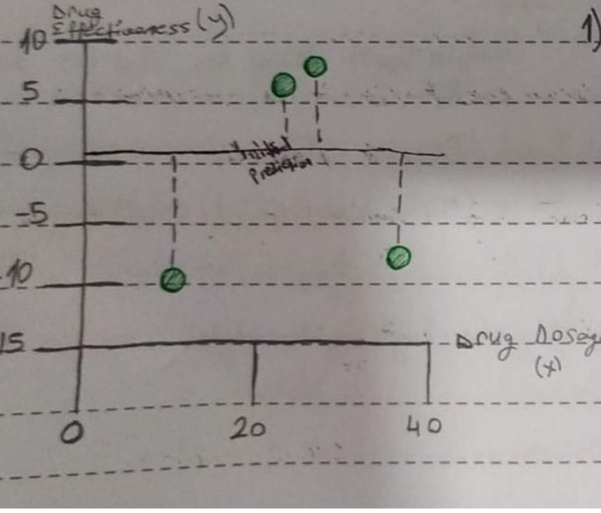
XGBoost

a) Regression :

Örneği kolay anlatmak için super simple bir training set kullanacağız.



1) The first step in fitting XGBoost to the Training Data is to make initial prediction. Bu initial herhangi bir sayı olabilir. Ancak Default'da hem Regresyon hem Sınıflandırma için 0.5 alınıyor.

2) Calculate first (initial) residuals. Difference betwen Observed and Predicted values.

3) Tüm ağaçlar single leaf ile başlar. Bu single leaf'te ilk olarak Residualler olur.

4) Bu residualler için "Similarity Score" hesaplanır.

$$\text{Similarity Score} = \frac{(\text{Sum of the Residuals})^2}{\text{Number of residuals} + \lambda}$$

λ → (Regularization parameter)

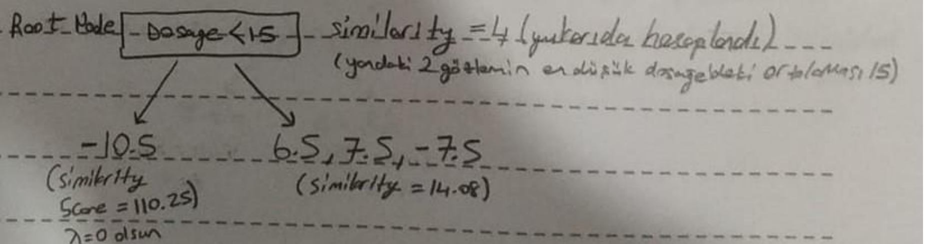
$\boxed{-10.5,\ 6.5,\ 7.5,\ -7.5}$ → ilk leaf ilk residualler ile başlar demiştik.

$$\text{Sim. score} = \frac{(-10.5+6.5+7.5-7.5)^2}{4+\lambda} = 4$$

λ → 0 olsun şimdilik.

5) İlk leaf similarity = 4 oldu. Şimdi şunu düşünmeliyiz, Residualleri 2 gruba böler isek daha iyi cluster etmiş olur muyuz.

6) Bunun için ilk leaf node'u en iyi bölecek Dosage (x) karar vermek lazım. En iyi demek en fazla "Gain" sağlayan demek. Bunun için Bağımsız değişkenin en küçük 2 değerinin ortalamasını alarak denemeye başlarız ve her denemede bir sonraki 2 gözlemin ortalamasını alarak leaf denenir.

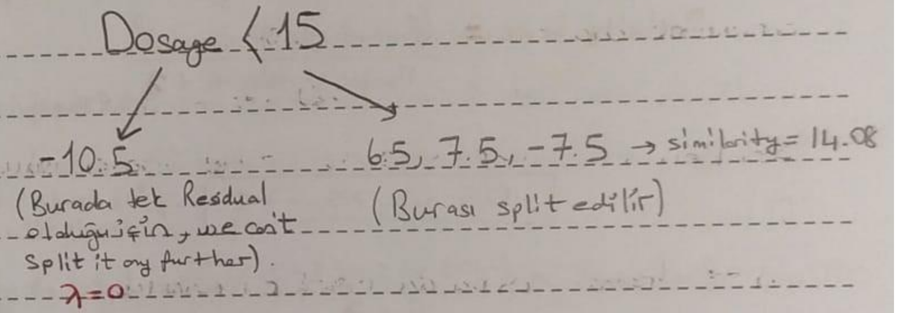

Root Node
Dosage < 15
similarity = 4 (yukarıda hesaplandı)
(yandaki 2 gözlemin en düşük dosagelerinin ortalaması 15)
-10.5
(similarity Score = 110.25)
λ=0 olsun
6.5, 7.5, -7.5
(similarity = 14.08)

Gain = Left similarity + Right sim - Root sim ⇒ 110.25 + 14.08 - 4 = 120.33

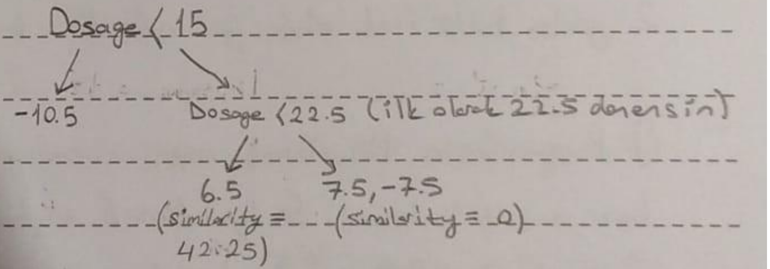
../..../....

Now, we calculated the Gain for the threshold "Dosage < 15", we can compare it to the "Gain" calculated for other Thresholds.

→ Dosage <15, Dosage <22.5, Dosage <30 Gain'lerine bakılır. En yüksek Gain veren Root olarak seçilir. Bu örnekte en yüksek Gain Dosage <15 iken oluyor.

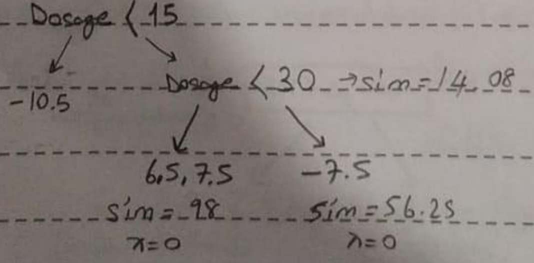

Dosage < 15
-10.5
6.5, 7.5, -7.5 → similarity = 14.08
(Burada tek Residual olduğu için, we can't split it any further).
λ=0
(Burası split edilir)

22.5 için: Bu durumda; split edilecek node için yine yukarıdaki gibi threshold'lar denenmeye başlanır.



Gain = 42.25 + 0 - 14.08 = 28.17

30 için:



Gain = 98 + 56.25 - 14.08 = 140.17 } Dosage 30 seçildi.

7) Tree'yi Prune etme i γ(gamma) = 130 olsun diyelim

★ Gain − γ(gamma) < 0 ise, we remove the Branch.

Mesela → Dosage < 30 olduğu durumda Gain = 140.17 idi

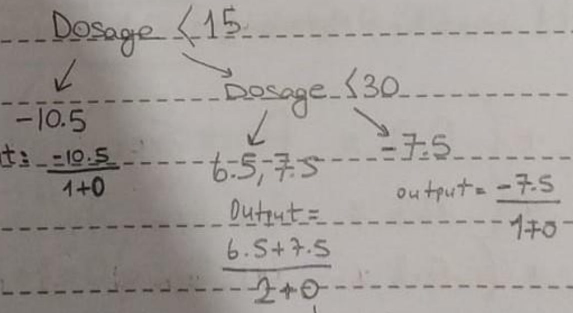
O node'u remove edemeyiz çünkü 140.17 > 130.

8) λ = 0 aldık hep işlem kolaylığı için. λ → Regularization parameter and reduce the prediction's sensitivity to individual observations. ①" When λ > 0, the similarity scores are smaller." ②" When λ > 0, the Gain values are smaller also."

λ → Training Data'da overfit'i engeller.

9) Leaf Output Value = $\frac{\text{Sum of Residuals}}{\text{Number of Residuals} + \lambda}$

Yani ;


Dosage < 15
-10.5
output: $\frac{-10.5}{1+0}$
Dosage < 30
6.5, 7.5
Output = $\frac{6.5+7.5}{2+0}$
-7.5
output = $\frac{-7.5}{1+0}$

λ > 0 olduğunda, it reduce the amout that this individual observation adds to the overall predistion.

../..../....

10) Bu durumda, örneğin;

New Predicted value, for observation with Dosage = 10 is calculated as follows:

Learning Rate

Original Prediction + (ε(eta) × -10.5)

Dosage < 15

$0.5 + 0.3 \times -10.5 = -2.65$

-10.5

output = -10.5

$\lambda = 0$ iken

(new prediction)

★ New residual is smaller than the previous (initial) residual.

Aynı sekilde; New prediction for observation with Dosage = 20

$0.5 + (0.3 \times 7) = 2.6$

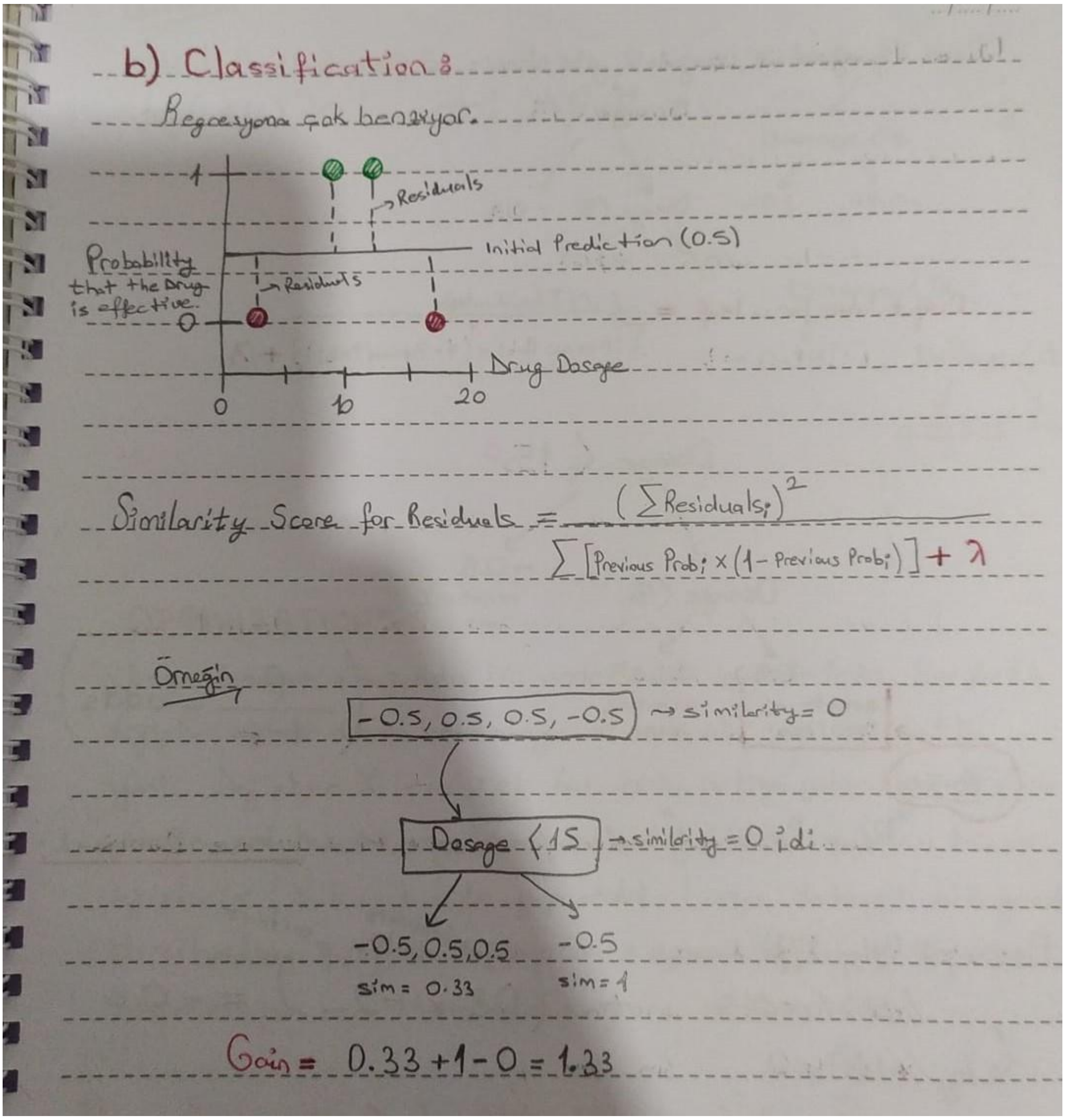
→ New residual is smaller than before again.

11) Build another tree based on the newest "Residuals."

$0.5 + (0.3 \times \text{First Tree})$

$+ (0.3 \times \text{2nd Tree})$

$+ (0.3 \times \text{3rd Tree})$



b) Classification:

Regresyona çok benziyor.

1
Residuals
Initial Prediction (0.5)
Probability that the Drug is effective
Residuals
0
Drug Dosage
0
10
20

$$\text{Similarity Score for Residuals} = \frac{\left(\sum \text{Residuals}_i\right)^2}{\sum\left[\text{Previous Prob}_i \times \left(1 - \text{Previous Prob}_i\right)\right] + \lambda}$$

Örneğin

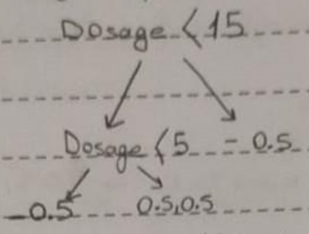
$\boxed{-0.5,\ 0.5,\ 0.5,\ -0.5}$ → similarity = 0

$\boxed{\text{Dosage} < 15}$ → similarity = 0 idi.

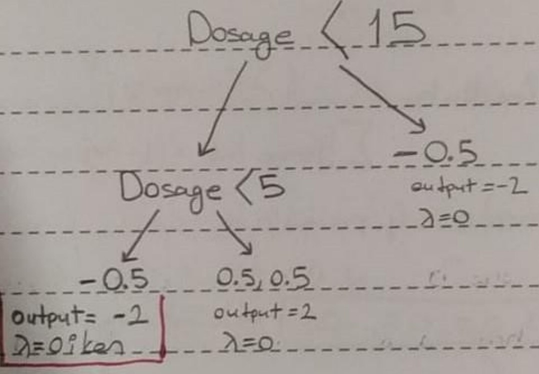
-0.5, 0.5, 0.5 | -0.5
sim = 0.33 | sim = 1

Gain = 0.33 + 1 − 0 = 1.33

Oluşan ilk ağacımız aşağıdaki gibi olsun:



Output value for a leaf $= \dfrac{(\sum \text{Residuals}_i)}{\sum[\text{Previous Prob}_i \times (1 - \text{Previous Prob}_i)] + \lambda}$



$\left(\dfrac{p}{1-p} = \text{odds}\right)$

örneğin

"New Predicted" value for observation whose Dosage = 2;

Predicted Drug Effectiveness

Initial = 0.5 + $(\overset{\text{Learning rate}}{0.3} \times \overset{\text{output}}{-2}) = -0.6$

↳ log (odds) = 0

↳ Log(odds) = -0.6 ; Convert log(odds) into probability.

Use Logistic Func.

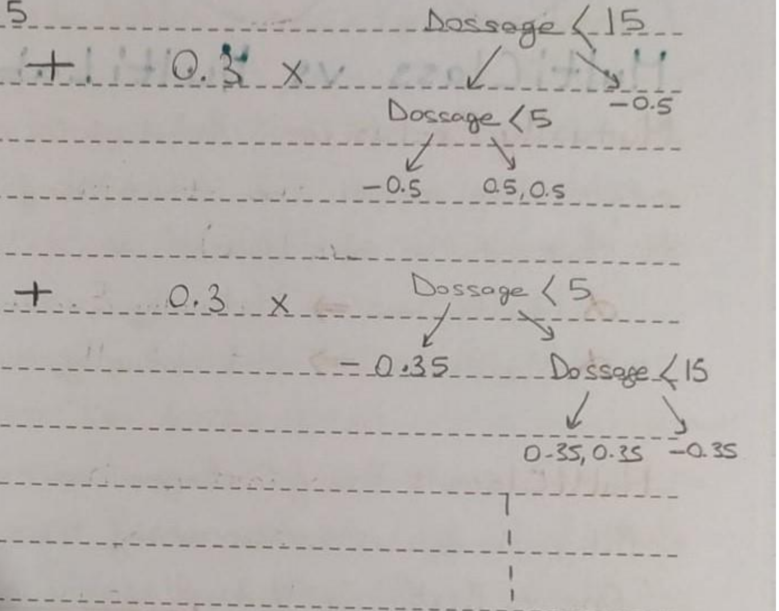
$$\text{Probability} = \frac{e^{-0.6}}{1+e^{-0.6}} = 0.35$$

→ Original residual was -0.5, with new prediction it becomes -0.35.

Gösterim:

Initial Prediction = 0.5

log(odds) = 0



OPTIMIZATIONS:

Large dataset olduğunda bir node karar vermek için, her farklı değişkenin her threshold değerine bakmak çok maliyetli. Yani, Değişken X için tek tek arka arkaya gelen her 2 gözlemin ortalamasına bakıp en iyi threshold buluyor idik ve bunu tüm değişkenler (Y, Z ...) için yapıyorduk. Large dataset olduğunda thresholdları quantile noktalardan seçer XGBoost, böylece bakması gereken threshold sayısını azaltmış olur.

How Xgboost Deal with Missing Values?

Training kümesinde missing değerlerin sağ veya sol node'dan hangisine gideceğine loss'u minimize etmesi bakımından karar verir ve test kümesindeki missing değerler o node'a gider default olarak. Training kümede NaN yok ise bu durumda her node için Default missing yönü seçer ve Test için bu yön kullanılır.