

Decision Tree Uygulama Alanları

- Sınıflandırma Problemleri (Kategorik variable)
- Regresyon Problemleri (Numerik variable) ^{kategorik ve sayısal değişken}
- Pazarlama ve Satış Analizi
- Risk Değerlendirmesi
- Sağlık ve Tıp
- Endüstriyel Defekt ve Kalite Kontrol

Avantajları

Anlaşılabilirlik

Değerli önemi

Gözlu Güzışli Modeller Uygulama

Veri ön işleme ihtiyacının Azalması

Kategorik değişkenler de ele alınabilir.

Dezavantajları

Aşırı uyum (overfitting)

Yavaşlık

Dengesiz Veri kümeleri

Rasgele Orman - Random Forest

• Birçok karar ağacının bir araya gelerek oluşturduğu topluluk türü. (ensemble learning (topluluk ağ))

• Her bir karar ağacı farklı bir örneklem ve veri alt özellikleri kullanılarak eğitilir

(100 sample dan 80 tane olup 1 ağac, 70 tane sample 2 ağacı, 90 tane olup 3 ağacı eğitebiliriz)

Her bir ağacın elimizde bulunan featureların hepsini kullanılarak eğitilir)

• Sınıflandırma, Regresyon

• Decision Tree lere meydana gelen overfitting gibi dezavantajları ortadan kaldırmak için ağırmıştır.

Reggele Oran Hiperparametreler

- 1) Ağac sayısı (n-estimators) : Model daha seilir,
- 2) Öznitelik sayısı (max-features) : Her ağac için seçilen sayı
max(10) ke 8, 5, 4, 9 danc gibi seçebilm
- 3) Ağac Derinliği (max-depth) : Overfitting'in sayısı önemli.

Uygulama Alanları

• Sınıflandırma ve regresyon problemleri.

Yayıp Yöntem

Güçlü Yöntem

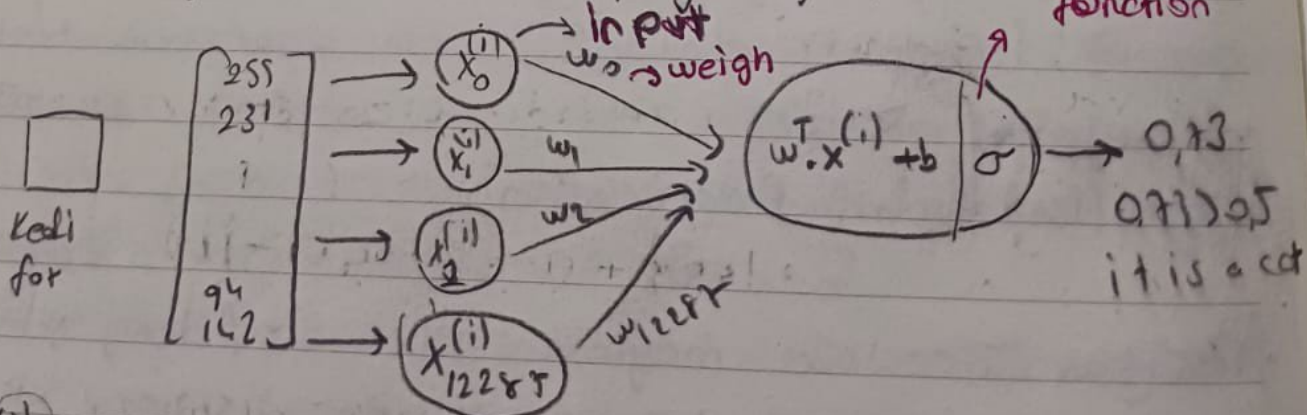
- Eğitim süreci yavaş olabilir. • Overfitting'e karşı dirençli
- Yüksek ve düşük boyutlu veri ve genelleme yeteneği yüksek
- Aynı performansı sağlar. • Özelliklerin önem derecesini belirleme yeteneğine sahip.

Lojistik Regresyon

- İstatistik ve makine öğrenme alanlarında sınıflandırma problemleri çözmek için kullanılan bir yöntemdir. Genellikle kategorik değişkenlerde kullanılır. (ör: Bir müşterinin bir ürünü satın alıp almadığı olasılığını tahmin etmek gibi durumlarda kullanılır)

• İleri Yayılm (Forward Propagation) ?

Logistic Sigmoid function



$x^{(i)}$ → feature (pixel)
1. kedi 1. nesne

$$\begin{bmatrix} 1 & 1 \\ 3 & 24 \end{bmatrix} \Rightarrow \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 3 \\ 4 \end{bmatrix}$$

w : weight } parametrelerdir. Training işlemi için
 b : bias } bu parametreler optimize edilmediği için

$w \cdot x^{(i)} + b = z$ buradaki transpose işlemi matris
 çarpımını ↓ gerçekleştirilebiliriz için yapıyoruz
 bias

σ : Bu logistic sigmoid fonksiyon sayesinde input
 ve weigh çarpımından çıkan değeri $[0, 1]$ aralığına
 çevirebiliriz. (2 yi sokarsak)

ileri yayılım : inputın verilmesi, inputların x e eşitlenmesi
 w ile çarpılması ve bias ile toplanması. Çıkan z 'nin
 sigmoid fonksiyonu sonucu 0 ve 1 arasında bir
 % çıkarmasıdır.

Lojistik Regresyon Eğitimi

• w, b parametrelerinin belirlenmesidir.

1) Ağırlık ve biasların başlatılması

2) İleri Yayılım (Forward Propagation)

3) Loss (maliyet, kayıp) fonksiyonunun hesaplanması
 $\rightarrow \hat{y} = \sigma(z)$

$\rightarrow L(\hat{y}, y) \rightarrow$ Burda merbinden çıkan sonucu gerçek
 sonuç ile karşılaştırarak formül hesaplanır.

$$L(\hat{y}, y) = - (y \log \hat{y} + (1-y) \log (1-\hat{y}))$$

4) Geri Yayılım (Backward Propagation) : Türev
 hesaplanabilir. yapıcı kayıp fonksiyonuna göre türev
 alarak w ve b değerlerini optimize etmeye
 çalışır. Loss fonksiyonun min noktasına (0) a yaklaştı-
 rmaya çalışır.

• Yeni loss'un w ve b ye göre kısmi türevlerini alalım.
 5) Ağırlıkların güncellenmesi: Gradient descent algoritması ile w ve b güncellenir.

$$\frac{\partial L}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial z} \frac{\partial z}{\partial w_1} \rightarrow \frac{\partial L}{\partial w_1} \text{ e göre türevini bulalım. (Geri yayılım)}$$

▼ Hatırlatma

$$\frac{d \log x}{dx} = \frac{1}{x} \text{ 'tir. } \leftarrow \text{türev}$$

$$L(\hat{y}, y) = - (y \log \hat{y} + (1-y) \log (1-\hat{y})) \rightarrow L(\hat{y}, y)$$

$$\frac{\partial L}{\partial \hat{y}} = \frac{-y}{\hat{y}} + \frac{1-y}{1-\hat{y}}$$

$$\sigma = \frac{1}{1+e^{-z}} \rightarrow \frac{\partial \hat{y}}{\partial z}$$

$$\frac{\partial z}{\partial w_1} = x_1$$

$$\hat{y}^2 e^{-z}$$

$$\frac{\partial L}{\partial w_1} = (\hat{y} - y) x_1$$

$$\frac{\partial L}{\partial b} = (\hat{y} - y)$$

5) gradient descent: $w_1 \leftarrow w_1 - \alpha (\hat{y} - y) x_1$
 $b \leftarrow b - \alpha (\hat{y} - y)$
 adım büyüklüğü \leftarrow hyper parametre hızla abakalı