**WEEK 1:**

## 1. Write a program to implement the naïve Bayesian classifier for a sample training data set stored as a .CSV file. Compute the accuracy of the classifier, considering few test data sets.

## Objective: To implement the naïve Bayesian classifier for a sample training data.

## Outcome: Student will able to implement the Bayesian classifier.

## Algorithm:

## 

(Ignoring P(D) since it is a constant)

**Gaussian Naive Bayes:**

A Gaussian Naive Bayes algorithm is a special type of Naïve Bayes algorithm. It’s specifically

used when the features have continuous values. It’s also assumed that all the features are

following a Gaussian distribution i.e., normal distribution.

**Representation for Gaussian Naive Bayes:**

We calculate the probabilities for input values for each class using a frequency. With realvalued inputs, we can calculate the mean and standard deviation of input values (x) for each class to summarize the distribution.

This means that in addition to the probabilities for each class, we must also store the mean and

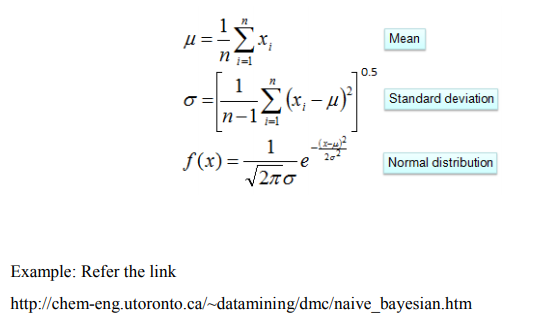
standard deviations for each input variable for each class.

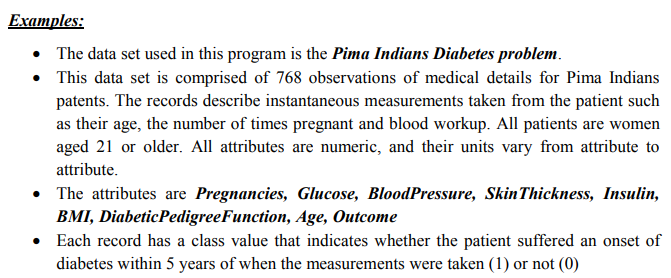
**Gaussian Naive Bayes Model from Data:**

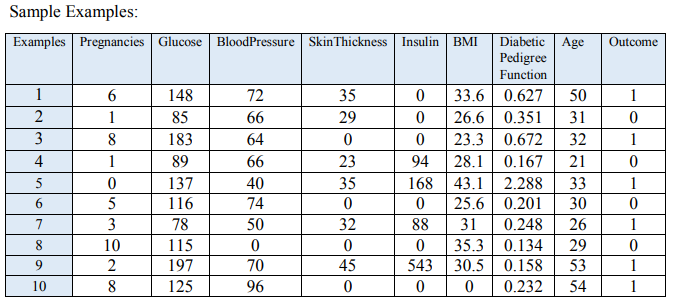
The probability density function for the normal distribution is defined by two parameters (mean

and standard deviation) and calculating the mean and standard deviation values of each input

variable (x) for each class value.







**Program:**

# 1.Naive Bayes (NB)

import pandas as pd

from sklearn.model\_selection import train\_test\_split

from sklearn.naive\_bayes import GaussianNB

from sklearn import metrics

df = pd.read\_csv("pima\_indian.csv")

feature\_col\_names = ['num\_preg', 'glucose\_conc', 'diastolic\_bp', 'thickness', 'insulin', 'bmi', 'diab\_pred', 'age']

predicted\_class\_names = ['diabetes']

X = df[feature\_col\_names].values # these are factors for the prediction

y = df[predicted\_class\_names].values # this is what we want to predict

#splitting the dataset into train and test data

xtrain,xtest,ytrain,ytest=train\_test\_split(X,y,test\_size=0.33)

print ('\n the total number of Training Data :',ytrain.shape)

print ('\n the total number of Test Data :',ytest.shape)

# Training Naive Bayes (NB) classifier on training data.

clf = GaussianNB().fit(xtrain,ytrain.ravel())

predicted = clf.predict(xtest)

predictTestData= clf.predict([[6,148,72,35,0,33.6,0.627,50]])

#printing Confusion matrix, accuracy, Precision and Recall

print('\n Confusion matrix')

print(metrics.confusion\_matrix(ytest,predicted))

print('\n Accuracy of the classifier is',metrics.accuracy\_score(ytest,predicted))

print('\n The value of Precision', metrics.precision\_score(ytest,predicted))

print('\n The value of Recall', metrics.recall\_score(ytest,predicted))

print("Predicted Value for individual Test Data:", predictTestData)

**Input: pima\_indian.csv**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| num\_preg | glucose\_conc | diastolic\_bp | thickness | insulin | bmi | diab\_pred | age | diabetes |
| 6 | 148 | 72 | 35 | 0 | 33.6 | 0.627 | 50 | 1 |
| 1 | 85 | 66 | 29 | 0 | 26.6 | 0.351 | 31 | 0 |
| 8 | 183 | 64 | 0 | 0 | 23.3 | 0.672 | 32 | 1 |
| 1 | 89 | 66 | 23 | 94 | 28.1 | 0.167 | 21 | 0 |
| 0 | 137 | 40 | 35 | 168 | 43.1 | 2.288 | 33 | 1 |
| 5 | 116 | 74 | 0 | 0 | 25.6 | 0.201 | 30 | 0 |
| 3 | 78 | 50 | 32 | 88 | 31 | 0.248 | 26 | 1 |
| 10 | 115 | 0 | 0 | 0 | 35.3 | 0.134 | 29 | 0 |
| 2 | 197 | 70 | 45 | 543 | 30.5 | 0.158 | 53 | 1 |
| 8 | 125 | 96 | 0 | 0 | 0 | 0.232 | 54 | 1 |
| 4 | 110 | 92 | 0 | 0 | 37.6 | 0.191 | 30 | 0 |
| 10 | 168 | 74 | 0 | 0 | 38 | 0.537 | 34 | 1 |
| 10 | 139 | 80 | 0 | 0 | 27.1 | 1.441 | 57 | 0 |
| 1 | 189 | 60 | 23 | 846 | 30.1 | 0.398 | 59 | 1 |
| 5 | 166 | 72 | 19 | 175 | 25.8 | 0.587 | 51 | 1 |
| 7 | 100 | 0 | 0 | 0 | 30 | 0.484 | 32 | 1 |
| 0 | 118 | 84 | 47 | 230 | 45.8 | 0.551 | 31 | 1 |
| 7 | 107 | 74 | 0 | 0 | 29.6 | 0.254 | 31 | 1 |
| 1 | 103 | 30 | 38 | 83 | 43.3 | 0.183 | 33 | 0 |
| 1 | 115 | 70 | 30 | 96 | 34.6 | 0.529 | 32 | 1 |
| 3 | 126 | 88 | 41 | 235 | 39.3 | 0.704 | 27 | 0 |

**Output:**

the total number of Training Data : (514, 1)

the total number of Test Data : (254, 1)

Confusion matrix

[[125 32]

[ 41 56]]

Accuracy of the classifier is 0.7125984251968503

The value of Precision 0.6363636363636364

The value of Recall 0.5773195876288659

Predicted Value for individual Test Data: [1]

**Discussion Topics:**

1. **Define Bayesian belief networks.**

Bayesian Belief Network is a graphical representation of different probabilistic relationships among random variables in a particular set. It is a classifier with no dependency on attributes i.e it is condition independent.

1. **State Bayes theorem.**

Bayes Theorem is a method to determine conditional probabilities – that is, the probability

of one event occurring given that another event has already occurred.

Equating right hand side of both the equations, we will get:

Bayes theorem in Artificial intelligence

(A|B) is known as posterior, which we need to calculate, and it will be read as Probability of hypothesis A when we have occurred an evidence B.

P(B|A) is called the likelihood, in which we consider that hypothesis is true, then we calculate the probability of evidence.

P(A) is called the prior probability, probability of hypothesis before considering the evidence

P(B) is called marginal probability, pure probability of an evidence.

1. **Why naïve Bayes is naïve?**

* Naïve Bayes algorithm is a supervised learning algorithm, which is based on Bayes theorem and used for solving classification problems.
* It is mainly used in text classification that includes a high-dimensional training dataset.
* **Naïve**: It is called Naïve because it assumes that the occurrence of a certain feature is independent of the occurrence of other features. Such as if the fruit is identified on the bases of color, shape, and taste, then red, spherical, and sweet fruit is recognized as an apple. Hence each feature individually contributes to identify that it is an apple without depending on each other.
* **Bayes**: It is called Bayes because it depends on the principle of [Bayes' Theorem](https://www.javatpoint.com/bayes-theorem-in-artifical-intelligence" \t "_blank).