

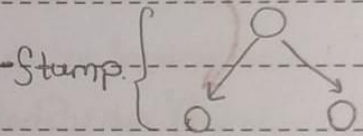
GBM ve AdaBoost

AdaBoost :

Three ideas behind "AdaBoost" are :

1) AdaBoost combines a lot of "weak learners" to make classifications. The weak learners are almost always stumps.

Only use
a variable to make
decision.



2) Some stumps have more chance to affect final decision.

3) Each stump is made by taking the previous stump's mistakes into account.

Özet olarak :

1) Başlangıçta datasetteki her sample'a aynı ağırlığı verir.

2) Kullanılan feature'lara bağlı her feature için "Stump" lar oluşturur. Bu weak learner'lerden en iyi tahmin yapanı seçer.

3) Final kararı ne kadar etkileyeceği tamamen o tree'nin (Stump) başarısına bağlıdır.

4) En iyi tahmin yapan içinde hangi gözlemleri yanlış tahminlemişe bakılır.

5) Bunlar hangi gözlemler ise bir sonraki adımda bu sample'ların ağırlığı artırılır. Ağırlığın artırılması demek, 2. oluşturulacak stump (weak tree) için oluşturulan datasette 1'den çok kez o hatalı gözlemlerin yer alması demek.

6) 2. weak learner ağacı oluşturulurken aynı size'da dataset farklı çokluğun gözlemler olur.

7) Her sample'a yine aynı ağırlık vererek (zaten datada o gözlemler çok olur) 2. stump'lar ayrı ayrı her değişken ile oluşturulur. En iyisi seçilir.

8) Bu adımlar tekrar eder. (Stump sayısı kadar).

9) Her ağacın başarı oranına göre final decision etkisi faktörü ve final karar verilir. Her adımda oluşan en iyi ağaç kendi başarı oranı ile finale etki ederek süreç biter.

GBM :

a) Regression :

Height	Color	Gender	Weight (Target)
1.6	Blue	M	88
1.5	Blue	F	56
1.5	Green	M	77

1) İlk olarak $\frac{88+56+77}{3} = 73$ average weight hesaplanır.

This is the first attempt at predicting everyone's weight.

2) Build a tree based on the errors from First Tree

(Yeni tüm herkes için ilk adımda "Average" prediction yapılır. İlk adımıdaki error'ları hesaplayacağız şimdi.)

Height	color	Gender	Weight(Target)	Residual
1.6	B	M	88	$88 - 73 = 15$
1.5	B	F	56	$56 - 73 = -17$
1.5	G	M	77	$77 - 73 = 4$

* (Bu adımda 1. tree build edeceğiz (Stump değil), bu tree'yi build eder iken Weight(Target) predict etmek yerine Residual'leri predict edeceğiz.)

Build edilen tree bu olsun:

(Ağaç daha büyük olabilir, göstermek için)

Gender = F

Height < 1.6

Color not Blue

-17

4

15

(2 gözlem buraya düşse idi aralarında yansattık)

3) We now combine the original leaf with new one

Average weight (initial prediction) = 73

1. gözlem \rightarrow Predicted weight = $73 + (0.1 \times 15) = 74.5$

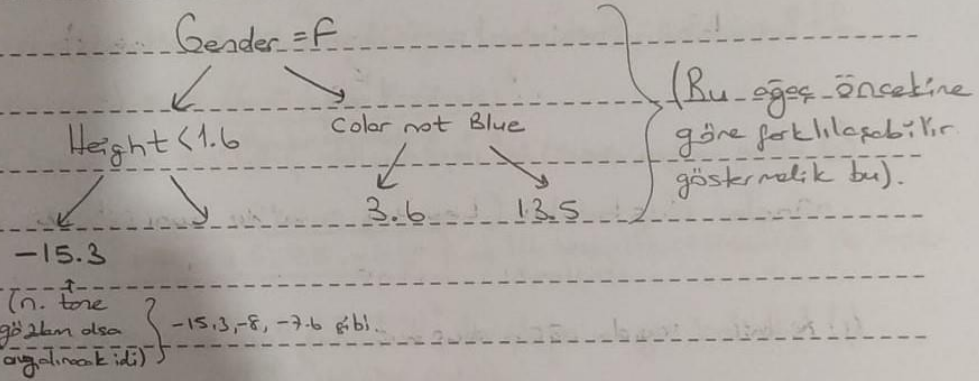
2. " " } Bunlarda aynı şekilde hesaplanır

3. " " }

Learning rate (Her ağaç için aynıdır ve overfitti engeller). Küçük adımlar ile gelişim sağlanır.

	Weight (Target)	Initial Res.	2nd Res.
4) 1. gözlem	88	15	$13.5 = 88 - 74.5$
2. gözlem	56	-17	$-15.3 = 56 - 71.3$
3. gözlem	77	4	$3.6 = 77 - 73.4$

5) Yeni ağaç bu kez yeni Residüaller ile oluşturulur.
(original weight'ler ile değil)



6) Şimdi elimizde 1 tane initial prediction (Avg)

2 // tree var

1. gözlem > Predicted weight = $73 + (0.1 \times 15) + (0.1 \times 13.5) = 75.85$

2. // > // " = $73 + (0.1 \times -17) + (0.1 \times -15.3) = 69.77$

3. // > // " = $73 + (0.1 \times 4) + (0.1 \times 3.6) = 73.76$

	Weight (Target)	Initial Res	2nd Res	3rd Res
1. gözlem	88	15	13.5	12.15
2. "	56	-17	-15.3	-13.77
3. "	77	4	3.6	3.24

7) Belirlediğimiz ağaç sayısına bağlı yukarıdaki işlemler yapılır ve finalize edilir.

b) Classification:

Regressyona çok benzer. (Logistic Regresyon prob. hesapları mantığı ile aynı)

1) Initial prediction belirlenir. (Regressyonda Avg.; sınıflandırmada log-odds).

$$\rightarrow \log\left(\frac{\text{count class 1}}{\text{count class 2}}\right)$$

2) Buradaki Residual = (Observed - Predicted)

1 veya 0 olacaktır.

3) Aynı şekilde residualler predict edilmesi için

ağaçlar oluşturulur. (Learning rate'de aynı şekilde yapılır)

4) İstenilen sayıda ağaç oluşturulduktan sonra final karar verilir.