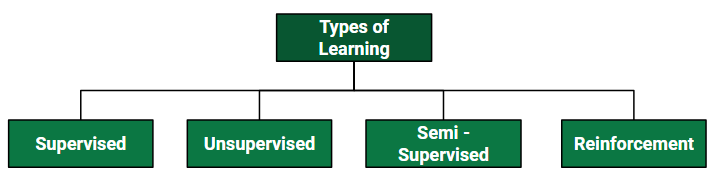
**What is Learning for a machine?**

A machine is said to be learning from **past Experiences**(data feed in) with respect to some class of **Tasks**, if it’s **Performance** in a given Task improves with the Experience.



|  |  |
| --- | --- |
| **Supervised** | |
| **Regression** | **Classification** |
| Linear Regression | Logistic Regression |
| Polynomial Regression | K-NN |
| Support Vector Regression (SVR) | Support Vector Machine(SVM) |
| Decision Tree Regression | Naïve Bayes |
| Random Forest Regression | Decision Tree Classification |
|  | Random Forest |

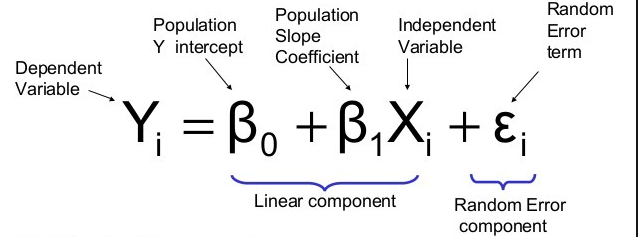
|  |  |
| --- | --- |
| **Unsupervised** | |
| **Clustering** | **Association** |
| k-means | Apriori |

ML | Linear Regression

**Linear Regression** is a machine learning algorithm based on **supervised learning**. It performs a **regression task**. Regression models a target prediction value based on independent variables. It is mostly used for finding out the relationship between variables and forecasting. Different regression models differ based on – the kind of relationship between dependent and independent variables, they are considering and the number of independent variables being used.  


Linear regression performs the task to predict a dependent variable value (y) based on a given independent variable (x). So, this regression technique finds out a linear relationship between x (input) and y(output). Hence, the name is Linear Regression.  
In the figure above, X (input) is the work experience and Y (output) is the salary of a person. The regression line is the best fit line for our model.

**Hypothesis function for Linear Regression :**



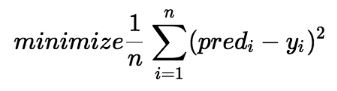
While training the model we are given :

Here, the β1 it’s are the parameters (also called weights) βo is the y-intercept and Єi is the random error term whose role is to add bias. The above equation is the linear equation that needs to be obtained with the minimum error. To check the error we have to calculate the sum of squared error and tune the parameters to try to reduce the error.

Key:  
1. Y(predicted) is also called the hypothesis function.  
2. J(θ) is the cost function which can also be called the error function. Our main goal is to minimize the value of the cost.  
3. y(i) is the predicted output.  
4. hθ(x(i)) is called the hypothesis function which is basically the Y(predicted) value

**How to update θ1 and θ2 values to get the best fit line ?**

**Cost Function (J):**  
By achieving the best-fit regression line, the model aims to predict y value such that the error difference between predicted value and true value is minimum. So, it is very important to update the b0 and b1 values, to reach the best value that minimize the error between predicted y value (pred) and true y value (y).





Cost function(J) of Linear Regression is the **Root Mean Squared Error (RMSE)** between predicted y value (pred) and true y value (y).

[**Gradient Descent**](https://www.geeksforgeeks.org/gradient-descent-in-linear-regression/)**:**

To update θ1 and θ2 values in order to reduce Cost function (minimizing RMSE value) and achieving the best fit line the model uses Gradient Descent. The idea is to start with random θ1 and θ2 values and then iteratively updating the values, reaching minimum cost.

# Polynomial Regression

**Polynomial Regression**is a form of linear regression in which the relationship between the independent variable x and dependent variable y is modeled as an nth degree polynomial

**y** = a + b1x + b2^2 + e

Here y is dependent variable, a is y intercept, b is the slope and e is the error rate.

## Support Vector Machine?

The objective of the support vector machine algorithm is to find a hyperplane in an N-dimensional space(N — the number of features) that distinctly classifies the data points.

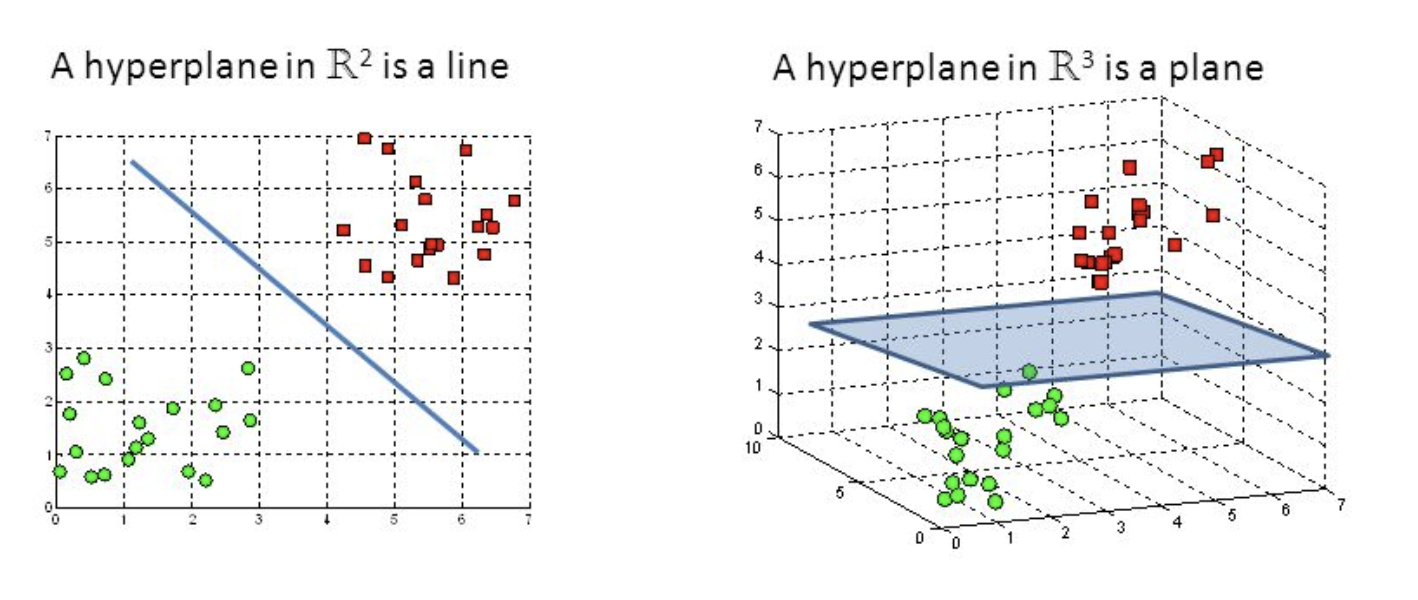


Support Vector Machine” (SVM) is a supervised [machine learning algorithm](https://courses.analyticsvidhya.com/courses/introduction-to-data-science-2?utm_source=blog&utm_medium=understandingsupportvectormachinearticle) which can be used for both classification or regression challenges. However,  it is mostly used in classification problems. In the SVM algorithm, we plot each data item as a point in n-dimensional space (where n is number of features you have) with the value of each feature being the value of a particular coordinate. Then, we perform classification by finding the hyper-plane that differentiates the two classes very well (look at the below snapshot).

[](https://www.analyticsvidhya.com/wp-content/uploads/2015/10/SVM_1.png)

How it works:

To separate the two classes of data points, there are many possible hyperplanes that could be chosen. Our objective is to find a plane that has the maximum margin, i.e the maximum distance between data points of both classes. Maximizing the margin distance provides some reinforcement so that future data points can be classified with more confidence.



Hyperplanes are decision boundaries that help classify the data points. Data points falling on either side of the hyperplane can be attributed to different classes. Also, the dimension of the hyperplane depends upon the number of features. If the number of input features is 2, then the hyperplane is just a line. If the number of input features is 3, then the hyperplane becomes a two-dimensional plane. It becomes difficult to imagine when the number of features exceeds 3.



# Decision Tree Regressor explained in depth

Decision Tree can be used both in classification and regression problem

Decision Trees can be summarized with the below bullet points:

* Decision trees are predictive models that use a set of binary rules to calculate a target value.
* Each individual tree is a fairly simple model that has branches, nodes and leaves.

## **Important Terminology**

**Root Node:**It represents entire population or sample and this further gets divided into two or more homogeneous sets.

**Splitting:**It is a process of dividing a node into two or more sub-nodes

**Decision Node:**When a sub-node splits into further sub-nodes, then it is called decision node.

**Leaf/Terminal Node:**Nodes do not split is called Leaf or Terminal node.

**Pruning:**When we remove sub-nodes of a decision node, this process is called pruning. You can say opposite process of splitting.

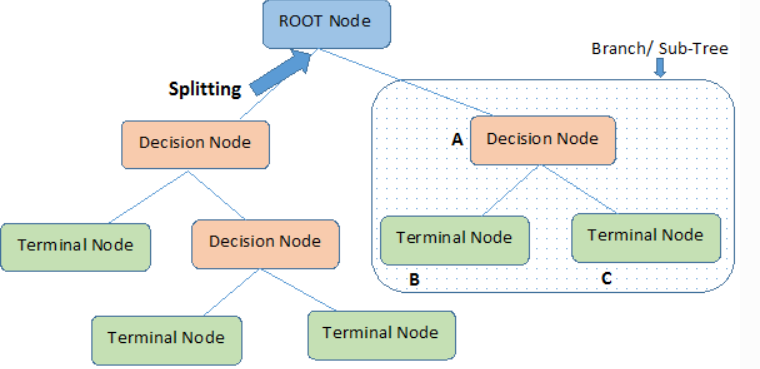
Pruning reduces the size of decision trees by removing parts of the tree that do not provide power to classify instances.

(How: Minimal cost complexity pruning recursively finds the node with the “weakest link”. The weakest link is characterized by an effective alpha, where the nodes with the smallest effective alpha are pruned first.

<https://scikit-learn.org/stable/auto_examples/tree/plot_cost_complexity_pruning.html>

 )

**Branch / Sub-Tree:**A sub section of entire tree is called branch or sub-tre



## **How is Splitting Decided for Decision Trees?**

The decision of making strategic splits heavily affects a tree’s accuracy. The decision criteria is different for classification and regression trees. Decision trees regression normally use [mean squared error (MSE)](https://en.wikipedia.org/wiki/Mean_squared_error) to decide to split a node in two or more sub-nodes. Split the data into two subset. For each subset, it will calculate the MSE separately. The tree chooses the value with results in smallest MSE value.

Note: For k classes there are 2k – 1 – 1 splits, which is computationally prohibitive if k is a large number.

# Random Forest Regressor

Random Forest is a flexible, easy to use machine learning algorithm that produces great results most of the time with minimum time spent on hyper-parameter tuning. It has gained popularity due to its simplicity and the fact that it can be used for both classification and regression tasks**.**

Random Forest is an ensemble machine learning technique capable of performing both regression and classification tasks using multiple decision trees and a statistical technique called **bagging.**Bagging along with boosting aretwo of the most popular ensemble techniques which aim to tackle **high variance** and **high bias**.

Random forest builds multiple decision trees and merge their predictions together to get a more **accurate** and **stable** prediction rather than relying on individual decision trees.

## **Random sampling of training observations**

Each tree in a random forest learns from a **random** sample of the training observations. The samples are drawn with replacement, known as bootstrapping, which means that some samples will be used multiple times in a single tree. The idea is that by training each tree on different samples, although each tree might have high variance with respect to a particular set of the training data, overall, the entire forest will have lower variance but not at the cost of increasing the bias

## Why a Random Forest is better than a single decision tree?

The fundamental idea behind a random forest is to combine the predictions made by many decision trees into a single model. Individually, predictions made by decision trees may not be accurate but combined together, the predictions will be closer to the true value on average.

## **Why a Random Forest reduces overfitting**

The objective of a machine learning model is to generalize well to **new data**it has never seen before. Overfitting occurs when a very flexible model (high capacity) **memorizes** the training data by fitting it closely. The problem is that the model learns not only the actual relationships in the training data but **also any noise** that is present. A flexible model is said to have high ***variance***because the learned parameters (such as the structure of the decision tree) will vary considerably with the training data.

On the other hand, an inflexible model is said to have high ***bias*** because it makes **assumptions** about the training data (it’s biased towards pre-conceived ideas of the data). An inflexible model may not have the capacity to fit even the training data and **in both cases — high variance and high bias** — the model is not able to generalize well to new data.

## **How does it work?**

**Step 1:** Samples are taken repeatedly from the training data so that each data point is having an equal probability of getting selected, and all the samples have the same size as the original training set.

Let's say we have the following data:

x= 0.1,0.5,0.4,0.8,0.6, y=0.1,0.2,0.15,0.11,0.13 where x is an independent variable with 5 data points and y is dependent variable.

Now Bootstrap samples are taken with replacement from the above data set. **n\_estimators**is set to 3 (no of tree in random forest), then:

The first tree will have a bootstrap sample of size 5 (same as the original dataset), assuming it to be: x1={0.5,0.1,0.1,0.6,0.6} likewise

x2={0.4,0.8,0.6,0.8,0.1}

x3={0.1,0.5,0.4,0.8,0.8}

**Step 2:**A Random Forest Regressor model is trained at each bootstrap sample drawn in the above step, and a prediction is recorded for each sample.

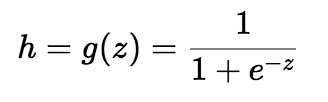
**Step 3:** Now the ensemble prediction is calculated by averaging the predictions of the above trees producing the final prediction

**Classification**

# Logistic Regression

Logistic regression is basically a supervised classification algorithm. In a classification problem, the target variable(or output), y, can take only discrete values for given set of features(or inputs), X.

The model builds a regression model to predict the probability that a given data entry belongs to the category numbered as “1”. Just like Linear regression assumes that the data follows a linear function, Logistic regression models the data using the **sigmoid function**.





Logistic regression becomes a classification technique only when a decision threshold is brought into the picture. The setting of the threshold value is a very important aspect of Logistic regression and is dependent on the classification problem itself.

The decision for the value of the threshold value is majorly affected by the values of [precision and recall.](https://www.geeksforgeeks.org/confusion-matrix-machine-learning/)

Based on the number of categories, Logistic regression can be classified as:

1. **binomial:** target variable can have only 2 possible types: “0” or “1” which may represent “win” vs “loss”, “pass” vs “fail”, “dead” vs “alive”, etc.
2. **multinomial:** target variable
3. can have 3 or more possible types which are not ordered(i.e. types have no quantitative significance) like “disease A” vs “disease B” vs “disease C”.
4. **ordinal:** it deals with target variables with ordered categories. For example, a test score can be categorized as:“very poor”, “poor”, “good”, “very good”. Here, each category can be given a score like 0, 1, 2, 3.

**Confusion Matrix:**

A confusion matrix is a summary of prediction results on a classification problem. The number of correct and incorrect predictions are summarized with count values and broken down by each class. This is the key to the confusion matrix. The confusion matrix shows the ways in which your classification model is confused when it makes predictions

**Performance Matrix:**

<https://medium.com/thalus-ai/performance-metrics-for-classification-problems-in-machine-learning-part-i-b085d432082b>



Here,

* Class 1 : Positive
* Class 2 : Negative

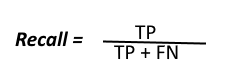
**Definition of the Terms:**

* Positive (P) : Observation is positive (for example: is an apple).
* Negative (N) : Observation is not positive (for example: is not an apple).
* True Positive (TP) : Observation is positive, and is predicted to be positive.
* False Negative (FN) : Observation is positive, but is predicted negative.
* True Negative (TN) : Observation is negative, and is predicted to be negative.
* False Positive (FP) : Observation is negative, but is predicted positive.

**Classification Rate/Accuracy:**



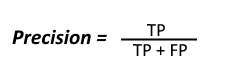
**Recall:**



A recall is a ratio of the true positive rate against the actual positive rate. It ranges from 0 to 1.

The precise definition of **recall** is the number of true positives divided by the number of true positives plus the number of false negatives.

**Precision:**



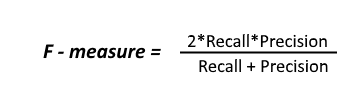
Precision is the most commonly used error metric is n classification mechanism. Its range is from 0 to 1, where 1 represents 100%

To get the value of precision we divide the total number of correctly classified positive examples by the total number of predicted positive examples. High Precision indicates an example labelled as positive is indeed positive (a small number of FP).

**9**: <https://towardsdatascience.com/beyond-accuracy-precision-and-recall-3da06bea9f6c>

**High recall, low precision:**This means that most of the positive examples are correctly recognized (low FN) but there are a lot of false positives.

**Low recall, high precision:**This shows that we miss a lot of positive examples (high FN) but those we predict as positive are indeed positive (low FP)

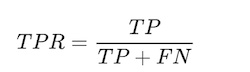
**F-measure:**  
Since we have two measures (Precision and Recall) it helps to have a measurement that represents both of them. We calculate an F-measure which uses Harmonic Mean in place of Arithmetic Mean as it punishes the extreme values more.  
The F-Measure will always be nearer to the smaller value of Precision or Recall.  


### ROC curve:

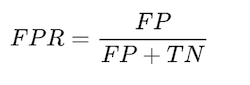
An ROC curve(receiver operating characteristic curve) graph shows the performance of a classification model at all classification thresholds.

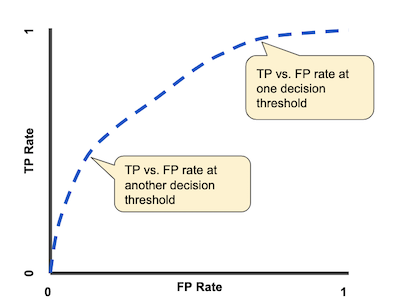
It plots 2 parameters:

* **True positive rate** (Recall)



**. False Positive rate**





### AUC:

**AUC** stands for *Area under the ROC Curve.*It provides an aggregate measure of performance across all possible classification thresholds.

The higher the **area under the ROC curve**(AUC), the better the classifier. A perfect classifier would have an AUC of 1.

### **K Nearest Neighbors - Classification**

K nearest neighbors is a simple algorithm that stores all available cases and classifies new cases based on a similarity measure (e.g., distance functions). KNN has been used in statistical estimation and pattern recognition

|  |  |  |
| --- | --- | --- |
| **Algorithm** |  |  |
| A case is classified by a majority vote of its neighbors, with the case being assigned to the class most common amongst its K nearest neighbors measured by a distance function. If K = 1, then the case is simply assigned to the class of its nearest neighbor. |  |  |
|  |  |  |



|  |  |  |
| --- | --- | --- |
| It should also be noted that all three distance measures are only valid for continuous variables. In the instance of categorical variables the Hamming distance must be used. It also brings up the issue of standardization of the numerical variables between 0 and 1 when there is a mixture of numerical and categorical variables in the dataset. |  |  |
|  |  |  |

### **Naive Bayesian**

The Naive Bayesian classifier is based on Bayes’ theorem with the independence assumptions between predictors

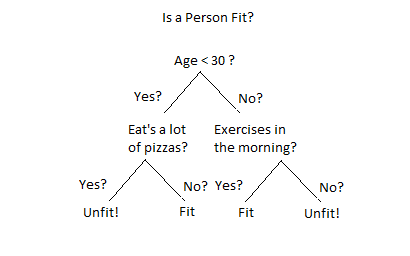
The Naive Bayes Algorithm model is based on the Bayes Theorem. It describes the probability of an event. It is based on prior knowledge of conditions which might be related to that specific event.

|  |  |  |
| --- | --- | --- |
| **Algorithm** |  |  |
| Bayes theorem provides a way of calculating the posterior probability, *P*(*c|x*), from *P*(*c*), *P*(*x*), and *P*(*x|c*). Naive Bayes classifier assume that the effect of the value of a predictor (*x*) on a given class (*c*) is independent of the values of other predictors. This assumption is called class conditional independence. |  |  |
|  |  |  |
| * *P*(*c|x*) is the posterior probability of *class* (*target*) given *predictor* (*attribute*). * *P*(*c*) is the prior probability of *class*. * *P*(*x|c*) is the likelihood which is the probability of *predictor* given *class*. * *P*(*x*) is the prior probability of *predictor*. |  |  |
| *Example 1:* |  |  |
| *We use the same simple Weather dataset here.* |  |  |
|  |  |  |
| *The posterior probability can be calculated by first, constructing a frequency table for each attribute against the target. Then, transforming the frequency tables to likelihood tables and finally use the Naive Bayesian equation to calculate the posterior probability for each class. The class with the highest posterior probability is the outcome of prediction.* |  |  |
|  |  |  |

# Decision Trees Explained Easily

Decision Trees (DTs) are a non-parametric supervised learning method used for classification and regression.

Decision tree builds classification or regression models in the form of a tree structure. It breaks down a data set into smaller and smaller subsets while at the same time an associated decision tree is incrementally developed. The final result is a tree with decision nodes and leaf nodes. A decision node has two or more branches. Leaf node represents a classification or decision. The topmost decision node in a tree which corresponds to the best predictor called root node. Decision trees can handle both categorical and numerical data.

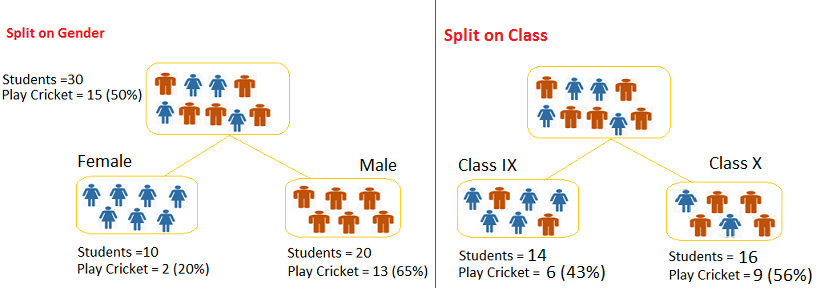


# How Do Decision Trees Work?

There are several steps involved in the building of a decision tree

# Splitting

The process of partitioning the data set into subsets. Splits are formed on a particular variable



# Pruning:

# The shortening of branches of the tree. Pruning is the process of reducing the size of the tree by turning some branch nodes into leaf nodes, and removing the leaf nodes under the original branch. Pruning is useful because classification trees may fit the training data well, but may do a poor job of classifying new values. A simpler tree often avoids over-fitting

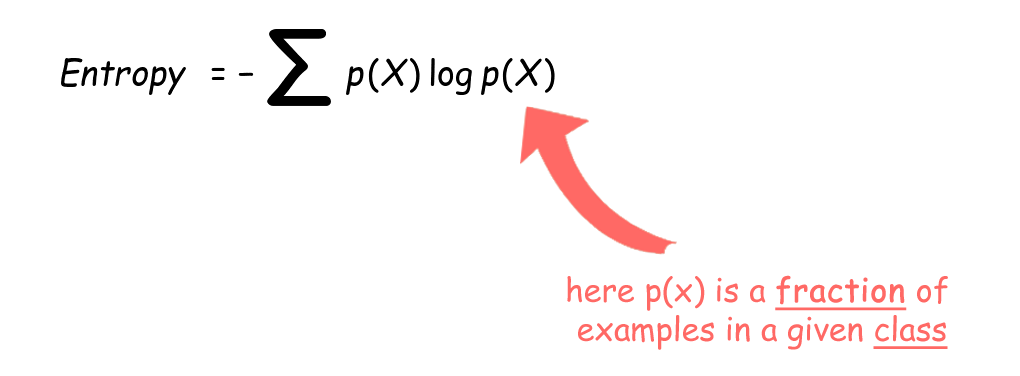
# Tree Selection

# The process of finding the smallest tree that fits the data. Usually this is the tree that yields the lowest cross-validated error.

# ****Key Factors****:

# 1. Entropy:

Entropy controls how a Decision Tree decides to **split**the data. It actually effects how a **Decision Tree** draws its boundaries



A decision tree is built top-down from a root node and involves partitioning the data into subsets that contain instances with similar values (homogeneous). ID 3 algorithm uses entropy to calculate the homogeneity of a sample. If the sample is completely homogeneous the entropy is zero and if the sample is an equally divided it has entropy of one.

# 2. Information Gain:

# The information gain is based on the decrease in entropy after a dataset is split on an attribute. Constructing a decision tree is all about finding attribute that returns the highest information gain (i.e., the most homogeneous branches).

# *Steps Involved:*

# Step 1:

# Calculate entropy of the target.

# Step 2:

# The dataset is then split on the different attributes. The entropy for each branch is calculated. Then it is added proportionally, to get total entropy for the split. The resulting entropy is subtracted from the entropy before the split. The result is the Information Gain, or decrease in entropy.

# Step 3

# Choose attribute with the largest information gain as the decision node, divide the dataset by its branches and repeat the same process on every branch.

# Pros:

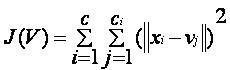
Computionally cheap to use, easy for humans to understand results and it can deal with irrelevant feaures also

# Cons:

Prone to Overfitting.(It refers to the process when models is trained on training data too well that any noise in testing data can bring negative impacts to performance of model.)

### k-means clustering algorithm:

k-means is  one of  the simplest unsupervised  learning  algorithms  that  solve  the well  known clustering problem. The procedure follows a simple and  easy  way  to classify a given data set  through a certain number of  clusters (assume k clusters) fixed apriori. The  main  idea  is to define k centers, one for each cluster. These centers  should  be placed in a cunning  way  because of  different  location  causes different  result. So, the better  choice  is  to place them  as  much as possible  far away from each other. The  next  step is to take each point belonging  to a  given data set and associate it to the nearest center. When no point  is  pending,  the first step is completed and an early group age  is done. At this point we need to re-calculate k new centroids as barycenter of  the clusters resulting from the previous step. After we have these k new centroids, a new binding has to be done  between  the same data set points  and  the nearest new centre. A loop has been generated. As a result of  this loop we  may  notice that the k centers change their location step by step until no more changes  are done or  in  other words centers do not move any more. Finally, this  algorithm  aims at  minimizing  an objective function know as squared error function given by:



where,  
                           *‘||xi- vj||’* is the Euclidean distance between *xi* and *vj.*

*‘ci’* is the number of data points in *ith* cluster.

*‘c’* is the number of cluster centers.

**Algorithmic steps for k-means clustering:**

Let  X = {x1,x2,x3,……..,xn} be the set of data points and V = {v1,v2,…….,vc} be the set of centers.

1) Randomly select *‘c’* cluster centers.

2) Calculate the distance between each data point and cluster centers.

3) Assign the data point to the cluster center whose distance from the cluster center is minimum of all the cluster centers..

4) Recalculate the new cluster center using:

5) Recalculate the distance between each data point and new obtained cluster centers.

6) If no data point was reassigned then stop, otherwise repeat from step 3).

**Advantages**

1) Fast, robust and easier to understand.

2) Relatively efficient: O(tknd), where n is # objects, k is # clusters, d is # dimension of each object, and t  is # iterations. Normally, k, t, d << n.

3) Gives best result when data set are distinct or well separated from each other.

**Disadvantages**

1) The learning algorithm requires apriori specification of the number of  cluster centers.

2) The use of  Exclusive Assignment - If  there are two highly overlapping data then k-means will not be able to resolve       that there are two clusters.

4) Euclidean distance measures can unequally weight underlying factors.

5) The learning algorithm provides the local optima of the squared error function.

6) Randomly choosing of the cluster center cannot lead us to the fruitful result. Pl. refer [Fig](https://sites.google.com/site/dataclusteringalgorithms/k-means-clustering-algorithm/k-means_initial_cluster_selection).

7) Applicable only when mean is defined i.e. fails for categorical data.

8) Unable to handle noisy data and outliers*.*

9) Algorithm fails for non-linear data set.

# K-Nearest Neighbor(KNN):

* K-Nearest Neighbour is one of the simplest Machine Learning algorithms based on Supervised Learning technique
* K-NN algorithm stores all the available data and classifies a new data point based on the similarity. This means when new data appears then it can be easily classified into a well suite category by using K- NN algorithm.
* K-NN algorithm can be used for Regression as well as for Classification but mostly it is used for the Classification problems.
* K-NN is a **non-parametric algorithm**, which means it does not make any assumption on underlying data.

It is also called a **lazy learner algorithm** because it does not learn from the training set immediately instead it stores the dataset and at the time of classification, it performs an action on the dataset.

## **How does K-NN work?**

The K-NN working can be explained on the basis of the below algorithm:

* **Step-1:** Select the number K of the neighbors
* **Step-2:** Calculate the Euclidean distance of **K number of neighbors**
* **Step-3:** Take the K nearest neighbors as per the calculated Euclidean distance.
* **Step-4:** Among these k neighbors, count the number of the data points in each category.
* **Step-5:** Assign the new data points to that category for which the number of the neighbor is maximum.
* **Step-6:** Our model is ready.

## **How to select the value of K in the K-NN Algorithm?**

* There is no particular way to determine the best value for "K", so we need to try some values to find the best out of them. The most preferred value for K is 5.
* A very low value for K such as K=1 or K=2, can be noisy and lead to the effects of outliers in the model.
* Large values for K are good, but it may find some difficulties.

**Note: we can use L graph**

**Advantages:**

* It is simple to implement.
* It is robust to the noisy training data
* It can be more effective if the training data is large.

**Disadvantages:**

* Always needs to determine the value of K which may be complex some time.
* The computation cost is high because of calculating the distance between the data points for all the training samples.

**Short Course**

**Name three types of biases that can occur during sampling**

* Selection bias
* Under coverage bias
* Survivorship bias

Selection Bias occurs when there is no specific randomization achieved while picking individuals or groups or data to be analysed

**What is bias?**

Bias is an error introduced in your model because of the oversimplification of a machine learning algorithm." It can lead to underfitting.

**What is Ensemble Learning?**

The ensemble is a method of combining a diverse set of learners together to improvise on the stability and predictive power of the model. Two types of Ensemble learning methods are:

Bagging

Bagging method helps you to implement similar learners on small sample populations. It helps you to make nearer predictions.

Boosting

Boosting is an iterative method which allows you to adjust the weight of an observation depends upon the last classification. Boosting decreases the bias error and helps you to build strong predictive models.

**Define the term cross-validation**

Cross-validation is a validation technique for evaluating how the outcomes of statistical analysis will generalize for an Independent dataset. This method is used in backgrounds where the objective is forecast, and one needs to estimate how accurately a model will accomplish.

**What is Back Propagation?**

Back-propagation is the essence of neural net training. It is the method of tuning the weights of a neural net depend upon the error rate obtained in the previous epoch. Proper tuning of the helps you to reduce error rates and to make the model reliable by increasing its generalization.

**Ex**

**plain p-value**

When you conduct a hypothesis test in statistics, a p-value allows you to determine the strength of your results. It is a numerical number between 0 and 1. Based on the value it will help you to denote the strength of the specific result

**What is Normal Distribution**

A normal distribution is a set of a continuous variable spread across a normal curve or in the shape of a bell curve. You can consider it as a continuous probability distribution which is useful in statistics. It is useful to analyze the variables and their relationships when we are using the normal distribution curve.

Normal distribution equally distributed as such the mean, median and mode are equal

**What is skewed Distribution & uniform distribution?**

Skewed distribution occurs when if data is distributed on any one side of the plot whereas uniform distribution is identified when the data is spread is equal in the range.

**When underfitting occurs in a static model?**

Underfitting occurs when a statistical model or machine learning algorithm not able to capture the underlying trend of the data.

**State the difference between a Validation Set and a Test Set.**

A Validation set mostly considered as a part of the training set as it is used for parameter selection which helps you to avoid overfitting of the model being built.

While a Test Set is used for testing or evaluating the performance of a trained machine learning model.

**While working on a data set, how can you select important variables? Explain**

Following methods of variable selection you can use:

* Remove the correlated variables before selecting important variables
* Use linear regression and select variables which depend on that p values.
* Use Backward, Forward Selection, and Stepwise Selection
* Use Xgboost, Random Forest, and plot variable importance chart.
* Measure information gain for the given set of features and select top n features accordingly.

**Is it possible to capture the correlation between continuous and categorical variable?**

Yes, we can use analysis of covariance technique to capture the association between continuous and categorical variables.

**Treating a categorical variable as a continuous variable would result in a better predictive model?**

Yes, the categorical value should be considered as a continuous variable only when the variable is ordinal in nature. So it is a better predictive model.

### **How can you avoid the overfitting your model?**

Overfitting refers to a model that is only set for a very small amount of data and ignores the bigger picture. There are three main methods to avoid overfitting:

1. Keep the model simple—take fewer variables into account, thereby removing some of the noise in the training data
2. Use cross-validation techniques, such as k folds cross-validation
3. Use regularization techniques, such as LASSO, that penalize certain model parameters if they're likely to cause overfitting

### **Differentiate between univariate, bivariate, and multivariate analysis**

**Univariate**

Univariate data contains only one variable. The purpose of the univariate analysis is to describe the data and find patterns that exist within it.

Example: height of students

**Bivariate**

Bivariate data involves two different variables. The analysis of this type of data deals with causes and relationships and the analysis is done to determine the relationship between the two variables.

Example: temperature and ice cream sales in the summer season

**Multivariate**

Multivariate data involves three or more variables, it is categorized under multivariate. It is similar to a bivariate, but contains more than one dependent variable.

Example: data for house price prediction

### **What are the feature selection methods used to select the right variables?**

There are two main methods for feature selection:

**Filter Methods**

This involves:

* Linear discrimination analysis
* ANOVA
* Chi-Square

The best analogy for selecting features is "bad data in, bad answer out." When we're limiting or selecting the features, it's all about cleaning up the data coming in.

**Wrapper Methods**

This involves:

* Forward Selection: We test one feature at a time and keep adding them until we get a good fit
* Backward Selection: We test all the features and start removing them to see what works better
* Recursive Feature Elimination: Recursively looks through all the different features and how they pair together

### **How can you select k for k-means?**

We use the elbow method to select k for k-means clustering. The idea of the elbow method is to run k-means clustering on the data set where 'k' is the number of clusters.

### **What is the significance of p-value?**

**p-value typically ≤ 0.05**

This indicates strong evidence against the null hypothesis; so you reject the null hypothesis.

**p-value typically > 0.05**

This indicates weak evidence against the null hypothesis, so you accept the null hypothesis.

**Standard deviation:**

<https://www.mathsisfun.com/data/standard-deviation.html>

Choosing Machine Learning Algorithm.

<https://hackernoon.com/choosing-the-right-machine-learning-algorithm-68126944ce1f>

Pandas:

<https://www.w3resource.com/python-exercises/pandas/index.php>