We at The Data Monk hold the vision to make sure everyone in the IT industry has an equal stand to work in an open domain such as analytics. Analytics is one domain where there is no formal under-graduation degree and which is achievable to anyone and everyone in the World.

We are a team of 30+ mentors who have worked in various product-based companies in India and abroad, and we have come up with this idea to provide study materials directed to help you crack any analytics interview.

Every one of us has been interviewing for at least the last 6 to 8 years for different positions like Data Scientist, Data Analysts, Business Analysts, Product Analysts, Data Engineers, and other senior roles. We understand the gap between having good knowledge and converting an interview to a top product-based company.

Rest assured that if you follow our different mediums like our blog cum questions-answer portal www.TheDataMonk.com , our youtube channel - The Data Monk, and our e-books, then you will have a very strong candidature in whichever interview you participate in.

There are many blogs that provide free study materials or questions on different analytical tools and technologies, but we concentrate mostly on the questions which are asked in an interview. We have a set of 100+ books which are available both on Amazon and on The Data Monk e-shop page

We would recommend you to explore our website, youtube channel, and e-books to understand the type of questions covered in our articles. We went for the question-answer approach both on our website as well as our e-books just because we feel that the best way to go from beginner to advance level is by practicing a lot of questions on the topic.

We have launched a series of 50 e-books on our website on all the popular as well as niche topics. Our range of material ranges from SQL, Python, and Machine Learning algorithms to ANN, CNN, PCA, etc.

We are constantly working on our product and will keep on updating it. It is very necessary to go through all the questions present in this book.

Give a rating to the book on Amazon, do provide your feedback and if you want to help us grow then please subscribe to our Youtube channel.

Ada Boost

Below is how we proceed with the article:

Module 1:

- First let's revise some basic concepts of machine learning, decision trees, ensemble learning and terms related to it (if you are comfortable with decision tree concepts and terms feel free to skip to module 2).

Module 2:

- In this module we will see exactly how Ada boost works and uncover the math behind it (don't worry we have got examples covered for all basic concepts to make it clear).

Module 3:

- In this module to get the flavour of algorithm, we take a small data set (2 features) and do the math and built the Ada boost model and see how exactly each step goes on.

Module 4:

- In the Final section, we use SKLEARN module and built AdaBoost model from scratch (we use breast cancer data set for this module) and achieve an accuracy of almost 96%.

Finally

Summary: main ideas behind Ada Boost

All of this is going to be in questionnaire format so that you get hands on knowledge of what type of questions, you might face in interview with respect to this algorithm.

Module 1:

1) **What is machine learning?**
   Machine learning is training the computer program to form a model using the data, which is expected to produce accurate predictions when the model is applied on a new data.

2) **What is Bias?**
   Error or difference between model prediction and ground truth.
   The more the Bias the more difference Between the actual shape and function shape.
   If the bias is low then the error is low, so the function could adapt to the ground truth.

3) **What is Variance?**
   Variance is the error caused due to too much of adaptability. When we say the variance is high, what we mean is, the function is too well adapted that any new data point even if it is a bit away from the current points the function couldn't identify the new point.

4) **What are the effects of high bias and high variance?**
   High bias leads to:
   - Under fitting model
   - High error on both train data and validation/test data
   - Over-simplified model

   High variance leads to:
   - Over fitting model
   - High error on test data and least error on train data (since it is too well adapted to train data could not perform well on test data)
   - Over-complex model

5) **What is bias-variance trade off?**
   High variance-low bias leads to accurate but inconsistent. Inconsistent we mean by testing with test data it does not give proper results

Low variance-high bias leads to inaccurate but consistent. New data doesn't affect the model much.

6) **What is supervised machine learning model?**
Machine learning models that require labelled training data and they correct or better the model by finding the difference between the actual output and predicted output.
Ex: linear regression, neural networks, decision trees

7) **What is unsupervised machine learning model?**
Unsupervised models do not take labels for modelling. They form patterns or clustering groups based on some criteria.
Ex: hierarchical, k-means, Gaussian mixture models

8) **What is the difference between inductive and deductive learning?**
Inductive learning is the process of using observations to draw conclusions.
Deductive learning is the process of using conclusions to form observations.

9) **Explain one hot encoding and label encoding in classification problems?**
One hot encoding is converting categorical labels into binary numbers by extending the dimensionality of the data set. Label encoding is converting label form into numeric form.
One hot encoding increases the dimensionality whereas label encoding does not increase the dimensionality.

10) **What is cross validation?**
Cross-validation is dividing the data into namely three parts: training data, testing data, validation data.
Data is split into k parts and k-1 parts are used for training and the remaining part is used for testing the model which was trained on k-1 parts.
The above mentioned is done for all the k parts (at each stage 1 part is left for testing) and by the end of cross validation, all the samples will be exposed as both testing and training.

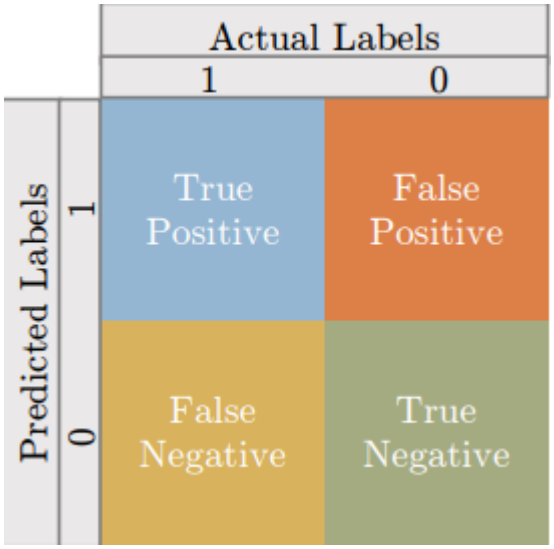
11) **What are some of the performance metrices of classification models apart from accuracy and why is accuracy not always the best choice?**

Accuracy does not always give the true picture of the situation.
For example, if they are total 100 labels out of which 90 are into label 1 ,10 are into label 2, even when you say all 100 as label 1, you are correct 90 out of 100 times but categorizing all labels into one side is not an ideal way to model the data.

We have other performance metrices to check the quality of classification model.

- Precision
- Recall
- F1 score
- Specificity

12) Give formulas to all the performance metrics used for classification problems?



1) Precision= $\frac{TP}{TP+FP}$

2)Specificity= $\frac{TN}{TN+FP}$

3) Recall= $\frac{TP}{TP+FN}$

4) F1 Score= $2 * \left(\frac{prec*recall}{prec+recall}\right)$

5) Accuracy=$\frac{TP+TN}{TP+TN+FN+FP}$

13) What is an ROC curve used in classification analysis?

Roc stands for Receiving Operating Characteristics is a graph plot against True positive rate and false positive rate.
The higher the area inside curve the better the model/classification at predicting labels. A y=x line in ROC is worst/useless.

14) How to avoid overfitting?

These are some of the methods to avoid overfitting:

- Don't make the model complex: take fewer variable which will reduce variance, thereby removing some noise in the data set.
- Use techniques such as cross validation.
- Use regularization techniques such as ridge and lasso to penalize certain model parameters.

15) What is the difference between classification and regression model?

Classification model is used when the outputs are discrete like the binary numbers or t shirt sizes.

On the other side regression deals with continuous data like predicting the weather tomorrow.

16) What is Clustering?

Clustering is the process of grouping the samples based on some measure such as distance. They must be similar to one group and dissimilar to the other.

A few types of clustering:

- Hierarchical Clustering
- K means clustering
- Density-based clustering
- Fuzzy clustering

17) What is reinforcement learning?

Reinforcement learning is different from both supervised and unsupervised, it neither has data nor labels. Reinforcement learning has some kind of reward system, where some kinds of actions are given points while other movements are not encouraged.

Example of this could be the machine which learns chess game and beats even the world class chess champions.

18) What is a model learning rate?

The learning rate is a tuning parameter that specifies the step size or the leap taken during the model training. In layman terms, the learning rate decides how big the step is towards minimizing error.

19) How to choose a classifier based on data?

If the data has fewer training examples, the last thing we need is an algorithm which quickly overfits so when the samples are small choose an algorithm that has correct bias and less variance.

Conversely, when we have a large number of samples, we choose the algorithm which gives high variance since it involves complex relationships.

20) What are decision trees?

Decision tree algorithm is a supervised machine learning algorithm. which appears like a top-down tree where at each node we take a question from one attribute and we move downwards based on the answers we get from each node and when we reach base node, we form a decision -make a classification.

21) Why does the decision tree suffer from overfitting often?

Decision trees when they have less samples and many variables then they tend to overfit because there are lots of nodes so it adapts to training data too well thus increasing the variance.

22) What is the difference between parameters and hyper-parameters?

A parameter is a variable that is internal to the model. These values depend on the type of data. These are often part of trained models.

Example: weights, bias, etc

A hyper parameter is a variable external to the model whose values have nothing to do with how the data is structured or what variables are present in the data. They are often used to estimate model parameters.

Learning rate, hidden layers, etc

23) What are position terms used in decision trees?

- root node: no incoming edge zero or more outgoing edge
- internal node: one incoming edge and 2 or more outgoing edge
- leaf node: exactly one incoming, no outgoing edges

24) What is pruning in decision trees?

It is done in order to reduce overfitting in decision trees. Pruning is nothing but removing branches from decision trees and clubbing up the samples which that branch split.

25) Are decision trees sensitive to outliers?

No, decision trees in fact, all tree algorithms are not sensitive to outliers. They are robust to outliers since one odd sample doesn't affect too much at the node level decision of trees.

26) What is information gain?

Before information gain let's define entropy. Entropy suggests the total randomness in a split of a decision tree. Information gain is decrease or increase in entropy when a node is split.

$$IG(Y, X) = E(Y) - E(Y/X)$$

27) What is the disadvantage of information gain?

Information gain tends to prefer nodes with large number of partitions with small samples for each node and pure instead of the nodes with less partitions which has some high level of impurity. This sometimes might not be the ideal choice.

28) What are some of the advantages of decision trees?

Decision trees are easier to understand. In layman terms they are just "if-else loops" except that boundary conditions are set by the algorithm itself based on the data and labels.

Decision trees are non-parametric hence decision trees are robust to outliers, and have relatively few parameters to tune.

Another advantage could be: it needs less data cleaning cases of missing values and outliers have less significance of how the tree structure is formed.

The data can be used to generate important insights on the probabilities, costs, etc.

29) What are some of the disadvantages of decision trees?

Decision trees are not suitable when your data set is of completely continuous values or when most of the attributes are continuous.
Decision trees do not perform well when the number of attributes are more and samples are less.
Pruning algorithms used to reduce overfitting can be quite computationally expensive.

30) How does the decision tree handle continuous variables?

Decision tree identifies certain threshold values by having information gain as a parameter in deciding the threshold value

31) What are the different types of attributes we encounter while performing decision trees?

- Continuous: continuous range of values
- Nominal: two or more distinct values (yes/no, sex: male/female/other)
- Ordinal: nominal attributes but have certain increasing or decreasing level of degree or measure (good, better, best; low, high)

32) When can a categorical value be treated as a continuous variable and what effect does it have when done so?

A categorical predictor can be treated as a continuous one when the nature of data points it represents is ordinal. If the predictor variable is having ordinal data, then it can be treated as continuous and its inclusion in the model increases the performance of the model.

For example: good, better, best can be 1,2,3.

33) List some of the cross validation techniques that could be applied to classification problems?

- K-fold
- Stratified k fold
- Leave one out
- Bootstrapping
- Random search cv

- Grid search cv

34) Show all the splitting criteria of decision trees in one graph to have a holistic view?

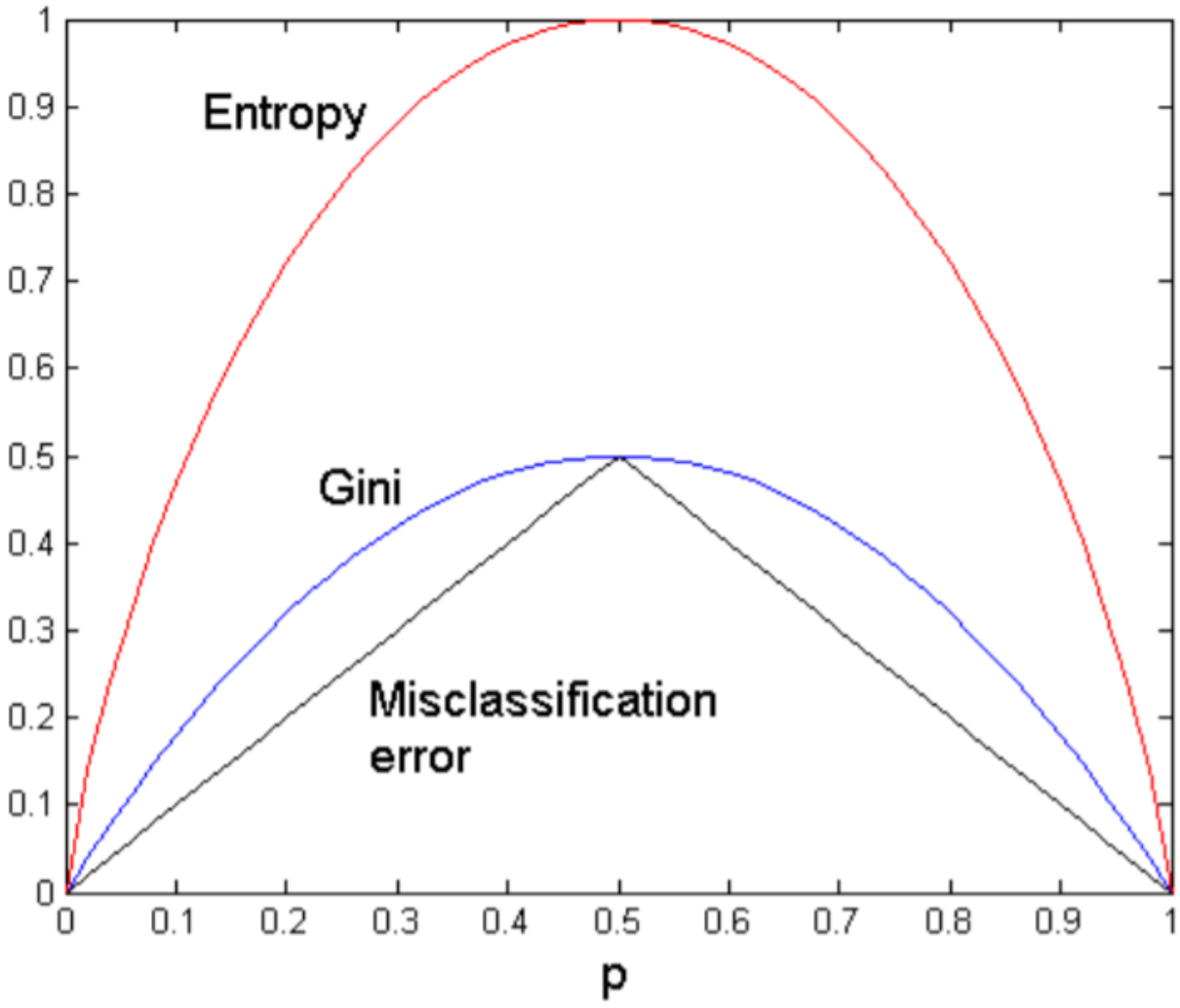


Fig 1.1

35) What are some of the methods of decision tree pruning?

some of the methods are:

i. Minimum error pruning
ii. Reduced error pruning
iii. Minimum cost complexity error pruning

36) What needs to be done when the decision tree overfits?

Pruning, generally overfit happens when we have many nodes. These nodes continuously split and will very less samples at leaf node eventually over fitting so we merge some of the end nodes thus pruning.

Early stopping: stop the algorithm before it becomes a fully grown tree

37) What is ensemble Learning?

It is a machine learning technique where many weak learners are trained and combined in a particular order/sequence. The prediction is formed by combining the results of all weak learners.

38) What are the different types of Ensemble learning methods and how do they differ from each other?

(Previously we said ensemble learning is about combing weak learners and using them in decision making process, so these categories in ensemble learning are based on how these weak learners combine)

Ensemble techniques can be broadly classified into 2 types

- Bagging (Bootstrap Aggregation): In bagging technique we parallelly (training of one weak learner doesn't depend on training the other) train all the weak learners and combine them with some kind of deterministic averaging process.
- Boosting: In boosting we train them in a sequential manner (training one model will have an effect on the next trained weak learner).

39) If your dataset is having high variance, how would you handle it?

If the dataset has high variance, then we can always use bagging ensemble technique. Bagging algorithm splits the data into subgroups with sampling done from random data. After we are done with split, random data is used to produce model or algorithm. Then we use polling technique to combine all the predicted outcomes of the model.

40) What are strong learners?

It consists of single classifier which produce high accuracy with arbitrarily small classification error.

- ex: Logistic regression, SVM, decision trees

41) What are weak learners?

a classifier that is just better than random. It usually doesn't produce high accuracy. These are used as base learners for ensemble methods

ex: bagging and boosting methods

Module 2:

42) In short how does the Ada Boost algorithm works?

Fig 2.1

Ada boost is an ensemble boosting technique, where, initially each sample (each row) gets a sample weight (initially it gets equal weight) and we train stumps (single variable decision tree) with all variables and select the stump which best classifies them (using metrices which we will discuss further).

Now, we are ready with our first stump, the next tasks at hand are:

1) Find the total error (sum of weights of misclassified samples) and using it Calculate the "amount of say" or a quantity that says how important this stump of all the stumps (which we will generate as we go further)

2) Then, we update the weights of each sample -giving high weight to those samples which were predicted wrong by our previous stump (we will discuss methods for doing this later) and lower weights to those which were classified correctly by our previous stump
3) Now we generate a new data set from our original data set such that the samples(rows) with high weights dominate the new data set and low weight sample don't repeat as often or as much as high weighted samples.
4) Now we repeat the same process again -creating a stump and updating weights and calculating the amount of say updating weights it goes on till we reach a certain number of stumps (which is given as input to the Ada Boost instance).

As we go further into the article, we will know how these steps are actually implemented.

43) What are the metrics used to measure the quality of decision trees (that are used in making stumps in Ada boost)?

In 42th question (about Ada boost) we discussed that out of all the initial stumps we select one stump, we do this by using a metric which defines the quality of stump

These can be namely:

1) Gini Index/impurity
2) Entropy (measure the randomness in the tree)

44) How does Gini index work show with an example?

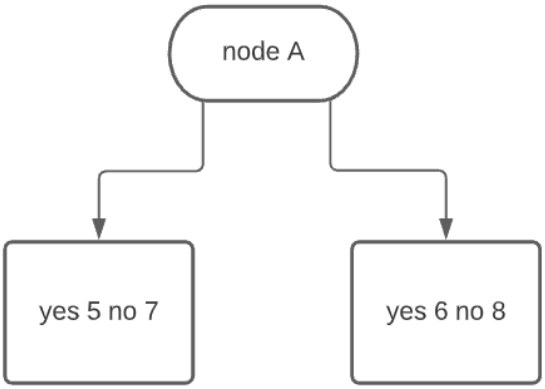


Fig 2.2

Let’s consider this stump (decision tree) and calculate Gini index for it

Gini index formula:

$$gini_{index} = 1 - \sum_j \left[p\left(\frac{j}{t}\right)\right]^2$$

Now using this for leaf node 1

$G_{\text{left node}}=1-(5/12)^2-(7/12)^2=0.4861$

Now for leaf node 2

$G_{\text{right node}} =1-(6/14)^2-(8/14)^2$

$=0.489$

Now we find $G_{\text{Node A}}$

Total samples are (5+7) + (6+8) =26

leaf node1 =12, node 2=14

$G_{\text{node A}}=(12/26)*G_{\text{left node}} +(14/26)*G_{\text{right node}}$

(12/26)*0.4861 + (14/26)*0.489 = 0.627

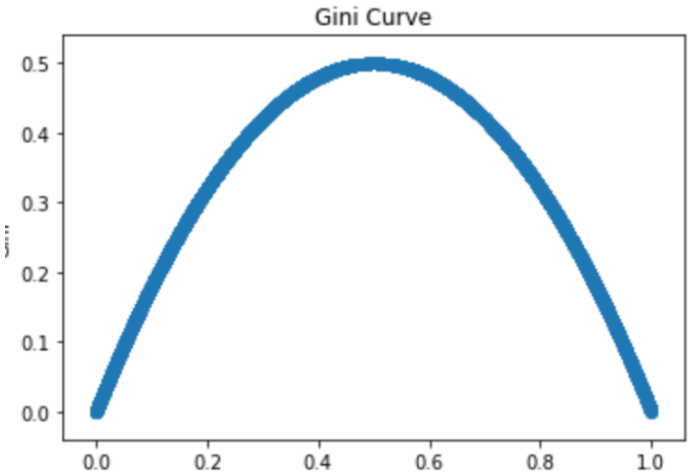


Fig 2.3

Gini index graph (max is 0.5 when a leaf node equally divides the samples(5-yes,5-no))

Gini should be as low as possible for stumps to perform better, for perfect classification Gini is 0.

45) what are different important cases of Gini Index?

Case 1: when we have a perfect tree – all are classified on the right node.

$$\text{Gini index/impurity} = 1-(1^2+0^2) = 0$$

Case 2: the most imperfect tree -half of samples are classified on one side other half on the other side.

$$\text{Gini index/impurity} = 1-(0.5^2+0.5^2) = 0.5$$

Case 3: another perfect case -all are classified on left node.

$$\text{Gini index /impurity} = 1-(0^2+1^2) = 0$$

46) Out of the 2 (Gini impurity, Entropy) which one should be selected while training?

When dealing with small data sets, it doesn't matter whichever we take, they tend to give similar results.

To answer the above question, it really depends on the requirement, each has its own advantage

- Gini impurity is computationally efficient since entropy involves logarithm function, it is a bit computationally expensive.
- Whereas entropy tends to produce stable trees by scattering the most frequent class whereas Gini impurity doesn't suffice the purpose.

47) When 2 nodes have Gini index 0.2,0.32 which node would you consider for your stump?

For a good classifier Gini index is always low, so classifier with Gini index 0.2 is better for the model as a stump.

48) How does entropy metrics work?

$$\boldsymbol{entropy} = -\sum_{\boldsymbol{j}} \boldsymbol{p}\left[\frac{\boldsymbol{j}}{\boldsymbol{t}}\right] * \mathbf{log_2}\, \boldsymbol{p}[\frac{\boldsymbol{j}}{\boldsymbol{t}}]$$

P[j/t] is the relative frequency of class j at node t

Let's calculate entropy for the above tree (fig 2.2)

Entropy of left node:

$$E1 = -(\frac{5}{12} * \frac{5}{12} + \frac{7}{12} * \frac{7}{12})$$

After calculating we get E1=0.979(appx)

Similarly for right node(E2):

$$E2 = -(\frac{6}{14} * \frac{6}{14} + \frac{8}{14} * \frac{8}{14})$$

We get E2 =0.9852

Total Entropy of node a = (12/26) *E1 + (14/26) *E2

=0.982

Entropy varies from 0 to 1 (the lower the entropy the better the tree at classifying)

49) Formula to calculate the amount of say or "the importance of classifier "in Ada Boost?

$$amount\ of\ say = 0.5 * \log_e \frac{1 - Total_{error}}{Total_{error}}$$

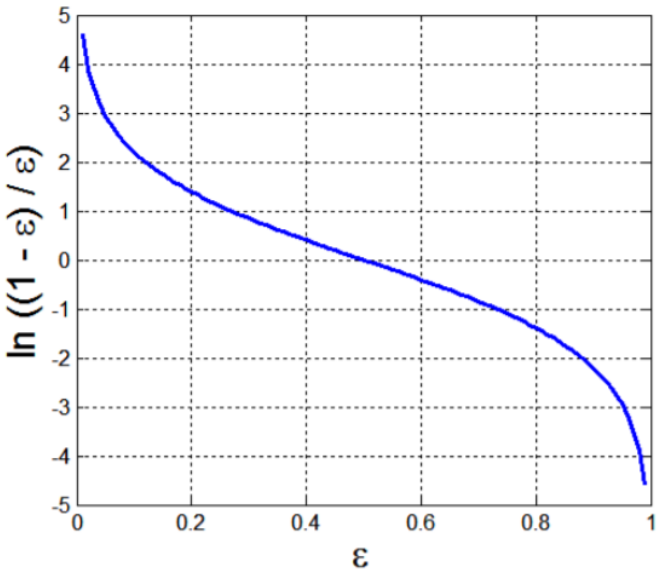


Fig 2.4

This is how the graph would look like for "amount of say".
Where Total error is the sum of weights for samples that are misclassified by our selected stump (in module 2 with live example you will see how this formula exactly works and where it works).

Inference: we observe that if our total error is large then we would have a less weight and with less total error, we get more weight for the tree (that makes sense right, since if tree performs bad (high total error) then we should not let it have high importance and the opposite is also true).

50) How do we update the sample weights in Ada Boost?

(Go back to question 42 and see step 2, earlier we said we update the sample weight, now let's see how we exactly do that)

Basic intuition: This updating process should be in such a way that we concentrate more on samples which couldn't be classified correctly earlier, so our method should give high weight to misclassified samples and low weight to correctly classified.
That's exactly how it happens

For the samples which were misclassified

$$weight_{new} = weight_{previous} * e^{amount\ of\ say}$$

For the samples which were classified correctly

$$weight_{new} = weight_{previous} * e^{-amount\ of\ say}$$

Now we Normalize the weights (we convert them in such a way that sum of all sample weights)

$Weight_{updated}$ = $weight_{new(i)}/(\sum(weight_{new}))$

51) how to produce <u>new data set</u> for the next stump using the present stump and dataset?

(In the previous question we saw how to update the weights for each sample)

Suppose these are our new weights for each sample (assuming there are only 7 rows in the dataset).

0.07
0.07
0.07
0.49
0.07
0.07
0.07
0.09

Fig 2.5

these are normalized since the sum is 1

Now we find cumulative of table in fig 2.6

We get

0.07
0.14
0.21
0.70
0.77
0.84

0.91
1.00

Fig 2.6

whenever we take cumulative, the last row needs to be 1 -since the sum of weights is 1

Okay we are all set, what we do now, is we generate a random number from 0-1 and if the number is between 0- 0.07 we take 1st sample into our new dataset.

If it's between 0.07-0.14 we choose 2nd sample and if its 0.21-0.70(observation: wow, that quite a big range compared to our previous ranges).

And we go on that way.

Intuition:

So, what's happening in the background? why are we doing this?

When we take cumulative and find ranges for respective sample, we observe that sample with high weight is having high range and this indicates that it has a higher probability of entering the new data set because the probability of choosing a number between 0-0.07 is less compared to choosing 0.21-0.70

What is the outcome?

Samples which have high weights are going to repeat more often than the samples with low weight.

52) Finally, after forming these stumps how do we do classifications/predictions?

Let's take an imaginary sample and imagine passing it to all the stumps and display the classification it did.

For example, our classification is to find, if Chintu went to school

Let's ask our stumps regarding it.

Stumps that said, Chintu went to school.

	Weight of stump
Stump1	0.6
Stump3	0.25
Stump5	0.4

Fig 2.7

Stumps that say Chintu did not go to school.

	Weight of stump
Stump 2	0.4
Stump 4	0.54

Fig 2.8

Sum of weights: table 1=0.6+0.25+0.4=1.25

Sum of weights: table 2 = 0.4+0.54=0.94

Since table1>table 2 decision made by table 1 prevails and Chintu did go to school (such a good boy).

Now we are done with basic understanding of how Ada boost forms the model and how we use multiple weak learners to predict/classify the samples

Module 3:

Let us take a small data set and see how the whole process is done in creating Ada Boost model in a stepwise manner

Dataset

There are 2 features

Feature1

Feature2

And 3rd column is the target variable.

8 samples in the data set

index	feature 1	feature2	classification
1	yes	yes	yes
2	no	yes	yes
3	yes	no	yes

4	yes	yes	yes
5	no	yes	no
6	no	yes	no
7	yes	no	no
8	yes	yes	no

Fig 3.1

Step 1:

53) What is the first step you do for a data set to train it in Ada boost?

We add weights to each row, initially they are all same

Initial_weight = 1/ (total no of samples)

index	feature 1	feature2	classification	initial_sample_weight
1	yes	yes	yes	1.0/8
2	no	yes	yes	1.0/8
3	yes	no	yes	1.0/8
4	yes	yes	yes	1.0/8
5	no	yes	no	1.0/8
6	no	yes	no	1.0/8
7	yes	no	no	1.0/8
8	yes	yes	no	1.0/8

Fig 3.2

See in the above figure the initial weights were 1/8 since there are 8 samples

Step2:

54) What would you do after initializing the weights?

We form stumps with all the features and see which performs better and select it as base learner

Stump formed using feature 1(to understand have a look at data set fig 3.1)

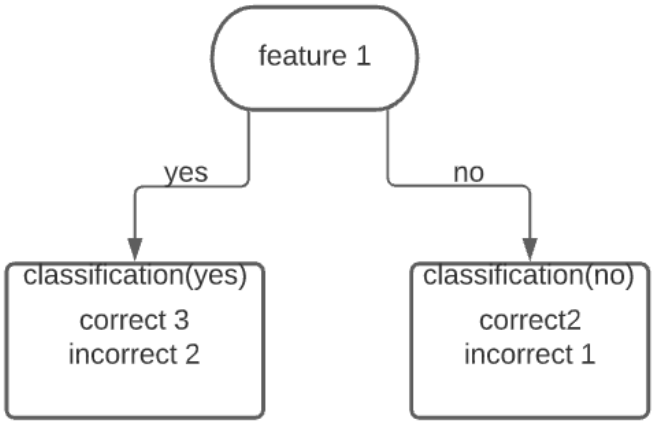


Fig 3.3

Stump formed using feature 2

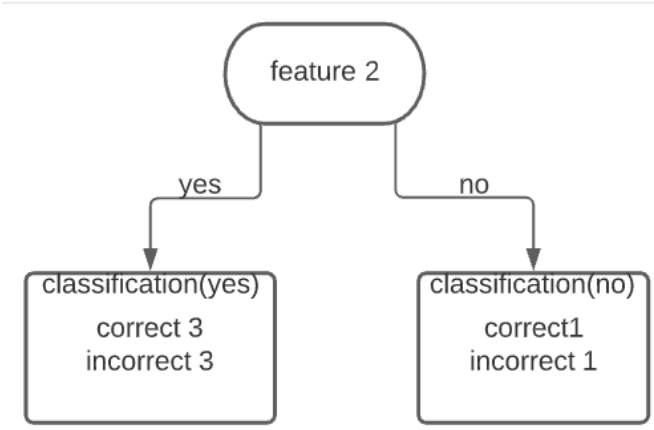


Fig 3.4

Step 3:

55) From all the stumps, select the stump which performs good

We use Gini index to find which stump performs better

Gini index for stump 1

$Gini_{stump1=}(5/8)*gini_{leftnode}+(3/8)*gini_{rightnode}$ (5/8,3/8 are total samples on that node /total no of samples on both nodes)

Gini left node= $1-(3/5)^2-(2/5)^2=0.48$

Gini right node=$1-(2/3)^2-(1/3)^2=0.45$

$Gini_{stump1} = (5/8) *(0.48) + (3/8) *(0.45) = 0.468$

Similarly for feature 2

$Gini_{stump2} = (6/8) * gini_{leftnode}+(2/8) * gini_{rightnode}$

Gini left node= $1-(3/6)^2-(3/6)^2=0.5$(worst Gini index possible)

Gini right node $=1-(1/2)^2-(1/2)^2=0.5$

$Gini_{stump2} = (0.75) *0.5 + 0.25 *0.5 =0.5$

Since $Gini_{stump1} < Gini_{stump2}$ we consider stump 1

Step 4:

56) Calculate the total error and amount of say for the selected stump

We selected stump 1 and for it we have a total of 3 incorrect classified samples

index	feature 1	feature2	classification	initial_sample_weight
1	yes	yes	yes	1.0/8
2	no	yes	yes	1.0/8
3	yes	no	yes	1.0/8
4	yes	yes	yes	1.0/8
5	no	yes	no	1.0/8
6	no	yes	no	1.0/8
7	yes	no	no	1.0/8
8	yes	yes	no	1.0/8

Fig 3.5 The highlighted samples are misclassified

Total Error =sum of weights of misclassified =1/8 + 1/8 +1/8 =3/8

Amount of say $=1/2*log\left(\frac{1-\frac{3}{8}}{\frac{3}{8}}\right)$ =0.25541

57) How do we update the weights?

the highlighted samples in the previous question indicate misclassified samples now they get some more additional weights

$$weight_{new} = weight_{previous} * e^{amount\ of\ say}$$

$Weight_{new} = 1/8 * e^{0.25541} = 0.161 \sim = 0.16$

For mis-classified samples

$$weight_{new} = weight_{previous} * e^{-amount\ of\ say}$$

$Weight_{new}$ =1/8*$e^{-0.25541}$=0.097 ~=0.1

index	feature 1	feature2	classification	initial_sample_weight	update_weights
1	yes	yes	yes	1.0/8	0.1
2	no	yes	yes	1.0/8	0.16
3	yes	no	yes	1.0/8	0.1
4	yes	yes	yes	1.0/8	0.1
5	no	yes	no	1.0/8	0.1
6	no	yes	no	1.0/8	0.1
7	yes	no	no	1.0/8	0.16
8	yes	yes	no	1.0/8	0.16

Fig 3.6 We now need to normalize the weights:

Final_updated_weights = updated_weights/sum_of_updated_weights

Here sum is 0.98, so we divide by 0.98

index	feature 1	feature2	label	initial_sample_weight	update_weights	normalized_weights
1	yes	yes	yes	1.0/8	0.1	0.102040816
2	no	yes	yes	1.0/8	0.16	0.163265306
3	yes	no	yes	1.0/8	0.1	0.102040816
4	yes	yes	yes	1.0/8	0.1	0.102040816
5	no	yes	no	1.0/8	0.1	0.102040816
6	no	yes	no	1.0/8	0.1	0.102040816
7	yes	no	no	1.0/8	0.16	0.163265306
8	yes	yes	no	1.0/8	0.16	0.163265306

Fig 3.7 The last column shows normalized updated weights

58) How do we generate new data set from the updated weights?

index	feature 1	feature2	classification	initial_sample_weight	update_weights	normalized_weights	cummulative
1	yes	yes	yes	1.0/8	0.1	0.102040816	0.102040816
2	no	yes	yes	1.0/8	0.16	0.163265306	0.265306122
3	yes	no	yes	1.0/8	0.1	0.102040816	0.367346939
4	yes	yes	yes	1.0/8	0.1	0.102040816	0.469387755
5	no	yes	no	1.0/8	0.1	0.102040816	0.571428571
6	no	yes	no	1.0/8	0.1	0.102040816	0.673469388
7	yes	no	no	1.0/8	0.16	0.163265306	0.836734694
8	yes	yes	no	1.0/8	0.16	0.163265306	1

Fig 3.8

see the last column which indicates cumulative values of previous column (normalized weights) (see the sum at the end became 1 because we made the previous column sum 1).

Now we generate random numbers and if the number is between 0-0.102 then we put sample 1 (contents of row 1) in the new data set.

Similarly, if we generate a number between 0.102-0.265 then sample 2 is put in 2nd data set.

We do this with the intuition that since high weight sample covers larger range, they have high chance of getting into our new data set than the other sample with low range.

Now after we generate new data set, we again repeat step 1(give equal weights to each sample train the stumps and so on). We stop the process based on number of stumps, we gave as parameter to the function or if it perfectly classifies the data.

Module 4:

Now let's code for the Ada Boost using SKLEARN module (coding lines are indicated by italics to differentiate it from comments below and above the coding lines).

Step 1:

 # Importing libraries
import pandas as pd
from sklearn.ensemble import AdaBoostClassifier
from sklearn import datasets

Scikit-learn metrics module for accuracy calculation
from sklearn.metrics import accuracy_score

Train_test_split function from model_selection module
from sklearn.model_selection import train_test_split

Step2:
About the dataset

Classes	**2**
Samples per class	**212(M),357(B)**
Samples total	569
Dimensionality	30
Features	**Real, positive**

Importing data set
'''lets take a default data set present in SKLEARN'''
df=datasets.load_breast_cancer()
X_input=df.data
y=df.target

Step3:
#Split the dataset 70 percent to train and 30 percent to test you can also use stratify parameter so that the outcomes types are equally distributed among training and testing

X_train, X_test, y_train, y_test = train_test_split(X_input, y, test_size=0.3)

'''70% training and 30% test'''
Step4:
Now, we enter the core area where we build the actual Ada Boost model
Here we take 50 stumps to form our model

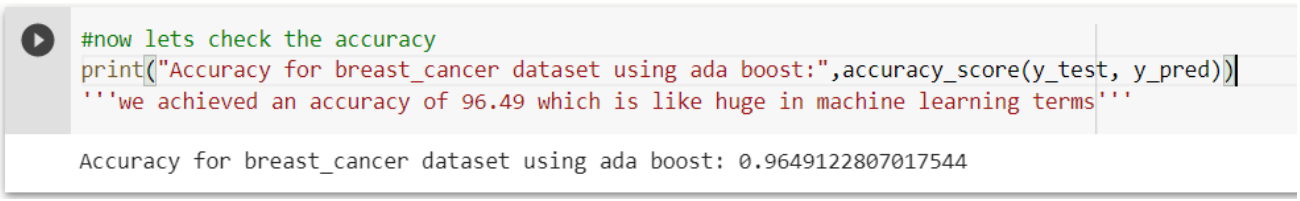
model = AdaBoostClassifier(n_estimators=50,
learning_rate=1)

Train Adaboost Classifer

model = model.fit(X_train, y_train)

#Predict the response for test dataset
y_pred = model.predict(X_test)
Step 5:
#Now let's check the accuracy.

print("Accuracy for breast_cancer dataset using ada boost:",accuracy_score(y_test, y_pred))

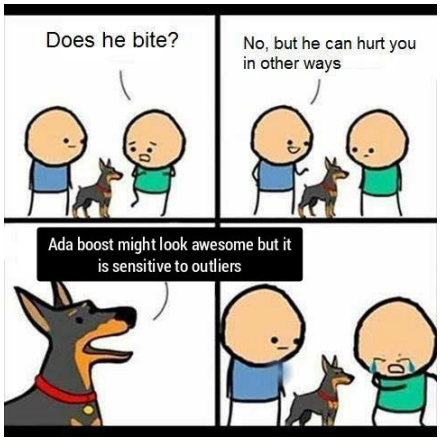


We achieved an accuracy of 96.4 %

59) What are the parameters of AdaBoost classifier in SKLEARN module?

1) Base estimator (initially set to None (decision tree classifier)
2) N_estimator (no of estimators-no of stumps to be formed)
3) Learning_rate (weights applied to each classifier at boosting iteration)
4) Algorithm= **{'SAMME', 'SAMME.R'}, default='SAMME.R'**
5)Random_state

60) Is Ada boost prone to outliers?



Outliers are bad for Ada Boost, since if outliers are misclassified for one stump, then these outlier samples repeated for the next dataset which is not an ideal thing to happen since because of this normal samples are neglected, so yes, Ada boost, which is a boosting technique is sensitive to outlier.

61) How does Ada Boost weight training reach optimal stage?

we make more and more stumps at every stage by concentrating on difficult aka mis-classified data points made by previous weak learners

62) state any disadvantage of Ada boost algorithm?

Being overfit is a common disadvantage to most of ensemble learning methods. Something that is unique to Ada boost is, from empirical evidence and particularly vulnerable or highly receptive to uniform noise.it is also not robust to outliers.