Name :- Shefali Chaurasia

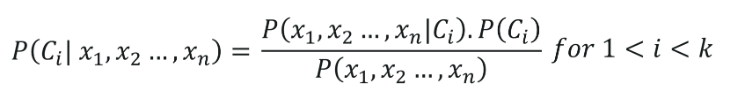
Admission no. 18SCSE1010271

5. Write a program to demonstrate the working of Naive Bayes classifier. Compute the accuracy of the classifier, considering few test data sets.

**Naive Bayes Classifier**

It is a classification algorithm based on **Bayes’s theorem** which gives an assumption of independence among predictors. In simple terms, a Naive Bayes classifier assumes that the presence of a particular feature in a class is unrelated to the presence of any other feature.

Even if the features depend on each other, all of these properties contribute to the probability independently. Naive Bayes model is easy to make and is particularly useful for comparatively large data sets. Even with a simplistic approach, Naive Bayes is known to outperform most of the classification methods in machine learning. Following is the Bayes theorem to implement the Naive Bayes Theorem.



**Advantages and Disadvantages**

The Naive Bayes classifier requires a small amount of training data to estimate the necessary parameters to get the results. They are extremely fast in nature compared to other classifiers.

The only disadvantage is that they are known to be a bad estimator.

* It is a kind of classifier that works on Bayes theorem.
* Prediction of membership probabilities is made for every class such as the probability of data points associated to a particular class.
* The class having maximum probability is appraised as the most suitable class.
* This is also referred as Maximum A Posteriori (MAP).
* The MAP for a hypothesis is:
  + 𝑀𝐴𝑃 (𝐻) = max 𝑃((𝐻|𝐸))
  + 𝑀𝐴𝑃 (𝐻) = max 𝑃((𝐻|𝐸)  ∗ (𝑃(𝐻)) /𝑃(𝐸))
  + 𝑀𝐴𝑃 (𝐻) = max(𝑃(𝐸|𝐻) ∗ 𝑃(𝐻))
  + 𝑃 (𝐸) is evidence probability, and it is used to normalize the result. Result will not be affected by removing 𝑃(𝐸).
* NB classifiers conclude that all the variables or features are not related to each other.
* Existence or absence of a variable does not impact the existence or absence of any other variable.
* Example:
  + A fruit may be observed to be an apple if it is red, round, and about 4″ in diameter.
  + In this case also even if all the features are interrelated to each other, a NB classifier will observe all of these independently contributing to the probability that the fruit is apple.
* We experiment the hypothesis in real datasets, given multiple features.
* So, computation becomes complex.

### **Types Of Naive Bayes Algorithms**

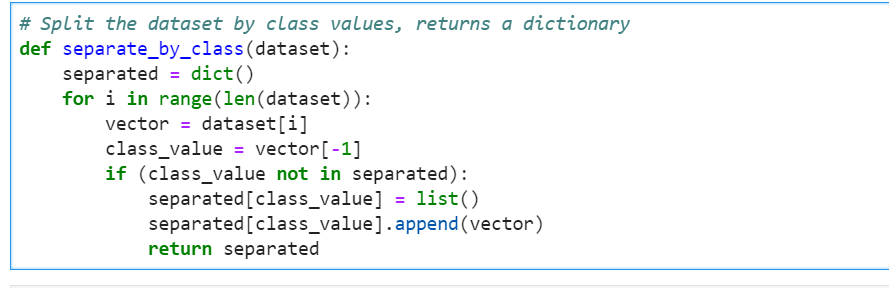
**1. Gaussian Naïve Bayes:**When characteristic values are continuous in nature then an assumption is made that the values linked with each class are dispersed according to Gaussian that is Normal Distribution.

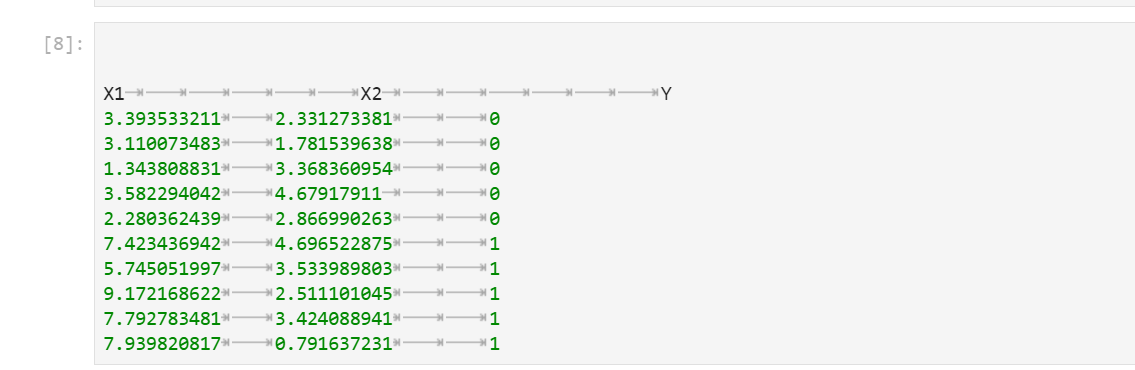
**2. Multinomial Naïve Bayes:**Multinomial Naive Bayes is favoured to use on data that is multinomial distributed. It is widely used in text classification in NLP.  Each event in text classification constitutes the presence of a word in a document.

**3. Bernoulli Naïve Bayes:**When data is dispensed according to the multivariate Bernoulli distributions then Bernoulli Naive Bayes is used. That means there exist multiple features but each one is assumed to contain binary value. So, it requires features to be binary valued.

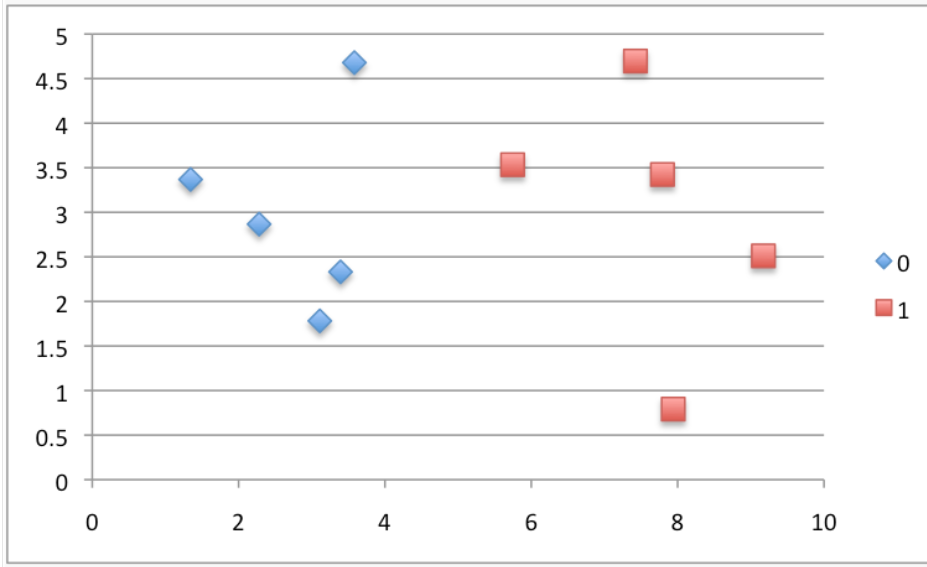
### 1.Separate By Class :-

This means that we will first need to separate our training data by class. A relatively straightforward operation. We can create a dictionary object where each key is the class value and then add a list of all the records as the value in the dictionary.





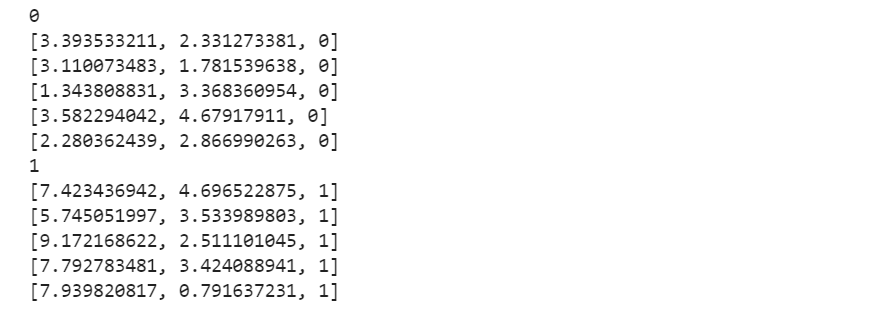
We can plot this dataset and use separate colors for each class.



Putting this all together, we can test our separate\_by\_class() function on the contrived dataset.



Output:-



### 2: Summarize Dataset

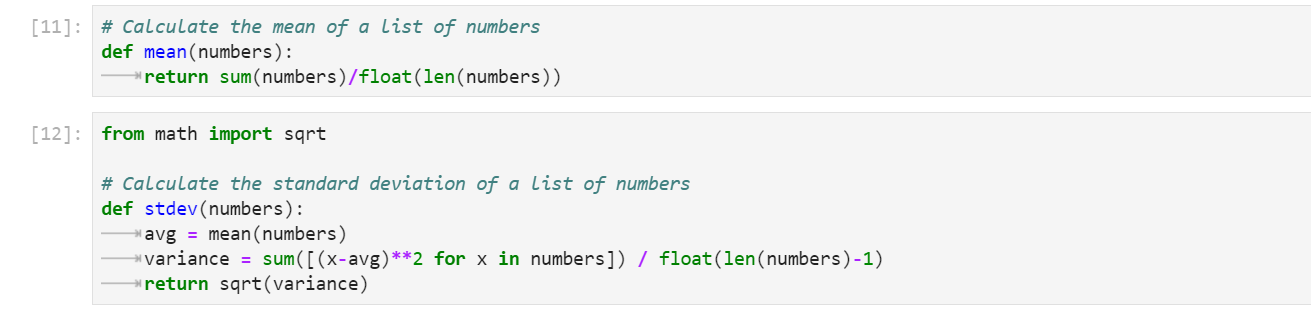
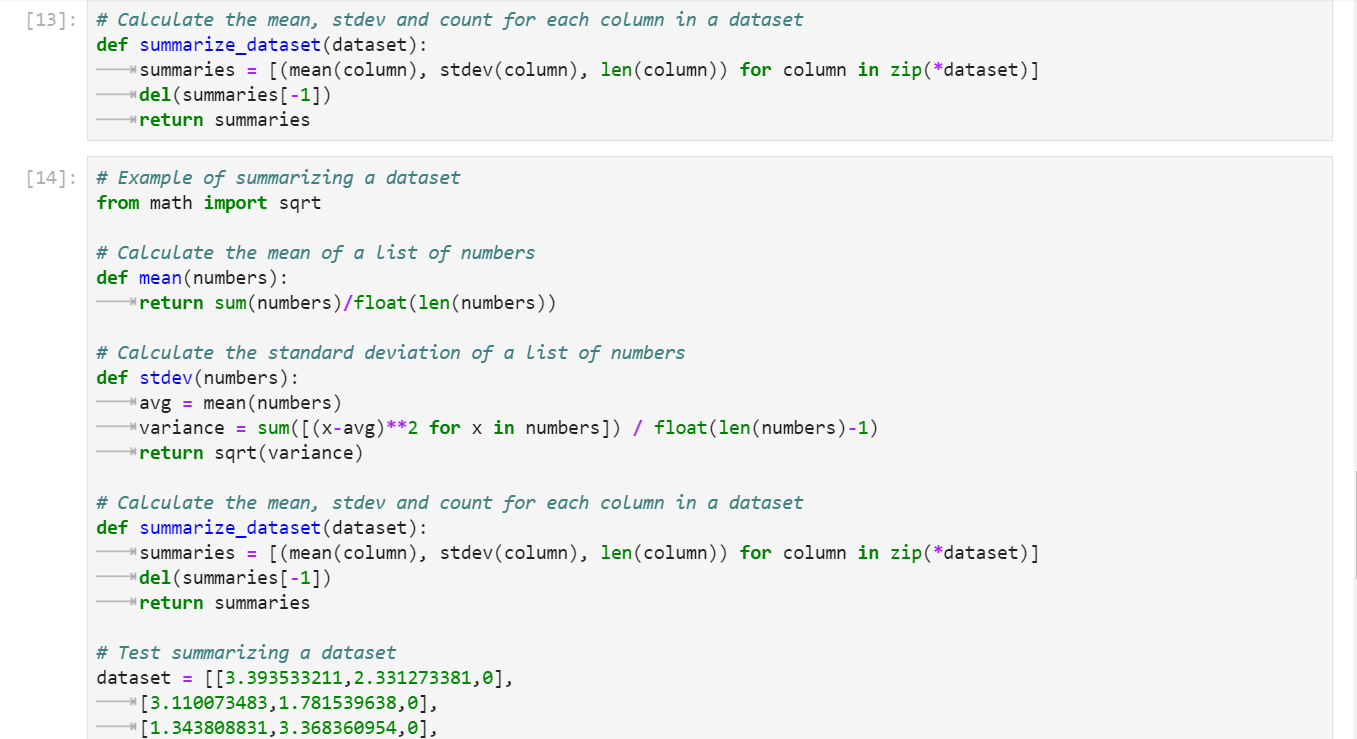
We’ll see how these statistics are used in the calculation of probabilities in a few steps. The two statistics we require from a given dataset are the mean and the standard deviation (average deviation from the mean).

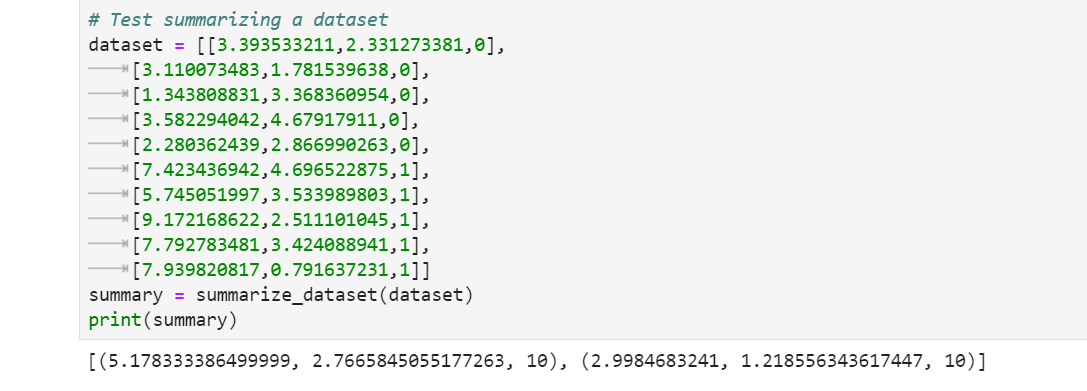
The mean is the average value and can be calculated as:

* mean = sum(x)/n \* count(x)

Where *x* is the list of values or a column we are looking.

Below is a small function named *mean()* that calculates the mean of a list of numbers

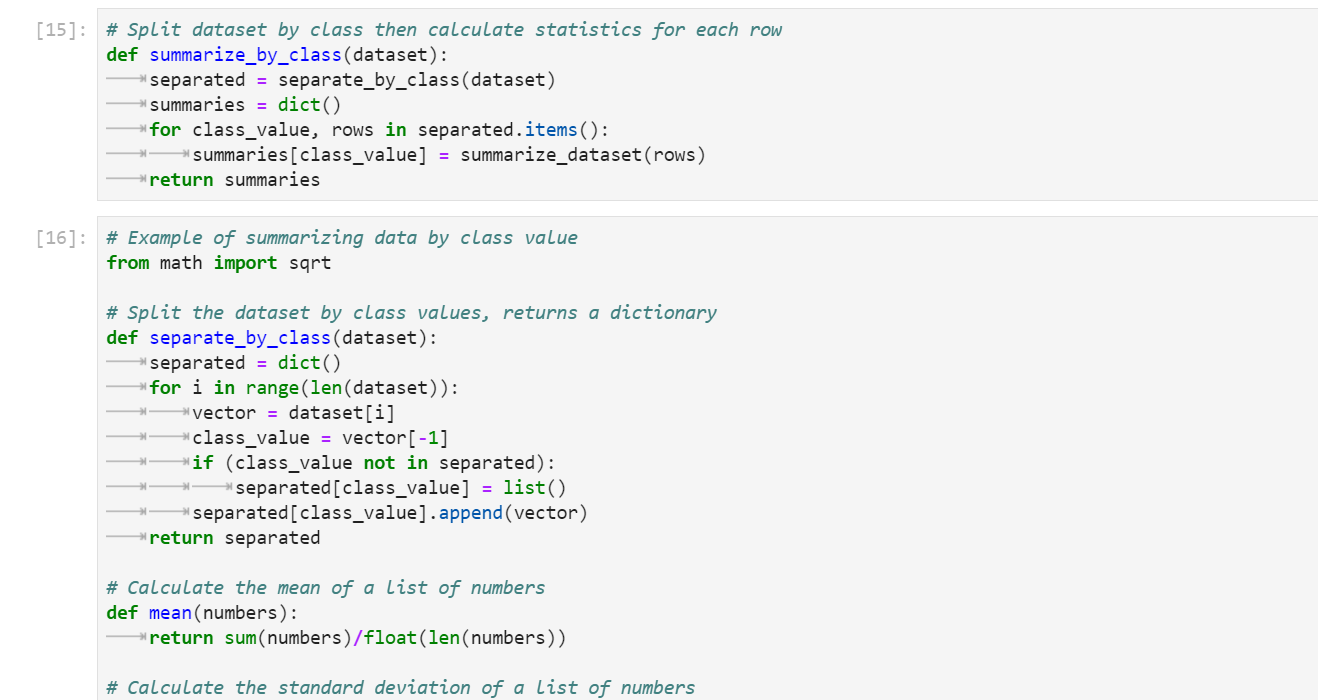
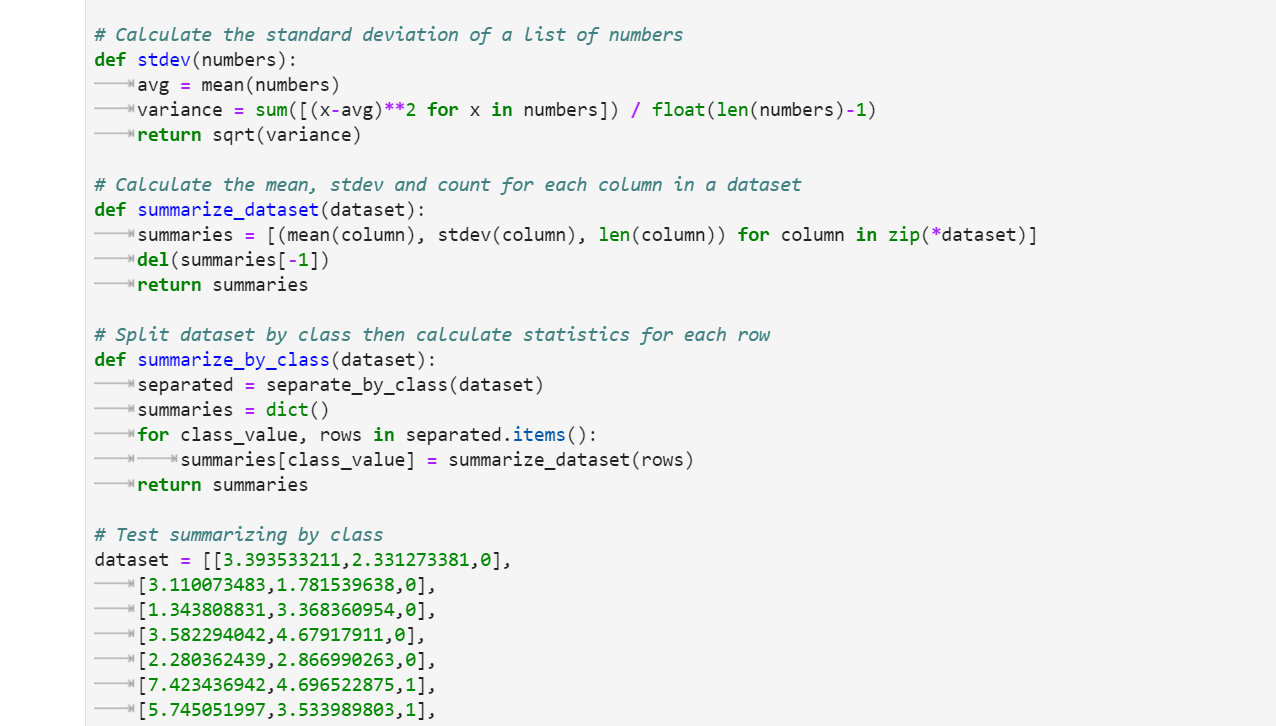


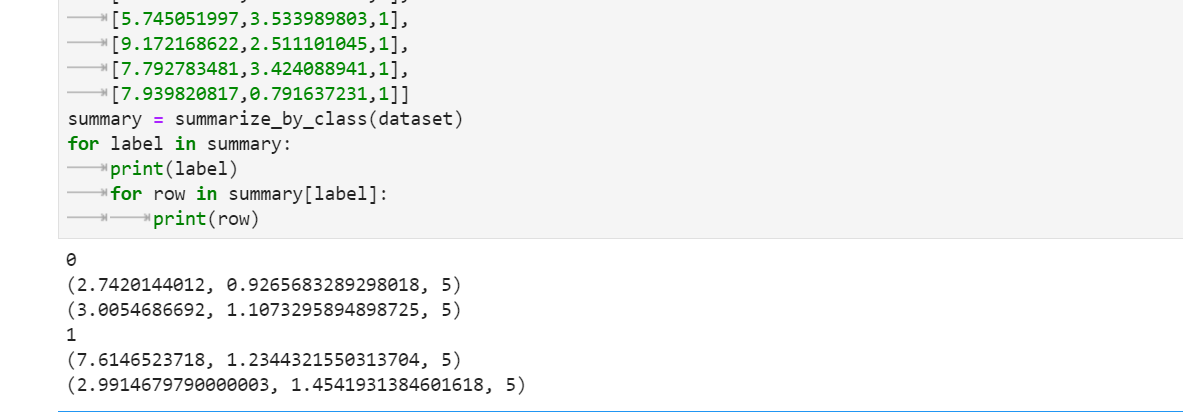
### 3: Summarize Data By Class

we have developed the *separate\_by\_class()* function to separate a dataset into rows by class. And we have developed *summarize\_dataset()* function to calculate summary statistics for each column.

We can put all of this together and summarize the columns in the dataset organized by class values.

Below is a function named *summarize\_by\_class()* that implements this operation. The dataset is first split by class, then statistics are calculated on each subset. The results in the form of a list of tuples of statistics are then stored in a dictionary by their class value



Running this example calculates the statistics for each input variable and prints them organized by class value. Interpreting the results, we can see that the X1 values for rows for class 0 have a mean value of 2.7420144012.

### 4: Gaussian Probability Density Function

One way we can do this is to assume that X1 values are drawn from a distribution, such as a bell curve or Gaussian distribution.

A [Gaussian distribution](https://machinelearningmastery.com/continuous-probability-distributions-for-machine-learning/) can be summarized using only two numbers: the mean and the standard deviation. Therefore, with a little math, we can estimate the probability of a given value. This piece of math is called a Gaussian [Probability Distribution Function](https://en.wikipedia.org/wiki/Gaussian_function) (or Gaussian PDF) and can be calculated as:

* f(x) = (1 / sqrt(2 \* PI) \* sigma) \* exp(-((x-mean)^2 / (2 \* sigma^2)))

Where *sigma* is the standard deviation for *x*, *mean* is the mean for *x* and *PI* is the value of pi.

Below is a function that implements this. I tried to split it up to make it more readable

