UNIT II SUPERVISED LEARNING

Regression

Regression analysis is a statistical method that helps us to analyze and understand the relationship between two or more variables of interest. The process that is adapted to perform regression analysis helps to understand which factors are important, which factors can be ignored, and how they are influencing each other.

For the regression analysis is be a successful method, we understand the following terms:

- Dependent Variable: This is the variable that we are trying to understand or forecast.
- Independent Variable: These are factors that influence the analysis or target variable and provide us with information regarding the relationship of the variables with the target variable.

In regression, we normally have one dependent variable and one or more independent variables. Here we try to "regress" the value of dependent variable "Y" with the help of the independent variables. In other words, we are trying to understand, how does the value of 'Y' change w.r.t change in 'X'.

What is Regression Analysis? General Uses of Regression Analysis

Regression analysis is used for prediction and forecasting. This has a substantial overlap to the field of machine learning. This statistical method is used across different industries such as,

- Financial Industry- Understand the trend in the stock prices, forecast the prices, evaluate risks in the insurance domain
- Marketing- Understand the effectiveness of market campaigns, forecast pricing and sales of the product.
- Manufacturing- Evaluate the relationship of variables that determine to define a better engine to provide better performance
- Medicine- Forecast the different combination of medicines to prepare generic medicines for diseases.

Terminologies used in Regression Analysis

Outliers

Suppose there is an observation in the dataset that has a very high or very low value as compared to the other observations in the data, i.e. it does not belong to the population, such an observation is called an outlier. In simple words, it is an extreme value. An outlier is a problem because many times it hampers the results we get.

Multicollinearity

When the independent variables are highly correlated to each other, then the variables are said to be multicollinear. Many types of regression techniques assume multicollinearity should not be present in the dataset. It is because it causes problems in ranking variables based on its importance, or it makes the job difficult in selecting the most important independent variable.

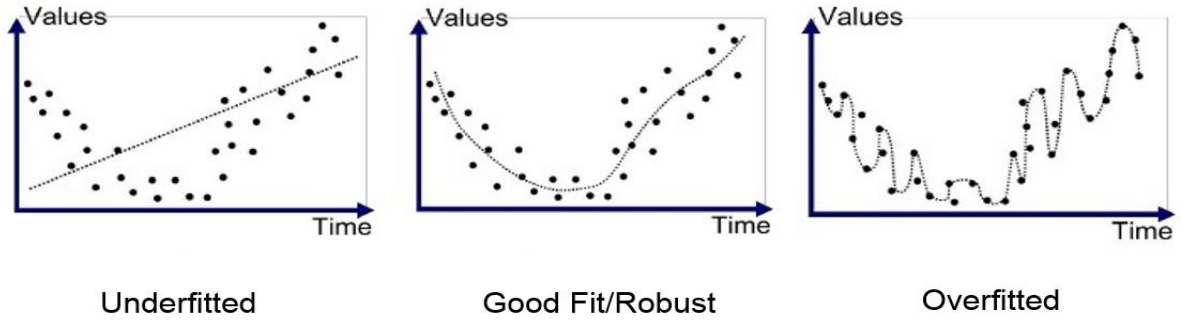
Heteroscedasticity

When the variation between the target variable and the independent variable is not constant, it is called heteroscedasticity. Example-As one's income increases, the variability of food consumption will increase. A poorer person will spend a rather constant amount by always eating inexpensive food; a wealthier person may occasionally buy inexpensive food and at other times, eat expensive meals. Those with higher incomes display a greater variability of food consumption.

Underfit and Overfit

When we use unnecessary explanatory variables, it might lead to overfitting. Overfitting means that our algorithm works well on the training set but is unable to perform better on the test sets. It is also known as a problem of **high variance**.

When our algorithm works so poorly that it is unable to fit even a training set well, then it is said to underfit the data. It is also known as a problem of **high bias**.



Types of Regression

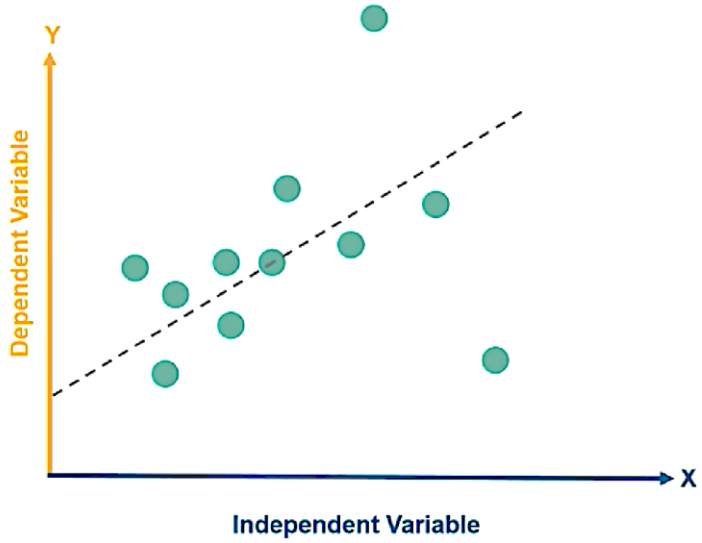
For different types of Regression analysis, there are assumptions that need to be considered along with understanding the nature of variables and its distribution.

Linear Regression

The simplest of all regression types is Linear Regression where it tries to establish relationships between Independent and Dependent variables. The Dependent variable considered here is always a continuous variable.

What is Linear Regression?

Linear Regression is a predictive model used for finding the ***linear*** relationship between a dependent variable and one or more independent variables.



Here, 'Y' is our dependent variable, which is a continuous numerical and we are trying to understand how does 'Y' change with 'X'.

So, if we are supposed to answer, the above question of "What will be the GRE score of the student, if his CCGPA is 8.32?" our go-to option should be linear regression.

Examples of Independent & Dependent Variables:

• x is Rainfall and y is Crop Yield

• x is Advertising Expense and y is Sales

• x is sales of goods and y is GDP

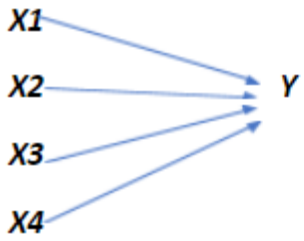
If the relationship with the dependent variable is in the form of single variables, then it is known as Simple Linear Regression

Simple Linear Regression

X ——> Y

If the relationship between Independent and dependent variables are multiple in number, then it is called Multiple Linear Regression

Multiple Linear Regression



Simple Linear Regression Model

As the model is used to predict the dependent variable, the relationship between the variables can be written in the below format.

$\boldsymbol{Y_i} = \beta_0 + \beta_1 X_i + \varepsilon_i$

Where,

$\boldsymbol{Y_i}$ – Dependent variable

β_0 — Intercept

β_1 – Slope Coefficient

X_i – Independent Variable

ε_i – Random Error Term

The main factor that is considered as part of Regression analysis is understanding the variance between the variables. For understanding the variance, we need to understand the measures of variation.

$$SST = SSR + SSE$$

Total Sum of Squares	Regression Sum of Squares	Error Sum of Squares

- **SST = total sum of squares (Total Variation)**
    - Measures the variation of the Y i values around their mean Y
- **SSR = regression sum of squares (Explained Variation)**
    - Variation attributable to the relationship between X and Y
- **SSE = error sum of squares (Unexplained Variation)**
    - Variation in Y attributable to factors other than X

With all these factors taken into consideration, before we start assessing if the model is doing good, we need to consider the assumptions of Linear Regression.

Assumptions:

Since Linear Regression assesses whether one or more predictor variables explain the dependent variable and hence it has 5 assumptions:

1. Linear Relationship

2. Normality

3. No or Little Multicollinearity

4. No Autocorrelation in errors

5. Homoscedasticity

With these assumptions considered while building the model, we can build the model and do our predictions for the dependent variable. For any type of machine learning model, we need to understand if the variables considered for the model are correct and have been analysed by a metric. In the case of Regression analysis, the statistical measure that evaluates the model is called the ***coefficient of determination which is represented as r^2.***

The coefficient of determination is the portion of the total variation in the dependent variable that is explained by variation in the independent variable. A higher value of r^2 better is the model with the independent variables being considered for the model.

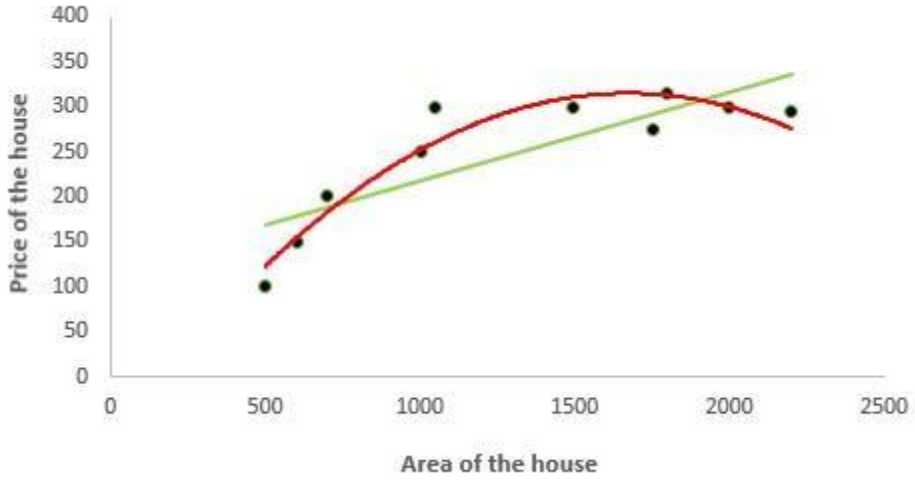
$$r^2 = SSR$$

Note: The value of r^2 is the range of $0 \leq r^2 \leq 1$

Polynomial Regression

This type of regression technique is used to model nonlinear equations by taking polynomial functions of independent variables.

In the figure given below, you can see the red curve fits the data better than the green curve. Hence in the situations where the relationship between the dependent and independent variable seems to be non-linear, we can deploy **Polynomial Regression Models**.



Thus a polynomial of degree k in one variable is written as:

$$y = \beta_0 + \beta_1 X + \beta_2 X^2 + \ldots + \beta_k X^k + \varepsilon$$

Here we can create new features like

$$X_1 = x, X_2 = x^2, \ldots, X_k = x^k$$

and can fit linear regression in a similar manner.

In case of multiple variables say X1 and X2, we can create a third new feature (say X3) which is the product of X1 and X2 i.e.

$$X_3 = X_1 * X_2$$

The main drawback of this type of regression model is if we create unnecessary extra features or fitting polynomials of higher degree this may lead to overfitting of the model.

Logistic Regression

Logistic Regression is also known as Logit, Maximum-Entropy classifier is a supervised learning method for classification. It establishes a relation between dependent class variables and independent variables using regression.

The dependent variable is categorical i.e. it can take only integral values representing different classes. The probabilities describing the possible outcomes of a query point are modelled using a logistic function. This model belongs to a family of discriminative classifiers. They rely on attributes which discriminate the classes well. This model is used when we have 2 classes of dependent variables. When there are more than 2 classes, then we have another regression method which helps us to predict the target variable better.

There are two broad categories of Logistic Regression algorithms

1. Binary Logistic Regression when the dependent variable is strictly binary
2. Multinomial Logistic Regression when the dependent variable has multiple categories.

There are two types of Multinomial Logistic Regression

1. Ordered Multinomial Logistic Regression (dependent variable has ordered values)
2. Nominal Multinomial Logistic Regression (dependent variable has unordered categories)

Process Methodology:

Logistic regression takes into consideration the different classes of dependent variables and assigns probabilities to the event happening for each row of information. These probabilities are found by assigning different weights to each independent variable by understanding the relationship between the variables. If the correlation between the variables is high, then

positive weights are assigned and in the case of an inverse relationship, negative weight is assigned.

As the model is mainly used to classify the classes of target variables as either 0 or 1, thus the Sigmoid function is obtained by implementing the log-normal function on these probabilities that are calculated on these independent variables.

The Sigmoid function:

P(y= 1) = Sigmoid(Z) = 1/(1 + e -z)

P(y= 0) = 1 –P(y =1) = 1 –(1/(1 + e -z)) = e –z/ (1 + e -z)

y = 1 if P(y=1|X) > .5, else y = 0

where the default probability cut off is taken as 0.5.

$$logLoss = \frac{-1}{N}\sum_{i=1}^{N}(y_i(logp_i) + (1 - y_i)log(1 - p_i))$$

This method is also called the Odds Log ratio.

Assumptions:

1. The dependent variable is categorical. Dichotomous for binary logistic regression and multi-label for multi-class classification

2. Attributes and log odds i.e. log(p / 1-p) should be linearly related to the independent variables

3. Attributes are independent of each other (low or no multicollinearity)

4. In binary logistic regression class of interest is coded with 1 and other class 0

5. In multi-class classification using Multinomial Logistic Regression or OVR scheme, class of interest is coded 1 and rest 0(this is done by the algorithm)

Note: the assumptions of Linear Regression such as homoscedasticity, normal distribution of error terms, a linear relationship between the dependent and independent variables are not required here.

Some examples where this model can be used for predictions.

1. **Predicting the weather:** you can only have a few definite weather types. Stormy, sunny, cloudy, rainy and a few more.

2. **Medical diagnosis:** given the symptoms predicted the disease patient is suffering from.

3. **Credit Default:** If a loan has to be given a particular candidate depend on his identity check, account summary, any properties he holds, any previous loan, etc

4. **HR Analytics:** IT firms recruit a large number of people, but one of the problems they encounter is after accepting the job offer many candidates do not join. So, this results in cost overruns because they have to repeat the entire process again. Now when you get an application, can you actually predict whether that applicant is likely to join the organization (Binary Outcome – Join / Not Join).

5. **Elections:** Suppose that we are interested in the factors that influence whether a political candidate wins an election. The outcome (response) variable is binary (0/1); win or lose. The predictor variables of interest are the amount of money spent on the campaign and the amount of time spent campaigning negatively.

Linear Discriminant Analysis (LDA)

Discriminant Analysis is used for classifying observations to a class or category based on predictor (independent) variables of the data.

Discriminant Analysis creates a model to predict future observations where the classes are known.

LDA comes to our rescue in situations when logistic regression is unstable when

1. Classed are well separated
2. Data is small
3. When we have more than 2 classes

Working Process of LDA Model

The LDA model uses Bayes' Theorem to estimate probabilities. They make predictions upon the probability that a new input dataset belongs to each class. The class which has the highest probability is considered as the output class and then the LDA makes a prediction.

The prediction is made simply by the use of Bayes' theorem which estimates the probability of the output class given the input. They also make use of the probability of each class and also the data belonging to that class:

$P(Y=x|X=x) = [(Plk* fk(x))] / [sum(Pll* fl(x))]$

Where

k=output class

Plk= Nk/n or base probability of each class observed in the training data. It is also called prior probability in Bayes' theorem.

fk(x) = estimated probability of x belonging to class k.

Regularized Linear Models

This method is used to solve the problem of overfitting of the model which arises due to the model performing poorly on test data. This model helps us to solve the problem by adding an error term to the objective function to reduce the bias in the model.

Regularization is generally useful in the following situations:

- A large number of variables
- Low ratio of number observations to the number of variables
- High Multicollinearity

L1 Loss function or L1 Regularization

In L1 regularization we try to minimize the objective function by adding a penalty term to the sum of the absolute values of coefficients. This is also known as the least absolute deviations method. **Lasso Regression (Least Absolute Shrinkage Selector Operator)** makes use of L1 regularization. It takes the minimum absolute values of the coefficients.

The cost function for lasso regression

Min(||Y – X(theta)||^2 + λ||theta||)

λ is the hypermeter, whose value is equal to the alpha in the Lasso function

It is generally used when we have more number of features because it automatically does feature selection.

1. L2 Loss function or L2 Regularization

In L2 regularization we try to minimize the objective function by adding a penalty term to the sum of the squares of coefficients. **Ridge Regression** or shrinkage regression makes use of L2 regularization. This model assumes the square of the absolute values if coefficient.

The cost function for ridge regression

Min(||Y – X(theta)||^2 + λ||theta||^2)

Lambda is the penalty term. λ given here is actually denoted by an alpha parameter in the ridge function. So by changing the values of alpha, we are basically controlling the penalty term. Higher the values of alpha, bigger is the penalty and therefore the magnitude of coefficients is reduced.

It shrinks the parameters, therefore it is mostly used to prevent multicollinearity

It reduces the model complexity by coefficient shrinkage

Value of alpha, which is a hyperparameter of Ridge, which means that they are not automatically learned by the model instead they have to be set manually.

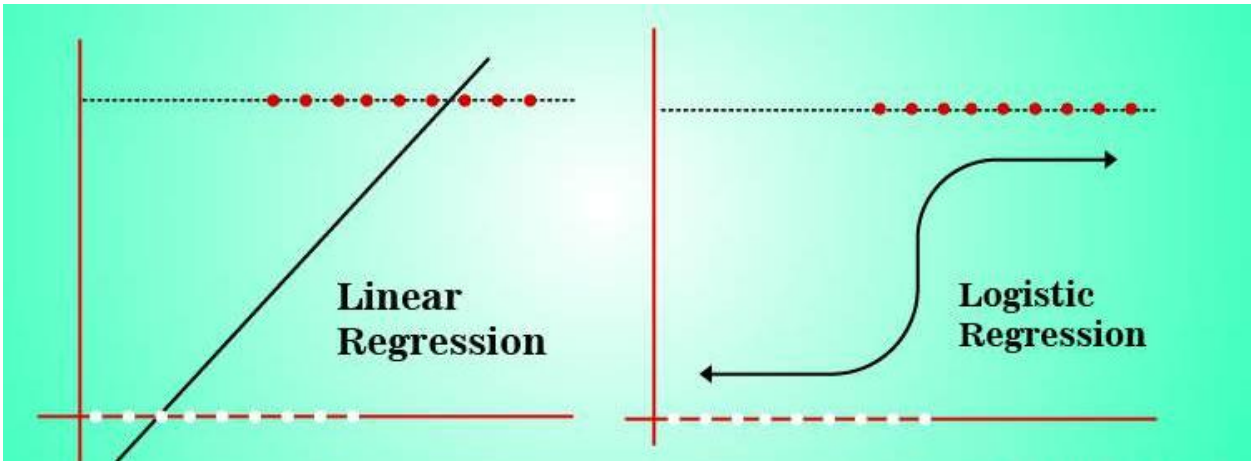
A combination of both Lasso and Ridge regression methods brings rise to a method called Elastic Net Regression where the cost function is :

Min(||Y-Xtheta||^2 + Lambda1||theta|| + lambda2||theta||^2)

Linear versus Logistic Regression

Linear Regression	Logistic Regression
■Target is an interval variable. ■Input variables have any measurement level. ■Predicted values are the mean of the target variable at the given values of the input variables.	■Target is a discrete (binary or ordinal) variable. ■Input variables have any measurement level. ■Predicted values are the probability of a particular level(s) of the target variable at the given values of the input variables.

8



K-Nearest Neighbors

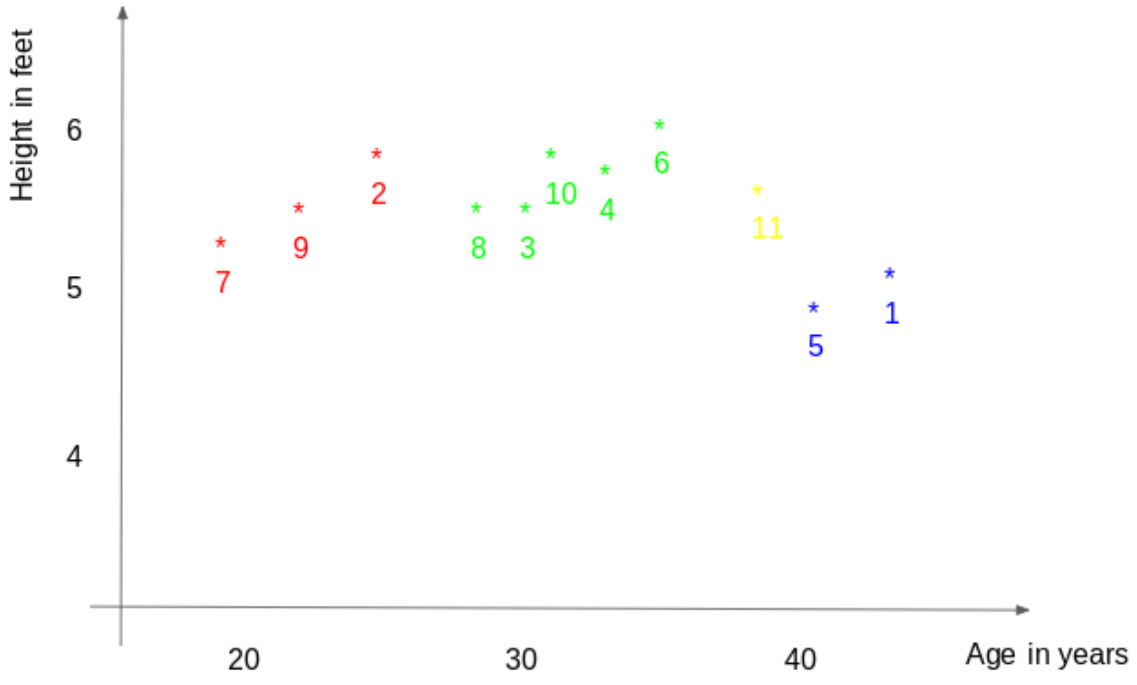
KNN algorithm can be used for both classification and regression problems. The KNN algorithm uses 'feature similarity' to predict the values of any new data points

Let us start with a simple example. Consider the following table – it consists of the height, age and weight (target) value for 10 people. As you can see, the weight value of ID11 is missing. We need to predict the weight of this person based on their height and age.

Note: The data in this table does not represent actual values. It is merely used as an example to explain this concept.

ID	Height	Age	Weight
1	5	45	77
2	5.11	26	47
3	5.6	30	55
4	5.9	34	59
5	4.8	40	72
6	5.8	36	60
7	5.3	19	40
8	5.8	28	60
9	5.5	23	45
10	5.6	32	58
11	5.5	38	?

For a clearer understanding of this, below is the plot of height versus age from the above table:



In the above graph, the y-axis represents the height of a person (in feet) and the x-axis represents the age (in years). The points are numbered according to the ID values. The yellow point (ID 11) is our test point.

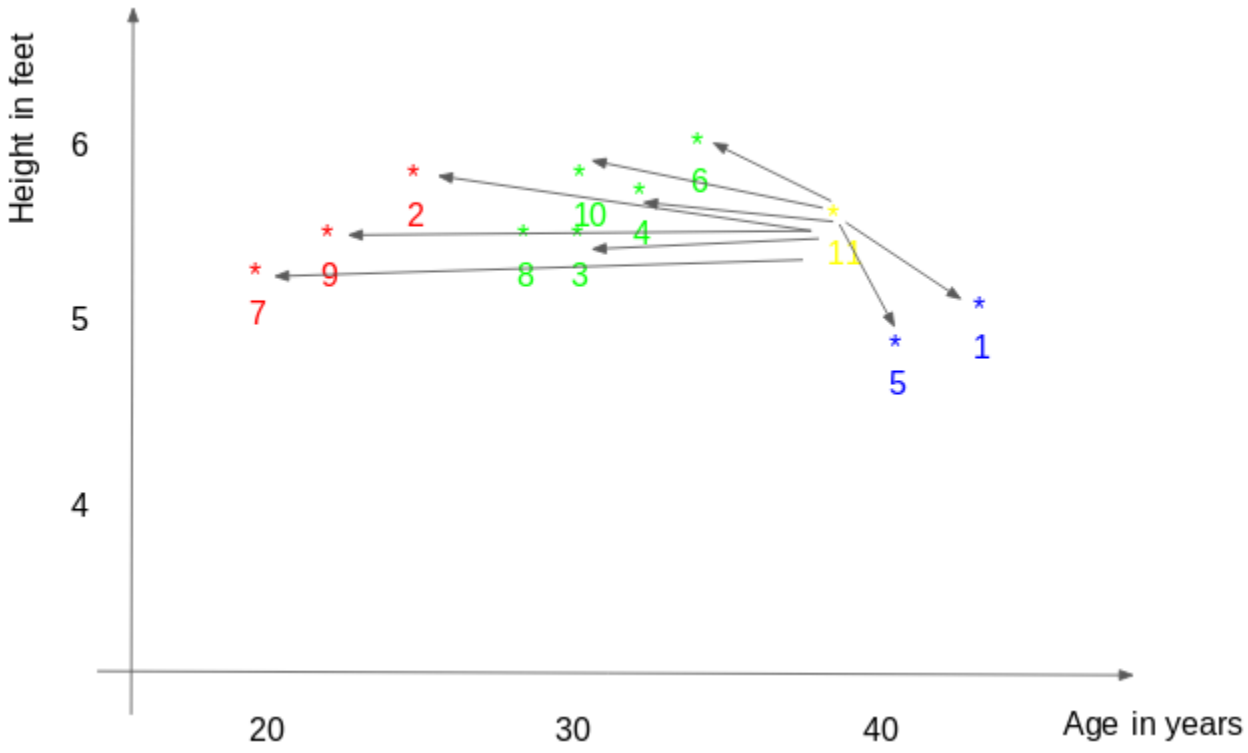
If I ask you to identify the weight of ID11 based on the plot, what would be your answer? You would likely say that since ID11 is **closer** to points 5 and 1, so it must have a weight similar to these IDs, probably between 72-77 kgs (weights of ID1 and ID5 from the table). That actually makes sense, but how do you think the algorithm predicts the values? We will find that out in this article.

As we saw above, KNN algorithm can be used for both classification and regression problems. The KNN algorithm uses '**feature similarity**' to predict the values of any new data points. This means that the new point is assigned a value based on how closely it resembles the points in the training set. From our example, we know that ID11 has height and age similar to ID1 and ID5, so the weight would also approximately be the same.

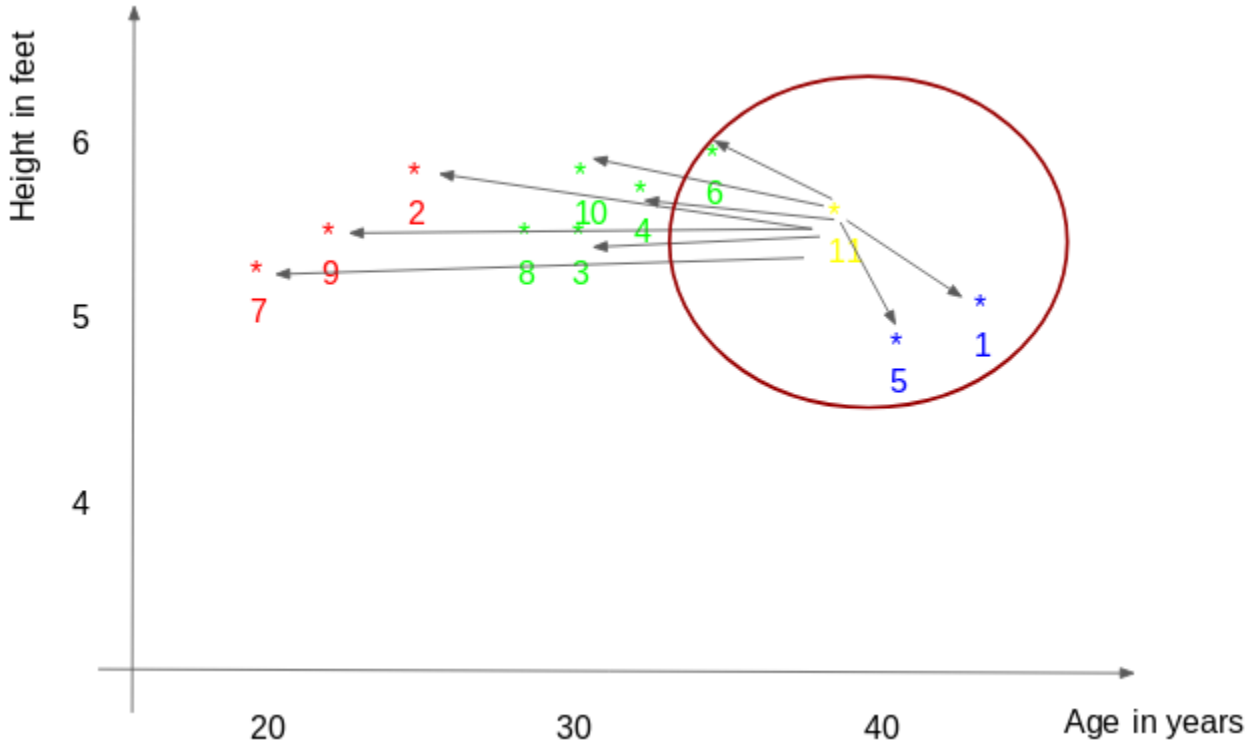
Had it been a classification problem, we would have taken the mode as the final prediction. In this case, we have two values of weight – 72 and 77. Any guesses on how the final value will be calculated? The average of the values is taken to be the final prediction.

Below is a stepwise explanation of the algorithm:

1. First, the distance between the new point and each training point is calculated.



2. The closest k data points are selected (based on the distance). In this example, points 1, 5, 6 will be selected if the value of k is 3. We will further explore the method to select the right value of k later in this article.



3. The average of these data points is the final prediction for the new point. Here, we have weight of ID11 = (77+72+60)/3 = 69.66 kg.

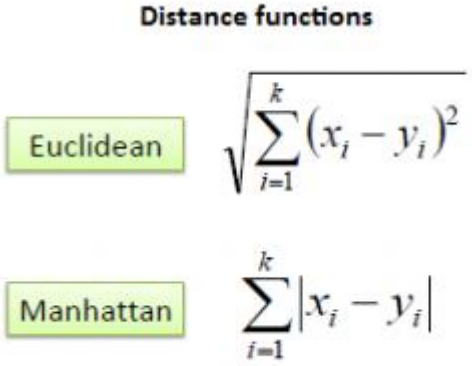
In the next few sections, we will discuss each of these three steps in detail.

3. Methods of the calculating distance between points

The **first step** is to calculate the distance between the new point and each training point. There are various methods for calculating this distance, of which the most commonly known methods are – Euclidian, Manhattan (for continuous) and Hamming distance (for categorical).

1. **Euclidean Distance:** Euclidean distance is calculated as the square root of the sum of the squared differences between a new point (x) and an existing point (y).

2. **Manhattan Distance**: This is the distance between real vectors using the sum of their absolute difference.



3. **Hamming Distance**: It is used for categorical variables. If the value (x) and the value (y) are the same, the distance D will be equal to 0 . Otherwise D=1.

$$D_H = \sum_{i=1}^{k} |x_i - y_i|$$

$$x = y \Rightarrow D = 0$$

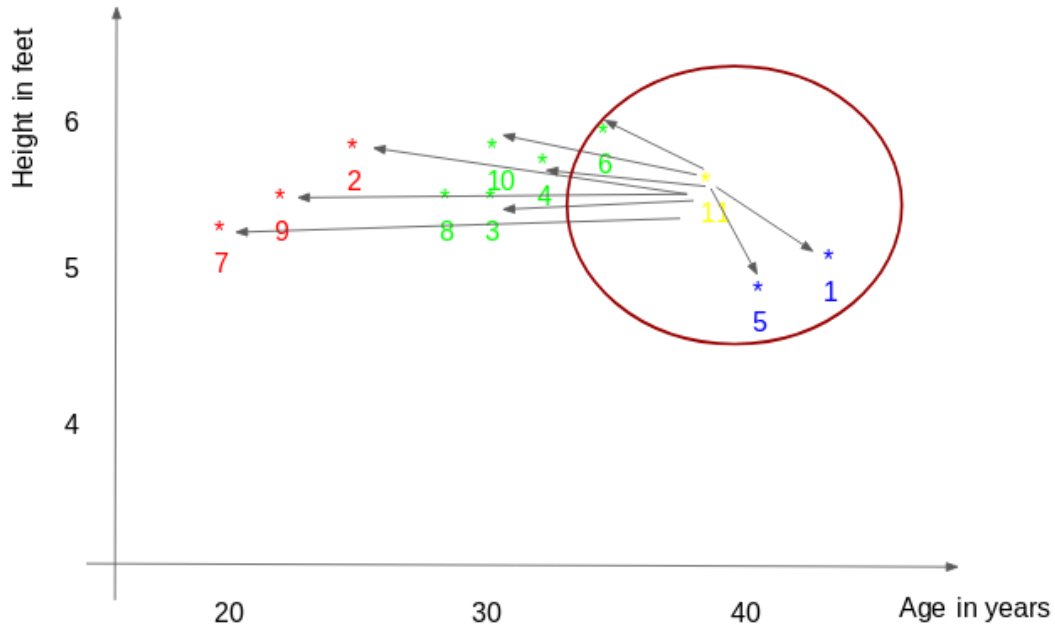
$$x \neq y \Rightarrow D = 1$$

Once the distance of a new observation from the points in our training set has been measured, the next step is to pick the closest points. The number of points to be considered is defined by the value of k.

4. How to choose the k factor?

The **second step** is to select the k value. This determines the number of neighbors we look at when we assign a value to any new observation.

In our example, for a value k = 3, the closest points are ID1, ID5 and ID6.



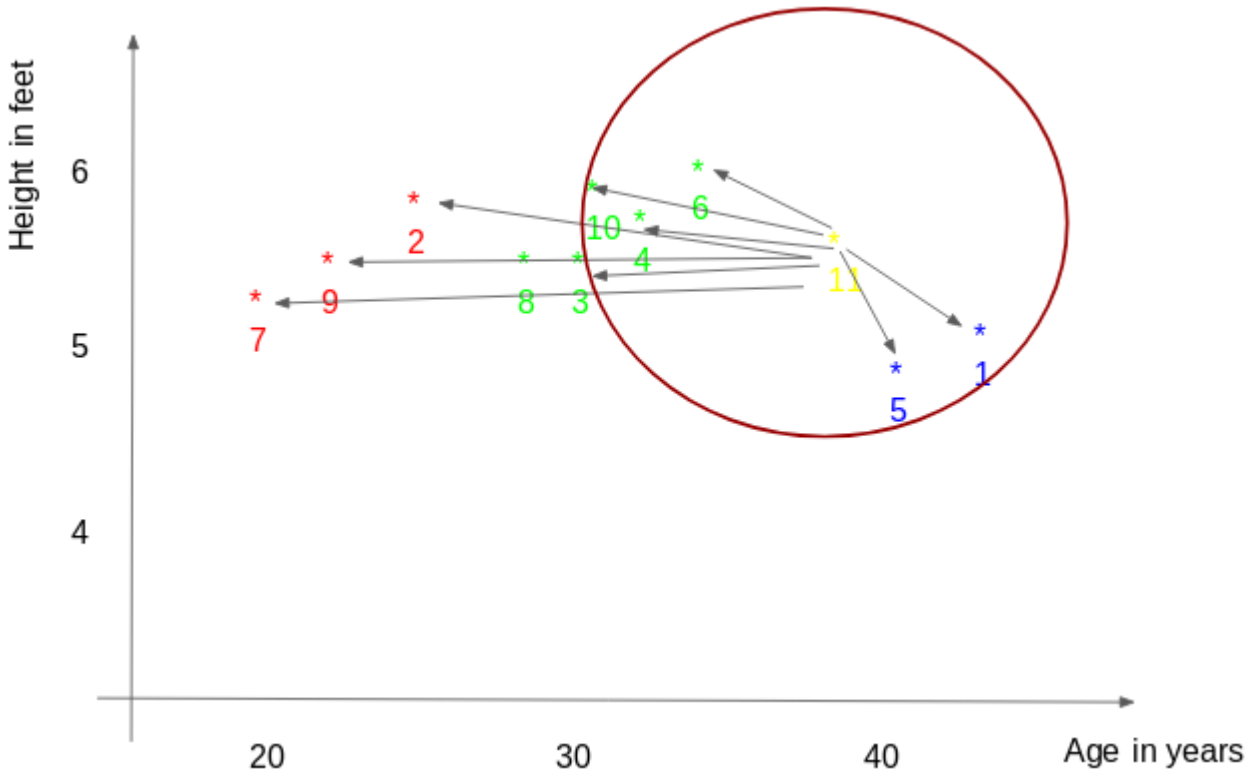
ID	Height	Age	Weight
1	5	45	77
5	4.8	40	72
6	5.8	36	60

The prediction of weight for ID11 will be:

ID11 = (77+72+60)/3

ID11 = 69.66 kg

For the value of k=5, the closest point will be ID1, ID4, ID5, ID6, ID10.



ID	Height	Age	Weight
1	5	45	77
4	5.9	34	59
5	4.8	40	72
6	5.8	36	60
10	5.6	32	58

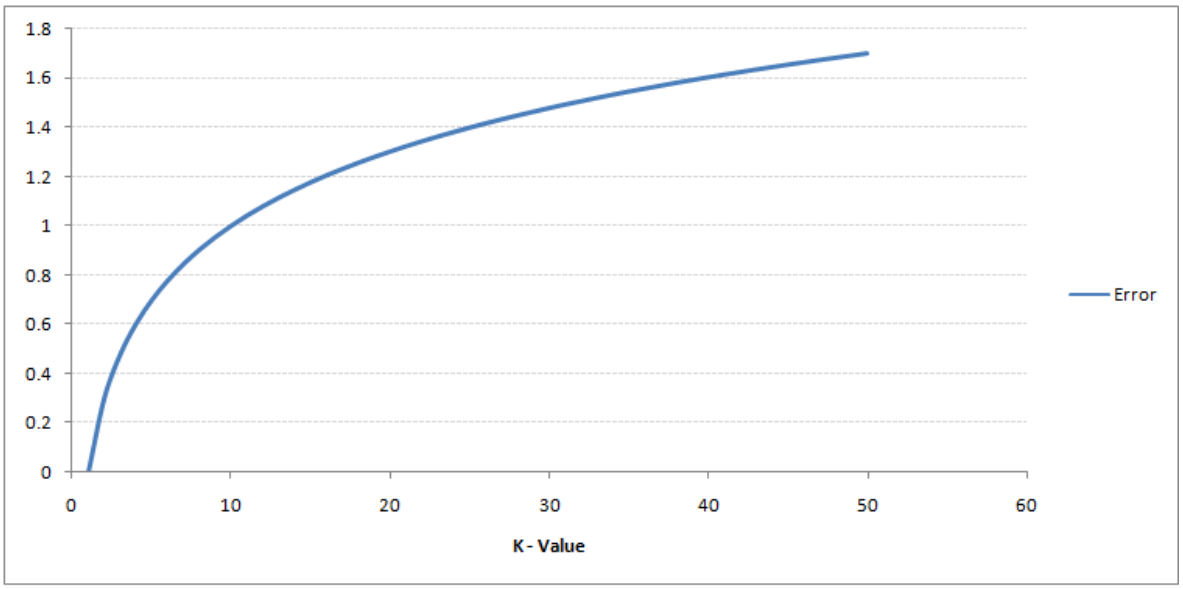
The prediction for ID11 will be :

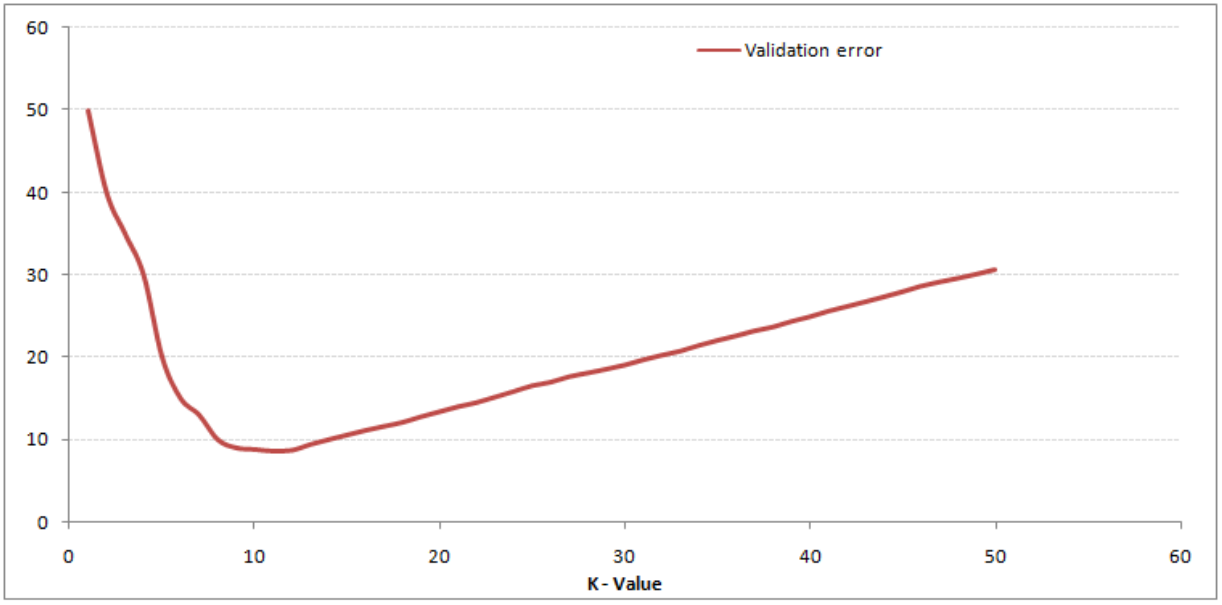
ID 11 = (77+59+72+60+58)/5

ID 11 = 65.2 kg

We notice that based on the k value, the final result tends to change. Then how can we figure out the optimum value of k? Let us decide it based on the error calculation for our train and validation set (after all, minimizing the error is our final goal!).

Have a look at the below graphs for training error and validation error for different values of k.





For a very low value of k (suppose k=1), the model overfits on the training data, which leads to a high error rate on the validation set. On the other hand, for a high value of k, the model performs poorly on both train and validation set. If you observe closely, the validation error curve reaches a minima at a value of k = 9. This value of k is the optimum value of the model (it will vary for different datasets). This curve is known as an '**elbow curve**' (because it has a shape like an elbow) and is usually used to determine the k value.

You can also use the grid search technique to find the best k value.

Decision Tree Algorithm

Decision Tree algorithm belongs to the family of supervised learning algorithms. Unlike other supervised learning algorithms, the decision tree algorithm can be used for solving **regression and classification problems** too.

The goal of using a Decision Tree is to create a training model that can use to predict the class or value of the target variable by **learning simple decision rules** inferred from prior data(training data).

In Decision Trees, for predicting a class label for a record we start from the **root** of the tree. We compare the values of the root attribute with the record's attribute. On the basis of comparison, we follow the branch corresponding to that value and jump to the next node.

Types of Decision Trees

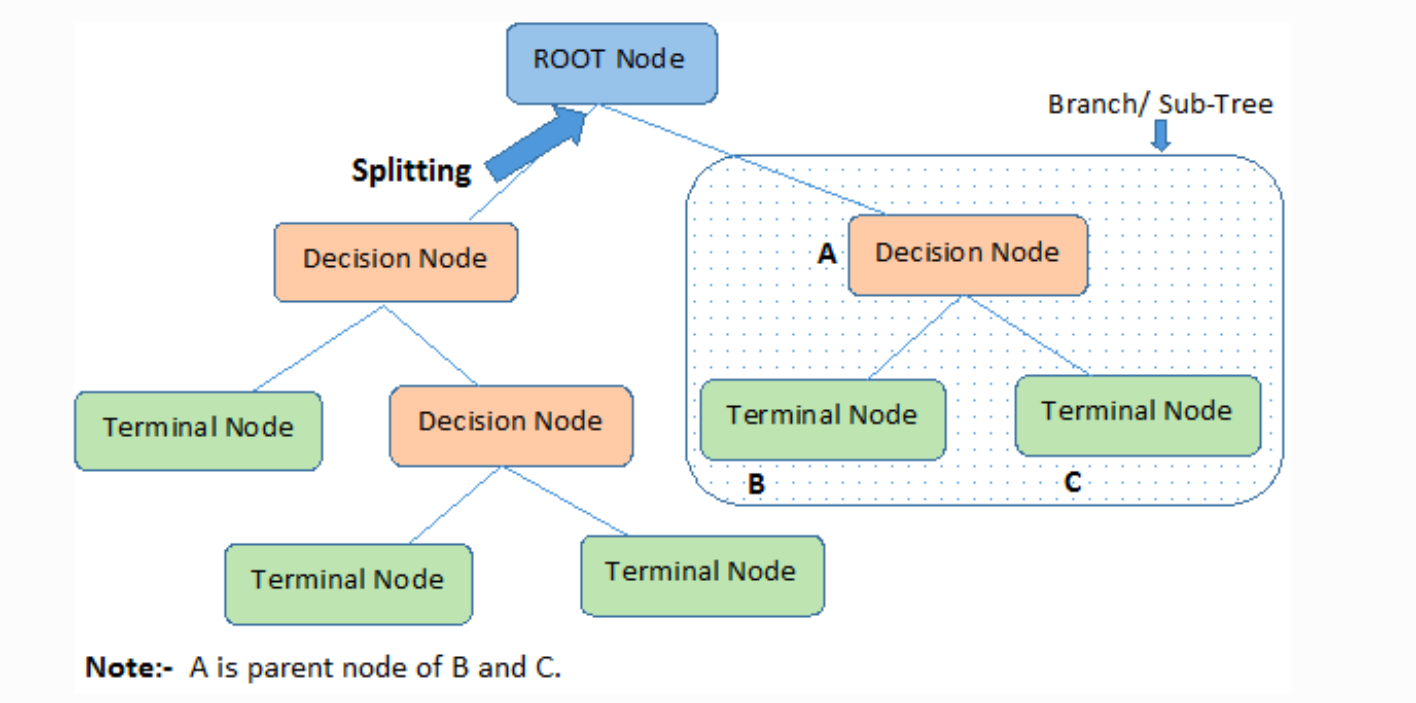
Types of decision trees are based on the type of target variable we have. It can be of two types:

1. **Categorical Variable Decision Tree:** Decision Tree which has a categorical target variable then it called a **Categorical variable decision tree.**
2. **Continuous Variable Decision Tree:** Decision Tree has a continuous target variable then it is called **Continuous Variable Decision Tree.**

Example:- Let's say we have a problem to predict whether a customer will pay his renewal premium with an insurance company (yes/ no). Here we know that the income of customers is a significant variable but the insurance company does not have income details for all customers. Now, as we know this is an important variable, then we can build a decision tree to predict customer income based on occupation, product, and various other variables. In this case, we are predicting values for the continuous variables.

Terminology related to Decision Trees

1. **Root Node:** It represents the 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 the decision node.
4. **Leaf / Terminal Node:** Nodes do not split is called Leaf or Terminal node.
5. **Pruning:** When we remove sub-nodes of a decision node, this process is called pruning. You can say the opposite process of splitting.
6. **Branch / Sub-Tree:** A subsection of the entire tree is called branch or sub-tree.
7. **Parent and Child Node:** A node, which is divided into sub-nodes is called a parent node of sub-nodes whereas sub-nodes are the child of a parent node.



Decision trees classify the examples by sorting them down the tree from the root to some leaf/terminal node, with the leaf/terminal node providing the classification of the example.

Each node in the tree acts as a test case for some attribute, and each edge descending from the node corresponds to the possible answers to the test case. This process is recursive in nature and is repeated for every subtree rooted at the new node.

Assumptions while creating Decision Tree

Below are some of the assumptions we make while using Decision tree:

- In the beginning, the whole training set is considered as the **root.**
- Feature values are preferred to be categorical. If the values are continuous then they are discretized prior to building the model.
- Records are **distributed recursively** on the basis of attribute values.
- Order to placing attributes as root or internal node of the tree is done by using some statistical approach.

Decision Trees follow **Sum of Product (SOP) r**epresentation. The Sum of product (SOP) is also known as **Disjunctive Normal Form**. For a class, every branch from the root of the tree to a leaf node having the same class is conjunction (product) of values, different branches ending in that class form a disjunction (sum).

The primary challenge in the decision tree implementation is to identify which attributes do we need to consider as the root node and each level. Handling this is to know as the attributes selection. We have different attributes selection measures to identify the attribute which can be considered as the root note at each level.

How do Decision Trees work?

The decision of making strategic splits heavily affects a tree's accuracy. The decision criteria are different for classification and regression trees.
Decision trees use multiple algorithms to decide to split a node into two or more sub-nodes. The creation of sub-nodes increases the homogeneity of resultant sub-nodes. In other words, we can say that the purity of the node increases with respect to the target variable. The decision tree splits the nodes on all available variables and then selects the split which results in most homogeneous sub-nodes.

The algorithm selection is also based on the type of target variables. Let us look at some algorithms used in Decision Trees:

ID3 → (extension of D3)
C4.5 → (successor of ID3)
CART → (Classification And Regression Tree)
CHAID → (Chi-square automatic interaction detection Performs multi-level splits when computing classification trees)
MARS → (multivariate adaptive regression splines)
The ID3 algorithm builds decision trees using a top-down greedy search approach through the space of possible branches with no backtracking. A greedy algorithm, as the name suggests, always makes the choice that seems to be the best at that moment.
Steps in ID3 algorithm:

1. It begins with the original set S as the root node.
2. On each iteration of the algorithm, it iterates through the very unused attribute of the set S and calculates **Entropy(H)** and **Information gain(IG)** of this attribute.
3. It then selects the attribute which has the smallest Entropy or Largest Information gain.
4. The set S is then split by the selected attribute to produce a subset of the data.
5. The algorithm continues to recur on each subset, considering only attributes never selected before.

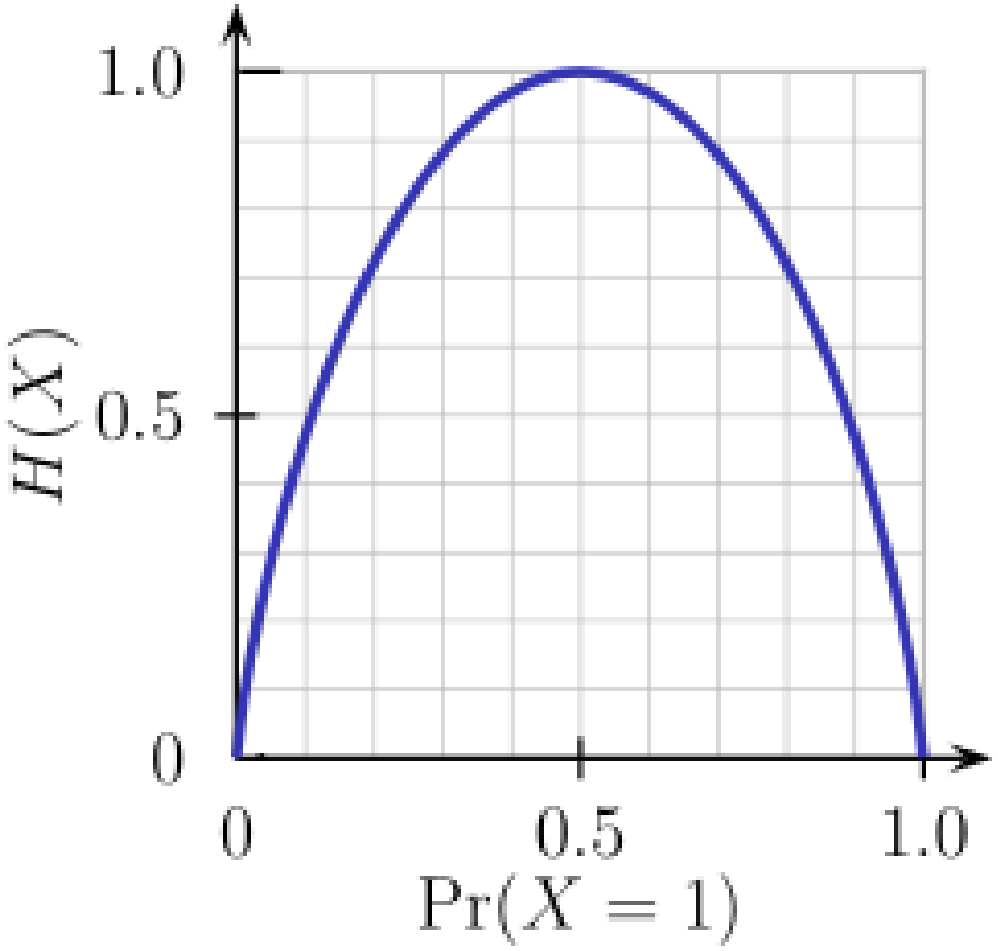
Attribute Selection Measures

If the dataset consists of **N** attributes then deciding which attribute to place at the root or at different levels of the tree as internal nodes is a complicated step. By just randomly selecting any node to be the root can't solve the issue. If we follow a random approach, it may give us bad results with low accuracy.
For solving this attribute selection problem, researchers worked and devised some solutions. They suggested using some *criteria* like :
Entropy,
Information gain,
Gini index,
Gain Ratio,
Reduction in Variance
Chi-Square
These criteria will calculate values for every attribute. The values are sorted, and attributes are placed in the tree by following the order i.e, the attribute with a high value(in case of information

gain) is placed at the root.
While using Information Gain as a criterion, we assume attributes to be categorical, and for the Gini index, attributes are assumed to be continuous.

Entropy

Entropy is a measure of the randomness in the information being processed. The higher the entropy, the harder it is to draw any conclusions from that information. Flipping a coin is an example of an action that provides information that is random.



From the above graph, it is quite evident that the entropy H(X) is zero when the probability is either 0 or 1. The Entropy is maximum when the probability is 0.5 because it projects perfect randomness in the data and there is no chance if perfectly determining the outcome.

ID3 follows the rule — A branch with an entropy of zero is a leaf node and A brach with entropy more than zero needs further splitting.
Mathematically Entropy for 1 attribute is represented as:

$$E(S) = \sum_{i=1}^{c} - p_i \log_2 p_i$$

Play Golf	
Yes	No
9	5

Entropy(PlayGolf) = Entropy (5,9)
= Entropy (0.36, 0.64)
= - (0.36 $\log_2$ 0.36) - (0.64 $\log_2$ 0.64)
= 0.94

Where **S → Current state, and Pi → Probability of an event *i* of state S or Percentage of class *i* in a node of state S.**
Mathematically Entropy for multiple attributes is represented as:

$$E(T, X) = \sum_{c \in X} P(c) E(c)$$

		Play Golf		
		Yes	No	
Outlook	Sunny	3	2	5
	Overcast	4	0	4
	Rainy	2	3	5
				14

E(PlayGolf, Outlook) = **P**(Sunny)*__E__(3,2) + **P**(Overcast)*__E__(4,0) + **P**(Rainy)*__E__(2,3)

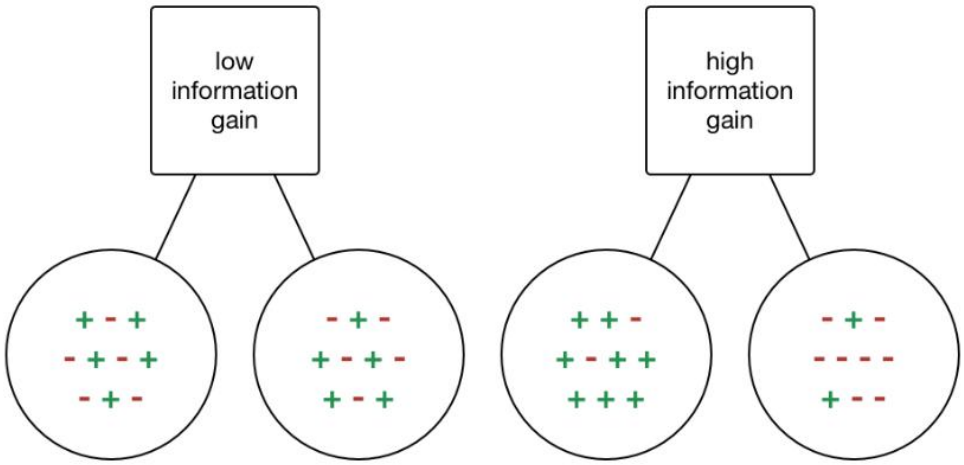
= (5/14)*0.971 + (4/14)*0.0 + (5/14)*0.971

= 0.693

where **T→ Current state and X → Selected attribute**

Information Gain

Information gain or **IG** is a statistical property that measures how well a given attribute separates the training examples according to their target classification. Constructing a decision tree is all about finding an attribute that returns the highest information gain and the smallest entropy.



Information Gain

Information gain is a decrease in entropy. It computes the difference between entropy before split and average entropy after split of the dataset based on given attribute values. ID3 (Iterative Dichotomiser) decision tree algorithm uses information gain.

Mathematically, IG is represented as:

$$\text{Information Gain}(T,X) = \text{Entropy}(T) - \text{Entropy}(T, X)$$

$$\begin{aligned} \text{IG(PlayGolf, Outlook)} &= \text{E(PlayGolf)} - \text{E(PlayGolf, Outlook)} \\ &= 0.940 - 0.693 \\ &= 0.247 \end{aligned}$$

In a much simpler way, we can conclude that:

$$Information\ Gain\ =\ Entropy(before) - \sum_{j=1}^{K} Entropy(j,\ after)$$

Information Gain

Where “before” is the dataset before the split, K is the number of subsets generated by the split, and (j, after) is subset j after the split.

Gini Index

You can understand the Gini index as a cost function used to evaluate splits in the dataset. It is calculated by subtracting the sum of the squared probabilities of each class from one. It favors larger partitions and easy to implement whereas information gain favors smaller partitions with distinct values.

$$Gini = 1 - \sum_{i=1}^{C} (p_i)^2$$

Gini Index

Gini Index works with the categorical target variable “Success” or “Failure”. It performs only Binary splits.

Higher value of Gini index implies higher inequality, higher heterogeneity.

Steps to Calculate Gini index for a split

1. Calculate Gini for sub-nodes, using the above formula for success(p) and failure(q) (p^2+q^2).
2. Calculate the Gini index for split using the weighted Gini score of each node of that split.

CART (Classification and Regression Tree) uses the Gini index method to create split points.

Gain ratio

Information gain is biased towards choosing attributes with a large number of values as root nodes. It means it prefers the attribute with a large number of distinct values.
C4.5, an improvement of ID3, uses Gain ratio which is a modification of Information gain that reduces its bias and is usually the best option. Gain ratio overcomes the problem with information gain by taking into account the number of branches that would result before making the split. It corrects information gain by taking the intrinsic information of a split into account.

Let us consider if we have a dataset that has users and their movie genre preferences based on variables like gender, group of age, rating, blah, blah. With the help of information gain, you split at ‘Gender’ (assuming it has the highest information gain) and now the variables ‘Group of Age’ and ‘Rating’ could be equally important and with the help of gain ratio, it will penalize a variable with more distinct values which will help us decide the split at the next level.

$$Gain\ Ratio = \frac{Information\ Gain}{SplitInfo} = \frac{Entropy\ (before) - \sum_{j=1}^{K} Entropy(j, after)}{\sum_{j=1}^{K} w_j\ log_2\ w_j}$$

Gain Ratio

Where "before" is the dataset before the split, K is the number of subsets generated by the split, and (j, after) is subset j after the split.

Reduction in Variance

Reduction in variance is an algorithm used for continuous target variables (regression problems). This algorithm uses the standard formula of variance to choose the best split. The split with lower variance is selected as the criteria to split the population:

$$Variance = \frac{\Sigma(X - \overline{X})^2}{n}$$

Above X-bar is the mean of the values, X is actual and n is the number of values.

Steps to calculate Variance:

1. Calculate variance for each node.
2. Calculate variance for each split as the weighted average of each node variance.

Chi-Square

The acronym CHAID stands for *Chi*-squared Automatic Interaction Detector. It is one of the oldest tree classification methods. It finds out the statistical significance between the differences between sub-nodes and parent node. We measure it by the sum of squares of standardized differences between observed and expected frequencies of the target variable.
It works with the categorical target variable "Success" or "Failure". It can perform two or more splits. Higher the value of Chi-Square higher the statistical significance of differences between sub-node and Parent node.

It generates a tree called CHAID (Chi-square Automatic Interaction Detector).

Mathematically, Chi-squared is represented as:

$$\chi^2 = \sum \frac{(O-E)^2}{E}$$

Where:

χ^2 = Chi Square obtained
$\sum$ = the sum of
O = observed score
E = expected score

Steps to Calculate Chi-square for a split:

1. Calculate Chi-square for an individual node by calculating the deviation for Success and Failure both
2. Calculated Chi-square of Split using Sum of all Chi-square of success and Failure of each node of the split

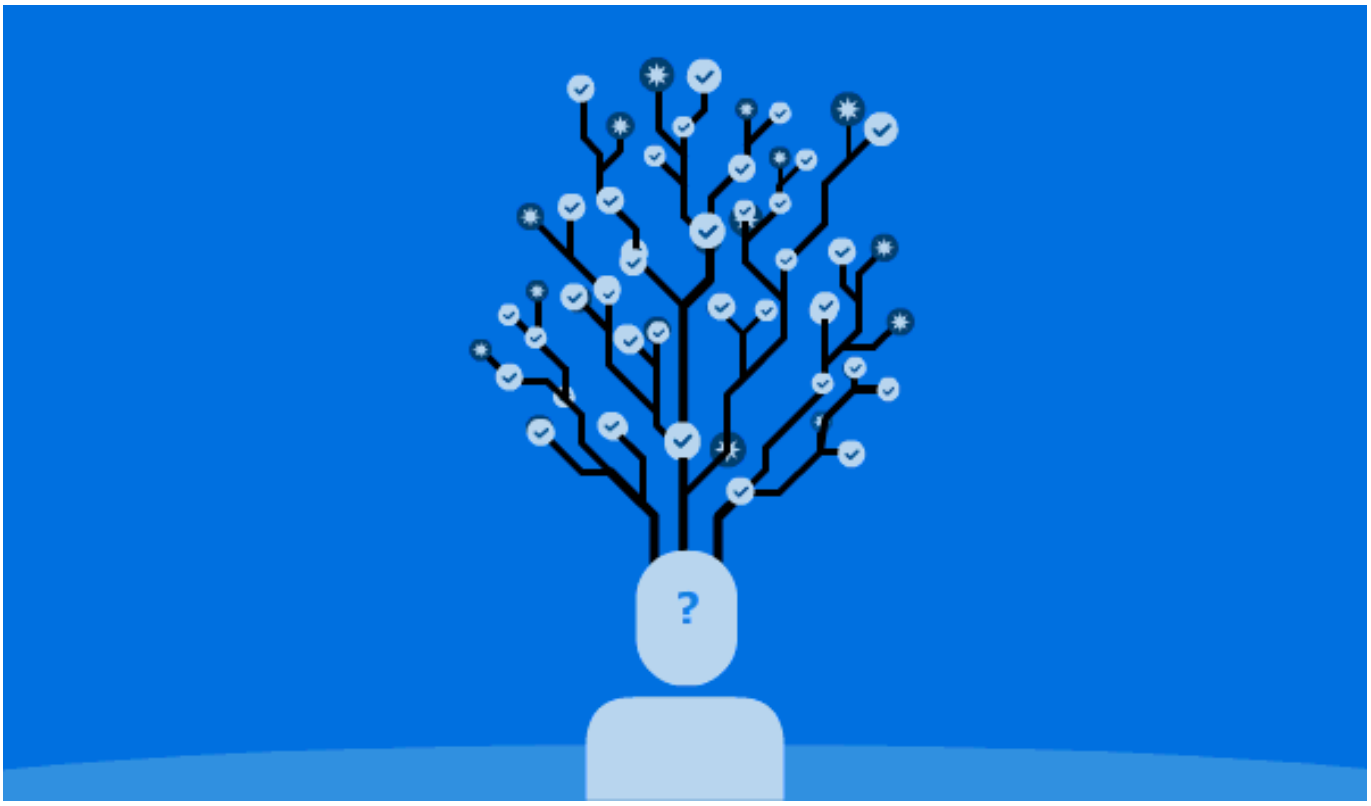
How to avoid/counter Overfitting in Decision Trees?

The common problem with Decision trees, especially having a table full of columns, they fit a lot. Sometimes it looks like the tree memorized the training data set. If there is no limit set on a decision tree, it will give you 100% accuracy on the training data set because in the worse case it will end up making 1 leaf for each observation. Thus this affects the accuracy when predicting samples that are not part of the training set.
Here are two ways to remove overfitting:

1. Pruning Decision Trees.
2. Random Forest

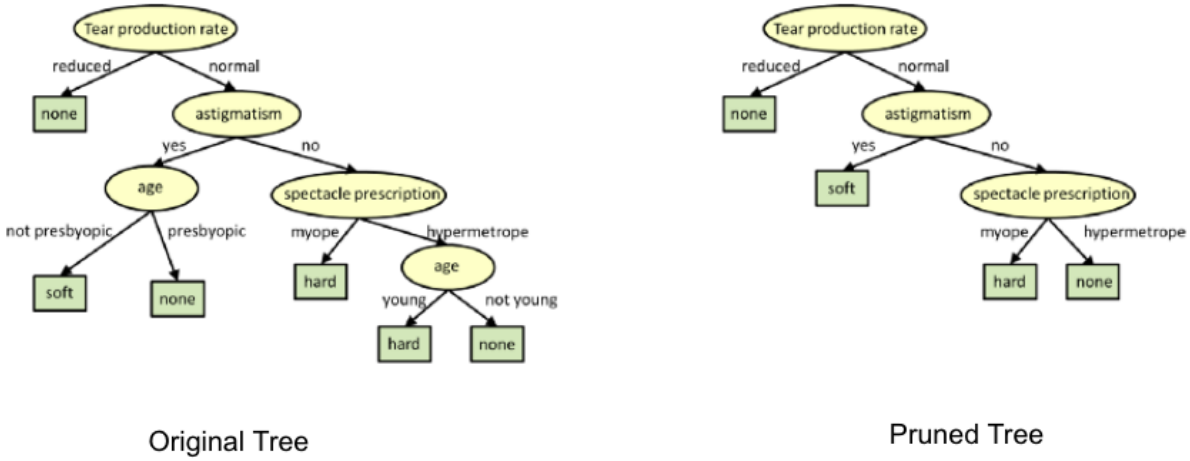
Pruning Decision Trees

The splitting process results in fully grown trees until the stopping criteria are reached. But, the fully grown tree is likely to overfit the data, leading to poor accuracy on unseen data.



Pruning in action

In **pruning**, you trim off the branches of the tree, i.e., remove the decision nodes starting from the leaf node such that the overall accuracy is not disturbed. This is done by segregating the actual training set into two sets: training data set, D and validation data set, V. Prepare the decision tree using the segregated training data set, D. Then continue trimming the tree accordingly to optimize the accuracy of the validation data set, V.



Pruning

In the above diagram, the 'Age' attribute in the left-hand side of the tree has been pruned as it has more importance on the right-hand side of the tree, hence removing overfitting.

Random Forest

Random Forest is an example of ensemble learning, in which we combine multiple machine learning algorithms to obtain better predictive performance.

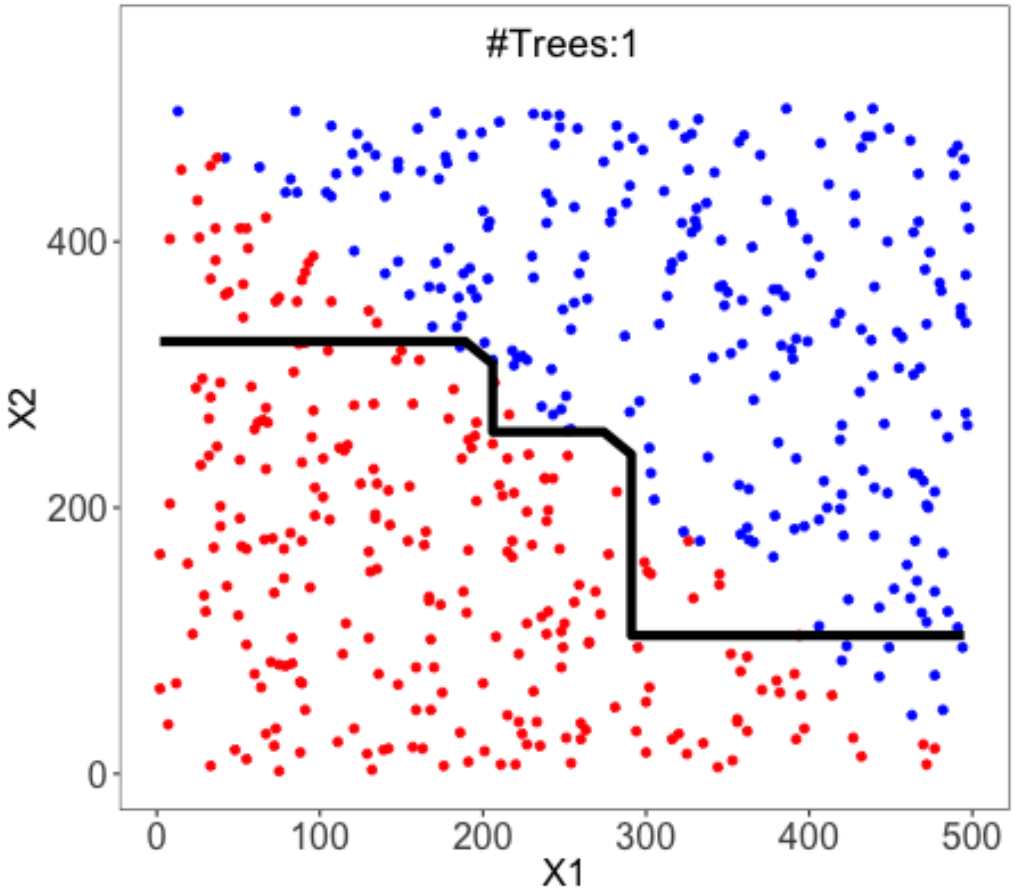
Why the name "Random"?

Two key concepts that give it the name random:

1. A random sampling of training data set when building trees.
2. Random subsets of features considered when splitting nodes.

A technique known as bagging is used to create an ensemble of trees where multiple training sets are generated with replacement.

In the bagging technique, a data set is divided into **N** samples using randomized sampling. Then, using a single learning algorithm a model is built on all samples. Later, the resultant predictions are combined using voting or averaging in parallel.



Random Forest in action

Which is better Linear or tree-based models?

Well, it depends on the kind of problem you are solving.

1. If the relationship between dependent & independent variables is well approximated by a linear model, linear regression will outperform the tree-based model.
2. If there is a high non-linearity & complex relationship between dependent & independent variables, a tree model will outperform a classical regression method.
3. If you need to build a model that is easy to explain to people, a decision tree model will always do better than a linear model. Decision tree models are even simpler to interpret than linear regression!