Gradient Boost - Classification

Aravindan T S

- Research paper titled ‘Machine learning-based prediction of COVID-19 diagnosis based on symptoms’ published on nature.com .
- This paper uses a machine-learning approach that trained on records from 51,831 tested individuals (of whom 4769 were confirmed to have COVID-19)

Article | Open Access | Published: 04 January 2021

Machine learning-based prediction of COVID-19 diagnosis based on symptoms

Yazeed Zoabi, Shira Deri-Rozov & Noam Shomron

npj Digital Medicine **4**, Article number: 3 (2021) | Cite this article

81k Accesses | **94** Citations | **48** Altmetric | Metrics

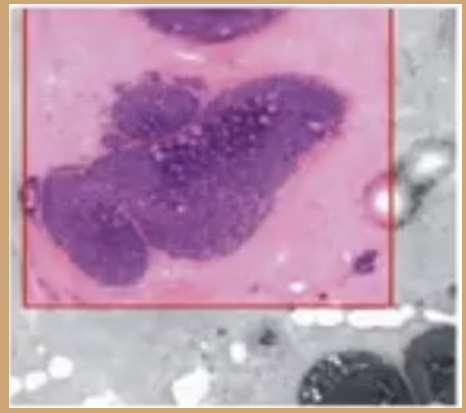
Abstract

Effective screening of SARS-CoV-2 enables quick and efficient diagnosis of COVID-19 and can mitigate the burden on healthcare systems. Prediction models that combine several features to estimate the risk of infection have been developed. These aim to assist medical staff worldwide in triaging patients, especially in the context of limited healthcare resources. We established a machine-learning approach that trained on records from 51,831 tested individuals (of whom 4769 were confirmed to have COVID-19). The test set contained data from the subsequent week (47,401 tested individuals of whom 3624 were confirmed to have COVID-19). Our model predicted COVID-19 test results with high accuracy using only eight binary features: sex, age ≥60 years, known contact with an infected individual, and the appearance of five initial clinical symptoms. Overall, based on the nationwide data publicly reported by the Israeli Ministry of Health, we developed a model that detects COVID-19 cases by simple features accessed by asking basic questions. Our framework can be used, among

- The model predicted COVID-19 test results with high accuracy using only eight binary features: sex, age ≥60 years, known contact with an infected individual, and the appearance of five initial clinical symptoms.
- This model can be implemented globally for effective screening and prioritization of testing for the virus in the general population.
- Predictions were generated using a gradient-boosting machine model built with decision-tree base-learners.
- Gradient boosting is widely considered state of the art in predicting tabular and is used by many successful algorithms in the field of machine learning.

Dataset

- This is a human cell sample and its attributes are given below the image.
- Is this a benign or malignant cell?
- The list below is a **dataset** with the attributes of several cell samples of patients believed to be at risk of developing cancer.
- Analysing this showed that many characteristics differed between benign and malignant cells.
- One can use this analysis to calculate whether a new sample will be benign or malignant with high accuracy.
- This is done by cleaning the data, selecting an algorithm and training the model to understand patterns of benign or malignant cells within the data.

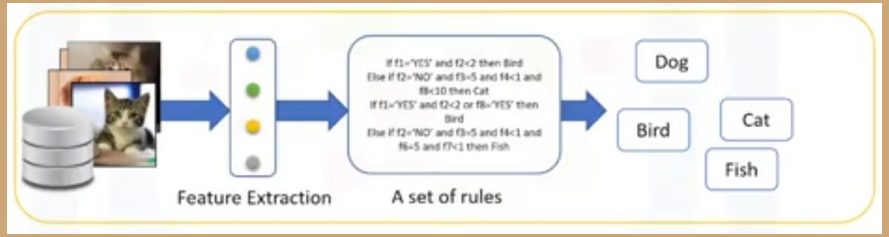


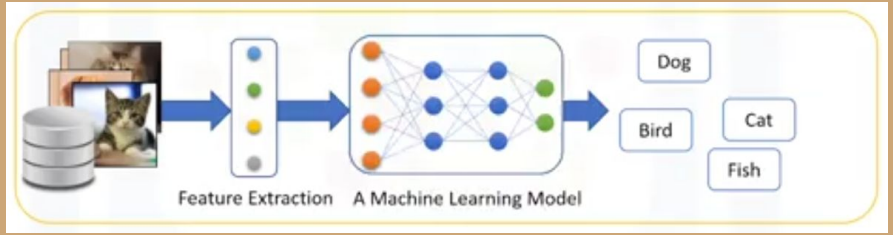
ID	Clump	UnifSize	UnifShape	MargAdh	SingEpiSize	BareNuc	BlandChrom	NormNucl	Mit	Class
1000015	6	1	1	1	7	1	3	1	1	

ID	Clump	UnifSize	UnifShape	MargAdh	SingEpiSize	BareNuc	BlandChrom	NormNucl	Mit	Class
1000025	5	1	1	1	2	1	3	1	1	benign
1002945	5	4	4	5	7	10	3	2	1	benign
1015425	3	1	1	1	2	2	3	1	1	malignant
1016277	6	8	8	1	3	4	3	7	1	benign
1017023	4	1	1	3	2	1	3	1	1	benign
1017122	8	10	10	8	7	10		7	1	malignant
1018099	1	1	1	1	2	10	3	1	1	benign
1018561	2	1	2	H	2	1	3	1	1	benign
1033078	2	1	1	1	2	1	1	1	5	benign
1033078	4	2	1	1	2	1	2	1	1	benign

Machine Learning Model

- A **machine learning model** has the ability to learn without being explicitly programmed.
- Assume that you want to develop a software to recognise and differentiate between different animals.
- Prior to machine learning, rules or methods were written to get computers to detect the animals.
- This was a failure because it needed a lot of rules highly dependent on the dataset used and was not generalised enough to detect out of sample cases.
- Machine learning allows us to look at the whole dataset with the corresponding types of animals.

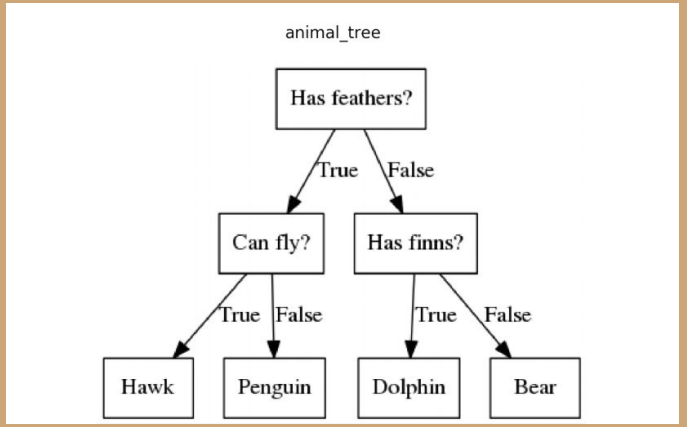


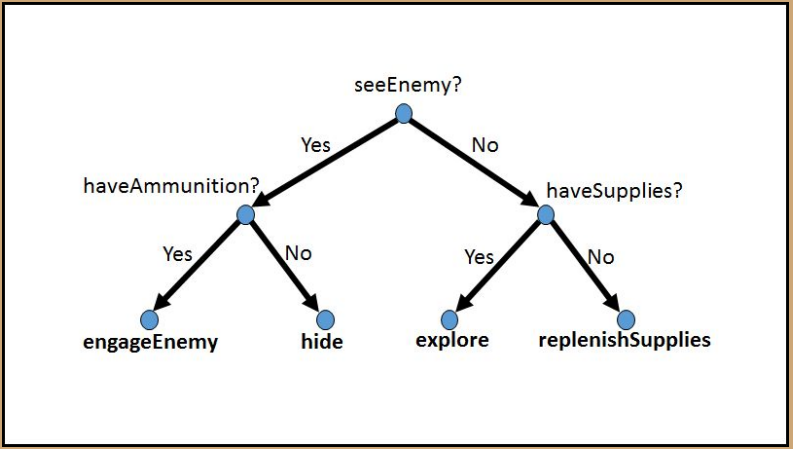


- It learns the pattern of each animal. These patterns are then used to predict animals from their attributes.
- Applications of machine learning : chat bots, websites recommending movies tv shows, ads, etc.

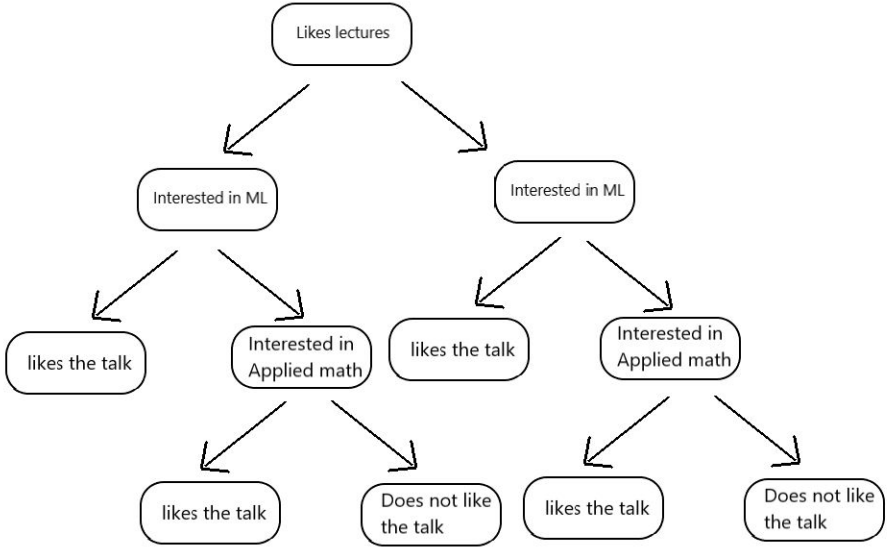
Decision Trees

- Decision Trees are a type of algorithm where the data is continuously split according to a certain parameter.
- Few examples of decision trees are given on the right.
- In general, a decision tree asks a question and classifies based on the answer.
- Simply put, you start at the top and work your way down till you get to a point where you cannot go any further. This is how you classify a sample.
- The very top of the tree is called the ‘Root Node’ or the ‘Root’.
- ‘Internal Nodes’ or ‘Nodes’ have arrows pointing to and away from them.





- ‘Leaf Nodes’ or ‘Leaves’ have arrows pointing to them but not away from them.
- We use the training set of the dataset to build the decision tree. We test the accuracy of the model using the validation set. The model should not be overfit.



Gradient Boost

- The Gradient boosting algorithm (popularly known as the Gradient boosting machine or GBM) is one of the most powerful algorithms in the field of machine learning.
- The principle behind boosting algorithms is first we built a model on the training dataset, then a second model is built to rectify the errors present in the first model.
- The main idea behind Gradient boosting algorithm is to build models sequentially and these subsequent models try to reduce the errors of the previous model.

Input : $\{(x_i, y_i)\}^n_{i=1}$, and a differentiable Loss Function $L(y_i, F(x))$

Step 1: Initialize model with a constant value : $F_0(x) = \text{argmin}_{\gamma} \Sigma L(y_i, \gamma)$

Step 2: for m = 1 to M:

(A) Compute $r_{im} = -\left[\frac{\partial L(y_i, F(x_i))}{\partial F(x_i)}\right]_{F(x)=F_{m-1}(x)}$ for $i = 1,\ldots,n$

(B) Fit a regression tree to the r_{im} values and create terminal regions R_{jm}, for $j = 1 \ldots J_m$

(C) For $j = 1 \ldots J_m$ compute $\gamma_{jm} = \Sigma L(y_i, F_{m-1}(x_i) + \gamma)$

(D) Update $F_m(x) = F_{m-1}(x) + v \Sigma \gamma_m I(x \in R_{jm})$

Step 3: Output $F_M(x)$

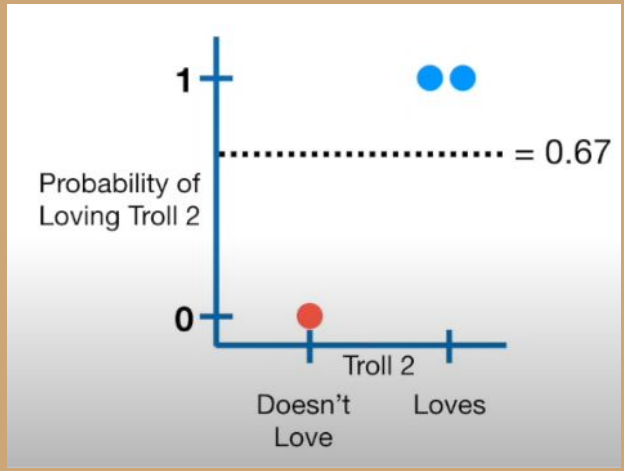
- x_i refers to each row of attributes that we use to predict the value. y_i refers to the output values. So, $\{(x_i, y_i)\}^n_{i=1}$ refers to the training dataset.
- A loss function is a measure of how good your prediction model does in terms of being able to predict the expected outcome.

- The blue dots represent people who love troll 2 and the red dot represents the person who does not love troll 2.
- So, predicted probability here is 0.67.
- Log(Likelihood) =

$$\sum_{i=1}^{N} y_i \times \log(p) + (1 - y_i) \times \log(1 - p)$$

- Here, p is the predicted probability and y_i are the observed values for loves troll 2.
- Upon plugging in the values in the equation, log(likelihood) of the blue dots is log(0.67) and that of the red dot is log(1-0.67).
- The better the prediction, the larger the log(likelihood).

Likes Popcorn	Age	Favorite Colour	Loves Troll 2
Yes	12	Blue	Yes
No	87	Green	Yes
No	44	Blue	No



- Hence, if we want to use log(likelihood) as a loss function, where smaller values represent better fitting models, we have to multiply the log(likelihood) by -1.
- So, it is : $$-\sum_{i=1}^{N} y_i \times \log(p) + (1 - y_i) \times \log(1 - p)$$

- Removing the summation (since the loss function sometimes only deals with one sample at a time) and replacing y_i with observed, the equation becomes,

$$-\left[\textbf{Observed} \times \log(p) + (1 - \textbf{Observed}) \times \log(1 - p) \right]$$

- ⇔ - Observed x log(p) - (1 - Observed) x log(1 - p)
- ⇔ - Observed x log(p) - log(1 - p) + Observed x log(1 - p)
- ⇔ - Observed x [log(p) - log(1 - p)] - log(1 - p)

$$\text{since } \log(p) - \log(1-p) = \frac{\log(p)}{\log(1-p)} = \log\left(\frac{p}{1-p}\right) = \log(\text{odds})$$

- ⇔ - Observed x log(odds) - log(1 - p)

$$\log(1-p) = \log\left(1 - \frac{e^{\log(\text{odds})}}{1+e^{\log(\text{odds})}}\right) = \log\left(\frac{1+e^{\log(\text{odds})}}{1+e^{\log(\text{odds})}} - \frac{e^{\log(\text{odds})}}{1+e^{\log(\text{odds})}}\right) = \log\left(\frac{1}{1+e^{\log(\text{odds})}}\right)$$

$$= \log(1) - \log(1+e^{\log(\text{odds})}) = -\log(1+e^{\log(\text{odds})})$$

- $\Leftrightarrow$ - Observed x log(odds) + log(1 + $e^{log(odds)}$)

  This will serve as the Loss Function.

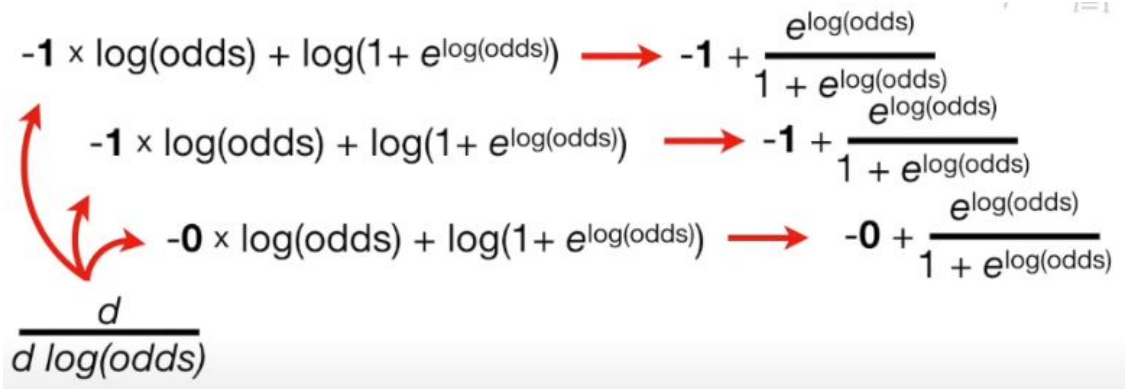
- Now we have to show that it is differentiable.

$$\frac{d}{d\ log(odds)} \text{-Observed} \times \log(odds) + \log(1 + e^{\log(odds)}) =$$

$$\text{-Observed} + \frac{e^{\log(odds)}}{1 + e^{\log(odds)}}$$

$$= \text{-Observed} + p$$

- Now, we start with step 1.
- We use maximum likelihood estimation to find the optimal initial prediction.
- Since we multiplied the log(likelihood) by -1, maximum likelihood estimation would minimise the predicted value.
- Taking the derivative of each term with respect to the log(odds),



- Replacing with p and setting the sum of the derivatives to p,

$$\mathbf{-1} + \boldsymbol{p} + \mathbf{-1} + \boldsymbol{p} + \mathbf{-0} + \boldsymbol{p} = 0$$

- p = ⅔. Now we convert the predicted probability into the log(odds).

  log(odds) = log(p/(1-p))

  = log((⅔)/(1-⅔)) = log(2/1) = 0.69

- So, this is the initial leaf $F_0(x)$. The model is initialized with this constant value.

- This is when we build the decision trees.
- We start by setting m = 1 (where m stands for the decision tree).
- We then compute the pseudo residuals:

$$r_{im} = -\left[\frac{\partial L(y_i, F(x_i))}{\partial F(x_i)}\right]_{F(x)=F_{m-1}(x)} \quad \text{for } i = 1,\ldots,n$$

- This is the derivative of the loss function with respect to the predicted log (odds). Plugging in the loss function,

$$\frac{d}{d\ log(odds)} -\textbf{Observed} \times \log(odds) + \log(1 + e^{\log(odds)})$$

- => $$-\text{Observed} + \frac{e^{\log(\text{odds})}}{1 + e^{\log(\text{odds})}}$$
- Multiplying by -1,

$$(\text{Observed} - \frac{e^{\log(\text{odds})}}{1 + e^{\log(\text{odds})}})$$

- => (Observed - p)
- Plugging in the most recent value of p,

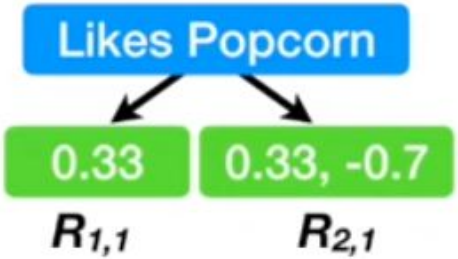
  => (Observed - 0.67)
- Now, we can compute the pseudo residuals for each sample.

- $r_{1,1} = (1 - 0.67) = 0.33$

  $r_{2,1} = (1 - 0.67) = 0.33$

  $r_{3,1} = (0 - 0.67) = -0.67$

- Now, we build a decision tree to predict the residuals. Here we take it as,



- $R_{1,1}$ and $R_{2,1}$ are the the terminal regions R_{jm}
- We have now fit a regression tree to the residuals and labelled the leaves.

- Now, we move on to the next step (calculating the output values).

$$\textbf{(C) For } j = 1 \ldots J_m \textbf{ compute } \quad \gamma_{jm} = \underset{\gamma}{\operatorname{argmin}} \sum_{x_i \in R_{ij}} L(y_i, F_{m-1}(x_i) + \gamma)$$

- The output value for each leaf is the value for gamma that minimizes this summation.
- Output value for $R_{1,1}$:

$$\gamma_{1,1} \quad = \underset{\gamma}{\operatorname{argmin}} \sum_{x_i \in R_{ij}} L(y_i, F_{m-1}(x_i) + \gamma)$$

- =>

$$\gamma_{1,1} \quad = \underset{\gamma}{\operatorname{argmin}} \sum_{x_i \in R_{ij}} - y_i \times [F_{m-1}(x_i) + \gamma] + \log(1 + e^{F_{m-1}(x_i)+\gamma})$$

- Since only x_1 goes to $R_{1,1}$, we can remove the sigma and substitute i with 1.

$$L(y_1, F_{m-1}(x_1) + \gamma) = -y_1 \times [F_{m-1}(x_1) + \gamma] + \log(1 + e^{F_{m-1}(x_1)+\gamma})$$

- For ease of computation, we approximate the loss function with a second

$$L(y_1, F_{m-1}(x_1) + \gamma) \approx L(y_1, F_{m-i}(x_1)) + \frac{d}{dF()}(y_1, F_{m-1}(x_1))\gamma + \frac{1}{2}\frac{d^2}{dF()^2}(y_1, F_{m-1}(x_1))\gamma^2$$

- Taking the derivative:

$$\frac{d}{d\gamma}L(y_1, F_{m-1}(x_1) + \gamma) \approx \frac{d}{dF()}(y_1, F_{m-1}(x_1)) + \frac{d^2}{dF()^2}(y_1, F_{m-1}(x_1))\gamma$$

- Now we solve for gamma:

$$\gamma = \frac{-\frac{d}{dF()}(y_1, F_{m-1}(x_1))}{\frac{d^2}{dF()^2}(y_1, F_{m-1}(x_1))}$$

- =>

$$\gamma = \frac{\textbf{Observed - } \boldsymbol{p}}{\frac{d^2}{dF()^2}(y_1, F_{m-1}(x_1))}$$

- =>

$$\gamma = \frac{\textbf{Residual}}{\frac{d^2}{dF()^2}(y_1, F_{m-1}(x_1))}$$

- The denominator, on expansion, is:

$$\frac{d^2}{d\ log(odds)^2}\textbf{-Observed} \times \log(odds) + \log(1 + e^{\log(odds)})$$

=> $$\frac{d}{d\ log(odds)}\textbf{-Observed} + \frac{e^{\log(odds)}}{1 + e^{\log(odds)}}$$

- => $\frac{d}{d\ log(odds)}$ **-Observed** $+ (1 + e^{log(odds)})^{-1} \times e^{log(odds)}$
- => $-(1 + e^{log(odds)})^{-2} e^{log(odds)} \times e^{log(odds)} + (1 + e^{log(odds)})^{-1} \times e^{log(odds)}$
- => $\frac{-e^{2 \times log(odds)}}{(1 + e^{log(odds)})^2} + \frac{e^{log(odds)}}{(1 + e^{log(odds)})}$
- => $\frac{-e^{2 \times log(odds)}}{(1 + e^{log(odds)})^2} + \frac{e^{log(odds)}}{(1 + e^{log(odds)})} \times \frac{(1 + e^{log(odds)})}{(1 + e^{log(odds)})}$
- => $\frac{-e^{2 \times log(odds)}}{(1 + e^{log(odds)})^2} + \frac{e^{log(odds)} + e^{2 \times log(odds)}}{(1 + e^{log(odds)})^2}$

- => $$\frac{-e^{2 \times \log(odds)} + e^{\log(odds)} + e^{2 \times \log(odds)}}{(1 + e^{\log(odds)})^2}$$
- => $$\frac{e^{\log(odds)}}{(1 + e^{\log(odds)})(1 + e^{\log(odds)})}$$
- => $$\frac{e^{\log(odds)}}{(1 + e^{\log(odds)})} \times \frac{1}{(1 + e^{\log(odds)})}$$
- => $$\frac{e^{\log(odds)}}{(1 + e^{\log(odds)})} \times \frac{1}{(1 + e^{\log(odds)})}$$
- => p (1 - p)

- So,

$$\gamma = \frac{\textbf{Residual}}{\boldsymbol{p} \times (1 - \boldsymbol{p})}$$

- => $$\gamma_{1,1} = \frac{0.33}{0.67 \times (1 - 0.67)}$$
- => $\gamma_{1,1}$ = 1.5

- Now we calculate the output value for the next leaf.
- Samples x_2 and x_3 go to leaf $R_{2,1}$.
- $$\gamma_{2,1} = \underset{\gamma}{\operatorname{argmin}} \sum_{x_i \in R_{ij}} L(y_i, F_{m-1}(x_i) + \gamma)$$
- => $$\gamma_{2,1} = \underset{\gamma}{\operatorname{argmin}} \left[L(y_2, F_{m-1}(x_2) + \gamma) + L(y_3, F_{m-1}(x_3) + \gamma) \right]$$
- $$L(y_2, F_{m-1}(x_2) + \gamma) \approx L(y_2, F_{m-1}(x_2)) + \frac{d}{dF()}(y_2, F_{m-1}(x_2))\gamma + \frac{1}{2}\frac{d^2}{dF()^2}(y_2, F_{m-1}(x_2))\gamma^2$$
$$L(y_3, F_{m-2}(x_3) + \gamma) \approx L(y_3, F_{m-1}(x_3)) + \frac{d}{dF()}(y_3, F_{m-1}(x_3))\gamma + \frac{1}{2}\frac{d^2}{dF()^2}(y_3, F_{m-1}(x_3))\gamma^2$$

- $$L(y_2, F_{m-2}(x_2) + \gamma) + L(y_3, F_{m-2}(x_3) + \gamma) \approx L(y_2, F_{m-1}(x_2)) + L(y_3, F_{m-1}(x_3))$$
$$+ \left[\frac{d}{dF()}(y_2, F_{m-1}(x_2)) + \frac{d}{dF()}(y_3, F_{m-1}(x_3)) \right] \gamma$$
$$+ \frac{1}{2} \left[\frac{d^2}{dF()^2}(y_3, F_{m-1}(x_3)) + \frac{d^2}{dF()^2}(y_3, F_{m-1}(x_3)) \right] \gamma^2$$

- $$\frac{d}{d\gamma} L(y_2, F_{m-2}(x_2) + \gamma) + L(y_3, F_{m-2}(x_3) + \gamma) \approx \left[\frac{d}{dF()}(y_2, F_{m-1}(x_2)) + \frac{d}{dF()}(y_3, F_{m-1}(x_3)) \right]$$
$$+ \left[\frac{d^2}{dF()^2}(y_3, F_{m-1}(x_3)) + \frac{d^2}{dF()^2}(y_3, F_{m-1}(x_3)) \right] \gamma = 0$$

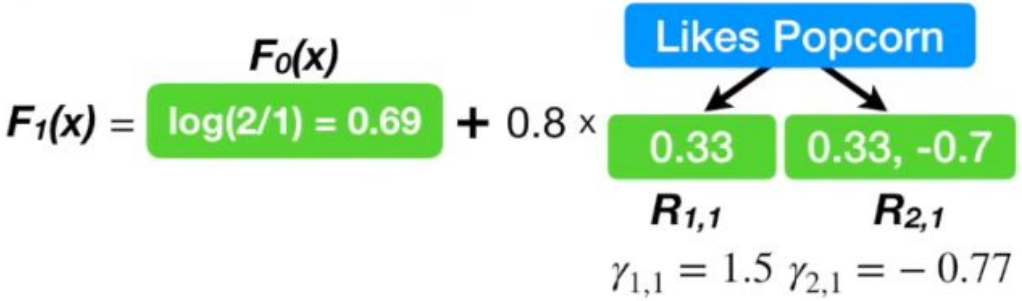
- $$\gamma = \frac{-\left[\frac{d}{dF()}(y_2, F_{m-1}(x_2)) + \frac{d}{dF()}(y_3, F_{m-1}(x_3))\right]}{\left[\frac{d^2}{dF()^2}(y_3, F_{m-1}(x_3)) + \frac{d^2}{dF()^2}(y_3, F_{m-1}(x_3))\right]}$$
- $$\gamma = \frac{-\left[-y_2 + p_2 + -y_3 + p_3\right]}{\left[\frac{d^2}{dF()^2}(y_3, F_{m-1}(x_3)) + \frac{d^2}{dF()^2}(y_3, F_{m-1}(x_3))\right]}$$

- $$\gamma = \frac{(y_2 - p_2) + (y_3 - p_3)}{\left[\frac{d^2}{dF()^2}(y_3, F_{m-1}(x_3)) + \frac{d^2}{dF()^2}(y_3, F_{m-1}(x_3))\right]}$$
- $$\gamma = \frac{\text{Residual}_2 + \text{Residual}_3}{\left[\frac{d^2}{dF()^2}(y_3, F_{m-1}(x_3)) + \frac{d^2}{dF()^2}(y_3, F_{m-1}(x_3))\right]}$$
- $$\gamma = \frac{\text{Residual}_2 + \text{Residual}_3}{\left[p_2 \times (1 - p_2) + p_3 \times (1 - p_3)\right]}$$

- So,

$$\gamma_{2,1} = \frac{0.33 + -0.67}{[\,0.67 \times (1 - 0.67)\,] + [\,0.67 \times (1 - 0.67)\,]}$$

- => $\gamma_{2,1} = -0.77$

- Now, we make a new prediction for each sample.
- 

- New prediction for sample 1 : $0.69 + 0.8 \times 1.5 = 1.89$

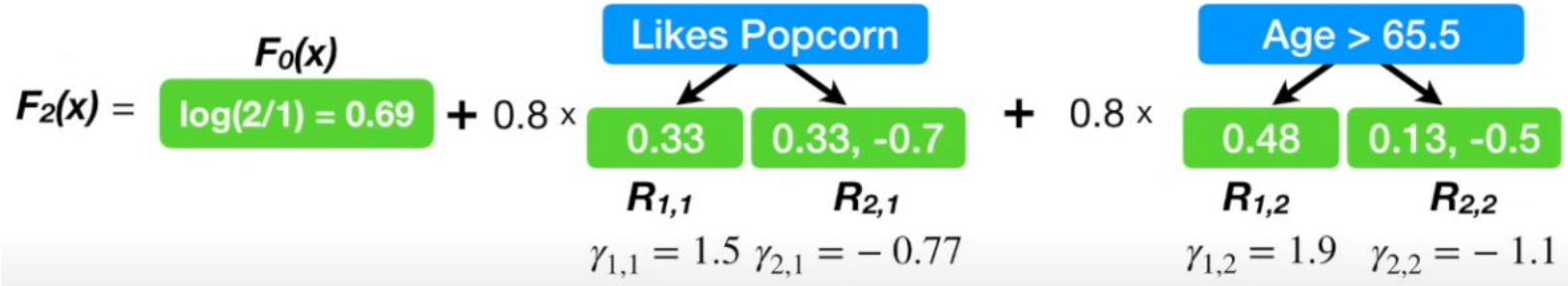
  New prediction for sample 2 : $0.69 + 0.8 \times -0.77 = 0.07$

  New prediction for sample 3 : $0.69 + 0.8 \times -0.77 = 0.07$

- Now, we set m = 2 and repeat the whole process.

  We calculate new residuals for the new predictions, create a new regression tree, calculate output values and make new predictions.

- This is then repeated for all m < M. Here, let us assume M = 2.
- If M = 2, $F_2(x)$ is the output from the Gradient Boost algorithm.



- Now, if we receive some new data, we would use $F_2(x)$ to predict it's output.
- Assume that we received the following new data:

Likes Popcorn	Age	Favorite colour	Loves Troll 2
Yes	90	Green	

- Predicted log(odds) that this person loves troll 2 = log(2/1) + (0.8 x 1.5) + (0.8 x 1.9) = 3.4
- Predicted probability = $e^{3.4}/(1 + e^{3.4})$ = 0.97

References

- Zoabi, Y., Deri-Rozov, S. & Shomron, N. Machine learning-based prediction of COVID-19 diagnosis based on symptoms. npj Digit. Med. 4, 3 (2021). https://doi.org/10.1038/s41746-020-00372-6
- Starmer, Josh. [StatQuest with Josh Starmer]. (2022, April 20). *Gradient Boost Part 4 (of 4): Classification Details.* [Video]. Youtube. https://www.youtube.com/watch?v=StWY5QWMXCw
- Prasanna Sahoo, *Probability and Mathematical Statistics* (Louisville: University of Louisville, 2008).

Thank You