**6. Supervised Learning**

**6.1. Introduction**

Supervised learning is commonly used in real world applications, such as face and speech recognition, products or movie recommendations, and sales forecasting.

Supervised learning involves building a machine learning model that is based on **labeled samples**. For example, if we build a system to estimate the price of a plot of land or a house based on various features, such as size, location, and so on, we first need to create a database and label it. We need to teach the algorithm what features correspond to what prices. Based on this data, the algorithm will learn how to calculate the price of real estate using the values of the input features.

Supervised learning deals with learning a function from available training data. Here, a learning algorithm analyzes the training data and produces a derived function that can be used for mapping new examples.

There are many **supervised learning algorithms** such as

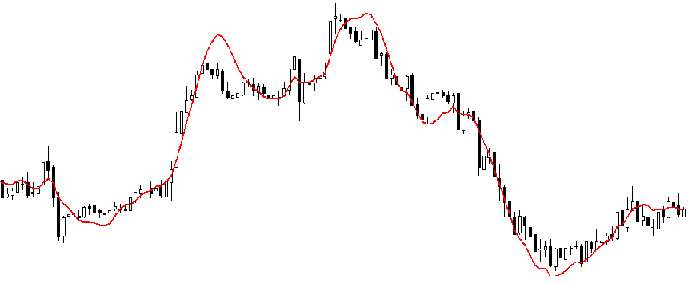
1. Regression
2. Classification
3. Decision Tree
4. KNN
5. Random Forest
6. Logistic Regression
7. Neural networks
8. Support Vector Machines (SVMs)
9. Naive Bayes classifiers

**6.2. Regression**

**When Regression is chosen?**  
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 hyperplane which goes through the points.

**What is Regression Analysis?**

Regression analysis is a form of predictive modeling technique which investigates the relationship between a **dependent**(target) and **independent variable (s)** (predictor). This technique is used for forecasting, time series modeling and finding the [causal effect relationship](https://www.analyticsvidhya.com/blog/2015/06/establish-causality-events/) between the variables. For example, relationship between rash driving and number of road accidents by a driver is best studied through regression.



Regression analysis is an important tool for modelling and analysing data. Here, we fit a curve / line to the data points, in such a manner that the differences between the distances of data points from the curve or line is minimized.

## Why do we use Regression Analysis?

As mentioned above, regression analysis estimates the relationship between two or more variables. Let’s understand this with an easy example:

Let’s say, you want to estimate growth in sales of a company based on current economic conditions. You have the recent company data which indicates that the growth in sales is around two and a half times the growth in the economy. Using this insight, we can predict future sales of the company based on current & past information.

There are multiple benefits of using regression analysis. They are as follows:

1. It indicates the **significant relationships** between dependent variable and independent variable.
2. It indicates the **strength of impact** of multiple independent variables on a dependent variable.

Regression analysis also allows us to compare the effects of variables measured on different scales, such as the effect of price changes and the number of promotional activities. These benefits help market researchers / data analysts / data scientists to eliminate and evaluate the best set of variables to be used for building predictive models.

**Types of Regression –**

* Linear regression
* Logistic regression
* Polynomial regression
* Stepwise regression
* Stepwise regression
* Ridge regression
* Lasso regression

[**Linear regression**](https://www.geeksforgeeks.org/ml-linear-regression/)is used for predictive analysis. Linear regression is a linear approach for modeling the relationship between the criterion or the scalar response and the multiple predictors or explanatory variables. Linear regression focuses on the conditional probability distribution of the response given the values of the predictors. For linear regression, there is a danger of overfitting. The formula for linear regression is: Y’ = bX + A.

[**Logistic regression**](https://www.geeksforgeeks.org/understanding-logistic-regression/) is used when the dependent variable is dichotomous. Logistic regression estimates the parameters of a logistic model and is form of binomial regression. Logistic regression is used to deal with data that has two possible criterions and the relationship between the criterions and the predictors. The equation for logistic regression is: l = .

[**Polynomial regression**](https://www.geeksforgeeks.org/python-implementation-of-polynomial-regression/) is used for curvilinear data. Polynomial regression is fit with the method of least squares. The goal of regression analysis to model the expected value of a dependent variable y in regards to the independent variable x. The equation for polynomial regression is: l = .

**Stepwise regression** is used for fitting regression models with predictive models. It is carried out automatically. With each step, the variable is added or subtracted from the set of explanatory variables. The approaches for stepwise regression are forward selection, backward elimination, and bidirectional elimination. The formula for stepwise regression is .

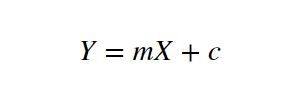
**Ridge regression** is a technique for analyzing multiple regression data. When multicollinearity occurs, least squares estimates are unbiased. A degree of bias is added to the regression estimates, and a result, ridge regression reduces the standard errors. The formula for ridge regression is .

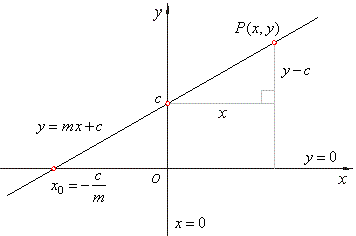
**Lasso regression** is a regression analysis method that performs both variable selection and regularization. Lasso regression uses soft thresholding. Lasso regression selects only a subset of the provided covariates for use in the final model. Lasso regression is .

**6.2.1. Linear Models**

In statistics, linear regression is a linear model approach to modelling the relationship between a dependent variable and one or more independent variables. In the case of one independent variable it is called simple linear regression. For more than one independent variable, the process is called multiple linear regression.

Let **X** be the independent variable and **Y** be the dependent variable. We will define a linear relationship between these two variables as follows:





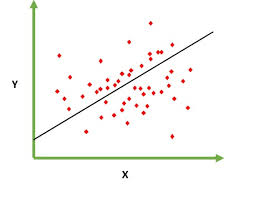
# Least Squares method

Least Square method is used to find the best fit line in the linear model.

In regression for the given data set we find the best line which can fit the data samples.

There are number of possibilities to draw the regression line, but to get a best fit line number of trails are to be made.

Least square method is one of the important statistical techniques used to find the regression line or best fit for the given model. It is used to compare relationship between two variables.



Simple linear regression

**Steps to be followed for least square method are**

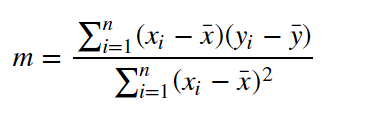
Consider the data set as (x1, y1) (x2, y2) (x3, y3)…. (xn, yn)

**Step 1 :** Calculate the mean of the sample for x and y as,





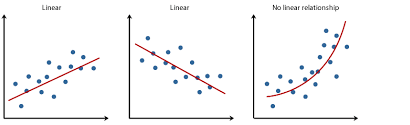
**Step 2 :** Calculate the slope of the line for best fit using the given equation.



**Step 3 :** Compute the y-intercept of the line.

**Step 4 :** Use the slope m and the y-intercept b to form the Equation of the line.

The given figure shows linear and non-linear plots for plotting the dataset. Fitting data on the line is the linear model where as fitting data on the curve is non linear model.



1. Linear Model b) Non-Linear Model

**6.2.2. Regression Trees**

Decision tree is a tree like structure which breaks down the dataset into smaller subsets. It is an iterative process and on every iteration tree increment itself. Decision tree mainly consists of nodes and links, nodes are further of two types decision nodes and leaf nodes.

Decision nodes are the nodes which on some condition are further divided into various other decision nodes or leaf nodes Leaf nodes are also known as terminals and they cannot be further divided. Decision trees can handle both categorical as well as numerical data.

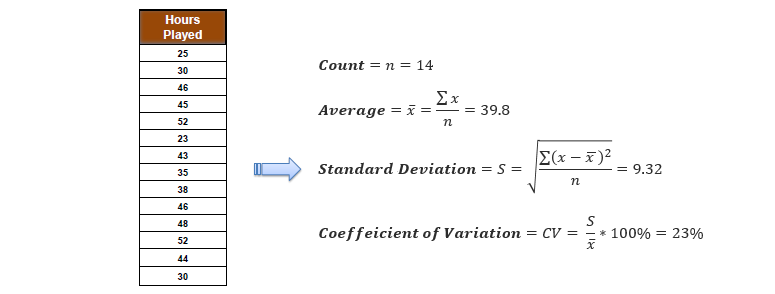
Decision tree when developed for regression is known as regression tree. Decision tree can be developed for the classification also and main difference in both is in regression tree standard deviation reduction is donw where as in classification tree gain is calculated.

Consider the Table given below to understand the decision tree.

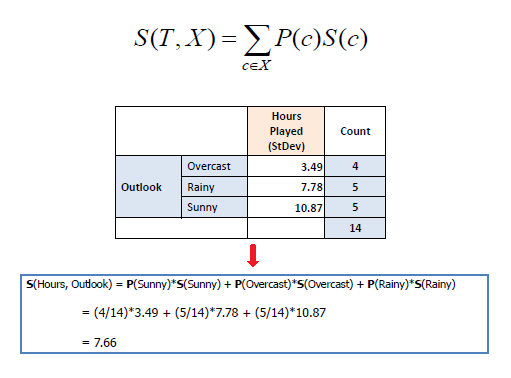
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Day** | **Outlook** | **Temp.** | **Humidity** | **Wind** | **Golf Players** |
| 1 | Sunny | Hot | High | Weak | 25 |
| 2 | Sunny | Hot | High | Strong | 30 |
| 3 | Overcast | Hot | High | Weak | 46 |
| 4 | Rain | Mild | High | Weak | 45 |
| 5 | Rain | Cool | Normal | Weak | 52 |
| 6 | Rain | Cool | Normal | Strong | 23 |
| 7 | Overcast | Cool | Normal | Strong | 43 |
| 8 | Sunny | Mild | High | Weak | 35 |
| 9 | Sunny | Cool | Normal | Weak | 38 |
| 10 | Rain | Mild | Normal | Weak | 46 |
| 11 | Sunny | Mild | Normal | Strong | 48 |
| 12 | Overcast | Mild | High | Strong | 52 |
| 13 | Overcast | Hot | Normal | Weak | 44 |
| 14 | Rain | Mild | High | Strong | 30 |

Standard deviation is used for calculating the homogeneity of a numerical sample, in case if in any problem all numerical samples are homogenous then the standard deviation is zero. Standard deviation can be calculated for single or multiple attributes.

**Standard deviation for single attribute**



**Standard deviation for multiple attribute**



Now let us see the standard deviation reduction. Standard deviation reduction is generally based on the low value of standard deviation once a dataset is split on an attribute. Here the aim to find a decision tree is to identify the attribute which returns the highest value of standard deviation reduction.

**Following are the steps carried out for forming the tree**

**Step 1 : Calculate the standard deviation of the target.**

Golf players = {25, 30, 46, 45, 52, 23, 43, 35, 38, 46, 48, 52, 44, 30}

Average of golf players = (25 + 30 + 46 + 45 + 52 + 23 + 43 + 35 + 38 + 46 + 48 + 52 + 44 + 30  
)/14 = 39.78

Standard deviation of golf players =  √[( (25 – 39.78)2 + (30 – 39.78)2 + (46 – 39.78)2 + … + (30 – 39.78)2 )/14] = 9.32

**Step 2 : Calculate the standard deviation for each branch, and then find the standard deviation reduction.**

### For Outlook

Outlook can be sunny, overcast and rain. We need to calculate standard deviation of golf players for all of these outlook candidates.

#### Sunny outlook

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Day | Outlook | Temp. | Humidity | Wind | Golf Players |
| 1 | Sunny | Hot | High | Weak | 25 |
| 2 | Sunny | Hot | High | Strong | 30 |
| 8 | Sunny | Mild | High | Weak | 35 |
| 9 | Sunny | Cool | Normal | Weak | 38 |
| 11 | Sunny | Mild | Normal | Strong | 48 |

Golf players for sunny outlook = {25, 30, 35, 38, 48}

Average of golf players for sunny outlook = (25+30+35+38+48)/5 = 35.2

Standard deviation of golf players for sunny outlook = √(((25 – 35.2)2 + (30 – 35.2)2 + … )/5) = 7.78

#### Overcast outlook

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Day | Outlook | Temp. | Humidity | Wind | Golf Players |
| 3 | Overcast | Hot | High | Weak | 46 |
| 7 | Overcast | Cool | Normal | Strong | 43 |
| 12 | Overcast | Mild | High | Strong | 52 |
| 13 | Overcast | Hot | Normal | Weak | 44 |

Golf players for overcast outlook = {46, 43, 52, 44}

Average of golf players for overcast outlook = (46 + 43 + 52 + 44)/4 = 46.25

Standard deviation of golf players for overcast outlook = √(((46-46.25)2+(43-46.25)2+…)= 3.49

#### Rainy outlook

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Day | Outlook | Temp. | Humidity | Wind | Golf Players |
| 4 | Rain | Mild | High | Weak | 45 |
| 5 | Rain | Cool | Normal | Weak | 52 |
| 6 | Rain | Cool | Normal | Strong | 23 |
| 10 | Rain | Mild | Normal | Weak | 46 |
| 14 | Rain | Mild | High | Strong | 30 |

Golf players for overcast outlook = {45, 52, 23, 46, 30}

Average of golf players for overcast outlook = (45+52+23+46+30)/5 = 39.2

Standard deviation of golf players for rainy outlook = √(((45 – 39.2)2+(52 – 39.2)2+…)/5)=10.87

### Summarizing standard deviations for the outlook feature

|  |  |  |
| --- | --- | --- |
| Outlook | Stdev of Golf Players | Instances |
| Overcast | 3.49 | 4 |
| Rain | 10.87 | 5 |
| Sunny | 7.78 | 5 |

Weighted standard deviation for outlook = (4/14)x3.49 + (5/14)x10.87 + (5/14)x7.78 = 7.66

You might remember that we have calculated the global standard deviation of golf players 9.32 in previous steps. Standard deviation reduction is difference of the global standard deviation and standard deviation for current feature. In this way, maximized standard deviation reduction will be the decision node.

Standard deviation reduction for outlook = 9.32 – 7.66 = 1.66

**Note : likewise we will calculate for the rest one.**

**For Temperature**

### Summarizing standard deviations for temperature feature

|  |  |  |
| --- | --- | --- |
| Temperature | Stdev of Golf Players | Instances |
| Hot | 8.95 | 4 |
| Cool | 10.51 | 4 |
| Mild | 7.65 | 6 |

Weighted standard deviation for temperature = (4/14)x8.95 + (4/14)x10.51 + (6/14)x7.65 = 8.84

Standard deviation reduction for temperature = 9.32 – 8.84 = 0.47

**For Humidity**

### Summarizing standard deviations for humidity feature

|  |  |  |
| --- | --- | --- |
| Humidity | Stdev of Golf Player | Instances |
| High | 9.36 | 7 |
| Normal | 8.73 | 7 |

Weighted standard deviation for humidity = (7/14)x9.36 + (7/14)x8.73 = 9.04

Standard deviation reduction for humidity = 9.32 – 9.04 = 0.27

### For Wind

### Summarizing standard deviations for wind feature

|  |  |  |
| --- | --- | --- |
| Wind | Stdev of Golf Player | Instances |
| Strong | 10.59 | 6 |
| Weak | 7.87 | 8 |

Weighted standard deviation for wind = (6/14)x10.59 + (8/14)x7.87 = 9.03

Standard deviation reduction for wind = 9.32 – 9.03 = 0.29

So, we’ve calculated standard deviation reduction values for all features. The winner is outlook because it has the highest score.

|  |  |
| --- | --- |
| **Feature** | **Standard Deviation Reduction** |
| Outlook | 1.66 |
| Temperature | 0.47 |
| Humidity | 0.27 |
| Wind | 0.29 |

**Step 3 : Choose the decision node with the largest value of standard deviation reduction.**

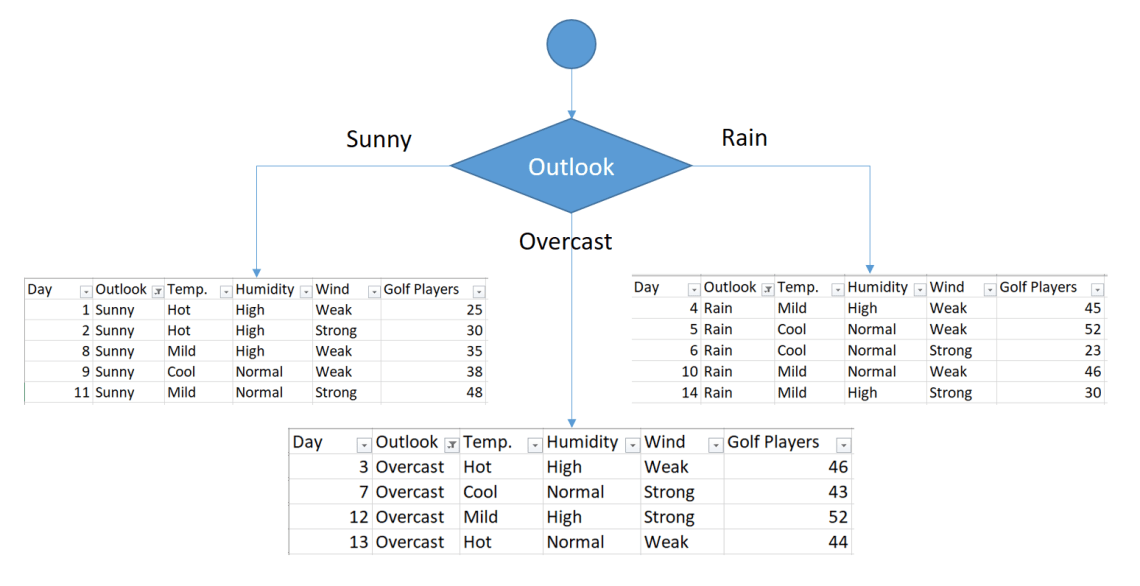
From the above, we would come to know standard deviation reduction for **outlook** is maximum and hence it is selected.

|  |  |  |
| --- | --- | --- |
| Outlook | Stdev of Golf Players | Instances |
| Overcast | 3.49 | 4 |
| Rain | 10.87 | 5 |
| Sunny | 7.78 | 5 |

Standard deviation reduction for outlook = 9.32 – 7.66 = 1.66

**Step 4 : In this step the dataset is divided with respect to the values of selected attribute. It is a recursive process and it is done until complete data is processed.**

We’ll put outlook decision at the top of decision tree. Let’s monitor the new sub data sets for the candidate branches of outlook feature.



### For Sunny Outlook

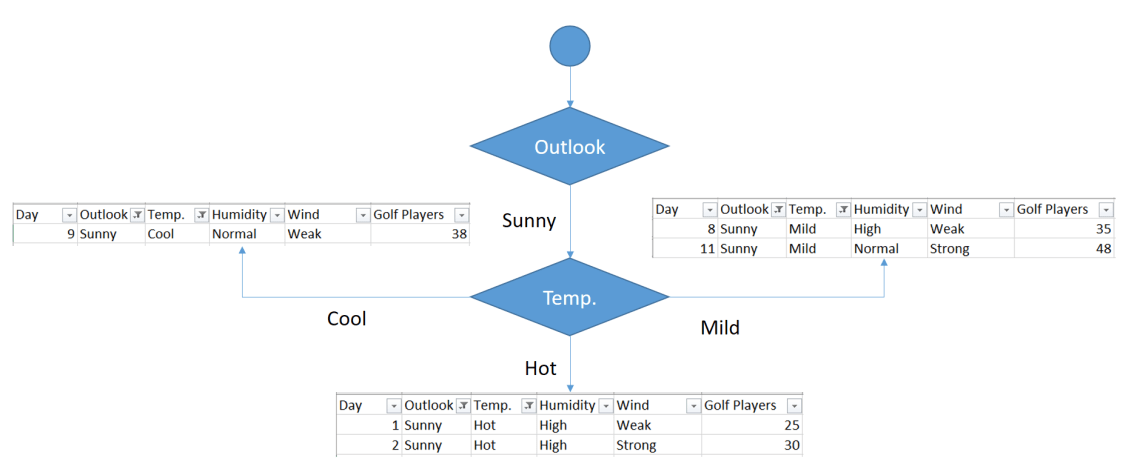
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Day | Outlook | Temp. | Humidity | Wind | Golf Players |
| 1 | Sunny | Hot | High | Weak | 25 |
| 2 | Sunny | Hot | High | Strong | 30 |
| 8 | Sunny | Mild | High | Weak | 35 |
| 9 | Sunny | Cool | Normal | Weak | 38 |
| 11 | Sunny | Mild | Normal | Strong | 48 |

Golf players for sunny outlook = {25, 30, 35, 38, 48}

Standard deviation for sunny outlook = 7.78

We’ve calculated standard deviation reductions for sunny outlook. The winner is temperature.

|  |  |
| --- | --- |
| **Feature** | **Standard Deviation Reduction** |
| Temperature | 4.18 |
| Humidity | 3.33 |
| Wind | 0.85 |



### For Overcast outlook

Overcast outlook branch has already 4 instances in the sub data set. We can terminate building branches for this leaf. Final decision will be average of the following table for overcast outlook.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Day | Outlook | Temp. | Humidity | Wind | Golf Players |
| 3 | Overcast | Hot | High | Weak | 46 |
| 7 | Overcast | Cool | Normal | Strong | 43 |
| 12 | Overcast | Mild | High | Strong | 52 |
| 13 | Overcast | Hot | Normal | Weak | 44 |

If outlook is overcast, then there would be (46+43+52+44)/4 = 46.25 golf players.

### For Rainy Outlook

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Day | Outlook | Temp. | Humidity | Wind | Golf Players |
| 4 | Rain | Mild | High | Weak | 45 |
| 5 | Rain | Cool | Normal | Weak | 52 |
| 6 | Rain | Cool | Normal | Strong | 23 |
| 10 | Rain | Mild | Normal | Weak | 46 |
| 14 | Rain | Mild | High | Strong | 30 |

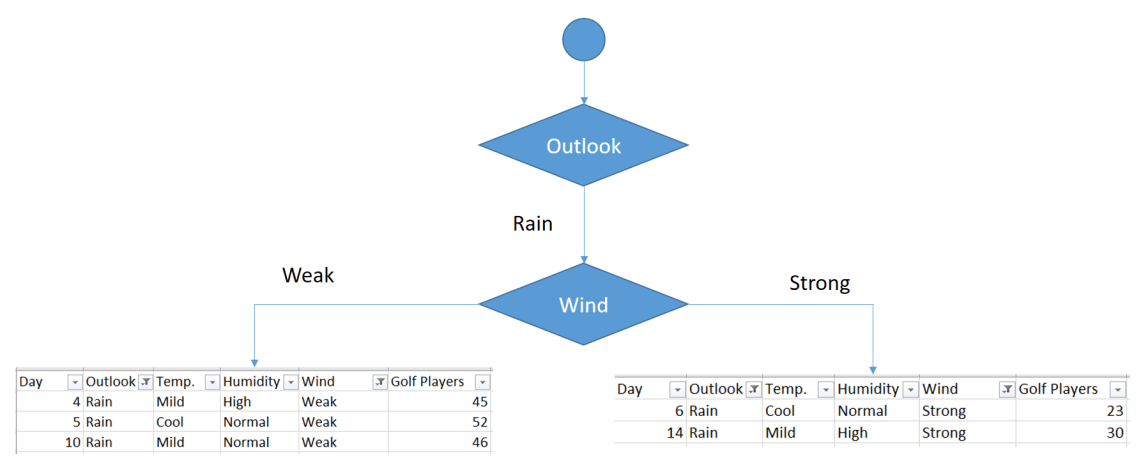
Golf players for sunny outlook = {45,52,23,46,30}

Standard deviation for rainy outlook = 10.87

### Summarizing rainy outlook

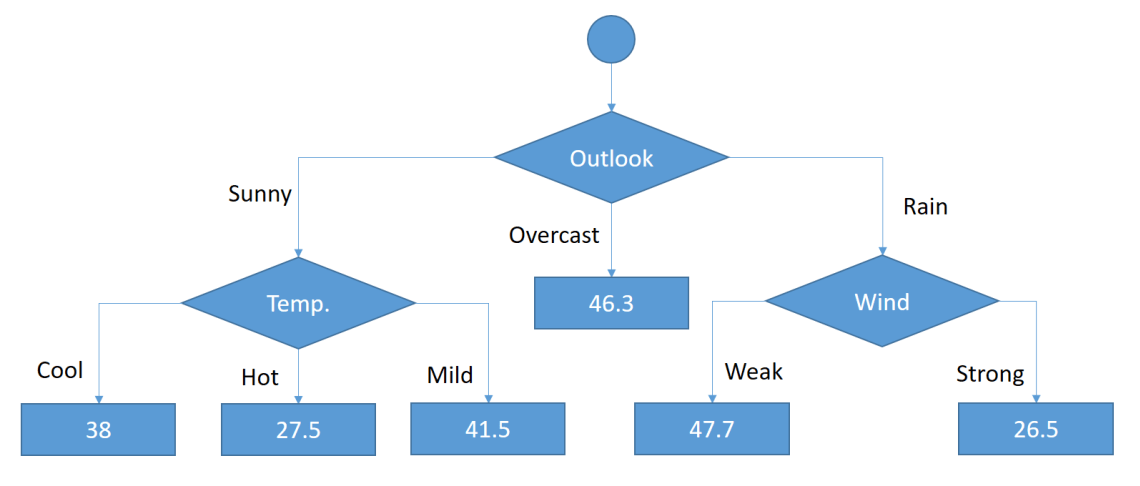
As illustrated below, the winner is wind feature.

|  |  |
| --- | --- |
| **Feature** | **Standard deviation reduction** |
| Temperature | 0.67 |
| Humidity | 0.37 |
| Wind | 7.62 |



As seen, both branches have items less than 5. Now, we can terminate these leafs based on the termination rule.

So, Final form of the decision tree is demonstrated below.



Final form of the regression tree

**6.2.3. Time Series Analysis**

Time series is a sequence of observations of categorical or numeric variables indexed by a date, or timestamp. A clear example of time series data is the time series of a stock price. In the following table, we can see the basic structure of time series data. In this case the observations are recorded every hour.

|  |  |
| --- | --- |
| **Timestamp** | **Stock - Price** |
| 2015-10-11 09:00:00 | 100 |
| 2015-10-11 10:00:00 | 110 |
| 2015-10-11 11:00:00 | 105 |
| 2015-10-11 12:00:00 | 90 |
| 2015-10-11 13:00:00 | 120 |

Normally, the first step in time series analysis is to plot the series, this is normally done with a line chart.

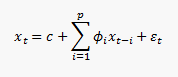
The most common application of time series analysis is forecasting future values of a numeric value using the temporal structure of the data. This means, the available observations are used to predict values from the future.

The temporal ordering of the data, implies that traditional regression methods are not useful. In order to build robust forecast, we need models that take into account the temporal ordering of the data.

The most widely used model for Time Series Analysis is called **Autoregressive Moving Average** (ARMA). The model consists of two parts, an **autoregressive** (AR) part and a **moving average** (MA) part. The model is usually then referred to as the *ARMA(p, q)* model where *p* is the order of the autoregressive part and *q* is the order of the moving average part.

## Autoregressive Model

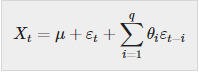
The *AR(p)* is read as an autoregressive model of order p. Mathematically it is written as −



where {φ1, …, φp} are parameters to be estimated, c is a constant, and the random variable εt represents the white noise. Some constraints are necessary on the values of the parameters so that the model remains stationary.

## Moving Average

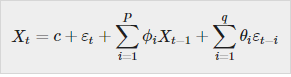
The notation *MA(q)* refers to the moving average model of order *q* −

****

where the θ1, ..., θq are the parameters of the model, μ is the expectation of Xt, and the εt, εt − 1, ... are, white noise error terms.

## Autoregressive Moving Average

The *ARMA(p, q)* model combines p autoregressive terms and q moving-average terms. Mathematically the model is expressed with the following formula −

****

We can see that the *ARMA(p, q)* model is a combination of *AR(p)* and *MA(q)* models.

To give some intuition of the model consider that the AR part of the equation seeks to estimate parameters for Xt − i observations of in order to predict the value of the variable in Xt. It is in the end a weighted average of the past values. The MA section uses the same approach but with the error of previous observations, εt − i. So in the end, the result of the model is a weighted average.

**Time Series Components**

1. **Trend**

It is considered to behaviour of the feature at a particular amount of time, it can be categorized as increasing trend, decreasing trend or constant trend.

When the particular feature value increases in particular amount of time it is increasing trend, similarly if it decreases it is decreasing trend and when it does not change over the period of time then it is constant trend.

The population, agricultural production, items manufactured, number of births and deaths, number of industry or any factory, number of schools or colleges are some of its example showing some kind of tendencies of movement.

### ****Linear and Non-Linear Trend****

If we plot the time series values on a graph in accordance with time t. The pattern of the data clustering shows the type of trend. If the set of data cluster more or less round a straight line, then the trend is linear otherwise it is non-linear (Curvilinear).

1. **Seasonality**

Seasonality operates in a regular and periodic manner. Seasonality is a pattern which repeats at the constant frequency. For example, demand for umbrellas will be in rainy season only.

1. **Cycles**

Cycles are type of seasonality pattern but it does not repeat at regular frequency. Cycle can be generally considered as the task completion time. One complete period is a cycle. This cyclic movement is sometimes called the ‘Business Cycle’.

It is a four-phase cycle comprising of the phases of prosperity, recession, depression, and recovery. The cyclic variation may be regular are not periodic.

1. **Random or Irregular moments**

There is another factor which causes the variation in the variable under study. They are not regular variations and are purely random or irregular. These fluctuations are unforeseen, uncontrollable, unpredictable, and are erratic. These forces are earthquakes, wars, flood, famines, and any other disasters.

**6.2.4. Forecasting**

It is the process of making predictions of the future based on the present and the past data most commonly by analysis of trends. Predictions is similar term to the forecasting but not exactly the same.

**Steps involved in forecasting are as follows**

1. **Identify the Problem**
2. **Collect the required Information**
3. **Perform a Preliminary Analysis**
4. **Choose the Forecasting Model**
5. **Data analysis**
6. **Verify Model Performance**

**Forecasting Methods :**

##### **Qualitative Methods**

These are subjective and are based on the judgment and opinion of experts or consumers. We use them no past data is available.

People use qualitative methods for making medium-to-long-range decisions. Market research is a type of qualitative forecasting method.

##### **Quantitative Methods**

With quantitative methods, we forecast future data as a function of past data. These methods are appropriate when we have past numerical data.

They are also appropriate when we can reasonably assume that some of the data patterns are likely to continue in the future.

We generally use quantitative methods for making short-term and medium-term decisions.

##### **Average Method**

Forecasts of all future values equal the mean of the historical data. This method is appropriate for any type of data where past data is available.

**6.3. Classification**

Supervised learning is a type of machine learning technique in which a teacher or a reference is required.

Classification belongs to supervised learning technique. Classification is considered to be most common task in machine learning. In classification there are finite numbers of classes and these classes can be denoted as C1,C2,…Cn.

Input in the classification can be of any size and the aim of the classification is to map that input or samples to the respective classes. Classification deals with assigning various observations into separate categories.

**Binary Classification**

Binary Classification would generally fall into the domain of Supervised Learning since the training dataset is labeled. And as the name suggests it is simply a special case in which there are only two classes which can be represented as positive and negative or +1 or -1.  
Some typical examples include:

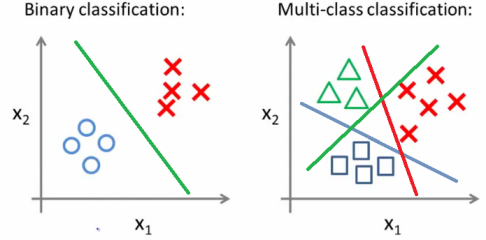
* Credit Card Fraudulent Transaction detection
* Medical Diagnosis
* Spam Detection

Now there are various paradigms that are used for learning binary classifiers which include:

* Decision Trees
* Neural Networks
* Bayesian Classification
* Support Vector Machines

**Multiclass Classification**

In machine learning, multiclass classification also known as multinomial classification is the problem of classifying instances into two or three or more classes. It is popular problem in supervised machine learning.



**6.3.1. Classification Trees**

Decision tree is a tree like structure which breaks down the dataset into smaller subsets. It is an iterative process and on every iteration tree increment itself. Decision tree mainly consists of nodes and links, nodes are further of two types decision nodes and leaf nodes.

Decision nodes are the nodes which on some condition are further divided into various other decision nodes or leaf nodes Leaf nodes are also known as terminals and they cannot be further divided. Decision trees can handle both categorical as well as numerical data.

Decision tree when developed for regression is known as regression tree. Decision tree can be developed for the classification also and main difference in both is in regression tree standard deviation reduction is donw where as in classification tree gain is calculated.

Consider the Table given below to understand the decision tree.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Day** | **Outlook** | **Temp.** | **Humidity** | **Wind** | **Decision** |
| 1 | Sunny | Hot | High | Weak | No |
| 2 | Sunny | Hot | High | Strong | No |
| 3 | Overcast | Hot | High | Weak | Yes |
| 4 | Rain | Mild | High | Weak | Yes |
| 5 | Rain | Cool | Normal | Weak | Yes |
| 6 | Rain | Cool | Normal | Strong | No |
| 7 | Overcast | Cool | Normal | Strong | Yes |
| 8 | Sunny | Mild | High | Weak | No |
| 9 | Sunny | Cool | Normal | Weak | Yes |
| 10 | Rain | Mild | Normal | Weak | Yes |
| 11 | Sunny | Mild | Normal | Strong | Yes |
| 12 | Overcast | Mild | High | Strong | Yes |
| 13 | Overcast | Hot | Normal | Weak | Yes |
| 14 | Rain | Mild | High | Strong | No |

ID3 is a core algorithm used for building decision tree, it uses a top-down approach and also works on the greedy search principal and has no backtracking. To calculate various nodes ID3 uses entropy and information gain method.

We can summarize the ID3 algorithm as illustrated below

Entropy(S) = ∑ – p(I) . log2p(I)

Gain(S, A) = Entropy(S) – ∑ [ p(S|A) . Entropy(S|A) ]

## Entropy

We need to calculate the entropy first. Decision column consists of 14 instances and includes two labels: yes and no. There are 9 decisions labeled yes, and 5 decisions labeled no.

Entropy(Decision) = – p(Yes) . log2p(Yes) – p(No) . log2p(No)

Entropy(Decision) = – (9/14) . log2(9/14) – (5/14) . log2(5/14) = 0.940

Now, we need to find the most dominant factor for decisioning.

## Wind factor on decision

Gain(Decision, Wind) = Entropy(Decision) – ∑ [ p(Decision|Wind) . Entropy(Decision|Wind) ]

Wind attribute has two labels: weak and strong. We would reflect it to the formula.

Gain(Decision, Wind) = Entropy(Decision) – [ p(Decision|Wind=Weak) . Entropy(Decision|Wind=Weak) ] – [ p(Decision|Wind=Strong) . Entropy(Decision|Wind=Strong) ]

Now, we need to calculate (Decision|Wind=Weak) and (Decision|Wind=Strong) respectively.

### Weak wind factor on decision

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Day** | **Outlook** | **Temp.** | **Humidity** | **Wind** | **Decision** |
| 1 | Sunny | Hot | High | Weak | No |
| 3 | Overcast | Hot | High | Weak | Yes |
| 4 | Rain | Mild | High | Weak | Yes |
| 5 | Rain | Cool | Normal | Weak | Yes |
| 8 | Sunny | Mild | High | Weak | No |
| 9 | Sunny | Cool | Normal | Weak | Yes |
| 10 | Rain | Mild | Normal | Weak | Yes |
| 13 | Overcast | Hot | Normal | Weak | Yes |

There are 8 instances for weak wind. Decision of 2 items are no and 6 items are yes as illustrated below.

1- Entropy(Decision|Wind=Weak) = – p(No) . log2p(No) – p(Yes) . log2p(Yes)

2- Entropy(Decision|Wind=Weak) = – (2/8) . log2(2/8) – (6/8) . log2(6/8) = 0.811

Notice that if the number of instances of a class were 0 and total number of instances were n, then we need to calculate -(0/n) . log2(0/n). Here, log(0) would be equal to -∞, and we cannot calculate 0 times ∞. This is a special case often appears in decision tree applications.

### Strong wind factor on decision

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Day** | **Outlook** | **Temp.** | **Humidity** | **Wind** | **Decision** |
| 2 | Sunny | Hot | High | Strong | No |
| 6 | Rain | Cool | Normal | Strong | No |
| 7 | Overcast | Cool | Normal | Strong | Yes |
| 11 | Sunny | Mild | Normal | Strong | Yes |
| 12 | Overcast | Mild | High | Strong | Yes |
| 14 | Rain | Mild | High | Strong | No |

Here, there are 6 instances for strong wind. Decision is divided into two equal parts.

1- Entropy(Decision|Wind=Strong) = – p(No) . log2p(No) – p(Yes) . log2p(Yes)

2- Entropy(Decision|Wind=Strong) = – (3/6) . log2(3/6) – (3/6) . log2(3/6) = 1

Now, we can turn back to Gain(Decision, Wind) equation.

Gain(Decision, Wind) = Entropy(Decision) – [ p(Decision|Wind=Weak) . Entropy(Decision|Wind=Weak) ] – [ p(Decision|Wind=Strong) . Entropy(Decision|Wind=Strong) ] = 0.940 – [ (8/14) . 0.811 ] – [ (6/14). 1] = 0.048

Calculations for wind column is over. Now, we need to apply same calculations for other columns to find the most dominant factor on decision.

## Other factors on decision

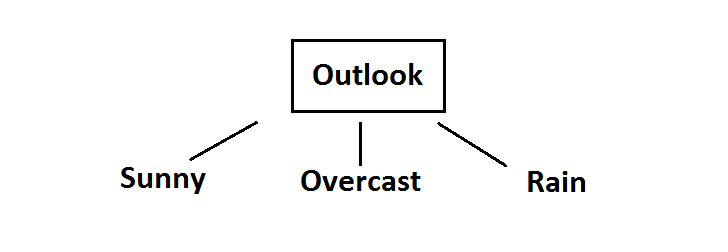
We have applied similar calculation on the other columns.

1- Gain(Decision, Outlook) = 0.246

2- Gain(Decision, Temperature) = 0.029

3- Gain(Decision, Humidity) = 0.151

As seen, outlook factor on decision produces the highest score. That’s why, outlook decision will appear in the root node of the tree.



Root decision on the tree

Now, we need to test dataset for custom subsets of outlook attribute.

## Overcast outlook on decision

Basically, decision will always be yes if outlook were overcast.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Day** | **Outlook** | **Temp.** | **Humidity** | **Wind** | **Decision** |
| 3 | Overcast | Hot | High | Weak | Yes |
| 7 | Overcast | Cool | Normal | Strong | Yes |
| 12 | Overcast | Mild | High | Strong | Yes |
| 13 | Overcast | Hot | Normal | Weak | Yes |

## Sunny outlook on decision

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Day** | **Outlook** | **Temp.** | **Humidity** | **Wind** | **Decision** |
| 1 | Sunny | Hot | High | Weak | No |
| 2 | Sunny | Hot | High | Strong | No |
| 8 | Sunny | Mild | High | Weak | No |
| 9 | Sunny | Cool | Normal | Weak | Yes |
| 11 | Sunny | Mild | Normal | Strong | Yes |

Here, there are 5 instances for sunny outlook. Decision would be probably 3/5 percent no, 2/5 percent yes.

1- Gain(Outlook=Sunny|Temperature) = 0.570

2- Gain(Outlook=Sunny|Humidity) = 0.970

3- Gain(Outlook=Sunny|Wind) = 0.019

Now, humidity is the decision because it produces the highest score if outlook were sunny.

At this point, decision will always be no if humidity were high.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Day** | **Outlook** | **Temp.** | **Humidity** | **Wind** | **Decision** |
| 1 | Sunny | Hot | High | Weak | No |
| 2 | Sunny | Hot | High | Strong | No |
| 8 | Sunny | Mild | High | Weak | No |

On the other hand, decision will always be yes if humidity were normal

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Day** | **Outlook** | **Temp.** | **Humidity** | **Wind** | **Decision** |
| 9 | Sunny | Cool | Normal | Weak | Yes |
| 11 | Sunny | Mild | Normal | Strong | Yes |

Finally, it means that we need to check the humidity and decide if outlook were sunny.

## Rain outlook on decision

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Day** | **Outlook** | **Temp.** | **Humidity** | **Wind** | **Decision** |
| 4 | Rain | Mild | High | Weak | Yes |
| 5 | Rain | Cool | Normal | Weak | Yes |
| 6 | Rain | Cool | Normal | Strong | No |
| 10 | Rain | Mild | Normal | Weak | Yes |
| 14 | Rain | Mild | High | Strong | No |

1- Gain(Outlook=Rain | Temperature)

2- Gain(Outlook=Rain | Humidity)

3- Gain(Outlook=Rain | Wind)

Here, wind produces the highest score if outlook were rain. That’s why, we need to check wind attribute in 2nd level if outlook were rain.

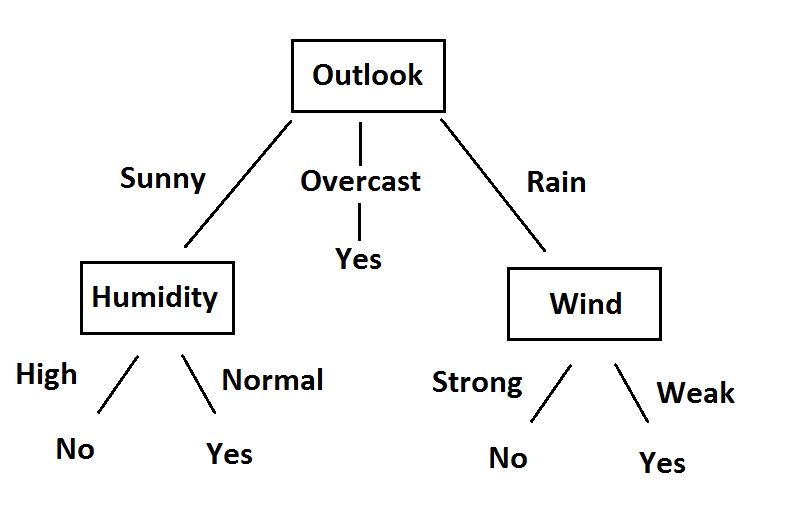
So, it is revealed that decision will always be yes if wind were weak and outlook were rain.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Day** | **Outlook** | **Temp.** | **Humidity** | **Wind** | **Decision** |
| 4 | Rain | Mild | High | Weak | Yes |
| 5 | Rain | Cool | Normal | Weak | Yes |
| 10 | Rain | Mild | Normal | Weak | Yes |

What’s more, decision will be always no if wind were strong and outlook were rain.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Day** | **Outlook** | **Temp.** | **Humidity** | **Wind** | **Decision** |
| 6 | Rain | Cool | Normal | Strong | No |
| 14 | Rain | Mild | High | Strong | No |

So, decision tree construction is over. We can use the following rules for decisioning.



**6.3.4. Logistic regression**

Logistic regression is a statistical method for predicting binary classes. The outcome or target variable is dichotomous in nature. Dichotomous means there are only two possible classes such as positive or negative, 0 or 1, true or false, on or off. For example, it can be used for cancer detection problems. It computes the probability of an event occurrence.

It is a special case of linear regression where the target variable is categorical in nature. It uses a log of odds as the dependent variable. Logistic Regression predicts the probability of occurrence of a binary event utilizing a logit function.

Linear Regression Equation:

http://res.cloudinary.com/dyd911kmh/image/upload/f_auto,q_auto:best/v1534281880/image1_ga8gze.png

Where, y is dependent variable and x1, x2 ... and Xn are explanatory variables.

Sigmoid Function:

http://res.cloudinary.com/dyd911kmh/image/upload/f_auto,q_auto:best/v1534281880/image2_kwxquj.png

Apply Sigmoid function on linear regression:

http://res.cloudinary.com/dyd911kmh/image/upload/f_auto,q_auto:best/v1534281880/image3_qldafx.png

### Linear Regression Vs. Logistic Regression

Linear regression gives you a continuous output, but logistic regression provides a constant output. An example of the continuous output is house price and stock price. Example's of the discrete output is predicting whether a patient has cancer or not, predicting whether the customer will churn. Linear regression is estimated using Ordinary Least Squares (OLS) while logistic regression is estimated using Maximum Likelihood Estimation (MLE) approach.

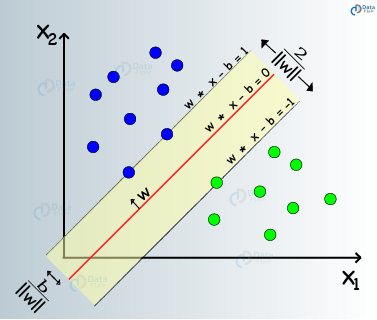


**6.3.5. separating hyperplanes**

Support Vector Machine (SVM) supports the concept of separating hyperplane which is used for creating a classification boundary between two sample set.

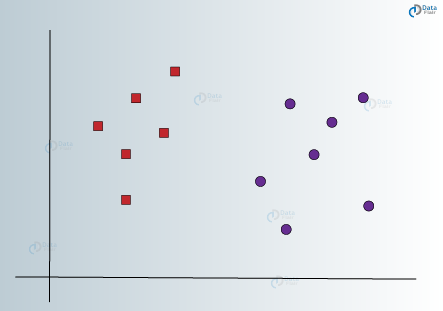
Support Vector Machines are a type of supervised machine learning algorithm that provides analysis of data for classification and regression analysis. While they can be used for regression, SVM is mostly used for classification. We carry out plotting in the n-dimensional space. Value of each feature is also the value of the specific coordinate. Then, we find the ideal hyperplane that differentiates between the two classes.

These support vectors are the coordinate representations of individual observation. It is a frontier method for segregating the two classes.

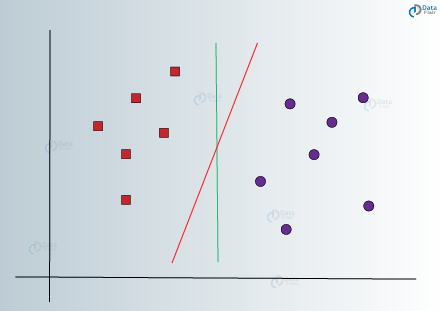


**Working of SVM**

SVM creates a hyperplane that separates the dataset into classes. Suppose that for a given dataset, you have to classify red triangles from blue circles. Your goal is to create a line that classifies the data into two classes, creating a distinction between red triangles and blue circles.



There are many lines that can separates the two classes. Let us visualize some of the lines that can differentiate between the two classes as follows –

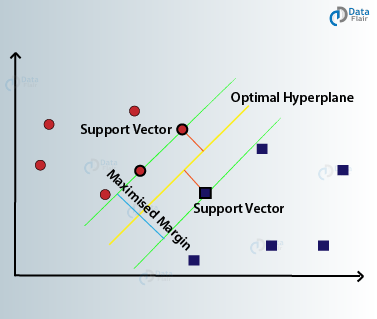


In the above visualizations, we have a green line and a red line.

Which one would be better differentiate the data into two classes?

If you choose the red line, then it is the ideal line that partitions the two classes properly because it lies at same distance from two classes. The green line cannot be the ideal line as it lies too close to the red class.

According to SVM, we have to find the points that lie closest to both the classes. These points are known as support vectors. In the next step, we find the proximity between our dividing plane and the support vectors. The distance between the points and the dividing line is known as margin. The aim of an SVM algorithm is to maximize this very margin. When the margin reaches its maximum, the hyperplane becomes the optimal one.



The SVM model tries to enlarge the distance between the two classes by creating a well-defined decision boundary. In the above case, our hyperplane divided the data. While our data was in 2 dimensions, the hyperplane was of 1 dimension. For higher dimensions, say, an n-dimensional Euclidean Space, we have an n-1 dimensional subset that divides the space into two disconnected components.

**6.3.4. K Nearest Neighbors - Classification**

K-Nearest Neighbors is simple and one of the most basic yet essential classification algorithms in Machine Learning. It belongs to the supervised learning domain and finds intense application in pattern recognition, data mining and intrusion detection.

KNN stands for K-Nearest Neighbors. K is the number of neighbors in KNN.

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).

K nearest neighbor algorithm based on minimum distance from the query instance to the training samples to determine the K-nearest neighbors.

After we gather K Nearest neighbor, we get simple majority of these K-Nearest neighbors to be the prediction of the query instance.

|  |  |  |
| --- | --- | --- |
| **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. |  |  |
|  |  |  |
| https://www.saedsayad.com/images/KNN_similarity.png |  |  |
| 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. |  |  |
| https://www.saedsayad.com/images/KNN_hamming.png |  |  |

Choosing the optimal value for K is best done by first inspecting the data. In general, a large K value is more precise as it reduces the overall noise but there is no guarantee. Cross-validation is another way to retrospectively determine a good K value by using an independent dataset to validate the K value. Historically, the optimal K for most datasets has been between 3-10. That produces much better results than 1NN.

**Advantages and Disadvantages of KNN algorithm.**

**Advantages of KNN**  
  
**1. No Training Period:** KNN is called **Lazy Learner (Instance based learning)**. It does not learn anything in the training period. It does not derive any discriminative function from the training data. In other words, there is no training period for it. It stores the training dataset and learns from it only at the time of making real time predictions. This makes the KNN algorithm much faster than other algorithms that require training e.g. SVM, Linear Regression etc.

**2.** Since the KNN algorithm requires no training before making predictions, **new data can be added seamlessly** which will not impact the accuracy of the algorithm.

**3.** KNN is very **easy to implement**. There are only two parameters required to implement KNN i.e. the value of K and the distance function (e.g. Euclidean or Manhattan etc.)

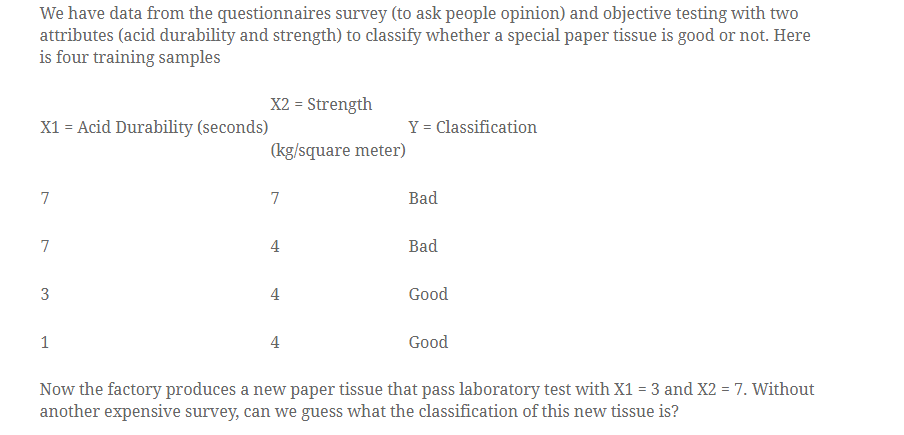
**Disadvantages of KNN**  
  
**1. Does not work well with large dataset:**In large datasets, the cost of calculating the distance between the new point and each existing points is huge which degrades the performance of the algorithm.  
  
**2. Does not work well with high dimensions:**The KNN algorithm doesn't work well with high dimensional data because with large number of dimensions, it becomes difficult for the algorithm to calculate the distance in each dimension.  
  
**3. Need feature scaling:** We need to do feature scaling (standardization and normalization) before applying KNN algorithm to any dataset. If we don't do so, KNN may generate wrong predictions.  
  
**4. Sensitive to noisy data, missing values and outliers**: KNN is sensitive to noise in the dataset. We need to manually impute missing values and remove outliers.

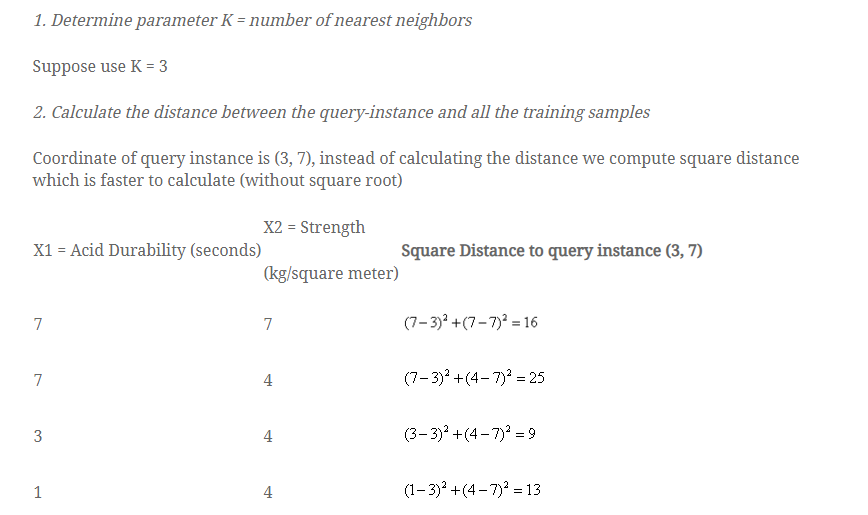
**Numerical Example of K Nearest Neighbor Algorithm**

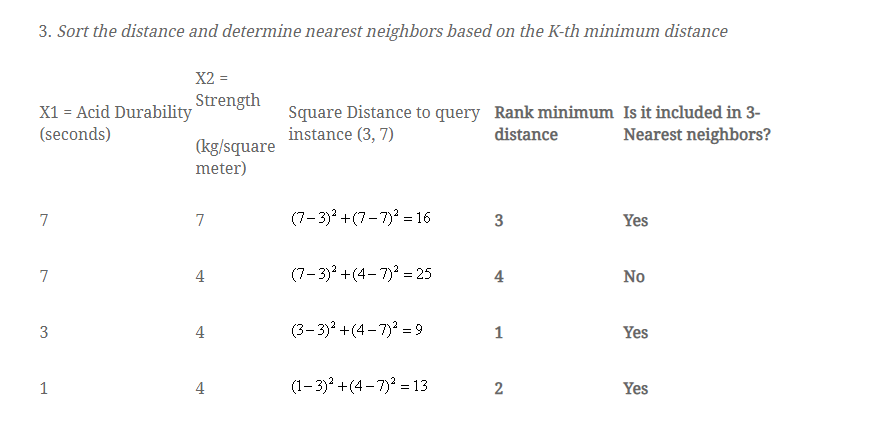
Here is step by step on how to compute K-Nearest neighbors KNN algorithm :

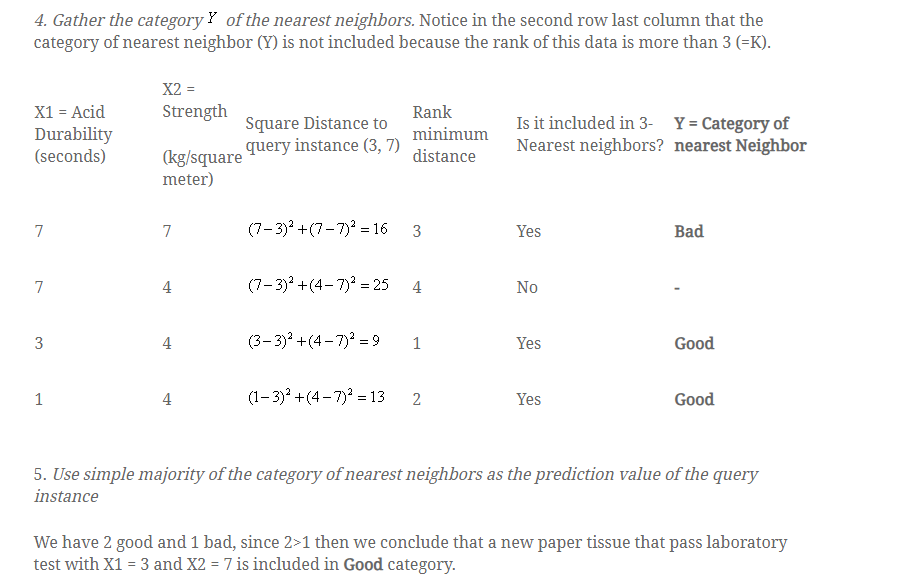
1. Determine parameters K = number of nearest neighbors.
2. Calculate the distance between the query-instance and all the training samples.
3. Sort the distance and determine nearest neighbors based on the K-th minimum distance.
4. Gather the category Y of the nearest neighbors.
5. Use simple majority of the category of nearest neighbors as the prediction value of the query instance.

**Example :**

****

****

****

****

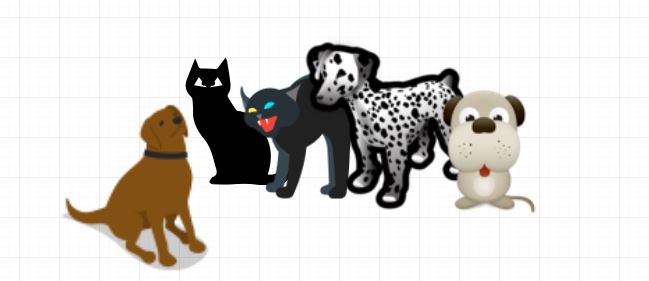
**7. Unsupervised Learning**

**7.1. Introduction**

Unsupervised learning is the training of machine using information that is neither classified nor labelled and allowing the algorithm to act on that information without guidance.

Unlike supervised learning, no teacher is provided that means no training will be given to the machine. Therefore machine is restricted to find the hidden structure in unlabelled data by our-self.

**For instance**, suppose it is given an image having both dogs and cats which have not seen ever.



Thus the machine has no idea about the features of dogs and cat so we can’t categorize it in dogs and cats. But it can categorize them according to their similarities, patterns, and differences i.e., we can easily categorize the above picture into two parts. First first may contain all pictures having **dogs** in it and second part may contain all pictures having **cats** in it. Here you didn’t learn anything before, means no training data or examples.

Unsupervised learning classified into two categories of algorithms:

* **Clustering**: A clustering problem is where you want to discover the inherent groupings in the data, such as grouping customers by purchasing behavior.

**e**.g. k-means , k-medoid

* **Association**: An association rule learning problem is where you want to discover rules that describe large portions of your data, such as people that buy X also tend to buy Y.

**e**.g. Apriori algorithm

**7.2. Principal Component Analysis**

**Principal component analysis** (**PCA**) is used to summarize the information in a data set described by multiple variables.

Note that, the information in a data is the total **variation** it contains.

**PCA reduces the dimensionality** of data containing a large set of variables. This is achieved by transforming the initial variables into a new small set of variables without loosing the most important information in the original data set.

These new variables corresponds to a **linear combination** of the originals and are called **principal components**.

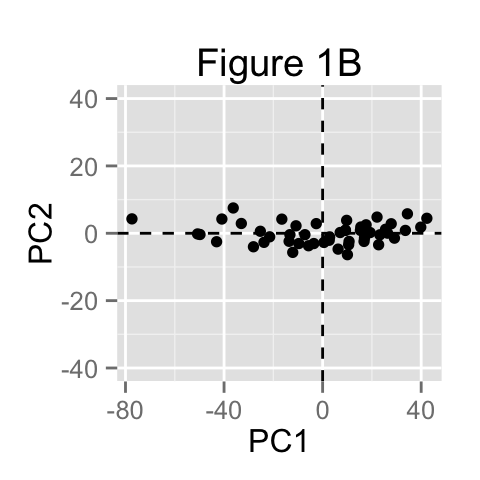
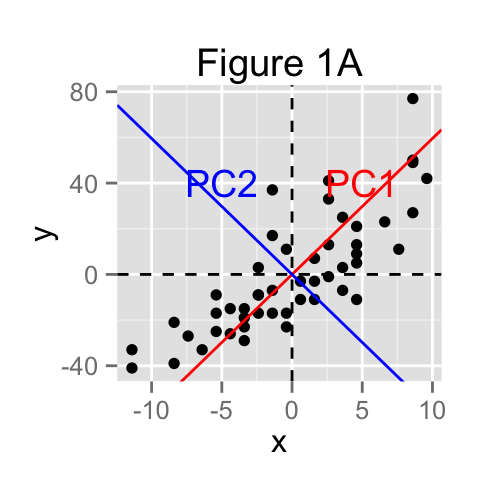
**7.2.1. PCA Basics**

In the Figure 1A below, the data are represented in the X-Y coordinate system. The dimension reduction is achieved by identifying the principal directions, called **principal components**, in which the data varies.

**PCA** assumes that the directions with the largest variances are the most “important” (i.e, the most principal).

In the figure below, the PC1 axis is the **first principal direction** along which the samples show the largest variation. The **PC2 axis** is the **second most important direction** and it is **orthogonal** to the PC1 axis.

The dimensionality of our two-dimensional data can be reduced to a single dimension by projecting each sample onto the first principal component (Figure 1B)

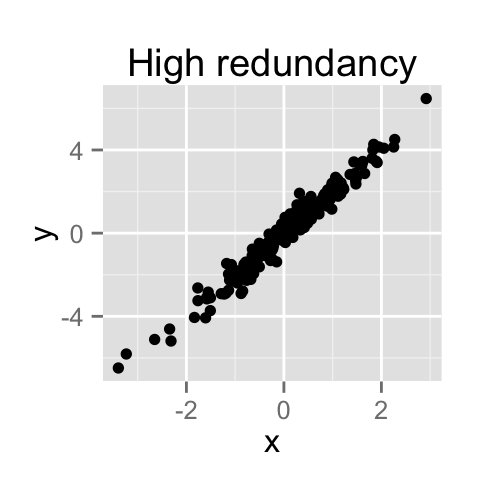
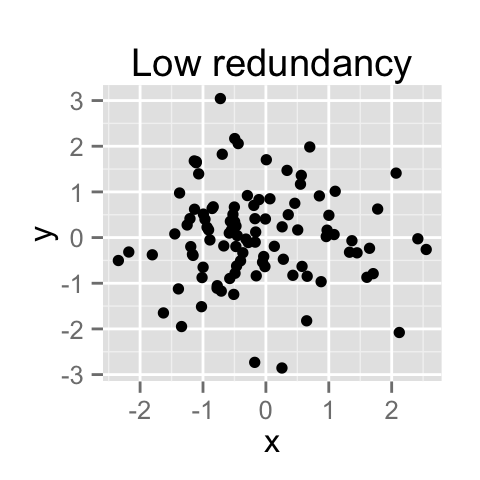


**7.2.2. Main purpose of PCA**

The main goals of **principal component analysis** is :

* to identify hidden pattern in a data set
* to reduce the dimensionnality of the data by removing the noise and redundancy in the data
* to identify correlated variables

PCA method is particularly useful when the variables within the data set are highly correlated.

**Correlation** indicates that there is **redundancy** in the data. Due to this redundancy, PCA can be used to reduce the original variables into a smaller number of new variables ( = **principal components**) explaining most of the variance in the original variables.

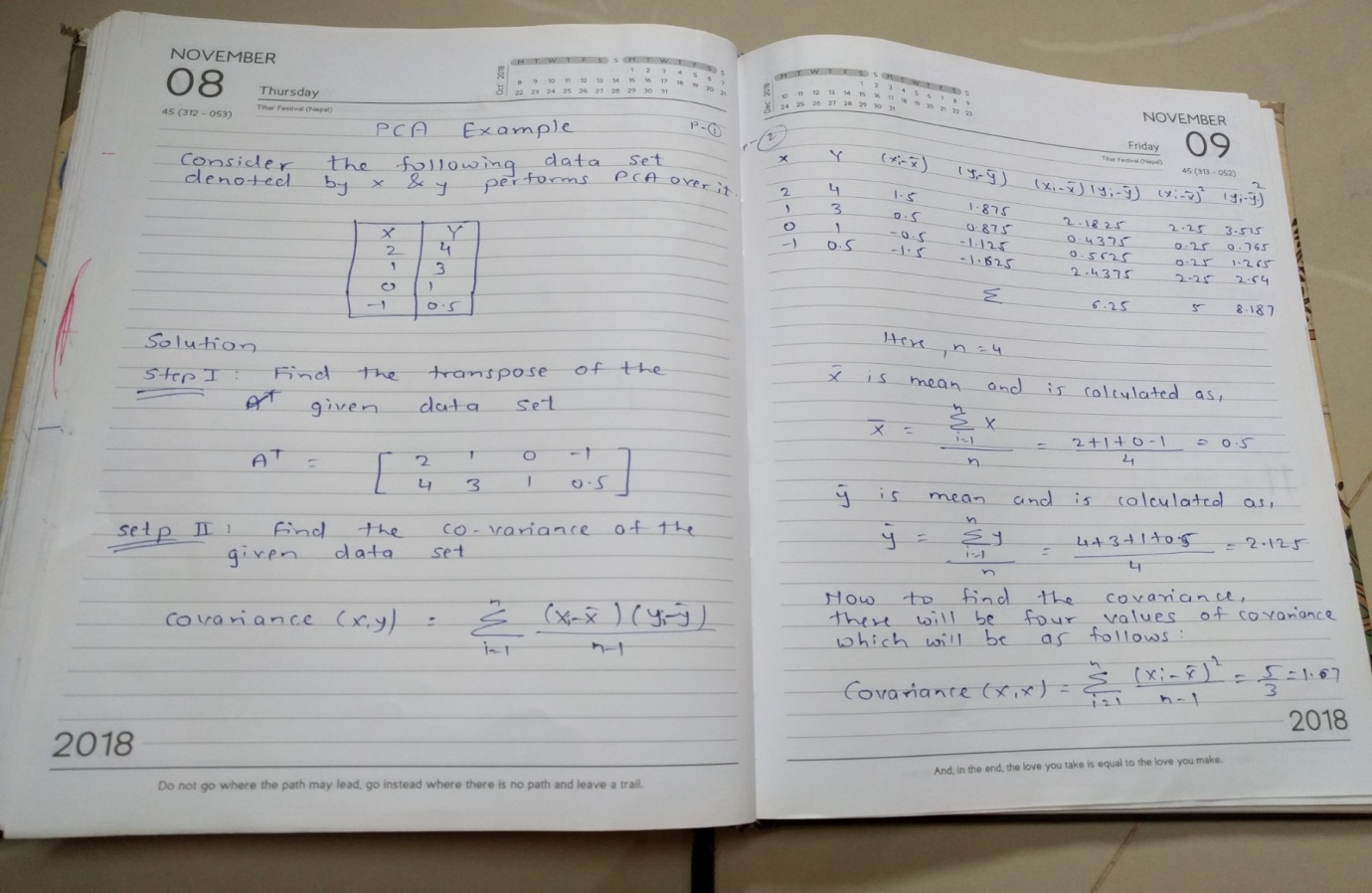
How to remove the redundancy?

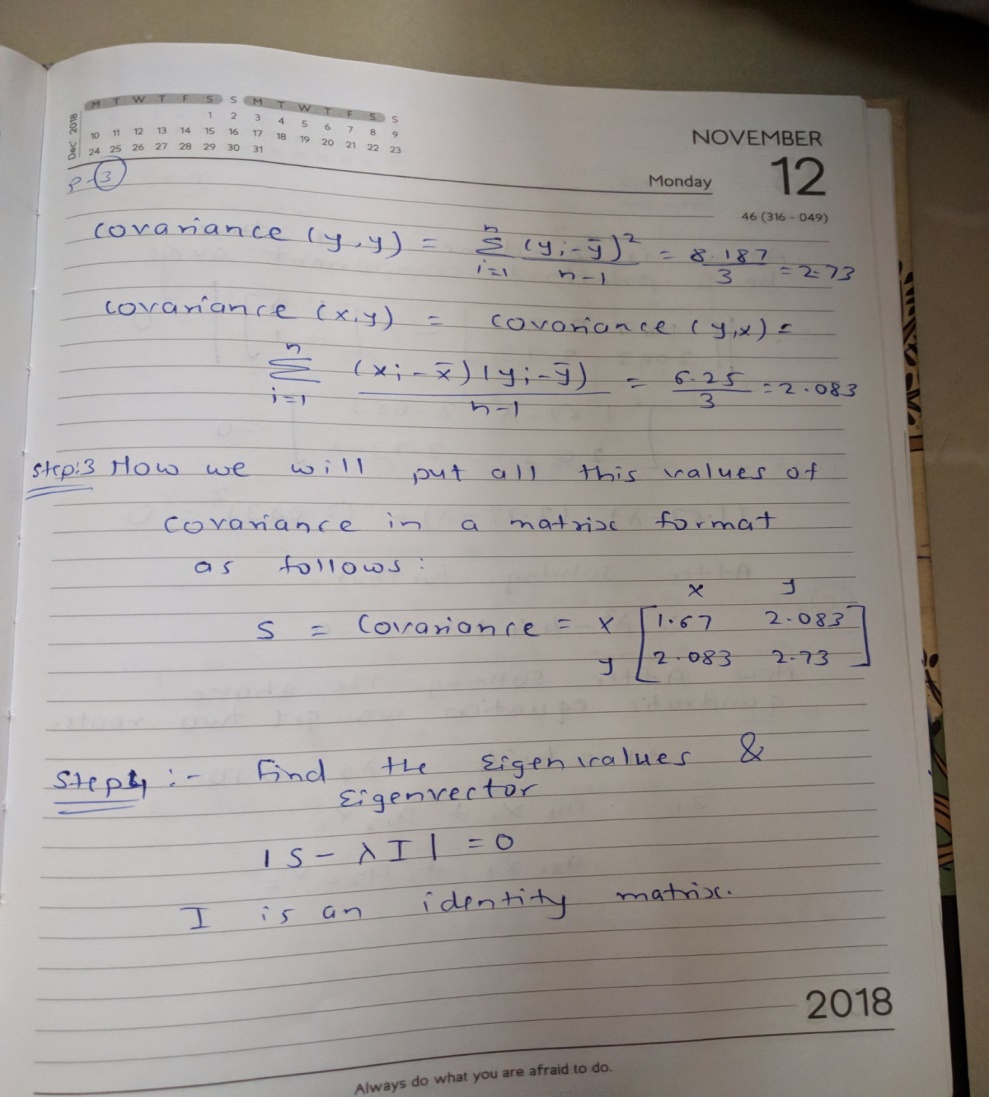
PCA is traditionally performed on covariance matrix or correlation matrix.

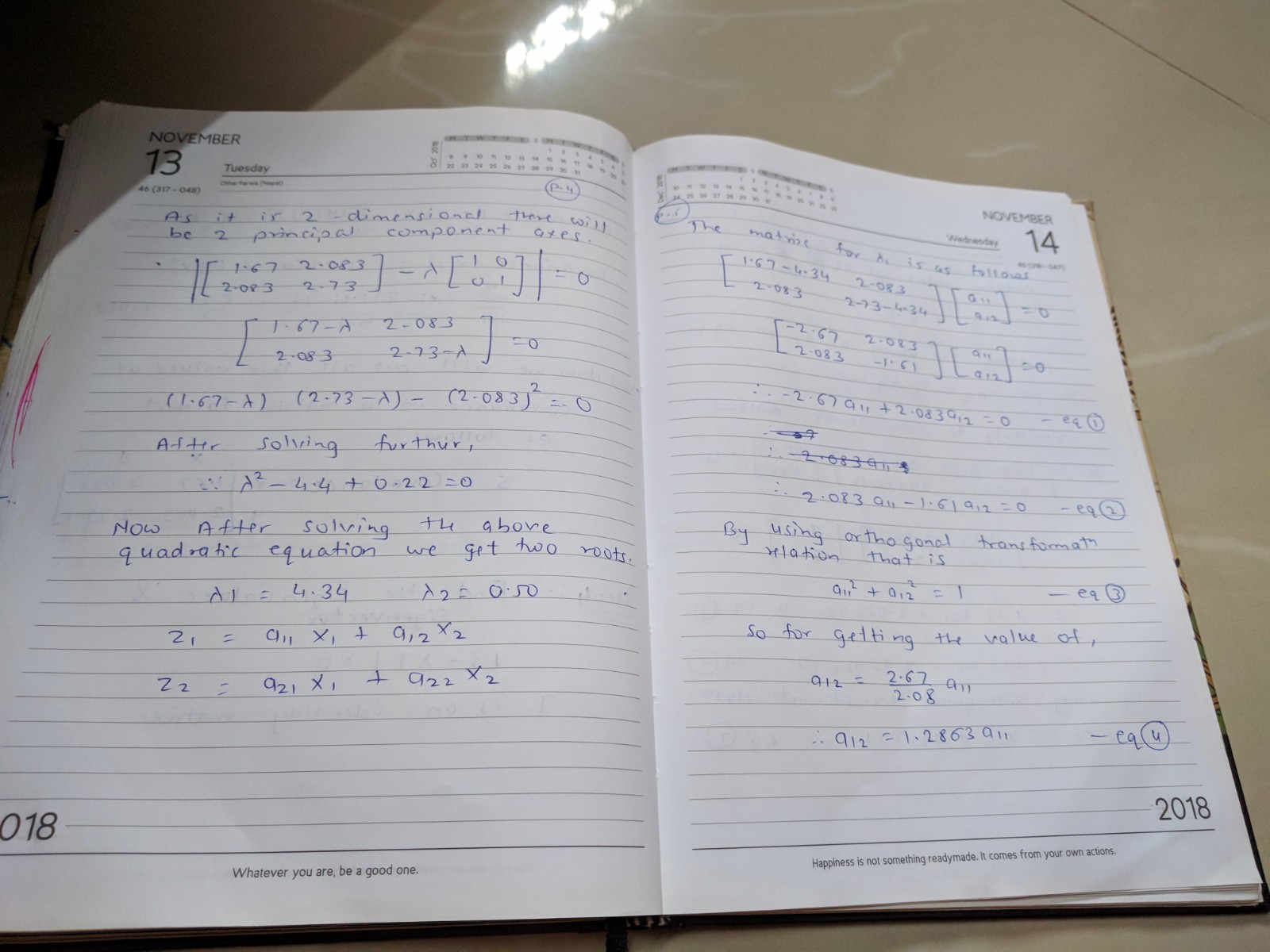
**7.2.3. Steps for Principal Component Analysis**

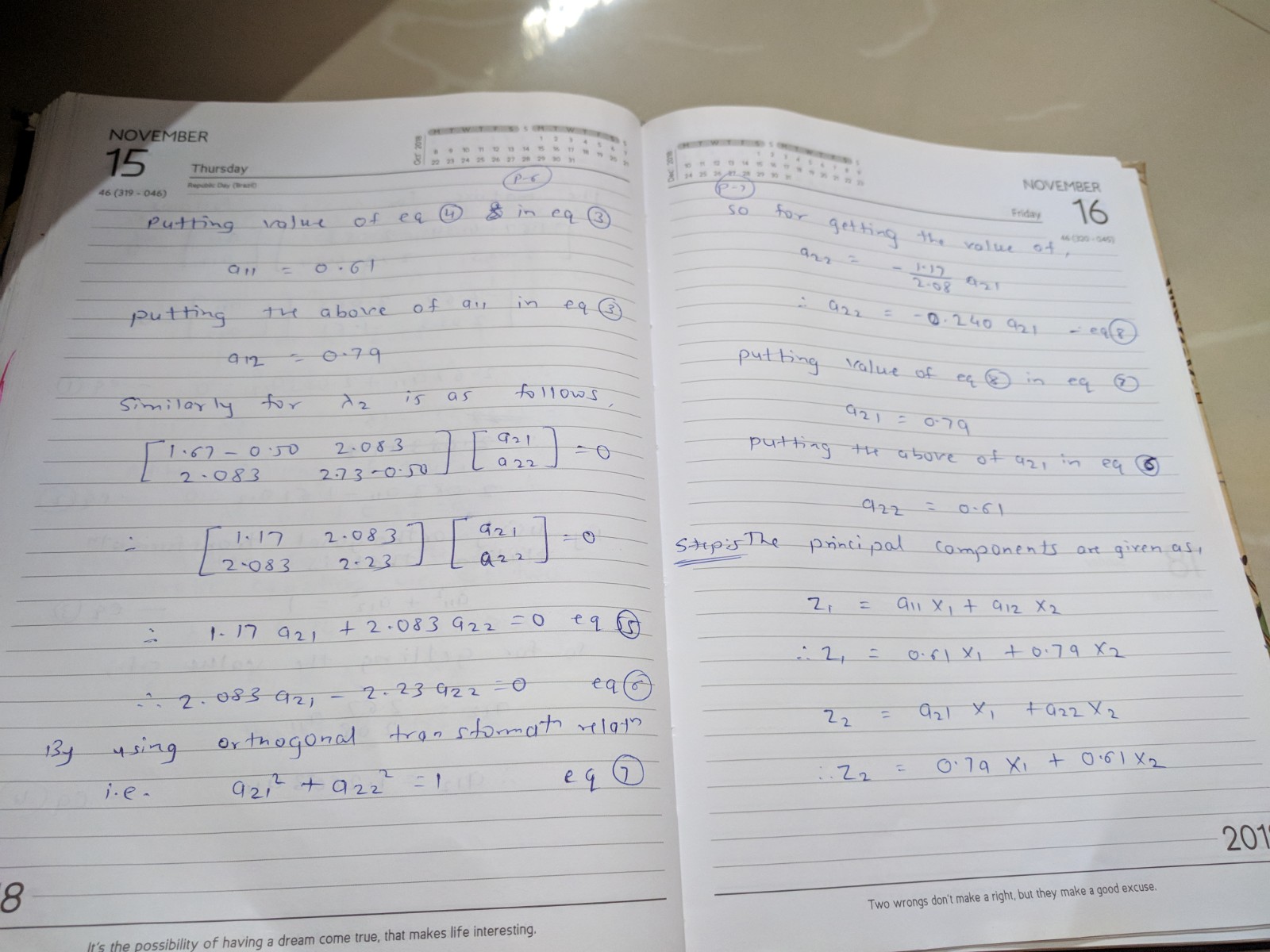
1. Find the Transpose of the given data set.
2. Find the covariance of the given data set.
3. Put the values of the covariance vector in covariance matrix.
4. Find Eigen values and Eigen vector.
5. Calculate the Principal Components.

**7.2.5. Example of PCA**









**7.3. K-Means**

K-means clustering is a type of unsupervised learning, which is used when you have unlabelled data (i.e., data without defined categories or groups).

K-means clustering is an exploratory data analysis technique.

It is non-hierarchical method of grouping objects together.

K-means clustering is an iterative clustering algorithm in which items are moved among set of clusters until the desired set is reached.

A high degree of similarity among elements in clusters is obtained, while a high degree of dissimilarity among elements in different clusters is achieved simultaneously.

The cluster mean of Ki = {ti1,t i2,…,t im}is defined as

mi = 1/m tij

mean is identical to the centroid.

**7.3.1. Steps for K-means clustering Algorithm**

1. Start
2. Input number of clusters.(This is usually given in problem else consider 3 cluster for better results).
3. Calculate Centroid or mean.
4. Calculate distance
5. Group the objects depending upon the minimum distance.
6. Go to Step 3 and continue till all the objects are clustered.

**7.3.2. Example**

Apply K-means algorithm on given data set {2,4,10,12,3,20,30,11,25} by dividing the set into cluster k=2.

**Solution :**

Given data set is {2,4,10,12,3,20,30,11,25}

Divide the given dataset into K number of clusters which is given 2.

We initially assign the means to first two values m1 = 2 and m2 = 4.

|  |  |  |  |
| --- | --- | --- | --- |
| m1 | m2 | K1 | K2 |
| 2 | 4 | {2,3} | {4,10,11,12,20,25,30} |
| Value 3 is equally close to both means, so we choose K1, recalculate means to get,  m1 is the means of first cluster(K1) and m2 is the mean of second cluster(K2).  m1 = (2+3)/2 = 2.5m2 = (4+10+11+12+20+25+30)/7 = 16 | | | |
| 2.5 | 16 | {2,3,4} | {10,11,12,20, 25,30} |
| m1 = (2+3+4)/3 = 3m2 = (10+11+12+20+25+30)/6 = 18 | | | |
| 3 | 18 | {2,3,4,10} | {11,12,20, 25,30} |
| 4.75 | 19.6 | {2,3,4,10,11,12} | {20, 25,30} |
| 7 | 25 | {2,3,4,10,11,12} | {20, 25,30} |

Note that clusters in the last two steps are identical.

This will yield identical mean, thus the mean have converged

Finally, we have 2 clusters :

K1 = {2,3,4,10,11,12}

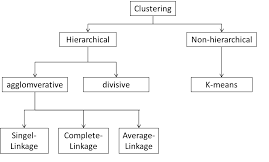
K2 = {20, 25,30}

**7.4. Hierarchical clustering**

Hierarchical clustering is another unsupervised learning algorithm that is used to group together the unlabeled data points having similar characteristics. Hierarchical clustering algorithms falls into following two categories.

**Agglomerative hierarchical algorithms** – Agglomerative algorithms start with each individual item in its own cluster and iteratively merge clusters  (bottom-up approach) until all items belong in one cluster. The hierarchy of the clusters is represented as a dendrogram or tree structure.

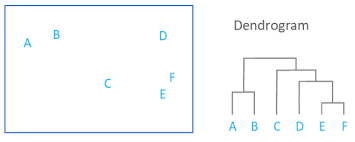
**Divisive hierarchical algorithms** − On the other hand, in divisive hierarchical algorithms, all the data points are treated as one big cluster and the process of clustering involves dividing (Top-down approach) the one big cluster into various small clusters.



**Dendrogram**

Dendrogram is a tree like structure which is used to represent the hierarchical clustering technique. In this structure the individual elements are known as Leaf, these leafs are combined together to form the cluster and this cluster is known as root.

In given figure, A and B are individual leaf item similarly C,D,E,F. When these leaf join together they form the cluster. For example, A and B combine together to form the cluster similarly the other pair EF join together to form the cluster and then is merged with D to form new cluster DEF and then DEF is again merged with C to form new cluster CDEF. Finally cluster AB is merged with CDEF and form final cluster ABCDEF.



**7.4.1. Steps to Perform Agglomerative Hierarchical Clustering**

We are going to explain the most used and important Hierarchical clustering i.e. agglomerative. The steps to perform the same is as follows −

* **Step 1** − Treat each data point as single cluster. Hence, we will be having, say K clusters at start. The number of data points will also be K at start.
* **Step 2** − Now, in this step we need to form a big cluster by joining two closet datapoints. This will result in total of K-1 clusters.
* **Step 3** − Now, to form more clusters we need to join two closet clusters. This will result in total of K-2 clusters.
* **Step 4** − Now, to form one big cluster repeat the above three steps until K would become 0 i.e. no more data points left to join.
* **Step 5** − At last, after making one single big cluster, dendrograms will be used to divide into multiple clusters depending upon the problem.

Before any clustering is performed, it is required to determine the proximity matrix containing the distance between each point using a distance function. Then, the matrix is updated to display the distance between each cluster. The following three methods differ in how the distance between each cluster is measured.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **1. Single Linkage** | | | |  |  |
| In single linkage hierarchical clustering, the distance between two clusters is defined as the *shortest*  distance between two points in each cluster. For example, the distance between clusters “r” and “s”  to the left is equal to the length of the arrow between their two closest points. | | | |  |  |
| https://www.saedsayad.com/images/Clustering_single.png   |  |  |  | | --- | --- | --- | | **2. Complete Linkage** |  |  | | In complete linkage hierarchical clustering, the distance between two clusters is defined as the *longest*  distance between two points in each cluster. For example, the distance between clusters “r” and “s”  to the left is equal to the length of the arrow between their two furthest points. |  |  | | https://www.saedsayad.com/images/Clustering_complete.png |  |  | | | | |  |  |
| **3. Average Linkage** |  |  |
| In average linkage hierarchical clustering, the distance between two clusters is defined as the average  distance between each point in one cluster to every point in the other cluster. For example, the distance  between clusters “r” and “s” to the left is equal to the average length each arrow between connecting  the points of one cluster to the other. |  |  |
| https://www.saedsayad.com/images/Clustering_average.png |  |  |

**7.4.2. Example of Hierachical Agglomerative**

Apply agglomerative clustering algorithm on the given data and draw dendrogram using single link and complete link

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 1 | 2 | 3 | 4 | 5 |
| 1 | 0 | 9 | 3 | 6 | 11 |
| 2 | 9 | 0 | 7 | 5 | 10 |
| 3 | 3 | 7 | 0 | 9 | 2 |
| 4 | 6 | 5 | 9 | 0 | 8 |
| 5 | 11 | 10 | 2 | 8 | 0 |

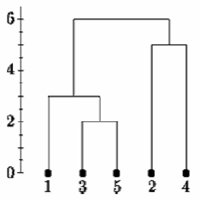
**Dendrogram for given data using single link**

Now lets start clustering.  The smallest distance is between three and five and they get linked up or merged first into a the cluster '35'.

To obtain the new distance matrix, we need to remove the 3 and 5 entries, and replace it by an entry "35" .  Since we are using single linkage clustering, the distance between "35" and every other item is the minimum of the distance between this item and 3 and this item and 5.  For example, d(1,3)= 3 and d(1,5)=11.  So, D(1,"35")=3.  This gives us the new distance matrix.  The items with the smallest distance get clustered next.  This will be 2 and 4.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 35 | 1 | 2 | 4 |
| 35 | 0 | 3 | 7 | 8 |
| 1 | 3 | 0 | 9 | 6 |
| 2 | 7 | 9 | 0 | 5 |
| 4 | 8 | 6 | 5 | 0 |

Continuing in this way, after 6 steps, everything is clustered. This is summarized below.  On this plot, the y-axis shows the distance between the objects at the time they were clustered.  This is called the cluster height.  Different visualizations use different measures of cluster height.

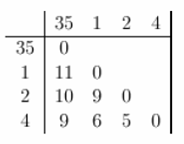


Single Linkage

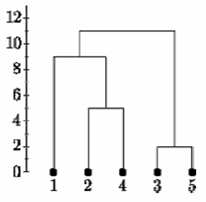
**Dendrogram for given data using complete link**

Now lets start clustering.  The smallest distance is between three and five and they get linked up or merged first into a the cluster '35'.

 To obtain the new distance matrix, we need to remove the 3 and 5 entries, and replace it by an entry "35" .  Since we are using complete linkage clustering, the distance between "35" and every other item is the maximum of the distance between this item and 3 and this item and 5.  For example, d(1,3)= 3 and d(1,5)=11.  So, D(1,"35")=11.  This gives us the new distance matrix.  The items with the smallest distance get clustered next.  This will be 2 and 4.



Continuing in this way, after 6 steps, everything is clustered. This is summarized below.  On this plot, the y-axis shows the distance between the objects at the time they were clustered.  This is called the cluster height.  Different visualizations use different measures of cluster height.

  
Complete Linkage

**7.5. Ensemble Learning**

Ensemble modelling is one of the powerful techniques used to improve the performance of any machine learning model. In Ensemble Modelling a combined solution or decision is formed out of various available approaches.

**7.5.1. Combining Multiple Models**

Ensemble Modelling is defined as an art of combining various approaches for performing certain task in order to create an improved, stable and efficient predictive model or system.

Ensemble methods are Meta – algorithms that combines several machine learning techniques into one predictive model in order to decrease variance (bagging), bias(boosting), or improve performance predictions (stacking).

Depending on the basis of the base learners involved in creating the ensemble model there are two types of ensemble methods.

1. Sequential Ensemble Method
2. Parallel Ensemble Method
3. **Sequential Ensemble Method (SEM)**

In SEM the base learners are generated in the sequential manner. The idea behind this method is to exploit the dependence between the base learners. Here the overall performance can be boosted by weighing previously mislabelled examples with higher weight. For e.g. AdaBoost.

1. **Parallel Ensemble Method (PEM)**

In PEM the base learner are generated in the parallel manner. The idea behind this method is to exploit the independence between the base learners. Here the idea is to reduce the error by performing averaging. For e.g. Random Forest.