*T.C*

*SAKARYA ÜNİVERSİTESİ*
**BİLGİSAYAR VE BİLİŞİM BİLİMLERİ FAKÜLTESİ**
**BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ**

# BSM 310 – YAPAY ZEKA

**3. Sınıf 1 / B**

**Grup üyeleri:**

**B1612.10001 ENES ÇAVUŞ**

**B1712.10091 AHMET CAN TURNA**

**B1612.10069 ERAY KAYA**

**B1612.10065 ERSİN YILMAZ**

**Sakarya**
**2020**

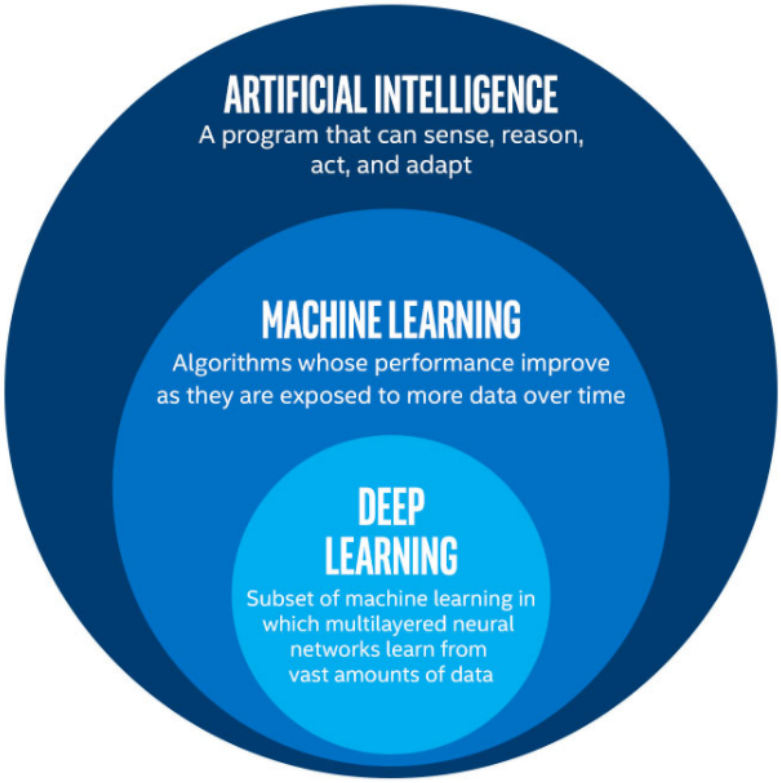
# Yapay Zeka ve KNN

Araştırmamızın konusu olan K – En Yakın Komşu algoritması bir makine öğrenmesi algoritmasıdır. Makine öğrenmesi ise bir yapay zeka alt alanıdır. Bu yüzden büyük resimi anlamak ve bu algoritmanın hangi büyük alanların bir parçası olduğunu görebilmek adına önce yapay zeka biraz araştırıldı daha sonra makine öğrenmesi algoritmalarının türleri ve kendi içlerinde sınıflandırmaları incelendi. Son olarak ise sıra algoritmamızın detaylı araştırılmasına geldi.

Yapay Zeka son 5 yıl içerisinde çok büyük bir ivme kazanarak genişledi. Hatta son zamanlarda dünya çapında her gün yani bir yapay zeka algoritması ya da yeni çalışmalar ortaya atılmaya başlandı. Özellikle sağlık sektöründe büyük bir ilerleme kaydetti.

Yapay zekanın bu kadar ilerlemesi neredeyse her alana hitap edebilmesinde ve birçok alanda büyük zaman ve maliyet kazancı sağlamasındandır. Her alana nasıl hitap edebileceğini araştırdığımızda makine öğrenesi ile karşılaşıyoruz. Makine öğrenmesi ise çeşitli algoritma ve yöntemler ile gerçekleştirilebilmekte. Bu algoritma alanlarından en büyüğü Derin Öğrenme denen insan beynini örnek alan sinirsel algoritmalar bütünü.

Bütün bunları anladıktan sonra büyük resmin bir parçası olan Makine Öğrenmesinin algoritmalarına sıra geliyor.



*Makine Öğrenmesi algoritmaları 2 ana alt başlıkta incelenebilir:*

- Supervised Learning (Gözetimli Öğrenme)

- Unsupervised Learning (Gözetimsiz Öğrenme)

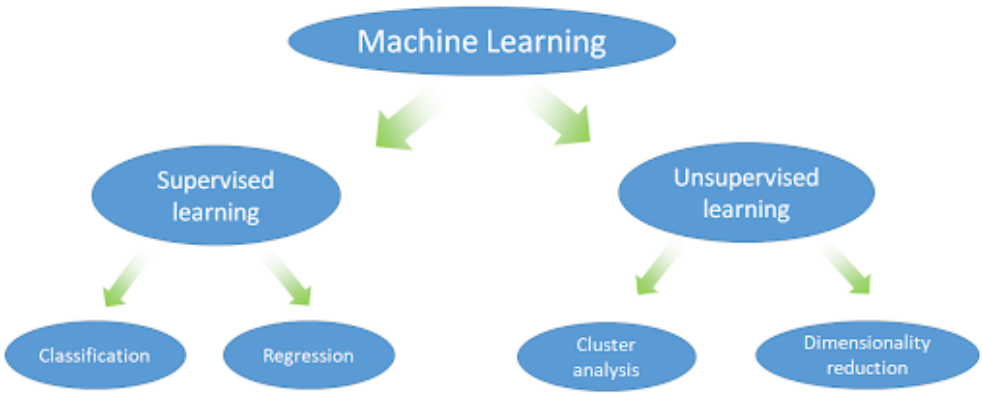
Bizim algoritmamız gözetimli öğrenmenin içerisinde, peki bu gözetimle öğrenme nedir? Gözetimli öğrenmede elimizde hazır ve işaretlenmiş (labelling ) veriler bulunmaktadır yani bir verinin nereye yerleştirilebileceği, hangi kümenin için yer alabileceğini geçmiş verileri değerlendirerek öğrenmemizi sağlayan şeydir gözetimli öğrenme.

Gözetimli öğrenme de 2 ana alt başlıklara ayrılmaktadır. Bunlar:

- Classification ( Sınıflandırma )

- Regression ( Regresyon )

Bizim algoritmamız çoğu algoritmanın aksine her iki tarafta da kullanılabilmektedir.



# KNN nedir?

KNN en popüler makine öğrenmesi algoritmalarından bir tanesidir.
Sınıflandırma - regresyon algoritmaları arasında bulunur.

KNN, anlaması ve uygulaması kolay bir algoritmadır. Bu yüzden Machine Learning alanına yeni girenlerin çoğu bu algoritmaya mutlaka inceler.

Elimizde kümelenebilecek alanları içeren bir veri seti olması bizim için yeterlidir.

Örneğin 3 adet bitki türünü yaprak / gövde boylarına göre kümelemek buna güzel bir örnektir.

Bu alanda ilerlemeye başlandığında algoritmanın popülerliği düşer. Çünkü diğer algoritmalara kıyasla 'tembel algoritma' olarak bilinir, fazla yer kaplar.

Algoritma herhangi bir öğrenme / eğitim aşaması içermemektedir.
Tüm veri setini alır, kümeler ve bu kümeler arasında sınırlarını çizer.

Daha sonra bir input değeri alır ve bu değeri kümelemiş olduğu veri seti arasında hangi kümeye ait olabileceğini bulmaya çalışır.

Bu aşamada belirlemiş olduğumuz "k" değeri devreye girer.

## 1-)K değerini seçme yöntemleri

### 1A-)K değeri genelde şu formülle bulunur

K değeri genelde şu formülle bulunur.

K = sqrt(N)

N : Eğitilen veri sayısı

Elimizde bulunan verilerin sıklığı ve tekrar sayısı gibi veriler k değerini seçmede önemli bir rol oynamaktadır. Dolayısıyla en iyi formül elimizde var olan veriye göre değişmektedir.

### 1B-)K değerini Schwarz methoduyla seçmek

**K değerini Schwarz Criterion gibi metodlar ile de belirleyebiliriz.**

Kmin = bozulma + $\lambda Dk \log N$

D : Problemin boyutu

k : Küme sayısı

N : Eğitilen veri sayısı

$\lambda$ : Özelleştirilmiş parametre

K değeri çeşitli methodlar ile seçilebilir. Yine elinizdeki verinin türü, veri miktarı ve veri akış hızı önemli olmakla birlikte verinin zamanla değişen niteliklerine uyum sağlayan

methodlar seçilebilir. Yazılımcının hayal gücü geniş nasıl olsa..

## 2-)K değerinin seçilmesinde uyulması gereken kurallar

### 2A-)K değeri tek sayı olmalı

K değeri, gruplaştırma yapılacak olan elemana olan uzaklıkların küçükten büyüğe sıralanan listeden kaç tane eleman alınacağı bilgisidir. Dolayısı ile 3'e 3 gibi x'e x'lik gibi bir durumla karşılaşılmaması için K değeri tek sayı seçilmelidir.

3'e 3'lük gibi durumlar iki sınıf arasında kararsızlığa yol açmaktadır.

Bu koşulun başta sınıf sayısının 2 olmasıyla bağdaştırılması doğru gibi gözükse de sınıf sayısı 3 5 gibi değerlerde olduğunda k'nın tek seçilmesinin yine faydası görülecektir.

sınıf = {a, b, c, d}, sınıf sayısı = 4, k =5
a = 0, b = 1, c = 2, d = 2 kararsız!

Ne kadar kurallara uyulsa da kararsızlık durumu ortaya çıkabilir. Bu kurallar kararsızlık durumunun oluşma ihtimalini düşürse de engel olamaz. Bu gibi durumlarda eşit olan sınıflar belirli algoritmalar ile aradaki farka bakılabilir. 2'şer tane sayıldı fakat hangisi daha yakın diye ağırlıklara bakılabilir. Veya tamamen bu duruma göz yumup ilk satırda okunan kod c ise yeni üyenin grubu c olarak seçilebilir.

sınıf = {a, b, c, d, e}, sınıf sayısı = 5, k =5
a = 1, b = 1, c = 1, d = 1, e = 1 kararsız!
Bu durumda dikkat edilmesi gereken madde **2B**'dir.

### 2B-)K değeri ile sınıf sayısı aralarında asal olmalı

K değeri sınıf sayısının bir çarpanı olması durumunda şu durumların gerçekleşmesi olasıdır:

sınıf = {a, b}, sınıf sayısı = 2, k = 2;

a = 1, b = 1 kararsız!

sınıf = {a, b, c}, sınıf sayısı = 3, k = 3;

a = 1, b = 1, c = 1 kararsız!

sınıf = {a, b, c} sınıf sayısı = 3, k = 6;

a = 2, b = 2, c = 2 kararsız!

a = 0, b = 3, c = 3 kararsız!

sınıf = {a, b, c} sınıf sayısı = 3, k = 9;

a = 3, b = 3, c = 3 kararsız!

a = 1 , b = 4, c = 4 kararsız!

Örneğin elimizde 5 farklı sınıf var. Bu durumda k değerini 5, 15, 25, 35 gibi değerler olarak seçmemeliyiz.

## 2C-)K değeri çok büyük seçilmemeli

K değeri büyük bir sayı seçilirse daha pürüzsüz bir seçilim elde edilir. Fakat fazla büyük seçilirse varyansın azalması ve biasın artması anlamına gelmektedir.

Yani K = N durumunda bütün verilere bakılacağı için kim fazlaysa daha çok o gruba dahil olunacaktır. Koyun misali, fazla önyargı..

Bellek israfı da cabası..

Hele ki çalıştığınız veri büyük bir veritabanın oluşturuyorsa hem var olan verilerin ağırlığı hem de verilere eklenen yeni elemanların ağırlığı da hesaba katılmalıdır.

Bellekte oluşturacağınız veri zinciri belleği aşmamalıdır.

K * her bir veri için bellekte işgal edilen alan < bellek alanı * 2 / 3 şeklinde olmalıdır. Hızın önemli olduğu sistemler ve verinin sürekli algoritmaya aktığı sistemlerde bu oran(2/3) daha da düşmektedir.

## 2D-)K değeri küçük seçilebilir fakat

K değeri küçük bir sayı seçilirse gürültü oluşur ve sonuca büyük bir etkisi olur.

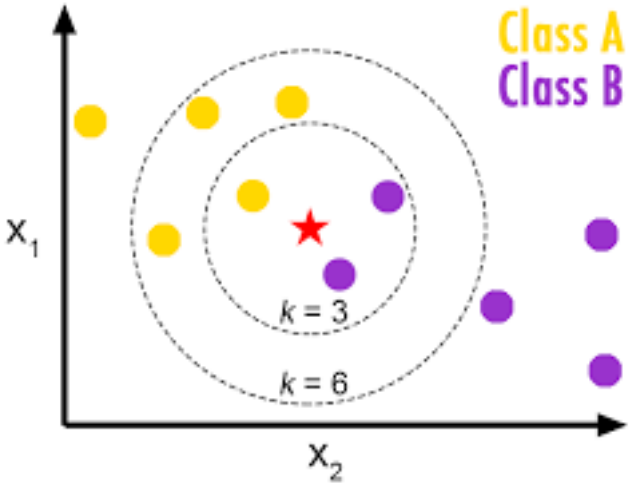
Yani K = 1 seçilmesi gruplaştırma yapılacak olan elemana en yakın birimin grubuna dahil olması demektir. Sert bir filtredir.

Gördüğü ilk elemanın grubuna dahil olma isteği. Körü körüne arkadaşına inanma.

Bu durum veride acayip bir yığılmaya neden olabilir. Fakat yerine görede kullanım alanı baya geniştir.

K değerinin bir diğer sağladığı avantaj ise bellek tasarrufudur. Algoritmanın doğası gereği k = 1 olduğunda veya küçük olduğunda oluşacak döngü ve belleğe alınacak veri zincirinin boyutuda küçük olacak demektir. Bu büyük verilerde müthiş bir hıza işaret ederken algoritmanın yazılış mantığına ve işe yararlılığına büyük bir darbe indirir.

## 3-)K değeri görsel örnek




K = 3 seçilirse yıldız B grubuna,
K = 6 seçilirse de yıldız A grubuna dahil olacaktır.
Kaynakça

# KNN Algoritmasının Uygulama Süreci Adım Adım ve Tüm Parametreleriyle Detaylı Anlatım

KNN algoritması KNN ve PREDICT_KNN depolanmış prosedürlerinde uygulanır. KNN modelini yazdırmak için PRINT_MODEL depolanmış prosedürünü kullanın.

Depolanmış tüm prosedürler, <parametre> = <değer> giriş çiftlerini içeren zorunlu tek dizeli bir parametreden oluşur. Bu girişler virgülle ayrılır. Parametrenin veri türü VARCHAR'dır (herhangi biri).

Geçerli <parametre> = <değer> girdileri, her depolanmış prosedür için parametre açıklamalarında listelenir.

- IDAX.KNN - Bir KNN modeli oluşturun

Bir k-En yakın Komşular modeli oluşturmak için bu saklı yordamı kullanın.

### Yetki

İfadenin yetki kimliğinin sahip olduğu ayrıcalıklar, IDAX_USER rolünü içermelidir.

## Syntax

```
IDAX.PREDICT_KNN(in parameter_string varchar(32672))
```

### Parametre açıklamaları

### parameter_string

Virgülle ayrılmış <parametre> = <değer> girdi çiftlerini içeren zorunlu tek dize parametresi.

Veri türü: VARCHAR (32672)

Aşağıdaki listede parametre değerleri gösterilmektedir:

### Model

Zorunlu. İnşa edilecek KNN modelinin adı. Veri türü: VARCHAR (64)

### Intable

Zorunlu. Giriş tablosunun adı. Veri türü: VARCHAR (128)

### Dd

Zorunlu. Giriş tablosunun benzersiz bir örnek kimliğini tanımlayan sütunu. Veri türü: VARCHAR (128)

### Target

Zorunlu. Girdi tablosunun tahmin sütununu temsil eden sütunu. Veri türü: VARCHAR (128)

## Incolumn

İsteğe bağlı. Girdi tablosunun, noktalı virgül (;) ile ayrılmış belirli özelliklere sahip sütunları. Her sütun, aşağıdaki özelliklerden bir veya daha fazlasıyla gerçekleşir:

Nominal türe (": nom", ": nominal") veya sürekli türe göre (": cont", ": continuous"). Varsayılan olarak, sayısal türler süreklidir ve diğer tüm türler nominaldir.

Rol kimliği (": id"), hedef (": target"), giriş (": active", " in", ": input") veya yoksay (": ignore", ": inactive).

Bu parametre belirtilmezse, giriş tablosunun tüm sütunları varsayılan özelliklere sahiptir.

Varsayılan: hiçbiri Veri türü: VARCHAR (32000)

## Coldeftype

İsteğe bağlı. Giriş tablosundaki sütunların varsayılan türü.

İzin verilen değerler:

nominal tip için "nom" ve "nominal"

sürekli tip için " cont" ve " continuous"

Parametre belirtilmezse, sayısal sütunlar süreklidir ve diğer tüm sütunlar nominaldir.

Varsayılan: hiçbiri Veri türü: VARCHAR (10)

## Coldefrole

İsteğe bağlı. Giriş tablosunun sütunlarının varsayılan rolü.

İzin verilen değerler:

Giriş rolü için " active", " in" ve " input"

Yoksayma rolü için " ignore" ve " inactive"

Parametre belirtilmezse, tüm sütunlar giriş sütunlarıdır.Varsayılan: giriş Veri türü: VARCHAR (8)

## ColPropertiesTable

İsteğe bağlı.Giriş tablosundaki sütunların özelliklerinin depolandığı giriş tablosu.

Bu tablonun biçimi, COLUMN_PROPERTIES () saklı yordamının çıktı biçimidir.

Varsayılan: hiçbiri Veri türü: VARCHAR (128)

### Maxsize

İsteğe bağlı. Modelde saklanan giriş verisi örneğinin maksimum boyutu.Giriş verileri maksimum boyuttan daha az kayıt içeriyorsa, giriş verileri modelde saklanır.

Giriş verileri maksimum boyuttan daha fazla kayıt içeriyorsa, örnekleme aşağıdaki yollardan biriyle yapılır:

Hedef sütun nominalse, hedef sütunda tabakalı örnekleme yapılır.

Hedef sütun sayısalsa, modelde giriş tablosunun rastgele bir örneği saklanır.

Varsayılan: 10000 Minimum: 1 Veri türü: BIGINT

### Randseed

İsteğe bağlı. Rastgele sayı üreteci için tohum. Varsayılan: 12345 Veri türü: INTEGER

### Örnek

```
CALL IDAX.KNN('model=customer_censor_mdl, intable=customer_churn,
id=cust_id, target=censor');
```

- IDAX.PREDICT_KNN - Bir KNN modeli uygulayın

Bir veri kümesi için sınıflandırma tahminleri veya regresyon tahminleri oluşturmak üzere bir KNN modeli uygulamak için bu depolanmış prosedürü kullanın.

### Yetki

İfadenin yetki kimliğinin sahip olduğu ayrıcalıklar, IDAX_USER rolünü içermelidir. Ayrıca, modelin sahibi olmanız veya modeli değiştirme yetkisine sahip olmanız gerekir.

## Syntax

```
IDAX.PREDICT_KNN(in parameter_string varchar(32672))
```

### Parametre açıklamaları

### parameter_string

Virgülle ayrılmış <parametre> = <deger> girdi çiftlerini içeren zorunlu tek dize parametresi.

Veri türü: VARCHAR (32672)

Aşağıdaki listede parametre değerleri gösterilmektedir:

## Model

Zorunlu. Uygulanacak KNN modelinin adı. Veri türü: VARCHAR (64)

## Intable

Zorunlu. Giriş tablosunun adı. Veri türü: VARCHAR (128)

## Outtable

Zorunlu. Giriş tablosunun kayıtları için öngörülen değerlerin saklandığı çıktı tablosunun adı. Veri türü: VARCHAR (128)

## ID

İsteğe bağlı. Giriş tablosunun benzersiz bir kaydı tanımlayan sütunu.

Varsayılan: Modeli oluşturmak için kullanılan kimlik sütununun adı.

Veri türü: VARCHAR (128)

## Target

İsteğe bağlı. Girdi tablosunun tahmin hedefini temsil eden sütunu.

Belirtilen hedef sütun tahmin için kullanılmaz. Varsayılan: Modeli oluşturmak için kullanılan hedef sütunun adı.

Veri türü: VARCHAR (128)

## Distance

İsteğe bağlı. En yakın komşuları hesaplamak için kullanılacak mesafe.

İzin verilen değerler "öklid", "canberra", "manhattan" ve "maksimum" dur.

Varsayılan: 'öklid'

Veri türü: VARCHAR (16)

## K

İsteğe bağlı. Dikkate alınması gereken en yakın komşuların sayısı.

Varsayılan: 3 Aralık:> = 1 Veri türü: INTEGER

## Stand

İsteğe bağlı. Giriş tablosunun ve model tablosunun sürekli giriş sütunlarını standartlaştırdığını gösteren bir işaret.

Sürekli giriş sütunlarının farklı ölçekleri yok sayılır, böylece daha büyük değerlere sahip sütunlar artık dezavantajlı olmaz.

Varsayılan: doğru Veri türü: BOOLEAN

### Fast

İsteğe bağlı. Kesin yöntem yerine daha hızlı yaklaşık hesaplama yönteminin kullanılıp kullanılmayacağını gösteren bir işaret.

Varsayılan: doğru Veri türü: BOOLEAN

### Weights

İsteğe bağlı. Model tablosu için sınıf ağırlıklarını içeren bir tablo.

Modelin hedef sütunu "sürekli" türündeyse sınıf ağırlıkları yoksayılır.

Bu parametreyi belirtmezseniz, tüm ağırlıkların 1'e eşit olduğu varsayılır.

Ağırlık tablosu aşağıdaki sütunları içerir:

Sayısal pozitif ağırlıklar içeren bir AĞIRLIK sütunu

Modelin hedef sütununun değerlerini içeren bir SINIF sütunu

Veri türü: VARCHAR (128)

### İade bilgileri

Sonuç kümesi olarak bir tahminin yapıldığı girdi dizilerinin sayısı.

### Örnek

CALL IDAX.PREDICT_KNN('model=customer_censor_mdl, intable=customer_churn, outtable=customer_censor_score');

# Örnekler

Bu örnek, CUSTOMER_CHURN örnek veri kümesinde KNN modelinin nasıl oluşturulacağını gösterir.

**İlk olarak, CUSTOMER_CHURN tablosuna dayanan CUSTOMER_CHURN_VIEW örnek veri kümesini aşağıdaki gibi oluşturursunuz:**

```
CREATE VIEW CUSTOMER_CHURN_VIEW AS (SELECT CUST_ID, DURATION, CASE WHEN CENSOR=1
THEN 'yes' ELSE 'no' END AS CHURN,
AVG_SPENT_RETAIN_PM, AVG_SQ_SPENT_RETAIN_PM IN_B2B_INDUSTRY,  ANNUAL_REVENUE_MIL
TOTAL_EMPLOYEES,
TOTAL_BUY TOTAL_BUY_FREQ, TOTAL_BUY_FREQ_SQ
FROM CUSTOMER_CHURN);
```

**Daha sonra CUSTOMER_CHURN_VIEW örnek veri kümesini bir eğitim veri kümesine ve bir doğrulama veri kümesine aşağıdaki gibi bölebilirsiniz:**

```
CALL IDAX.SPLIT_DATA('intable=customer_churn_view,
traintable=customer_churn_train,
testtable=customer_churn_test, id=cust_id, fraction=0.35');
```

**Aşağıdaki çağrı, customer_churn_train veri kümesinde algoritmayı çalıştırır ve KNN modelini oluşturur.**

```
CALL IDAX.KNN('model=customer_churn_mdl, intable=customer_churn_train,
id=cust_id, target=churn');
```

**PREDICT_KNN saklı yordamı, CHURN sütununun değerini tahmin eder.**

**Aşağıdaki çağrı, değerlerin yeni işlemlerle nasıl ilişkilendirileceğini gösterir.**

```
CALL IDAX.PREDICT_KNN('model= customer_churn_mdl, intable= customer_churn_test,
outtable=customer_churn_score');
```

**KNN customer_churn_mdl modelini oluşturmak için kullanılmayan CUSTOMER_CHURN_TEST veri kümesinin kayıt değerlerindeki kayıp değerlerini customer_churn_score tahminiyle karşılaştırarak önceki adımdaki tahminleri doğrulayabilirsiniz:**

```
SELECT s.id, s.class churn_predicted, churn from customer_churn_test i,
customer_churn_score s where i.cust_id=s.id;
```

# SORULAR

SORU 1 ) 3 farklı sınıfın bulunduğu bir veri kümesine göre k değeri için aşağıdakilerden hangisi yanlıştır?

A ) Tek sayı olması
B ) 3'ün katı olmaması
C ) 1 olarak seçilmemesi
D ) N olarak seçilmemesi
E ) Çok büyük bir değer seçilmemesi

SORU 2 ) Aşağıdakilerden hangisi KNN uygulama adımlarından yanlıştır?

A ) Veri Seti Bulma
B ) Veri Seti Analizi
C ) Veri Görselleştirme İşlemleri
D ) Optimum k değeri bulma
E ) K değeri bulduktan sonra veri temizleme işlemi yapma

SORU 1 ) KNN algoritması için söylenenlerden hangisi yanlıştır?

A ) Öğrenmesi ve entegre etmesi kolaydır
B ) Veriler arası mesafeden kaynaklanan dengesizliklerden etkilenmez
C ) Hafıza kullanımı fazladır
D ) Büyük verilerde sürekli işlem gerektiğinden yavaş ( tembel ) kalır
E ) Herhangi bir ön eğitim aşaması içermemektedir

SORU 4 ) Aşağıdakilerden hangisi KNN algoritmasına göre yanlıştır?

A ) Veri seti boyutu ile algoritma hızı arasında ters orantı vardır.
B ) K  değerini 1 seçmek yanıltıcı tahminlere yol açacaktır.
C ) K değeri seçimi veri setine göre değişiklik göstermez, k = sqrt ( n ) formulu yeterlidir.
D ) KNN algoritması çoğunluk oylama mekanizması kullanır.
E ) Algoritma önceden öğrenme gibi bir adımı içermemektedir.

SORU 5 ) 6 farklı bitki türü için KNN algoritması kullanılsa aşağıdakilerden hangi k değeri bu sınıflandırma için uygun olabilir?

A ) 1
B ) 9
C ) 24
D ) 48
D ) 600

# KAYNAKLAR

https://stackoverflow.com/questions/11568897/value-of-k-in-k-nearest-neighbor-algorithm

https://www.quora.com/How-can-I-choose-the-best-K-in-KNN-K-nearest-neighbour-classification?share=1

https://qiita.com/yshi12/items/26771139672d40a0be32

https://towardsdatascience.com/top-10-algorithms-for-machine-learning-beginners-149374935f3c

https://www.mindmeister.com/export/image/969402186?height=600&t=MITG7ZIKaz&variable_size=1&width=1200

https://towardsdatascience.com/machine-learning-basics-with-the-k-nearest-neighbors- algorithm-6a6e71d01761