DLA

09.11.2022
- Week 7 - (1)

Neden Normalization yapıyorum?

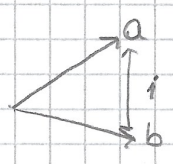
* Veriler arasında farklılığın çok fazla olduğu durumlarda verileri tek bir düzen içerisinde ele alabilmek için
* Ayrıca, farklı ölçekleme sisteminde bulunan verilerin birbiri ile karşılaştırılabilmesi için.

↳ Complex sayılarla daha rahat çalışmak için.
↳ 0-1 aralığı için.

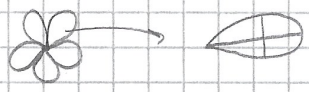
$X = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \end{bmatrix}$ $x_1 \rightarrow 1500, 75000$ → 1500'den 75000'e stajyerden Ceo'ya maaş

$x_2 \rightarrow 20, 60$

KNN $d = \sqrt{(x_1^a - x_1^b)^2 + \dots + (x_n^a - x_n^b)^2}$



* → Why doesn't we need normalization in iris data?



iris datada genişlik ve boy bilgileri içerir ve bu bilgiler hep cm cinsi old. için burada normaliz. gerek yoktu. Bu dataseti norm. olmadan kurtarır. Fakat diğer datasetlerde önce normal. yapmalısın.

Bazı datasetlerde verilerin birbirleri ile birimleri tutmayabilir. veya aralıkları çok yüksek olabilir. Mesela;

Yaş	Ücret
20	500
↓	↓
40	800

→ burada knn distance kullanırken $\sqrt{(800-500)^2 + (40-20)^2} - - -$ bu çok büyük çıkacağı için diğer x_n'leri önemsiz gibi gösterir

* Hepsini dikkate alıp düzgünce değerlendirebilmek ve gerçekçi olsun diye normalizasyon yapmalısın.

$X = \frac{X - X_{min}}{X_{max} - X_{min}}$ → [0,1] → 0-1 arasında olması için normalize yapıyoruz.

Standart scaler = $X = \frac{X - \mu}{\sigma}$

DLA

09.11.2022 (2)

Dataset

Preliminary Analysis → generate new data → PCA

Normalization

Split

Bu sırayla yapmalısın.

* Tüm datayı birlikte normalize edip sonra split etmelisin. Önce split edip sonra normalize edersen train & test'deki normalizasyonlar birbirini tutmayacaktır.

* Buyüzden önce Normalizasyon sonra Split.

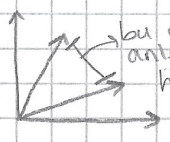
Normalization, KNN, DT & SVM

KNN'de normalization:

x_1'in ölçeğindeki küçük bir değişiklik x_2'deki değişimi yok sayıyorsa yani maaşım 10.000 iken 11.000'e çıktığında değişim $(1000)^2$ iken [maaşa göre küçük bir ölçek fakat sayısal olarak $(bin)^2$ yüksek old. için] yaş 20 → 60 değişim $(40)^2$ olduğunda yaşın değişimi kendi scalasında çok olmasına $(1000)^2$'nin yanında hiçbir anlamı kalmaz. Buyüzden normalize ediyoruz.

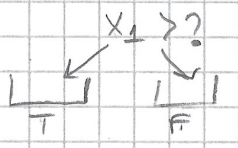
Yaş:	20	40	60
Normalize Yaş:	0	0.5	1

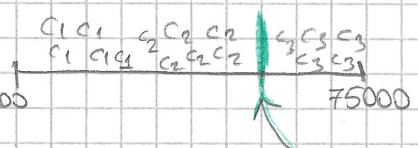
böyle yapınca



Decision Tree'de normalization: min-max, standart dene hiçbir şey fark etmeyecek

Çünkü DT şöyle çalışır:

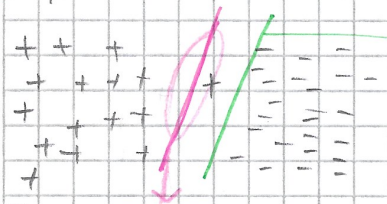




C_1: 50
C_2: 50
C_3: 50

* DT ben bunu nereden ayırırsam en az hetero olur diyor. Buraya koyarım diyor. Yani 0'dan 1'e gitmiş, 1500'den 75000'e gitmiş fark etmiyor. O yine dataseti ortadan bir yerden bölüyor.

Support Vector Machine'de (SVM) normalization: C = margin



→ hard margin böyle kesiyor

* Ben buna baktığımda hard margin kullanırsam generalize olmaz. Soft margin kullanırsan sonuç daha smooth olur.

Benim samplem böyle ama benim göremediğim yerlerde belkide +, − karışık var.

C önemli!!

C = 1 old. daha lose (?) öğrenme yapıyorsun özelliklede birbirine geçmiş datalarda