TEORİ VE UYGULAMADA MAKİNE ÖĞRENMESİ

Doç. Dr. Serkan Savaş

K-Means Algoritması

K-Ortalama Kümeleme Algoritması, örnek uygulama üzerinden açıklanmıştır.

Örnek: Iris veriseti ile kümeleme

K – Ortalama Kümeleme Algoritması

K-Ortalama, 1967'de James MacQueen tarafından tanıtılmıştır. Bu alanda pek çok çalışma yapıldığı görülmektedir.

K-ortalama kümeleme, en çok kullanılan kümeleme algoritmasıdır. Bilgi alma, bilgisayarla görme ve örüntü tanıma dahil olmak üzere çok sayıda alanda kullanılmaktadır.

K-ortalama kümeleme, karşılaştırılabilir veri noktalarını gruplamak için n veri noktasını k kümeye göndermektedir.

Her düğümü en yakın ağırlık merkezi ile kümeye tahsis eden yinelemeli bir süreçtir. Ardından, değerlerinin ortalamasını alarak grubun ağırlık merkezini yeniden hesaplamaktadır.

K – Ortalama Kümeleme Algoritması

K-ortalama kümelemesinin temel adımları:

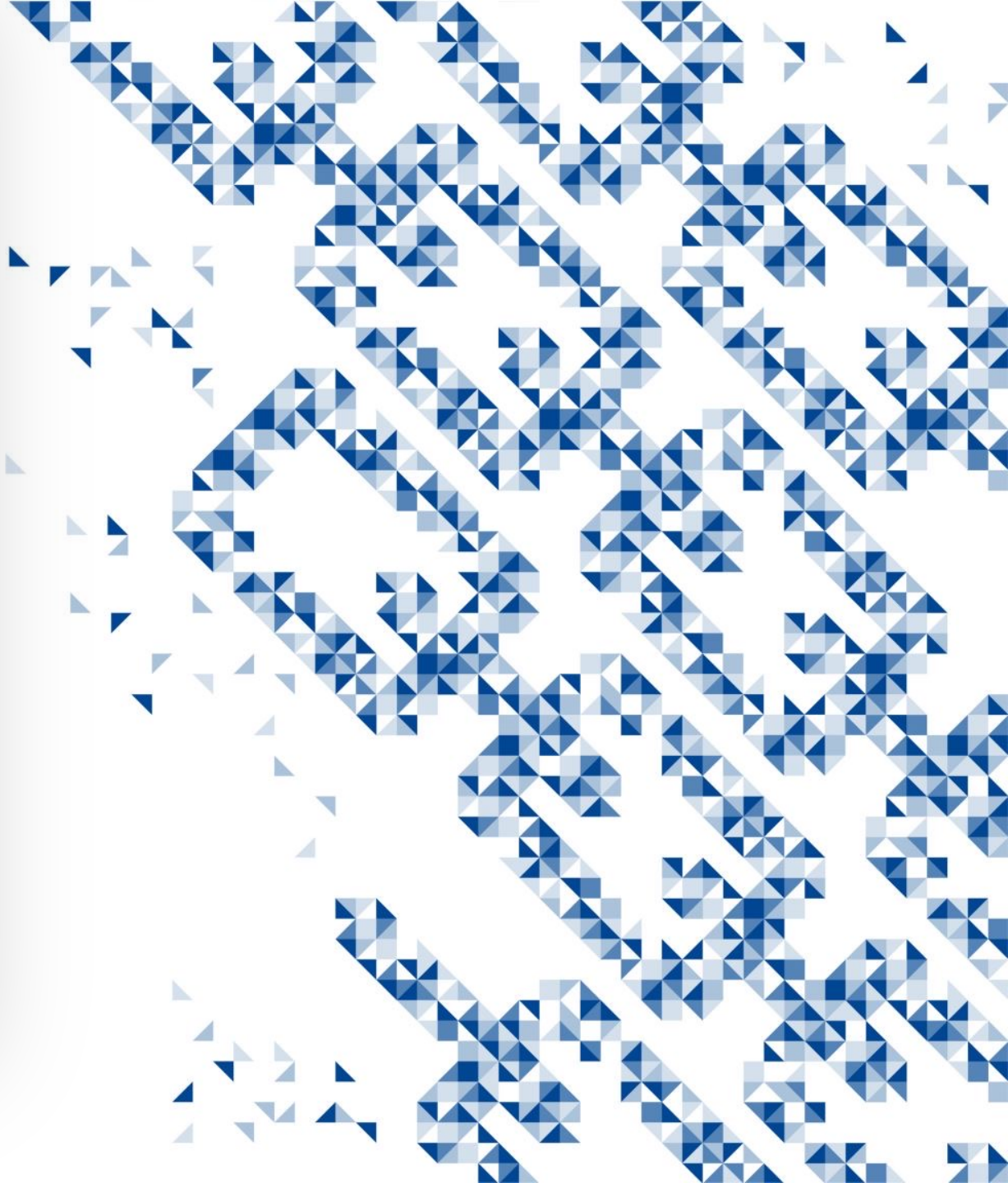
1. Kaç küme (k) istediğinize karar verin,
2. K kümenin her birine rastgele bir merkez atayın,
3. Tüm noktaların k merkezinin her birine olan mesafesini hesaplayın,
4. Noktaları en yakın merkeze atayın,
5. Her kümedeki tüm noktaların ortalamasını alarak merkezin yeni konumunu bulun,
6. Merkezler pozisyon değiştirmeyene kadar 3-5 arasındaki adımları tekrarlayın.

K – Ortalama Kümeleme Algoritması

Kümeleme hedefiyle ilgili olarak, gruplamak istediğimiz bir dizi etiketlenmemiş veri noktamız vardır. Bu gruplara genellikle algoritma tarafından (0, 1,...) sayılar atanır.

Dolaysıyla, burada bir kararla belirlenen gruplar arasında bir ayrım çizgisi olmalıdır. Çünkü algoritma, bir veri noktasını ilgili bir küme ile ilişkilendirerek çalışır ve bu nedenle karar sınırının önemli olduğu yer burasıdır.

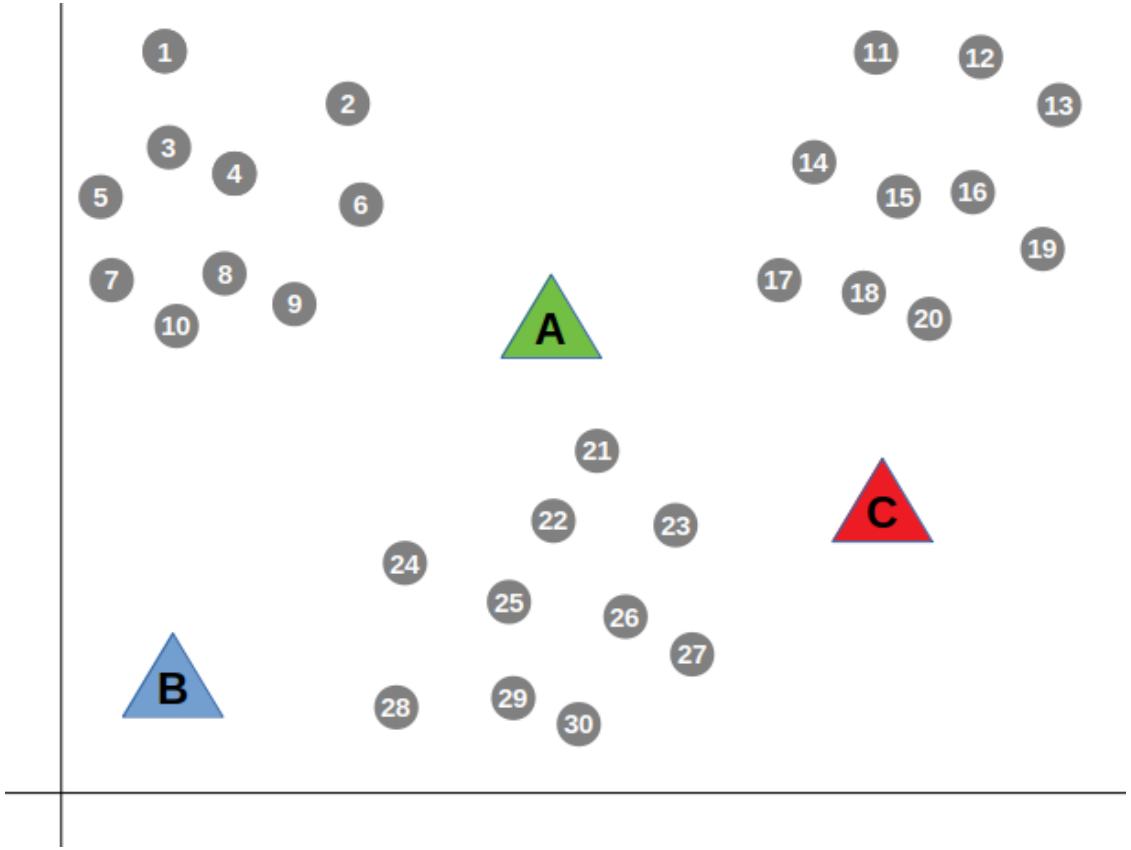
K-ortalama kümeleme, sahip olduğumuz eğitim noktalarını kümelere yerleştirmekle ilgilidir.



K – Ortalama Kümeleme Algoritması

Algoritma, değerleri ya rasgele başlatılmış ya da düzlemdeki gerçek veri noktalarına ayarlanmış **k** farklı ortalama yerleştirerek başlatılmaktadır.

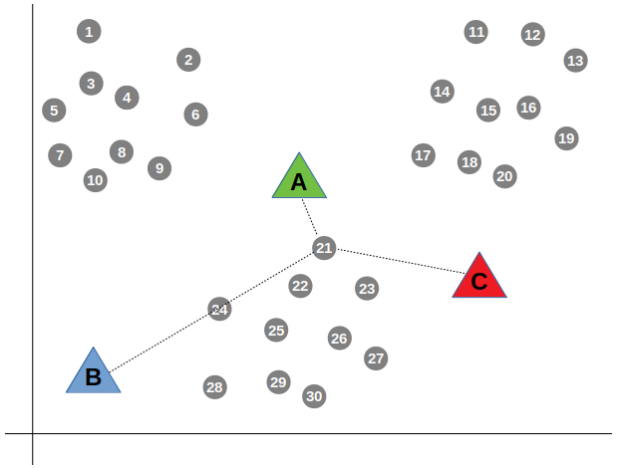
Yandaki şekilde veriler üç gruba ayrıldığı için k=3 ile başlamaktadır. Şekilde, ortalamaların değerlerinin (yani konumlarını) rastgele atandığı görülmektedir.



K – Ortalama Kümeleme Algoritması

Sonra, algoritma veri noktalarından birer birer geçer ve her nokta ile üç merkez (A, B ve C) arasındaki mesafeyi ölçer.

Algoritma daha sonra veri noktasını en yakın merkez noktasına (yani en yakın mesafeye) atayarak gruplandırır.



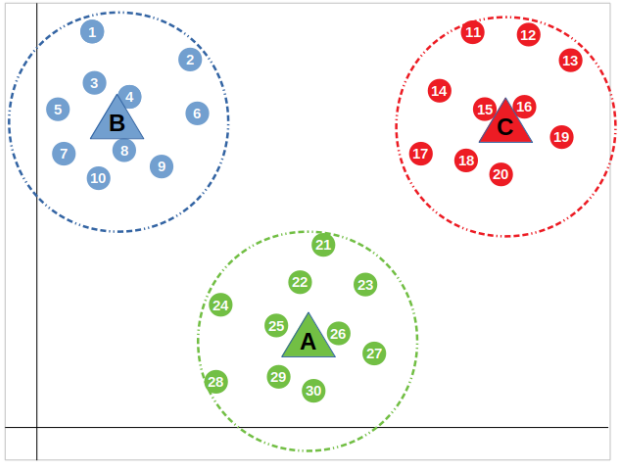
21 numaralı veri noktası, merkez A'ya daha yakın olduğu için yeşil renkte A grubuna ait olacaktır.

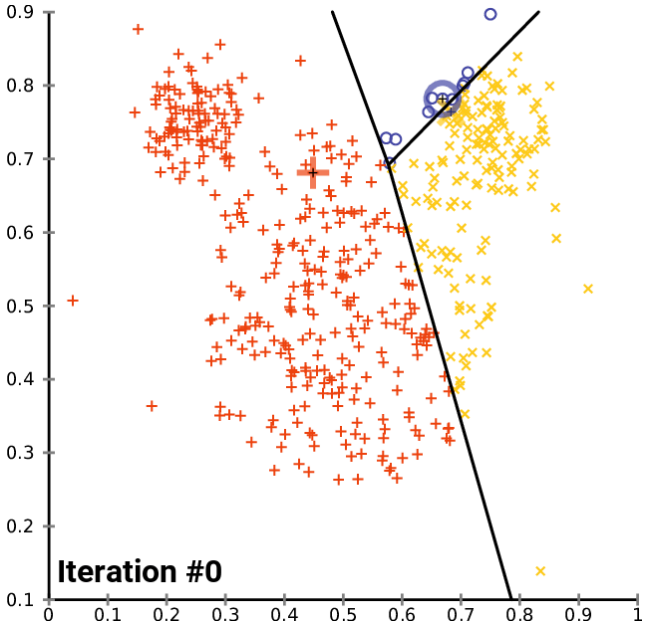
K – Ortalama Kümeleme Algoritması

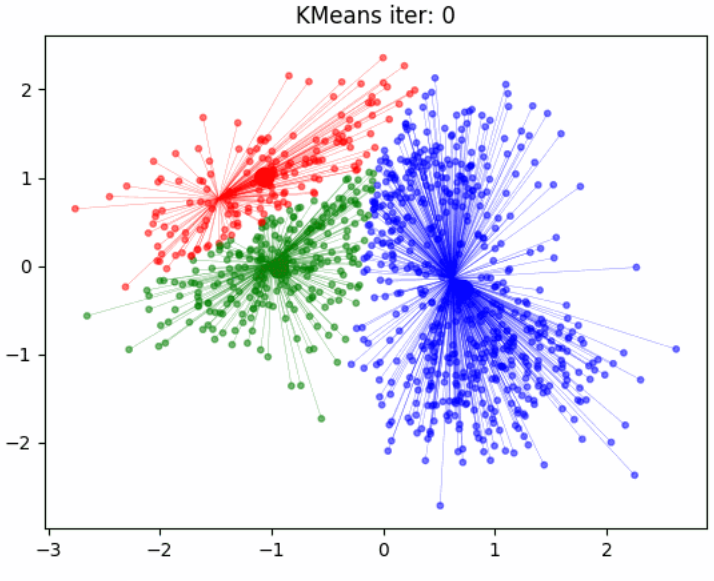
Her bir veri noktasını en yakın ağırlık merkeziyle ilişkilendirmeyi bitirir bitirmez, ortalamaları yeniden hesaplanır.

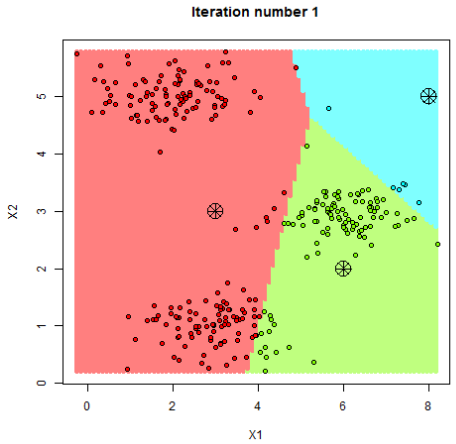
Ağırlık merkezlerinin değerleri ise; bir ağırlık merkezinin yeni değeri, o merkeze ait tüm noktaların toplamının, gruptaki noktaların sayısına bölümüdür.

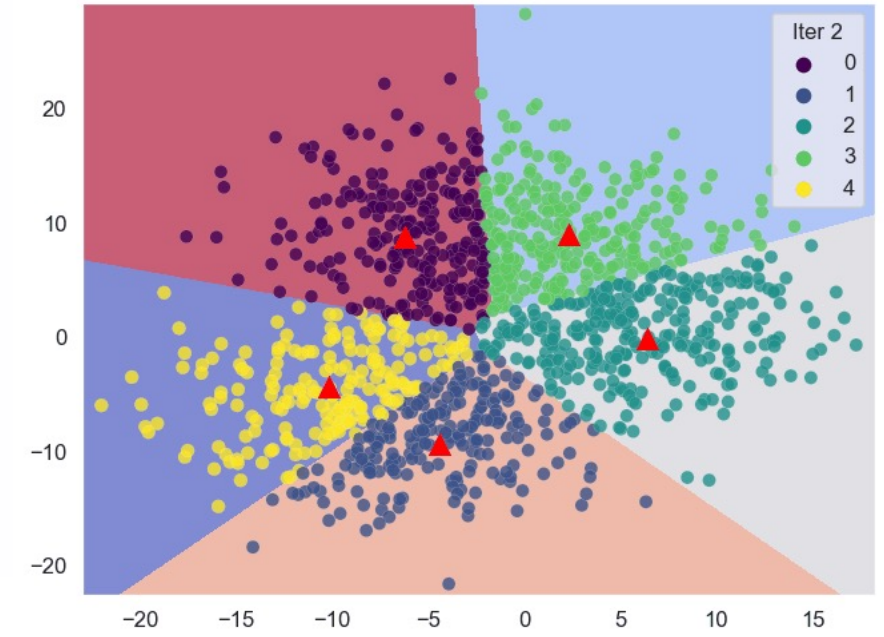
Yeniden hesaplama hiçbir ağırlık merkezi değerini değiştirmeyene kadar yukarıdaki adımlar devam edilir. Bu işlem, her bir merkezin, kendi dairesel karar sınırı ile çevrili olan kümesinin ortasında kendisini merkezlediği anlamına gelir.




Iteration #0

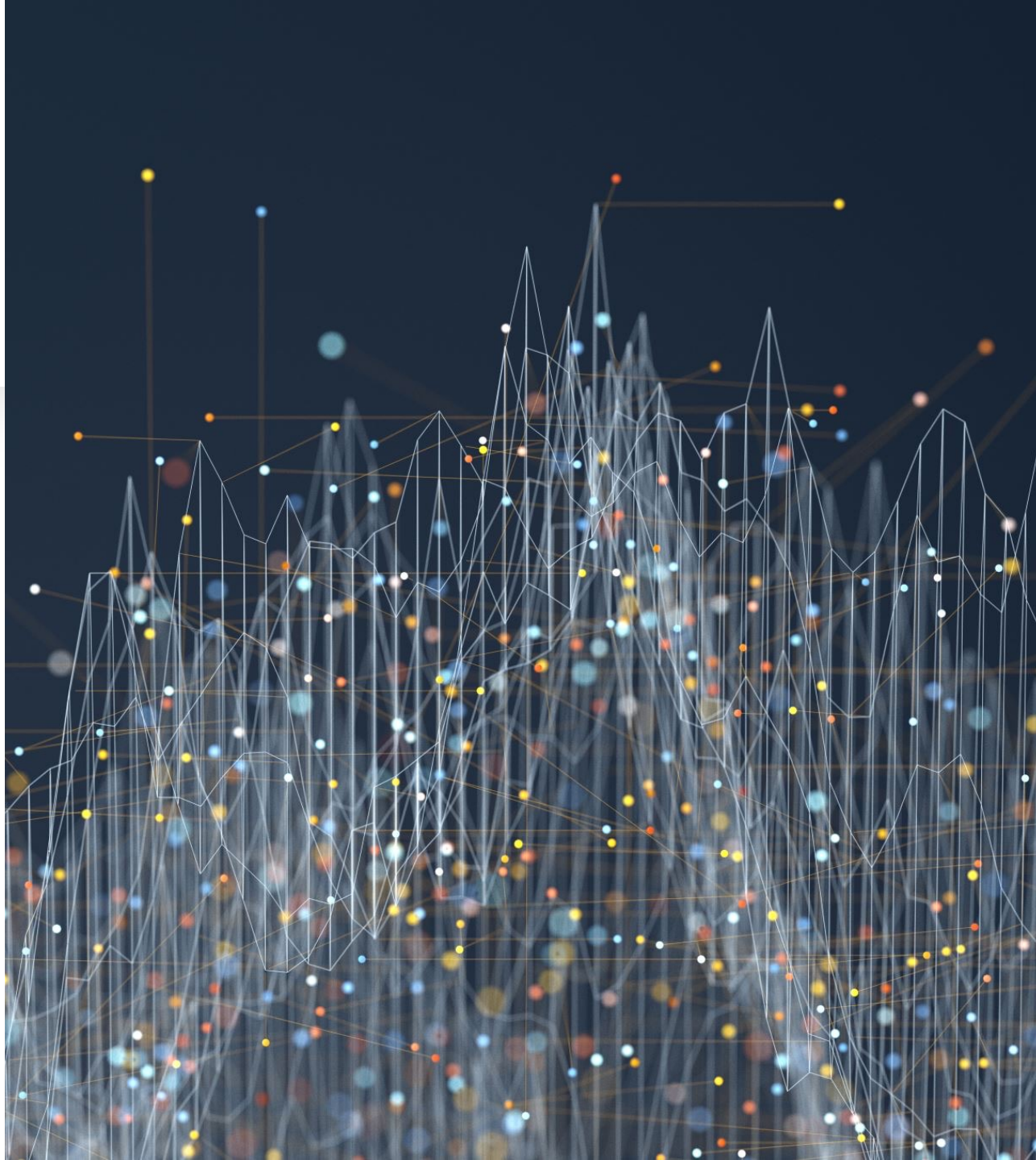

KMeans iter: 0


Iteration number 1
X1
X2


Iter 2
0
1
2
3
4

K-Ortalama Kümeleme Algoritmasının Kullanım Avantajları

- Uygulaması basittir.
- Büyük bir veri kümesine ölçeklenebilir ve ayrıca büyük veri kümelerine hızlıdır.
- Farklı şekil ve boyutlar için kümelerin genelleştirilmesi mümkündür.



K-Ortalama Kümeleme Algoritmasının Kısıtlamaları

- Aykırı değerlere karşı hassastır.
- K değerlerini manuel olarak seçmek zor bir işlemdir.
- Boyut sayısı arttıkça ölçeklenebilirliği azalır.

Kullanım Alanları

- Kanser ile ilgili Verileri Tanımlamada Kümeleme Algoritması
- Arama Motorlarında Kümeleme Algoritması
- Akademisyenlerde Kümeleme Algoritması
- Kablosuz Sensör Ağı Tabanlı Uygulamada Kümeleme Algoritması
- …

Örnek Uygulama

- from sklearn.cluster import KMeans
- from sklearn import datasets
- from sklearn.utils import shuffle
- from matplotlib import pyplot as plt

import some data to play with
- iris = datasets.load_iris()
- X = iris.data
- y = iris.target
- names = iris.feature_names
- X, y = shuffle(X, y, random_state=42)

Örnek Uygulama

```
model = KMeans(n_clusters=3, random_state=42)
iris_kmeans = model.fit(X)

from sklearn.metrics import confusion_matrix
conf_matrix=confusion_matrix(y, iris_kmeans.labels_)

fig, ax = plt.subplots(figsize=(7.5, 7.5))
ax.matshow(conf_matrix, cmap=plt.cm.Blues, alpha=0.3)
for i in range(conf_matrix.shape[0]):
    for j in range(conf_matrix.shape[1]):
        ax.text(x=j, y=i,s=conf_matrix[i, j], va='center',
                ha='center', size='xx-large')

plt.xlabel('Predictions', fontsize=18)
plt.ylabel('Actuals', fontsize=18)
plt.title('Confusion Matrix', fontsize=18)
plt.show()
```

Örnek Uygulama

- iris_kmeans.cluster_centers_

- fig = plt.figure(figsize=(20, 10))
- ax1 = fig.add_subplot(1, 2, 1, projection='3d')
- ax1.scatter(X[:, 3], X[:, 0], X[:, 2],
- c=iris_kmeans.labels_.astype(float),
- edgecolor="k", s=150)
- ax1.view_init(20, -50)
- ax1.set_xlabel(names[3], fontsize=12)
- ax1.set_ylabel(names[0], fontsize=12)
- ax1.set_zlabel(names[2], fontsize=12)
- ax1.set_title("K-Means Clusters for the Iris Dataset", fontsize=12)

- ax2 = fig.add_subplot(1, 2, 2, projection='3d')

Örnek Uygulama

```
for label, name in enumerate(['virginica','setosa','versicolor']):
    ax2.text3D(
        X[y == label, 3].mean(),
        X[y == label, 0].mean(),
        X[y == label, 2].mean() + 2,
        name,
        horizontalalignment="center",
        bbox=dict(alpha=0.2, edgecolor="w", facecolor="w"),)

ax2.scatter(X[:, 3], X[:, 0], X[:, 2],
            c=y, edgecolor="k", s=150)
ax2.view_init(20, -50)
ax2.set_xlabel(names[3], fontsize=12)
ax2.set_ylabel(names[0], fontsize=12)
ax2.set_zlabel(names[2], fontsize=12)
ax2.set_title("Actual Labels for the Iris Dataset", fontsize=12)
fig.show()
```

Kaynaklar

- Ahmed, M. (2019). Data summarization: a survey. *Knowledge and Information Systems*, *58*(2), 249–273. doi:10.1007/s10115-018-1183-0
- Akkaya, K., Senel, F. ve McLaughlan, B. (2009). Clustering of wireless sensor and actor networks based on sensor distribution and connectivity. *Journal of Parallel and Distributed Computing*, *69*(6), 573–587. doi:10.1016/j.jpdc.2009.02.004
- Alsabti, K., Ranka, S. ve Singh, V. (1998). An Efficient k-means Clustering Algorithm. *IPPS: 11th International Parallel Processing Symposium* içinde .
- Estlick, M., Leeser, M., Theiler, J. ve Szymanski, J. J. (2001). Algorithmic transformations in the implementation of K-means clustering on reconfigurable hardware. *ACM/SIGDA International Symposium on Field Programmable Gate Arrays - FPGA* içinde (ss. 103–110). Association for Computing Machinery (ACM). doi:10.1145/360276.360311
- Girolami, M. ve He, C. (2003). Probability density estimation from optimally condensed data samples. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *25*(10), 1253–1264. doi:10.1109/TPAMI.2003.1233899
- Kijsipongse, E. ve U-Ruekolan, S. (2012). Dynamic load balancing on GPU clusters for large-scale K-Means clustering. *JCSSE 2012 - 9th International Joint Conference on Computer Science and Software Engineering* içinde (ss. 346–350). doi:10.1109/JCSSE.2012.6261977
- Liu, T., Rosenberg, C. ve Rowley, H. A. (2007). Clustering billions of images with large scale nearest neighbor search. *Proceedings - IEEE Workshop on Applications of Computer Vision, WACV 2007* içinde . doi:10.1109/WACV.2007.18
- MacQueen, J. (1967). Classification and Analysis of Multivariate Observations. *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability* içinde (ss. 281–297).
- Nigam, K., Mccallum, A. K., Thrun, S. ve Mitchell, T. (2000). Text classification from labeled and unlabeled documents using EM. *Machine Learning*, *39*(2), 103–134. doi:10.1023/a:1007692713085
- Oyelade, O. J., Oladipupo, O. O. ve Obagbuwa, I. C. (2010). Application of k Means Clustering algorithm for prediction of Students Academic Performance. *arxiv.org*, *7*(1). https://arxiv.org/abs/1002.2425 adresinden erişildi.
- Saurabh, A. ve Naik, A. (2011). Wireless sensor network based adaptive landmine detection algorithm. *ICECT 2011 - 2011 3rd International Conference on Electronics Computer Technology* içinde (C. 1, ss. 220–224). doi:10.1109/ICECTECH.2011.5941593
- Wang, X.-Y. ve Garibaldi, J. (2005). A comparison of fuzzy and non-fuzzy clustering techniques in cancer diagnosis. *Proceedings of the 2nd International Conference in Computational Intelligence in Medicine and Healthcare, BIOPATTERN Conference, Costa da Caparica, Lisbon, Portugal*, *28*, 7.