# SUPERVISED LEARNING

Supervised learning as the name indicates the presence of a supervisor as a teacher. Basically supervised learning is a learning in which we teach or train the machine using data which is well labeled that means some data is already tagged with the correct answer. After that, the machine is provided with a new set of examples(data) so that supervised learning algorithm analyses the training data(set of training examples) and produces a correct outcome from labeled data.

Supervised learning classified into two categories of algorithms:

* **Classification**: A classification problem is when the output variable is a category, such as “disease” and “no disease”.
* **Regression**: A regression problem is when the output variable is a real value, such as “dollars” or “weight”.

**A regression problem** is when the output variable is a real or continuous value, such as “salary” or “weight”. Many different models can be used, the simplest is the linear regression. It tries to fit data with the best hyper-plane which goes through the points.

**Supervised learning models are the following:**

* Linear regression
* Logistic regression
* Decision tree
* Random forest
* K nearest neighbours
* SVM
* Naive Bayes
* Boosting and Bagging

# Training Data and Test Data

**TRAIN DATA:**

The observations in the training set form the experience that the algorithm uses to learn. In supervised learning problems, each observation consists of an observed output variable and one or more observed input variables.

**TEST DATA:**

The test set is a set of observations used to evaluate the performance of the model using some performance metric. It is important that no observations from the training set are included in the test set. If the test set does contain examples from the training set, it will be difficult to assess whether the algorithm has learned to generalize from the training set or has simply memorized it.

In addition to the training and test data, a third set of observations, called a validation or hold-out set, is sometimes required. The validation set is used to tune variables called hyper parameters, which control how the model is learned. The program is still evaluated on the test set to

provide an estimate of its performance in the real world; its performance on the validation set should not be used as an estimate of the model's real-world performance since the program has been tuned specifically to the validation data. It is common to partition a single set

of supervised observations into training, validation, and test sets. There are no requirements for the sizes of the partitions, and they may vary according to the amount of data available. It is common to allocate 50 percent or more of the data to the training set, 25 percent to the test set, and the remainder to the validation set.

**Cross-Validation**

Cross-validation is a technique in which we train our model using the subset of the data-set and then evaluate using the complementary subset of the data-set.

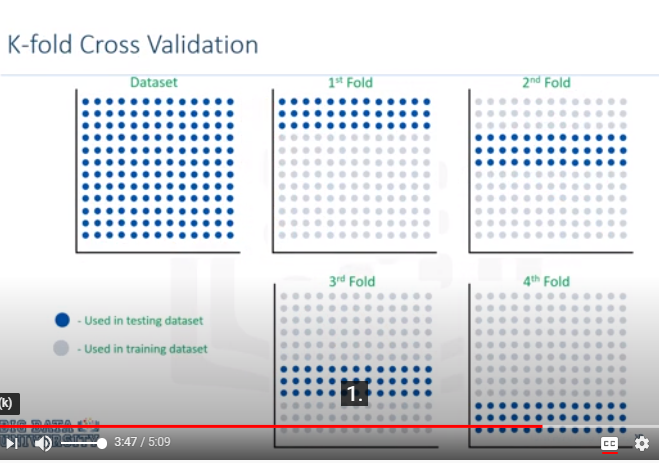
The three steps involved in cross-validation are as follows :

1.Reserve some portion of sample dataset

2.Using the rest data-set train the model.

3.Test the model using the reserve portion of the data-set.

1. **Fold Cross Validation**  
   In this method, we split the data-set into k number of subsets(known as folds) then we perform training on the all the subsets but leave one(k-1) subset for the evaluation of the trained model. In this method, we iterate k times with a different subset reserved for testing purpose each time.



# Performance Measures − Bias and Variance

# Over fitting: When a model performs too well with training data.But performance with the test data will be too bad.

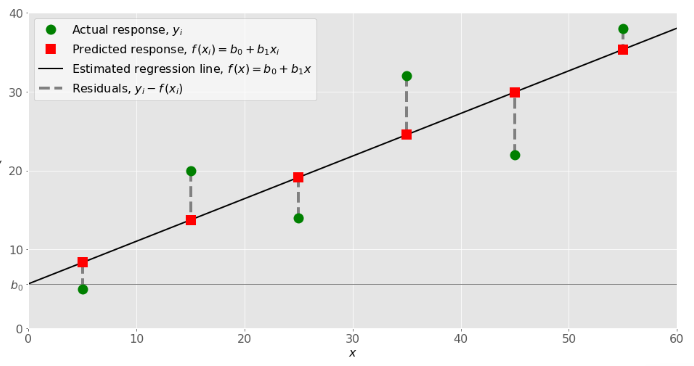
# Under fitting: When the model neither performs well with train data nor with test data.

# High error in the training dataset is called ‘ bias’.High error in test dataset is called ‘variance’.In underfitting we have ‘high bias and high variance’ whereas in overfitting we have ‘low bias and high variance’.

Model should be selected in such a way that it has ‘**low bias and low variance’**

LINEAR REGRESSION

Linear regression is probably one of the most important and widely used regression techniques.Simple or single-variate linear regression is the simplest case of linear regression with a single independent variable,



# Finds out the linear relationship between independent and dependent variable and the dependent variable should be continuous in nature.

Relation will be linear in nature.

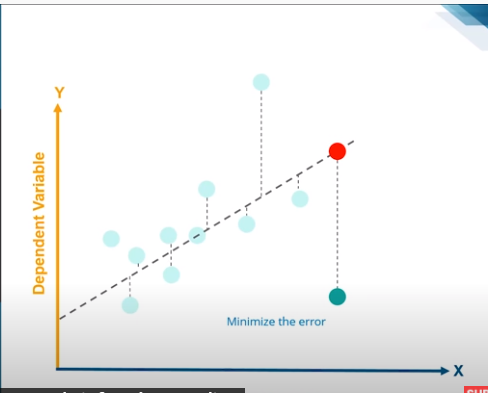
**Simple linear regression uses only one independent feature and Multiple linear regression uses more than 1 independent feature.**

Simple linear regression:

Y=mx+c

Y=dep variable

X=independent var c=reg.coeff(intercept) m=reg.coeff(slope)



# We consider the best fit line.when x=0,point in y axis is the c(intercept)

# With change in x what is the change in y it is the slope.

# Multiple Linear Regression

Y=c+m1 x1+m2 x2+m3 x3……….mn xn

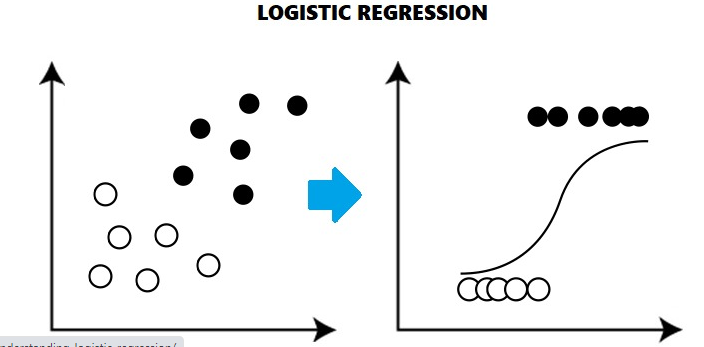
# LOGISTIC REGRESSION

Used for binary classification usually

Data should be linearly separable

We use a sigmoid function

Usually sets a threshold for classification



* Logistic regression uses **sigmoid function** or logistic function which is a complex cost function. This sigmoid function is used to model the data in logistic regression. The function can be represented as:

IMG_256

* f(x)= Output between the 0 and 1 value.
* x= input to the function
* e= base of natural logarithm.

## Types of Logistic Regression

Generally, logistic regression means binary logistic regression having binary target variables, but there can be two more categories of target variables that can be predicted by it. Based on those number of categories, Logistic regression can be divided into following types −

### Binary or Binomial

In such a kind of classification, a dependent variable will have only two possible types either 1 and 0. For example, these variables may represent success or failure, yes or no, win or loss etc.

### Multinomial

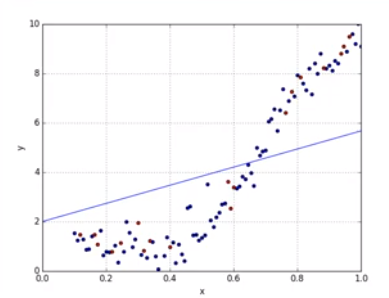
In such a kind of classification, dependent variable can have 3 or more possible ***unordered*** types or the types having no quantitative significance. For example, these variables may represent “Type A” or “Type B” or “Type C”.

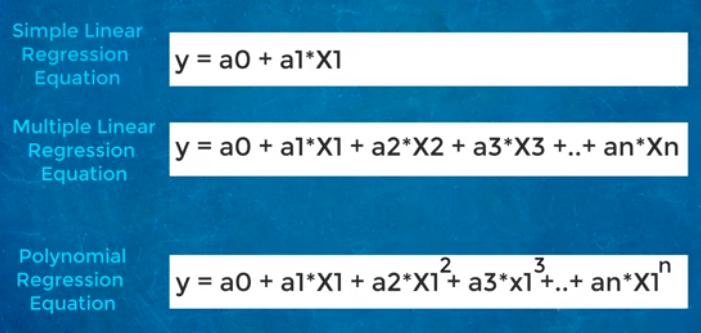
### Ordinal

In such a kind of classification, dependent variable can have 3 or more possible ***ordered*** types or the types having a quantitative significance. For example, these variables may represent “poor” or “good”, “very good”, “Excellent” and each category can have scores like 0,1,2,3.

### Polynomial Regression:

* Polynomial Regression is a type of regression which models the **non-linear dataset** using a linear model.
* It is similar to multiple linear regression, but it fits a non-linear curve between the value of x and corresponding conditional values of y.
* Suppose there is a dataset which consists of datapoints which are present in a non-linear fashion, so for such case, linear regression will not best fit to those datapoints. To cover such datapoints, we need Polynomial regression.
* I**n Polynomial regression, the original features are transformed into polynomial features of given degree and then modeled using a linear model.** Which means the datapoints are best fitted using a polynomial line.



* 
* Here Y is the **predicted/target output, a0, a1,... an are the regression coefficients**. x is our **independent/input variable**.
* The model is still linear as the coefficients are still linear with quadratic
* Used mainly to find the progression eg.growth rate of tissues,progression of disease

#### Note: This is different from Multiple Linear regression in such a way that in Polynomial regression, a single element has different degrees instead of multiple variables with the same degree.

**Regularisation** is the process of introducing additional information in order to avoid overfitting

### Ridge Regression:

* Ridge regression is one of the most robust versions of linear regression in which a small amount of bias is introduced so that we can get better long term predictions.
* The amount of bias added to the model is known as **Ridge Regression penalty**. We can compute this penalty term by multiplying with the lambda to the squared weight of each individual feature.
* The equation for ridge regression will be:



* A general linear or polynomial regression will fail if there is high collinearity between the independent variables, so to solve such problems, Ridge regression can be used.
* Ridge regression is a regularization technique, which is used to reduce the complexity of the model. It is also called as **L2 regularization**.
* 3rd term is called L2 penality
* Lambda is the regularisation parameter.
* It helps to solve the problems if we have more parameters than samples.

### Lasso Regression:

* Lasso regression is another regularization technique to reduce the complexity of the model.
* It is similar to the Ridge Regression except that penalty term contains only the absolute weights instead of a square of weights.
* Since it takes absolute values, hence, it can shrink the slope to 0, whereas Ridge Regression can only shrink it near to 0.
* It is also called as **L1 regularization**. The equation for Lasso regression will be:

# SUPPORT VECTOR MACHINE(SVM)

Support vector machines (SVMs) are powerful yet flexible supervised machine learning algorithms which are used both for classification and regression. But generally, they are used in classification problems.

An SVM model is basically a representation of different classes in a hyperplane in multidimensional space. The hyperplane will be generated in an iterative manner by SVM so that the error can be minimized. The goal of SVM is to divide the datasets into classes to find a maximum marginal hyperplane (MMH).

# IMG_256

The followings are important concepts in SVM −

**Support Vectors** − Datapoints that are closest to the hyperplane is called support vectors. Separating line will be defined with the help of these data points.

**Hyperplane** − As we can see in the above diagram, it is a decision plane or space which is divided between a set of objects having different classes.

**Margin** − It may be defined as the gap between two lines on the closet data points of different classes. It can be calculated as the perpendicular distance from the line to the support vectors. Large margin is considered as a good margin and small margin is considered as a bad margin.

The main goal of SVM is to divide the datasets into classes to find a maximum marginal hyperplane (MMH) and it can be done in the following two steps −

First, SVM will generate hyperplanes iteratively that segregates the classes in best way.

Then, it will choose the hyperplane that separates the classes correctly.

We will start by importing following packages −

import pandas as pd

import numpy as np

from sklearn import svm, datasets

import matplotlib.pyplot as plt

Now, we need to load the input data −

iris = datasets.load\_iris()

From this dataset, we are taking first two features as follows −

X = iris.data[:, :2]

y = iris.target

Next, we will plot the SVM boundaries with original data as follows −

x\_min, x\_max = X[:, 0].min() - 1, X[:, 0].max() + 1

y\_min, y\_max = X[:, 1].min() - 1, X[:, 1].max() + 1

h = (x\_max / x\_min)/100

xx, yy = np.meshgrid(np.arange(x\_min, x\_max, h), np.arange(y\_min, y\_max, h))

X\_plot = np.c\_[xx.ravel(), yy.ravel()]

Now, we need to provide the value of regularization parameter as follows −

C = 1.0

Next, SVM classifier object can be created as follows −

Svc\_classifier = svm.SVC(kernel='linear', C=C).fit(X, y)

Z = svc\_classifier.predict(X\_plot)

Z = Z.reshape(xx.shape)

plt.figure(figsize=(15, 5))

plt.subplot(121)

plt.contourf(xx, yy, Z, cmap=plt.cm.tab10, alpha=0.3)

plt.scatter(X[:, 0], X[:, 1], c=y, cmap=plt.cm.Set1)

plt.xlabel('Sepal length')

plt.ylabel('Sepal width')

plt.xlim(xx.min(), xx.max())

plt.title('Support Vector Classifier with linear kernel')

**Output**

Text(0.5, 1.0, 'Support Vector Classifier with rbf kernel')

**Pros of SVM classifiers**

SVM classifiers offer great accuracy and work well with high dimensional space. SVM classifiers basically use a subset of training points hence in result uses very less memory.

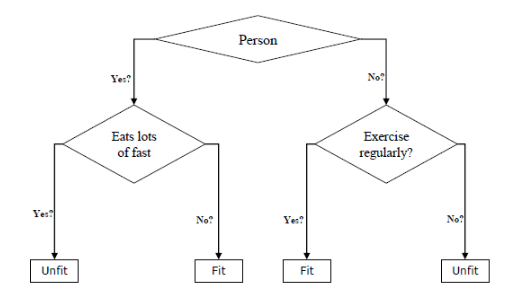
**Cons of SVM classifiers**

They have high training time hence in practice not suitable for large datasets. Another disadvantage is that SVM classifiers do not work well with overlapping classes.

**DECISION TREE**

In general, Decision tree analysis is a predictive modelling tool that can be applied across many areas. Decision trees can be constructed by an algorithmic approach that can split the dataset in different ways based on different conditions. Decision trees are the most powerful algorithms that fall under the category of supervised algorithms.

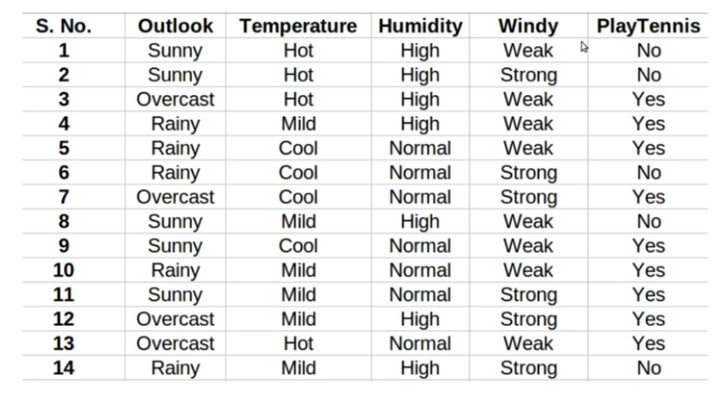
They can be used for both classification and regression tasks. The two main entities of a tree are decision nodes, where the data is split and leaves, where we get outcome. The example of a binary tree for predicting whether a person is fit or unfit providing various information like age, eating habits and exercise habits, is given below −



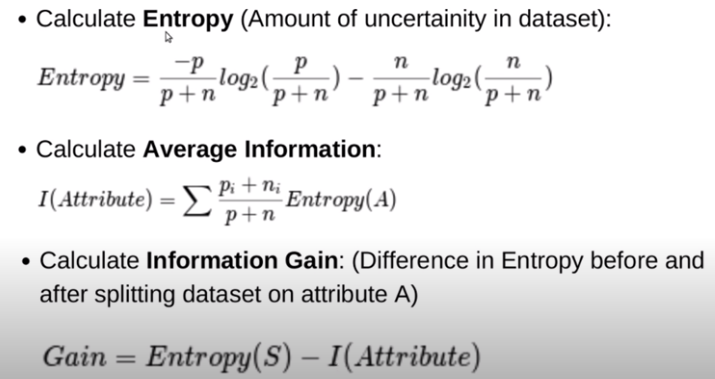
In the above decision tree, the questions are decision nodes and final outcomes are leaves. We have the following two types of decision trees.

**Classification decision trees** − In this kind of decision trees, the decision variable is categorical. The above decision tree is an example of a classification decision tree.

**Regression decision trees** − In this kind of decision trees, the decision variable is continuous.

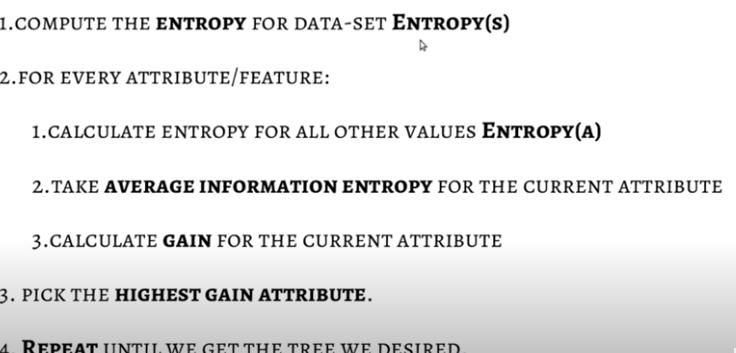


To make a decision tree we have to first select the root node.The attribute which best classifies the training data is chosen as the root node. To choose the best attribute we use the **ID3 algorithm.**



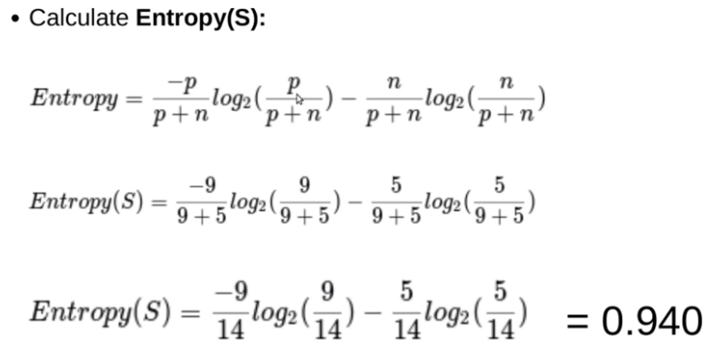
First we will calculate the entropy for the entire dataset. Then we will calculate entropy for each column and find the average . It is known as **average information.**

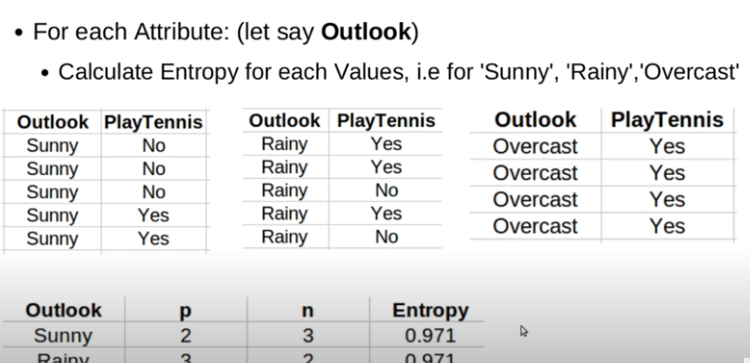
Then we will subtract average information from total entropy. It is called as **Information gain**



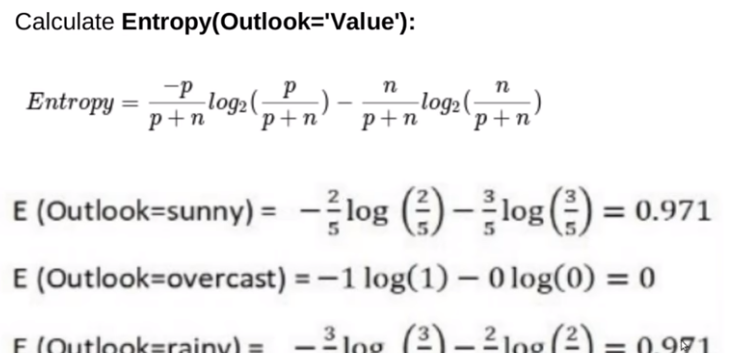
Step 3=Node

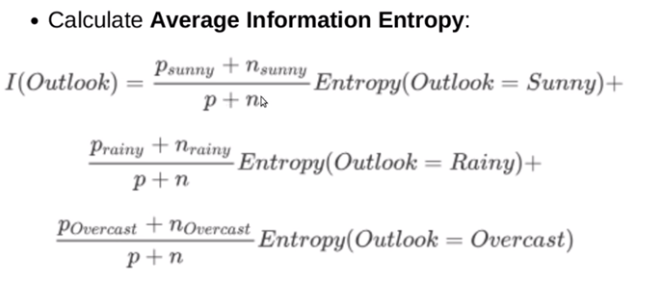
This is the entropy of the entire dataset.

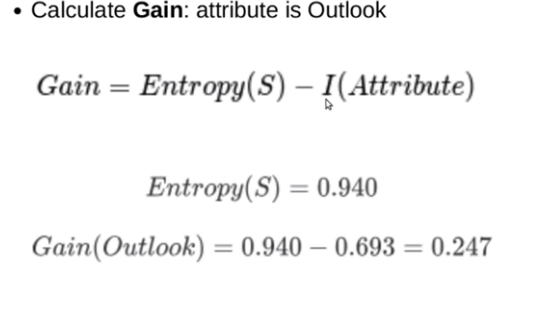


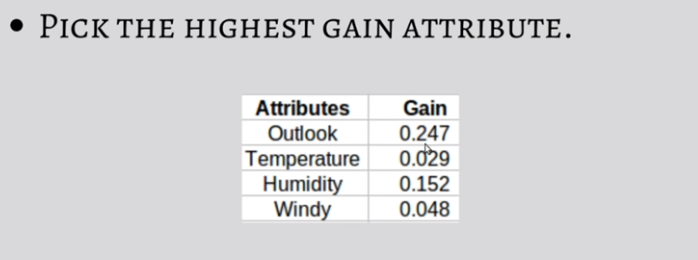


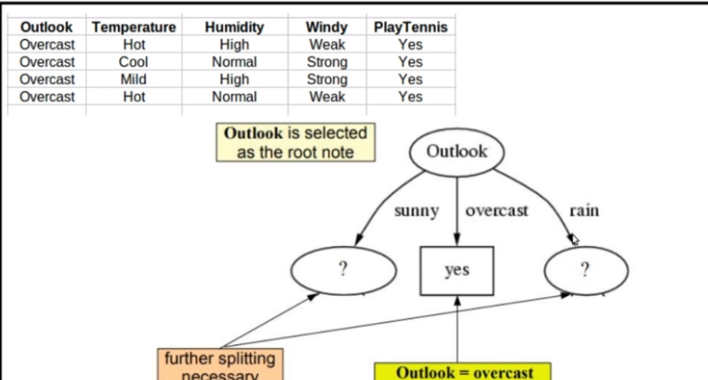
Overcast entropy=0

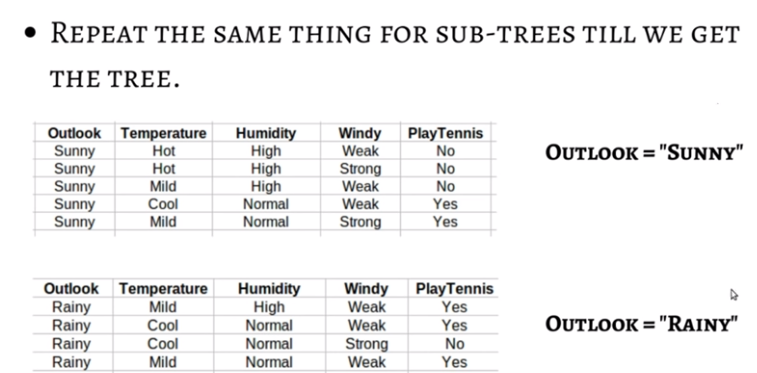








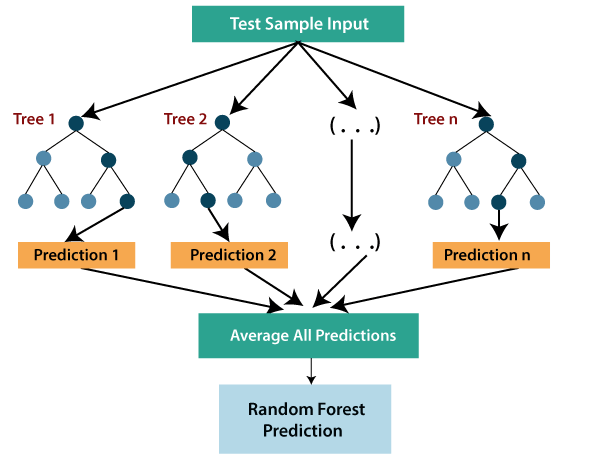




**RANDOM FOREST ALGORITHM**

Random forest is a supervised learning algorithm which is used for both classification as well as regression. But however, it is mainly used for classification problems. As we know that a forest is made up of trees and more trees means more robust forest. Similarly, random forest algorithm creates decision trees on data samples and then gets the prediction from each of them and finally selects the best solution by means of voting. It is an ensemble method which is better than a single decision tree because it reduces the over-fitting by averaging the result.

## Working of Random Forest Algorithm

−

**Step 1** − First, start with the selection of random samples from a given dataset.

**Step 2** − Next, this algorithm will construct a decision tree for every sample. Then it will get the prediction result from every decision tree.

**Step 3** − In this step, voting will be performed for every predicted result.

**Step 4** − At last, select the most voted prediction result as the final prediction result.

* Random forest uses **Bagging or Bootstrap Aggregation** technique of ensemble learning in which aggregated decision tree runs in parallel and do not interact with each other.

**KNN**

K-nearest neighbors (KNN) algorithm is a type of supervised ML algorithm which can be used for both classification as well as regression predictive problems. However, it is mainly used for classification predictive problems in industry.

K-nearest neighbors (KNN) algorithm uses ‘feature similarity’ to predict the values of new data points which further means that the new data point will be assigned a value based on how closely it matches the points in the training set. We can understand its working with the help of the following steps.

**Step 1** − For implementing any algorithm, we need dataset. So during the first step of KNN, we must load the training as well as test data.

**Step 2** − Next, we need to choose the value of K i.e. the nearest data points. K can be any integer.

**Step 3** − For each point in the test data do the following −

**3.1** − Calculate the distance between test data and each row of training data with the help of any of the method namely: Euclidean, Manhattan or Hamming distance. The most commonly used method to calculate distance is Euclidean.

**3.2** − Now, based on the distance value, sort them in ascending order.

**3.3** − Next, it will choose the top K rows from the sorted array.

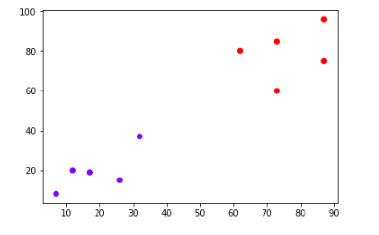
**3.4** − Now, it will assign a class to the test point based on most frequent class of these rows.

**Step 4** − End

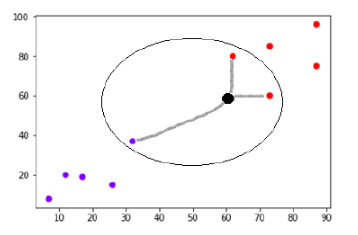
### Example

The following is an example to understand the concept of K and working of KNN algorithm −

Suppose we have a dataset which can be plotted as follows −

1

Now, we need to classify new data point with black dot (at point 60,60) into blue or red class. We are assuming K = 3 i.e. it would find three nearest data points. It is shown in the next diagram −



We can see in the above diagram the three nearest neighbors of the data point with black dot. Among those three, two of them lies in Red class hence the black dot will also be assigned in red class.Knn gets affected by outliers

## Applications of KNN

The following are some of the areas in which KNN can be applied successfully −

### Banking System

KNN can be used in the banking system to predict whether an individual is fit for loan approval? Does that individual have the characteristics similar to the defaulters one?

### Calculating Credit Ratings

KNN algorithms can be used to find an individual’s credit rating by comparing with the persons having similar traits.

### Politics

With the help of KNN algorithms, we can classify a potential voter into various classes like “Will Vote”, “Will not Vote”, “Will Vote to Party ‘Congress’, “Will Vote to Party ‘BJP’.

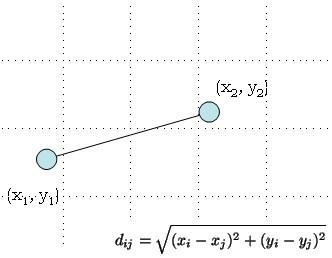
Other areas in which KNN algorithms can be used are Speech Recognition, Handwriting Detection, Image Recognition and Video Recognition.

## Euclidean distance

The Euclidean distance between two points in either the plane or 3-dimensional space measures the length of a segment connecting the two points. It is the most obvious way of representing distance between two points.

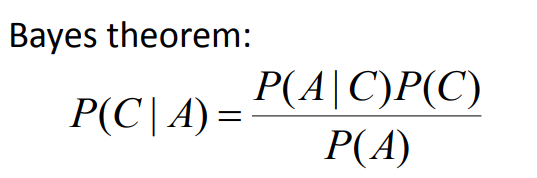
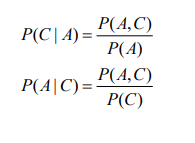
The Pythagorean Theorem can be used to calculate the distance between two points, as shown in the figure below. If the points (x1,y1)(x1,y1) and (x2,y2)(x2,y2) are in 2-dimensional space, then

The Euclidean distance between them



**NAIVE BAYES MODEL**

* Naïve Bayes algorithm is a supervised learning algorithm, which is based on **Bayes theorem** and used for solving classification problems.
* It is mainly used in *text classification* that includes a high-dimensional training dataset.
* Naïve Bayes Classifier is one of the simple and most effective Classification algorithms which helps in building the fast machine learning models that can make quick predictions.
* **It is a probabilistic classifier, which means it predicts on the basis of the probability of an object**.
* Some popular examples of Naïve Bayes Algorithm are **spam filtration, Sentimental analysis, and classifying articles**.
* **Naïve**: It is called Naïve because it assumes that the occurrence of a certain feature is independent of the occurrence of other features. Such as if the fruit is identified on the bases of color, shape, and taste, then red, spherical, and sweet fruit is recognized as an apple. Hence each feature individually contributes to identify that it is an apple without depending on each other.
* **Bayes**: It is called Bayes because it depends on the principle of [Bayes' Theorem](https://www.javatpoint.com/bayes-theorem-in-artifical-intelligence).
* Naive Bayes Classifiers rely on the Bayes’ Theorem, which is based on conditional probability or in simple terms, the likelihood that an event (A) will happen *given that*another event (B) has already happened. Essentially, the theorem allows a hypothesis to be updated each time new evidence is introduced. The equation below expresses Bayes’ Theorem in the language of probability:



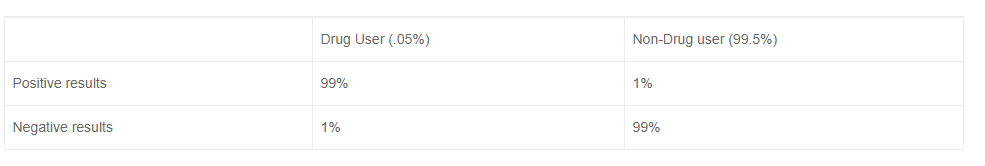
* “P” is the symbol to denote probability.
* P(A | B) = The probability of event A (hypothesis) occurring given that B (evidence) has occurred.
* P(B | A) = The probability of the event B (evidence) occurring given that A (hypothesis) has occurred.
* P(A) = The probability of event B (hypothesis) occurring.
* P(B) = The probability of event A (evidence) occurring.

# **Drug Testing**

To better illustrate the usefulness of Bayes’ Theorem, let’s walk through an example. Suppose that a drug-testing method has the following statistical results:

* 0.5 percent of the population uses drugs.
* The test is positive for drug users 99 percent of the time (and therefore, it misses 1 percent of drug users).
* The test is negative for a non-drug user 99 percent of the time (and positive for a non-drug user 1 percent of the time).

We would like to know the odds that a randomly selected individual who had a positive test result is actually a drug user, i.e. that the results are a *true positive*. In the language of probability, that looks something like this:



* P( User | Positive ) = The probability that an individual that had a positive test result is actually a drug user.

Following Bayes’ Theorem, we know that the probability is calculated as follows:

P( User | Positive ) = ( P( Positive | User ) \* P( User) ) / P(Positive)

Now, let’s calculate each of the terms on the right side of the equation.

* P( Positive | User ) = 0.99. This is the chance of returning *true positive* test results if you are a drug user.
* P( User) = 0.005. The percentage of the population that are drug users.
* P(Positive) = P(Positive | User) \* P(User) + P(Positive | Non-user) \* P(Non-user), which translates to: .99\*.005 + .01\*.995 = 0.0149. [This is the breakdown of how a positive result— a user testing positive(*true positive*) and a non-user testing positive (*false positive*)—can occur.

Applying these figures to the original formula gives you:

P( User | Positive ) = (.99 \* .005 ) / .99 = 33%

What a surprising result! At first, the test seems highly accurate. However, upon further calculation, we see that it’s far more likely that even after testing positive a person is, in fact, not a drug-user.

## Types of Naïve Bayes Model:

There are three types of Naive Bayes Model, which are given below:

* **Gaussian**: The Gaussian model assumes that features follow a normal distribution. This means if predictors t**ake continuous values instead of discrete,** then the model assumes that these values are sampled from the Gaussian distribution.
* **Multinomial**: The Multinomial Naïve Bayes classifier is used when the data is multinomial distributed. It is primarily used for **document classification problems,** it means a particular document belongs to which category such as Sports, Politics, education, etc.  
  The classifier uses the frequency of words for the predictors.
* **Bernoulli**: The Bernoulli classifier works similar to the Multinomial classifier, but the predictor variables are the independent Booleans variables. Such as if a particular word is present or not in a document. This model is also famous for document classification tasks.

Advantages of Naive Bayes Classifiers

* Simple model
* Fast
* Scalable
* Requires little data

Disadvantages of Naive Bayes Classifiers

* Assumes feature independence
* Must choose the likelihood function

USES

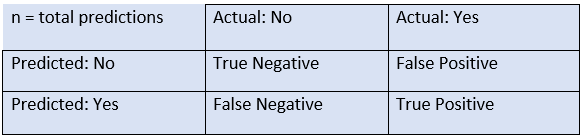
* It is used for Credit Scoring.
* It is used in medical data classification.
* It can be used in real-time predictions because Naïve Bayes Classifier is an eager learner.
* It is used in Text classification such as Spam filtering and Sentiment analysis.

**METRICS IN CLASSIFICATION**

**1. CONFUSION MATRIX**

The confusion matrix is a matrix used to determine the performance of the classification models for a given set of test data. It can only be determined if the true values for test data are known. The matrix itself can be easily understood, but the related terminologies may be confusing. Since it shows the errors in the model performance in the form of a matrix, hence also known as an **error matrix**. Some features of Confusion matrix are given below:

* For the 2 prediction classes of classifiers, the matrix is of 2\*2 table, for 3 classes, it is 3\*3 tables, and so on.
* The matrix is divided into two dimensions, that are **predicted values** and **actual values** along with the total number of predictions.
* Predicted values are those values, which are predicted by the model, and actual values are the true values for the given observations.



* **True Negative:** Model has given prediction No, and the real or actual value was also No.
* **True Positive:** The model has predicted yes, and the actual value was also true.
* **False Negative:** The model has predicted no, but the actual value was Yes, it is also called as **Type-II error**.
* **False Positive:** The model has predicted Yes, but the actual value was No. It is also called a **Type-I error.**

## Need for Confusion Matrix in Machine learning

* It evaluates the performance of the classification models, when they make predictions on test data, and tells how good our classification model is.
* It not only tells the error made by the classifiers but also the type of errors such as it is either type-I or type-II error.
* With the help of the confusion matrix, we can calculate the different parameters for the model, such as accuracy, precision, etc.

## Calculations using Confusion Matrix:

We can perform various calculations for the model, such as the model's accuracy, using this matrix. These calculations are given below:

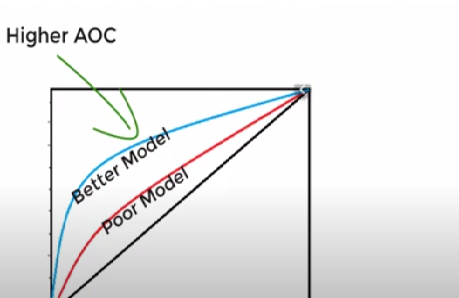
* **Classification Accuracy:** It is one of the important parameters to determine the accuracy of the classification problems. It defines how often the model predicts the correct output. It can be calculated as the ratio of the number of correct predictions made by the classifier to all numbers of predictions made by the classifiers. The formula is given below:

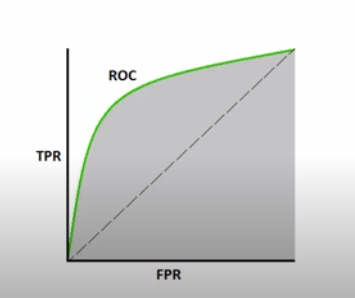
IMG_256

* **Misclassification rate:** It is also termed as Error rate, and it defines how often the model gives the wrong predictions. The value of error rate can be calculated as the number of incorrect predictions to all numbers of the predictions made by the classifier. The formula is given below:  
  IMG_256
* **Precision:** It can be defined as the number of correct outputs provided by the model or out of all positive classes that have predicted correctly by the model, how many of them were actually true. It can be calculated using the below formula:  
  IMG_258
* **Recall:** It is defined as the out of total predicted positive classes, how our model predicted correctly. The recall must be as high as possible.  
  
* **F-measure:** If two models have low precision and high recall or vice versa, it is difficult to compare these models. So, for this purpose, we can use F-score. This score helps us to evaluate the recall and precision at the same time. The F-score is maximum if the recall is equal to the precision. It can be calculated using the below formula:

Other important terms used in Confusion Matrix:

* **Null Error rate:** It defines how often our model would be incorrect if it always predicted the majority class. As per the accuracy paradox, it is said that "*t*he best classifier has a higher error rate than the null error rate."
* **ROC Curve(RECEIVER OPERATING CHARACTERISTICS):**
* Used for **binary classification** problem
* The ROC is a graph displaying a classifier's performance for all possible thresholds. The graph is plotted between the true positive rate (on the Y-axis) and the false Positive rate (on the x-axis).
* **AUC (Area under curve)**





## Learners in Classification Problems:

In the classification problems, there are two types of learners:

1. **Lazy Learners:** Lazy Learner firstly stores the training dataset and wait until it receives the test dataset. In the Lazy learner case, classification is done on the basis of the most related data stored in the training dataset. It takes less time in training but more time for predictions.  
   **Example:** K-NN algorithm, Case-based reasoning
2. **Eager Learners:** Eager Learners develop a classification model based on a training dataset before receiving a test dataset. Opposite to Lazy learners, Eager learners take less time in training and less time in prediction. **Example:** Decision Trees, Naïve Bayes, ANN.

**AUC-ROC curve:**

* ROC curve stands for **Receiver Operating Characteristics Curve** and AUC stands for **Area Under the Curve**.
* It is a graph that shows the performance of the classification model at different thresholds.
* To visualize the performance of the multi-class classification model, we use the AUC-ROC Curve.
* The ROC curve is plotted with TPR and FPR, where TPR (True Positive Rate) on Y-axis and FPR(False Positive Rate) on X-axis.

## Use cases of Classification Algorithms

Classification algorithms can be used in different places. Below are some popular use cases of Classification Algorithms:

* Email Spam Detection
* Speech Recognition
* Identifications of Cancer tumor cells.
* Drugs Classification
* Biometric Identification, etc.

**ENSEMBLE LEARNING**

Ensemble models in machine learning operate on a similar idea. They combine the decisions from multiple models to improve the overall performance. This can be achieved in various ways, which you will discover in this article.

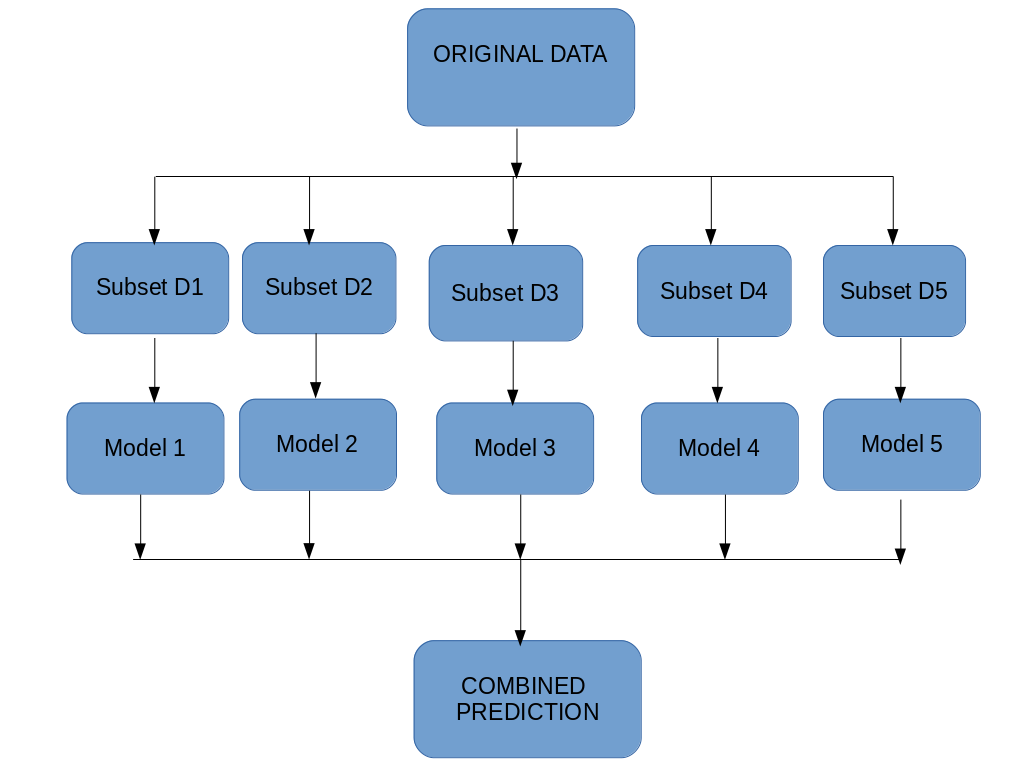
Bagging

The idea behind bagging is combining the results of multiple models (for instance, all decision trees) to get a generalized result. Here’s a question: If you create all the models on the same set of data and combine it, will it be useful? There is a high chance that these models will give the same result since they are getting the same input. So how can we solve this problem? One of the techniques is bootstrapping.

Bootstrapping is a sampling technique in which we create subsets of observations from the original dataset, **with replacement**. The size of the subsets is the same as the size of the original set.

Bagging (or Bootstrap Aggregating) technique uses these subsets (bags) to get a fair idea of the distribution (complete set). The size of subsets created for bagging may be less than the original set.

1. Multiple subsets are created from the original dataset, selecting observations with replacement.
2. A base model (weak model) is created on each of these subsets.
3. The models run in parallel and are independent of each other.
4. The final predictions are determined by combining the predictions from all the models



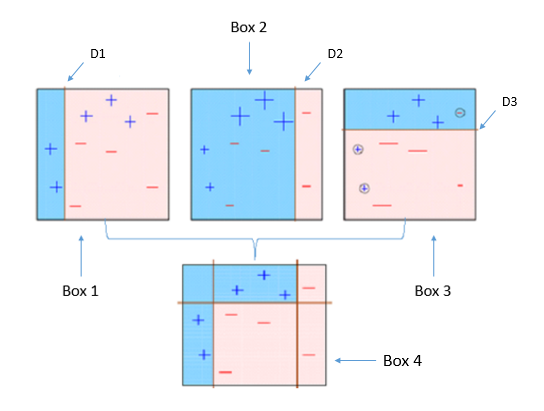
## Boosting:

Boosting is another ensemble procedure to make a collection of predictors. In other words, we fit consecutive trees, usually random samples, and at each step, the objective is to solve net error from the prior trees.

If a given input is misclassified by theory, then its weight is increased so that the upcoming hypothesis is more likely to classify it correctly by consolidating the entire set and at last converts weak learners into better performing models.

### AdaBoost

Adaptive boosting or AdaBoost is one of the simplest boosting algorithms. Usually, decision trees are used for modelling. Multiple sequential models are created, each correcting the errors from the last model. AdaBoost assigns weights to the observations which are incorrectly predicted and the subsequent model works to predict these values correctly.



As we see above, the first decision stump(D1) is made separating the (+) blue region from the ( — ) red region. We notice that D1 has three incorrectly classified (+) in the red region. The incorrect classified (+) will now carry more weight than the other observations and be fed to the second learner. The model will continue and adjust the error faced by the previous model until the most accurate predictor is built.

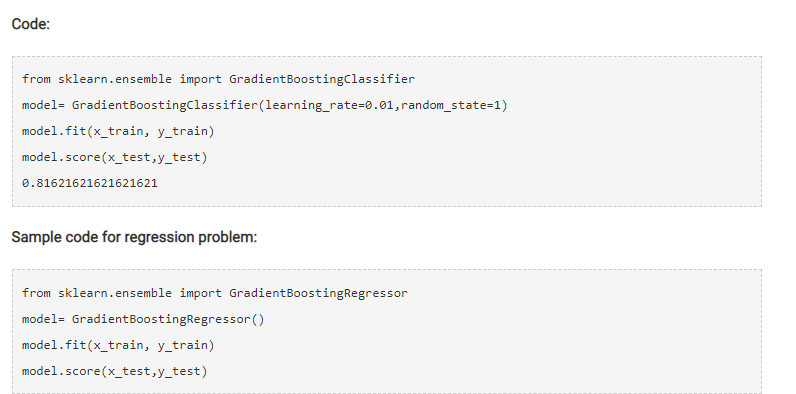
Below are the steps for performing the AdaBoost algorithm:

1. Initially, all observations in the dataset are given equal weights.
2. A model is built on a subset of data.
3. Using this model, predictions are made on the whole dataset.
4. Errors are calculated by comparing the predictions and actual values.
5. While creating the next model, higher weights are given to the data points which were predicted incorrectly.
6. Weights can be determined using the error value. For instance, higher the error more is the weight assigned to the observation.
7. This process is repeated until the error function does not change, or the maximum limit of the number of estimators is reached.



### Gradient Boosting (GBM)

Gradient Boosting or GBM is another ensemble machine learning algorithm that works for both regression and classification problems. GBM uses the boosting technique, combining a number of weak learners to form a strong learner. Regression trees used as a base learner, each subsequent tree in series is built on the errors calculated by the previous tree.



### XGBoost

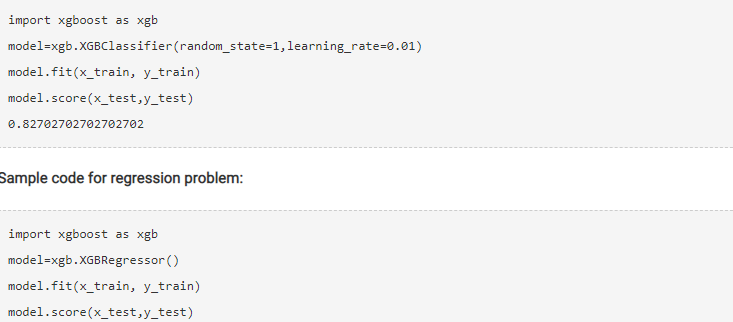
XGBoost (extreme Gradient Boosting) is an advanced implementation of the gradient boosting algorithm. XGBoost has proved to be a highly effective ML algorithm, extensively used in machine learning competitions and hackathons. XGBoost has high predictive power and is almost 10 times faster than the other gradient boosting techniques. It also includes a variety of regularization which reduces overfitting and improves overall performance. Hence it is also known as ‘**regularized boosting**‘ technique.

Let us see how XGBoost is comparatively better than other techniques:

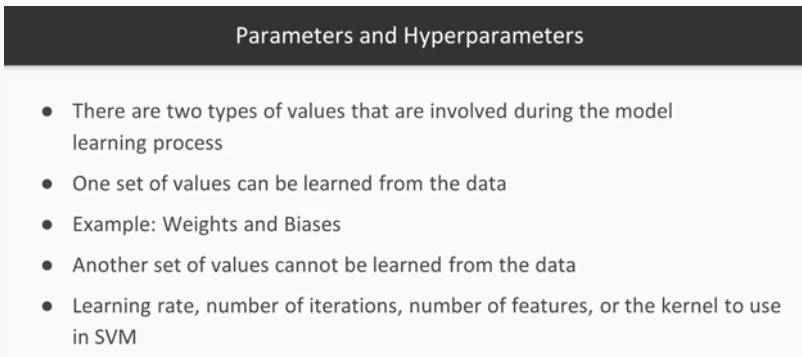
1. **Regularization:**
   1. Standard GBM implementation has no regularisation like XGBoost.
   2. Thus XGBoost also helps to reduce overfitting.
2. **Parallel Processing:**
   1. XGBoost implements parallel processing and is faster than GBM .
   2. XGBoost also supports implementation on Hadoop.
3. **High Flexibility:**
   1. XGBoost allows users to define custom optimization objectives and evaluation criteria adding a whole new dimension to the model.
4. **Handling Missing Values:**
   1. XGBoost has an in-built routine to handle missing values.
5. **Tree Pruning:**
   1. XGBoost makes splits up to the max\_depth specified and then starts pruning the tree backwards and removes splits beyond which there is no positive gain.
6. **Built-in Cross-Validation:**
   1. XGBoost allows a user to run a cross-validation at each iteration of the boosting process and thus it is easy to get the exact optimum number of boosting iterations in a single run.

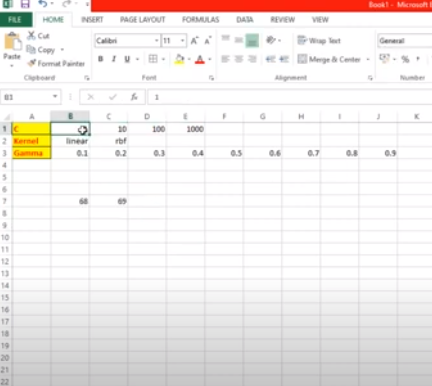
**Code:**

Since XGBoost takes care of the missing values itself, you do not have to impute the missing values. You can skip the step for missing value imputation from the code mentioned above. Follow the remaining steps as always and then apply xgboost as below.



**HYPERPARAMETER OPTIMIZATION**





In **Random search** cv value combinations are taken at random.We may not get the best result but somewhere around the best one.It is best used with very big data.It saves time

In **grid search**:All the possible permutations and combinations of parameters are done.And the best one is selected.It is very time consuming so we need to **consider rescaling** of data sometimes.

**HYPOTHESIS TESTING**

Hypothesis testing is an essential procedure in statistics. A hypothesis test evaluates two mutually exclusive statements about a population to determine which statement is best supported by the sample data. When we say that a finding is statistically significant, it’s thanks to a hypothesis test.

The null hypothesis states that there is no relationship between the [two variables being studied](https://www.simplypsychology.org/variables.html) (one variable does not affect the other)(H0)

The alternative hypothesis is the one you would believe if the null hypothesis is concluded to be untrue.(H1)

1. **Value**

The level of statistical significance is often expressed as a p-value between 0 and 1. The smaller the p-value, the stronger the evidence that you should reject the null hypothesis. A smaller p-value means that there is stronger evidence in favor of the alternative hypothesis.

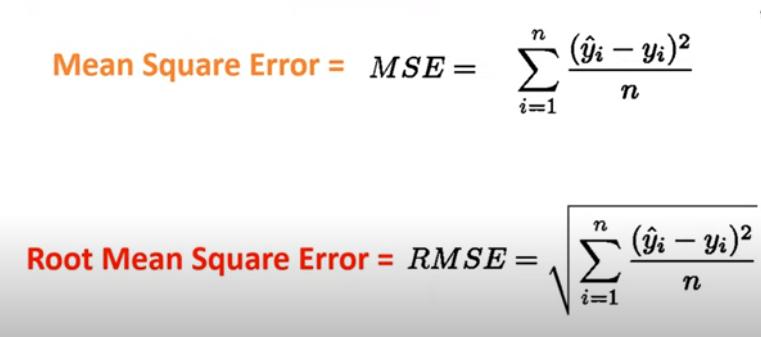
A *p*-value less than 0.05 (typically ≤ 0.05) is statistically significant. It indicates strong evidence against the null hypothesis, as there is less than a 5% probability the null is correct (and the results are random). Therefore, we reject the null hypothesis, and accept the alternative hypothesis.

A *p*-value higher than 0.05 (> 0.05) is not statistically significant and indicates strong evidence for the null hypothesis. This means we retain the null hypothesis and reject the alternative hypothesis. You should note that you cannot accept the null hypothesis, we can only reject the null or fail to reject it.

**REGRESSION EVALUATION METRICS:**

ERRORS IN MACHINE LEARNING MODELS

Errors in machine learning models are known as ‘**cost function’**.



We can find the root mean square error of each model and the model which gives the minimum value will be the best model.These are part of model evaluation.

**R squared** is a statistical measure that represents the goodness of fit of a regression model. The ideal value for r-square is 1. The closer the value of r-square to 1, the better is the model fitted.

**Note :** The value of R-square can also be negative when the models fitted is worse than the average fitted model

**FEATURE SCALING**

Standardization and normalization are the two most commonly used feature scaling techniques in machine learning.

**Normalization** is a scaling technique in which values are shifted and rescaled so that they end up ranging between 0 and 1. It is also known as **Min-Max scaling.**

**Standardization** is another scaling technique where the values are centered around the mean with a unit standard deviation. This means that the mean of the attribute becomes zero and the resultant distribution has a unit standard deviation.

Here’s the curious thing about feature scaling – it improves (significantly) the performance of some machine learning algorithms and does not work at all for others.

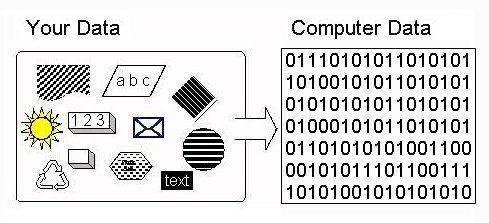
Machine learning algorithms like [**linear regression**](https://www.analyticsvidhya.com/blog/2017/05/neural-network-from-scratch-in-python-and-r/?utm_source=blog&utm_medium=feature-scaling-machine-learning-normalization-standardization)**,**[**logistic regression**](https://www.analyticsvidhya.com/blog/2017/05/neural-network-from-scratch-in-python-and-r/?utm_source=blog&utm_medium=feature-scaling-machine-learning-normalization-standardization), [**neural network**](https://www.analyticsvidhya.com/blog/2017/05/neural-network-from-scratch-in-python-and-r/?utm_source=blog&utm_medium=feature-scaling-machine-learning-normalization-standardization)**,** etc. that use gradient descent as an optimization technique require data to be scaled

Distance algorithms like[**KNN**](https://www.analyticsvidhya.com/blog/2018/03/introduction-k-neighbours-algorithm-clustering/?utm_source=blog&utm_medium=feature-scaling-machine-learning-normalization-standardization)**,**[**K-means**](https://www.analyticsvidhya.com/blog/2019/08/comprehensive-guide-k-means-clustering/?utm_source=blog&utm_medium=feature-scaling-machine-learning-normalization-standardization)**, and**[**SVM**](https://www.analyticsvidhya.com/blog/2017/09/understaing-support-vector-machine-example-code/?utm_source=blog&utm_medium=feature-scaling-machine-learning-normalization-standardization)are most affected by the range of features. This is because behind the scenes they are using distances between data points to determine their similarity

[**Tree-based algorithms**](https://www.analyticsvidhya.com/blog/2016/04/tree-based-algorithms-complete-tutorial-scratch-in-python/?utm_source=blog&utm_medium=feature-scaling-machine-learning-normalization-standardization), on the other hand, are fairly insensitive to the scale of the features. Think about it, a decision tree is only splitting a node based on a single feature. The decision tree splits a node on a feature that increases the homogeneity of the node. This split on a feature is not influenced by other features.So, there is virtually no effect of the remaining features on the split. This is what makes them invariant to the scale of the features!.we don’t need to scale for decision tree,boosting techniques,random forest.

## What is Categorical Encoding?

Typically, any structured dataset includes multiple columns – a combination of numerical as well as categorical variables. A machine can only understand the numbers. It cannot understand the text. That’s essentially the case with [Machine Learning algorithms](https://www.analyticsvidhya.com/blog/2017/09/common-machine-learning-algorithms/?utm_source=blog&utm_medium=one-hot-encoding-vs-label-encoding-using-scikit-learn) too.



*Categorical encoding is a process of converting categories to numbers*