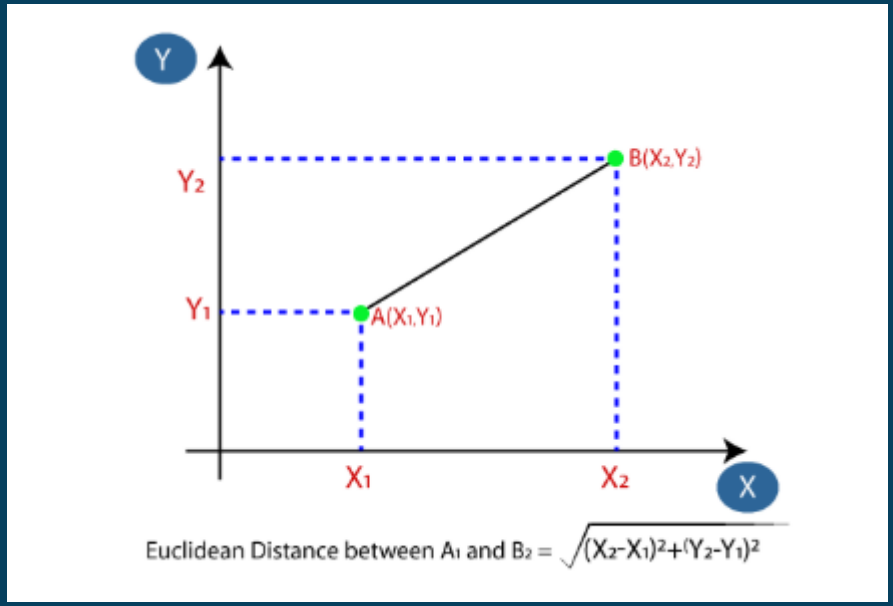
# K-NN (K-Nearest Neighbors)

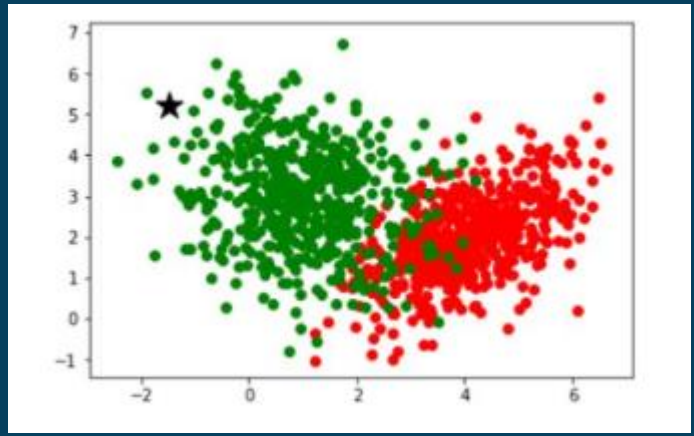
# K-En Yakın Komşu (kNN) Algoritması :

- 1967 yılında T. M. Cover ve P. E. Hart tarafından önerilen, örnek veri noktasının bulunduğu sınıfın ve en yakın komşunun, K değerine göre belirlendiği bir sınıflandırma yöntemidir. Denetimli öğrenmede sınıflandırma ve regresyon için kullanılan algoritmalardan biridir. En basit makine öğrenmesi algoritması olarak kabul edilir. KNN amacı, yeni bir örnek geldiğinde var olan öğrenme verisi üzerinde sınıflandırma yaparak onun en yakın K komşusuna bakarak örneğin sınıfına karar verir. K-NN algoritması sınıflandırma algoritmasıdır. Sınıflandırmak, belirli bir veri kümesini farklı sınıflara ayırma işlemidir. Sınıflandırma hem yapılandırılmış hem de yapılandırılmamış veri türleri üzerinde uygulanabilir

- KNN algoritması sınıflandırılmak istenen bir veriyi daha önceki verilerle olan yakınlık ilişkisine göre sınıflandıran bir algoritmadır. Algoritma adının içinde bulunduğu “K” algoritmaya dahil edilecek veri kümesindeki veri sayısını ifade etmektedir. Yani algoritmada “k” adet komşu aranır. Bir tahmin yapmak istediğimizde, tüm veri setinde en yakın komşuları arar. Algoritmanın çalışmasında bir K değeri belirlenir. Bu K değerinin anlamı bakılacak eleman sayısıdır. Bir değer geldiğinde en yakın K kadar eleman alınarak gelen değer arasındaki uzaklık hesaplanır. İlgili uzaklılardan en yakın k komşu ele alınır. Öznitelik değerlerine göre k komşu veya komşuların sınıfına atanır. Seçilen sınıf, tahmin edilmesi beklenen gözlem değerinin sınıfı olarak kabul edilir. Yani yeni veri etiketlenmiş (label) olur. K-NN non-parametric ( parametrik olmayan ), lazy ( tembel ) bir öğrenme algoritmasıdır. lazy kavramını anlamaya çalışırsak “eager learning” aksine “lazy learning”’in bir eğitim aşaması yoktur. Eğitim verilerini öğrenmez, bunun yerine eğitim veri kümesini “ezberler”. Uzaklık hesaplama işleminde genelde Öklid fonksiyonu kullanılır.



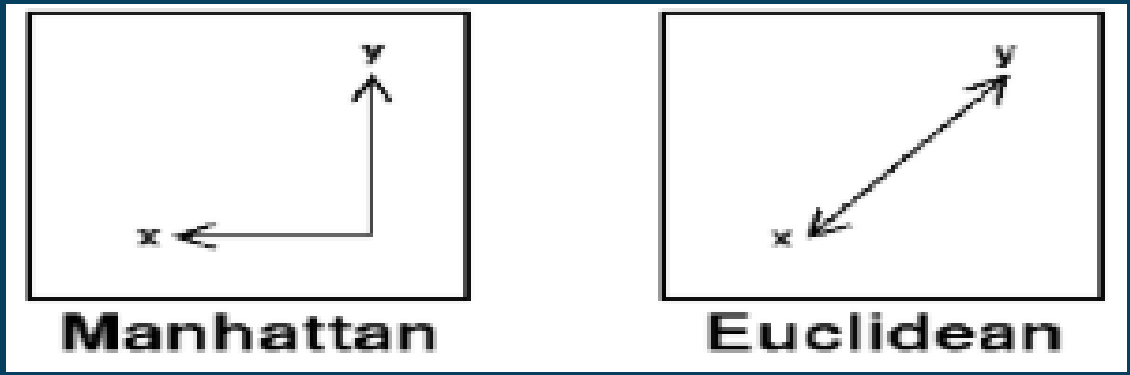
- Bir tahmin yapmak için KNN algoritması, lojistik veya doğrusal regresyonda olduğu gibi bir eğitim veri kümesinden öngörücü bir model hesaplamaz. Aslında, KNN'nin tahmine dayalı bir model oluşturmasına gerek yoktur. Bu nedenle, KNN için gerçek bir öğrenme aşaması yoktur. Bu yüzden genellikle tembel bir öğrenme yöntemi olarak kategorize edilir. Bir tahmin yapabilmek için, KNN herhangi bir eğitim aşaması olmadan bir sonuç üretmek için veri setini kullanır. KNN, bir tahmin yapmak için tüm veri kümesini depolar. KNN herhangi bir tahmine dayalı modeli hesaplamaz ve tembel öğrenme algoritmaları ailesinin bir parçasıdır. KNN, bir girdi gözlemi ile veri kümesindeki farklı gözlemler arasındaki benzerliği hesaplayarak tam zamanında (anında) tahminler yapar.




- Yukarıdaki şekilde, kırmızı veya yeşil olarak sınıflandırılan veri noktalarında siyah bir veri noktası kırmızı ya da yeşil olabilecek iki sınıfı göstermektedir.

KNN algoritmaları, sınıflandırılacak veri noktasına en yakın komşu olan bir k sayısına karar verir. K değeri 5 ise, o veri noktasına en yakın 5 Komşuyu arayacaktır. Bu örnekte, k = 4. KNN en yakın 4 komşuyu bulur. Bu veri noktasının bu komşulara yakın olması nedeniyle sadece bu sınıfa ait olacağı görülmektedir. K-en yakın komşu sınıflandırıcı algoritmalarının basit versiyonu, en yakın komşu sınıfı bularak hedef etiketini tahmin etmektir. Sınıflandırılacak noktaya en yakın sınıf, Öklid mesafesi kullanılarak hesaplanır. Mesafe, benzerliği ölçmek için kullanılır. İki örnek arasındaki mesafeyi ölçmenin birçok yolu vardır.

- Manhattan Distance, |X1-X2| + |Y1-Y2|
- Euclidean Distance, $\sqrt{((x1-x2)2}\ )+\sqrt{(y1-y2)2}$



# Mesafe Ölçüm yöntemleri:

**Minkowsky:**

$$D(x,y) = \left(\sum_{i=1}^{m} |x_i - y_i|^r\right)^{1/r}$$

**Euclidean:**

$$D(x,y) = \sqrt{\sum_{i=1}^{m} (x_i - y_i)^2}$$

**Manhattan / city-block:**

$$D(x,y) = \sum_{i=1}^{m} |x_i - y_i|$$

**Camberra:**

$$D(x,y) = \sum_{i=1}^{m} \frac{|x_i - y_i|}{|x_i + y_i|}$$

**Chebychev:**

$$D(x,y) = \max_{i=1}^{m} |x_i - y_i|$$

**Quadratic:**

$$D(x,y) = (x-y)^T Q (x-y) = \sum_{j=1}^{m} \left( \sum_{i=1}^{m} (x_i - y_i) q_{ji} \right) (x_j - y_j)$$

Q is a problem-specific positive definite $m \times m$ weight matrix

**Mahalanobis:**

$$D(x,y) = [\det V]^{1/m} (x-y)^{\mathrm{T}} V^{-1} (x-y)$$

$V$ is the covariance matrix of $A_1 .. A_m$, and $A_j$ is the vector of values for attribute $j$ occuring in the training set instances $1..n$.

**Correlation:**

$$D(x,y) = \frac{\sum_{i=1}^{m} (x_i - \overline{x_i})(y_i - \overline{y_i})}{\sqrt{\sum_{i=1}^{m} (x_i - \overline{x_i})^2 \sum_{i=1}^{m} (y_i - \overline{y_i})^2}}$$

$\overline{x_i} = \overline{y_i}$ and is the average value for attribute $i$ occuring in the training set.

**Chi-square:**

$$D(x,y) = \sum_{i=1}^{m} \frac{1}{sum_i} \left( \frac{x_i}{size_x} - \frac{y_i}{size_y} \right)^2$$

$sum_i$ is the sum of all values for attribute $i$ occuring in the training set, and $size_x$ is the sum of all values in the vector $x$.

**Kendall's Rank Correlation:**

$$D(x,y) = 1 - \frac{2}{n(n-1)} \sum_{i=1}^{m} \sum_{j=1}^{i-1} \mathrm{sign}(x_i - x_j)\,\mathrm{sign}(y_i - y_j)$$

sign(x)=-1, 0 or 1 if $x < 0$, $x = 0$, or $x > 0$, respectively.

Figure 1. Equations of selected distance functions.
($x$ and $y$ are vectors of $m$ attribute values).

- Mesafe ölçüsünün de yalnızca sürekli değişkenler için geçerli olduğu unutulmamalıdır. Kategorik değişkenler durumunda Hamming mesafesi kullanılmalıdır. Veri setinde sayısal ve kategorik değişkenlerin bir karışımı olduğunda, 0 ile 1 arasındaki sayısal değişkenlerin standardizasyonu konusunu da gündeme getirir.

**Hamming Distance**

$$D_H = \sum_{i=1}^{k} |x_i - y_i|$$

$$x = y \Rightarrow D = 0$$

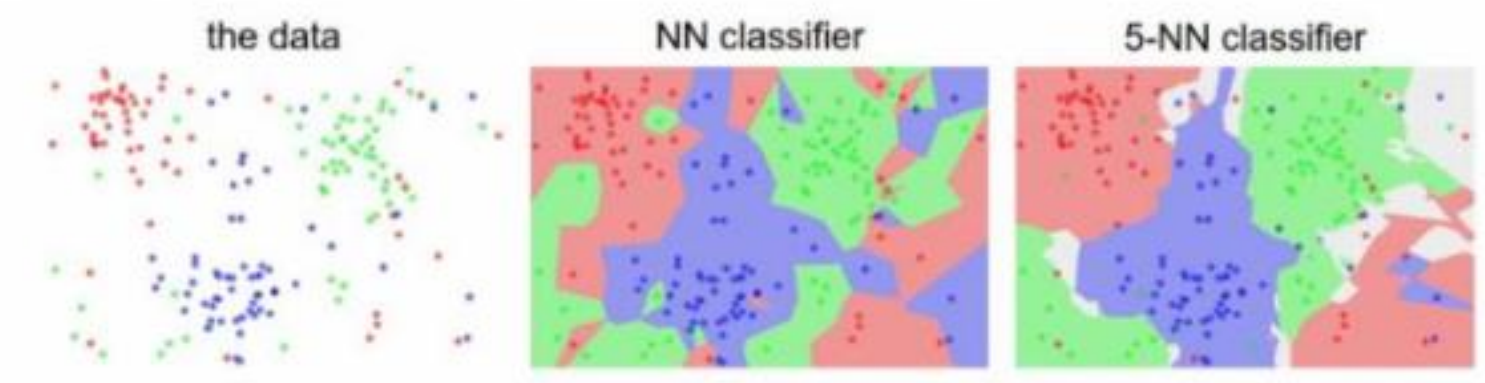
$$x \neq y \Rightarrow D = 1$$

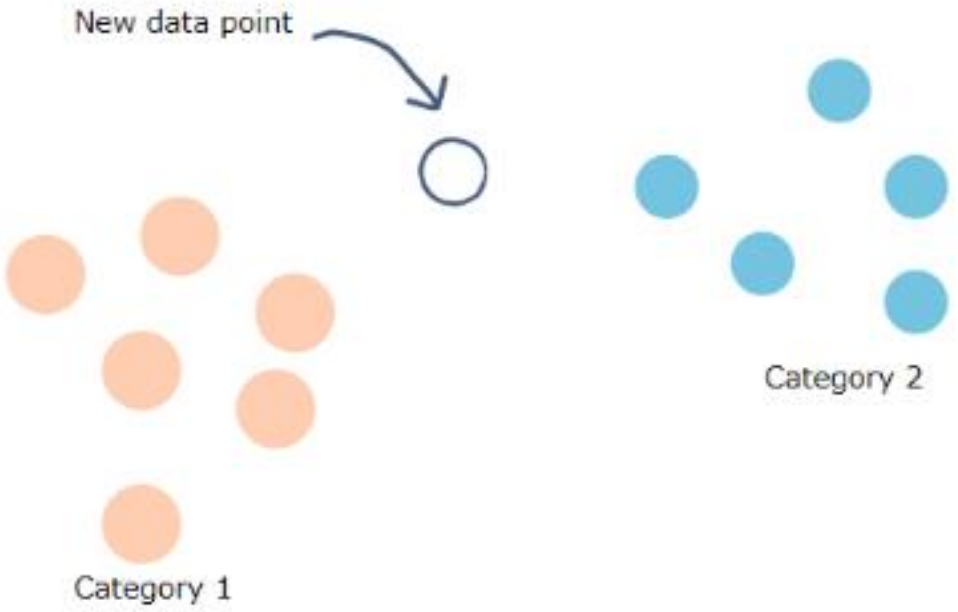
| X | Y | Distance |
|---|---|---|
| Male | Male | 0 |
| Male | Female | 1 |

**K'nin önemi nedir?** K değeri büyüdükçe tahmine duyulan güveni artırır. Öte yandan K çok büyük bir değere sahipse, kararlar çarpık olabilir.

**K nasıl seçilir?** Algoritma, yeni bir veri noktasının diğer tüm eğitim veri noktalarına olan mesafesini hesaplar. Mesafe herhangi bir türde olabilir, örneğin Öklid, Manhattan, vb. Algoritma daha sonra k'ye en yakın veri noktalarını seçer, burada k herhangi bir tam sayı olabilir. Sayısal değerlerin hangi özelliği temsil ettiğine bakılmaksızın, seçimini diğer veri noktalarına yakınlığına göre yapar. Son olarak, veri noktasını benzer veri noktalarının bulunduğu sınıfa atar. Seçilen veri kümesine uyan K değerini seçmek için, KNN algoritması farklı K değerleri ile defalarca çalıştırılr. Sonra, algoritma, yeni değerler için hassas tahminler yapma yeteneğini korurken karşılaşılan hata sayısını azaltan K'yi seçilir.

- K'ye karar vermek, K-en yakın Komşular'ın en kritik kısmıdır.
- K değeri küçükse, gürültü sonuca daha fazla bağımlı olacaktır. Bu gibi durumlarda modelin aşırı uyumu çok fazladır.
- K'nin değeri ne kadar büyükse, KNN'nin arkasındaki prensibi yok edecektir.
- Çapraz doğrulamayı kullanarak K'nin optimum değeri bulunabilir.


the data
NN classifier
5-NN classifier


New data point
Category 1
Category 2

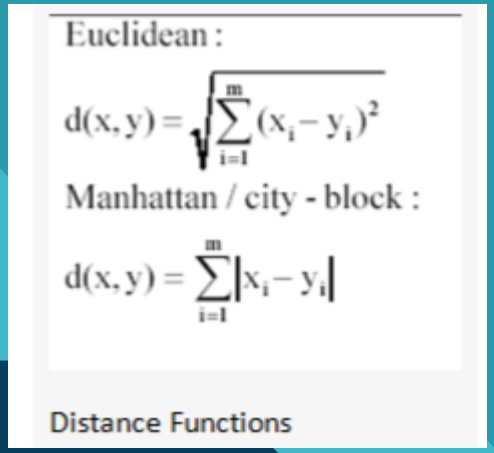
# KNN algoritması sözde kod uygulaması

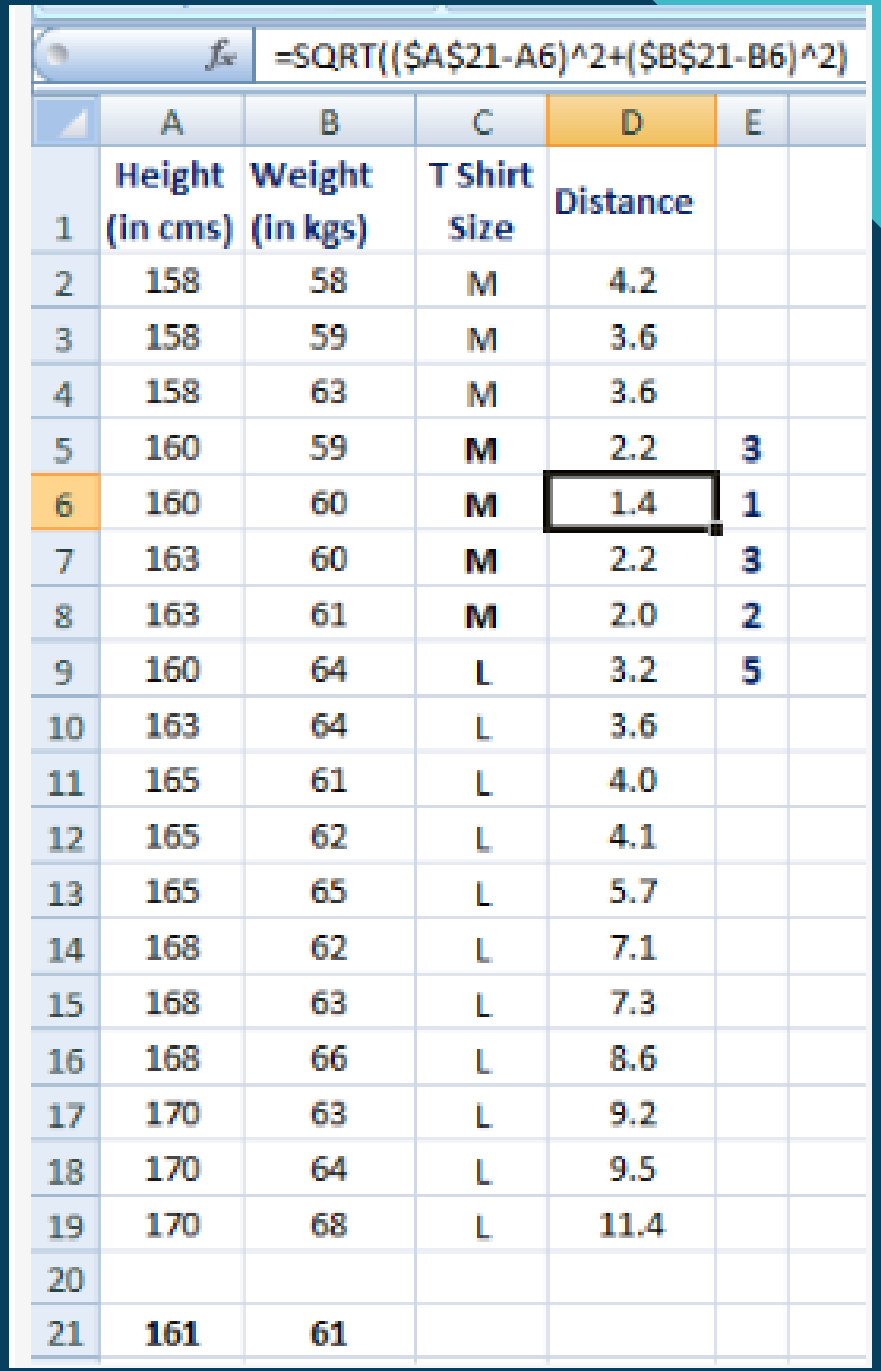
- İstenilen verileri yüklenir.
- k parametresi belirlenir. Bu parametre verilen bir noktaya en yakın komşuların sayısıdır. Örneğin: k=2 olsun. Bu durumda en yakın 2 komşuya göre sınıflandırma yapılacaktır.
- Örnek veri setine katılacak olan yeni verinin, mevcut verilere göre uzaklığı tek tek hesaplanır. İlgili uzaklık fonksiyonları yardımıyla.
- İlgili uzaklılardan en yakın k komşu ele alınır. Öznitelik değerlerine göre k komşu veya komşuların sınıfına atanır.
- Seçilen sınıf, tahmin edilmesi beklenen gözlem değerinin sınıfı olarak kabul edilir. Yani yeni veri etiketlenmiş (label) olur. KNN, anlaşılması oldukça basit bir algoritmadır. Bunun başlıca nedeni, tahmin yapabilmek için bir modele ihtiyaç duymamasıdır. Bunun tersi, tahminini yapabilmek için tüm gözlemlerini hafızasında tutması gerektiğidir. Bu nedenle, girdi veri kümesinin boyutuna dikkat etmeniz gerekir. Ayrıca, mesafenin hesaplanması için yöntemin seçimi ve komşuların K sayısı hemen belli olmayabilir. Kullanım durumunuz için tatmin edici bir sonuç elde etmek için birkaç kombinasyon denemeniz ve algoritmayı ayarlamanız gerekebilir.

- Örnek: KNN, eğitim seti yardımıyla veri noktasını belirli bir kategoriye sınıflandırmaya çalıştığımız parametrik olmayan denetimli bir öğrenme tekniğidir. Basit bir deyişle, tüm eğitim durumlarının bilgilerini yakalar ve yeni durumları benzerliğe göre sınıflandırır. Yeni bir örnek (x) için, en benzer K durum (komşular) için tüm eğitim seti aranarak ve bu K durumlar için çıktı değişkeni özetlenerek tahminler yapılır. Sınıflandırmada bu, mod (veya en yaygın) sınıf değeridir. KNN algoritması nasıl çalışır? Diyelim ki bazı müşterilerin boy, kilo ve tişört bedenleri var ve yeni bir müşterinin Tişört bedenini sadece sahip olduğumuz boy ve kilo bilgisine göre tahmin etmemiz gerekiyor. Boy, kilo ve tişört beden bilgilerini içeren veriler aşağıda gösterilmiştir.

| Height (in cms) | Weight (in kgs) | T Shirt Size |
|---|---|---|
| 158 | 58 | M |
| 158 | 59 | M |
| 158 | 63 | M |
| 160 | 59 | M |
| 160 | 60 | M |
| 163 | 60 | M |
| 163 | 61 | M |
| 160 | 64 | L |
| 163 | 64 | L |
| 165 | 61 | L |
| 165 | 62 | L |
| 165 | 65 | L |
| 168 | 62 | L |
| 168 | 63 | L |
| 168 | 66 | L |
| 170 | 63 | L |
| 170 | 64 | L |
| 170 | 68 | L |

- Adım 1: Uzaklık fonksiyonuna göre Benzerliği hesaplayın Pek çok uzaklık fonksiyonu vardır ama en yaygın olarak kullanılan ölçü Ökliddir. Esas olarak veriler sürekli olduğunda kullanılır. Manhattan mesafesi de sürekli değişkenler için çok yaygındır.
- Mesafe ölçümü kullanma fikri, yeni örnek ve eğitim durumları arasındaki mesafeyi (benzerliği) bulmak ve ardından boy ve ağırlık açısından yeni müşteriye en yakın kmüşterileri bulmaktır. 'Monica' adlı yeni müşteri 161cm boyunda ve 61kg ağırlığındadır. İlk gözlem ile yeni gözlem (monica) arasındaki Öklid uzaklığı aşağıdaki gibidir: =SQRT((161-158)^2+(61-58)^2) Benzer şekilde, yeni vaka ile tüm eğitim vakalarının mesafesini hesaplayacağız ve mesafe açısından sıralamayı hesaplayacağız. En küçük mesafe değeri 1 olarak sıralanır ve en yakın komşu olarak kabul edilir.
- 2. Adım : K-En Yakın Komşuları Bulunması K=5 olsun. Ardından algoritma, özellikler açısından Monica'ya en yakın, yani Monica'ya en çok benzeyen 5 müşteriyi arar ve bu 5 müşterinin hangi kategorilerde olduğunu görür. 4 tanesi 'Orta T shirt bedenleri' ve 1 tanesi ise 'Büyük T gömlek bedeni' vardı, o zaman Monica için en iyi tahmininiz 'Orta T gömlek. Aşağıdaki anlık görüntüde gösterilen hesaplamaya bakın

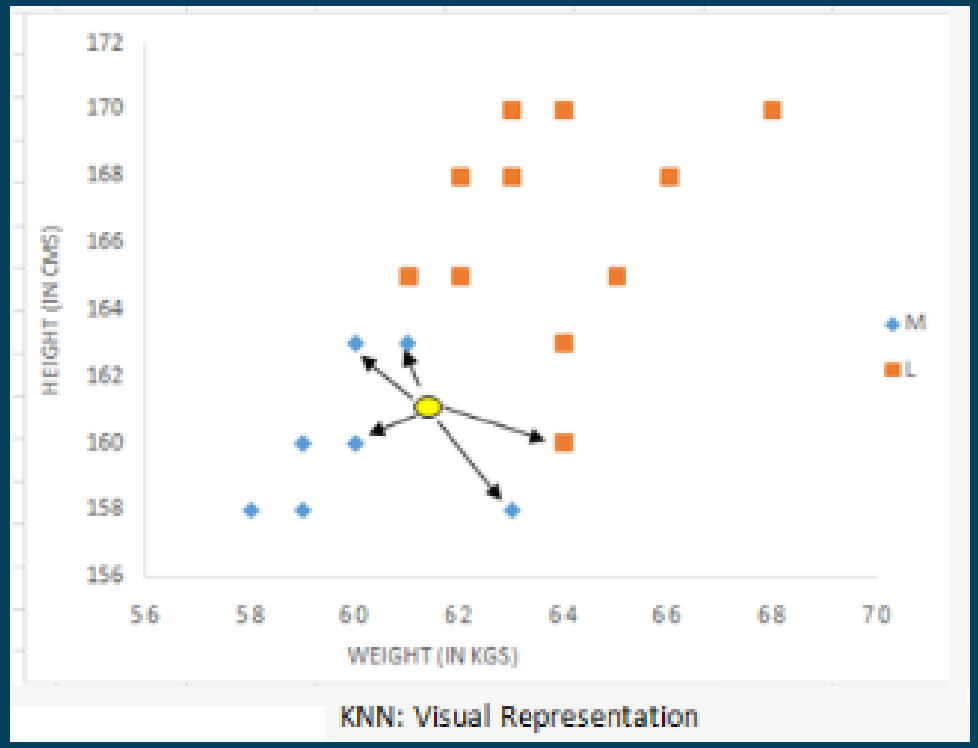




=SQRT(($A$21-A6)^2+($B$21-B6)^2)

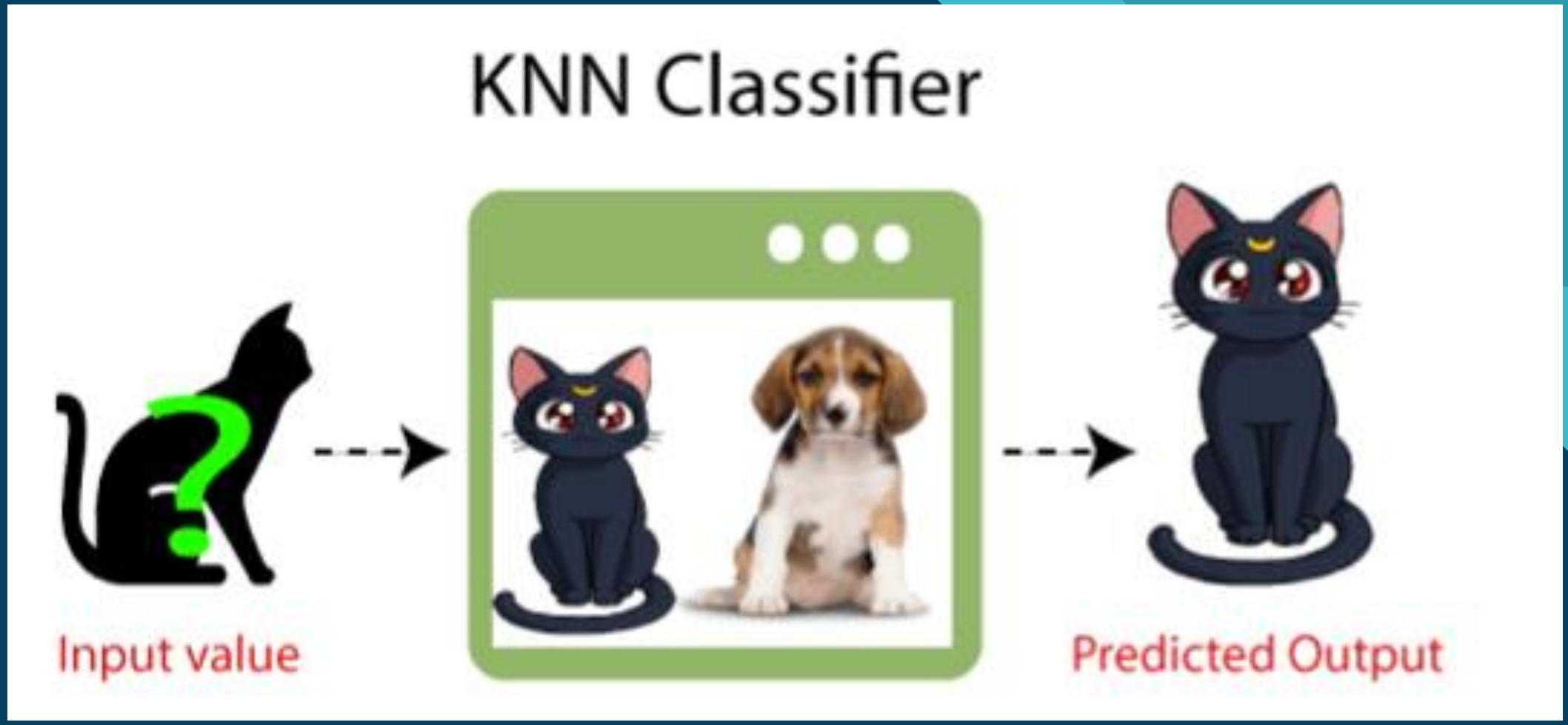
| | A | B | C | D | E |
|---|---|---|---|---|---|
| 1 | Height (in cms) | Weight (in kgs) | T Shirt Size | Distance | |
| 2 | 158 | 58 | M | 4.2 | |
| 3 | 158 | 59 | M | 3.6 | |
| 4 | 158 | 63 | M | 3.6 | |
| 5 | 160 | 59 | M | 2.2 | 3 |
| 6 | 160 | 60 | M | 1.4 | 1 |
| 7 | 163 | 60 | M | 2.2 | 3 |
| 8 | 163 | 61 | M | 2.0 | 2 |
| 9 | 160 | 64 | L | 3.2 | 5 |
| 10 | 163 | 64 | L | 3.6 | |
| 11 | 165 | 61 | L | 4.0 | |
| 12 | 165 | 62 | L | 4.1 | |
| 13 | 165 | 65 | L | 5.7 | |
| 14 | 168 | 62 | L | 7.1 | |
| 15 | 168 | 63 | L | 7.3 | |
| 16 | 168 | 66 | L | 8.6 | |
| 17 | 170 | 63 | L | 9.2 | |
| 18 | 170 | 64 | L | 9.5 | |
| 19 | 170 | 68 | L | 11.4 | |
| 20 | | | | | |
| 21 | 161 | 61 | | | |

- Aşağıdaki grafikte ikili bağımlı değişken (T-shirt bedeni) mavi ve turuncu renkte görüntülenmektedir. 'Orta boy tişört' mavi renkte ve 'Büyük boy tişört' turuncu renktedir. Yeni müşteri bilgileri sarı daire içinde sergilenir. Dört mavi vurgulanmış veri noktası ve bir turuncu vurgulanmış veri noktası sarı daireye yakındır. bu nedenle yeni vaka için tahmin, Orta T-shirt boyutu olan mavi vurgulu veri noktasıdır.



KNN: Visual Representation

- **K-En Yakın Komşu (KNN)** algoritması, denetimli öğrenme tekniğine dayanan en basit makine öğrenmesi algoritmalarından biridir.
- K-NN algoritması, yeni bir durum/veri ile mevcut durumlar arasındaki benzerliği varsayar ve yeni durumu mevcut kategorilere en çok benzeyen kategoriye yerleştirir.
- K-NN algoritması, mevcut tüm verileri saklar ve yeni bir veri noktasını benzerliğe dayalı olarak sınıflandırır. Bu, yeni veri ortaya çıktığında, K-NN algoritması kullanılarak kolayca uygun bir kategoriye sınıflandırılabileceği anlamına gelir.
- K-NN algoritması hem regresyon hem de sınıflandırma için kullanılabilir, ancak çoğunlukla sınıflandırma problemleri için kullanılır.
- K-NN, parametrik olmayan bir algoritmadır, yani temel veri hakkında herhangi bir varsayımda bulunmaz.
- Ayrıca, "tembel öğrenici" (lazy learner) algoritma olarak da adlandırılır çünkü eğitim setinden hemen öğrenmek yerine veri kümesini saklar ve sınıflandırma sırasında veri kümesi üzerinde işlem yapar.
- KNN algoritması eğitim aşamasında sadece veri kümesini saklar ve yeni veri aldığında, bu veriyi yeni veriye en çok benzeyen kategoriye sınıflandırır.


KNN Classifier
Input value
Predicted Output