

MAKİNE ÖĞRENMESİ

Denetimli Öğrenme

Denetimli öğrenmede algoritma eğitmek için etiketli veri kümesi kullanılır . Yani hangi verinin hangi bilgiye karşılık geldiği bilinmektedir. Etiketli veriler , her girdiğinin doğru çıktısının bilindiği veri setleridir. Algoritmanın olası çıktıları halihazırda bilindiğinden dolayı algoritmayı eğitmek için kullanılan veriler de doğru cevaplarla etkilenmiştir.

Denetimli öğrenme iki ana alt kategoriye ayrılır :

- **Sınıflandırma Problemleri**

Sınıflandırma problemlerinde model , girdiyi belirli kategorilere ayırmayı öğrenir. Girdi verilerini ayrık değerdeki çıktı verisine eşleyen problemlere uygulanmaktadır; yani çıktı verileri kategorik tiplerdir.

Örnekler : Bir e-posta sınıflandırma modelinde , e-postaların spam ya da spam olmadığını belirlemek , hisse senedi fiyatlarını tahmin etmek , bankalarda kredi riskli değerlendirmesi , makine arızası tespiti.

Sınıflandırma metrikleri , makine öğrenimi modellerinin performansını değerlendirmek , karşılaştırmak ve iyileştirmek için kullanılır . İşte bazı sınıflandırma metrikleri :

1. Doğruluk (Accuracy)

Veri bilimi projelerinde en doğru modelin hangisi olması gerektiğine karar vermek için iş birimlerinden gelen talepleri iyi değerlendirmemiz gereklidir.

Accuracy bir modelin başarısını ölçmek için çok kullanılan ancak tek başına yeterli olmadığı görülen bir metriktir.

Accuracy değerini dikkate almak her sınıfta eşit sayıda gözlem varsa mantıklı olabilir. Bunun dışında oldukça yanıltıcıdır.

Doğruluk bir modelde nasıl hesaplanır ?

$$\text{Doğruluk} = \frac{\text{Doğru Tahminler}}{\text{Tüm Tahminler}} = \frac{TP + TN}{TP + TN + FP + FN}$$

Toplam 4 ihtimal vardır. Bunu önce “Yabancı Çoban” hikayesi ile anlatmak benim için daha öğretici oldu .Çoban, eğittiğimiz modele, kurt ise bağımlı değişkenimizin sınıfına karşılık geliyor .

- 1- **True Positive (TP)** : Modelimiz (Çoban) “Kurt var!” der ve gerçekte de kurt vardır. Modelimizin doğru tahmin ettiği ihtimallerden biridir. (1-1)
- 2- **True Negative(TN)** : Modelimiz “Kurt yok!” der ve kurt yoktur. Modelimizin doğru tahmin ettiği ihtimallerden biridir.(0 – 0)
- 3- **False Positive(FP)** : Modelimiz “Kurt var!” der ama kurt yoktur. Modelimizin yanlış tahmin ettiği ihtimallerden biridir.(1 – 0)
- 4- **False Negative(FN)** : Modelimiz “ Kurt yok!” der ama kurt vardır . Modelimizin yanlış tahmin ettiği ihtimallerden biridir. (0 – 1)

Özellikle eşit dağılmayan unbaised veri kümelerinde model doğruluğu tek başına yeterli değildir. Örneğin kanser olan ve olmayan hastaların olduğu 100 kişilik bir veri kümemiz olduğunu düşünelim . Tüm hastalar içinde sadece 10 tanesinde kanser teşhisi konulmuştur. Böyle bir durumda kanser olan ancak teşhis edilemeyen (False Negative) hastalar olmasını istemeyiz Bu nedenle diğer metriklerin sonuçlarını birlikte değerlendirmeliyiz.

- Sınıflar dengeliyse ve hataların maliyeti eşitse kullanılır.

2. Kesinlik (Precision)

Pozitif olarak tahminlediğimiz değerlerin gerçekten kaçta kaçının pozitif olduğunu göstermektedir .

$$Precision = \frac{TP}{TP + FP}$$

Örneğin:

- Toplam 100 e-posta geldi..
- Modelin 20 tanesini spam olarak işaretledi.
- Bu 20 e-postadan 15 tanesi gerçekten spam çıktı (TP)
- Ama 5 tanesi aslında spam değildi, yanlışlıkla spam kutusuna gitti (FP)

- **Precision**= $TP+FP/TP = 15/(5+15) = 15/20=0.75$
- Yani spam klasöründeki maillerin gerçekten %75'i gerçekten spam . Ama %25'i yanlışlıkla oraya atılmış (hatalı tahmin).
- Önemli olan bir e-postayı yanlışlıkla spam olarak işaretlerse , bu precision düşüklüğüne işaret eder. Bu durum, bizim için önemli e-postaların kaybolmasına yol açar. Bu nedenle precision değerinin yüksek olması , model seçiminde önemli bir kriterdir.
- Yanlış pozitiflerin maliyeti yüksekse kullanılır.

3. Duyarlılık (Recall)

Recall pozitif olarak tahmin etmemiz gereken işlemlerin ne kadarını pozitif olarak tahmin ettiğimizi gösteren bir metriktir.

Recall , yanlış pozitiflerin maliyeti yüksekse kullanılır.

$$\text{Recall} = \frac{TP}{TP + FN}$$

Mümkün olduğunca yüksek olması gereklidir.

Havalimanında Bavul Kontrolü Örneği :

Bir güvenlik sistemi , gelen bavulları tarayarak içinde tehlikeli madde olup olmadığını tespit ediyor.

TP: 40 (Tehlikeli denildi ve gerçekte tehlikeli çıktı)

FN : 10 (Temiz dendi ama tehlikeli çıktı)

Recall: $TP / (TP+FN) = 40 / 40+10 = 40 / 50 = 0.80 = \%80$

Yani gerçekten tehlikeli olan 50 bavuldan 40 tanesini tespit ettik , ama 10 tanesi kaçtı. Sistem bazı tehlikeli bavulları kaçırıyor (FN) , bu büyük bir güvenlik açığı .

Bu nedende burada recall değeri çok önemlidir çünkü tehlikeli bir şeyi atlamak, büyük zarara yol açabilir.

4. F1 Score

F1 Score değeri bize precision (kesinlik) ve recall (duyarlılık) değerlerinin harmonik ortalamasını göstermektedir.

Basit bir ortalama yerine harmonik ortalama olmasının sebebi ise uç durumları da gözardı etmemiz gerektiğidir. Eğer basit bir ortalama hesaplaması olsaydı Precision değeri 1 ve Recall değeri 0 olan bir modelin F1 Score'u 0.5 gelecektir ve bu bizi yanıltacaktır.

Precision ve recall ikisi de önemliyse , veri dengesizse kullanılır.

$$F_1 = 2 * \frac{precision * recall}{precision + recall}$$

Accuracy yerine F1 Score değerinin kullanılmasının en temel sebebi eşit dağılmayan veri kümelerinde hatalı bir model seçimi yapmamaktadır. Ayrıca sadece False Negative ya da False Positive değil tüm hata maliyetlerini içerek bir ölçme metriğine ihtiyaç duyulduğu içinde F1 Score bizim için çok önemlidir.

- NOT :
Precision daha önemlidir , eğer yanlış yere “evet” demek büyük sorun yaratıyorsa. Recall daha önemlidir, eğer “evet” olması gereken bir şeyi atlamak büyük sorun yaratıyorsa.

5. AUC (Area Under Curve)

Modelin sınıfları ne kadar başarılı ayırt edebildiğini anlatır. AUC, arttıkça, model 0'ları 0 ve 1'leri 1 olarak tahmin etmede daha iyidir. Eğri altında kalan alan olarak geçer.

Örneğin hasta olan ve olmayan kullanıcıların olduğu bir veri setinde AUC ne kadar yüksekse, model hastalığı olan ve olmayan hastaları ayırt etmekte daha iyi performans gösterir.

AUC Skoru 1'e ne kadar yakınsa , modelin performansı o kadar iyidir.

Aşağıdaki grafiklerde görüldüğü gibi AUC değeri yükseldikçe yeşil ve kırmızı olan sınıflar birbirinden iyi bir oranda ayrılmaktadır. Son grafikte görülen AUC değeri 0.5 olan modelin performansının kötü olduğunu ve rastgele tahminleme yaptığı söylenebilir.

6. ROC (Receiver Operating Characteristic)

ROC eğrisi sınıflandırma problemleri için çok önemli bir performans ölçümüdür. İşlem karakteristik eğrisi olarak da geçer.

ROC eğrisinde X ekseninde FPR(Yanlış Pozitif Oran) ve Y ekseninde ise TPR(Gerçek Pozitif Oranı) bulunmaktadır.(TPR=Recall)

ROC bir olasılık eğrisidir ve altında kalan alan AUC ayrılabilirliğinin derecesini veya ölçüsünü temsil eder

- AUC VE ROC EĞRİSİ MODEL BAŞARISI

```
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import roc_curve, roc_auc_score

X, y = make_classification(n_samples=1000, n_features=10, n_classes=2, random_state=42)

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)

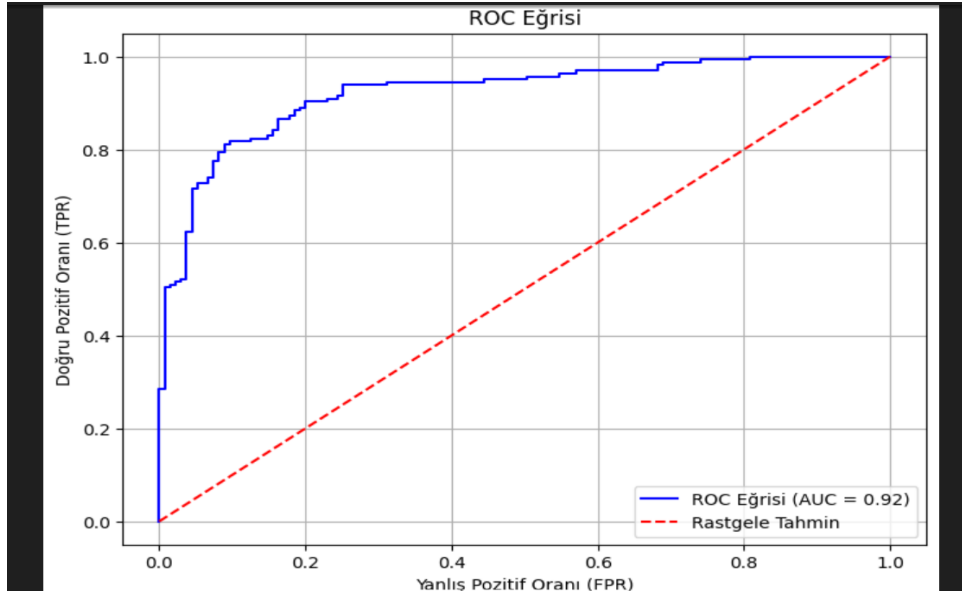
model = LogisticRegression()
model.fit(X_train, y_train)

y_pred_prob = model.predict_proba(X_test)[:, 1]

fpr, tpr, thresholds = roc_curve(y_test, y_pred_prob)
auc_score = roc_auc_score(y_test, y_pred_prob)

print(f"AUC Skoru: {auc_score:.4f}")

# ROC eğrisi
plt.figure(figsize=(8, 6))
plt.plot(fpr, tpr, color='blue', label=f'ROC Eğrisi (AUC = {auc_score:.2f})')
plt.plot([0, 1], [0, 1], color='red', linestyle='--', label='Rastgele Tahmin')
plt.xlabel('Yanlış Pozitif Oranı (FPR)')
plt.ylabel('Doğru Pozitif Oranı (TPR)')
plt.title('ROC Eğrisi')
plt.legend()
plt.grid(True)
plt.show()
```



AUC değeri 1'e ne kadar yakınsa model o kadar iyidir.

0.5 rastgele tahmin anlamına gelir.

Eğri sol üst köşeye ne kadar yakınsa model o kadar iyi performans gösteriyor demektir.

7. Hata Matrisi (Confusion Matrix)

Veri setindeki var olan durum ile sınıflandırma modelimizin doğru ve yanlış tahminlerinin sayısını tablo olarak göstermektedir.

	positive	negative
positive	TP = 20	FN = 24
negative	FP = 20	TN = 72

REGRESYON PROBLEMLERİ

1-Mean Absolute Error(MAE)

Ortalama mutlak hata iki sürekli değişken arasındaki farkın ölçüsüdür.

MAE, yönlerini dikkate almadan bir dizi tahmindeki hataların ortalama büyüklüğünü ölçen, tüm tekil hataların ortalamada eşit olarak ağırlıklandırıldığı doğrusal bir skordur.

Hataların mutlak büyüklüğünü eşit şekilde cezalandırmak istediğimizde kullanılır.

MAE'nin Özellikleri

Yorumu Kolay: Hata birimi orijinal veriyle aynı birimdedir

Aşırı Değerlere Duyarsız: MSE'ye göre aykırı değerlerden daha az etkilenir

Pozitif Değer: Mutlak değer kullanıldığı için her zaman pozitifdir

Küçük Değer İyidir: 0'a ne kadar yakınsa model o kadar iyidir

$$MAE = \frac{1}{n} \sum_{j=1}^n |y_j - \hat{y}_j|$$

2-Mean Squared Error(MSE)

Basitçe, ortalama kare hata bir regresyon eğrisinin bir dizi noktaya ne kadar yakın olduğunu söyler. MSE, bir makine öğrenmesi modelinin, tahminleyicinin performansını ölçer, her zaman pozitif değerlidir ve MSE değeri sıfıra yakın olan tahminleyicilerin daha iyi bir performans gösterdiği söylenebilir.

Büyük hataları aşırı cezalandırmak ve model eğitimi sırasında optimizasyon yapmak istediğimizde kullanılır.

$$MSE = \frac{1}{n} \sum \left(\underbrace{y - \hat{y}}_{\substack{\text{The square of the difference} \\ \text{between actual and} \\ \text{predicted}}} \right)^2$$

MAE: Dengeli hata ölçümü (robust)

MSE: Büyük hatalara odaklı

3-Root Mean Square Error (RMSE)

Regresyon modellerinde tahmin hatalarının ortalama büyüklüğünü ölçer. Büyük hataları daha sert cezalandırır (çünkü hataların karesi alınır).

MSE'nin birim tutarlı halini sunarak büyük hataları vurgulamak istediğimizde kullanılır.

$$RMSE = \sqrt{MSE}$$

4-R-KARE (R²)

R-Kare (R²), bir regresyon modelinin bağımlı değişkenin varyansını ne kadar iyi açıkladığını gösteren bir metriktir. 0 ile 1 arasında değer alır.

Modelin bağımlı değişkenin varyansını ne kadar iyi açıkladığını ölçmek istediğimizde kullanılır.

$$R^2 = 1 - \frac{\text{Açıklanamayan Varyans}}{\text{Toplam Varyans}} = 1 - \frac{\sum (y_{\text{gerçek}} - y_{\text{tahmin}})^2}{\sum (y_{\text{gerçek}} - \bar{y})^2}$$

Ek sorular :

- RMSE daha önemlidir, çünkü gerçek dünyada anlamlı bir metriktir. MSE'nin karekökünü alarak hatayı orijinal birimle (TL, kg vb.) ifade eder ve büyük hataları yine cezalandırırken, yorumlaması kolaydır.
- F1 Skoru, dengesiz verilerde ve hata maliyetleri farklı olduğunda daha güvenilirdir.
- Hata toleransının düşük olduğu durumlarda RMSE kullanılır, çünkü büyük hataları aşırı cezalandırarak modelin kritik sapmalardan kaçınmasını sağlar.

KAYNAKÇA

[MSE, RMSE, MAE, MAPE ve Diğer Metrikler – Veri Bilimcisi – YENİLENİYOR ...](#)

[Adım Adım Makine Öğrenmesi Bölüm 1: Makine Öğrenmesi Nedir? | by Hatice Candan | Machine Learning Türkiye | Medium](#)

[Denetimli ve Denetimsiz Öğrenme: Veri Biliminin İki Temel Taşı | Patika.dev](#)

[Doğruluk \(Accuracy\) , Kesinlik\(Precision\) , Duyarlılık\(Recall\) ya da F1 Score ? | by Gülcan Öğündür | Medium](#)

[ROC Eğrisi ve AUC Değeri – Bilişim IO](#)

ChatGpt

DeepSeek