Örsan Özener
Makine Öğrenmesi
08.08.2023
Yıldız Holding

PERFORMANS METRİKLERİ

Metrikler hem train hem test datasındaki sonucu ölçmek için kullanılır. Bu noktada eğer modeli iyileştirmek istiyorsak bunun için train-test arasında bir validasyon datası oluşturmaya ve validasyon datasındaki sonuca göre modelimizi iyileştirmeye dikkat etmeliyiz. Aksi takdirde test datasına göre modeli iyileştirmek data leakage'a neden olur. Bu da modelin gerçek hayatta başarısız olmasına neden olur. Burada varsayımımız train, validasyon ve test datasının benzer dağılımlardan gelmiş olması. Zaten bunlar farklı farklı datalarsa aynı dağılımda değillerse train ve validasyon datasından iyi performans alamayız. Bu farklılık ne zaman olur? Doğru bir şekilde rastgelelik yapılmadığı zaman veya zamansal bir şeye baktığımız zaman olur. Örneğin ev fiyatlarında tahminleme yapıyorum. Train datası zamanında çok güzel kredi veriliyordu, fakat validasyon ve test datası zamanında artık kredi verilmiyor yani başka bir environmentın sonucu olarak data geldi. Bunun olmaması gerekiyor.

Regresyon Problemleri İçin

Mean square error: Gerçek değer ile tahminimiz arasındaki farkın karelerinin toplamının ortalama değeri.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

Mean absolute error: Gerçek değer ile tahminimiz arasındaki farkın mutlak değerlerinin toplamının ortalama değeri.

Mean absolute percentage error: Gerçek değer ile tahminimiz arasındaki farkın gerçek değere bölünüp mutlak değerinin alınması sonrasında bunların toplanıp ortalamasının alınması. Fakat yüzdesel hata çok güvenilmeyen bir şey. Çünkü gerçek değer 0'a yaklaştığı zaman sonsuza doğru gidebilecek bir metrik.

Classification Problemleri İçin

Error Rate: Yanlış tahminlerimin sayısının ortalaması.

$$Error\ Rate = \sum_{i=1}^{n} I(y_i \neq \hat{y}_i) / n$$

Accuracy: Doğru tahminlerimin sayısının ortalaması.

$$Accuracy = \frac{Number\ of\ Correct\ predictions}{Total\ number\ of\ predictions\ made}$$

Bu metrikleri kullanmak eğer iki kümeden de eşit sayıda sample varsa mantıklıdır. Kovidlileri tahmin etmek istiyorum. 100 kişi var. 100 kişinin 99'u kovid değil zaten. Tahmin yaptığımda model hiçbiri kovid değil dediği zaman çok iyi çalışan bir modelmiş gibi görünebilir. Çünkü datada dengesizlik var.

Confusion Matrix

n=165	**Predicted: NO**	**Predicted: YES**
Actual: NO	50	10
Actual: YES	5	100

Burada 50 tanesi gerçekten hayır iken ben hayır diye tahmin etmişim, 100 tanesini de evet iken ben evet diye tahmin etmişim. 10 tanesi hayır iken evet diye tahmin etmişim, 5 tanesi evet iken ben hayır diye tahmin etmişim. Bizim istediğimiz 5 ile 10'u minimize etmek. Fakat modele bağlı olarak bu 5 ve 10'dan bir tanesi bizim için çok değerli iken diğeri çok değerli olmayabilir. Örneğin bir firma için kimler churn edecek (ayrılıp başka firmaya gidecek) diye tahmin etmeye çalıştığımızda churn edecekleri doğru tahmin etmek bizim için daha kritik olur.

Bu mantıkla confusion matrix bize genel bir fikir söyler. Confusion matrixin kötü tarafı 2-3-5 farklı modeli karşılaştırdığımız zaman tek bir karşılaştırma rakamı vermemesi. Matris veriyor. Matris de eğer domine ediyorsa biri diğerini bu diğerinden iyi

diyebilirsiniz ama domine etmiyorsa o zaman bunu karşılaştırmak için basit bir metriğe ihtiyacımız var.

Bu karşılaştırma için aşağıdaki metrikleri kullanabiliriz:

Area Under the Curve (AUC Score): Binary sınıflandırma problemlerinde kullanılır.
True Positive Rate (Sensitivity)

$$TruePositiveRate = \frac{TruePositive}{FalseNegative + TruePositive}$$

Elimdeki toplam pozitif datalardan kaçını doğru tahmin etmişim? Burada False Negative 0'a gittiği sürece (negatif demişim ve yanlış çıkmış) True Positive Rate değeri 1'e gider. Bu değeri nasıl maksimize ederiz, yani 1 yaparız? False Negative'i 0 yaparak. Bunu nasıl yaparız? Hiç negative demeyerek, hepsine pozitif diyerek.

Kovid örneğini düşünelim. İyi bir model herkese pozitif demeden populasyondaki bütün kovidleri yakalayabilen bir modeldir. Burada istediğimiz şey model o kadar iyi çalışsın ki False Negative yapmadan ben mümkün olduğu kadar True Positive Rate'i 1'e götürmeye çalışayım.

Diyelim ki kovidi anlamak için vücuttaki x değerine bakıyorum (ateş örneğin). Ben bu x değerinin tresholdunu düşürerek pozitif deme oranını artırabilirim (en düşük eşikte herkese kovid diyorum), yükselterek pozitif deme oranını düşürebilirim. Ve benim modelimin performansı bu tresholdu nereye koyduğuma göre değişir.

False Positive Rate (1-Specificity):

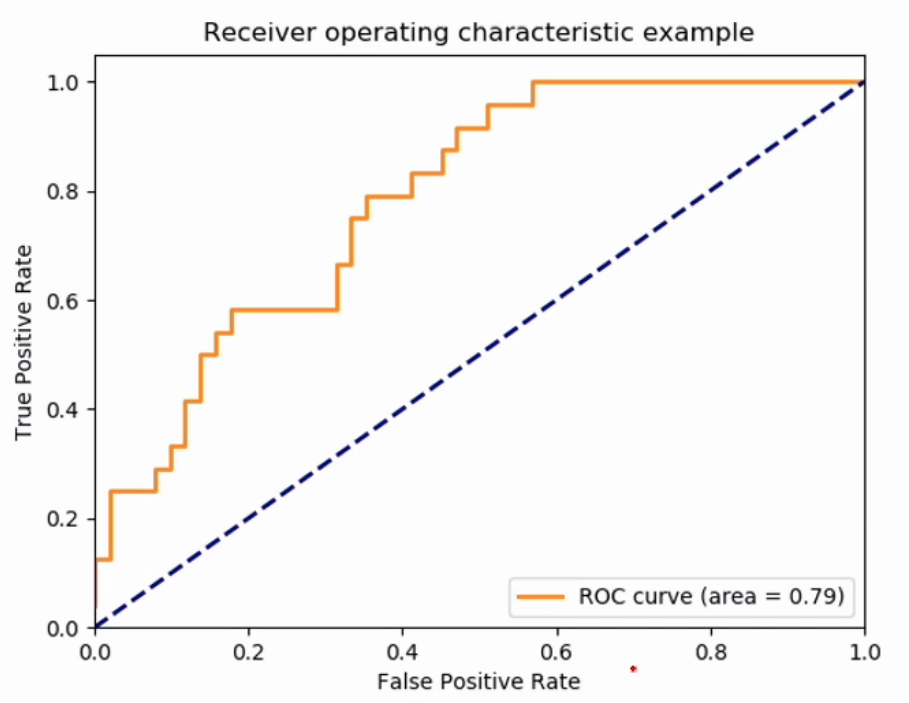
$$FalsePositiveRate = \frac{FalsePositive}{FalsePositive + TrueNegative}$$

Elimdeki negatif datadan kaçına pozitif dedim? Kovid örneğinde populasyonda sağlıklı insanlar var bunların kaçına kovid dedim? Bunu minimize etmek istiyorum. Eğer False

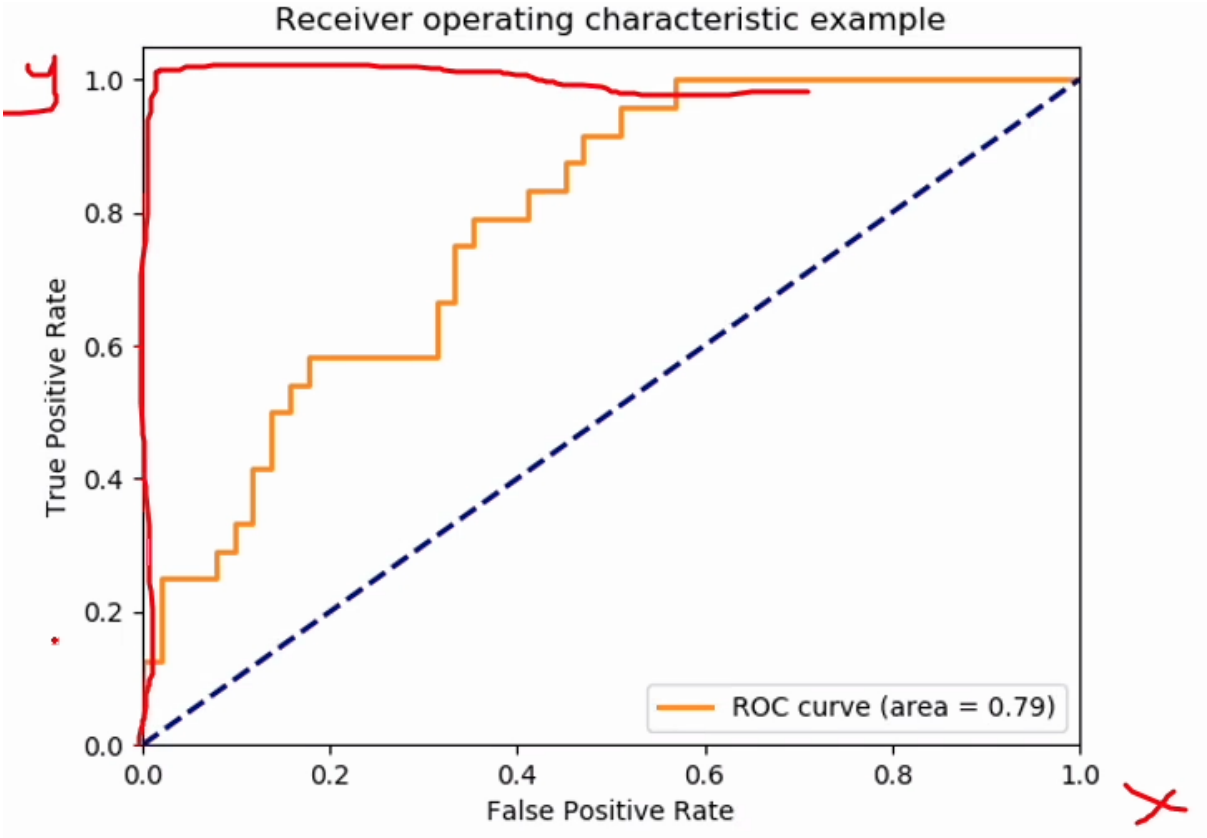
Positive'ler 0 olursa 0/(0 + TrueNegative) 'den sonuç 0 çıkar. Bu değeri nasıl minimize ederiz garantili olarak? Kimseye kovid demezsem minimize etmiş olurum.

Fakat iki amaç birbiriyle çelişiyor. True Positive Rate'te ne kadar kovid dersem 1'e gidiyor. Diğerinde de ne kadar kovid değil dersem o kadar 0'a gidiyor. x (ateş) değerinin tresholduyla oynayarak bu oranları değiştirebilirim. Ateşi 39'a çekersem True Positive Rate'i düşürürüm. Çünkü paydadaki False Negative değerini artırmış olurum. 30'a çekersem de False Positive Rate'ini artırmış olurum. İyi bir model False Positive Rate'i artırmadan True Positive Rate'i mümkün olduğu kadar yukarı çeker. Area Under the Curve bu mantıkla ortaya çıkmış bir şey.

Area Under the Curve



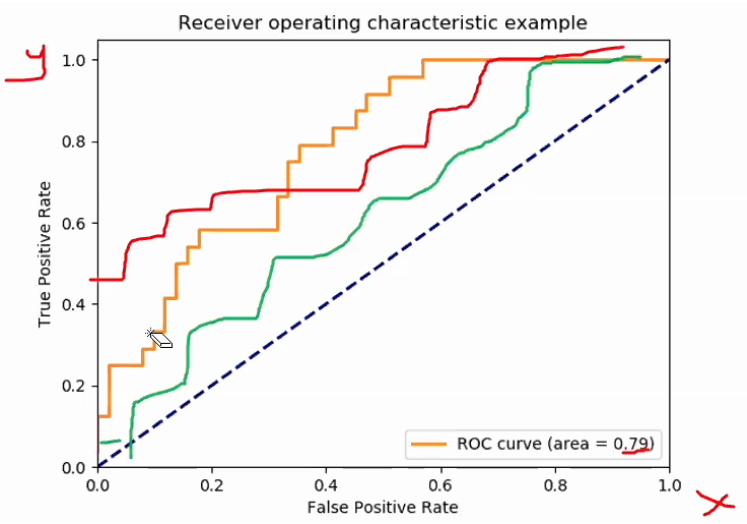
Sarıyla çizilen eğriyle ilgileniyoruz. x ekseninde False Positive Rate, y ekseninde ise True Positive Rate var. İdeal bir model False Positive'ı hiç artırmadan True Positive Rate'i yukarı çıkarır. Buradaki değerler tresholda göre değişiyor. Burada tresholddan kastımız 0-1 olma olasılığıyla ilgili verdiğimiz treshold (neyin yukarısına 1 classını aşağısına 0 classını atayacaım? Normalde 0.5 kullanıyoruz).



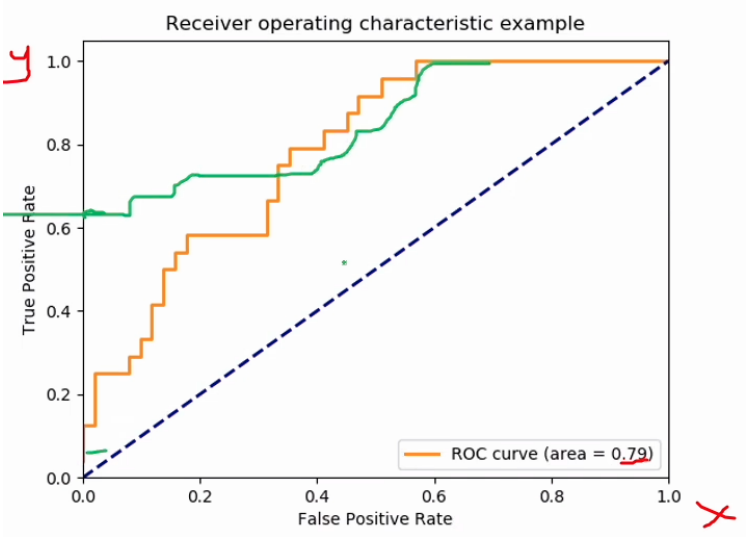
İdeal modelde True Positive Rate 0'dan bir anda 1'e doğru çıkıyor ve o şekilde devam ediyor. Kırmızıyla çizilen grafikte bunu gösterdik. Bunda çok az treshold değişiminde bile True Positive Rate'i tavana vurduruyorum ve hep 1'e gidiyor. Altında kalan alan bize skoru veriyor. Kırmızıyla çizilen grafikte yani idealde bu skorun maksimum değeri 1.

Sarı eğriye baktığımızda ise True Positive rate arttıkça istemesem de False Positive Rate de artıyor. Bazı yerlerde True Positive rate sabit kalıyor False Positive rate değişiyor. False Positive Rate 0.58'lere geldiğinde ise True Positive Rate'i 1'e vurduruyorum. Bunun alanını hesapladığımızda 0.79 çıkıyor. 0.79, 1 üzerinden çok da iyi performans olmayabilir. Bu tek başına bir anlam ifade etmeyebilir. Ancak elimizde farklı benchmarklar (farklı modellerin çıktıları) olacak ki karşılaştırma yapabilelim.

AUC skorunun doğru modeli işaret etmediği zaman olur mu?



Burada yeşil grafikle gösterdiğimiz modelin diğerlerinden daha kötü performans gösterdiğine emin olabiliyoruz. Bu alanı daha düşük olduğundan değil, her noktada daha kötü performans gösterdiğinden dolayı. Kırmızı ve sarı grafiğin altında kalan aynı da olabilir. Neye göre karar vereceğim hangi modeli kullanacağıma?



Bu örnekte yeşil grafiğin altında kalan alanın sarı grafiğin altında kalan alandan daha büyük olduğunu anlayabiliyoruz. Fakat sarı grafiğin olduğu modeli tercih ettiğimiz durumlar olur mu?

AUC skoru yüksek olan model demek; elimizde iki farklı sınıf varsa elimizde bu iki farklı sınıfı birbirinden kolay bir şekilde ayrıştırabilen model anlamına geliyor. Fakat benim spesifik kullandığım treshold seviyesinde (normalde 0.5 kullanıyoruz) AUC skoru yüksek olan modelin performansının teknik olarak daha kötü olma ihtimali var. Bu kadar detaya girip bakılmayabilir ama bunu bilmekte fayda var.

Precision - Recall:

Precision: Pozitif tahmin ettiklerimizin kaçını doğru tutturduk?

Recall: Gerçekte pozitif olanların yüzde kaçını doğru tutturduk?

F1 Skor:

Precision ve Recall bilgisine beraber de bakabiliriz. Bunun yöntemlerinden biri F1 skoruna bakmak. F1 skor; precision ve recall'ın harmonik ortalamasıdır.

$$F1 = 2 * \frac{1}{\frac{1}{precision} + \frac{1}{recall}}$$

Bunların dışında;

- AUC skor yerine precision ve recall skor da kullanılabilir. Bunda True Positive Rate-False Positive Rate yerine Precision ve Recall yazılıyor. Bunu özellikle azınlıkta olan bir sınıf varsa ve azınlıkta olan sınıf mevcut sınıftan daha önemliyse kullanıyoruz. Yani 100 tane hasta var bunların belli bir kısmı covid veya 100 müşteri var bunların sadece %3'lük kısmı rakip firmaya gidecek şeklinde verimiz var. Bu durumda AUC skor yerine F1 skor veya Precision ve Recall grafiğinin altındaki alana bakmak daha anlamlı sonuçlar verebilir.
- Multiclass classificationda logarithmic loss metriği kullanılabilir.
- Clustering problemleri için Dunn index, Silhouette Coefficient gibi metrikler kullanılıyor.