Merhabalar, bu yazıda makine öğrenmesi modellerinin doğruluğunu ölçmek için kullanılan metriklerden bahsedeceğim. Bu yazıyı bir cheatsheet gibi düşünebilirsiniz, umarım yardımcı olur.

En temelde bir makine öğrenmesi modeli çıkarırken hepimizin yaptığı şey hipotezde bulunmak. Modele en çok etki ettiğini düşündüğümüz feature'ları ekliyoruz, burada bir hipotez var. Çok basit bir örnekle açıklayacak olursam, elimizde öğrencilerin derslerden aldığı nihai harf notları, derse harcadıkları zaman, ödev verme oranları gibi bilgiler var, bunlardan ders notlarına gerçekten etki ettiğini düşündüklerimizi alıp lineer regresyon modeline, geçme kalma tahmini yaparken de lojistik regresyon modeline koyuyoruz. Burada bu feature'ların etki ettiğini savunmamız bir hipotezdir; ve hipotezlerin doğruluğu sensitivity, specificity, F1 score, p-value gibi değerlere bakılarak ölçülür; bu yüzden akademi tarafında çalışanlar ya da istatistik üzerine çalışanlar bu kavramlarla haşır neşirdir.

En klasik örnek olarak bir hastalık için test yaptığınızı varsayalım.

True positive: Hastalığınız olduğunu düşünüyorsunuz (testin pozitif çıkacağını tahmin ettiniz) ve test pozitif çıktı. Yani öne sürdüğünüz hipotezin doğru olduğunu düşündünüz ve doğru çıktı.

False positive: Hastalığınız olduğunu düşünüyorsunuz (tahmininiz pozitif) ama yaptığınız test negatif çıktı.

False negative: Hastalığınız olmadığını düşünüyorsunuz (tahmininiz negatif) ama test yaptınız ve pozitif çıktı.

True negative: Hastalığınız olmadığını düşünüyorsunuz (tahmininiz negatif) test yaptınız ve negatif çıktı.

Başka bir örnek verelim. Wimbledon'ı kimin kazanacağı üzerinden bir tahminde bulunuyorsunuz, Djokovic üzerinden bir sıfır-hipotezinde bulundunuz.

True positive: Djokovic'in kazanacağını düşündünüz ve kazandı.

True negative: Djokovic kaybeder dediniz ve kaybetti.

False positive: Djokovic kazanır dediniz ve kaybetti.

False negative: Djokovic'in kaybedeceğini düşündünüz ama kazandı.

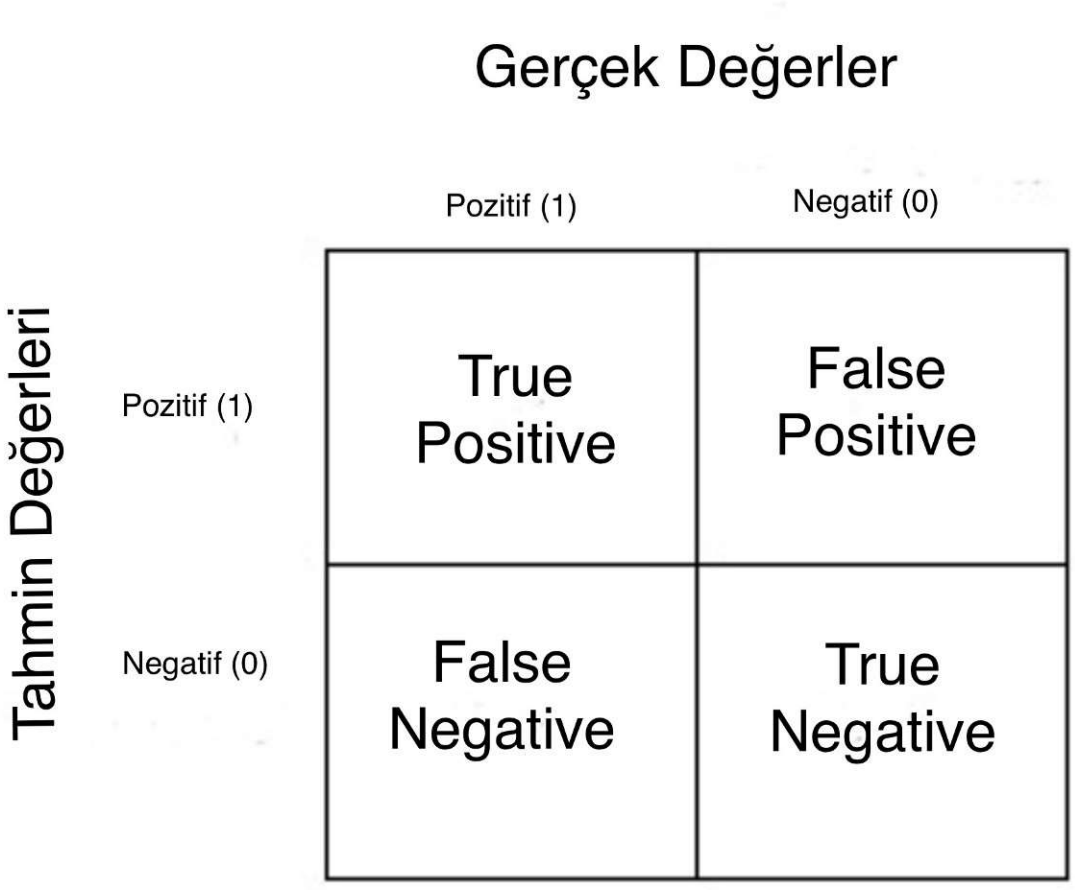
Eğer kendinizi çok kaptırıp bahis oynarsanız istatistikte dillere pelesenk olmuş tip I ve tip II hata yapabilirsiniz. Burada sıfır hipotezi (sizin gerçekleşeceğine inandığınız durum, varsayımınız, null hypothesis) Djokovic'in kazanacağını düşünmeniz.

Tip I hata: Djokovic'in kazanacağı üzerine bahis oynadınız ve kaybetti. (False Positive)

Tip II hata: Djokovic'in kaybedeceğini düşündünüz, bunun üzerine bahis oynadınız ve kazandı. (False Negative)

Karmaşıklık Matrisi (Confusion Matrix)

Karmaşıklık matrisi yukarıda bahsettiğim veriyi düzenleyip üzerinden hesaplamalar yapmamızı sağlar. Karmaşıklık matrisini matrisin kendisi üzerinden anlatmak daha iyi olacaktır.



Karmaşıklık matrisinde satırlarda tahmin edilen pozitif ve negatiflerin sayıları, sütunlarda da gerçekte olan pozitif ve negatif çıktı sayıları yer alır. Karmaşıklık matrisi

sadece pozitif negatif gibi dikotom veriler dışında da kullanılır, ama bu şu an konumuzun dışında, eğer merak ediyorsanız yazının sonunda bir tane görebilirsiniz.

Yine hastalık üzerinden örnek verelim, yüz binde bir ihtimal yakalanacağınız bir hastalık üzerine test yapmak istiyorsunuz ve test size %99,999 doğruluk (accuracy) vaat ediyor, bu testi yapar mıydınız? Sizce accuracy burada yeterli bir metrik midir?

Buradaki asıl sorun bu sınıflandırma (hasta/hasta değil) problemindeki verilerin çoğunun hasta değil sınıfına ait olması, yani bir sınıf diğerini baskılıyor. Böyle durumlarda recall, precision gibi metriklere başvuruyoruz.

Recall

Bahsettiğim durumda true positive'ler hasta olarak tahmin edilen ve gerçekten hasta olan insanlar, false negative'ler hasta olmadığı tahmin edilen ama hasta olan insanlar. Recall'a bakma sebebimiz tamamiyle paydadaki false negative'ler, yani hasta olmadığı tahmin edilen ve hasta olan insanlar. Doktorsanız hasta olan birine hasta değil demenin maliyeti ağır olduğu için recall false negative'in gözardı edilemez olduğu durumlarda önemli bir metrik.

$$\text{Recall} = \frac{\text{True Positive}}{\text{True Positive + False Negative}} = \frac{\text{Hasta tahmin edilip hasta çıkanlar}}{\text{Hasta tahmin edilip hasta çıkanlar + Hasta değil tahmin edilip hasta çıkanlar}}$$

Bu mantıkla herkese hasta derseniz recall 1 çıkar, yani recall için oturup ideal bir değer aralığı belirlememiz çok da doğru değil. Zaten her problemde istenilen false negative değeri için ayrı recall ve precision değerleri belirliyoruz. ROC curve'de kullandığımız True Positive Rate ve Sensitivity teknik olarak recall'la aynıdır.

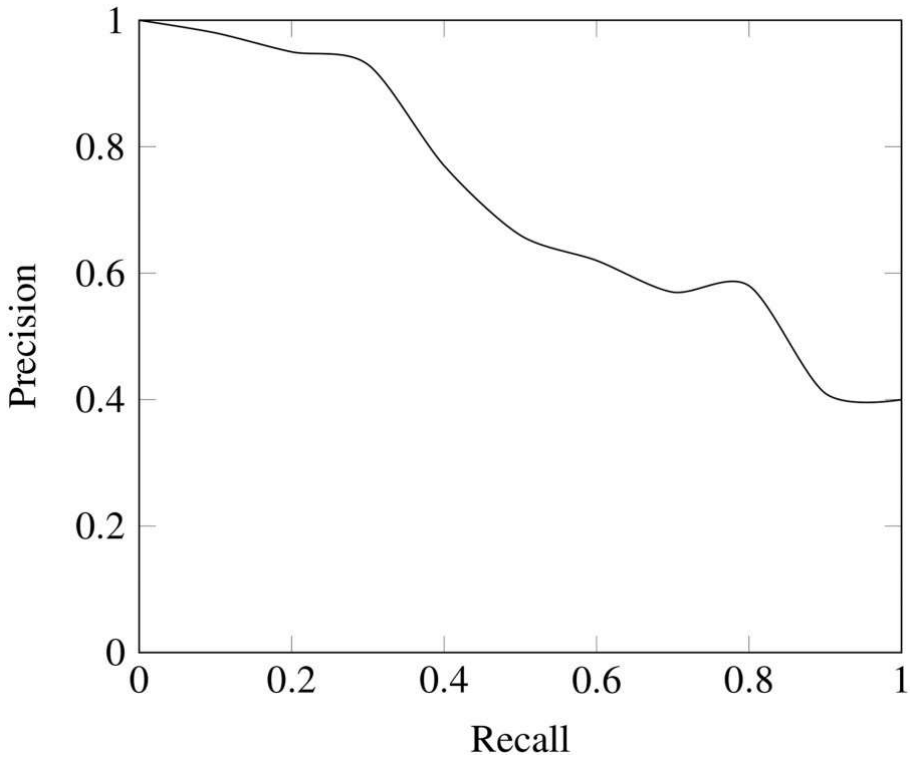
Precision

Precision'ın formül hali:

$$\text{Precision} = \frac{\text{True Positive}}{\text{True Positive + False Positive}} = \frac{\text{Hasta tahmin edilen ve gerçekten hasta olanlar}}{\text{Hasta tahmin edilip hasta olanlar + Hasta tahmin edilip hasta olmayanlar}}$$

Precision hasta tahmin ettiklerimizin kaçının gerçekten hasta olduğu ile ilgilenir, paydada bütün bir pozitif satırını alma sebebimiz de bu. Bu noktada false negative'lerimiz çok fazla olacaktır.

Precision'la recall ters orantılıdır ve ikisinin arasında bir denge tutturmak gerekir. (precision recall trade-off)



Kaynak: American Society of Mechanical Engineers asmedigitalcollection.asme.org

F1-Score

Precision ve recall'u dengelemede devreye F1-Score giriyor. F1-Score precision ve recall'un harmonik ortalaması.

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

Kaynak: towards data science

Burada normal ortalama alınmamasının sebebi de recall'un 1 precision'ın 0 olduğu bir durumda (ya da tam tersi) ortalama 0.5 çıkacaktır, bu da bize iyi bir iç görü vermez, başladığımız yere dönmüş oluruz, bunun yerine harmonik ortalamada recall'un 0.001 precision'ın 0.999 olduğu durumda F1-score 0.001 çıkar (0 olsaydı direkt 0 olacaktı). Eğer precision ve recall'un arasında dengeyi arıyorsak F1-Score'un maksimum olduğu durumlara bakarız, eğer az önce verdiğim doktor örneğine bakıyorsak recall'u maksimize edip precision'ı minimize ederiz. Özetle, hangi metriği maksimize edeceğimize karar vermek çok önemli.

Accuracy: Herkesin tahmin edebileceği üzere accuracy doğru tahminlerin toplam tahmin sayısına oranıdır. Accuracy'nin tersi de misclassification rate'tir, bu da (false positive+false negative)/toplam tahmin sayısı olarak ifade edilebilir, hata oranını verir.

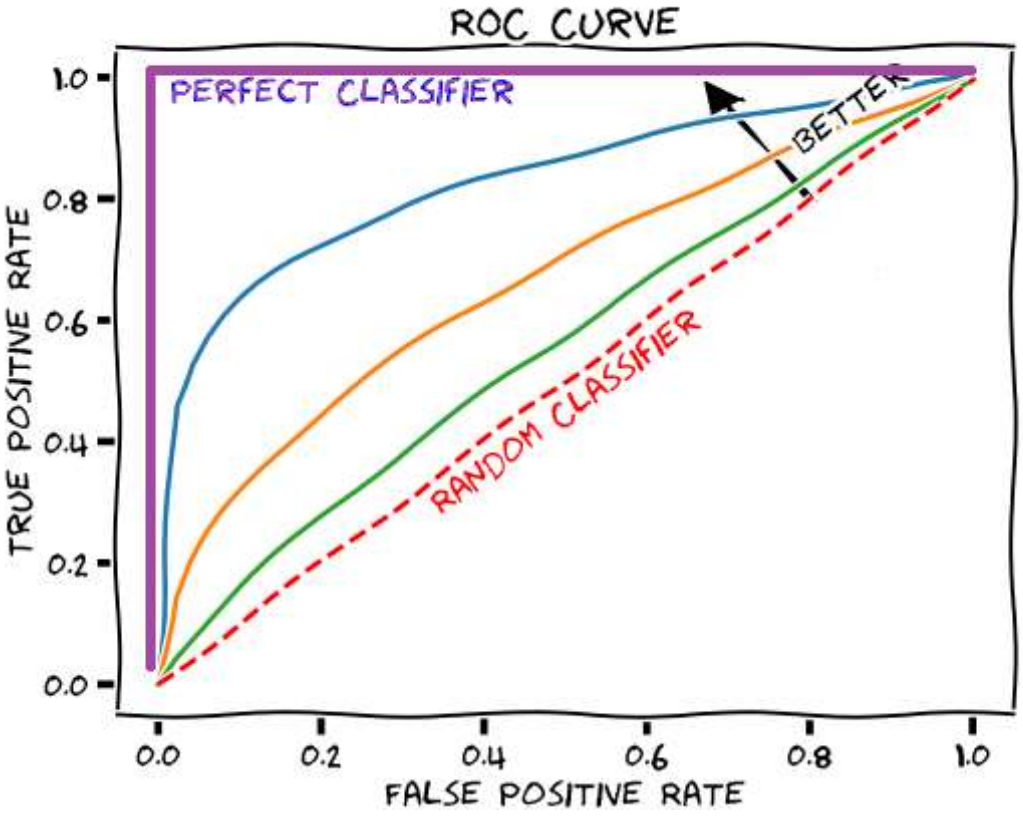
True Positive Rate: Gerçekte çıktımız (outcome) pozitif olduğu tahminlerin kaçını pozitif olarak tahmin ettik? (True Positive/Gerçekte pozitif olan durumlar)

False Positive Rate: Gerçekte çıktımızın negatif olduğu durumların kaçını pozitif olarak tahmin etmiştik? (False Positive/Gerçekte negatif olan durumlar)

True Negative Rate: Gerçekte çıktımızın negatif olduğu durumların kaçını negatif olarak tahmin etmiştik? (True Negative/Gerçekte negatif olan durumlar)

Receive Operating Characteristics (ROC) Curve & Area Under Curve

ROC ve AUC sınıflandırma problemlerinde en çok kullanılan performans ölçütlerinden ikisidir. ROC'un ne anlam ifade ettiğini ROC üzerinden anlatmak daha iyi olacak.



Kaynak: glassboxmedicine.com

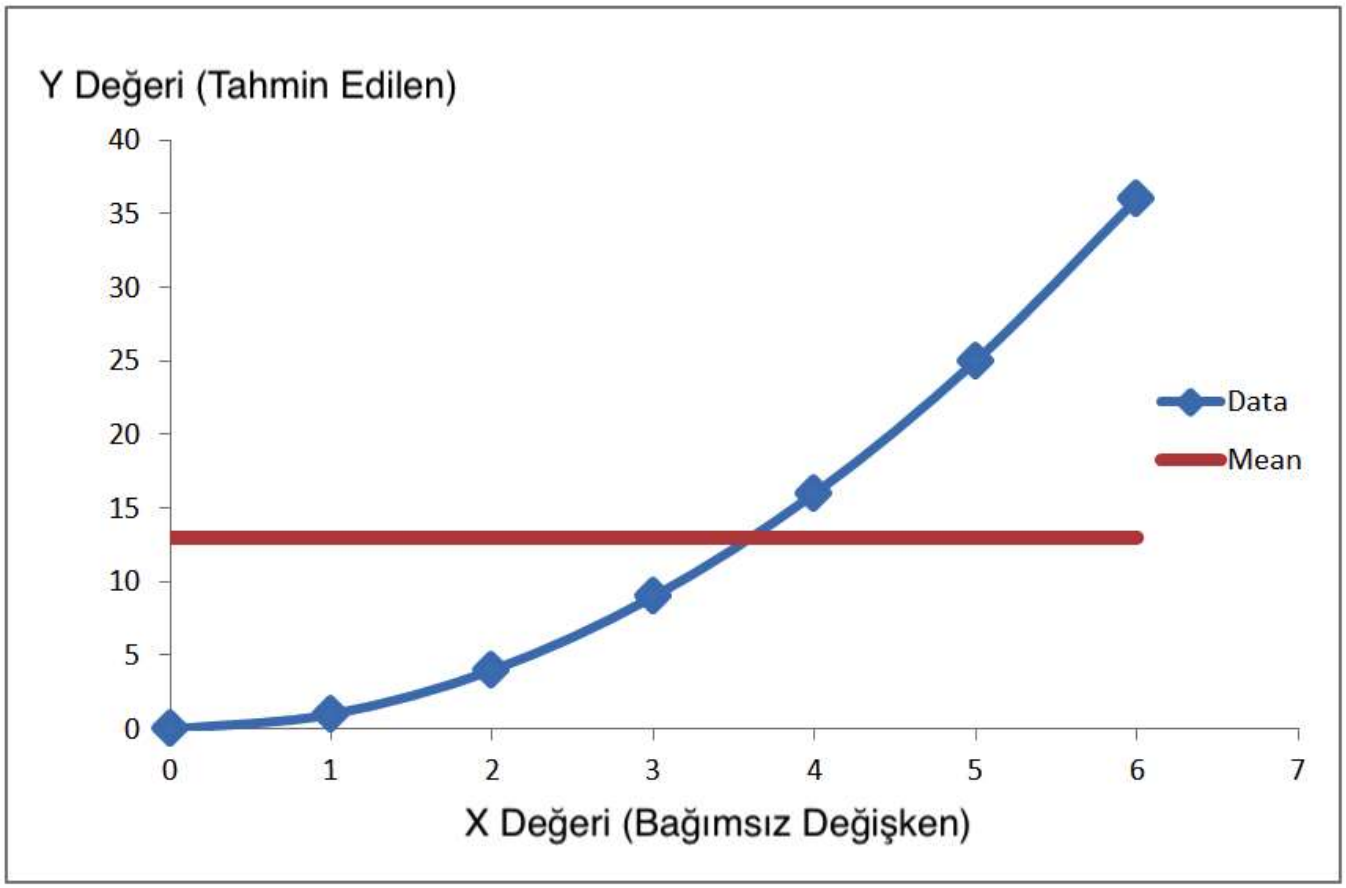
ROC bize modelin true positive rate'iyle false positive rate'i cinsinden ne kadar iyi ayrım yapabildiğini açıklar. AUC ise ROC eğrisinin altında kalan alanı verir, 0'la 1 arasındadır,

0'sa bütün tahminler yanlıştır. True positive rate kısaca gerçekte durum pozitifse bunların kaçını pozitif tahmin ettiğimizi gösterir, false positive rate de gerçekte durum negatifken bunların kaçını pozitif olarak tahmin ettiğimizi (yanlış alarm da denir) gösterir. Yani aslında ikisinde de pozitif tahmin ettik fakat çıktıları farklı. ROC grafiğinde 0.5,0.5 noktasında yani yukarıda random classifier yazan köşegende sınıflandırma becerisi bulunmayan modeller yer alır, yazı-tura örneğini düşünün, null hypothesis'iniz yazı gelmesi (yani pozitifiniz yazı) %50 ihtimal yazı gelecek ve %50 ihtimal gelmeyecek, hiçbir şey farketmiyor. Sol üste çıktıkça doğru tahmin sayımız artar. Eğer veri setiniz dengesizse accuracy'den daha kullanışlı bir metriktir. Eğer negatif örnekler pozitif örnekleri baskılıyorsa AUROC fazla optimist olabilir, çok basit bir mantıkla false positive rate'in paydasındaki true negative sayısı fazla olacağı için FPR düşük, TPR yüksek çıkacaktır. O yüzden önce confusion matrix'e bakmak önemli; eğer aksi bir durum varsa precision ve recall çok daha kullanışlı metriklerdir.

Regresyon Metrikleri

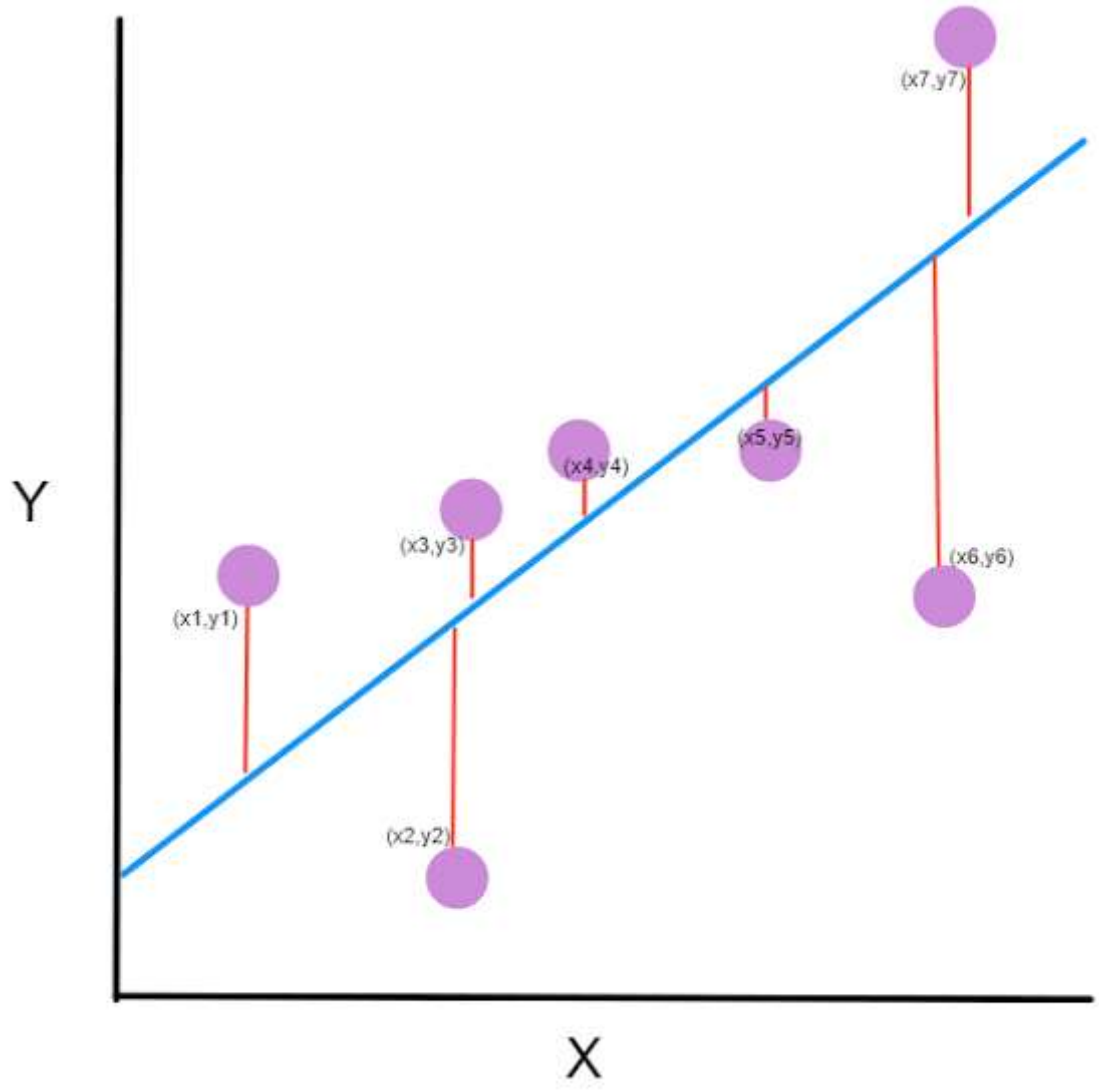
R-Squared (R-Kare)

R2 bize çizdiğimiz regresyon çizgisinin gerçek tabloya göre toplam varyasyonunu gösterir, istatistikte goodness of fit dediğimiz, veriyi bir çizgiye oturtuyoruz evet ama bu ne kadar iyi performans gösteriyor sorusunu cevaplar.



R2 yukarıda mean (veri setinin averajı) çizgisinden ne kadar daha iyi tahmin yapabildiğimizi ölçer. Elinizde bir veri seti var ve hiç bilgisayar yoksa ilk tahmininiz ortalamasını alıp ortalamayla tahmin etmek olur, bu noktada mean çizgisiyle datalarımızın arasındaki farkı alarak hata buluruz. En iyi R2 değeri 1'dir, 0.12 altındaki R2 değerleri modelinizin iyi olmadığını gösterir (Cohen, 1988). Normalde R2 değerlerinin 0'la 1 arasında olduğunu varsaysak da negatif çıkabilir. R2'nin negatif çıkması, modelinizin normalde mean kullandığınız durumdan çok çok daha kötü bir fit elde ettiğini gösterir, yani underfit eder. Bunun dışında, birden fazla değişkeni hesaba kattığımız regresyonda (multivariate regression) R2 yerine adjusted-R2 değerine bakarız, çünkü R2 çoğunlukla yanıltıcıdır.

Mean Squared Error



Kaynak: freecodecamp.com

Mean square error, -türkçeye hataların karelerinin ortalaması olarak çevrilebilir- çok basit şekilde sizin regresyon doğrunuzla data noktalarının aralarındaki uzaklığın karesini

alır ve kaç tane nokta varsa o sayıya böler. MSE'nin formülü aşağıda, az önce bahsettiğimden bir farkı yok.

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

MSE'yi minimum yapacak eğim ve intercept noktalarını bulmak için kısmi türevlerini 0'a eşitleriz.

$$\frac{\partial MSE}{\partial m} = \frac{\partial MSE}{\partial b} = 0$$

Buraya ispatını yapmayacağım, ama aşağıda MSE formülünün açılıp eğime (m) ve intercept'e (b) göre türevleri alınmış halleri var.

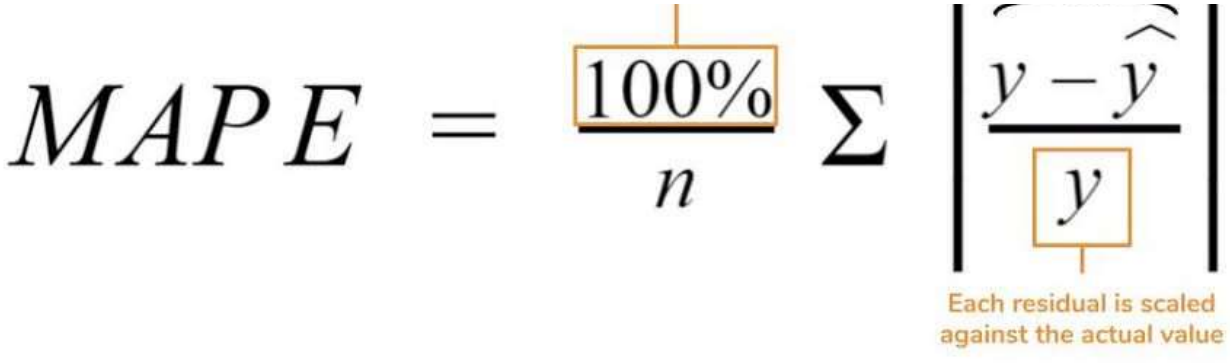
$$m = \frac{\overline{y} - \frac{\overline{xy}}{\overline{x}}}{\overline{x} - \frac{\overline{x^2}}{\overline{x}}} \cdot \frac{(\overline{x})}{(\overline{x})} = \frac{\overline{x}\,\overline{y} - \overline{xy}}{(\overline{x})^2 - \overline{x^2}}$$

MSE'yi en düşük yapacak eğim değeri

$$m\overline{x} + b = \overline{y} \Rightarrow b = \overline{y} - m\overline{x}$$

MSE'yi en düşük yapacak intercept (b) değeri

Mean Absolute Percentage Error



Kaynak: dataquest.io

MAPE'de veriden tahminlerimizi çıkarıp, gerçek veriye böleriz, ardından bunların mutlak değerlerini toplayıp 100ole çarpıp veri sayısına böleriz. MAPE, outlier veri noktalarına karşı iyi performans gösterse de verinizde 0 varsa paydada da yer alacağından sıkıntı çıkaracaktır. Gerçek değerin çok küçük olduğu durumlarda da hata büyük çıkacaktır. Bunun dışında tahmin ettiğimiz değerler gerçek değerlerin altındaysa MAPE bunun tam tersi durumdaki değerinden çok daha azdır, yani ortada bias var.

Mean Percentage Error (MPE)

$$MPE = \frac{100\%}{n} \sum \left(\frac{y - \hat{y}}{y} \right)$$

MPE'nin MAPE'den tek farkı mutlak değer operasyonunun olmaması, bunun kötü tarafı pozitif ve negatif yüzdelik hatalar birbirini götürmesi ve modelin ne kadar hata olduğuna dair fikir vermemesi, iyi tarafı da modelin underestimation mı(tahminin gerçek değerden küçük çıkması, negatif hata) yoksa overestimation mı (tahminin gerçek değerden büyük olması, abartı, pozitif hata) yaptığına dair fikir verebilmesidir.