LECTURE - 51

So, Now we can state Entropy as the MEASURE OF IMPURITY.

↳ A Decision Tree where the target variable takes a continuous value, usually numbers are called Regression Trees.

↳ The decision to split at each node is made according to the metric called PURITY.

↳ A node is 100% ~~im~~pure when all of its data belongs to a single class.

↳ A node is 100% impure when a node is split evenly 50-50.

∴ The Nodes should be very pure.

NOTE:

For which the Information Gain is high, According to that, we split the data into Nodes.

→ A Decision Tree is a flow-chart like structure, in which each "internal node represents a test on a attribute".

→ Each branch represents "the outcomes of the test".

→ Each leaf Node represents "a class label" (decision taken after computing all the attributes).

∴ leaf Nodes are the pure nodes.

*Decision Tree has a problem called Overfitting.

↳ Overfitting happens when the learning algorithm continues to develop hypotheses that reduce training set error at the cost of an increased test set error.

HOW TO KNOW A DECISION TREE IS OVERFITTED?

↳ The point is that we also need to check the validation accuracy beside them.

↳ If validation accuracy is falling down then we are on Overfitting zone.

OVERFITTING - HIGH VARIANCE, LOW BIASED

UNDERFITTING - LOW VARIANCE, HIGH BIASED

"VARIANCE IN DECISION TREE DOESN'T REFER

TO STATISTICAL VARIANCE".

"IN DECISION TREE VARIANCE MEANS HOW COMPLEX IS OUR MODEL".

GINI INDEX:

Gini Index or Impurity measures the degree or probability of a particular variable being wrongly classified when it is randomly chosen.

↳ A Gini Index of 0.5 denotes, equally distributed elements into some classes.

↳ Higher the Gini index indicates greater inequality and vice-versa.

→ Gini Index is represented as $I_G(Y)$.

Mathematically, the formulae of $I_G(Y)$ is

$$\Rightarrow \boxed{I_G(Y) = 1 - \sum_{i=1}^{k} [P(Y_i)]^2}$$

NOTE :

$I_G(Y)$ - the Input taken is similar to the Entropy.

↳ $I_G(Y)$ is the best one when compared to $H(Y)$ for calculation of IG.

↳ Earlier $H(Y)$ is used for computing of IG but these days $I_G(Y)$ is used since its very easy and takes less time for computation.

FOR EXAMPLE:

CASE-I:

$$\Rightarrow \text{Decision} - \begin{cases} Y(+ve) = 0 \\ Y(-ve) = 100 \end{cases}$$

$$\Rightarrow I_G(Y) = 1 - \sum_{i=1}^{k} [P(Y_i)]^2$$

$$\Rightarrow I_G(Y) = 1 - \{ P(Y(+ve)) + P(Y(-ve)) \}$$

$$\Rightarrow I_G(Y) = 1 - \left\{ \left(\frac{0}{100}\right)^2 + \left(\frac{100}{100}\right)^2 \right\}$$

$$\Rightarrow I_G(Y) = 1 - 1$$

$$\Rightarrow I_G(Y) = 0$$

CASE-II:

$$\Rightarrow \text{Decision} - \begin{cases} Y(+ve) = 50 \\ Y(-ve) = 50 \end{cases}$$

$$\Rightarrow I_G(Y) = 1 - \{ P(Y(+ve)) + P(Y(-ve)) \}$$

$$\Rightarrow I_G(Y) = 1 - \left\{ \left(\frac{50}{100}\right)^2 + \left(\frac{50}{100}\right)^2 \right\}$$

$$\Rightarrow I_G(Y) = 1 - \{ 0.25 + 0.25 \}$$

$$\Rightarrow I_G(Y) = 1 - 0.50$$

$$\Rightarrow I_G(Y) = 0.50$$

⇒ CASE - III :

⇒ Decision $\begin{cases} Y(+ve) = 100 \\ Y(-ve) = 0 \end{cases}$

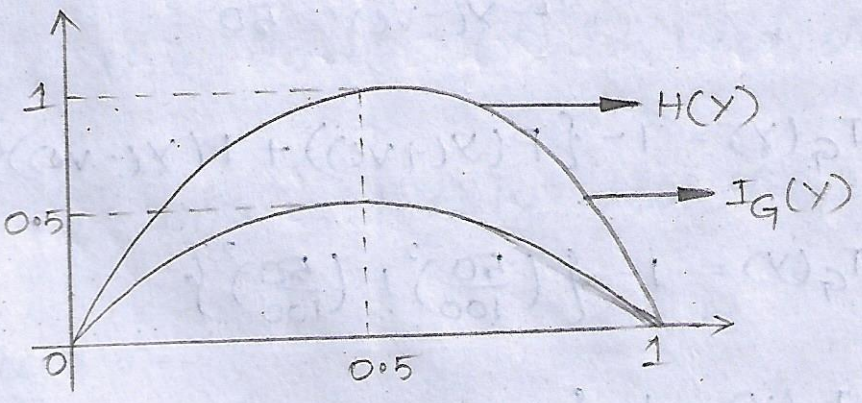
$$\Rightarrow I_G(Y) = 1 - \{P(Y(+ve)) + P(Y(-ve))\}$$

$$\Rightarrow I_G(Y) = 1 - \left\{\left(\frac{100}{100}\right)^2 + \left(\frac{0}{100}\right)^2\right\}$$

$$\Rightarrow I_G(Y) = 1 - \{1 - 0\}$$

$$\Rightarrow I_G(Y) = 0.$$

$H(Y)$	$I_G(Y)$
MAX - 1	MAX - 0.5
MIN - 0	MIN - 0



→ So, Instead of using Entropy we use Gini Index.

$$\Rightarrow \boxed{IG(Y, f_d) = I_G(Y) - \sum_{i=1}^{K} \left[\frac{|D_i|}{\|D\|} * I_G(Y) \right]}$$

* REASON: Using log function is very tricky.

↳ By using Sklearn, implementation of Gini Index is fast.

↳ the theory behind the Gini Index relies on the difference between a theoretical equality of some quantity and its actual value over the range of a related variable.

ENSEMBLE TECHNIQUE :

Ensemble methods are techniques that create multiple models and then combine them to produce improved results.

↳ Ensemble methods usually produces more accurate solutions than a single model would do.

↳ this is been the case in a number of machine learning competitions, where the winning solutions used ensemble methods.

RANDOM FOREST : COLLECTION OF DECISION TREE

Random forest is a supervised learning algorithm which is used for both classification and Regression.

↳ The Random forest algorithm creates decision trees on data samples and then gets the prediction from each of them and finally selects the best solution by means of voting.

↳ It is a flexible, easy to use which produces even without hyper-parameter tuning, a great result for most of the time.

↳ It is also one of the most used algorith-m, because of its simplicity and diversity (BECAUSE OF CLASSIFICATION & REGRESSION).

DIFFERENCE BETWEEN DECISION TREE & RANDOM FOREST :

↳ A 'Decision tree is built on the entire dataset, using all the features/variables of interest, whereas

↳ A Random forest randomly selects observations/rows and specific features/variables to build multiple decision trees from and then averages the results.

WHY RANDOM FOREST IS CALLED RANDOM?

"Forest" because there are several trees,

"random" because each tree is only trained on a random subset of samples drawn from the training set (with repetition)

and possibly a random subset of features.

NOTE:

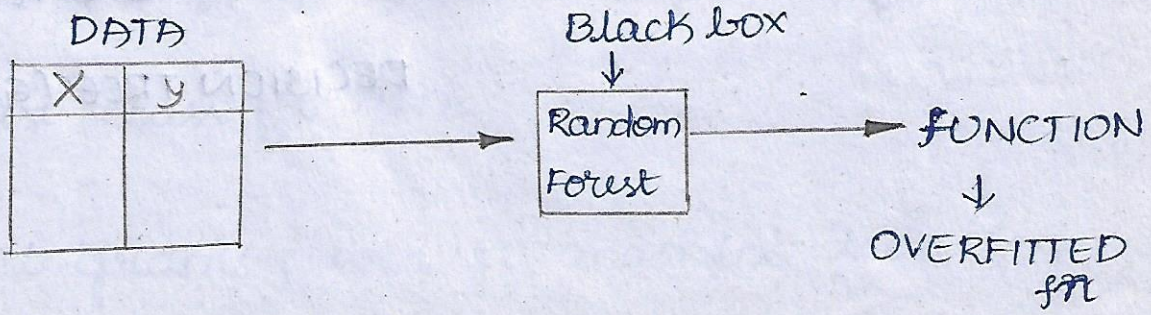
In order to overcome the overfitting we use Random Forest. i.e., helps in overfitting.

Overfitting - BY DEFAULT IT TAKES CLASSIFICATION
↳ WORKS ON REGRESSION IF MENTIONED.

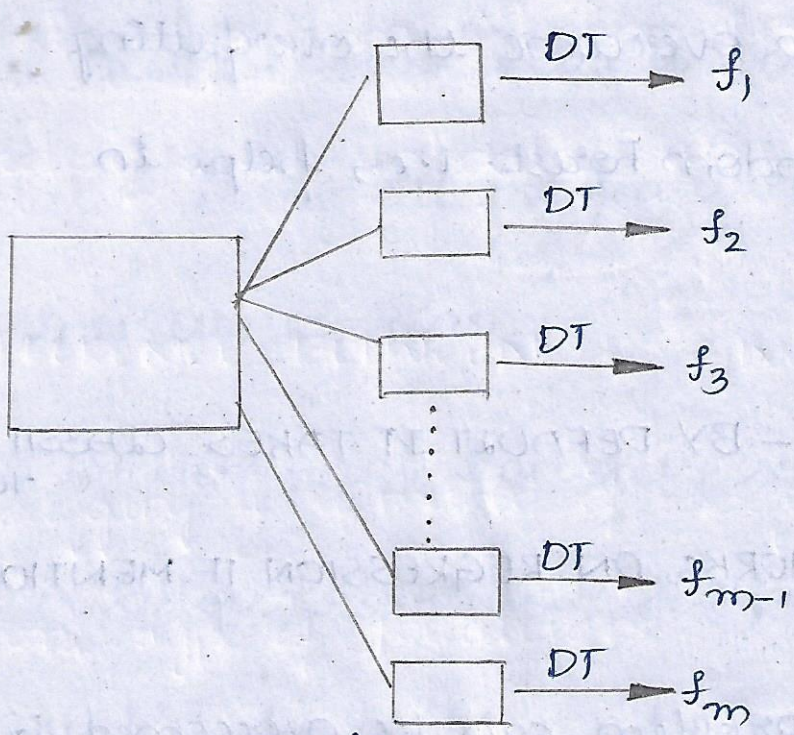
Overfitting problem can be overcomed by some other tricks i.e.,

HYPERPARAMETER → [HEIGHT/DEPTH
OVERFITTING .

↳ In production, they won't work on the overfitting.



Random Forest try to breaks the data into 'n' number of samples.



DT - DECISION TREE

↳ On each sample, we train a model of Decision Tree.

*Ensemble has many techniques.

ENSEMBLE

BAGGING

BOOSTING

RANDOM FOREST

GRADIENT BOOSTED DECISION TREE (GBDT)

BAGGING stands for "BOOTSTRAPPED AGGRIGATION".

BAGGING:

↳ way to decrease the variance in the Prediction by generating additional data for training from data set using the combinations with repetitions to produce multi-sets of the original data.

↳ It is a special case of the model averaging approach.

BOOSTING:

↳ Iterative technique which adjusts the weight of an observation based on the last classification.

↳ It grants power to machine learning

models to improve their accuracy of Prediction.

↳ This algorithm is one of the most widely used in data science competitions.

DIFFERENCE BETWEEN BAGGING & BOOSTING:

↳ Bagging is a method of merging the same type of predictions.

↳ Boosting is a method of merging different types of predictions.

→ Bagging decreases variance, not biased and solves overfitting issues in a model.

→ Boosting decreases bias, but not the variance.

RANDOM FOREST

DECISION TREE + ROW SAMPLING + COLUMN SAMPLING

ROW SAMPLING + COLUMN SAMPLING → BOOTSTRAPPING

Also called as BASE LEARNER → OVERFITTING → HIGH VARIANCE, LOW BIASED → In Decision Tree terms, DEPTH SHOULD BE AS MUCH AS POSSIBLE

BOOTSTRAPPING → SAMPLING MUST BE DONE FOR ROWS → FOR ROW → IT IS SAMPLED WITH REPLACEMENT.

* RANDOM - Randomly Bootstrapped
* FOREST - DECISION TREE as Base learner

THE STEPS INVOLVED IN RANDOM FOREST ALGORITHM:

1. Draw a random bootstrap sample of size 'n' (randomly choose n samples from the training set with replacement).

2. Grow a decision tree from the bootstrap sample. At each node:

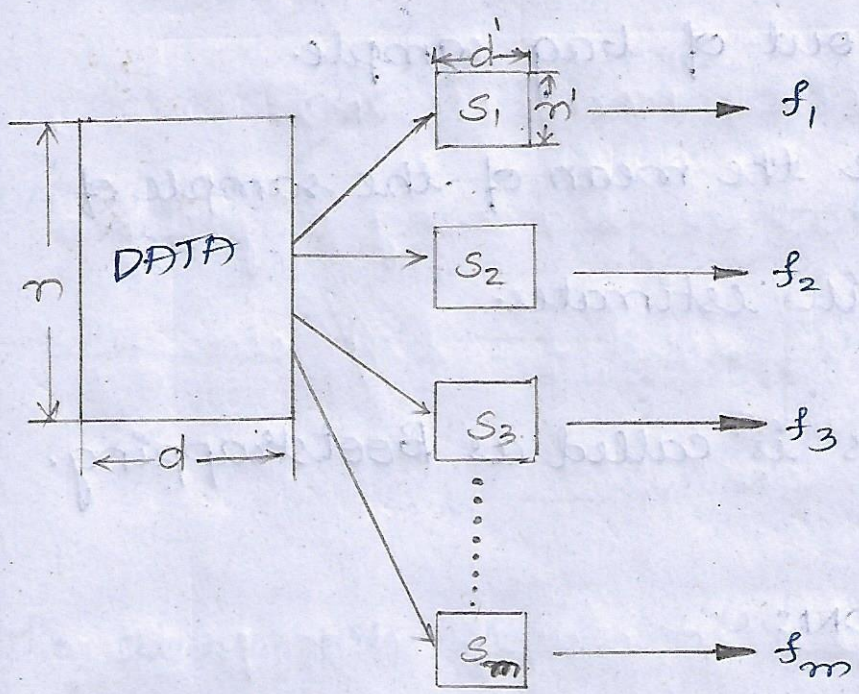
   ↳ Randomly select 'd' features without replacement.

   ↳ Split the node using the feature that provides the best split according to the objective function, for instance, by maximizing the information gain.

3. Repeat the steps 1 to 2k times.

4. Aggregate the Prediction by each tree to assign the class label by majority vote

or average.



the procedure of using bootstrap is to estimate the skill of the model and can be summarized as follows:

↳ choose a number of bootstrap samples to perform.

↳ choose a sample size.

↳ For each bootstrap sample,

* Draw a sample with replacement of choosen size
* Fit a model on the sample data.

* Estimate the skill of the model on the out-of-bag sample.

4. Calculate the mean of the sample of model skill estimates.

This process is called as Bootstrapping.

AGGREGATION:

→ It is the output of Ensemble learning.

→ Used to obtain better predictive performance that could be obtained from any constituent learning algorithm.

→ This is a method used for forming.

AGGREGATION

↓

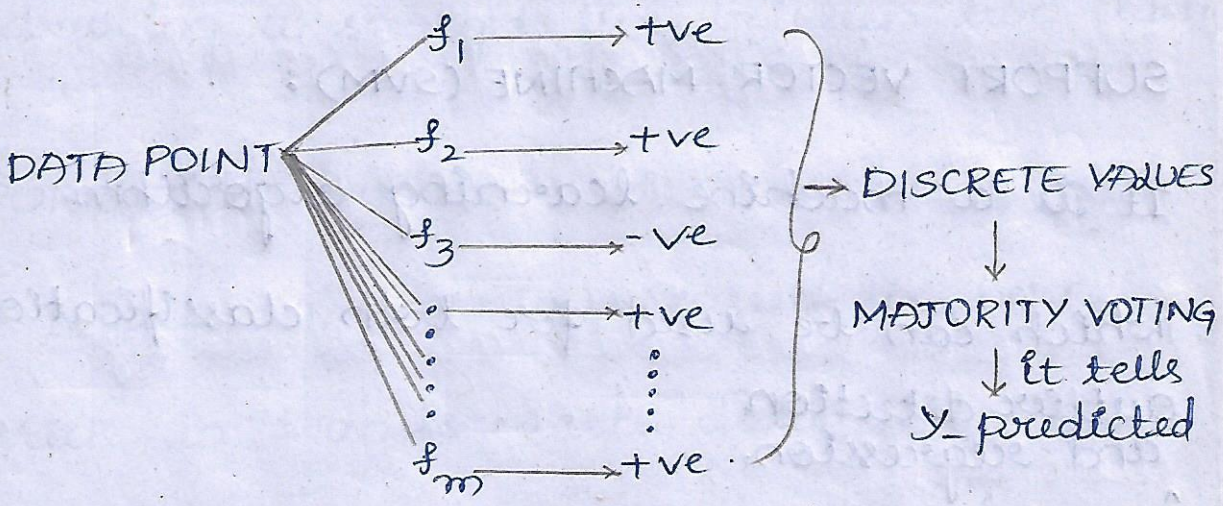
FOR FUTURE DATA POINT

↓

HOW TO CLASSIFY THE DATA POINT

i.e., 1/2/3 ....

In testing phase :



The above process is called as Aggregation.