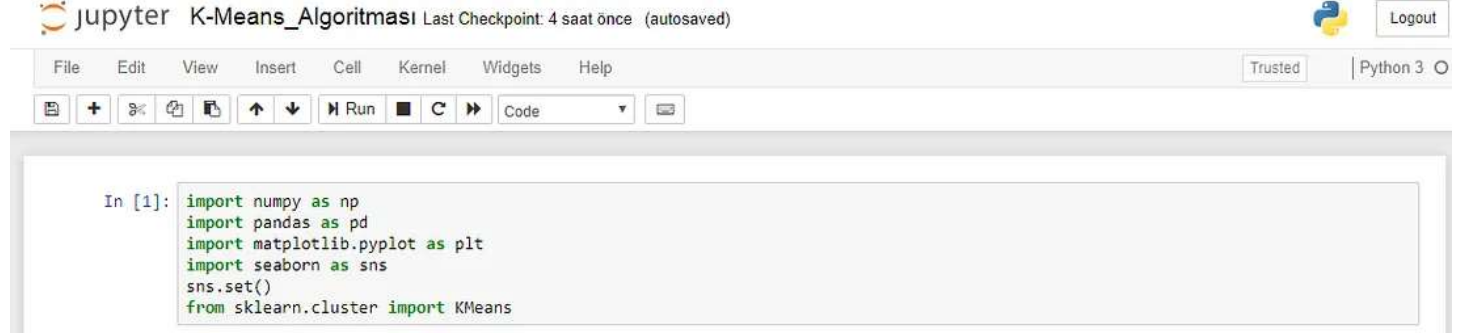
Şimdi pazar segmentasyonunun bir örneğini ele alacağız. Piyasadaki müşteri türlerini özelliklere göre analiz etmeye çalışacağız. Veri seti 30 örneklemden oluşmakta ve özellikleri sırasıyla memnuniyet ve bağlılıktır. Yani basitçe söylemek gerekirse elimizdeki müşteri verileriyle, müşterileri memnuniyet ve bağlılıklarına göre kümelendireceğiz.

#Adım adım giderek kodlarımızı yazmaya başlayalım.

#1. İlk olarak her zamanki gibi K-means Clustering'i gerçekleştirmek için gereken kütüphaneleri import edeceğiz.

- Numpy: Çok boyutlu dizilerle başa çıkmamıza yardımcı olan üçüncü taraf bir pakettir.
- Pandas: Verileri tablo şeklinde düzenlememize izin verir.

- Matplotlib: Numpy hesaplamasını görselleştirmeye yardımcı olur.
- Seaborn: Bu aynı zamanda Matplotlib'in görselleştirilmesine de katkıda bulunur.
- Scikit-learn: Bu, en yaygın kullanılan makine öğrenme kütüphanesidir.



Jupyter K-Means_Algoritması Last Checkpoint: 4 saat önce (autosaved)

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

```
In [1]: import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
sns.set()
from sklearn.cluster import KMeans
```

#2.Bu adımda pandas veri çerçevesini kullanarak değişken veri setine veri yükleyeceğiz.Müşteri memnuniyeti ve bağlılığını içeren 30 gözlem vardır. Buradaki veriler .csv formatındadır. Pandas, verileri tablo şeklinde düzenlememize yardımcı olur, bunu unutmayın.

In [2]:

```
data=pd.read_csv('segmentasyon.csv')
data
```

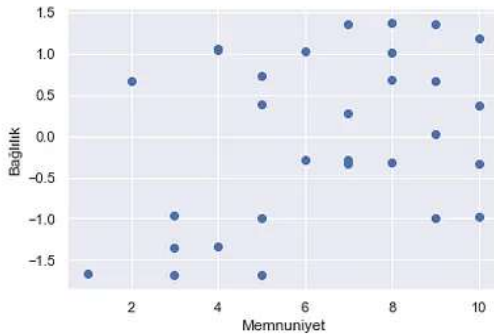
Out[2]:

	Memnuniyet	Bağlılık
0	4	-1.33
1	6	-0.28
2	5	-0.99
3	7	-0.29
4	4	1.06
5	1	-1.66
6	10	-0.97
7	8	-0.32
8	8	1.02
9	8	0.68
10	10	-0.34
11	5	0.39
12	5	-1.69
13	2	0.67
14	7	0.27
15	9	1.36
16	8	1.38
17	7	1.36
18	7	-0.34
19	9	0.67
20	10	1.18
21	3	-1.69
22	4	1.04
23	3	-0.96
24	6	1.03
25	9	-0.99
26	10	0.37
27	9	0.03
28	3	-1.36
29	5	0.73

#3.Şimdi plot komutu ile verilen verinin dağılım grafiğini çizeceğiz.

```
In [3]: plt.scatter(data['Memnuniyet'],data['Bağlılık'])
plt.xlabel('Memnuniyet')
plt.ylabel('Bağlılık')
plt.show
```

Out[3]: <function matplotlib.pyplot.show(*args, **kw)>



#4.Bu adımda ise kümeleme modelimizi oluşturuyoruz. Öncelikle sadece verilerimizin bir kopyasını oluşturuyoruz ve onu x değişkeninde saklıyoruz. Şimdi değişken k-means yaratacağız ve K-Means argümanı 2 olsun dedik.Yani 2

küme oluşturmak istediğimizi söyledik. Bu basit kod satırını k-means algoritmasını uygulayacak ve verilerimizi iki kümeye ayıracaktır.

Aşağıdaki açıklamaya bakarsak arka planda 300 adet iterasyon gerçekleştiğini görebilirsiniz.

```
In [4]: x=data.copy()

In [5]: kmeans=KMeans(2) # K=2 olsun dedik.
kmeans.fit(x) # Modelimizi oluşturduk.

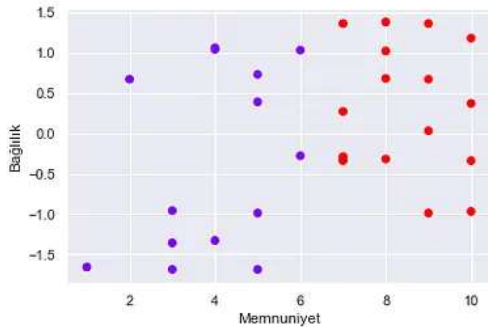
Out[5]: KMeans(algorithm='auto', copy_x=True, init='k-means++', max_iter=300,
n_clusters=2, n_init=10, n_jobs=None, precompute_distances='auto',
random_state=None, tol=0.0001, verbose=0)
```

#5.Tekrar x'in bir kopyasını oluşturacağız ve kümelerde saklayacağız. K-means algoritmamızın predict ettiği yani tahmin ettiği değere sahip olan cluster_predict adlı yeni bir sütun oluşturacağız. Daha sonra kümelenmiş verileri çizeceğiz, burada “Memnuniyet” ve “Bağlılık” olmak üzere iki parametremiz olduğunu unutmayın. İki kümeyi, biri kırmızı, diğeri mor olmak üzere ayıralım.

```
In [6]: clusters=x.copy()
clusters['cluster_pred']=kmeans.fit_predict(x) #Bir önceki satırda modelimizi oluşturmuştuk,bu satırda ise x datasına göre
#modelimizi tahminliyoruz.

In [7]: plt.scatter(clusters['Memnuniyet'],clusters['Bağlılık'],c=clusters['cluster_pred'],cmap='rainbow')
plt.xlabel('Memnuniyet')
plt.ylabel('Bağlılık')
plt.show

Out[7]: <function matplotlib.pyplot.show(*args, **kw)>
```



#6. Şimdi standartlaştırma yapmamız gerekiyor. Standartlaştırma , bizim daha sağlıklı ve doğru işlemler yapmamızı sağlar. Datamızda gördüğünüz üzere “Memnuniyet” değerleri “Bağlılık” değerlerinden çok daha büyüktür. Bu bizim algoritmamızın “Bağlılık” değerlerini görmezden gelmesine sebep olabilir,bu yüzden “Memnuniyet” değerlerini “Bağlılık” seviyesine indirmemiz gerekiyor.

```
In [8]: from sklearn import preprocessing
x_scaled=preprocessing.scale(x)
x_scaled
```

```
Out[8]: array([[ -0.93138063, -1.3318111 ],
 [ -0.15523011, -0.28117124],
 [ -0.54330537, -0.99160391],
 [  0.23284516, -0.29117733],
 [ -0.93138063,  1.05964534],
 [ -2.09560642, -1.6620122 ],
 [  1.39707095, -0.97159172],
 [  0.62092042, -0.32119561],
 [  0.62092042,  1.01962097],
 [  0.62092042,  0.67941378],
 [  1.39707095, -0.3412078 ],
 [ -0.54330537,  0.38923705],
 [ -0.54330537, -1.69203048],
 [ -1.70753116,  0.66940768],
 [  0.23284516,  0.26916393],
 [  1.00899568,  1.35982816],
 [  0.62092042,  1.37984035],
 [  0.23284516,  1.35982816],
 [  0.23284516, -0.3412078 ],
 [  1.00899568,  0.66940768],
 [  1.39707095,  1.17971847],
 [ -1.31945589, -1.69203048],
 [ -0.93138063,  1.03963316],
 [ -1.31945589, -0.96158562],
 [ -0.15523011,  1.02962706],
 [  1.00899568, -0.99160391],
 [  1.39707095,  0.36922486],
 [  1.00899568,  0.02901767],
 [ -1.31945589, -1.36182938],
 [ -0.54330537,  0.72944425]])
```

STANDARTLAŞTIRMA

Veriler sıfır ortalama civarında ölçeklenir. Şimdi her iki verinin de eşit şekilde ölçeklendiğini ve şimdi ikisinin de özellik olarak seçilme şansının eşit olacağını görebiliyoruz.

#7.Dirsek Yöntemi

Neden sadece 2 küme ile k-means başlattığımızı hiç merak ettiniz mi? Herhangi bir sayıda kümeye sahip olabilmeyi isteyip herhangi bir değerle başlatabilirdik. Ancak çok sayıda küme olduğu zaman analizler zorlaşmaktadır. Öyleyse, başlatılacak küme sayısını tam olarak nasıl bileceğiz? Elinizdeki problemlere göre K değeri değişeceği için kesin bir rakam söylenemez. Farklı problemlerde farklı değerler çıkacaktır.

```
In [9]: wcss=[] #Liste oluşturduk.

for i in range(1,30): #30 adet örneklememiz olduğu için 1'den 30'a kadar gitmesini sağlıyoruz.
    kmeans=KMeans(i)
    kmeans.fit(x_scaled) #standartlaştırdığımız datamıza göre fit et,yani modelimizi oluştur.
    wcss.append(kmeans.inertia_) #inertia ne demek? Her bir k değeri için wcss değerini bul.

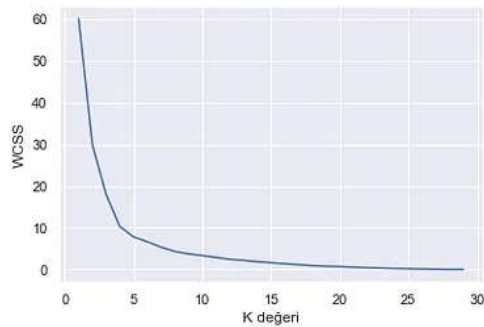
wcss
```

```
Out[9]: [60.0,
29.818973034723147,
17.913349527387965,
10.247181805928422,
7.792695153937187,
6.634898754940157,
5.326631124753926,
4.315662464700005,
3.731185392455718,
3.3238719798946406,
2.872671463648353,
2.4009836471184305,
2.1888663386948832,
1.8347282440516792,
1.6227677874312503,
1.3520410600630661,
1.146435576453863,
0.9061429801691203,
0.7698508420567366,
0.6800955326053162,
0.5451582152303865,
0.42313027513905704,
0.3227119817275011,
0.24721053307798668,
0.17170908442847233,
0.11383861748989679,
0.0559681505513213,
0.0014517677692203244,
0.00020024383023728806]
```

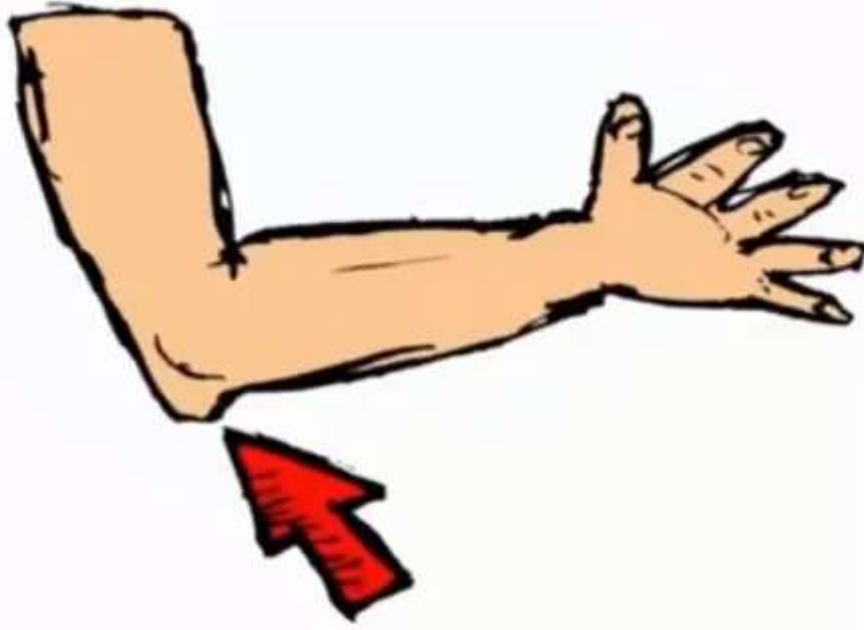
Şimdi de görselleştirelim..

```
In [10]: plt.plot(range(1,30),wcss)
plt.xlabel('K değeri')
plt.ylabel('WCSS')
plt.show
```

```
Out[10]: <function matplotlib.pyplot.show(*args, **kw)>
```



Elbow Technique



Burada dirsek yöntemi, kümelere nasıl ihtiyaç duyacağımız konusunda kafamız karıştığında kullanışlı olur. Grafiğimiz dirseğe benziyor ve bu dirsek noktasını belirlemek zorundayız.

Burada dirsek noktası 4 civarındadır ve bu, seçmemiz gereken yukarıdaki veriler için optimal küme sayımızdır.

Küme sayısını arttırmaya devam ettiğimizde 4'ten sonra dikkatle bakarsak wcss'lerde büyük bir değişiklik olmaz ve sabit kalır.

#8.Sorunumuz için optimal sayıda kümeye sahibiz.

Şimdi 4 sayıdaki yeni kümelenelemlerle birlikte kümelemeleri hızla gerçekleştireceğiz ve daha sonra bazı analizlere geçeceğiz.

```
In [11]:
kmeans_new=KMeans(4)
kmeans_new.fit(x_scaled)
cluster_new=x.copy()
cluster_new['cluster_pred']=kmeans_new.fit_predict(x_scaled)
cluster_new
```

Out[11]:

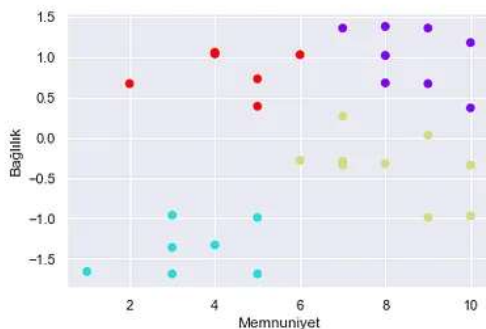
	Memnuniyet	Bağlılık	cluster_pred
0	4	-1.33	1
1	6	-0.28	2
2	5	-0.99	1
3	7	-0.29	2
4	4	1.06	3
5	1	-1.66	1
6	10	-0.97	2
7	8	-0.32	2
8	8	1.02	0
9	8	0.68	0
10	10	-0.34	2
11	5	0.39	3
12	5	-1.69	1
13	2	0.67	3
14	7	0.27	2
15	9	1.36	0
16	6	1.38	0
17	7	1.36	0
18	7	-0.34	2
19	9	0.67	0
20	10	1.18	0
21	3	-1.69	1
22	4	1.04	3
23	3	-0.96	1
24	6	1.03	3
25	9	-0.99	2
26	10	0.37	0
27	9	0.03	2
28	3	-1.36	1
29	5	0.73	3

Bu, 4 kümeyle kümelemeyi gerçekleştiren basit bir koddur. Burada dört küme vardır, bu yüzden tüm verilerimiz 0,1,2 veya 3 olarak sınıflandırılır.

#9.Yeni kümelerin çizilmesi

```
In [12]: plt.scatter(cluster_new['Memnuniyet'],cluster_new['Bağlılık'],c=cluster_new['cluster_pred'],cmap='rainbow')
plt.xlabel('Memnuniyet')
plt.ylabel('Bağlılık')
plt.show
```

Out[12]: <function matplotlib.pyplot.show(*args, **kw)>



#10.Analiz

1. Mavi noktalar, daha az memnun ve daha az bağlı olan ve bu nedenle yabancılaşmış olarak adlandırılabilirler insanlardır.
2. Kırmızı noktalar, bağlılığı yüksek ve memnuniyeti az olan insanlardır.
3. Mor noktalar yüksek bağlılık ve memnuniyeti olan insanlardır.
4. Açık yeşil noktalar, yüksek memnuniyet fakat orta derece bağlılıkta olan insanlardır.

Bir kurumun amacı daha fazla insanı mor nokta seviyesine getirmektir. Özellikle mavi noktalardaki insanların şikayetlerini analiz ve tespit edip düzeltebilirler. Böylece müşteri memnuniyeti ve bağlılığı artmış olur.

Evet,böylece yazımızın sonuna geldik. Bu yazımız da ;

- Kümeleme Nedir?Başlıca kullanılan kümeleme yöntemleri nelerdir?
- K-Means algoritması hangi alanlarda kullanılır? Unsupervised learning nedir ve neden kullanılır?
- Ve son olarak K-Means Algoritması ile bir uygulama gerçekleştirerek bazı incelikleri öğrenmiş olduk.

Umarım size faydalı bir yazı hazırlayabilmişimdir. Elimden geldiğince sade ve adım adım açıklamaya çalıştım. Herkese iyi çalışmalar diliyorum.

Yararlanılan Kaynaklar :

→<https://arxiv.org/ftp/arxiv/papers/1002/1002.2425.pdf>

→ <https://dzone.com/articles/10-interesting-use-cases-for-the-k-means-algorithm>

→ <https://lagunita.stanford.edu/c4x/HumanitiesScience/StatLearning/asset/unsupervised.pdf>

K Means

Clustering

Veri Bilimi



Follow

Written by Şevket Ay

358 Followers · Writer for Deep Learning Türkiye

<https://www.linkedin.com/in/sevketay>

More from Şevket Ay and Deep Learning Türkiye

Şevket Ay in Deep Learning Türkiye

Model Performansını Değerlendirmek—Metrikler

Bu yazıda, makine öğrenmesi modelimizi nasıl değerlendirebileceğimizi konuşacağız. Önce regresyon modellerimizi değerlendirmeyi daha sonra...

9 min read · Apr 30, 2020



1





Necmettin Çarkacı in Deep Learning Türkiye

Derin Öğrenme Uygulamalarında En Sık kullanılan Hiper-parametreler

Derin Öğrenme Uygulamalarında En Sık kullanılan Hiper-parametreler ve Bu Parametrelerin Bazı Temel Özellikleri.

14 min read · Jan 22, 2018



--



13





Yiğit Mesci in Deep Learning Türkiye

YOLO Algoritmasını Anlamak

Son yıllarda nesne tespiti alanında revaçta olan YOLO (You Only Look Once) algoritmasını ister istemez duymuşsunuzdur. Bu algoritma neden...

7 min read · Apr 29, 2019



--



4



...



Şevket Ay in Deep Learning Türkiye

Ensemble Learning—Bagging ve Boosting

Diyelim ki yeni bir telefon alacaksınız.Ama hangi model ve ya marka alacağınıza dair bir fikriniz yok. Mağazaya gittiniz ,mağazada çalışan...

8 min read · Dec 16, 2019



--



...

[See all from Şevket Ay](#)[See all from Deep Learning Türkiye](#)

Recommended from Medium



Gamze Deniz

Implementing k-means Clustering from Scratch

In my master's program for artificial intelligence and robotics, one of the five projects assigned in my machine learning course is to...

7 min read · Nov 22, 2023



--





Nirmal Sankalana

K-means Clustering: Choosing Optimal K, Process, and Evaluation Methods

In today's data-driven world, businesses and researchers encounter a huge amount of information from various sources. Extracting valuable...

16 min read · Sep 19, 2023



--



...

Lists



Practical Guides to Machine Learning

10 stories · 956 saves



data science and AI

39 stories · 49 saves



Tanner Overcash

From Points to Clusters: Spatial Clustering

Overview of Algorithms (K-means, K-medoids, DBSCAN) and Clustering Evaluation with Examples in Python

13 min read · Jan 10



--



...



Tahera Firdose

Comparing Hierarchical, K-Means, and DBSCAN Clustering Algorithms in Python

Hierarchical Clustering

✦ · 4 min read · Dec 4, 2023



--



Nur Hasna Amatullah

K-Means Clustering with PSO using RapidMiner

Optimization algorithm

3 min read · Sep 29, 2023



--





Jyoti Kapoor

Algorithm: K-Means Clustering

Example:

3 min read · Aug 17, 2023



--



See more recommendations