Mustafa Erdogan
I am Data Scientist and I have more than 4 years of experience in applications of Artificial Intelligence, and Machine Learning.
31 Followers
Follow

Mustafa Erdogan
Follow
Published in
Academy Team
7 min read
Apr 14
38
1

Makine öğrenme modellerindeki en temel hususun modelin performansının değerlendirilmesinden hareketle bu yazımızda özellikle makine öğrenmesinde sınıflandırma (classification) modellerinin performansını ölçmek için sıklıkla kullanılan metrikler örnekler yoluyla açıklanmıştır. Şimdi gelelim sınıflandırma modellerindeki başlıca metriklere.

Doğruluk (Accuracy): Modelin doğru tahmin ettiği örneklerin oranıdır. Doğruluğun maksimum değeri 1 olabilir. Örneğin, 100 örneğin 80'inin doğru sınıflandırıldığı bir modelin doğruluk skoru 0.8 olacaktır. Aşağıdaki formülden de anlaşılacağı üzere doğruluk doğru bilinen tahminlerin tüm tahminlere oranı olarak ifade edilebilir.

Formül:

Doğruluk = (TP + TN) / (TP + TN + FP + FN)

Burada;

TP (True Positive): doğru pozitif sayısı

TN (True Negative): doğru negatif sayısı

FP (False Positive): yanlış pozitif sayısı

FN (False Negative): yanlış negatif sayısıdır.

Örnek: 100 örneğin 80'ini doğru tahmin eden bir modelin doğruluğu (Accuracy) %80'dir.

Hassasiyet (Precision): Pozitif olarak sınıflandırılan örneklerin ne kadarının gerçekten pozitif olduğunu gösterir. Hassasiyet modelin pozitif sınıfı doğru sınıflandırma yeteneğini ölçmektedir. Hassasiyet skoru, yanlış pozitif sınıflandırmaların sayısını (yanlış pozitifler) gerçek pozitif sınıflandırmaların sayısına (doğru pozitifler) oranlaması ile hesaplanmaktadır. Aşağıdaki formülden de anlaşılacağı üzere hassasiyet pozitif olarak doğru bilinen tahminlerin tüm pozitif tahminlere oranı olarak ifade edilebilir.

Formül:

Hassasiyet = TP / (TP + FP)

TP (True Positive): doğru pozitif sayısı

FP (False Positive): yanlış pozitif sayısı

Örnek: 100 örneğin 80'ini pozitif olarak sınıflandıran bir modelin %75 hassasiyeti varsa, 60 örneğin gerçekten pozitif olduğunu söyleyebiliriz.

Duyarlılık (Recall): Gerçek pozitif örneklerin ne kadarının pozitif olarak sınıflandırıldığını gösterir. Aşağıdaki formül incelendiğinde duyarlılık, pozitif olarak doğru tahmin edilenlerin gerçek pozitiflere oranı olarak ifade edilebilir.

Formül:

Duyarlılık = TP / (TP + FN)

TP (True Positive): doğru pozitif sayısı

FN (False Negative): yanlış negatif sayısıdır.

Örnek: 100 gerçek pozitif örneğin 80'ini doğru olarak sınıflandıran bir modelin %80 duyarlılığı vardır.

F1 Skor (F1 Score): F1 Skor ise Hassasiyet (Precision) ve Duyarlılık (Recall) skorlarının harmonik ortalamasıdır.

Formül:

F1 Skor = 2 * ((Precision * Recall) / (Precision + Recall))

Precision: Hassasiyet

Recall: Duyarlılık

Örnek: 0,8 hassasiyeti ve 0,8 geri çağırması olan bir modelin F1 puanı 2*0,8x0,8/(0,8+0,8) = 0,8'dir.

Log Kaybı (Log Loss): Modelin tahmin ettiği olasılıkların gerçek olasılıklardan ne kadar uzak olduğunu ölçer. Model gerçek değerin olasılığına yakın bir tahminde bulunursa log kaybı düşük değer alırken gerçek değerin olasılığından uzaklaştıkça artan bir değer verir.

Formül:

Log Kaybı = -(1/N) * Σ(y * log(p) + (1-y) * log(1-p))

N: gözlem sayısı

y: gerçek değer olasılığı

p: ise tahmini değer olasılığı

Log kaybı (Log loss), gerçek değer olasılığına yakın bir tahminde daha düşük bir değer ve gerçek değerden uzaklaştıkça artan bir değer vermektedir.

Örnek: Bir örneğin gerçek değeri "A" ise ve model "A" için olasılık 0,8 tahmin ederken, gerçek olasılık 1 ise log kaybı -log(0,8) = 0,2231'dir.

Tüm bu işlemler için scikit-learn kütüphanesi, sınıflandırma modellerinde kullanılan bu metriklerin hesaplanmasını kolaylaştırır. Aşağıda, sınıflandırma modeli sonuçlarının hesaplanması için scikit-learn kütüphanesinin kullanımına ilişkin örnek kodlar verilmiştir.

```
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score, log_loss

# Gerçek sınıf etiketleri (y_true) ve tahmin edilen sınıf etiketleri (y_pred) örnek olarak verilmiştir.

y_true = [1, 0, 1, 1, 0]

y_pred = [1, 0, 1, 1, 1]

# Doğruluk (Accuracy) hesaplama:

acc = accuracy_score(y_true, y_pred)

print("Doğruluk:", acc)

# Hassasiyet (Precision) hesaplama:
```

```
prec = precision_score(y_true, y_pred)

print("Hassasiyet:", prec)

# Duyarlılık (Recall) hesaplama:

rec = recall_score(y_true, y_pred)

print("Duyarlılık:", rec)

# F1 Puanı (F1 Score) hesaplama:

f1 = f1_score(y_true, y_pred)

print("F1 Skor:", f1)

# Log Kaybı (Log Loss) hesaplama:

# Gerçek ve tahmin edilen sınıf olasılıkları örnek olarak verilmiştir.

y_true_probs = [[0.7, 0.3], [0.2, 0.8], [0.6, 0.4], [0.8, 0.2], [0.1, 0.9]]

y_pred_probs = [[0.9, 0.1], [0.4, 0.6], [0.8, 0.2], [0.1, 0.9], [0.2, 0.8]]

logloss = log_loss(y_true, y_pred_probs)

print("Log Kaybı:", logloss)

Output :

Doğruluk: 0.8

Hassasiyet: 0.75
```

```
Duyarlılık: 1.0


F1 Skor: 0.8571428571428571


Log Kaybı: 1.3086224330788454
```

Micro avg: Bu skor, hedef değişkende (target variable) tüm sınıfların gerçek pozitiflerinin toplamının tüm sınıfların tahmin edilen pozitiflerinin toplamına oranı şeklinde hesaplanmaktadır. Özellikle hedef değişkenin çoklu sınıf (**multi class**) olduğu durumlarda doğruluk (**Accuracy**) ölçüsü olarak kullanılır.

Formül (3 class için):

Micro avg = (TP1 + TP2 + TP3) / (TP1 + TP2 + TP3 + FP1 + FP2 + FP3)

Burada

TP1, TP2 ve **TP3**, sırasıyla sınıfların gerçek pozitif değerlerinin sayısını,

FP1, FP2 ve **FP3** ise sırasıyla sınıfların yanlış pozitif değerlerinin sayısını temsil eder.

Macro avg: Bu skor, hedef değişkende (target variable) mevcut her bir sınıfın hassasiyet ve duyarlılık değerlerinin ortalamasını alarak hesaplanmaktadır. Özellikle hedef değişkenin çoklu sınıf (**multi class**) olduğu durumlarda bir sınıfın diğer sınıflara kıyasla daha fazla önemli olup olmadığını belirlemek için kullanılır.

Formül (3 class için):

Macro avg = (PrecisionA + PrecisionB + PrecisionC) / 3

Weighted avg: Bu skor, hedef değişkende (target variable) mevcut her bir sınıfın hassasiyet ve duyarlılık değerlerinin ağırlıklı ortalamasını alır. Ağırlıklar, sınıf boyutlarına göre belirlenir.

Formül (3 class için):

Weighted avg = (n1 * PrecisionA + n2 * PrecisionB + n3 * PrecisionC) / (n1 + n2 + n3)

Burada

n1, n2 ve n3, sırasıyla sınıflar için gözlem (observation, row, vs.) sayısını temsil eder.

Tüm bu işlemler için aşağıda, sınıflandırma modeli sonuçlarının hesaplanması için scikit-learn kütüphanesinin kullanımına ilişkin örnek kodlar verilmiştir.

```
from sklearn.metrics import precision_score, recall_score, f1_score


from sklearn.metrics import precision_recall_fscore_support as score
```

```
# Tahmin ve gerçek değerleri veriler olarak tanımlayalım

y_true = ["A", "A", "A", "B", "B", "B", "B", "C", "C", "C", "C", "D", "D", "D", "D", "D"]

y_pred = ["A", "A", "B", "B", "B", "B", "C", "C", "C", "C", "D", "A", "D", "D", "D", "D"]

# Micro avg skoru

micro_precision, micro_recall, micro_fscore, _ = score(y_true, y_pred, average='micro')

print("Micro avg precision: ", micro_precision)

print("Micro avg recall: ", micro_recall)

print("Micro avg F1 score: ", micro_fscore)

# Macro avg skoru

macro_precision, macro_recall, macro_fscore, _ = score(y_true, y_pred, average='macro')

print("Macro avg precision: ", macro_precision)

print("Macro avg recall: ", macro_recall)

print("Macro avg F1 score: ", macro_fscore)

# Weighted avg skoru

weighted_precision, weighted_recall, weighted_fscore, _ = score(y_true, y_pred, average='weighted')

print("Weighted avg precision: ", weighted_precision)

print("Weighted avg recall: ", weighted_recall)
```

```
print("Weighted avg F1 score: ", weighted_fscore)


Output:


Micro avg precision:  0.625


Micro avg recall   :  0.625


Micro avg F1 score :  0.625


Macro avg precision:  0.6444444444444444


Macro avg recall   :  0.6555555555555555


Macro avg F1 score :  0.6446527777777777


Weighted avg precision:  0.6366666666666667


Weighted avg recall   :  0.625


Weighted avg F1 score :  0.6269444444444445
```

Peki Bu Skorlar Neye Göre ve Nasıl Yorumlanmalıdır?

Elde edilen bu sınıflandırma skorları, modelin performansı hakkında fikir vermektedir. Bu skorlar, modelin sınıflandırma yeteneğinin hangi ölçüde doğru olduğunu ve hangi sınıfların daha iyi tahmin edildiğini göstermektedir.

Hangi sınıflandırma skorunun daha önemli olduğu, uygulamaya ve veri setine göre değişmektedir. Yukarıda açıkladığımız üzere **Accuracy** (**Doğruluk**), özellikle hedef değişkendeki (target variable) sınıflar arasında bir denge var ise ve skorlar arası bir uyum söz konusu ise kullanılmaktadır. Dolayısıyla hedef değişkendeki sınıf dengesi değişken olduğunda yanıltıcı olabilmektedir. Bu sebeple precision, recall ve F1 skorların kullanılması önerilmektedir. Bu nedenle, bir modelin performansını değerlendirmek için, sınıf dağılımına bakmak ve özellikle sınıf dengesi değişken olduğunda **precision**, **recall** ve **F1** skorun yanı sıra **accuracy** skoruna da bakmak önemli olmaktadır.

Bu skorlardan Duyarlılık (**Recall)** skoru , modelin pozitif sınıfı doğru bir şekilde tespit etme yeteneğini ölçmektedir. Bu sebeple **Recall** skoru yanlış negatiflerin kabul edilemeyeceği uygulamalarda önemli hale

gelebilmektedir. Örneğin, hastalık teşhisinde kullanılan bir sınıflandırma modeli düşünelim. Bu durumda, yanlış negatif tahminler, yani gerçekten hasta olan kişilerin yanlışlıkla sağlıklı olarak sınıflandırılması sonucunda kaçırılan teşhisler, son derece önemlidir. Bu nedenle, bu uygulamaya yönelik **Recall** skoru daha önemlidir.

Bunun haricinde yanlış pozitiflerin kabul edilemeyeceği bir başka uygulamada hassasiyet (**Precision)** skoru önem arz etmektedir**.** Örneğin, spam filtresi sınıflandırması için bir model üzerinde çalıştığımızı düşünelim. Bu durumda, yanlış pozitif tahminler, yani spam olmayan e-postaların yanlışlıkla spam olarak sınıflandırılması sonucunda kaybedilen e-postalar, son derece önemli olabilir. Bu nedenle, **Precision** değeri bu senaryoda daha fazla önem arz etmektedir. Recall **s**koru da önemlidir, ancak **Precission**, spam filtresinin yanlış pozitif tahminler yapmamasını sağlamak için daha önemlidir.

Bunların dışında **F1 Skoru**, hem **Precission** hem de **Recall** skorlarının harmonik ortalaması sonucu hesaplanmaktadır. Dolayısıyla **F1 skoru**, yanlış pozitiflerin ve yanlış negatiflerin birbirine göre önemli olduğu durumlarda Kullanılmaktadır. Örnek vermek gerekirse bir arama motoru uygulamasında, hem doğru sonuçların bulunması hem de yanlış sonuçların azaltılması önemli ise **F1** skora bakılmaktadır. Bunun haricinde sadece bir skorlar modelimin hem hassasiyetini hemde duyarlılığı hakkında bilgi sahibi olmak istersem yine **F1** skora bakabilirim.

Micro avg, Macro Avg ve Weighted Avg için ise yukarıda açıklandığı üzere hedef değişkendeki (target variable) sınıfların 3 ve daha fazlası olduğu durumlarda kullanılması gerektiği önerilmektedir. Bunun için hangi skorlara bakılacağı noktasında eğer data seti balans (**balanced**) bir data seti ise **Micro avg** skoru önem arz etmekte iken, balans olmayan bir data setinde (**imbalanced**) ise **Weighted avg** skoru önemli hale gelmektedir. Bunun haricinde hedef değişkendeki bir sınıfa yönelik skorlar bizim için önemli ise **Macro avg** skorları önemli duruma gelmektedir.

Bu skorlardan hem binary hemde çok sınıflı (multiclass) sınıflandırma problemlerinde hesaplanabilen bir metriktir. Yukarıda açıklandığı üzere Log loss (logaritmik kayıp), tahmin edilen olasılık değerleri ile gerçek değerler arasındaki olasılık farkları hesaplar ve sonuç olarak bir kayıp değeri verir. Bu kayıp değeri ne kadar düşükse, modelin performansı o kadar iyidir.

Log loss, genellikle binary veya çok sınıflı sınıflandırma problemlerinde kullanılır ve bir olasılık dağılımı hesaplanır. Tahmin edilen olasılıkların doğruluğunu ölçmek için kullanılmaktadır. Log loss, model performansını ölçmek için sıkça kullanılan bir metrik olmasına rağmen, yorumu biraz zordur. Yüksek bir log loss değeri, düşük bir performansı gösterirken, düşük bir log loss değeri yüksek bir performansı gösterir. Örneğin bir kanser tanı testi üzerinde çalıştığımızı farz edelim. Modelimizin bu data setine özel bir kişinin kanserli olma olasılığını bulduğumuzu ve veri setimizde hedef değişkende (taget variable) test edilen kişilerin gerçek kanser durumları hakkında bilgimiz olduğunu varsayalım, bu yüzden modelimizin performansını log loss kullanarak ölçebiliriz.

Sonuç Olarak

Genellikle, sınıflandırma modelinin performansını değerlendirirken birden fazla skoru dikkate almak yapılacak en doğru davranıştır. Bu skorlardan **Accuracy** skoru, modelin genel performansını ölçerken, **Precision** ve **Recall** skorları, modelin sınıflandırma yeteneğinin belirli yönlerini ölçmektedir. **F1 skoru**, hem **precision** hem de **recall** skorlarının birleşik bir ölçüsü (**Harmonik Ortalaması**) olduğundan

ikisi arasındaki dengeyi göstermektedir. Micro, Macro ve Weighted Avg skorları ise hedef değişkendeki sınıfların 3 ve daha fazla olduğunda performans değerlendirmesinde kullanılması gereken skorlardır.

Faydalı Olması Dileğiyle…