T.C.

İSTANBUL GELİŞİM ÜNİVERSİTESİ

MÜHENDİSLİK VE MİMARLIK FAKÜLTESİ

BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ

2020-2021 GÜZ DÖNEMİ

BIL415 MAKİNE ÖĞRENMESİ DERSİ

DÖNEM SONU ÖDEV RAPORU

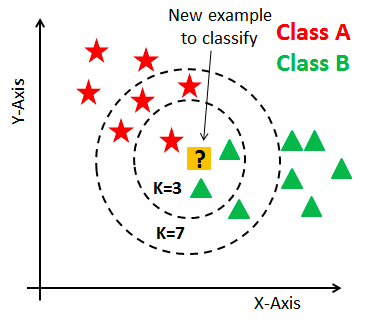
K NEAREST NEIGHBORHOOD

ÖMER FARUK ÖZTÜRK

170403057

06 OCAK 2020

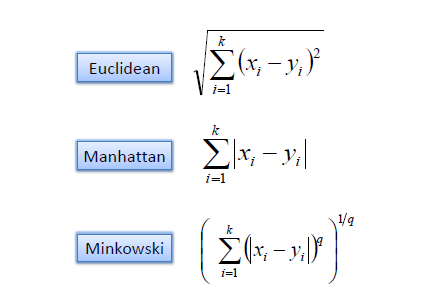
KNN (K NEAREST NEIGHBORHOOD)



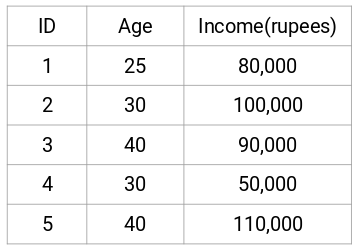
K Nearest Neighborhood (En yakın K komşu) , tembel olması ile bilinen bir makine öğrenmesi algoritmasıdır . "*Bana arkadaşını söyle, sana kim olduğunu söyleyeyim*" mantığında çalışmaktadır .

Yaptığı iş temel olarak girilen değeri ilgili veri setindeki verilerle kıyaslayıp , kullanıcıdan aldığı k değerine göre kıyasladığı verilerden k tanesini seçmektir . Kıyasladığı kriter ise uzaklıktır . Veri setindeki her sample her feature için birer uzaklık değeri alıp bu uzaklıkları topladıktan sonra o sample için bir uzaklık değeri belirler . Ardından kendisine en yakın k tane komşusundan y değerlerini alıp buna göre bir oylama yapar . Oylamadan çıkan değer bizim verimizin y değeri olarak belirlenir .

Şekil 1



Uzaklık değerini bulmak için çeşitli yöntemler vardır . Başlıca Euclidian , Manhattan , Minkowski metodlarını örnek gösterebiliriz . Ben mevcut programında Minkowski uzaklık metodunu kullandım çünkü , Manhattan metodunun bir nevi bir genel formu olması hasebiyle bu metodu kullanmak istedim .



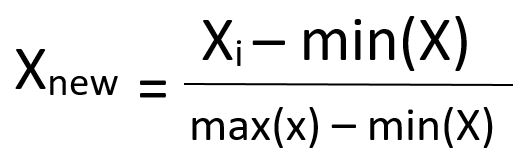
Şekil 2

Fakat her uzaklık değeri diğer labellardaki ile aynı aralıkta değildir ve bu da bizim toplam uzaklık değerimize etki eden labellardan bazılarını diğerlerine göre baskın kılmaktadır . Buna bir örnek vermek gerekirse sol tarafta bulunan Şekil 3 deki tabloya bakabiliriz .

Şekil 3

Gördüğünüz gibi tabloda (ID yi saymazsak) Age(Yaş) ve Income(Gelir) verilerimiz bulunmakta .

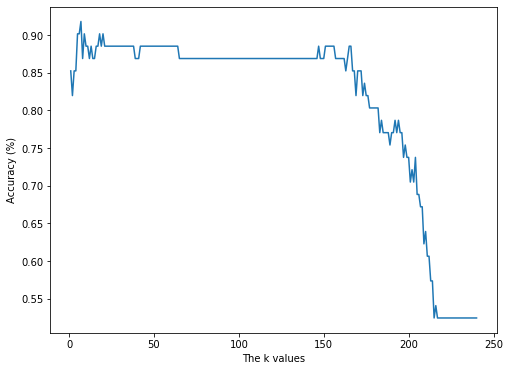
Bu veriler arasındaki fark çok fazla olduğu için verilerin birbirlerine baskınlıklarının olacağı barizdir ve bunun çözümü ise veri setimizi normalize etmekten geçmektedir .

 Üç çeşit normalizasyon metodu vardır ve ben bu programda z-score normalizasyon yöntemini kullandım . Formülünü göstermek gerekirse yan taraftaki Şekil 4 bunun için uygundur .

Şekil 4

Normalizasyon aşamasını geçtikten sonra tahminleme işlemimizi yapmamızın önünde hiçbir engel bulunmamaktadır .

Yazdığım kod satırına *findTheBestK* adında bir fonksiyon daha ekledim . Bu fonksiyonun yapmasını istediğim iş yüklediğimiz veri seti için en iyi K değerinin yani başarı oranı en fazla olan K değerinin alınmasıydı . Fonksiyon kullanıcıdan aldığı minimum ve maksimum K değerlerini tek tek tahminleme yaptığımız fonksiyon ile bir döngü içerisinde dener ve başarı oranlarını kıyaslar . Başarı oranı yüksek bir K değeri bulduğunda mevcut olan en iyi K değişkenindeki değeri bulunan K değeri ile değiştirir . Fonksiyonun sonunda da buna dair Şekil 5 dekine benzer bir grafik bastırır .



Şekil 5