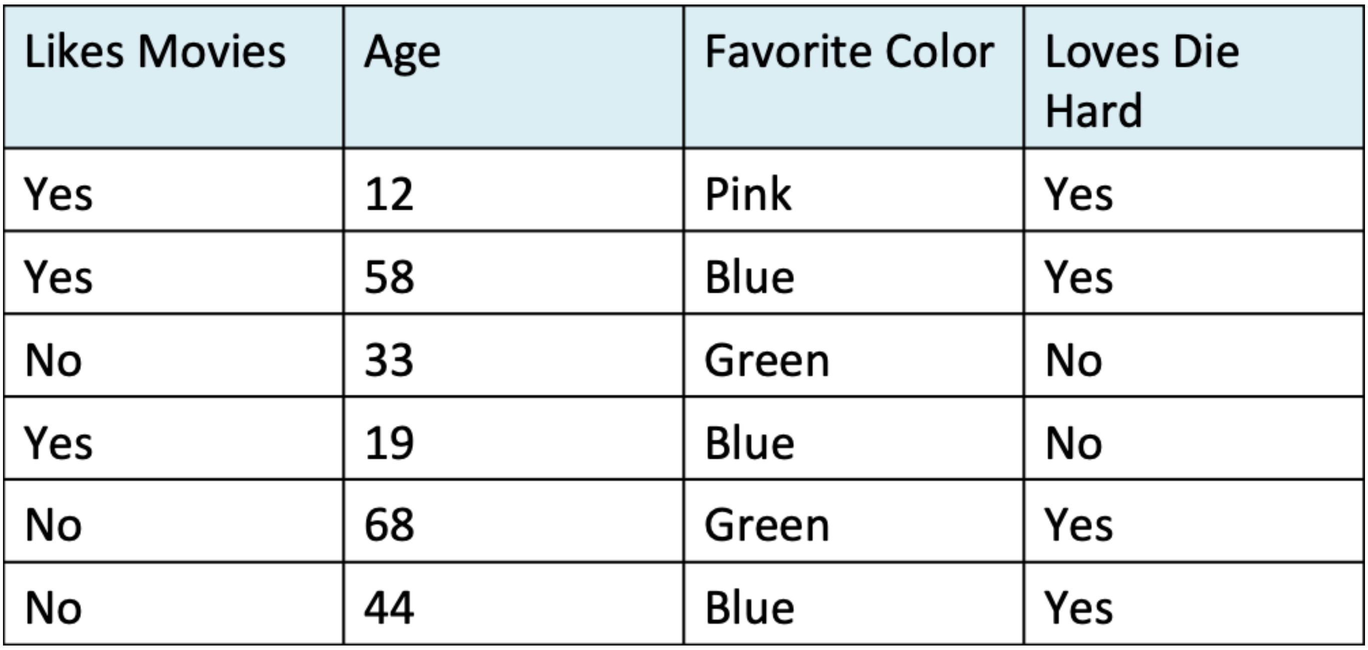
my notes: Gradient Boosting for Classification

reinec.medium.com/note-gradient-boosting-for-classification-c13f91e715b9

April 5, 2020

Boosting trains multiple weak learners (eg by taking the mean or a small DT) sequentially to form strong learner. Each model learn from the error made by the previous model

Weak learners are used in gradient boosting (eg using mean or a small DT) to predict if a person is a fan of die hard using the dataset below.



Likes Movies	Age	Favorite Color	Loves Die Hard
Yes	12	Pink	Yes
Yes	58	Blue	Yes
No	33	Green	No
Yes	19	Blue	No
No	68	Green	Yes
No	44	Blue	Yes

so an initial prediction is made using a weak learner by computing the log(odds), where odds is the number of yes-to-no ratio.

substituting the value of log(4/2) into the equation to derive the probability of loving die hard into the equation below, a value of 0.7 is obtained for all the data points.

$$probability\ of\ loving\ die\ hard = \frac{e^{\log\left(\frac{4}{2}\right)}}{1 + e^{\log\left(\frac{4}{2}\right)}}$$

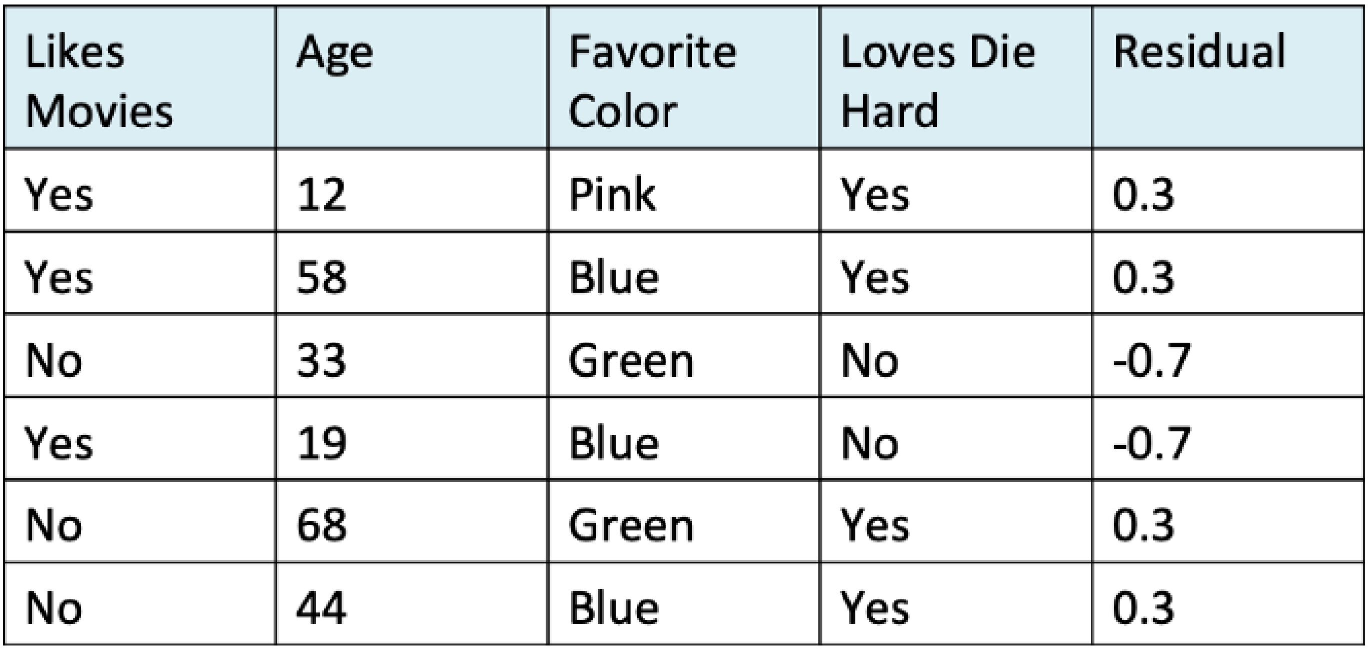
eq1

for most classification problem, a probability of > 0.5 would equate to a positive(in this case, loving die hard) label. Therefore, the weak learner has effectively classified all the data points as Loves Die Hard.

note: threshold value of 0.5 is just a commonly used value and can be adjusted accordingly

when the observed value for Loves Die Hard is Yes, probability = 1 and when observed value is No, probability = 0. we can then calculate residual using the equation, residual = y — ŷ.

residual (when yes) = 1–0.7 = 0.3



Likes Movies	Age	Favorite Color	Loves Die Hard	Residual
Yes	12	Pink	Yes	0.3
Yes	58	Blue	Yes	0.3
No	33	Green	No	-0.7
Yes	19	Blue	No	-0.7
No	68	Green	Yes	0.3
No	44	Blue	Yes	0.3

using the residual, we build another model to predict on it. (ie residual is used as target variable).

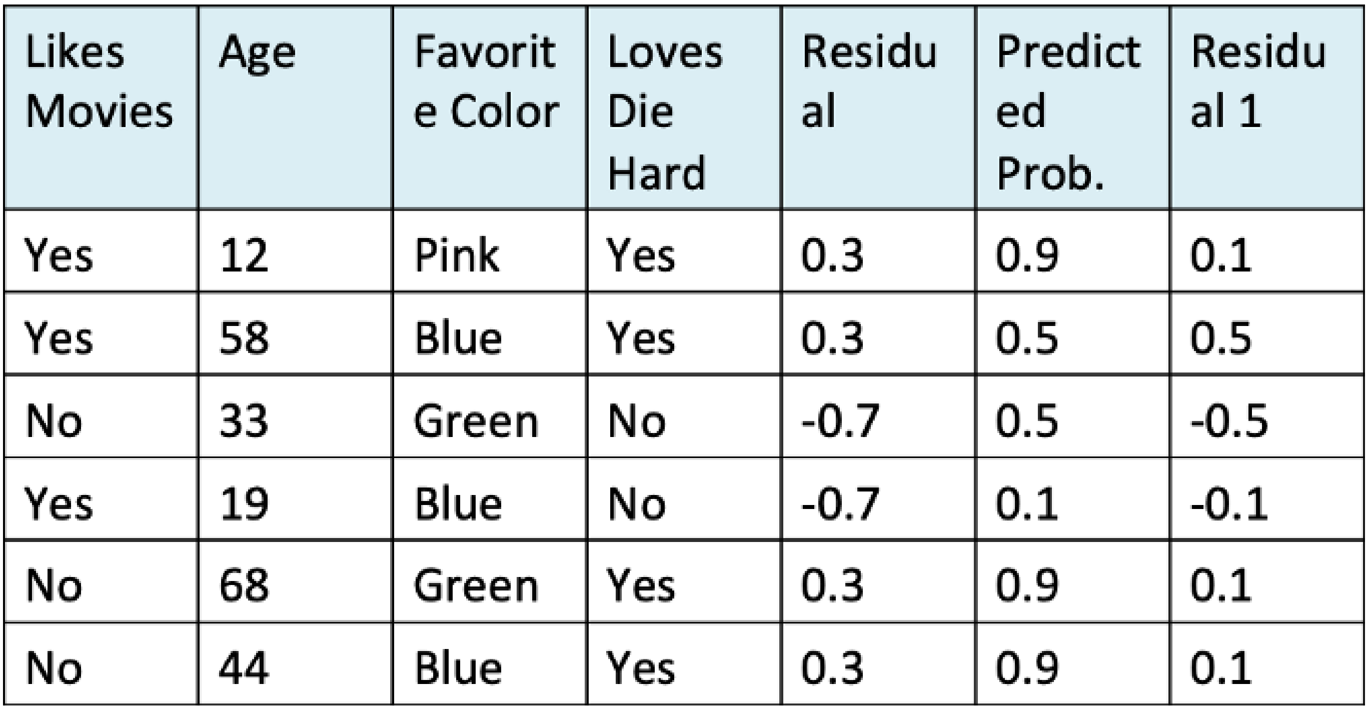
this 2nd model also restricts size of tree(weak learner). A scaling factor is applied to each tree (model), called the Learning Rate. The Learning rate (between 0–1) will prevent a high variance in the prediction by scaling(or limiting) the contribution from each tree.

Instead of taking the mean for of all the values that fall within each terminal node(as in Gradient Boosting for Regression), the output values for each terminal node is calculated using the formula below:

$$\frac{\Sigma Residual}{\Sigma[\text{Previous Probability x } (1 - \text{Previous Probability})]}$$

eq2

substituting the sum of log(odds) + (learning rate x terminal node output value from eq2) into eq1, we are able to obtain the predicted probability if a user loves die hard as shown in column Predicted Prob.



Likes Movies	Age	Favorite Color	Loves Die Hard	Residual	Predicted Prob.	Residual 1
Yes	12	Pink	Yes	0.3	0.9	0.1
Yes	58	Blue	Yes	0.3	0.5	0.5
No	33	Green	No	-0.7	0.5	-0.5
Yes	19	Blue	No	-0.7	0.1	-0.1
No	68	Green	Yes	0.3	0.9	0.1
No	44	Blue	Yes	0.3	0.9	0.1

The new residual values are calculated using column Loves Die Hard-Predicted Prob. to obtain column Residual 1. A new tree is created to train on Residual 1.

the steps are repeated until the max iterations or when error function does not change (ie additional trees fail to improve the fit) or when residuals becomes very small.

Note: the trees built for each iteration can be different (ie the splits and features used)