05-Nov-2022

Decision Tree

Machine learning

1. Supervised (Regression & classification)
2. Unsupervised (Clustering)

1. Supervised

Independent and dependent features

- 1. Regression Problem
- 2. classification Problem

Regression

1. Linear Regression (Linear approach)
2. Lasso
3. Ridge
4. Elastic Net
5. SVR

classification

1. Logistic Regression (Probabilistic approach)
2. SVM → SVC (linear Algebra)
   SVM Kernel

Decision Tree (Supervised ML algorithm)

- conditions based approach.
- Standalone (Single model)
- multiple model (ensemble technique) → Decision Tree

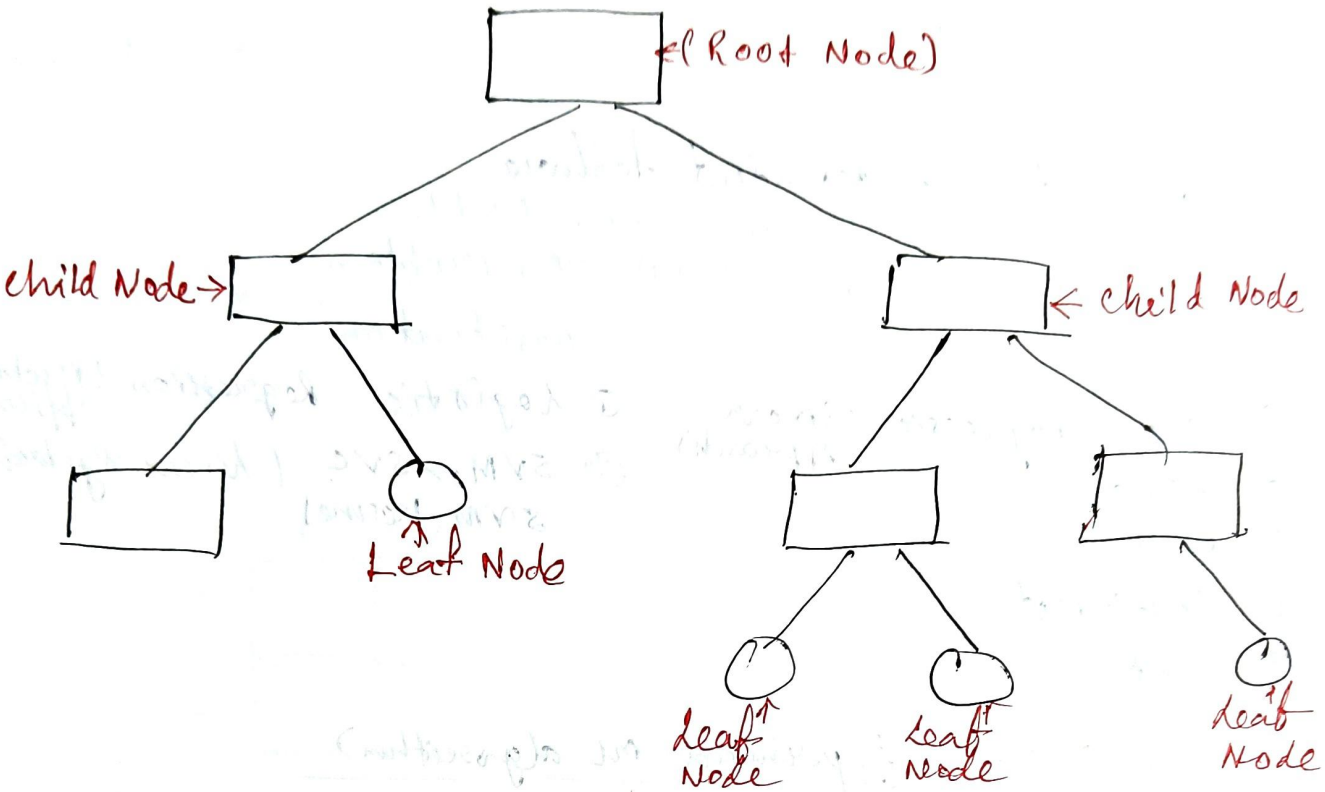
Ensemble Techniques

1. Bagging (Random Forest)
2. Boosting (XGBoost, adaboost, Gradient Boost)
3. Staking

→ Decision Tree can be used for both classification as well as Regression problem.

Decision Tree

① Decision Tree classifier
② Decision Tree Regressor



⇒ output must be present in the leaf node.

eg

Independent		Dependent / Target Variable
Height	Weight	Obese / Not obese
160	60	obese
170	70	Not obese
180	90	obese
⋮	⋮	⋮
180	45	

Que Which feature should we consider as root node?

Ans Height or Weight?

Two ways to decide root node

1. Entropy → Information Gain
2. Gini Indexing

Algorithm for Decision Tree

1. ID3 → ~~Based on Gini Indexing &~~ Entropy & Information Gain
2. CART → classification & Regression Tree based on Gini ~~Indexing~~ Impurity
3. C45 ~~Entropy~~

For Regression

1. We use standard deviation, MSE, RMSE etc to choose root node.

classification Problem (Decision Tree classifier)

Weight	Height	obese/No obese → categorical variable
50	150	Obese
60	160	No obese
70	170	obese
80	175	No obese
90	180	obese

Regression Problem (Decision Tree Regressor)

Weight	Height	BMI → continuous variable
50	150	21
60	160	22
70	170	23
80	175	25
90	185	26

→ weight & Height → Independent features

Entropy

- Measure of impurity

Why scaling is not required in Decision Tree?

Ans

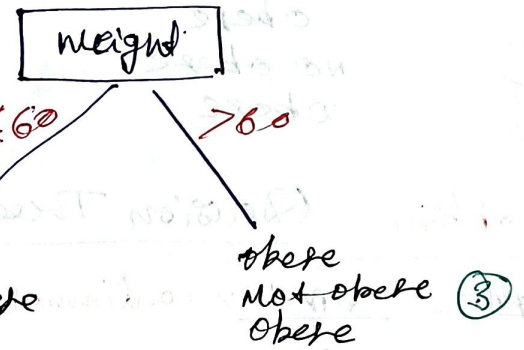
Weight	Height	Obese/No obese
50	165	obese
55	150	Not obese
65	155	obese
70	140	Not obese
75	175	obese

↓ Scaling of the data

Weight	Height	obese/No obese
5.0	16.5	obese
5.5	15.0	Not obese
6.5	15.5	obese ~~Not~~
7.0	14.0	Not obese
7.5	17.5	obese

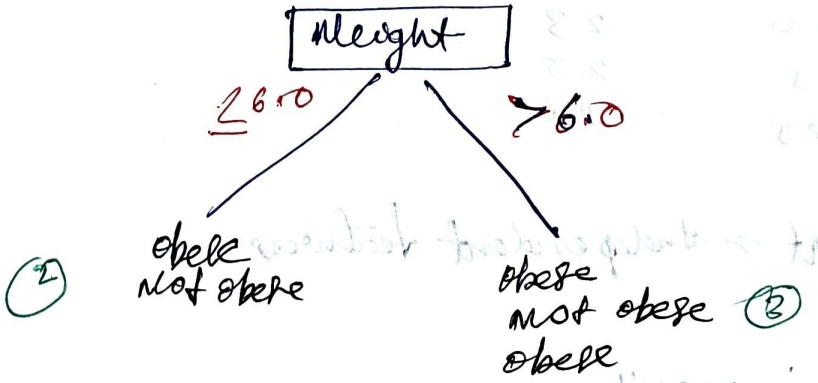
Root Node

Based on Gini Indexing & Entropy →



if weight ≤ 60
else

After scaling

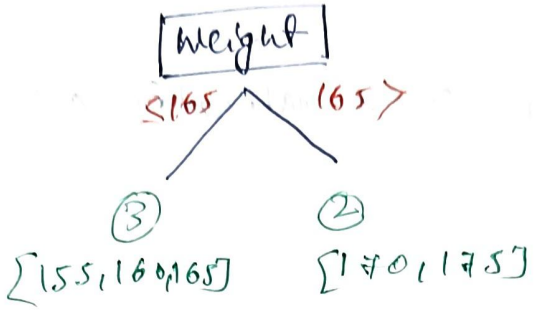


→ There no difference, we can use both, No scaling required.

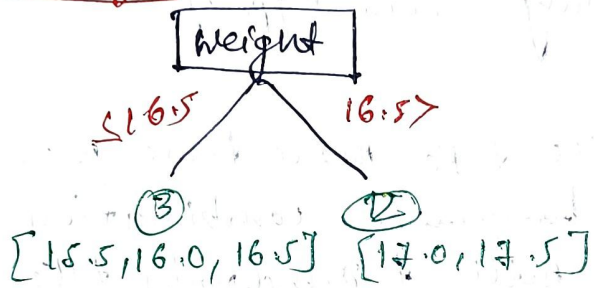
Weight
155
160
165
170
175

After Scaling →

Weight
15.5
16.0
16.5
17.0
17.5



After Scaling



→ No difference after scaling. Hence, No scaling is required in Decision Tree.

→ Even outliers ~~has~~ doesn't have any impact in Decision Tree algorithm.

GridSearchCV

- It is a method to implement hyperparameter tunning.

Hp Hyperparameter

criterion = 'gini',	(two option: 'gini' and 'entropy')
max_depth = none	(2, 3, 4, 5, 6)
min samples split =	(2, 4, 5, 6, 10)
'min_samples_leaf' =	(5, 10, 15, 20)

Hyperparameter Tunning

max_depth = 2, 3, 4, 5, 6
max_sample_leaf = 5, 10, 15, 20, 25
min_sample_split = 10, 15, 20, 25, 80
Criterion = Gini, Entropy

$5^3 = 125$ iterations

$5 \times 5 \times 5 \times 2 = 250$ iterations

① GridSearchCV

→ Use GridSearchCV to perform hyperparameter tunning for any model.

→ If CV included then cross validation will also happen.

② Random Search

→ It will pick randomly from hyperparameter.

Note:- GridSearchCV will take more time because it will take every ~~parameter~~ combination while Random Search will randomly take combination.

CROSS VALIDATION

weight	height	obese/No obese
50	180	Obese
60	160	No Obese
70	170	obese
80	175	No obese
90	180	obese

train_test_split = { train = 3, test = 2 } (Randomly)

Cross Validation

total data points = 100

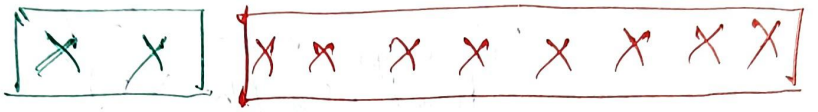
CV = 5

$\left\{ \frac{100}{5} = 20 \right\}$

X X X X X X X X X X

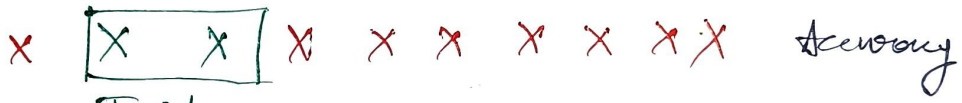
{ 100 - 20 = 80 (Train data points)
20 → (Test data points) }

Steps

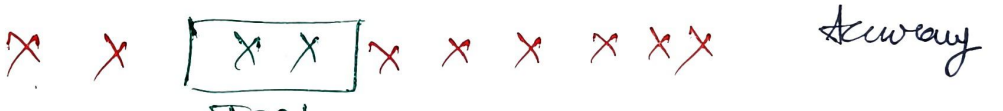


Test Data Points

Train Data Points



Accuracy



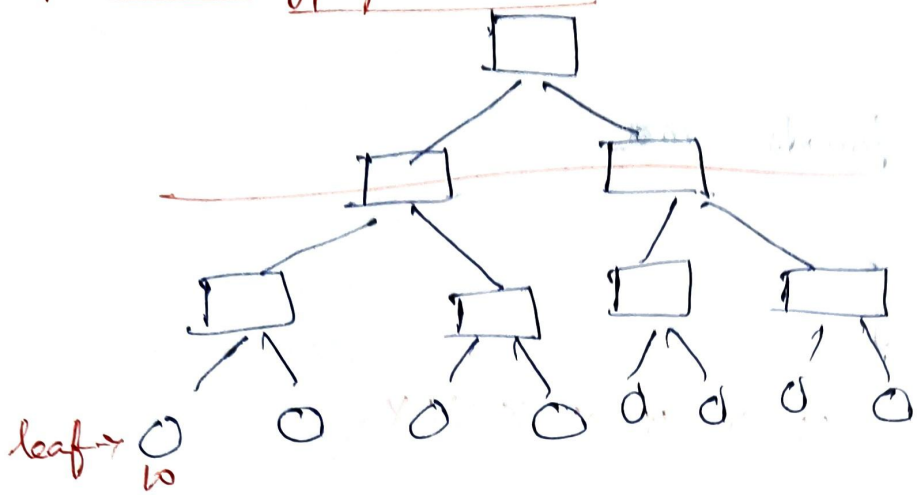
Accuracy

Accuracy.

Test

→ Either we can take maxm accuracy or average accuracy.

Explanation of hyperparameters



Max_depth = Height of the Tree

~~Max_sample_leaf~~

Min_sample_leaf ≥ 5 (always take leaf which have sample > 5)

Min_sample_split ≥ 10 (if min_sample_split conditions meet then don't proceed further).