



**MANIPAL INSTITUTE OF TECHNOLOGY**  
**MANIPAL**  
*(A constituent unit of MAHE, Manipal)*

**DEPARTMENT OF MECHATRONICS**

# **Bayesian Methods for Machine Learning**

Mini Project Report

Date: 29/05/2021 to 03/07/2021

*Submitted by*

**NAME: SARVESH P BHANDARY**

**REG NO: 180929094**

**SECTION A**

*Under the guidance of*

**MR. MAHESH INAMDAR**

**DEPARTMENT OF MECHATRONICS**

**June 2021**



**MANIPAL INSTITUTE OF TECHNOLOGY**

**MANIPAL**

*(A constituent unit of MAHE, Manipal)*

**DEPARTMENT OF MECHATRONICS ENGINEERING  
CERTIFICATE**

This is to certify that the mini-project titled “PROJECT TITLE” is carried out by “Sarvesh Bhandary (Reg. No. 180929094)” during May-July 2021. The report is submitted to the Department of Mechatronics and is treated as an alternative to the industrial training. Industrial training is an academic requirement for the award of B. Tech degree in Mechatronics.

Guide: Mr. Mahesh Inamdar

Head of the Department:  
Dr. Chandrashekhar Bhat

Date: 03/07/2021  
Place: Mumbai

<b>Contents</b>	
1) Introduction	4
2) Objectives and Weekly Breakup	5
3) Building the Foundation	7
4) Sampling from a Normal Distribution	10
5) Gradient Descent Algorithm	12
6) Expectation Maximization Algorithm	15
7) Logistic Regression using MSE	18
8) Conclusion	21
9) Annexure	22

## 1) Introduction

Humans are intelligent beings and can think and make their own decisions. But in many situations, the capabilities of humans are very limited. We cannot process large data in a limited amount of time, which is why we use machines to do that job for us. However, this does not mean that we can just sit back, and the machines will do all the work for us. We need to program the machines in the right and the most efficient way possible so that we get desired results as quickly as possible. This is known as machine learning. We *train* our machines with data of the past and the machine can make predictions for future data or make decisions with the help of the previous knowledge given.

Machine learning can be broadly classified into 3 categories:

### 1.1) Unsupervised learning:

This is a type of machine learning in which our data does not have *labels* and the algorithm would try to find meaning or patterns in the data and try clustering them.

### 1.2) Supervised learning:

This is a type of machine learning in which our data is *labeled*, for example, an email could be labelled as spam or non-spam. The algorithm tries to make a model which would correctly predict whether or not a new email, in this case, is spam or non-spam with the help of the data of previous emails.

### 1.3) Reinforcement learning:

Reinforcement learning is a type of machine learning in which the algorithm tries to maximize its reward function in a particular situation to find the best possible, say, path it should take while travelling on a road or any environment.

There are many efficient algorithms today used for different applications and research is going on to find better ones.

I am focusing on the Bayesian methods of machine learning which is a probabilistic form of learning and helps in handling missing data, finding uncertainty in our predictions which is particularly important in fields like medicine, and many more.

## **2) Objectives and Weekly Breakup**

### **2.1) Objectives:**

This training helped me achieve the following objectives:

- a) Understanding different concepts and fundamentals of machine learning.
- b) Getting acquainted with the different terminologies of statistics.
- c) Gaining and applying mathematical knowledge into a field that has vast applications of it.
- d) Understanding the working and use of different machine learning algorithms.
- e) Implementing various algorithms on different datasets from scratch.
- f) Learning how to represent data by plotting graphs, histograms etc. using matplotlib library in python.
- g) Using numpy arrays in python to make algorithms work fast and efficiently.
- h) To get better understanding on uses of Bayesian Methods in Machine Learning.
- i) Perform a comparative study on Bayesian and using Loss function.

### **2.2) Weekly breakup:**

My weekly breakup of activities for this mini project can be summarized in the following manner:

#### **Week 1**

- a) Parametric and Non-Parametric Methods
- b) Sampling data from Continuous and Discrete Distributions, MCMC
- c) Bayes Rule
- d) Project Report

#### **Week 2**

- a) Gradient Descent
- b) Building a Linear Regression Model using GDA
- c) Expectation Maximization algorithm
- d) Bayesian Linear Regression Model
- e) Project Report

#### **Week 3**

- a) Probabilistic Clustering
- b) Comparison of Lloyd's and Probabilistic Clustering
- c) Project Report

#### **Week 4**

- a) Paper Reading and Implementation
- b) Project Report

### 3) Building the Foundations

Before we jump into the implementations of machine learning algorithms, we must have a good grasp of the fundamentals on which these algorithms are built. These include:

#### 3.1) **Probability:**

Probability of an event is the likeliness of occurrence of the event.

#### 3.2) **Random variable:**

A random variable is an event described in the form of a number. (For example, if  $X$  denotes the number of times a head occurs when a coin is tossed 10 times, then  $X$  is the random variable and  $P(X = 5)$  is the probability of the random variable taking the value 5, or the probability of 5 heads occurring)

#### 3.3) **Probability density function (PDF):**

For a continuous random variable, the probability distribution is expressed in the form of a density function in which the probability of a random variable to lie in a certain range is the area under the curve in that range.

#### 3.4) **Cumulative distribution function (CDF):**

The CDF is a function that gives the probability of the random variable taking a value less than or equal to a variable (which is the argument of the function)

#### 3.5) **Moments of statistics:**

There are 4 moments of statistics, which are as follows:

3.5.1) Mean: Mean is the average value of all outputs of the data.

3.5.2) Variance: Variance describes how scattered the data is. More variance means the data is very scattered.

3.5.3) Skewness: Skewness is the measure of asymmetry of a distribution.

3.5.4) Kurtosis: Kurtosis explains how flat or pointed the distribution is. High kurtosis means that the curve is pointed, and the data has many outliers, whereas low kurtosis means that the curve is flat and there are very few outliers.

#### 3.6) **Probability distributions:**

A probability distribution is a function whose value  $y$ , is the probability of a random variable taking a certain value  $x$ . So, the function  $y = f(x)$  is the probability distribution of the random variable  $X$  (here,  $x$  is a value taken by the random variable  $X$ ).

There are many known distributions, to name a few, we have:

3.6.1) Normal distribution: A gaussian or normal distribution is also known as a bell curve, which takes its highest value at the mean and the value decreases as we

move away from it. The variance of the distribution tells us how close all values at different points are to the mean value of the function.

3.6.2) Binomial distribution: This is a discrete probability distribution of random variable taking one of two possible values a certain number of times.

3.6.3) Poisson's distribution: This is also a discrete probability distribution which tells the probability of a random variable taking a value in a fixed range of time or space (for example, the probability of, say, *4 accidents taking place* in some city *in a week*).

### 3.7) Central Limit Theorem:

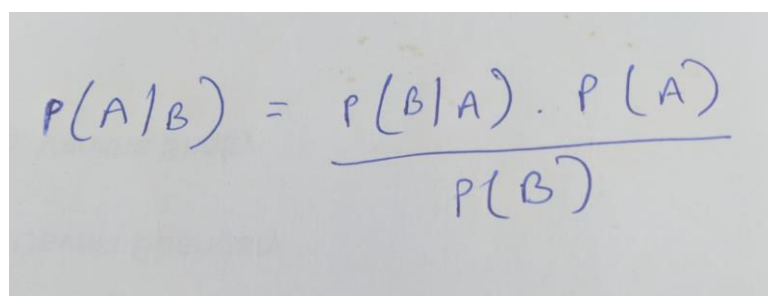
The central limit theorem states that if we collect random samples from a population and plot the distribution of its mean (mean of a collection of samples and plotting the mean of every collection), then regardless of what the original distribution was, the newly obtained distribution will always be normally distributed.

### 3.8) Markov chains:

A Markov chain is a linkage of different states or different events in a system in which every state has some probability of transitioning to every other state (which may be 0 for some or all states). We store these probabilities in a matrix known as a transition matrix. Markov chains help predict the behavior of a system in the long run or at any point of time with the help of the transition matrix. (For example, it could help in predicting how the weather could be in a region given the weather on previous days)

### 3.9) Bayes' Theorem:

Bayes' Theorem deals with conditional probability and helps us find the probability of the occurrence of an event A given that another event B has already occurred. It is as follows:


$$P(A|B) = \frac{P(B|A) \cdot P(A)}{P(B)}$$

Eq. 3.1

Here,

$P(A|B)$  is the probability of event A occurring given that event B has occurred.

$P(B|A)$  is the probability of event B occurring given that event A has occurred.

$P(A)$  is the probability of event A occurring.

$P(B)$  is the probability of event B occurring.

### **3.10) Bayesian Linear Regression:**

Linear regression is an algorithm to find the best fit line, plane, or hyperplane to model a given data with respect to different parameters. Ordinary Linear Regression gives us one 'best' value for each parameter whereas Bayesian Linear Regression gives us a probability distribution of all possible values of every parameter where the expected value of each parameter is typically the same as that obtained by Ordinary Linear Regression.

### **3.11) K-means and probabilistic clustering:**

K-means clustering is an algorithm in an unsupervised machine learning algorithm in which data is grouped in certain clusters for understanding the data and performing different operations based on it. While K-means and probabilistic clustering wish to achieve similar goals, their approaches are different. The main difference lies in the type of clustering. These are of 2 types:

- 3.11.1) Hard clustering: This approach restricts every data point to be a part of only 1 cluster, so no data point could be a part of multiple or no clusters. K-means follows this type of clustering.
- 3.11.2) Soft clustering: This approach makes every data point a part of all clusters but with a weight for each which describes how likely it is to be a part of a particular cluster (every data point has a different weight corresponding to every cluster). Probabilistic clustering follows this type of clustering.



#### 4) Sampling from a normal distribution

A normal distribution is a distribution that shows up in most real-life situations which is why it is given the name normal distribution. A normal distribution with mean 0 and variance 1 looks like this:

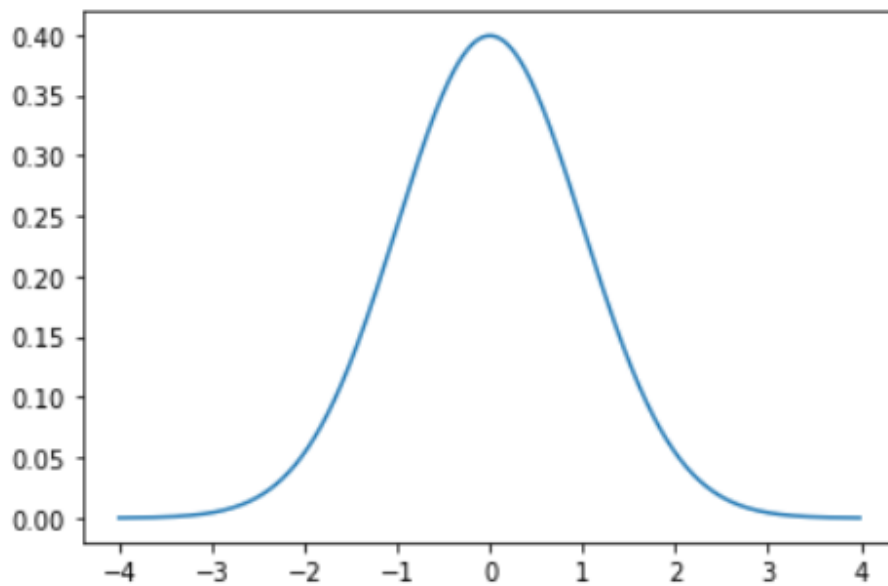


Fig. 4.1

In almost all situations, analyzing the entire population to come up with conclusions is impossible, hence, we try to take random samples out from the population and try to understand the population by analyzing these samples. Let us look at an example.

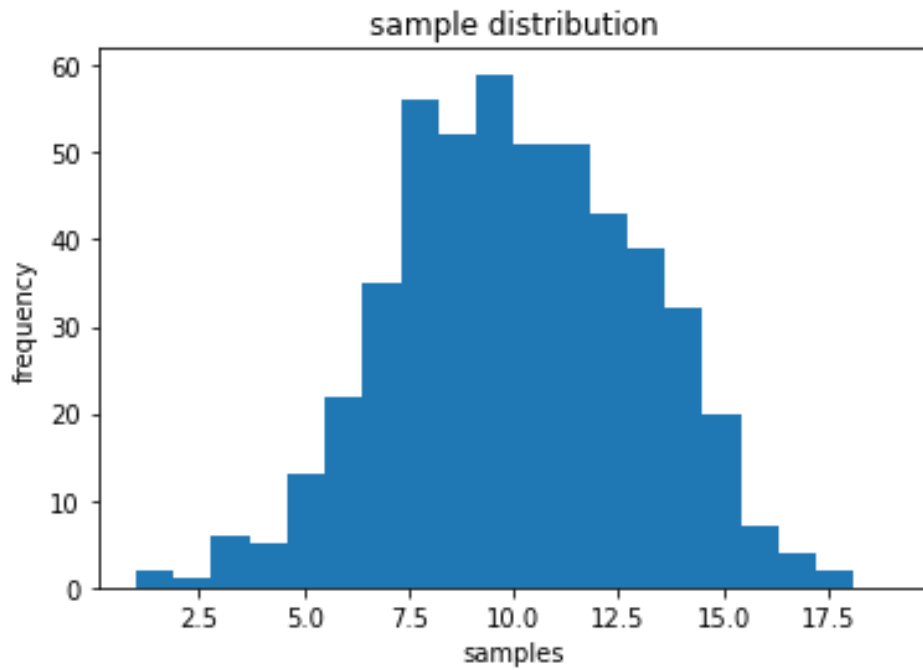


Fig. 4.2

The figure above is a collection of 500 samples from a normal distribution of mean 10 and standard deviation 3, and this graph represents the frequency of occurrence of the samples lying in different ranges. Even though this collection consists of only a small fraction of all the points belonging to the family of points of the original normal distribution, this gives us a pretty decent idea of what the actual population would look like. As the number of samples increase, the collection starts to resemble the true population even more.

## 5) Gradient Descent Algorithm

Gradient descent is an algorithm in which we analyze the graph of the error function (for example mean squared error) with respect to the different parameter values of our model, which tells us how inaccurate our model is, and from a point on the curve of the error function, move along the steepest slope down to reach the local minimum value of the function. The starting point may be randomly selected.

This iterative process can be broken down into 2 steps:

- a) Calculate the gradient.
- b) Update the parameters.

Now let us understand each of the steps in detail:

### 5.1) Calculate the gradient:

The error function is a function of the parameters as it represents the total error that we observe in all the data points taking our current parameter values as the parameters for our model (for example, in the line  $y = mx + c$ ,  $m$  and  $c$  are the parameters of the line that describe its slope and  $y$  intercept respectively). So, if we slightly change any of the parameters, the value of the error changes (may increase or decrease). The gradient (or the slope of the function) tells us about the direction in which our point should move on the curve or the amount by which each parameter should change with respect to each other in such a way that the reduction in error is the highest so that we reach the local minimum or the local optimum point with the least number of iterations.

### 5.2) Update the parameter values:

Now that we know the amount by which each parameter should change with respect to each other, we can decide a learning rate based on the accuracy that we want which would give the absolute values of the required change in each parameter and by adding those changes, we get our new parameter values which give a slightly lower error than the previous set of parameters.

We can keep iterating until the values converge and we reach a desired amount of accuracy. Note that the higher the desired accuracy, more the number of iterations it would take to reach there and hence, the algorithm could take very long to stop. Hence, we must choose the accuracy wisely.

So, I applied Gradient descent on the function given by  $y = x^2 + 2x$ , to find its minimum value. Here, we have a function itself that we have to minimize instead of minimizing an error term. So, I started with a random value of  $x$  where the output of the function was approximately 6 as we can see in the graph drawn below. This graph shows the change in value of the output  $y$  with increase in the number of iterations. As the minimum value of the function is -1, the function soon converged to the value -1. As we can see, the function had already converged by the time we reached 150 iterations and hence, carrying on till 400 iterations or represented below is just a waste of time unless we need that much accuracy.

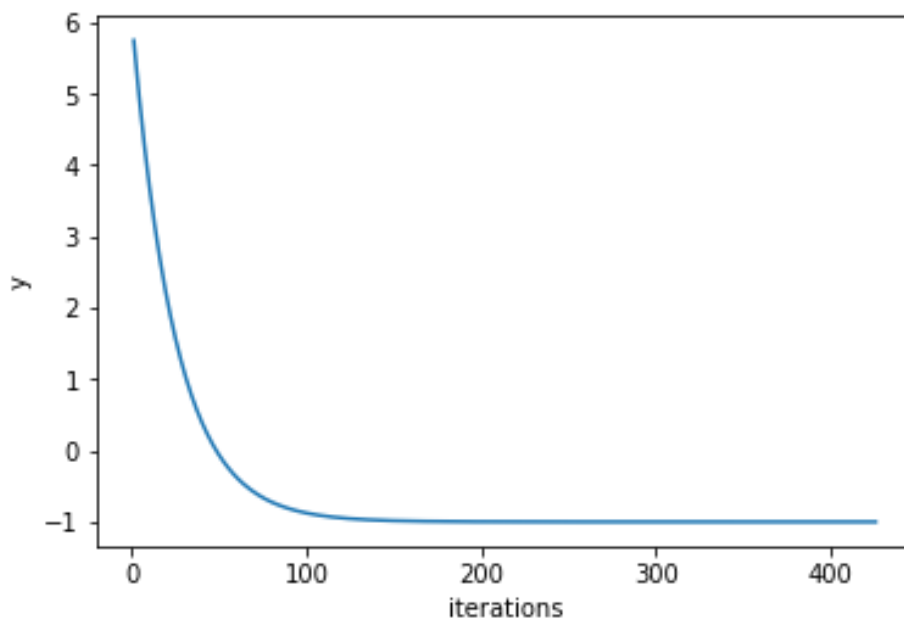


Fig. 5.1

Next, I had to apply Gradient descent to find the best fit line for a noisy data which had a true equation given by  $y = 2x + 5$ .

This problem requires us to find the values of parameters,  $a$  and  $b$ , for the equation  $y = ax + b$  such that the Mean Squared Error (MSE) is minimized. MSE can be represented as the sum of  $(y - y')^2$  for all data points, where  $y$  is the true output of the data for a particular  $x$ , and  $y'$  is the estimated output of  $x$  given by our model. The plot of the true data and our model can be seen in the figure below:

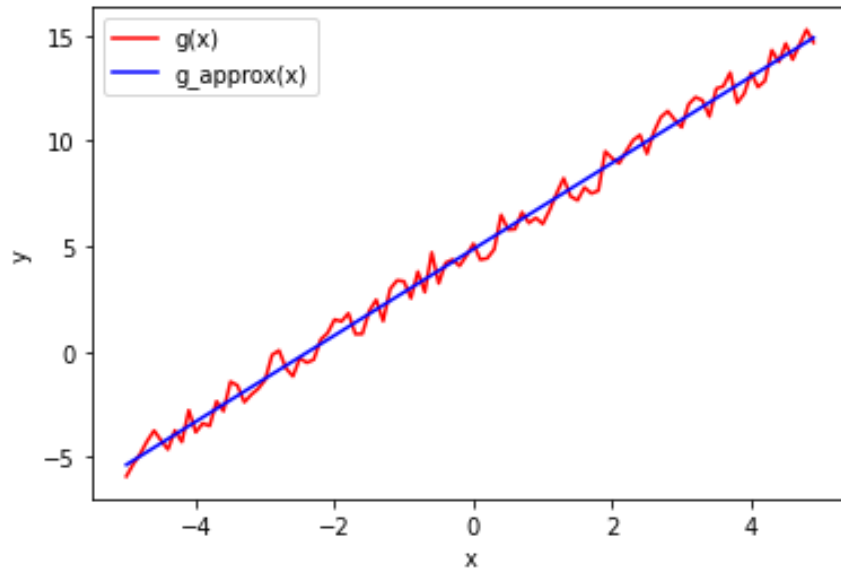


Fig. 5.2

Here,  $g\_approx(x)$  is the estimated function or the blue line in the above graph whereas  $g(x)$  is the true output function for every  $x$ . The mean squared error can be seen decreasing with every iteration in the graph below.

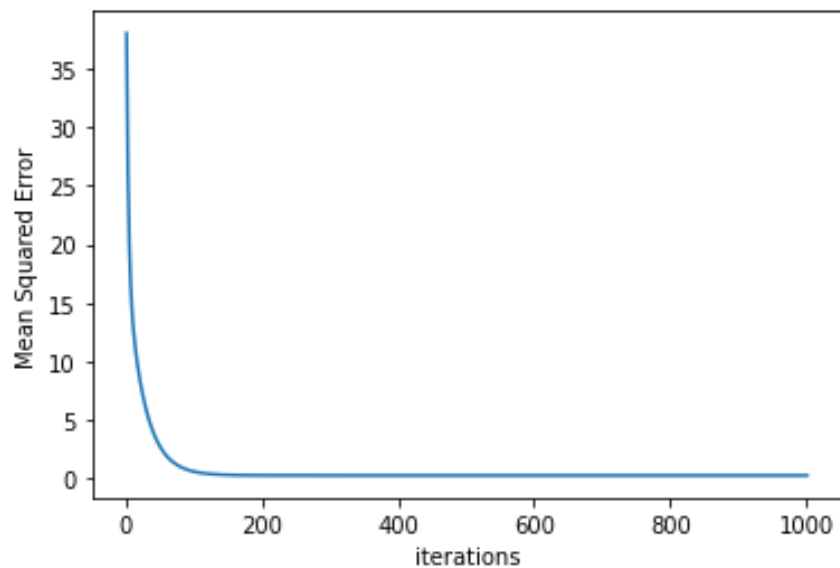


Fig. 5.3

The value of the MSE converged to almost 0 starting from a value as high as 40 when the parameters were randomly initiated.

## 6) Expectation Maximization Algorithm

Expectation maximization algorithm or the EM algorithm is an unsupervised learning algorithm in which we group the data into different gaussians probabilistically, which means that every data point has a certain probability of being a part of every gaussian depending on its relative distances with every gaussian and the parameters of each gaussian. These are also referred to as weights. More the weight of a point with respect to a gaussian, more is its likeliness to be a part of it and more influence it has on the positioning of the gaussian as well. For a given number of clusters or gaussians, the algorithm tries to position the gaussians in such a way that most data points belong to one cluster or the other. We begin the process by assuming a prior (a set of initial parameters of the gaussians which may be randomly assigned or by using prior knowledge of the data)

This iterative process consists of 2 steps:

- a) Expectation step
- b) Maximization step

Now let us understand each of the steps in detail:

### 6.1) Expectation step:

In the expectation step or the E-step, we determine the weights (or the expectation value) of each data point with respect to each gaussian and store it in a matrix known as the likelihood matrix. This is a  $(k \times n)$  matrix where  $k$  is the number of gaussians and  $n$  is the number of data points and the  $(i, j)$  element of the matrix is the weight of  $j$ th data point with respect to the  $i$ th gaussian which is found by dividing the pdf of the  $j$ th data point with respect to the  $i$ th gaussian by the sum of the pdfs of the  $j$ th data point with respect to all gaussians.

### 6.2) Maximization step:

In the maximization step or the M-step, we update the means and standard deviations of our gaussians depending on the weights such that the gaussians get closer to the points strongly related to them and hence cluster the points. First, the means are updated by making the new mean equal to the centre of mass or the weighted average of the data points (where the weights are the weights of the data points corresponding to the gaussian we are updating the mean of). Second, the standard deviations are updated by equating it to the square root of the weighted mean of the square of the distances between the data points and the updated means.

For the maximization step, the new mean and variance are as follows:

$$\mu = \frac{\sum_{i=1}^m y'_i}{m} = \frac{y'_1 + y'_2 + \dots + y'_m}{m}$$

Eq. 6.1

$$\sigma^2 = \frac{\sum_{i=1}^m (y'_i - y_i)^2}{m} = \frac{(y'_1 - y_1)^2 + (y'_2 - y_2)^2 + \dots + (y'_m - y_m)^2}{m}$$

Eq. 6.2

This iteration is made to continue until the means and standard deviations converge or become stable.

So, I applied the EM algorithm on a dataset that had a distribution represented by the image below. Many real-life distributions such as heights of students in a university etc follow normal distribution which is why we assume the data to normally distributed as well. I started by initiating equally spaced gaussians, the parameters of which, with successive iterations, converged to the true value of that of the data.

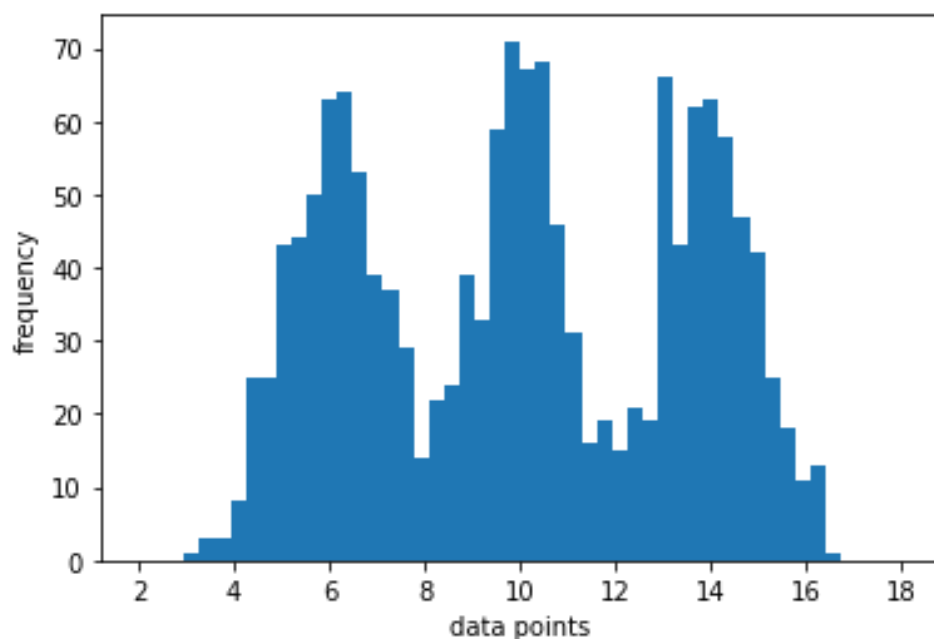


Fig. 6.1

The initial and final values of the parameters are as shown below:

```
The true (mean,variance) of the sets of data points are:
```

```
[[ 6  1]
 [10  1]
 [14  1]]
```

```
The (mean,variance) of the clusters are:
```

```
[[ 6.05975467  1.01506006]
 [ 9.97473906  0.97084038]
 [13.99860167  1.00268597]]
```

Fig. 6.2



## 7) Logistic Regression

In machine learning, logistic regression is an algorithm which is used to create groups in data and categorize them, for example, marking an email as spam or non-spam, predict whether a particular day would be sunny or cloudy or even create multiple groups using previous data. Many a times, we encounter problems when the answer is either a yes or a no and this is solved by logistic regression by giving an output of 1 if the answer is yes and giving the output as 0 if the answer is a no.

So, we use a function known as sigmoid function as an output function which outputs values very close to 0 and 1.

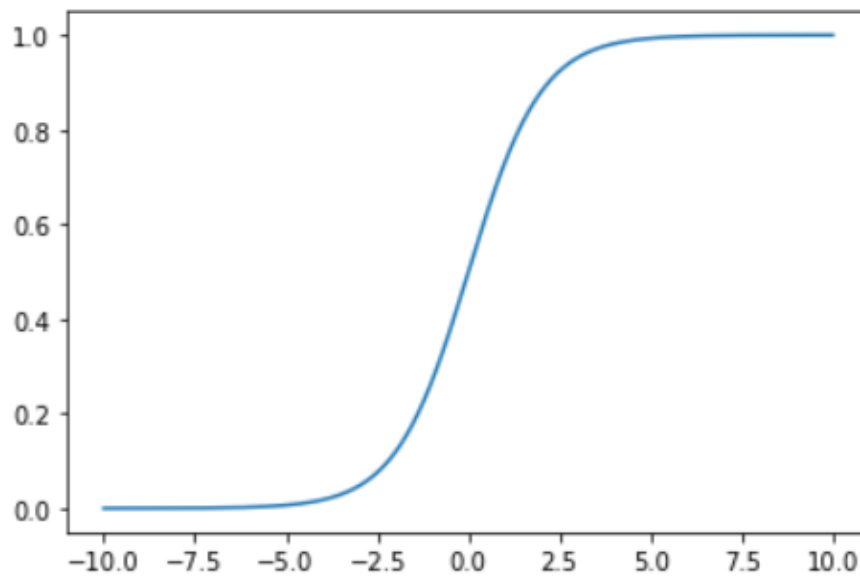


Fig. 7.1

The above figure represents a typical sigmoid function. Its equation is as follows:

$$\text{sig}(x) = \frac{1}{1 + e^{-(ax+b)}}$$

Eq. 7.1

Here,  $a$  and  $b$  represent the amount by which the graph is compressed or stretched along the  $x$  axis and the position of the centre respectively. By adjusting the parameters,  $a$  and  $b$ , we can fit many curves which would give an output of approximately 0 or 1 for any input data.

I used gradient descent to find the values, a and b, in order to minimize the error term which can be represented as follows:

$$J = \left( \frac{-1}{m} \right) \cdot \sum_{i=1}^m \left( y_i \cdot \log(\text{sig}(x)) + (1 - y_i) \cdot \log(1 - \text{sig}(x)) \right)$$

Eq. 7.2

Where:

J is the error term

m is the total number of samples

$y_i$  is the true output of a given  $x_i$

$\text{sig}(x)$  is the output of our model for our current values of a and b.

To calculate the gradient, we use chain rule to find the partial derivatives with respect to a and b. They are as follows:

$$\frac{\partial J}{\partial a} = \frac{\partial J}{\partial \text{sig}(x)} \times \frac{\partial \text{sig}(x)}{\partial a}$$

Eq. 7.3

$$\frac{\partial J}{\partial b} = \frac{\partial J}{\partial \text{sig}(x)} \times \frac{\partial \text{sig}(x)}{\partial b}$$

Eq. 7.4

The parameters a and b are updated in the following manner:

$$\begin{aligned} a &= a - \alpha \cdot (\partial J / \partial a) \\ b &= b - \alpha \cdot (\partial J / \partial b) \end{aligned}$$

Eq. 7.5

Here,  $\alpha$  is the learning rate which tells how fast the algorithm learns.

The data points along with our predicted model can be seen in the figure below:

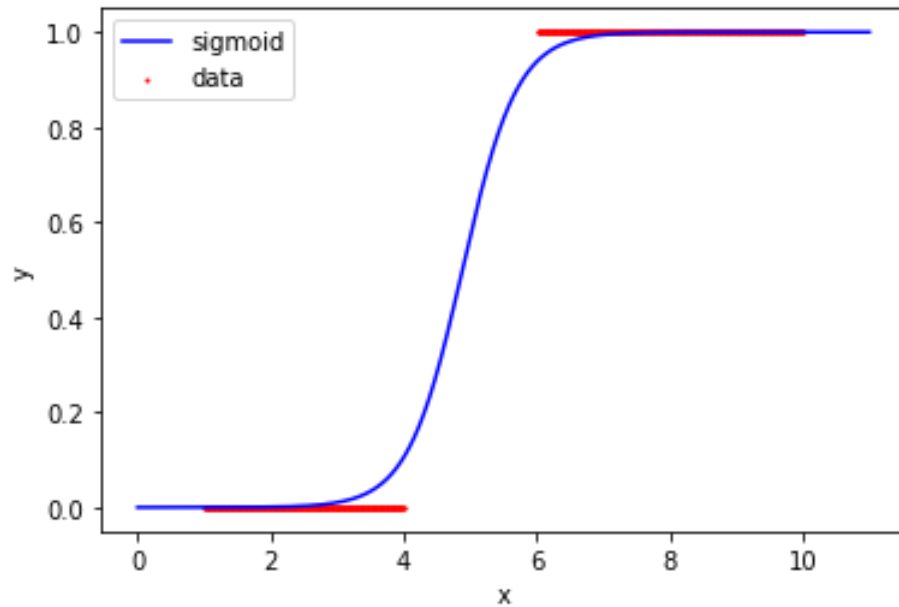


Fig. 7.2

Here, the red points represent the true output of the data and the blue curve is our predicted model for the same. We can see in the graph below that the error function converges to a small value starting from an extremely large amount produced for randomly initiated parameters with successive iterations.

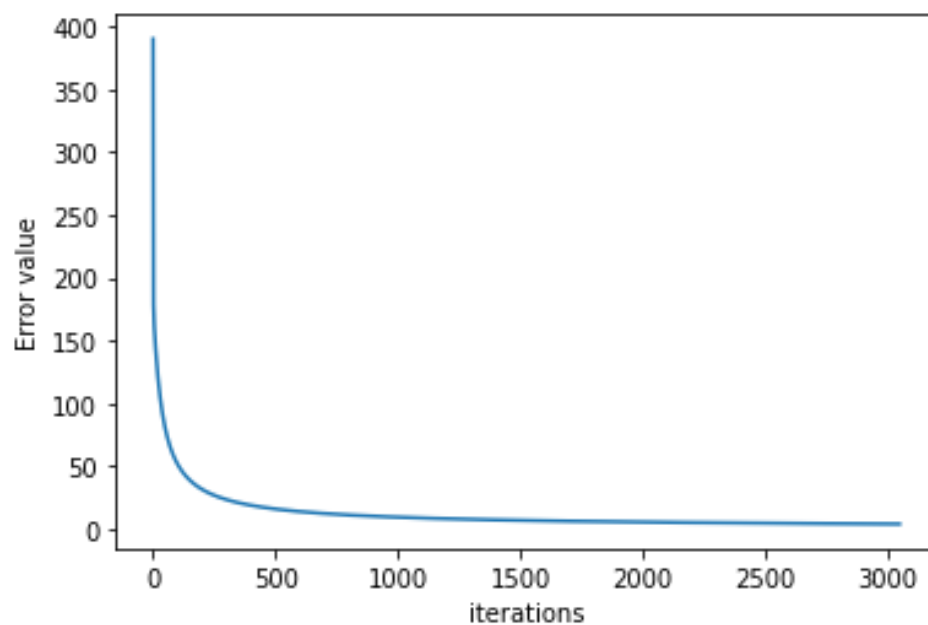


Fig. 7.

## **8) Conclusion**

Bayesian and non-Bayesian methods are both widely used in the industry. Bayesian methods come useful when we need to measure the uncertainty of our predictions and the parameters at all times. This is really important in fields such as medicine where knowing how confident we are in our parameters and result is necessary. Also, the use of a prior distribution, which can be made using domain knowledge, significantly affects the performance of the algorithms.

I learnt many different concepts of machine learning and have an intuitive idea of the same now. I also implemented the algorithms on python and the results and graphs are shown above along with the implementations. I thank my faculty for guiding me throughout this journey of learning and giving me valuable feedback.

## 9) Annexures

### 8.1) Annexure 1:

PO	• Tick	Page. No	Section No	Guides Observation
PO1	✓	12	5	
PO2	✓	10	4	
PO3	✓	18	7	
PO4	✓	15	6	
PO5	✓	15	6	
PO6	✓	7	3	
PO7				
PO8				
PO9				
PO10	✓	7	3	
PO11				
PO12				

Table 8.1: PO Mapping

PSO	• Tick	Pg. No		Section No	Guides Observation
PSO1					
PSO2	✓	12		5	

Table 8.2: PSO Mapping

8.2) Annexure 2:

Sl	PLO	• Tick	Pg. No	Section No	Guides Observation
1	C1.	✓	12	5	
2	C2.	✓	15	6	
3	C3.	✓	18	7	
4	C4.	✓	18	7	
5	C5.				
6	C6.	✓	10	4	
7	C7.				
8	C8.				
9	C9.				
10	C10.				
11	C11.				
12	C12.				
13	C13.	✓	15	6	
14	C14.				
15	C15.				
16	C16.				
17	C17.				
18	C18.				

Table 8.2: PLO Mapping