19 Ağustos 2020 - Muhammed Furkan Gülşen

Makine Öğrenimi Algoritmaları Ve Karşılaştırmaları

Makine öğrenimi kısaca; bilgisayarların programlamadan bir sorunu nasıl çözeceklerini öğrendikleri bilimsel bir tekniktir.

1- Doğrusal (Lineer) Regresyon
2- Lojistik (Logistic) Regresyon
3- K en yakın komşular
4- Karar (Decision) Ağaçları
5- Destek Vektör Makinesi (SVM)
6- Rastgele (Random) Orman
7- Naive Bayes

1- Doğrusal Regresyon (Linear)

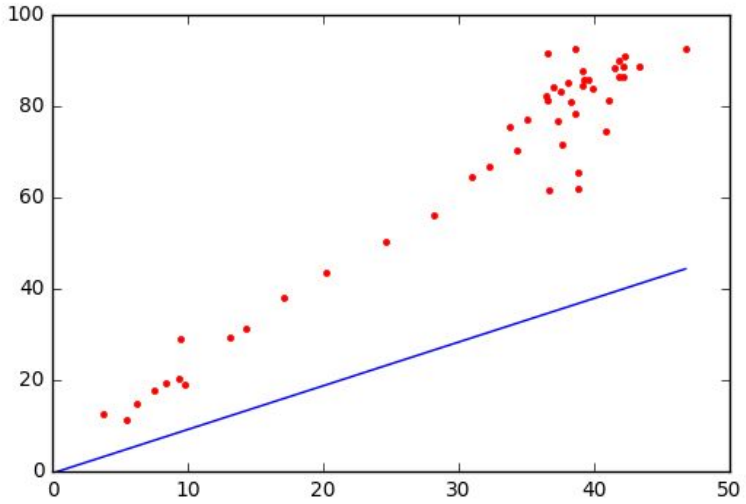
Özellikleri alır ve sürekli bir çıktı öngörür. Örneğin: hisse senedi fiyatı, maaş vb. Adından da anlaşılacağı gibi doğrusal regresyon, her soruna doğrusal bir eğri çözümü bulur.

Temel Teori

LR, eğitim özelliklerinin her biri için ağırlık parametresi olan teta'yı tahsis eder. Öngörülen çıktı (h (θ)), özelliklerin ve θ katsayılarının doğrusal bir fonksiyonu olacaktır.

$$h_\theta = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \dots$$

Θ değerlerini doğru yönde hizalamak için Gradient Descent" algoritması kullanılacaktır. Aşağıdaki diyagramda, her kırmızı nokta eğitim verilerini temsil eder ve mavi çizgi türetilen çözümü gösterir.



Kayıp İşlevi

LR'de, kayıp metriği olarak Mean Squared Error kullanırız. Beklenen ve gerçek çıktıların sapmasının karesi alınacak ve toplanacaktır. Bu kaybın türevi Gradient Descent algoritması tarafından kullanılacaktır.

Avantajları

- Kolay ve basit uygulama,
- Hızlı eğitim,
- Θ katsayılarının değeri, özellik önemi varsayımı verir.

Dezavantajları

- Yalnızca çözüm doğrusal ise uygulanabilir. Pek çok gerçek hayat senaryosunda, durum böyle olmayabilir.
- Algoritma, girdi artıklarının (hata) normal dağıldığını varsayar, ancak her zaman karşılanmayabilir.
- Algoritma, girdi özelliklerinin karşılıklı olarak bağımsız olduğunu varsayar (eş doğrusallık yoktur).

Hiperparametreler

- Düzenleme Parametresi (λ):

    - Verilere aşırı uymayı önlemek için düzenleme kullanılır. λ ne kadar yüksekse, düzenlileştirme o kadar yüksek olur ve çözüm yüksek oranda önyargılı olur. λ düşürmek, çözümü yüksek varyanslı yapar. Bir ara değer tercih edilmelidir.
- Öğrenme Oranı (α):
    - Eğitim sırasında Gradient Descent algoritması uygulanırken θ değerlerinin ne kadar düzeltilmesi gerektiğini tahmin eder. α ayrıca orta bir değer olmalıdır.

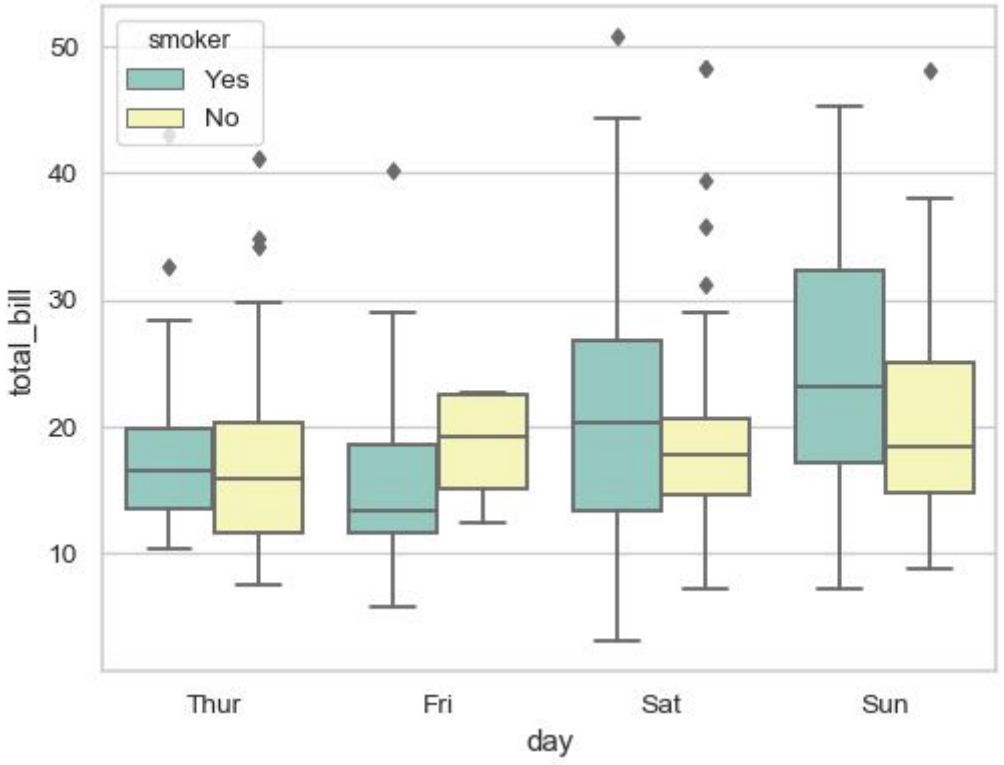
LR İçin Varsayımlar

- Bağımsız ve bağımlı değişkenler arasındaki doğrusal ilişki.
- Eğitim verilerinin homoskedastik olması, yani hataların varyansının bir şekilde sabit olması gerektiği anlamına gelir.
- Bağımsız değişkenler eş doğrusal olmamalıdır.

Eşdoğrusallık ve Aykırı Değerler:

Bir özelliğin diğerinden bir miktar doğrulukla doğrusal olarak tahmin edilebildiği durumlarda iki özelliğin eş doğrusal olduğu söylenir.

- Eşdoğrusallık basitçe standart hatayı şişirecek ve bazı önemli özelliklerin eğitim sırasında önemsiz kalmasına neden olacaktır. İdeal olarak, eğitimden önce eşdoğrusallığı hesaplamalı ve yüksek düzeyde ilişkili özellik kümelerinden yalnızca bir özelliği tutmalıyız.



Aykırı değer, eğitim sırasında karşılaşılan başka bir zorluktur. Normal gözlemlere göre aşırı olan ve modelin doğruluğunu etkileyen veri noktalarıdır.

- Aykırı değerler hata fonksiyonlarını şişirir ve doğrusal regresyonun eğri fonksiyonunu ve doğruluğunu etkiler. Düzenli hale getirme (L1), viol parametrelerinin şiddetli bir şekilde değişmesine izin vermeyerek aykırı değerleri düzeltebilir.
- Keşif veri analizi aşamasında, aykırı değerlere dikkat etmeli ve onları düzeltmeli / ortadan kaldırmalıyız. Bunları tanımlamak için kutu grafiği kullanılabilir.

Diğer Modellerle Karşılaştırma

LR algoritmasını diğer algoritmalardan ayıran en büyük farklardan biri, LR'nin yalnızca doğrusal çözümleri desteklemesidir. Makine öğreniminde

diğerlerinin hepsinden daha iyi performans gösteren en iyi modeller yoktur ve verimlilik, eğitim veri dağıtım türüne bağlıdır.

LR vs Karar Ağaçları

- Daha az veri setine sahip (düşük gürültülü) çok sayıda özellik olduğunda, doğrusal regresyonlar Karar ağaçlarından / rastgele ormanlardan daha iyi performans gösterebilir. Genel durumlarda, Karar ağaçları daha iyi bir ortalama doğruluğa sahip olacaktır.
- Kategorik bağımsız değişkenler için karar ağaçları doğrusal regresyondan daha iyidir.
- Karar ağaçları, eşdoğrusallığı LR'den daha iyi yönetir.

LR vs SVM

- SVM, çekirdek numarası kullanarak hem doğrusal hem de doğrusal olmayan çözümleri destekler.
- SVM, aykırı değerleri LR'den daha iyi yönetir.
- Her ikisi de eğitim verisi daha az olduğunda ve çok sayıda özellik olduğunda iyi performans gösterir.

LR vs KNN

- KNN parametrik olmayan bir modeldir, oysa LR parametrik bir modeldir.
- KNN, tüm eğitim verilerini takip etmesi ve komşu düğümleri bulması gerektiğinden gerçek zamanlı olarak yavaştır, oysa LR, ayarlanmış θ katsayılarından çıktıyı kolayca çıkarabilir.

LR vs Sinir Ağları

- Sinir ağları, LR modeline kıyasla daha büyük eğitim verilerine ihtiyaç duyarken, LR daha az eğitim verisiyle bile iyi çalışabilir.

- NN, LR'ye kıyasla yavaş olacaktır.
- Ortalama doğruluk, sinir ağlarında her zaman daha iyi olacaktır.

2- Lojistik (Logistic)Regresyon

Doğrusal regresyon gibi, Lojistik regresyon da sınıflandırma algoritmaları ile başlamak için doğru algoritmadır. 'Regresyon' adı olsa da, bu bir regresyon modeli değil, bir sınıflandırma modelidir. İkili çıktı modelini çerçevelemek için lojistik bir işlev kullanır. Lojistik regresyonun çıktısı bir olasılık ($0 \leq x \leq 1$) olacaktır ve çıktı olarak ikili 0 veya 1'i tahmin etmek için kullanılabilir ($x < 0.5$, çıktı = 0, aksi takdirde çıktı = 1).

Temel Teori

Lojistik Regresyon, doğrusal regresyona oldukça benzer davranır. Ayrıca doğrusal çıktıyı hesaplar, ardından regresyon çıktısı üzerinden bir saklama işlevi izler. Sigmoid işlevi, sıklıkla kullanılan lojistik işlevdir. Yanda, z değerinin Denklem (1) 'deki doğrusal regresyon çıktısı ile aynı olduğunu açıkça görebilirsiniz.

$$z = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \ldots.$$

$$h(\theta) = g(z)$$

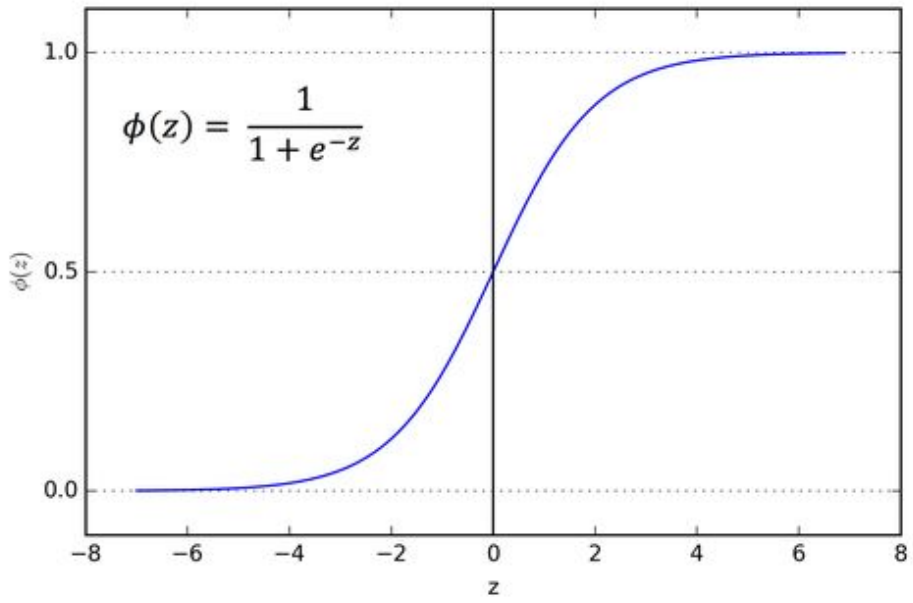
$$g(z) = \frac{1}{1 + e^{-z}}$$

Buradaki h (θ) değeri P(y = 1 / x) 'e karşılık gelir, yani, x girişi verildiğinde çıkışın ikili 1 olma olasılığı. P (y = 0 / x), 1-h (θ) 'ye eşit olacaktır.

z'nin değeri 0 olduğunda, g (z) 0.5 olacaktır. Z pozitif olduğunda, h (θ) 0,5'ten büyük olacak ve çıktı ikili 1 olacaktır. Benzer şekilde, z negatif olduğunda, y'nin değeri 0 olacaktır. Sınıflandırıcıyı bulmak için doğrusal bir

denklem kullandığımızda, çıktı modeli de doğrusal bir boyut olacaktır, yani giriş boyutunu bir alandaki tüm noktalar aynı etikete karşılık gelecek şekilde iki boşluğa böler.

Aşağıdaki şekil bir sigmoid fonksiyonun dağılımını göstermektedir.



Kayıp İşlevi

Mean Squared Error kayıp fonksiyonu olarak kullanamayız (doğrusal regresyon gibi), çünkü sonunda doğrusal olmayan bir sigmoid fonksiyonu kullanıyoruz. MSE işlevi yerel minimum değerleri getirebilir ve Gradient Descent algoritmayı etkileyecektir.

Yani burada kayıp fonksiyon olarak Cross Entropy kullanıyoruz. y=1 ve y= 0'a karşılık gelen iki denklem kullanılacaktır. Buradaki temel mantık, tahminim çok yanlış olduğunda (örneğin: y '= 1 & y = 0), maliyetin sonsuz olan -log(0) olacağıdır.

$$J(\theta) = \frac{1}{m}\sum cost(y', y)$$

$$cost(y', y) = -log(1 - y') \;\; if \;\; y = 0$$

$$cost(y', y) = -log(y') \;\; if \;\; y = 1$$

Verilen denklemde, m eğitim verisi boyutunu, y' tahmin edilen çıktıyı ve y gerçek çıktıyı temsil eder.

Avantajları

- Kolay, hızlı ve basit sınıflandırma yöntemi.
- θ parametreler, bağımsız değişkenlerin bağımlı değişken üzerindeki öneminin yönünü ve yoğunluğunu açıklar.
- Çok sınıflı sınıflandırmalar için de kullanılabilir.
- Kayıp işlevi her zaman dışbükeydir.

Dezavantajları

- Doğrusal olmayan sınıflandırma problemlerine uygulanamaz.
- Uygun özellik seçimi gereklidir.
- İyi sinyal-gürültü oranı beklenir. (!)
- Eşdoğrusallık ve aykırı değerler LR modelinin doğruluğunu bozar.

Hiperparametreler

Lojistik regresyon hiperparametreleri, lineer regresyonunkine benzer. Yüksek doğruluk elde etmek için öğrenme hızı (α) ve Düzenli hale getirme parametresi (λ) doğru şekilde ayarlanmalıdır.

LR İçin Varsayımlar

Lojistik regresyon varsayımları, doğrusal regresyon modeline benzer.

Diğer Modellerle Karşılaştırma

Lojistik Regresyon vs SVM

- SVM doğrusal olmayan çözümleri işleyebilirken lojistik regresyon yalnızca doğrusal çözümleri işleyebilir.
- Doğrusal SVM, maksimum marj çözümü elde ettiği için aykırı değerleri daha iyi yönetir.
- SVM'deki hinge loss, LR'deki günlük kaybından daha iyi performans gösterir.

Lojistik Regresyon vs Karar Ağaçları (Decision Tree)

- Karar ağacı, eşdoğrusallığı LR'den daha iyi yönetir.
- Karar ağaçları özelliklerin önemini çıkaramaz, ancak LR yapabilir.
- Karar ağaçları, kategorik değerler için LR'den daha iyidir.

Lojistik Regresyon vs Sinir Ağı (NN)

- NN, LR'nin destekleyemediği doğrusal olmayan çözümleri destekleyebilir.
- LR, dışbükey kayıp işlevine sahiptir, bu nedenle yerel bir minimumda asılı kalmaz, oysa NN askıda kalabilir.
- Eğitim verileri daha az ve özellikler büyük olduğunda LR, NN'den daha iyi performans gösterirken, NN'nin büyük eğitim verilerine ihtiyacı vardır.

Lojistik Regresyon vs Naive Bayes

- Naive Bayes üretken bir modeldir, LR ise ayırt edici bir modeldir.
- Naive Bayes küçük veri kümeleri ile iyi çalışır, oysa LR düzenlileştirme benzer performans sağlayabilir.
- Naive Bayes tüm özelliklerin bağımsız olmasını beklediğinden, LR eşdoğrusallık konusunda Naive Bayes ten daha iyi performans gösterir.

Lojistik Regresyon vs KNN

- KNN, LR'nin parametrik bir model olduğu parametrik olmayan bir modeldir.
- KNN, Lojistik Regresyondan nispeten yavaştır.
- KNN, LR'nin yalnızca doğrusal çözümleri desteklediği doğrusal olmayan çözümleri destekler.
- LR, (tahmini hakkında) güven düzeyini türetebilirken, KNN yalnızca etiketleri çıkarabilir.

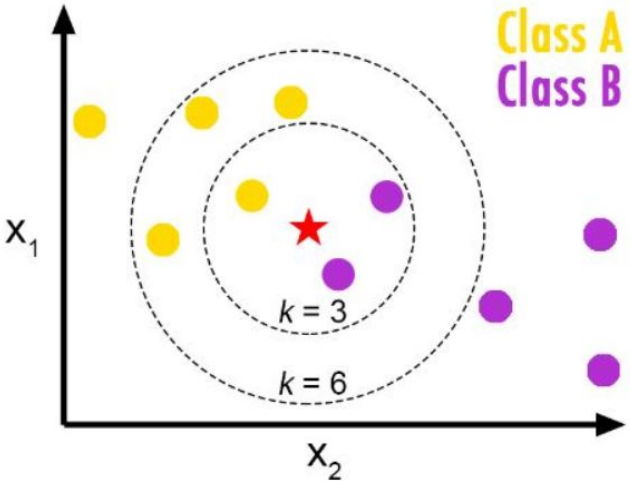
3- Doğrusal Regresyon K-Em Yakın Komşular (KNN)

K-en yakın komşular, sınıflandırma ve regresyon için kullanılan parametrik olmayan bir yöntemdir. Kullanılan en kolay makine öğrenimi tekniklerinden biridir. Yerel yaklaşımla tembel bir öğrenme modelidir.

Temel Teori

KNN'nin arkasındaki temel mantık, etrafındakileri keşfetmek, test veri noktasının onlara benzer olduğunu varsaymak ve çıktıyı türetmektir. KNN'de k komşular ararız ve öngörü ile geliriz.

KNN sınıflandırması durumunda, en yakın k veri noktasına çoğunluk oylaması uygulanırken, KNN regresyonunda, en yakın k veri noktasının ortalaması çıktı olarak hesaplanır. Genel bir kural olarak, tek sayıları k olarak seçeriz. KNN, hesaplamaların yalnızca çalışma zamanında gerçekleştiği tembel bir öğrenme modelidir. en yakın k veri noktası üzerinden çoğunluk oylaması uygulanırken, KNN regresyonunda, en yakın k veri noktasının ortalaması çıktı olarak hesaplanır. Genel bir kural olarak, tek sayıları k olarak seçeriz. KNN, hesaplamaların yalnızca çalışma zamanında gerçekleştiği tembel bir öğrenme modelidir.



Yukarıdaki diyagramda sarı ve mor noktalar, eğitim verilerinde Sınıf A ve Sınıf B'ye karşılık gelir. Kırmızı yıldız, sınıflandırılması gereken test verilerini gösterir. k = 3 olduğunda, B Sınıfını çıktı olarak ve K = 6 olduğunda çıktı olarak A Sınıfını tahmin ederiz.

Kayıp İşlevi

KNN ile ilgili herhangi bir eğitim yoktur. Test sırasında minimum mesafeli k komşu sınıflandırma / regresyonda yer alacaktır.

Avantajları

- Kolay ve basit makine öğrenimi modeli.
- Ayarlanacak az sayıda hiperparametre.

Dezavantajları

- k değeri akıllıca seçilmelidir.
- Örnek boyutu büyükse, çalışma süresi boyunca yüksek hesaplama maliyeti.
- Özellikler arasında adil muamele için uygun ölçeklendirme sağlanmalıdır.

Hiperparametreler

KNN esas olarak iki hiperparametre, K değeri ve mesafe fonksiyonu içerir.

- K değeri:
  KNN algoritmasına kaç komşu katılacağı. k, doğrulama hatasına göre ayarlanmalıdır.
- Uzaklık işlevi:
  Öklid mesafesi en çok kullanılan benzerlik işlevidir. Manhattan mesafesi, Hamming Distance, Minkowski mesafesi farklı alternatiflerdir.

Varsayımlar

- Giriş alanı hakkında net bir şart olmalıdır.
- Uygun şekilde orta düzeyde örneklem büyüklüğü (yer ve zaman kısıtlamaları nedeniyle).
- Eşdoğrusallık ve aykırı değerler eğitimden önce ele alınmalıdır.

Diğer Modellerle Karşılaştırma

KNN ve diğer modeller arasındaki genel bir fark, diğerlerine kıyasla KNN'nin ihtiyaç duyduğu büyük gerçek zamanlı hesaplamadır.

KNN vs Naive Bayes

- Naive bayes, KNN'nin gerçek zamanlı uygulaması nedeniyle KNN'den çok daha hızlıdır.
- Naive bayes parametrik iken KNN parametrik değildir.

KNN vs Doğrusal Regresyon

- Veriler yüksek SNR'ye sahip olduğunda KNN, doğrusal regresyondan daha iyidir.

KNN vs SVM

- SVM, aykırı değerlere KNN'den daha iyi bakıyor.
- SVM, büyük özellikler ve daha az eğitim verisi olduğunda KNN'den daha iyi performans gösterir.

KNN vs Sinir Ağları (NN)

- Sinir ağları, yeterli doğruluğu elde etmek için KNN'ye kıyasla daha büyük eğitim verilerine ihtiyaç duyar.
- NN, KNN'ye kıyasla çok fazla hiperparametre ayarına ihtiyaç duyar.

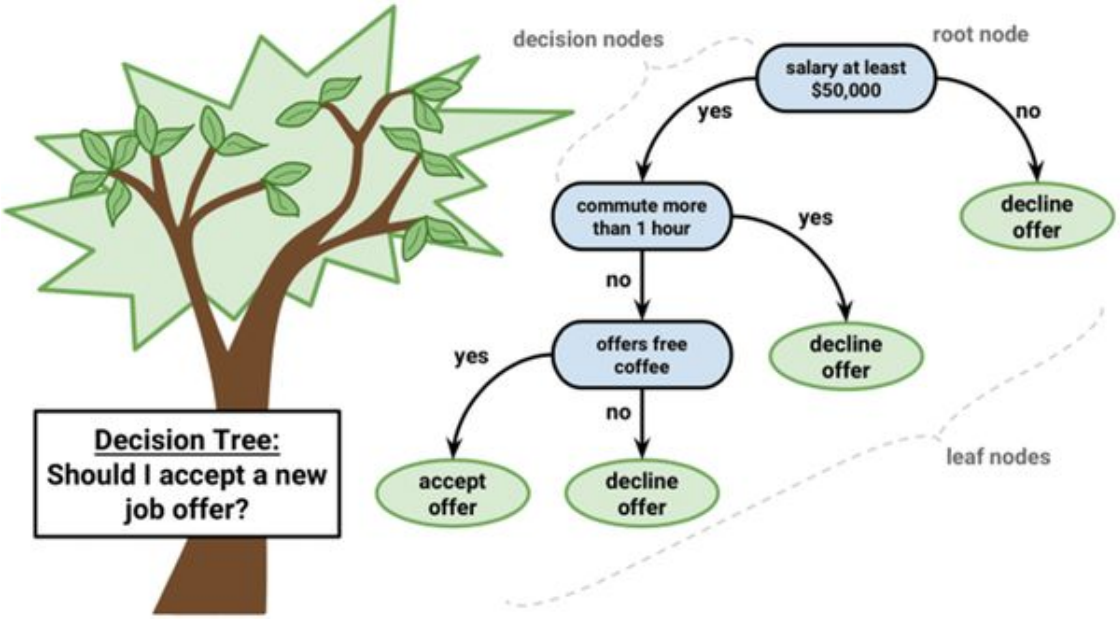
4- Karar Ağacı (Decision Tree)

Karar ağacı, regresyon ve sınıflandırma problemlerini çözmek için kullanılan ağaç tabanlı bir algoritmadır. Çıktı türetmek için homojen bir

olasılık dağıtılmış kök düğümünden oldukça heterojen yaprak düğümlerine dallanmış ters bir ağaç çerçevelenir. Regresyon ağaçları, sürekli değerlere sahip bağımlı değişken için ve ayrık değerli bağımlı değişken için sınıflandırma ağaçları kullanılır.

Temel Teori

Karar ağacı, her bir düğümün bir özellik üzerinde bir koşula sahip olduğu bağımsız değişkenlerden türetilir. Düğümler, koşula bağlı olarak bir sonraki düğümün hangi düğüme gideceğine karar verir. Yaprak düğüme ulaşıldığında, bir çıktı tahmin edilir. Doğru koşul dizisi, ağacı verimli kılar. entropi / bilgi kazancı, düğümlerdeki koşulları seçmek için kriter olarak kullanılır. Ağaç yapısını türetmek için recursive, greedy tabanlı bir algoritma kullanılır.



Yukarıdaki diyagramda, iç düğümler (koşullar) ve etiketlerle yaprak düğümler (teklifi reddet / kabul et) içeren bir ağaç görebiliriz.

Koşulları Seçmek İçin Algoritma

CART (sınıflandırma ve regresyon ağaçları) için, sınıflandırma ölçütü olarak gini indeksini kullanıyoruz. Veri noktalarının ne kadar iyi karıştırıldığını hesaplamak için bir ölçüdür.

$$giniindex = 1 - \sum P_t^2$$

maksimum gini indeksli özellik, karar ağacını oluşturmanın her aşamasında sonraki koşul olarak seçilir. Set eşit olmayan şekilde karıştırıldığında, gini puanı maksimum olacaktır.

Avantajları

- Veriler üzerinde ön işleme gerek yoktur.
- Verilerin dağıtımı konusunda varsayım yok.
- Eşdoğrusallığı verimli bir şekilde yönetir.
- Karar ağaçları, tahmin üzerinde anlaşılır bir açıklama sağlayabilir.

Dezavantajları

- Yüksek saflık elde etmek için ağacı inşa etmeye devam edersek, modele overfitting uygulamış olabiliriz. Bu sorunu çözmek için karar ağacı budama kullanılabilir.
- Aykırı değerlere eğilimli.
- Karmaşık veri kümelerini eğitirken ağaç çok karmaşık hale gelebilir.
- Sürekli değişkenleri işlerken değerli bilgileri kaybeder.

Hiperparametreler

Karar ağacı birçok hiperparametre içerir ve bunlardan birkaçını listeleyeceğim.

- kriter
  Sonraki ağaç düğümünü seçmek için hangi maliyet fonksiyonu. Çoğunlukla kullanılanlar gini / entropidir.
- maksimum derinlik
  Karar ağacının izin verilen maksimum derinliğidir.
- minimum numune bölünmesi
  Dahili bir düğümü bölmek için gereken minimum düğümdür.
- minimum numune yaprağı
  Yaprak düğümünde olması gereken minimum numune.

Diğer Modellerle Karşılaştırma

Karar Ağaçları vs Rastgele Orman

- Rastgele Orman, karar ağaçlarının bir koleksiyonudur ve ormanın ortalama / çoğunluk oyu tahmin edilen çıktı olarak seçilir.
- Rastgele Orman modeli, Karar ağacına göre fazla uyum göstermeye daha az meyilli olacak ve daha genel bir çözüm sunacaktır.
- Random Forest, karar ağaçlarından daha sağlam ve doğrudur.

Karar Ağaçları vs KNN

- Her ikisi de parametrik olmayan yöntemlerdir.
- Karar ağacı, otomatik özellik etkileşimini desteklerken KNN olamaz.
- KNN'nin pahalı gerçek zamanlı uygulaması nedeniyle karar ağacı daha hızlıdır.

Karar Ağaçları vs Naive Bayes

- Karar ağacı ayırt edici bir modeldir, Naive bayes ise üretici bir modeldir.

- Karar ağaçları daha esnek ve kolaydır.
- Karar ağacı budama, eğitim verilerindeki bazı temel değerleri ihmal edebilir ve bu da bir atış için doğruluğa yol açabilir.

Karar Ağaçları vs Sinir Ağları (NN)

- Her ikisi de doğrusal olmayan çözümler bulur ve bağımsız değişkenler arasında etkileşime sahiptir.
- Eğitim verilerinde büyük kategorik değerler kümesi olduğunda karar ağaçları daha iyidir.
- Senaryo, kararla ilgili bir açıklama talep ettiğinde, karar ağaçları NN'den daha iyidir.
- Yeterli eğitim verisi olduğunda NN, karar ağacından daha iyi performans gösterir.

Karar Ağaçları vs SVM

- SVM, doğrusal olmayan problemleri çözmek için çekirdek numarası kullanırken, karar ağaçları problemi çözmek için girdi uzayında hiper dikdörtgenler türetir.
- Karar ağaçları kategorik veriler için daha iyidir ve eşdoğrusallığı SVM'den daha iyi ele alır.

5-Destek Vektör Makinesi(SVM)

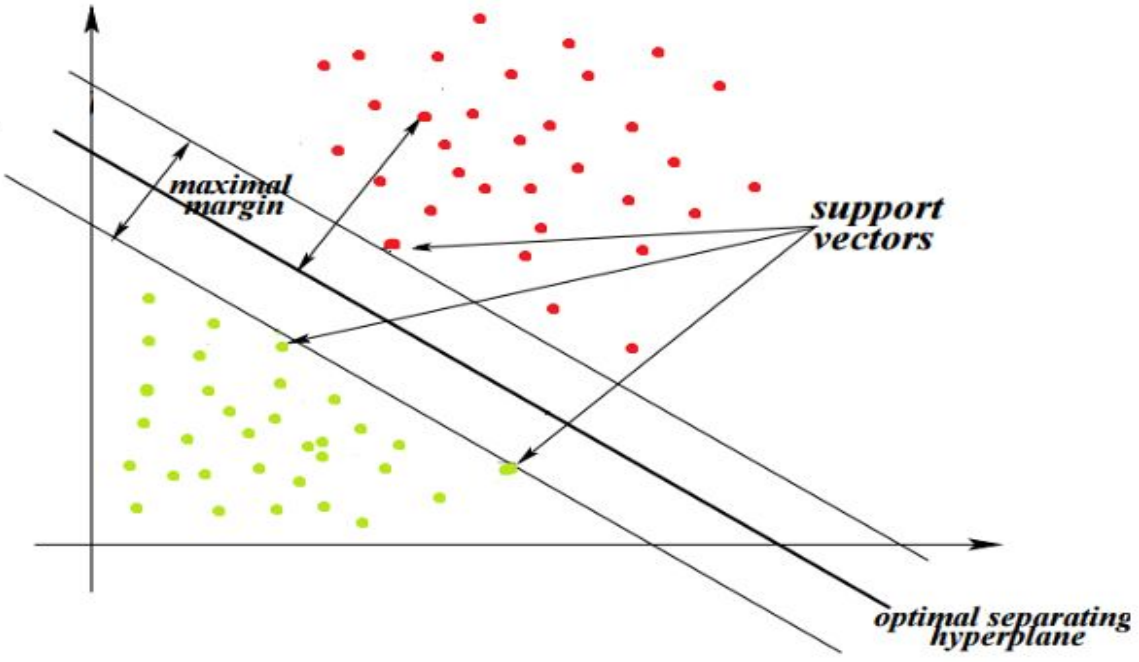
Destek Vektör makinesi, hem sınıflandırma hem de regresyon için kullanılabilen bir ML tekniği türüdür. Doğrusal ve doğrusal olmayan sorunları desteklemek için başlıca iki çeşidi vardır. Doğrusal SVM'nin

çekirdeği yoktur ve soruna minimum marjlı doğrusal çözüm bulur. Çekirdekli SVM, çözüm doğrusal olarak ayrılabilir olmadığında kullanılır.

Temel Teori

Destek Vektör Makinesi, metin sınıflandırması, görüntü sınıflandırması, biyoinformatik vb. Alanlarda yaygın olarak kullanılan denetimli bir öğrenme tekniğidir.

Model tarafından sınıflandırma marjını maksimize eden bir hiper düzlem türetilir. N özellik mevcutsa, hiper düzlem, N-1 boyutlu bir alt uzay olacaktır. Özellik uzayındaki sınır düğümlerine destek vektörleri denir. Göreceli konumlarına bağlı olarak, maksimum kenar boşluğu türetilir ve orta noktada bir optimal hiper düzlem çizilir.



Kenar boşluğunun (m) değeri, //w// ile ters orantılı olacaktır; burada w, ağırlık matrislerinin kümesidir. Marjı maksimize etmek için, //w// 'yi en aza indirmemiz gerekecek. Optimizasyon problemi,

$$\text{Minimize} \quad \frac{||\vec{w}||^2}{2} \quad \text{where} \quad y_i\,(\vec{w}.\vec{x}+b) \geq 1 \text{ for any } i = 1,\ldots,n$$

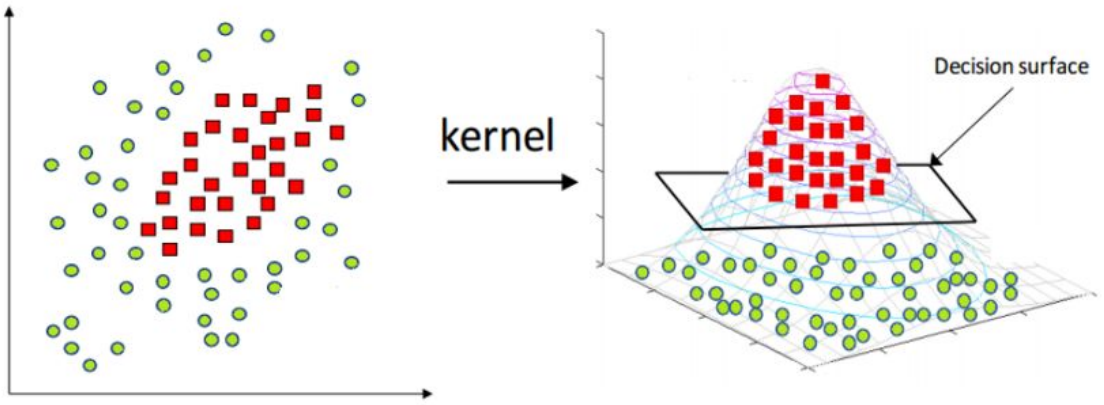
Yukarıdaki optimizasyon, tamamen doğrusal olarak ayrılabilir çözümler için iyi çalışır. Aykırı değerleri ele almak için, aşağıdaki gibi bir gevşek terime ihtiyacımız var. İkinci terim, slack değişken elde etmek için hinge kaybını kullanır.

$$\text{Minimize } \frac{1}{2}\|w\|^2 + C\sum_i \max(0, 1 - y_i(w^\intercal x_i + b))$$

C, ıskalama cezasını ve marj genişliğini dengeleyen düzenlilik parametresidir. Matematiksel açıklamalar bu hikayenin kapsamının çok üzerinde olduğundan, onları derinlemesine açıklamayacağım.

Temel mantık, maliyet işlevini en aza indirmek için, w'nin sınıflar arasında maksimum marjla ayar yapmaya zorlanmasıdır. C değeri, veri kümeleri üzerinde uygulanan düzenlilik düzeyine karar verecektir. Veri kümeleri üzerine uygulanacak düzey (yumuşak / sert) marjına karar verir. Kısacası, C, aykırı değerlere karşı bilgisizlik düzeyidir.

Veri kümesi doğrusal olarak ayrılabilir olmadığında N doğrusal SVM . Tüm eğitim verileri için yeni bir alt düzlem türetmek için bir çekirdek işlevi kullanılır. Etiketlerin yeni alt düzlemdeki dağılımı, eğitim verilerinin doğrusal olarak ayrılabilir olacağı şekilde olacaktır. Daha sonra, doğrusal bir eğri, alt düzlemdeki etiketleri sınıflandıracaktır. Sınıflandırma sonuçları özellik uzayına geri yansıtıldığında, doğrusal olmayan bir çözüm elde ederiz.



Buradaki denklemdeki tek değişiklik, yeni bir çekirdek fonksiyonunun tanıtılmasıdır. Yeni denklem şöyle görünecek:

$$\text{Minimize } \frac{||\vec{w}||^2}{2} + C\sum_i \zeta_i \quad \text{where } y_i(\vec{w}.\phi(x_i) + b) \geq 1 - \zeta_i \text{ for all } 1 \leq i \leq n, \zeta_i \geq 0$$

Xi, veri kümesini yeni hiper düzleme dönüştürecek olan ϕ (xi) ile değiştirilecektir.

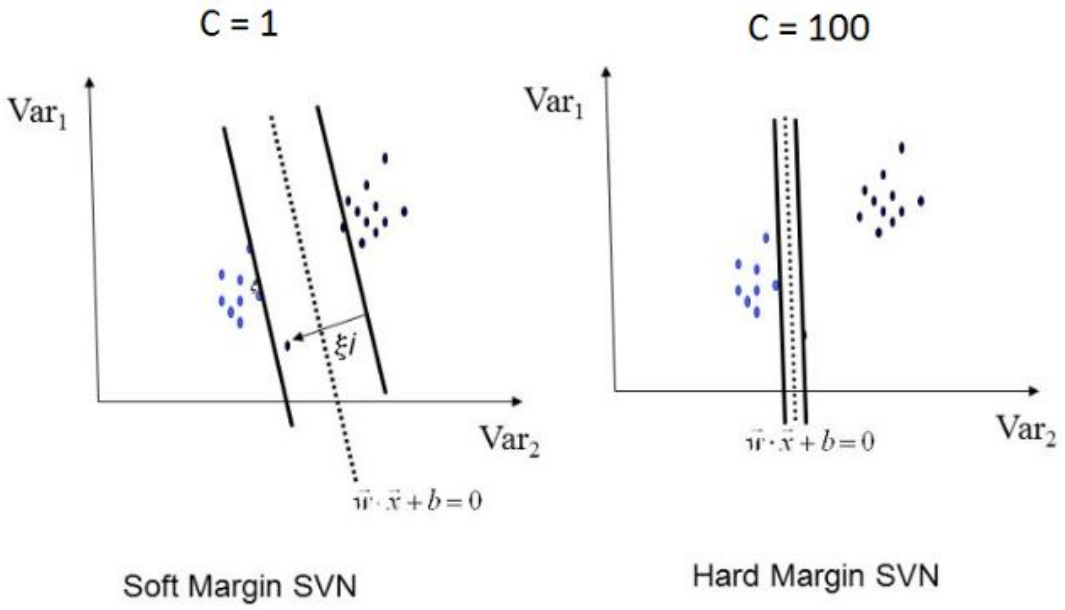
Kayıp İşlevi

Eşitlik 1'deki kayıp fonksiyonu aşağıdaki gibi iki kısma ayrılabilir:

$$1)\ \frac{1}{2}\|w\|^2 \qquad 2)\ C\sum_i \max(0, 1 - y_i(w^\top x_i + b))$$

İlk terim w parametrelerini en aza indirmeye ve yüksek marj elde etmeye çalışır. İkinci terim menteşe kaybına karşılık gelir . Her veri kümesi için bolluk değişkenini hesaplar. Herhangi bir veri seti marj arasına veya yanlış tarafa gelirse, menteşe kaybı ile ceza verilecektir.

İlk terimin küçültülmesi marjın genişlemesinin azalmasına ve genişlemesine neden olur. İkinci terimin en aza indirilmesi, menteşe kaybını azaltmak için marjın kısalmasına neden olur. C'nin değerine bağlı olarak, nihayet sabit bir marjı belirledik. C'nin değeri, eğride yumuşak / sert bir marj belirler.



Yukarıdaki diyagramda, marjın türetilmesinde C'nin etkileri hakkında açıktır.
Doğrusal olmayan çekirdeklerde Gauss çekirdeği, polinom çekirdeği, Sigmoid çekirdeği, Laplace RBF çekirdeği vb. Kullanabiliriz.

Avantajları

- SVM, karmaşık çözümleri çözmek için çekirdek numarası kullanır.
- SVM, küresel minimuma her zaman ulaşılabilen dışbükey bir optimizasyon işlevi kullanır.
- Hinge(menteşe) kaybı daha yüksek doğruluk sağlar.
- Aykırı değerler, yumuşak kenar boşluğu sabiti C kullanılarak iyi bir şekilde ele alınabilir.

Dezavantajları

- Hinge(menteşe) kaybı seyrekliğe yol açar.
- Hiper parametreler ve çekirdekler, yeterli doğruluk için dikkatlice ayarlanmalıdır.

- Daha büyük veri kümeleri için daha uzun eğitim süresi.

Hiperparametreler

- **Soft Margin Sabiti (C):**
  Aykırı değerler üzerindeki ceza düzeyine karar veren bir hiperparametredir. Düzenlileştirme parametresinin tersidir. C büyük olduğunda, Outliers'a yüksek ceza verilecek ve sert bir marj oluşacaktır. C küçük olduğunda, aykırı değerler ihmal edilir ve marj geniş olur.

- **Polinom Çekirdeğindeki (d) polinom derecesi:**
  d = 1 olduğunda, doğrusal bir çekirdeğe eşdeğerdir. D daha yüksek olduğunda, çekirdek karmaşık kalıpları yeni bir hiper düzleme yansıtarak ayırt edecek kadar esnektir.

- **Gauss Çekirdeğindeki Genişlik Parametresi (γ):**
  Gama, Gauss eğrisinin genişliğine karar verir. Gama artışı ile genişlik de artar.

Diğer Modellerle Karşılaştırma

SVM vs Karar Ağaçları

- Random Forest çok sınıflı sınıflandırmayı desteklerken, SVM'nin bunun için birden fazla modele ihtiyacı vardır.
- Rastgele Orman, tahmin üzerinde bir olasılık verebilir, oysa SVM veremez.

- Random Forest, kategorik verileri SVM'den daha iyi ele alır.

SVM vs Naive Bayes

- Her ikisi de düşük miktarda eğitim verisi ve büyük özelliklerle daha iyi performans gösterir.
- Özellikler karşılıklı olarak bağımlıysa SVM, Naive Bayes'ten daha iyi performans gösterir.
- SVM ayırt edici bir modeldir, NB ise üretken modeldir.

SVM vs Yapay Sinir Ağları (NN)

- SVM'nin dışbükey bir optimizasyon işlevi vardır, oysa NN yerel minimumda asılı olabilir.
- SVM, sınırlı eğitim verisi ve birçok özellik olduğunda NN'den daha iyi performans gösterebilir. NN, yeterli doğruluk için büyük eğitim verilerine ihtiyaç duyar.
- Çok sınıflı sınıflandırma, SVM için birden fazla model gerektirirken, NN bunu tek bir modelle yapabilir.

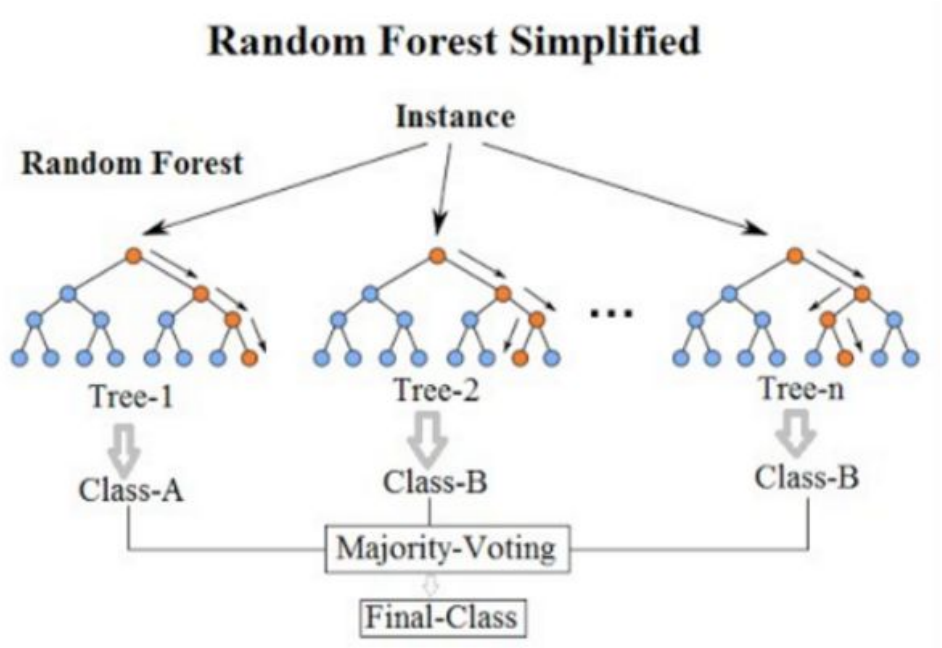
6- Rastgele Orman (Random Forest)

Random Forest, daha güçlü bir model elde etmek için çoklu karar ağaçlarının birleştirildiği bir topluluk modelidir. Türetilen model daha sağlam, doğru olacak ve aşırı uyumu kurucu modellerden daha iyi idare edecektir.

Temel Teori

Random Forest, sınıflandırma ve regresyon çıktıları elde etmek için " bagging method" ile birleştirilmiş bir dizi karar ağacına sahiptir. Sınıflandırmada, çıktıyı çoğunluk oylamasını kullanarak hesaplarken, regresyonda ortalama hesaplanır.

Random Forest, ikili, kategorik, sürekli özelliklerle çok çeşitli giriş verilerini işleyebilen sağlam ve doğru bir model ortaya çıkarır.



Kayıp İşlevi

Veri setlerinin kayıp değerini hesaplamak için entropi / Gini skorunu kullanırız..

Avantajları

- Doğru ve güçlü model.
- Aşırı uydurmayı verimli bir şekilde ele alır.
- Örtük özellik seçimini destekler ve özellik önemini türetir.

Dezavantajları

- Orman büyüdüğünde hesaplama açısından karmaşık ve daha yavaş.
- Tahmin üzerinde iyi tanımlayıcı bir model değil.

Hiperparametreler

- n_estimators:
  Ormandaki ağaçların sayısıdır. Çok sayıda ağaçla birlikte yüksek doğruluk, ancak yüksek hesaplama karmaşıklığı gelir.
- maksimum özellikler:
  tek bir ağaçta izin verilen maksimum özellik sayısı.
- minimum numune yaprağı :
  Dahili bir düğümü bölmek için gereken minimum numune sayısıdır.

Diğer Modellerle Karşılaştırma

Rastgele Orman karşılaştırması, Karar ağacı karşılaştırmalarına oldukça benzer.

Rastgele Orman vs Naive Bayes

- Rastgele Orman karmaşık ve büyük bir modelken, Naive Bayes nispeten daha küçük bir modeldir.
- Naive Bayes, küçük eğitim verileriyle daha iyi performans gösterirken, RF'nin daha büyük eğitim verileri setine ihtiyacı vardır.

Rastgele Orman vs Yapay Sinir Ağları(NN)

- Her ikisi de çok güçlü ve yüksek doğruluklu algoritmalardır.
- Her ikisinin de dahili olarak özellik etkileşimleri vardır ve daha az açıklanabilir.
- Random Forest özellik ölçeklendirmeye ihtiyaç duymazken, NN özelliklerin ölçeklendirilmesine ihtiyaç duyar.
- Her iki modelin toplu versiyonu güçlü olacak.

7- Naive Bayes

Naive bayes, sınıflandırma problemleri için kullanılan üretken bir olasılık modelidir. Özellik setinin çok büyük olduğu metin sınıflandırmaları için kullanılan ana modeldir. Duygu analizi, spam filtreleme vb. İçin yaygın olarak kullanılmaktadır.

Temel Teori

Naive bayes modeli Thomas Baye'nin bayes kuralına dayanmaktadır. Bayes kuralı şu şekilde ifade edilebilir:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

Yukarıdaki denklemde,

- P (A | B): (son olasılık) B olayı doğru olduğunda A olayının olma olasılığı.
- P (A), P (B): A olayının ve B olayının gerçekleşme olasılığı.
- P (B | A): (olasılık) A olayı doğru olduğunda B olayının gerçekleşme olasılığı.

Temel mantık, eğitim verilerinden Y olarak çıktı etiketi verilen özelliklerin (Xi) ayrı olasılıklarından X girdisine verilen çıktı etiketinin (Y) olasılığını türetmektir.

$$P(C_i|x_1, x_2, \ldots, x_n) = \frac{P(x_1, x_2, \ldots, x_n|C_i).P(C_i)}{P(x_1, x_2, \ldots, x_n)} \text{ for } 1 \leq i \leq k$$

Lütfen yukarıdaki denklemde naif bayes'ın tüm özelliklerin bağımsız olduğunu varsaydığına dikkat edin. "Naive" kelimesinin kendisi bunu hatırlatmak için kullanılır. Birden fazla sınıf etiketi olması durumunda, her etiket için P (Ci | X) hesaplanır ve çıktı olarak maksimum olasılığa sahip etiket seçilir.

Naive bayes teoreminin aşağıda listelendiği gibi birkaç alternatifi vardır:

- Gauss: Gauss'a özgü saf koylar, özelliklerin Gauss dağılımını takip ettiğini varsayar.
- Multinomial: Multinomial naive bayes dağılımı, verilerin multinom dağılımda olduğu varsayıldığında kullanılır.