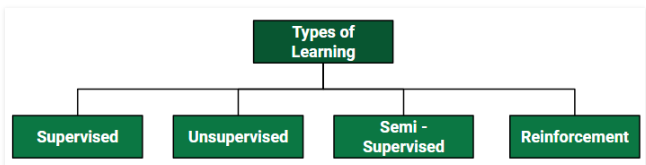
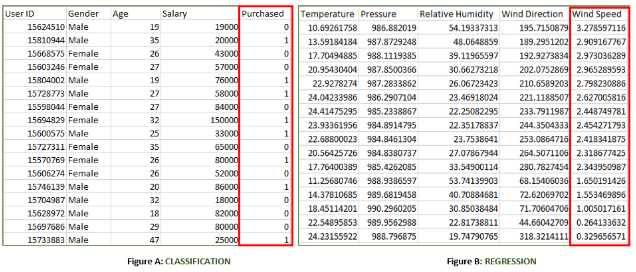
**Machine Learning**

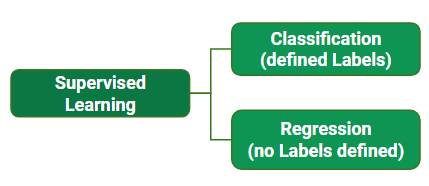


1. **Supervised Learning**

* Supervised learning is when the model is getting trained on a labelled dataset.
* **Labelled** dataset is one which have both input and output parameters.



**TYPES IN SUPERVISED LEARNING**

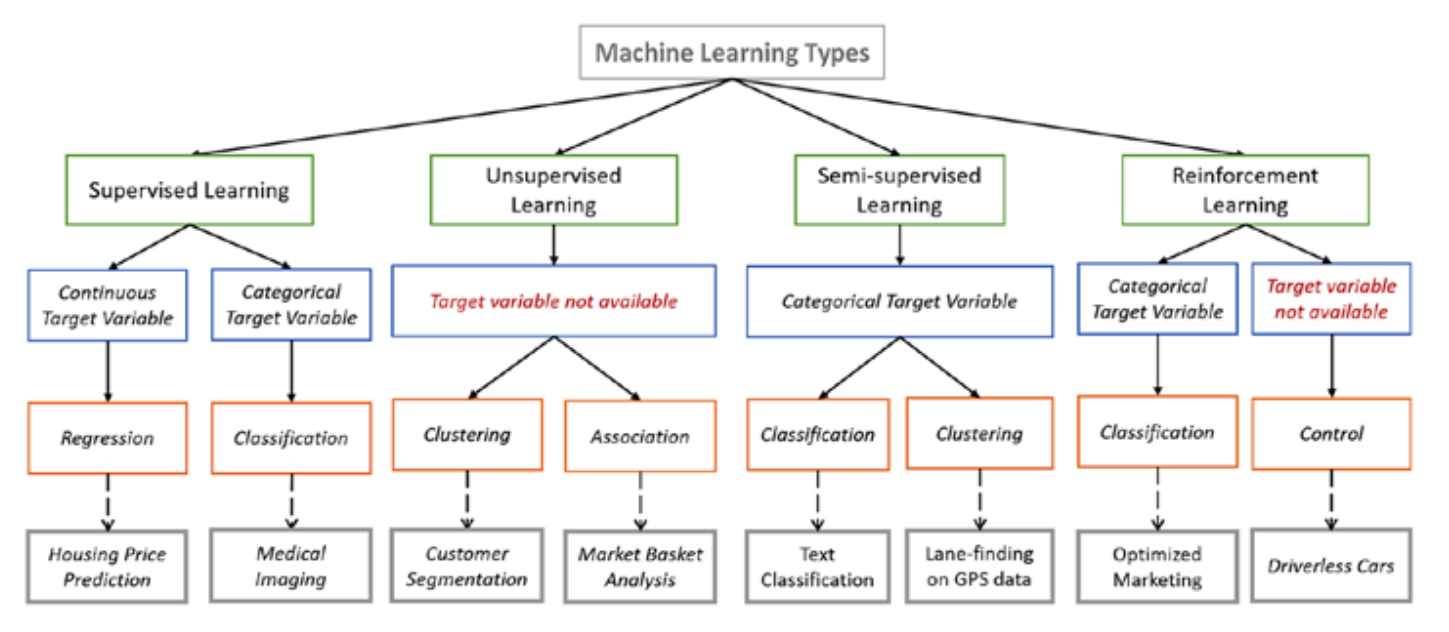


1. **CLASSIFICATION**

* It is a Supervised Learning task where output is having defined labels (discrete value).
* For example in above Figure A,
* **Figure A:**It is a dataset of a shopping store which is useful in predicting whether a customer will purchase a particular product under consideration or not based on his/ her gender, age and salary.
* **Input :** Gender, Age, Salary
* **Output** – Purchased has defined labels i.e. 0 or 1 ;
* 1 means the customer will purchase and 0 means that customer won’t purchase.
* The **goal** here is to predict discrete values belonging to a particular class and evaluate on the basis of accuracy.
* It can be either **binary** or **multi class classification**.
* In **binary** classification, model predicts either 0 or 1; yes or no but in case of **multi class** classification, model predicts more than one class.  
  **Example:** Gmail classifies mails in more than one classes like social, promotions, updates, and forum.
* **Classification Algorithms**
* Decision Tree
* Random Forest
* Adaptive Boosting
* K-Nearest Neighbors
* Naïve Bayes
* Gradient Boosting algorithms
* GBM
* XG Boost
* Light GBM
* Cat Boost

1. **REGRESSION**

* It is a Supervised Learning task where output is having continuous value.
* Example in above Figure B,
* **Figure B:**It is a Meteorological dataset which serves the purpose of predicting wind speed based on different parameters.
* **Input :** Dew Point, Temperature, Pressure, Relative Humidity, Wind Direction
* **Output :** Wind Speed
* Output – Wind Speed is not having any discrete value but is continuous in the particular range.
* The goal here is to predict a value as much closer to actual output value as our model can and then evaluation is done by calculating error value. The smaller the error the greater the accuracy of our regression model.
* **Regression Algorithms**
* Linear Regression



**Training the System**

* While training the model, data is usually split in the ratio of 80:20
* I.e. 80% as training data and rest as testing data.
* In training data, we feed input as well as output for 80% data.
* The model learns from training data only.
* We use different machine learning algorithms to build our model. By learning, it means that the model will build some logic of its own.
* Once the model is ready then it is good to be tested.
* At the time of testing, input is fed from remaining 20% data which the model has never seen before, the model will predict some value and we will compare it with actual output and calculate the accuracy.

<https://www.analyticsvidhya.com/blog/2017/09/common-machine-learning-algorithms/>

1. **DECISION TREE ALGORITHM**

* Decision tree algorithm falls under the category of supervised learning. They can be used to solve both regression and classification problems.
* Decision tree uses the tree representation to solve the problem in which each leaf node corresponds to a class label.

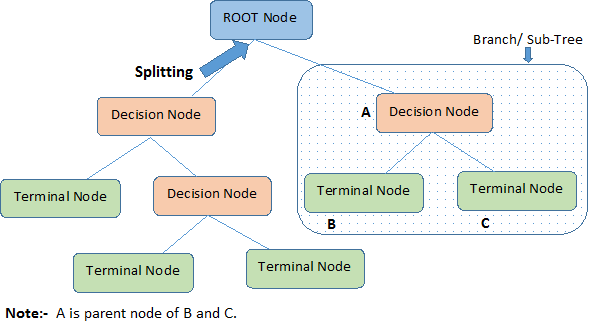
**Types of Decision Trees**

* Types of decision tree is based on the type of target variable we have. It can be of two types:
* **Categorical Variable Decision Tree:** Decision Tree which has categorical target variable then it called as categorical variable decision tree.
* **Continuous Variable Decision Tree:**Decision Tree has continuous target variable then it is called as Continuous Variable Decision Tree.

**Important Terminology related to Decision Trees**

Let’s look at the basic terminology used with Decision trees:

1. **Root Node:**It represents entire population or sample and this further gets divided into two or more homogeneous sets.
2. **Splitting:**It is a process of dividing a node into two or more sub-nodes.
3. **Decision Node:**When a sub-node splits into further sub-nodes, then it is called decision node.
4. **Leaf/ Terminal Node:**Nodes do not split is called Leaf or Terminal node.

[](https://www.analyticsvidhya.com/wp-content/uploads/2015/01/Decision_Tree_2.png)

1. **Pruning:**When we remove sub-nodes of a decision node, this process is called pruning. You can say opposite process of splitting.
2. **Branch / Sub-Tree:**A sub section of entire tree is called branch or sub-tree.
3. **Parent and Child Node:**A node, which is divided into sub-nodes is called parent node of sub-nodes whereas sub-nodes are the child of parent node.

**HOW TO CHOOSE ROOT NODE**

* In Decision Tree the major challenge is to identification of the attribute for the root node in each level. This process is known as attribute selection.
* We have two popular attribute selection measures:

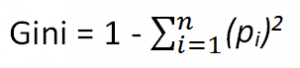
1. Information Gain
2. Gini Index
3. Id3 Algorithm(<https://www.youtube.com/watch?v=UdTKxGQvYdc&feature=youtu.be>)

**GINI INDEX/GINI IMPURITY**

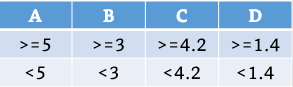
* Gini index or Gini impurity measures the degree or probability of a particular variable being wrongly classified when it is randomly chosen. But what is actually meant by ‘impurity’?
* If all the elements belong to a single class, then it can be called pure.
* The degree of Gini index varies between 0 and 1, where 0 denotes that all elements belong to a certain class or if there exists only one class, and 1 denotes that the elements are randomly distributed across various classes.
* A Gini Index of 0.5 denotes equally distributed elements into some classes.
* While building the decision tree, we would prefer choosing the attribute/feature with the least Gini index as the root node.

**FORMULA**

* **Formula for Gini Index**

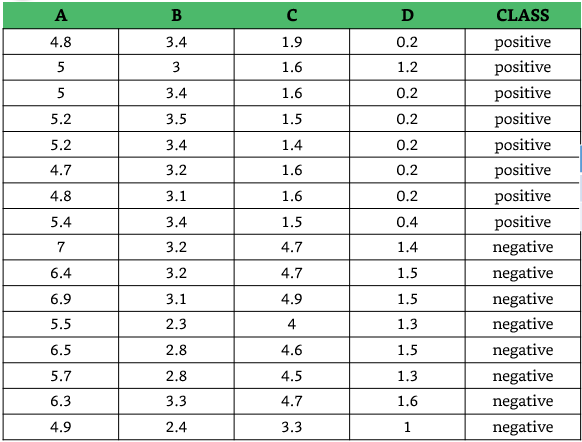


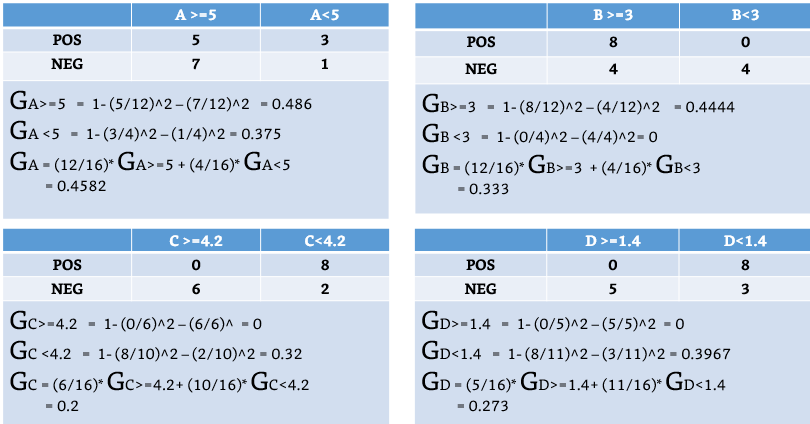
* Where piis the probability of an object being classified to a particular class.



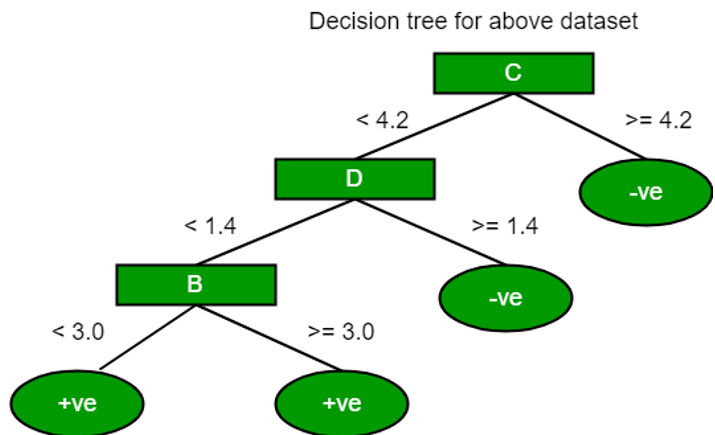
<https://blog.quantinsti.com/gini-index/>

<https://medium.com/@pytyagi/decision-tree-ac0c9e3b8258>





<https://www.linkedin.com/pulse/how-does-id3-algorithm-works-decision-trees-sagarnil-das/>



<https://www.xoriant.com/blog/product-engineering/decision-trees-machine-learning-algorithm.html>

**ADVANTAGES OF DECISION TREE**

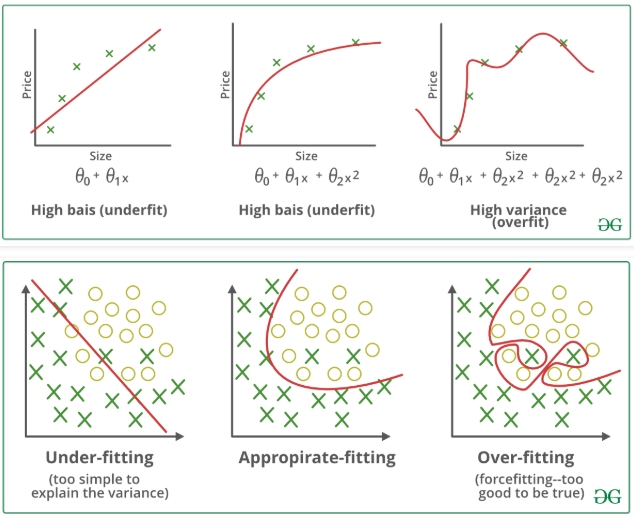
* A **decision tree** is simple to understand and after a brief exploration, we construct it.
* It can deal with numerical as well as categorical data.
* Resistant to outliers, hence require little data preprocessing.

**DISADVANTAGES OF DECISION TREE**

* Prone to overfitting.
* Greedy Approach: Splitting data according to the first best split and it will split whole data by that path only.
* But it is possible that different paths of the split are more informative so that split may not be the best split.

**OVER FITTING**

* A statistical model is said to be over fitted, when we train it with a lot of data.
* When a model gets trained with so much of data, it starts learning from the noise and inaccurate data entries in our data set.
* Then the model does not categorize the data correctly, because of too much of details and noise.
* The causes of overfitting are the non-parametric and non-linear methods because these types of machine learning algorithms have more freedom in building the model based on the dataset and therefore they can really build unrealistic models.
* Over fitting occurs specifically if the model or algorithm shows **high variance** and **low bias**.



**UNDER FITTING**

* A statistical model or a machine learning algorithm is said to have under fitting when it cannot capture the underlying trend of the data.
* <https://elitedatascience.com/overfitting-in-machine-learning>
* Its occurrence simply means that our model or the algorithm does not fit the data well enough.
* It usually happens when we have less data to build an accurate model and also when we try to build a linear model with a non-linear data
* Under fitting can be avoided by using more data and also reducing the features by feature selection.
* Specifically under fitting occurs if the model or algorithm shows low variance and high bias.

## **How to avoid overfitting the Decision tree model**

* To avoid decision tree from overfitting **we remove the branches that make use of features having low importance.**
* This method is called as **Pruning or post-pruning.**
* We can prune decision Tree by setting Max-depth of the tree or by setting minimum data points in each node.
* This way we will reduce the complexity of tree, and hence improves predictive accuracy by the reduction of overfitting.

**PARAMETER TUNING**

* **Max \_depth**
* The first parameter to tune is max \_depth.
* This indicates how deep the tree can be.
* The deeper the tree, the more splits it has and it captures more information about the data.
* **Min \_ samples \_split**
* Min \_samples \_split represents the minimum number of samples required to split an internal node.
* This can vary between considering at least one sample at each node to considering all of the samples at each node.
* When we increase this parameter, the tree becomes more constrained as it has to consider more samples at each node.