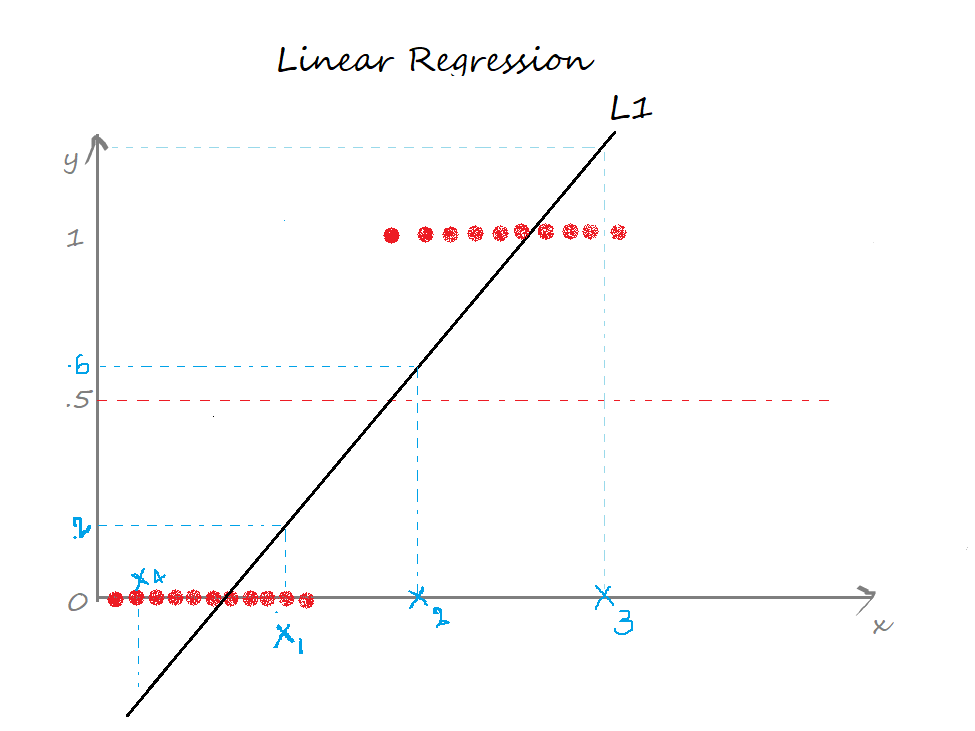
**WEEK 8**

**Logistic Regression**

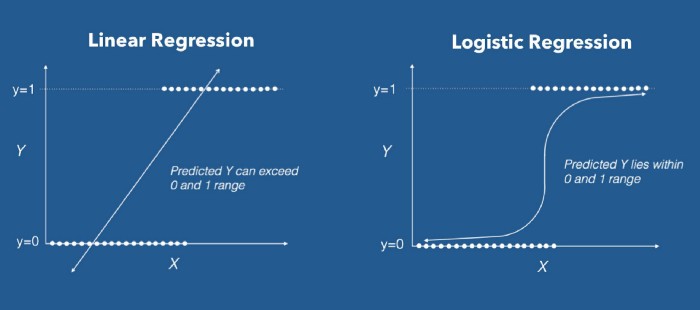
Assume we have a dataset that is linearly separable and has the output that is discrete in two classes (0, 1).  if we try to use linear regression to solve a binary class classification problem, the regression like looks as follows.



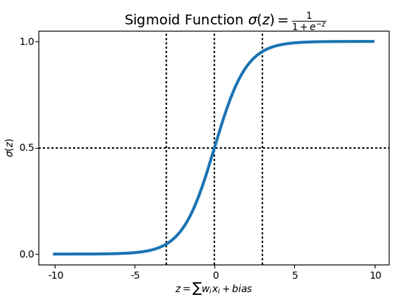
The two limitations of using a linear regression model for classification problems are:

* the predicted value may exceed the range (0,1)
* error rate increases if the data has outliers

**Logistic Regression is a Machine Learning algorithm which is used for the classification problems, it is a predictive analysis algorithm and based on the concept of probability.**



* Logistic Regression uses a more complex cost function, this cost function can be defined as the ‘**Sigmoid function**’ or also known as the ‘logistic function’ instead of a linear function.
* The hypothesis of logistic regression tends it to limit the cost function between 0 and 1.
* In order to map predicted values to probabilities, we use the Sigmoid function.
* The function maps any real value into another value between 0 and 1. In machine learning, we use sigmoid to map predictions to probabilities.



Linear model:   
Sigmoid function:    
Logistic regression model:

**Types of logistic regression**

Logistic Regression can be classified into three types:

* **Binomial:** In binomial Logistic regression, there can be only two possible types of the dependent variables, such as 0 or 1, Pass or Fail, etc.
* **Multinomial:** In multinomial Logistic regression, there can be 3 or more possible unordered types of the dependent variable, such as "cat", "dogs", or "sheep"
* **Ordinal:** In ordinal Logistic regression, there can be 3 or more possible ordered types of dependent variables, such as "low", "Medium", or "High".

## How does Logistic Regression Work? Use of Sigmoid function

* The logistic regression equation is quite similar to the linear regression model.
* Consider we have a model with one predictor “x” and one Bernoulli response variable “ŷ” and p is the probability of ŷ=1. The linear equation can be written as:

p = b0+b1x --------> eq 1

* The right-hand side of the equation (b0+b1x) is a linear equation and can hold values that exceed the range (0,1).
* But we know probability will always be in the range of (0,1).
* To overcome that, we predict odds instead of probability.

**Odds:**The ratio of the probability of an event occurring to the probability of an event not occurring. Odds = p/(1-p)

* The equation 1 can be re-written as:

p/(1-p) = b0+b1x      --------> eq 2

* Odds can only be a positive value, to tackle the negative numbers, we predict the **logarithm of odds**.

Log of odds = ln(p/(1-p))

* The equation 2 can be re-written as:

ln(p/(1-p)) = b0+b1x      --------> eq 3

* To recover p from equation 3, we apply exponential on both sides.

exp(ln(p/(1-p))) = exp(b0+b1x)

eln(p/(1-p)) = e(b0+b1x)

* Solving the equation for p, we will get,

p = 1 / (1 + e-(b0+b1x))

A linear equation (z) is given to a sigmoidal activation function (σ) to predict the output (ŷ).

**Assumptions of Logistic Regression**

### ****1. The Response Variable is Binary****

Logistic regression assumes that the response variable only takes on two possible outcomes

### ****2. The Observations are Independent****

Logistic regression assumes that the observations in the dataset are independent of each other. That is, the observations should not come from repeated measurements of the same individual or be related to each other in any way.

### ****3. There is No Multicollinearity Among Explanatory Variables****

Logistic regression assumes that there is no severe [multicollinearity](https://www.statology.org/multicollinearity-regression/) among the [explanatory variables](https://www.statology.org/explanatory-response-variables/).

Multicollinearity occurs when two or more explanatory variables are highly correlated to each other, such that they do not provide unique or independent information in the regression model. If the degree of correlation is high enough between variables, it can cause problems when fitting and interpreting the model.

For example, if the following variables are explanatory variables:

* **Player height**
* **Player shoe size**
* **Hours spent practicing per day**

In this case, **height** and **shoe size** are likely to be highly correlated since taller people tend to have larger shoe sizes.

### ****4. There are No Extreme Outliers****

Logistic regression assumes that there are no extreme outliers or influential observations in the dataset.

### ****5. There is a Linear Relationship Between Explanatory Variables and the Logit of the Response Variable****

Logistic regression assumes that there exists a linear relationship between each explanatory variable and the logit of the response variable. Recall that the logit is defined as:

Logit(p)  = log(p / (1-p)) where p is the probability of a positive outcome.

### ****6. The Sample Size is Sufficiently Large****

Logistic regression assumes that the sample size of the dataset if large enough to draw valid conclusions from the fitted logistic regression model. As a rule of thumb, you should have a minimum of 10 cases with the least frequent outcome for each explanatory variable. For example, if you have 3 explanatory variables and the expected probability of the least frequent outcome is 0.20, then you should have a sample size of at least (10\*3) / 0.20 = **150**.

# **Support Vector Machine**

* Support Vector Machine or SVM is one of the most popular Supervised Learning algorithms, which is used for Classification problems in Machine Learning.
* The goal of the SVM algorithm is to create the best line or decision boundary that can segregate n-dimensional space into classes so that we can easily put the new data point in the correct category in the future. This best decision boundary is called a hyperplane.
* SVM chooses the extreme points/vectors that help in creating the hyperplane. These extreme cases are called as support vectors, and hence algorithm is termed as Support Vector Machine.
* Consider the below diagram in which there are two different categories that are classified using a decision boundary or hyperplane:



**Hyperplane:**

* There can be multiple lines/decision boundaries to segregate the classes in n-dimensional space, but we need to find out the best decision boundary that helps to classify the data points. This best boundary is known as the hyperplane of SVM.
* The dimensions of the hyperplane depend on the features present in the dataset, which means if there are 2 features (as shown in image), then hyperplane will be a straight line. And if there are 3 features, then hyperplane will be a 2-dimension plane.
* We always create a hyperplane that has a **maximum margin**, which means the maximum distance between the data points.

**Support Vectors:**

* The data points or vectors that are the closest to the hyperplane and which affect the position of the hyperplane are termed as Support Vector.
* Since these vectors support the hyperplane, hence called a Support vector.

**SVM can be of two types:**

* **Linear SVM:** Linear SVM is used for linearly separable data, which means if a dataset can be classified into two classes by using a single straight line, then such data is termed as linearly separable data, and classifier is used called as Linear SVM classifier.
* **Non-linear SVM:** Non-Linear SVM is used for non-linearly separated data, which means if a dataset cannot be classified by using a straight line, then such data is termed as non-linear data and classifier used is called as Non-linear SVM classifier.

## How does SVM works?

**Linear SVM:**

* The working of the SVM algorithm can be understood by using an example.
* Suppose we have a dataset that has two tags (green and blue), and the dataset has two features x1 and x2.
* We want a classifier that can classify the pair(x1, x2) of coordinates in either green or blue.
* So as it is 2-d space so by just using a straight line, we can easily separate these two classes.
* But there can be multiple lines that can separate these classes. Consider the below image:



* Hence, the SVM algorithm helps to find the best line or decision boundary; this best boundary or region is called as a **hyperplane**.
* SVM algorithm finds the closest point of the lines from both the classes. These points are called support vectors.
* The distance between the vectors and the hyperplane is called as **margin**. And the goal of SVM is to maximize this margin.
* The **hyperplane** with maximum margin is called the **optimal hyperplane**.



**Non-Linear SVM:**

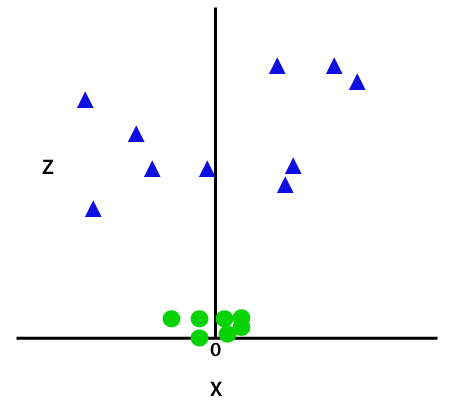
If data is linearly arranged, then we can separate it by using a straight line, but for non-linear data, we cannot draw a single straight line. Consider the below image:



So to separate these data points, we need to add one more dimension. For linear data, we have used two dimensions x and y, so for non-linear data, we will add a third-dimension z. It can be calculated as:

z=x2 +y2

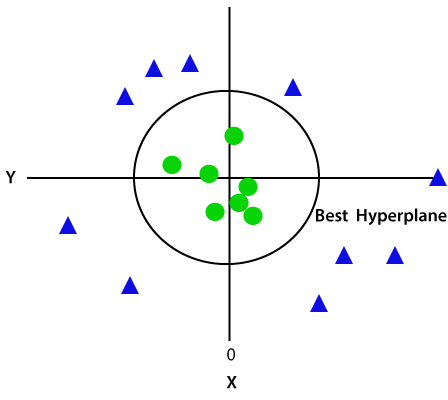
By adding the third dimension, the sample space will become as below image:



So now, SVM will divide the datasets into classes in the following way. Consider the below image:



Since we are in 3-d Space, hence it is looking like a plane parallel to the x-axis. If we convert it in 2d space with z=1, then it will become as:



Hence we get a circumference of radius 1 in case of non-linear data.

SVM algorithm can be used for

* **Face detection**
* **image classification**
* **text categorization,** etc.

# **Ensemble Learning**

Ensemble models in machine learning **combine the decisions from multiple models** to improve the overall performance of the task in hand.  Diversification in Machine Learning is achieved by a technique called Ensemble Learning.

## Simple Ensemble Techniques

### 1. Max Voting

* The max voting method is generally used for classification problems.
* In this technique, multiple models are used to make predictions for each data point.
* The predictions by each model are considered as a ‘vote’.
* The predictions which we get from the majority of the models are used as the final prediction.
* For example, when you asked 5 of your colleagues to rate your movie (out of 5); we’ll assume three of them rated it as 4 while two of them gave it a 5. Since the majority gave a rating of 4, the final rating will be taken as 4. **You can consider this as taking the mode of all the predictions.**
* The result of max voting would be something like this:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Colleague 1 | Colleague 2 | Colleague 3 | Colleague 4 | Colleague 5 | Final rating |
| 5 | 4 | 5 | 4 | 4 | 4 |

### 2. Averaging

* Similar to the max voting technique, multiple predictions are made for each data point in averaging.
* In this method, we take an average of predictions from all the models and use it to make the final prediction.
* Averaging can be used for making predictions in regression problems or while calculating probabilities for classification problems.
* For example, in the below case, the averaging method would take the average of all the values. i.e. (5+4+5+4+4)/5 = 4.4

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Colleague 1 | Colleague 2 | Colleague 3 | Colleague 4 | Colleague 5 | Final rating |
| 5 | 4 | 5 | 4 | 4 | 4.4 |

### 3. Weighted Average

* This is an extension of the averaging method.
* All models are assigned different weights defining the importance of each model for prediction.
* For instance, if two of your colleagues are critics, while others have no prior experience in this field, then the answers by these two friends are given more importance as compared to the other people.
* The result is calculated as [(5\*0.23) + (4\*0.23) + (5\*0.18) + (4\*0.18) + (4\*0.18)] = 4.41.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Colleague 1 | Colleague 2 | Colleague 3 | Colleague 4 | Colleague 5 | Final rating |
| weight | 0.23 | 0.23 | 0.18 | 0.18 | 0.18 |  |
| rating | 5 | 4 | 5 | 4 | 4 | 4.41 |

## Advanced Ensemble techniques

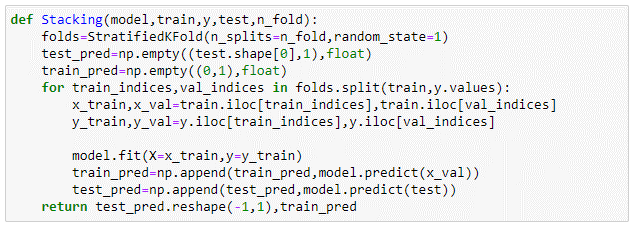
### 1. Stacking

* Stacking is an [ensemble learning](https://courses.analyticsvidhya.com/courses/ensemble-learning-and-ensemble-learning-techniques?utm_source=blog&utm_medium=comprehensive-guide-for-ensemble-models) technique that uses predictions from multiple models (for example decision tree, knn or svm) to build a new model.
* This model is used for making predictions on the test set.

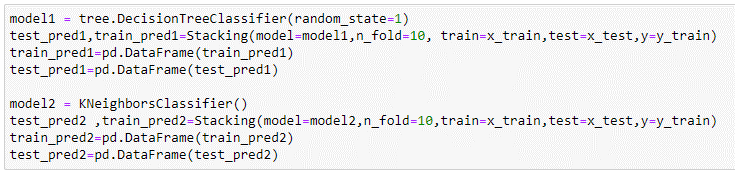
1. The train set is split into 10 parts.
2. A base model (suppose a decision tree) is fitted on 9 parts and predictions are made for the 10th part. This is done for each part of the train set.
3. The base model (in this case, decision tree) is then fitted on the whole train dataset.
4. Using this model, predictions are made on the test set.
5. Steps 2 to 4 are repeated for another base model (say knn) resulting in another set of predictions for the train set and test set.
6. The predictions from the train set are used as features to build a new model.
7. This model is used to make final predictions on the test prediction set.

**Sample code:**

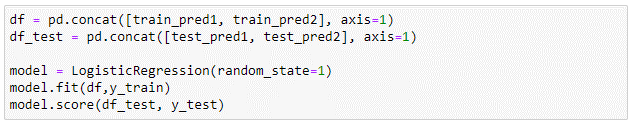
We first define a function to make predictions on n-folds of train and test dataset. This function returns the predictions for train and test for each model.



Now we’ll create two base models – decision tree and knn.



Create a third model, logistic regression, on the predictions of the decision tree and knn models.



We can also create multiple levels in a stacking model.

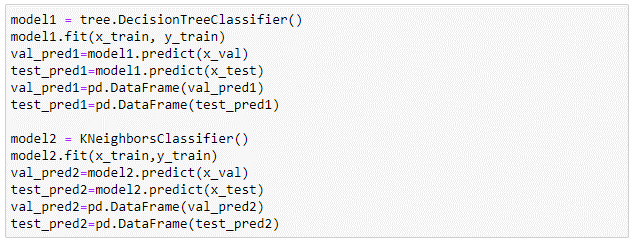
### 2. Blending

* Blending follows the same approach as stacking but uses only a holdout (validation) set from the train set to make predictions.
* In other words, unlike stacking, the predictions are made on the holdout set only.
* The holdout set and the predictions are used to build a model which is run on the test set.
* Here is a detailed explanation of the blending process:

1. The train set is split into training and validation sets.
2. Model(s) are fitted on the training set.
3. The predictions are made on the validation set and the test set.
4. The validation set and its predictions are used as features to build a new model.
5. This model is used to make final predictions on the test and meta-features.

**Sample Code:**

We’ll build two models, decision tree and knn, on the train set in order to make predictions on the validation set.

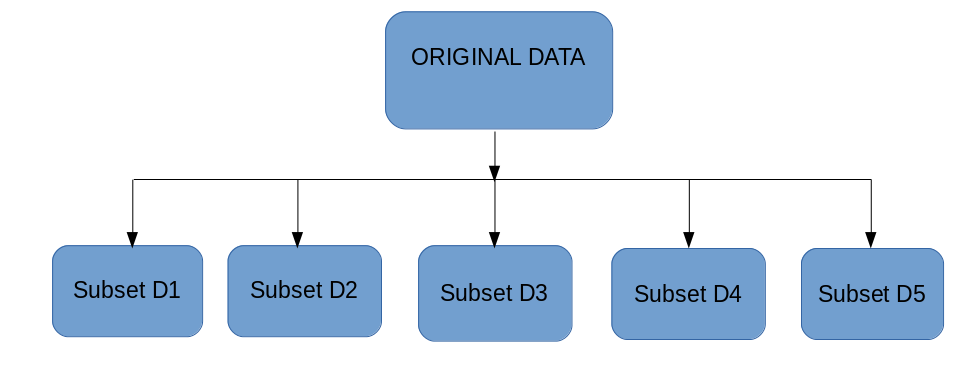


Combining the meta-features and the validation set, a logistic regression model is built to make predictions on the test set.

### 

### 3. Bagging

* The idea behind bagging is combining the results of multiple models (for instance, all decision trees) to get a generalized result.
* 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**.
* 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.

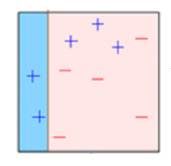


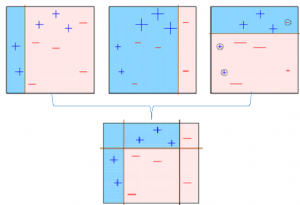
1. Multiple subsets are created from the original dataset, selecting observations with replacement.
2. A base model (weak model) is created on each of these subsets.
3. The models run in parallel and are independent of each other.
4. The final predictions are determined by combining the predictions from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

### 

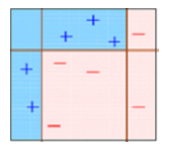
### 4. Boosting

* If a data point is incorrectly predicted by the first model, and then the next (probably all models), will combining the predictions provide better results? Such situations are taken care of by boosting.
* Boosting is a sequential process, where each subsequent model attempts to correct the errors of the previous model. The succeeding models are dependent on the previous model. Let’s understand the way boosting works in the below steps.

1. A subset is created from the original dataset.
2. Initially, all data points are given equal weights.
3. A base model is created on this subset.
4. This model is used to make predictions on the whole dataset.  
   
5. Errors are calculated using the actual values and predicted values.
6. The observations which are incorrectly predicted, are given higher weights. (Here, the three misclassified blue-plus points will be given higher weights)
7. Another model is created and predictions are made on the dataset. (This model tries to correct the errors from the previous model)
8. Similarly, multiple models are created, each correcting the errors of the previous model.
9. The final model (strong learner) is the weighted mean of all the models (weak learners).



1. Thus, the boosting algorithm combines a number of weak learners to form a strong learner. The individual models would not perform well on the entire dataset, but they work well for some part of the dataset. Thus, each model actually boosts the performance of the ensemble.



# **Random Forest Algorithm**

* Random Forest is a popular machine learning algorithm that belongs to the supervised learning technique.
* It is based on the concept of **ensemble learning,** which is a process of combining multiple classifiers to solve a complex problem and to improve the performance of the model.
* **"Random Forest is a classifier that contains a number of decision trees on various subsets of the given dataset and takes the average to improve the predictive accuracy of that dataset."**
* Instead of relying on one decision tree, the random forest takes the prediction from each tree and based on the majority votes of predictions, and it predicts the final output.
* **The greater number of trees in the forest leads to higher accuracy and prevents the problem of overfitting.**



## How does Random Forest algorithm work?

Random Forest works in two-phase first is to create the random forest by combining N decision tree, and second is to make predictions for each tree created in the first phase.

The Working process can be explained in the below steps and diagram:

**Step-1:** Select random K data points from the training set.

**Step-2:** Build the decision trees associated with the selected data points (Subsets).

**Step-3:** Choose the number N for decision trees that you want to build.

**Step-4:** Repeat Step 1 & 2.

**Step-5:** For new data points, find the predictions of each decision tree, and assign the new data points to the category that wins the majority votes.

## Applications of Random Forest

There are mainly four sectors where Random forest mostly used:

1. **Banking:** Banking sector mostly uses this algorithm for the identification of loan risk.
2. **Medicine:** With the help of this algorithm, disease trends and risks of the disease can be identified.
3. **Land Use:** We can identify the areas of similar land use by this algorithm.
4. **Marketing:** Marketing trends can be identified using this algorithm.

## Advantages of Random Forest

* Random Forest is capable of performing both Classification and Regression tasks.
* It is capable of handling large datasets with high dimensionality.
* It enhances the accuracy of the model and prevents the overfitting issue.

## Disadvantages of Random Forest

* Although random forest can be used for both classification and regression tasks, it is not more suitable for Regression tasks.