Unit - 3

Classification

Classification is a data mining function that assign items in a collection to target categories or classes.

The goal of classification is to accurately predict the target class for each class in the data.

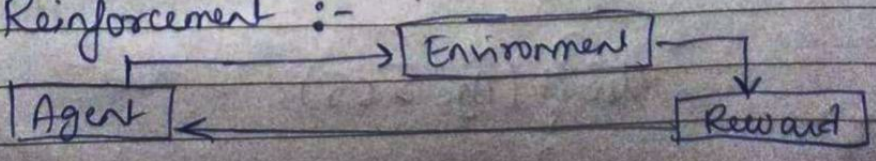
Example of cases where the data analysis task is classification :-

(a) A bank officer want to analyze the data in order to know which customer are risky or which are safe.

(b) A marketing manager at a company needs to analyze a customer with a given profile, who will buy a new computer.

ML (Machine Learning)

- Supervised Approach :- X_1 X_2 X_3 (Y) → output is known
- Unsupervised Approach :- output not known
- Reinforcement :-

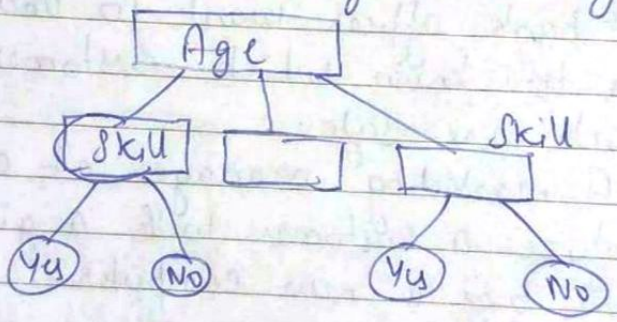


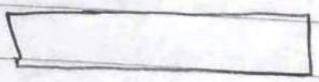
* Classification comes under supervised approach

In classification, basically what we do is to learn understand the data and then by using known output we can predict the output of unknown data.

Decision Tree Induction

Decision Tree is a technique used in classification.
It will not be always a binary tree.



for attribute → 

for result → 

Ye basically if else if pe kam kar raha.

```
If (Age < 20)
{
}
else if (Age > 60)
{
}
else { }
```

Now, whenever we choose different attributes on top decision tree will be different.

So, we need to find which attribute will be well suited for top. This process is known as Attribute Selection.

For this we measure Entropy.

$$\text{Entropy} = -\sum P_i \log_2(P_i) \text{ bits}$$

Q.

S.No	Age	Income	Class
1.	Youth	H	No
2.	Youth	H	No
3.	Middle age	H	Yes
4.	Senior	M	Yes
5.	Senior	L	Yes
6.	Senior	L	No
7.	Middle age	L	Yes
8.	Youth	M	No
9.	Youth	L	Yes
10.	Senior	M	Yes
11.	Youth	M	Yes
12.	Middle age	M	Yes
13.	Middle age	H	Yes
14	Senior	M	No

No. of classes = 2.

Step 1 → Count No. of Yes & No.

Yes = 9

No = 5

$P(Yes) = \frac{9}{14}$ $P(No) = \frac{5}{14}$

Step 2 → find information in data (Entropy)

$info(D) = -\sum P_i \log_2(P_i)$

$= -\left(\frac{9}{14}\log_2\left(\frac{9}{14}\right) + \left(\frac{5}{14} \times \log_2\left(\frac{5}{14}\right)\right)\right)$

$= -(-0.4098 + (-0.5305))$

$= 0.940$ bits

Step 3 :- find info of age :-

	Yes	No	Sum
Youth	2	3	5
Middle Age	4	0	4
Senior	3	2	5

$info(age) = -\left[\frac{5}{14} \times \left(\frac{2}{5}\log_2\left(\frac{2}{5}\right) + \frac{3}{5}\log_2\left(\frac{3}{5}\right)\right)\right.$

$+ \frac{4}{14} \times \left(\frac{4}{4} \times \log_2\left(\frac{4}{4}\right) + 0 \times \log_2 0\right)$

$\left. + \frac{5}{14} \times \left(\frac{3}{5}\log_2\left(\frac{3}{5}\right) + \frac{2}{5}\log_2\left(\frac{2}{5}\right)\right)\right]$

$= -[-0.347 + 0 + -0.347]$

$= 0.694$ bits.

Gain = info(D) − info(Age)

$= 0.940 - 0.694$

$= 0.246$ bits

Step 4:- find info of income :-

	Yes	No	Sum
H	2	2	4
M	3	1	4
L	4	2	6

$$\text{info(income)} = -\left[\frac{4}{14} \times \left(\frac{2}{4}\log_2\left(\frac{2}{4}\right) + \frac{2}{4}\log_2\left(\frac{2}{4}\right)\right) + \frac{4}{14} \times \left[\frac{3}{4}\log_2\left(\frac{3}{4}\right) + \frac{1}{4}\log_2\left(\frac{1}{4}\right)\right] + \frac{6}{14} \times \left(\frac{4}{6}\log_2\left(\frac{4}{6}\right) + \frac{2}{6}\log_2\left(\frac{2}{6}\right)\right)\right.$$

$= -[-0.2857 + -0.3936 + -0.3936]$

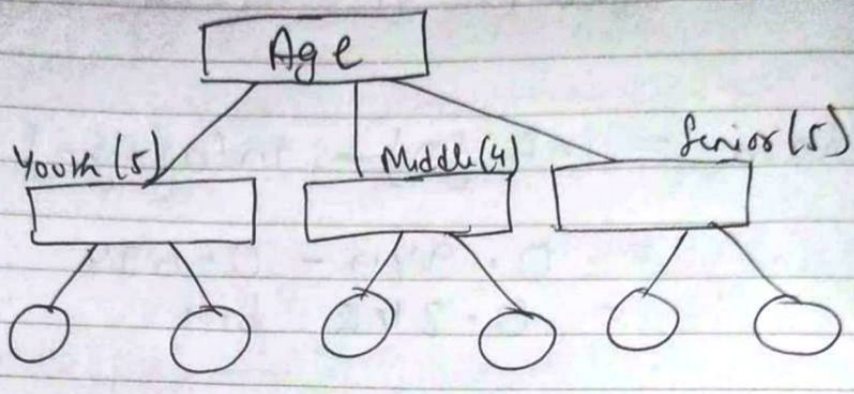
$= 0.911$ bits

Gain = info(D) − info(income)

$= 0.940 - 0.911$

$= 0.029$ bits

Gain of Age is higher than Gain of income. So it will come on top.



Drawback :- 1) Bias :- it is favouring the attributes which has more no. of partition.

Name of this Algo is ID3 :

Algorithm :-

Generate decision_tree (D, Attribute, selection method)

1) Create a Node N
2) If D contains the tuples belonging to same class.
   return N with label of class
3) If attribute list is empty then
   return N as leaf node with majority in D.
4) Apply attribute selection criteria to find the best split.
5) Attribute list = Attribute list - best selected attribute

6. For every j in the split
   If D_j is empty.
      Attach a leaf node with majority of class as labels.
   Else
      return Node Generate decision_tree(D_j)

7. Return N.

* $Split\_info_A(D) = -\sum_{i=1}^{v} \left|\frac{d_i}{D}\right| \log_2\left(\frac{D_i}{D}\right)$.

Split_info of Info(B)

$= -\left(\frac{4}{14}\log_2\frac{4}{14} + \frac{6}{14}\log_2\frac{6}{14} + \frac{4}{14}\log_2\frac{4}{14}\right)$.

$= 1.55$ bits

$\text{Gain Ratio} = \frac{Info(D)}{SplitInfo_A(D)}$

$= \frac{0.029}{1.55} = 0.018$

* Select that whose Gain ratio is high.

Gini Index :-

Gini Index measures ~~that~~ the impurity of data.

$$Gini(D) = 1 - \sum_{i=1}^{n} (P_i^2)$$

$$P(Yes) = \frac{9}{14}$$

$$P(No) = \frac{5}{14}$$

$$Gini(D) = 1 - \left[\left(\frac{9}{14}\right)^2 + \left(\frac{5}{14}\right)^2\right]$$

$$= 0.46$$

$$= 46\% \text{ impurity.}$$

$$Gini_A(D) = \sum_{i=1}^{v} \frac{D_i}{D} \times Gini(D).$$

* Gini is applied when split is Binary.

So, these 3 were the 3 techniques for designing Decision tree.

* Now the Question arises till what height we should do it.

We just need to draw till a certain level. We will not draw after that level.

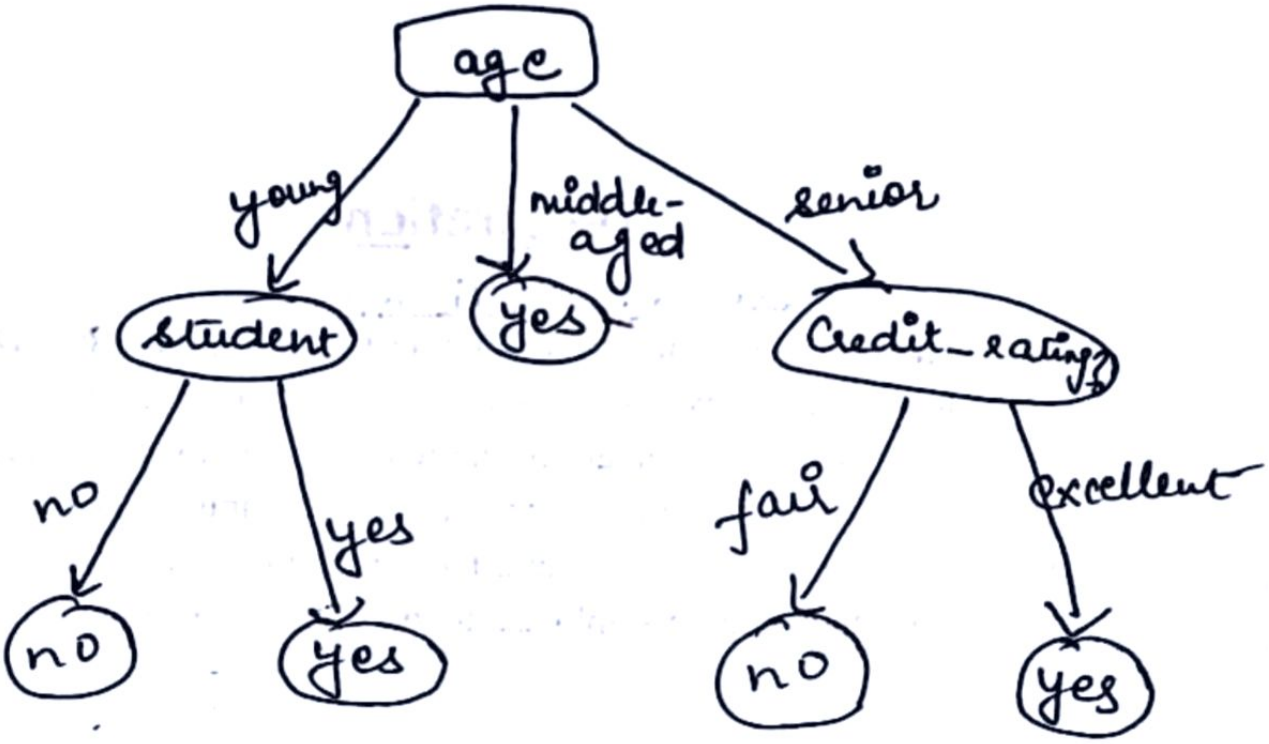
2 Ways to decide height

(i) Pre pruning
(ii) Post pruning.

Decision Tree induction

It is a method of learning the decision trees from the training sets. The dataset is broken down into smaller subsets and is present in the form of nodes of a trees. The tree structure has a root node, internal nodes or decision nodes, leaf nodes and branches. The root node is the topmost node.

→ Each internal node denotes a test of an attribute, each branch ~~nodes~~ denotes the outcome of a test, and each leaf node holds a class label. The topmost node in the tree is the root node.



Decision tree induction algorithm

developed a decision tree algorithm known as ID3 (Iterative Dichotomiser) later, he presented C4.5, which was the successor of ID3. ID3 and C4.5 adopt a greedy approach. In this algo, there is no backtracking; the trees are constructed in a top-down recursive divide and conquer manner.

Tree Pruning

Tree Pruning is performed in order to remove anomalies in the training data due to noise or outliers. The pruned trees are smaller and less complex.

Tree-Pruning Approaches

There are two approaches to prune a tree -

Pre-pruning → The tree is pruned by halting its construction early.

Post-pruning → This approach removes a sub-tree from a fully grown tree.

Key factors

Entropy :-

It refers to a common way to measure impurity. In the decision trees, it measures the randomness or impurity in data sets.

Information Gain :-

Information gain refers to the decline in entropy after the dataset is split. It is also called entropy reduction. Building a decision tree is all about discovering attributes that return the highest data gain.

Naive - Bayes Theorem

$$P(A|B) = \frac{P(B|A)\ P(A)}{P(B)}$$

Day	Outlook	Temp	Humidity	Wind	Play Cricket?
1	Sunny	Hot	High	Weak	No
2	Sunny	Hot	High	Strong	No
3	Overcast	Hot	High	Weak	Yes
4	Rain	mild	High	Weak	Yes
5	Rain	Cool	Normal	Weak	Yes
6	Rain	Cool	Normal	Strong	No
7	Overcast	Cool	Normal	Strong	Yes
8	Sunny	mild	High	Weak	No
9	Sunny	Cool	Normal	Weak	Yes
10	Rain	mild	Normal	Weak	Yes
11	Sunny	mild	Normal	Strong	Yes
12	Overcast	mild	High	Strong	Yes
13	Overcast	Hot	Normal	Weak	Yes
14	Rain	mild	High	Strong	No

Q→ Find the probability to play cricket on 15th day where conditions are —

Temp = cool Humidity = High

Wind = strong Outlook = Sunny

Step-1

Total(14)

Yes(9) NO(5)

$P(Yes) = \frac{9}{14}$ $P(NO) = \frac{5}{14}$ expected % for / Total no. of outcome

Step-2

Wind(14)

Weak(8) Strong(6)

Yes(6) NO(2) Yes(3) NO(3)

$P(Weak|Yes) = \frac{6}{9}$ $P(Strong|Yes) = \frac{3}{9}$

$P(Weak|NO) = \frac{2}{5}$ $P(Strong|NO) = \frac{3}{5}$

Step-3

Humidity(14)

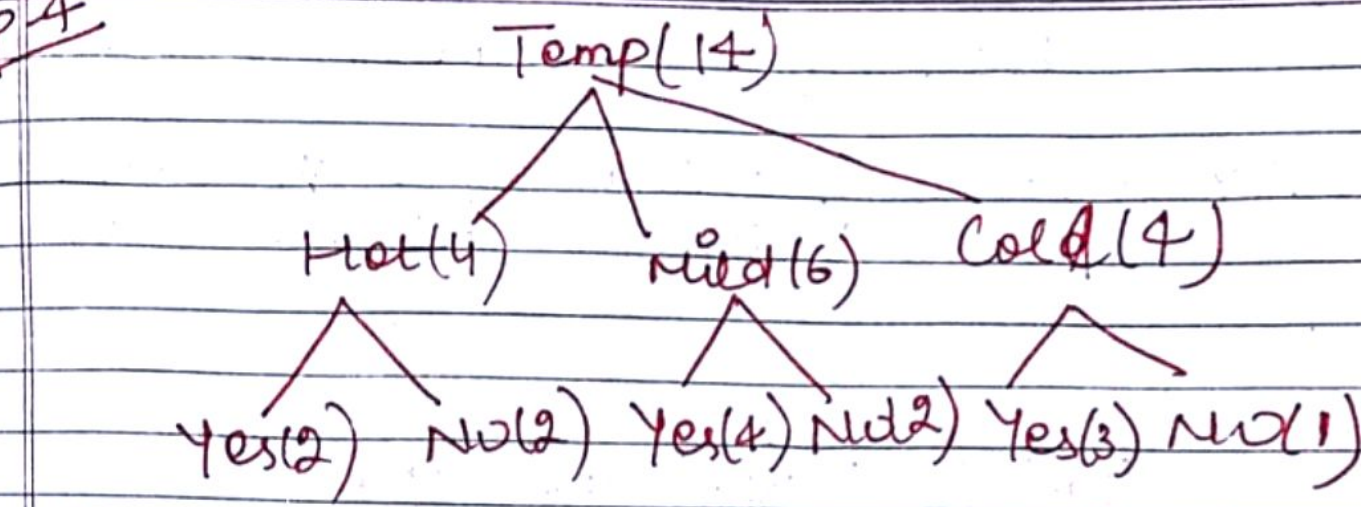
High(7) Normal(7)

Yes(3) NO(4) Yes(6) No(1)

$P(High|Yes) = \frac{3}{9}$ $P(Normal|Yes) = \frac{6}{9}$

$P(High|NO) = 4/5$ $P(Normal|NO) = 1/5$

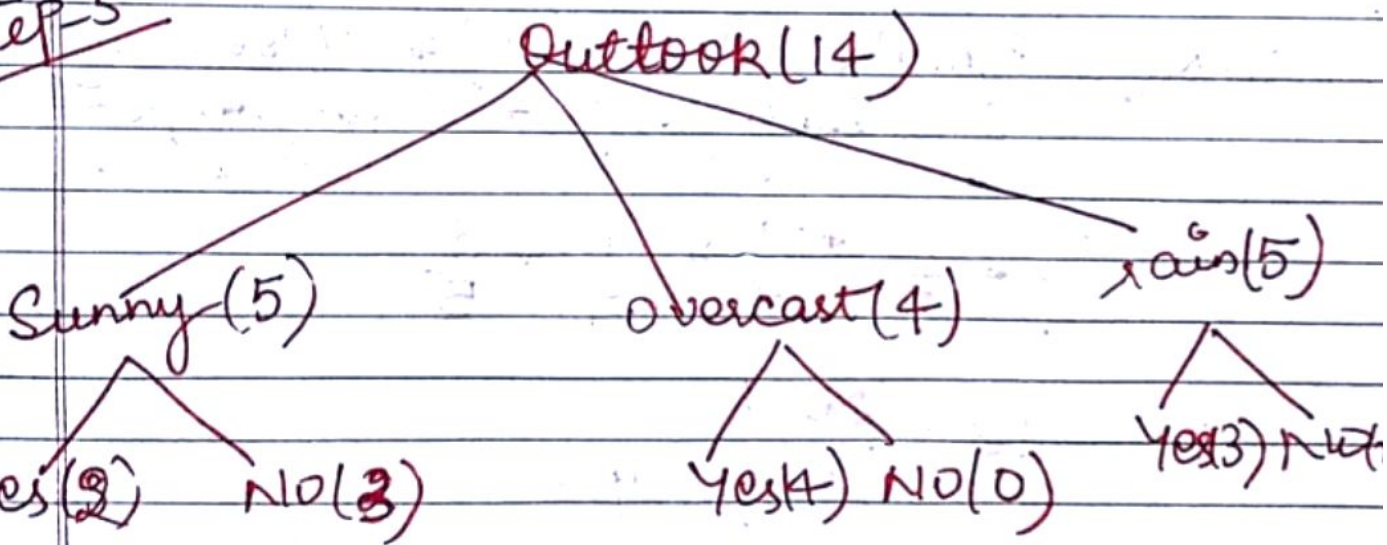
Step 4



$P(Hot/Yes) = \frac{2}{9}$ $P(Mild/Yes) = \frac{4}{9}$ $P(Cool/Yes) = \frac{3}{9}$

$P(Hot/No) = \frac{2}{5}$ $P(Mild/No) = \frac{2}{5}$ $P(Cool/No) = \frac{1}{5}$

Step-5



$P(Sunny/Yes) = \frac{2}{9}$ $P(Overcast/Yes) = \frac{4}{9}$ $P(rain/Yes) = \frac{3}{9}$

$P(Sunny/No) = \frac{3}{5}$ $P(Overcast/No) = \frac{0}{5}$ $P(rain/No) = \frac{2}{5}$

Let X = { Sunny, Cool, High, Strong }

$P(X/Yes) = P(Yes) * P(Sunny/Yes) * P(Cool/Yes) * P(High/Yes) * P(Strong/Yes)$

$= \frac{9}{14} * \frac{2}{9} * \frac{3}{9} * \frac{3}{9} * \frac{3}{9}$

$= 0.00529$

$P(X/No) = P(No) * P(Sunny/No) * P(Cool/No) * P(High/No) * P(Strong/No)$

$= \frac{5}{14} * \frac{3}{5} * \frac{1}{5} * \frac{4}{5} * \frac{3}{5}$

$= \frac{18}{7 \times 125} = 0.02057$

Probablity of No is High

So answer should be "No".

RULE BASED CLASSIFICATION:

Rule based classifier uses a set of IF THEN rules for classification

If {condition} Then {condition}

- If {condition} → Rule antecedent
- Then {condition} → Rule consequent

For eg-

Rid	Age	Income	Student	Comp.
1	youth	high	NO	no
2	youth	high	NO	no
3	medium	high	NO	yes
4	old	medium	NO	yes
5	old	low	yes	yes
6	old	low	yes	NO
7	medium	low	yes	yes
8	youth	medium	NO	NO
9	youth	low	yes	yes

10	old	medium	Yes	Yes
11	youth	medium	Yes	Yes
12	medium	medium	No	Yes
13	medium	high	Yes	Yes
14	old	medium	NO	NO

R1: If age = 'youth' & student = 'yes' then buy-computer = yes

R2: If age = 'old' & income > 25000 & student = No then buy comp = No

Rule has —

i) Coverage → fraction of records that satisfy antecedant of a rule.

ii) Accuracy → fraction of records that satisfy both antecedant & consequent of a rule.

for ex — R_1: If age = youth & student = 'yes' then buy-computer = yes.

∴ coverage = 2/14 = 14.28%
accuracy = 2/2 = 100%.

Characteristics of a Rule based classifier

If we convert the result to classification rules, then rules would be mutually exclusive & exhaustive at same time.

a) Mutually exclusive rules ($R_1 \cap R_2 = \emptyset$)

Classifier contains mutually exclusive rules. If then rules are independent of each other.

— Every record is covered by atmost one rule.

For eg — R_1 — Age & Income
R_2 — Kid & student

b) exhaustive — Classifier has exhaustive coverage if it accounts for every possible combination of attribute.

— Each record is covered by atleast one rule.

eg (id, student) (id, income) (age, income) (id, age, income)

But this is not possible since a record may trigger more than one rule or record may not trigger any rule.

- If more than one rule is triggered, need conflict resolution.

i) Size ordering → Assign the highest priority to the triggering rules that has the toughest requirement i.e. with most attribute test.

For eg -

R_1 – 3 Attribute
R_2 – 5 Attribute
R_3 – 11 Attribute → Chosen.

ii) Class based ordering – Rules that belong to same class.

$C_1 >= 25$
$C_2 <= 25 - 30$
$C_3 > 50$

iii) Rule-based ordering – rules are organised in one long priority list according to some measure of rule quality ie. by coverage, accuracy.

eg R_1: (give_birth = No) ∧ (can fly = yes) → Bird

R_2: (give birth = No) ∧ (live in water = yes) → Fish

R_3 (give-birth yes) ∧ (Blood type = warm) → Mammals

R_4: (give_birth = No) ∧ (can fly = No) → Reptiles

R_5: (live in water = sometimes) → Amphians

eg

Name	BT	GB	can fly	live in water	Cla
Hawk	warm	no	yes	No	?
Bear	warm	yes	No	No	?

R_1 covers hawk class bird.

R_3 covers bear class mammals.

eg →

Name	BT	GB	can fly	live in water	Cl
Turtle	cold	No	~~sometimes~~ No	Sometimes	?
Shark	cold	yes	No yes	yes	?

Turtle triggers both R_4 & R_5

↳ ordering rule or voting

default shark triggers none of rules

↳ default rules

Approaches for rule based classifier

i) Direct method → extract rules direct from data.
eg- sequential covering algorithm.

ii) Indirect method → extract rules from other classifier.
eg → decision tree.

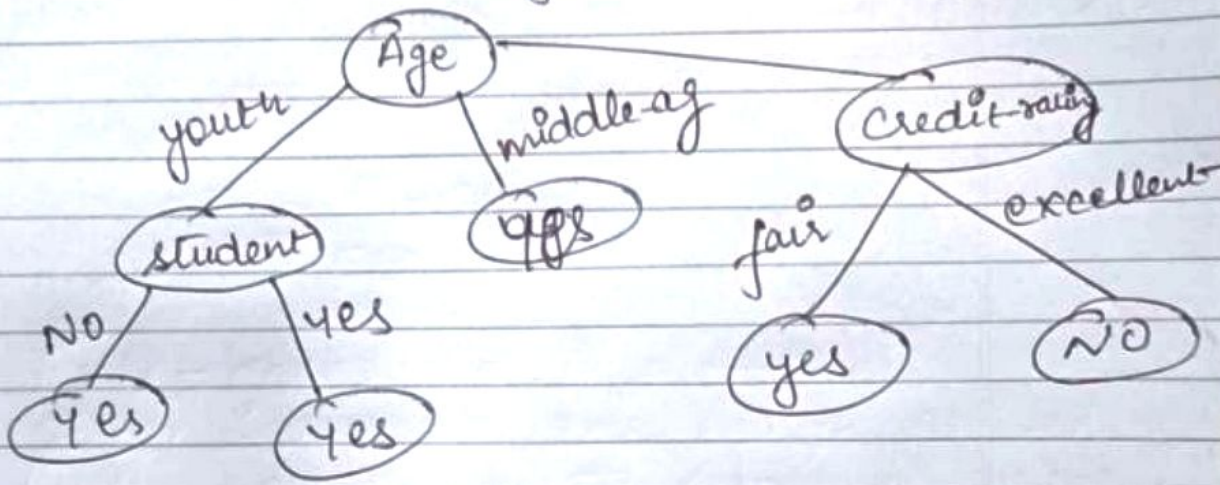
R1: If Age = youth AND student = no Then buys_computer = NO

R2: If Age = youth AND student = yes Then buy_computer = yes

R3: If Age = middle Then buy_computer = yes

R4: If Age = senior AND credit_rating = fair Then buy_computer = yes

R5: If Age = senior AND credit_rating = excellent Then buy_computer = NO



Methods for estimating a classifier's accuracy —

i) Hold out method, random subsampling
ii) Cross-validation
iii) Bootstrap

Cross-validation

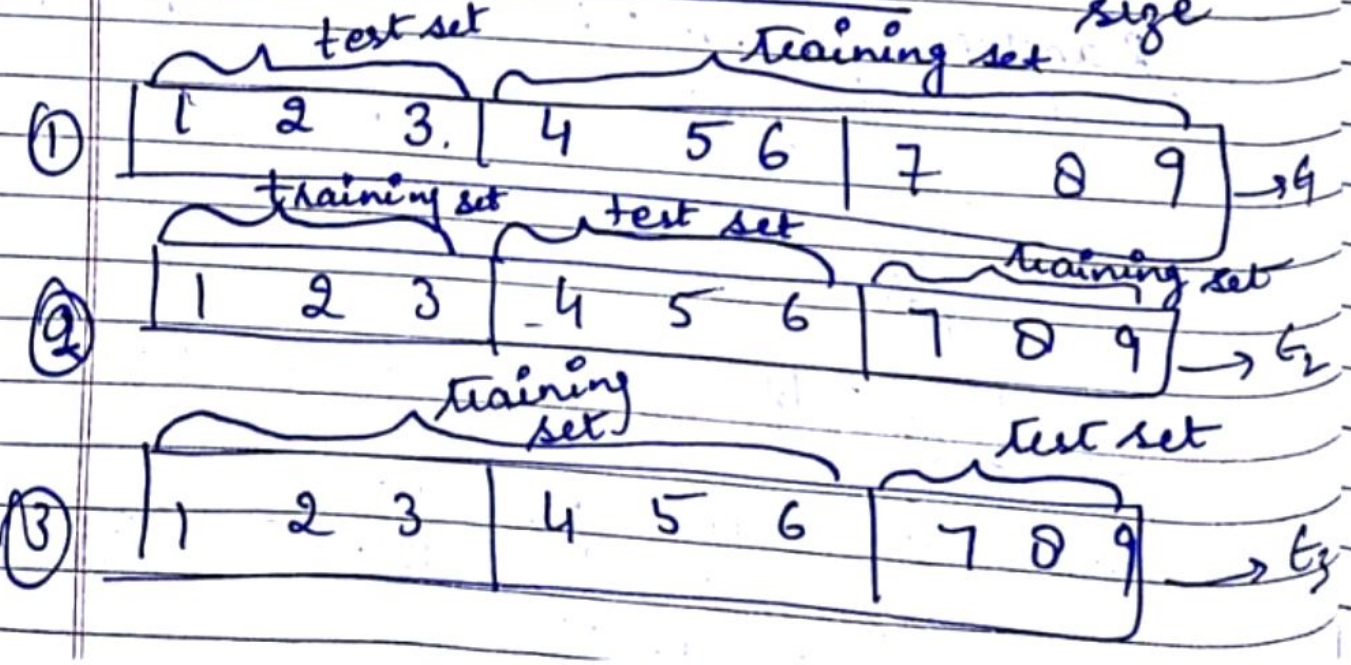
systematic sampling

Not random sampling

→ It overcome the problem of "overlapping of test set".

→ divide the whole data into 3 subsets of equal size

3-Fold cross-validation



$$E = \frac{1}{K} \sum_{i=1}^{K} E_i$$

In K-fold cross validation, the initial data are randomly partitioned into K mutually exclusive subsets or folds " $D_1, D_2, D_3 - - D_K$, each of approximately equal size. Training set and testing is performed K times. In iteration i, partition D_i is reserved as the test set and the remaining partitions are collectively used to train the model

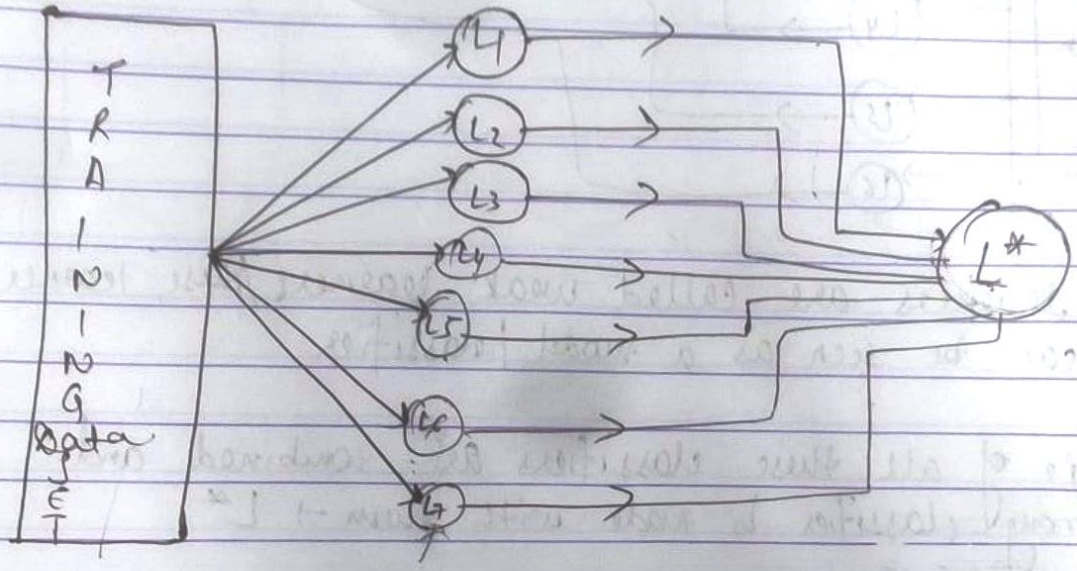
→ In the first iteration, subsets $D_2, D_3 - - D_K$ collectively as the training set to obtain a first model, which is tested on D_1

→ Second iteration is trained on subsets $D_1, D_3 - - D_K$ and tested on D_2 and so on.

⟹ Leave One-out is a special case of K-fold cross validation where K is the set to the no. of initial tuples. That is, only one sample is " left out " at a time for the test set.

Ensemble learning: Don't take decisions based on only one single model.

→ Consider the output of different other models to reach to a decision / conclusion.



→ L1, L2, L3, L4 - - - are the base learners.

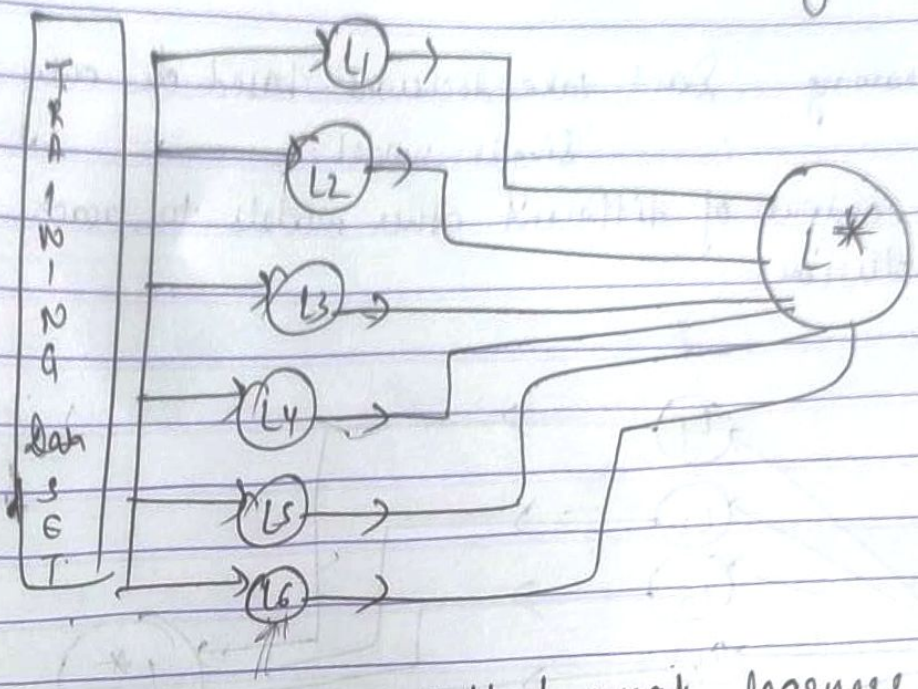
→ These learners can use different algorithms to maintain the diversity of their output.

→ This situation is called Heterogeneous situation.

→ If all the learners use the same algorithm, then the diversity of output will reduce.

→ To bring diversity in this case, we use different training data set

When learners use different Training data sets,



→ These learners are called weak learners / Base learner.
+ This can be seen as a model / classifier.

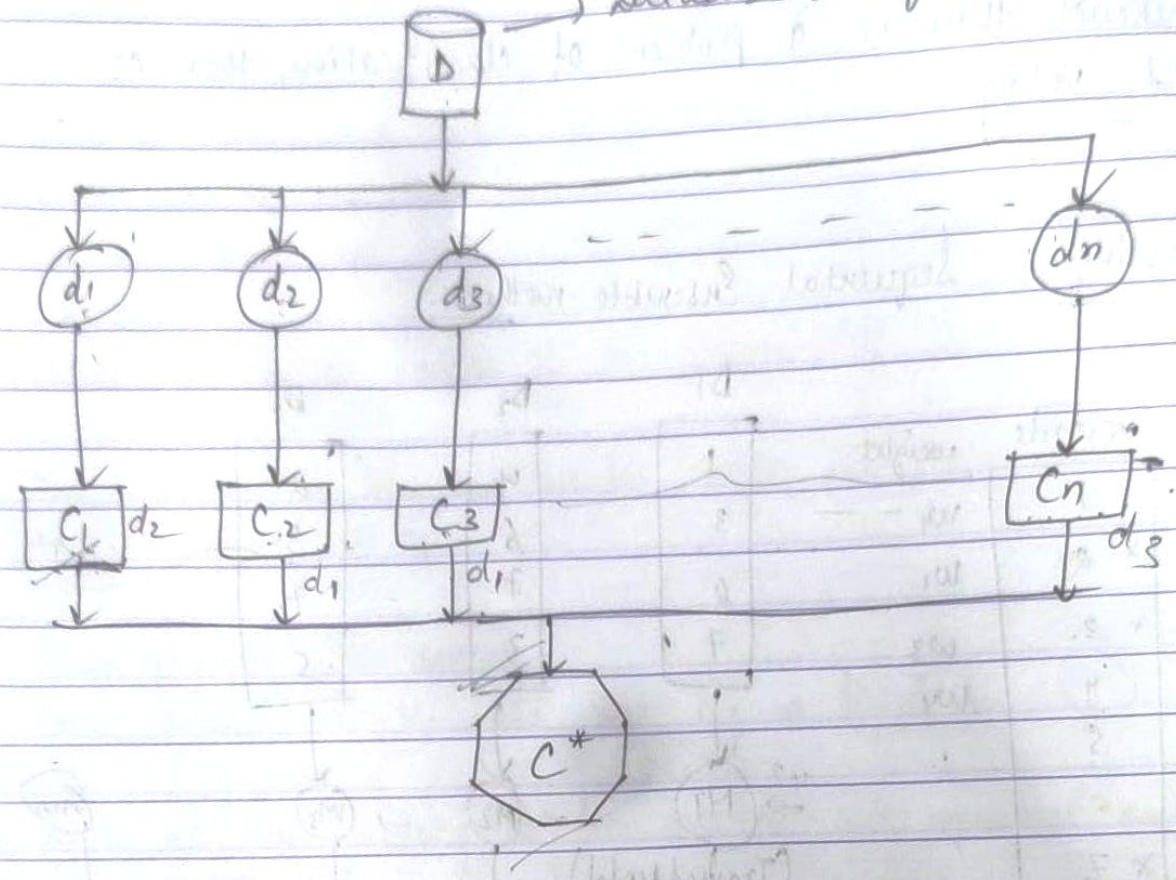
+ The o/p of all these classifiers are combined and a strong classifier is made with them → L^*.

→ Strong because the predictibility of L^* as compared to other ($L_1, L_2, \ldots$) is very high.

→ Three types of Ensemble Methods:

1. Bagging
2. Boosting
3. Random Forest

→ Bagging: Also known as Bootstrap Aggregation.



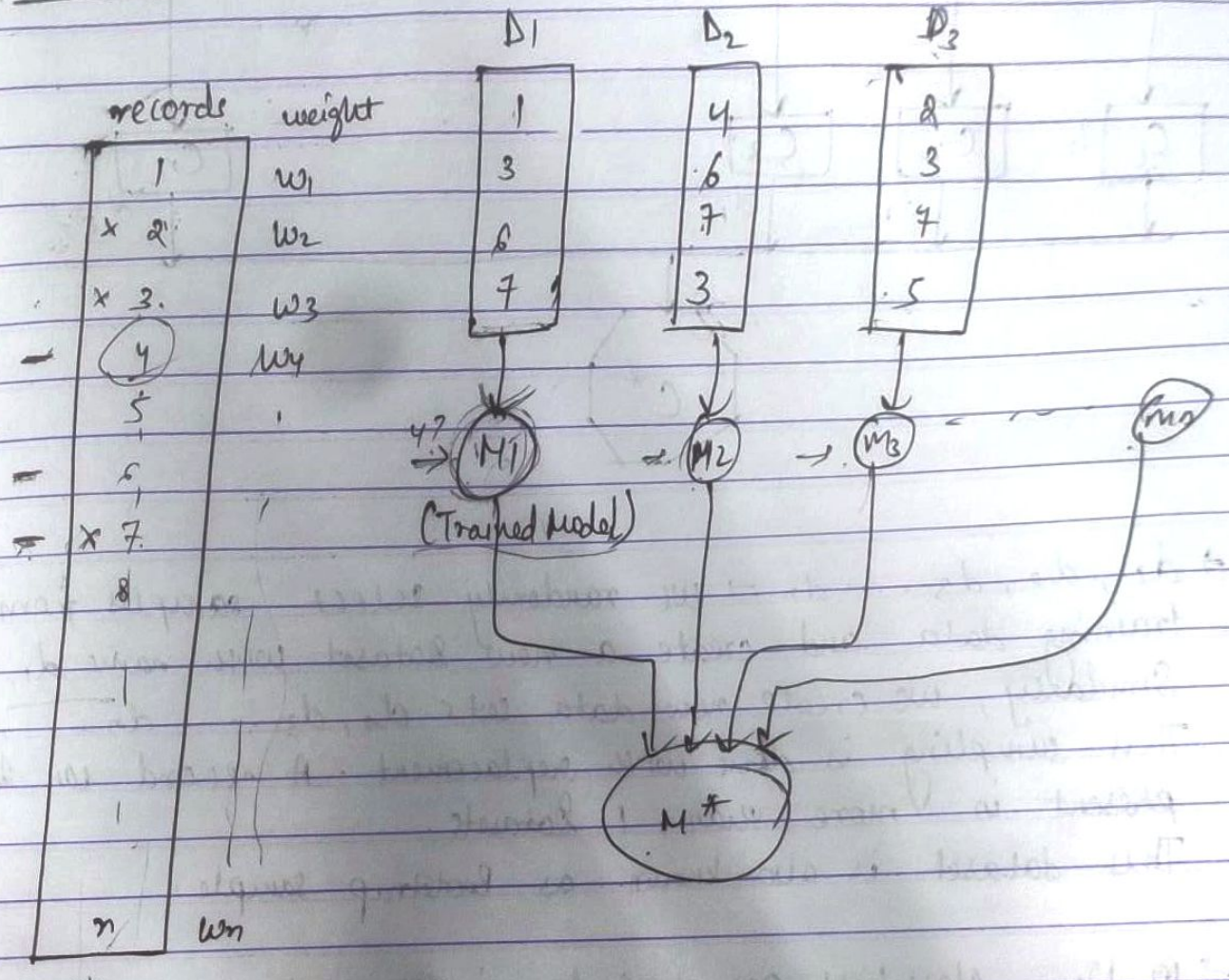
→ $d_1, d_2, d_3 \ldots d_n$ → we randomly select samples from training data and create a new Dataset with name d_1. Similarly, we create new data sets $d_2, d_3 \ldots d_n$.
This sampling is done with replacement. A record can be present in more than 1 Datasets.
This dataset is also known as Bootstrap sample.

→ Now, classifiers are trained using these data sets.
C_1 is trained from d_1, C_2 is trained from D_2, ... etc.

Now, these classifiers are combined and a final ensembled classifier C^* is formed. (with low error rate & more accurate result)

If we need to find to which class a data belongs, all the classifiers are asked. The result is the majority votes that have come.
Whenever there is a problem of classification, then we used voting.

Boosting : Sequential Ensemble method.



We assume that all the weights are equal.
↓
When we are creating a new dataset D_1, then all the records are equally probable to be a part of D_1.

M_1 will tell to which class the records belong. All the records will be checked one by one.

It is not necessary that M_1 tells the correct data, to overcome this, 2nd model is trained.

Suppose w_4, w_6, w_7 misclassified.

To move from one model to another, the weights need to be updated.

Focus more on the weights of the records that M_1 misclassified.

The records with misclassified weight now have more probability to get selected into D_2.

Same process will repeat

Same repeats with M_2 and then D_3 is filled.

Once all the models are formed, they are combined and a strong model M^* is formed.

In case of classification, voting is preferred.

Random forest : uses Decision trees.

It is a kind of Ensembled classifier that uses decision tree Algorithm

Initially, we must have original training data set.

	A	B	C	T
1.				Y
2.				N
3.				Y
4.				Y
5.				N

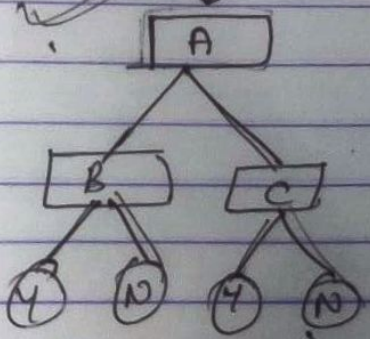
Bootstrap Data Set

	A	B	C	T
2.				N
2.1.				Y
3 1.				Y
3.3.				Y
4.				Y

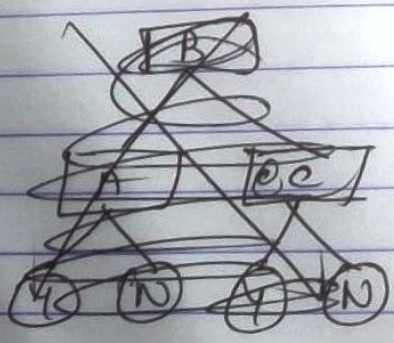
→ Create a Bootstrap Data Set using sampling.

Plot a decision Tree using this Bootstrap Data Set

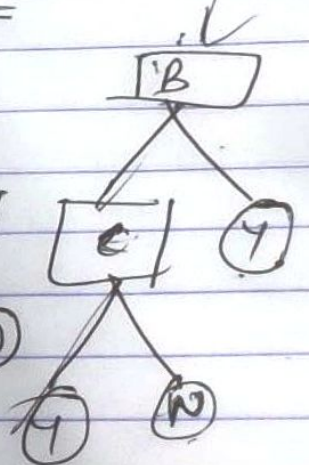
To decide the root node, out of 3 variables A, B, C we choose a subset of (A, B) and then assume that A is better for splitting



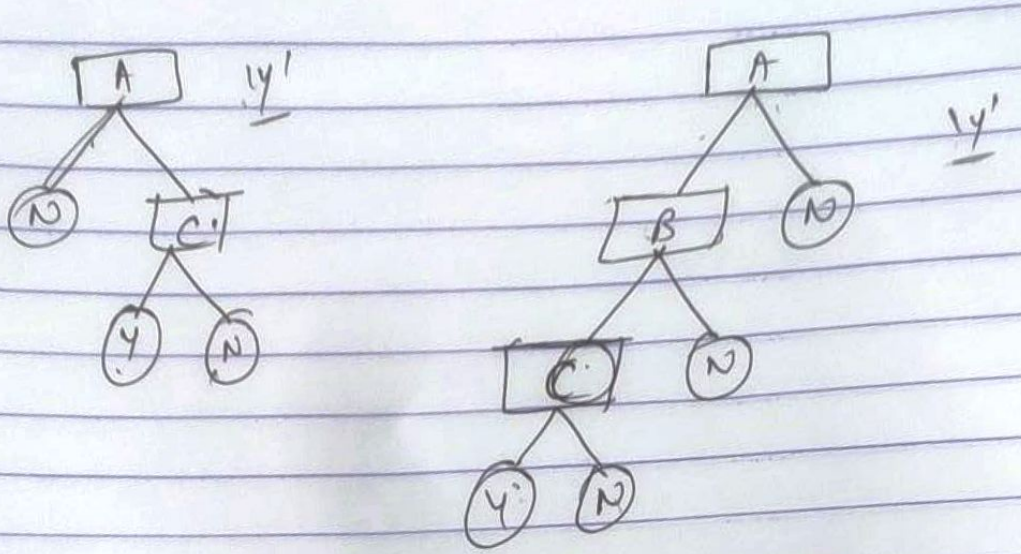
Same steps for (B, C)



B, C



Randomly generated Decision Trees.



Test Tuple → T

A	B	C	?

Find whether the target attribute will be Yes or No?

We apply this tuple to every decision tree.
We go with the maxm votes. → Yes
→ No

Bootstrap Approach → The training records are sampled with replacement

→ ensures accuracy.

OD

Rec	X	Y	Class
10	0	1	+
14	+	0	+
12	1	1	−

Sample

α_1

Rec	X	Y	Class
12	1	1	−
12	1	1	−
10	0	1	+

similarly → α_2, α_3

if original size is n, then sample will contain 63.21 % of N

Bootstrapping Method - How does it work?

Invented by Bradley Efron, the bootstrapping method is known to generate new samples or resamples out of the already existing samples in order to measure the accuracy of a sample statistic.

Using the replacement technique, the method creates new hypothetical samples that help in the testing of an estimated value.

Here are 3 quick steps that are involved in the Bootstrapping method -

1. Randomly choose a sample size.
2. Pick an observation from the training dataset in random order.
3. Combine this observation with the sample chosen earlier.

For the samples that are chosen in the representative sample size, they are referred to as the 'Bootstrapped samples' or the bootstrap sample size. On the other hand, the samples that are not chosen are referred to as the 'Out-of-the-bag' samples that serve as the testing dataset.