Ensembles Learning

1) In ensembles learning we look at multiple classifier & combining the output of the multiple classifier in order to get better prediction or classification accuracy.

2) In certain conditions, where the classifier outputs are independent of each other and make error in an independent manner it is possible that by combining the outputs of the multiple classifier in order to get better prediction or classification accuracy

Ensembles Classifier

1) In this we have multiple learner or classifier which give multiple output which may same or all different, or some are same & some are different.

2) Multiple learner will give different decisions & we want to combine a decision output by the multiple learner.

For this we have to generate a group

of base learner.

→ So How to generate a group of learners.

1) Use Different Algorithm
2) Different learner use different hyper parameters.
3) Or Same algorithm but different values of parameter you get different models as output will be different. They may have 4) different representation
5) Different training sets

Ex:- 1) In ANN we can seled different no of layer, weight, bias, neuron as hyper parameter
2) In decisiontree the stragtegy for deciding this attributes like entropy, gini Index. to modify the decision tree algorithm.

→ For ex 2 class - classification

Error of each learner is ϵ
if $\epsilon < 0.5$
means may be accuracy is not high but error should be $\epsilon < 0.5$.

* Different base learner satisfy certain independence conditions, then it is possible to combine weak learners, each having error ~~greater~~ less than 0.5 in order to get a very strong learner.

* Importance of Bias and Variance.

Bias:- Is a measure of how flexible the model is. So if the model is very flexible or very powerful then the bias is low

Variance:- Variance is high when if you give different subsets of data as training set, the models output are very different then we say variance is high.

* So in machine learning our objectives is reduce both bias & variance.

* So using ensembling learning we want low bias & low variance.

* Ensembles use multiple trained model

1) If we assume this models have low bias & high variance.
By combining the output, we can get low variance while maintaing low bias.
Even at intial individual hypothesis may have high bias by combining models we get classifier, which has low bias.

2) So by ensembles if we get low bias & low variance we will get less overfitting no need worry about stopping criteria.

* Combining weak learners
→ But error cannot be higher than 0.5; error must be less than 0.5.

$$\epsilon < 0.5$$

So to combinie weak learner following are conditions.

1) Assume n independent learner.
2) Eeach has accuracy 70'1. or each with error 0.3 or 0.7 accuraay.

3) If all n learners has same output
If all the learners agree on the output you are getting higher comitidence.

For ex.- Ten learners with accuracy 0.7

$\therefore$ Confidance - $1-(1-7)^{10} = 1-0.3^{10}$ close to 1.

* How to Combine the learner.

→ Combination can be done by voting.
based on majority combination or majority voting

→ Voting can be given based on weight.

1) Unweighted Voting
2) Weighted Voting.

Weight $\propto$ accuracy
Weight $\propto$ 1/variance

$y = \sum_{j=1}^{N} w_j d_j$ d_j is o/p of classifier
w_j is weight of classifier

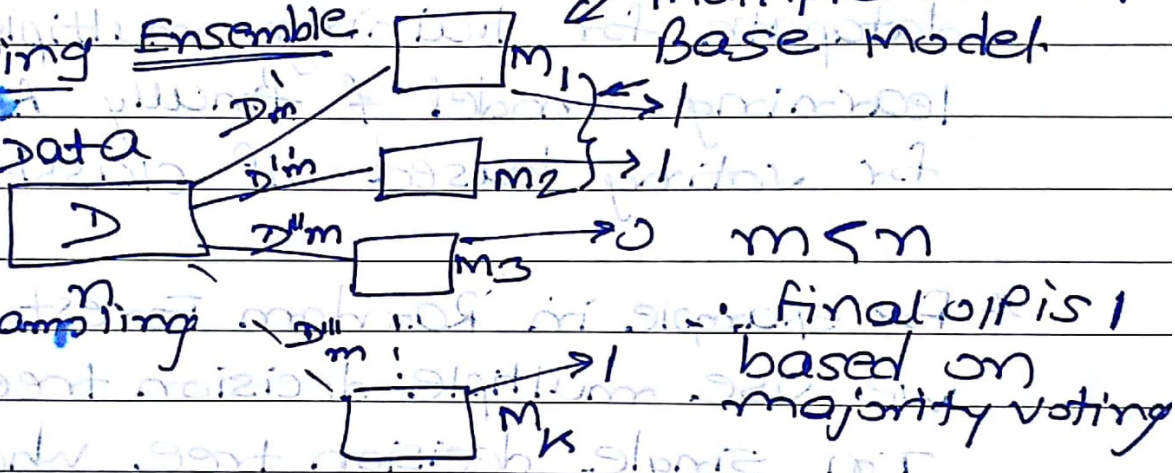
Each $w_j \geq 0$ $\sum_j w_j = 1$

ENSEMBLE TECHNIQUES

Two types of Ensemble techniques

1) Bagging - (Bootstrap Aggregation)
   1) Random Forest
2) Boosting
   1) ADABOOST
   2) Gradient Boosting
   3) XgBoost

1) Bagging Ensemble



Row Sampling with Replacement

* Ensemble means combining multiple models
* In bagging from a dataset having n sample, select D'm dataset & provide to each model M1, M2 ... MK with row sampling with replacement
* Resample data & give it other model

* So each is train with different data set
* Then using test dataset each model is tested for binary classifier & based on voting of majority final o/p is selected.

* In Bagging enother name is Bootstrap because it select multiple datapath for training multiple machine learning model & finally Aggration for voting based o/p classifier.

* For example in Random Forest Tree we use multiple decision tree. In single decision tree when we create tree with high depth we get Low Bias but High Variance

* But in Bagging Approach we use data row sampling with replacement we use data sampling & feature engineering & apply to multiple learner machine model & we take aggregation output based on

voting due to which we get low variance.

This multiple model becomes expert on different data set while getting high variance to low variance we are not depend on only one model output we take majority voting so get low variance.

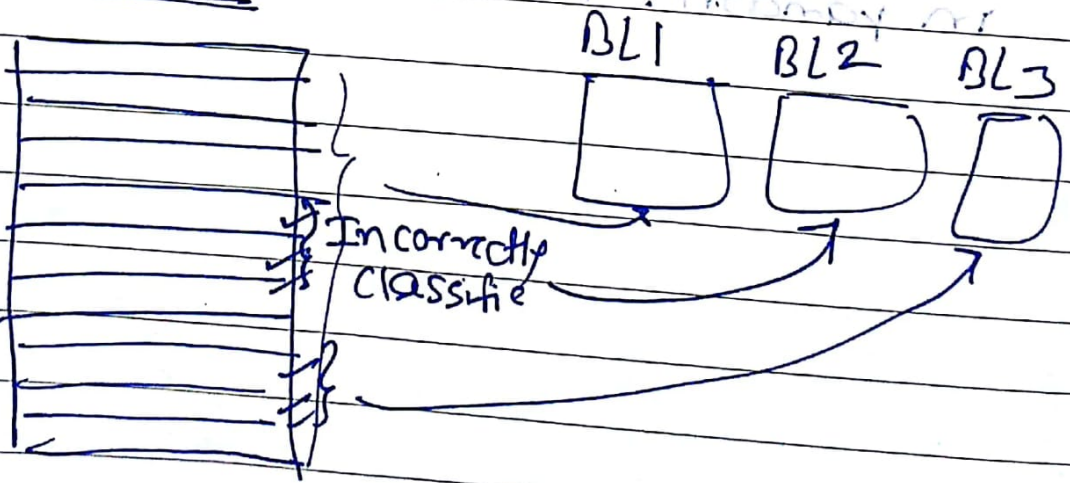
* So in classificater uses majorty vote
* In regression it uses mean or median as output.
* Hyperparameter : No of decision tree in random forest.

Boosting Techniques: (ADBoost)

* Boosting

1) Consider Data set: D
2) Base learner: BL1, BL2, BL3 ... BLN ← selected
3) From the dataset give all example to first base model & check the accuracy.
4) Those example from dataset ~~gives~~ missclassified, again give that example to another base model BL2
5) And ~~for~~ BL2 those example are missclassified give to BL3 & so on
6) It will go on unless & until we created base model with good accuracy.
7) Base model are created sequentially

* ~~ADABOOST~~



* ADABOOST

1) Consider feature f_1, f_2, f_3 O/P
2) With multiple feature training example
3) Consider all training example have same sample weight

Intial samp weight $W = 1/n$ n is no. of training example (no. of records)

4) Intially all record assign same weight for ex. n = 7

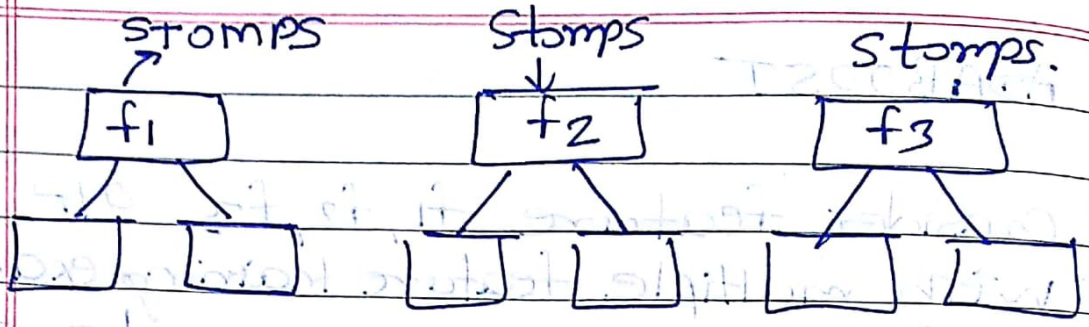
Step 1

	f_1	f_2	f_3	O/P	Sample weight
1				Yes	1/7
2				No	1/7
3				Yes	1/7
4				No	1/7
5					1/7
6					1/7
7					1/7

$\therefore W = \frac{1}{7}$

Assign the same weight to all record.

Step 2) 1) Create a first base model BL1 with decision tree
2) Create a base learner sequentially
3) Create a decision tree only with one depth
4) Consider feature f_1 & create decision tree with one depth called STOMPS

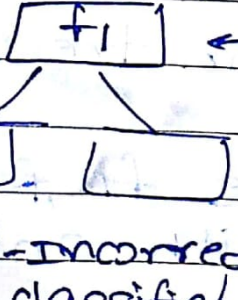


5) From this three stomps we have to select one decision tree as base learner model

6) How to select first base learner from above three stomps
→ Using Entropy or Gini Index.
Those decision tree entropy is less select that decision tree as base learning model. out of above 3 stomps one with less entropy.

Step 3) Once we select one decision tree from above stomps which have less entropy, check how many records it gives correct output

for ex →



2) Find out Total error = $\frac{\text{No of incorrectly classified O/P}}{\text{Total No. of O/P.}}$

for example in this table.

$$\text{Total error} = \frac{1}{7}$$

1 :- is incorrectly classified O/P.

7 :- total no. of output.

Step(4) Find out Performance of Stump.

$$\text{Performance of Stump} = \frac{1}{2}\log_e\left(\frac{1-TE}{TE}\right)$$

Performance of Stump means how the stump is basically classified.

$$\text{Performance of Stump} = \frac{1}{2}\log_e\left[\frac{1-\frac{1}{7}}{\frac{1}{7}}\right]$$

$$= \frac{1}{2}\log_e(6) = 0.896$$

→ Total error & performance of Stump is calculated to update the weight of data record.

So in boosting we need to pass wrong classified record to next stump.

So for the same we need to update weight in such a way that for wrong classified record increase the weight & for corrected classified record decrease the weight.

Step 3) Updation of weight

for ex: use following.

Performance of stump = 0.895 T.E = 1/7

1) Update the weight for incorrectly classified record o/p.

New Sample weight = Old weight × e^(Performance of Stump)

$= \frac{1}{7} \times e^{0.895} = 0.34$

so weight gets here increases

2) Update the weight for correctly classified record o/p

New Sample weight = Old weight × e^(−Performance of Stump)

$= \frac{1}{7} \times e^{-0.895} = 0.05$

So weight gets reduces

So using weight updation we update weight

i) for incorrectly classification weight is increase.

ii) for correctly classification record weight is decrease.

for ex. - Updated weight is 1

f_1	f_2	f_3	O/P	Old W	New W	update weight Normalized weight
			Yes	1/7	0.05	0.07
			No	1/7	0.05	0.07
			Yes	1/7	0.05	0.07
			Yes	1/7	0.05	0.07
			No ← incorrect classified	1/7	0.349	0.513
			Yes	1/7	0.05	0.017
			No	1/7	0.05	0.017
				$\Sigma=1$	$\Sigma=0.68$	$\Sigma=1$

⇒ As we take summation of old weight = 1

→ But summation of New weight is not equal to 1

∴ Normalised the weight for that.

$$= \frac{\text{New weight}}{\Sigma \text{ of New weight}} = \frac{0.05}{\Sigma=0.68} = 0.07$$

Use this Normalize weight.

Step 6 - Create New database Bucket.

by the range.

f1	f2	f3	OP	Normalised weight	Bucket
			Yes	0.07	→ 0 to 0.07
			No	0.07	→ 0.07 to 0.14
			Yes	0.07	
			Yes	0.07	→ 0.14 to 0.21
wrong classified record →			No	0.043	→ 0.21 to 0.727
			Yes	0.07	
			No	0.07	

So by dividing the weight into range we create a bucket & select the records in which we select wrong classified record.

i) Create different iteration on selected random value

ii) for ex 0.43 which is in the range of bucket 0.21 to 0.727 select the record which falls in this particular range.

iii) In this range wrong classified record is also present.

iv) Select this record, take new value of iteration 0.31 select the bucket & select record value which are in this bucket

v) In this way create a new dataset

f1 f2 f3 OP

1)
2)
3)
4)

vi) Use this new dataset for creating new decision tree stumps.

Step 4) Repeate Repeate the process again from step 2, i) Select decision tree based on less entropy value.

ii) Calculate correct & incorrect classified record

iii) find Total error & performace of stump.

iv) Update weight

v) Create bucket based on range.

vi) By iteration select record from bucket & create new feature table & use new decision stumps, find error again.

Reapeating this process again & again we get very less error which we find at start.

Less error means very less record are incorrectly classified.

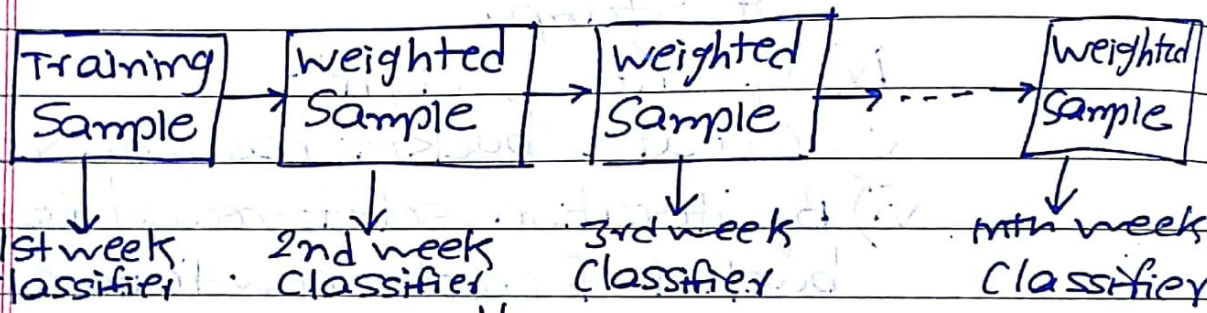
Date
Page

last
So at least we are taking voting of all stumps on test dataset.

So in boosting we are combining weak learner & creating strong learner

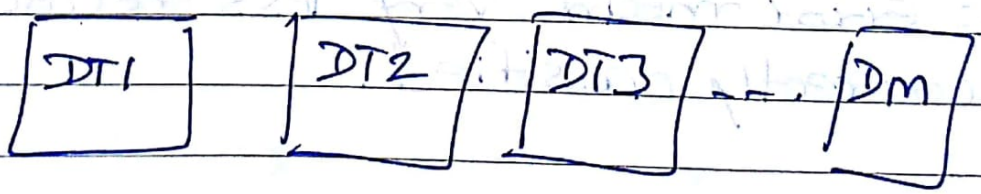
*) XGboost :—

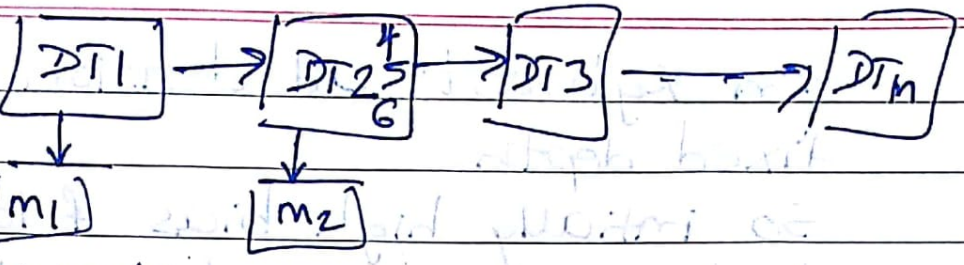
* Extream Gradient Boosting



→ In XGboost decision tree are created sequentially not parallely.

→ Sequential Ensemble Technique





Data Records	weigh			
1	w_1	m_1	m_2	
2	w_2	Apply Data		
3	w_3	Record 1 to 7		
4	w_4	Suppose Record		
5	w_5	4, 5, 6		
6	w_6	are misclassified.		
7	w_7	Re weight for the same		

1) Create a sequential decision tree.
2) Assign a weight to each record, but weight is based on probability
3) Weight decide what is the probability of the selection of record
Intiall this weight are equal, that probability of all data record are equal to select for decision tree training purpose
4) As per misclassification update weight, increase weight & apply that dataset to next Dicision tree & based on voting select final classifier output.

In xgboost all decision tree have fixed depth
So intially high bias & low variance which get converted into low bias & low variance. By combining weak learner & get final classifier with good accuracy.

Where as in decision tree, we get low bias & high variance.