Assignment Report of the Naïve Bayesian Algorithm

1. Introduction to Naïve Bayes

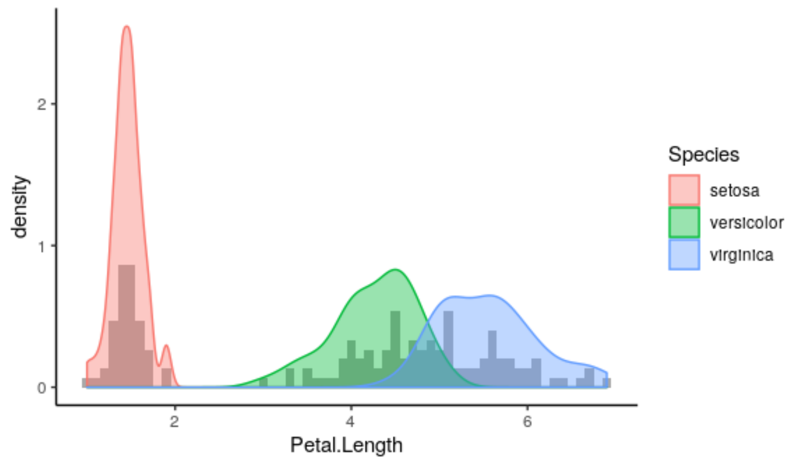
This assignment has to do with the classification of certain data a user will demand. For example, what is the probability of a certain type of flower being that type and not another. To classify the data, we use the Naïve Bayes Theorem. The formula is as follows

$$P(y|X) = \frac{P(X|y)P(y)}{P(X)}$$

This assignment which uses the Naïve Bayes Theorem to calculate the probabilities of each pattern uses two different methods.
The Naive Bayes Theorem generally has two different methods to calculate probabilities, each method for a different use.
Data classification using Histograms and Data Classification using the Gaussian Naïve Bayes model, which in our case is the best approach for our dataset.

1.1 Problems & Difficulties

One of the biggest difficulties I had in this assignment was to understand which classification method was the best approach for the Dataset. At first, I began to develop the Naïve Bayes Theorem by classifying my data using histograms, this resulted to the model giving inaccurate probabilities.



Notice the small conjunction between the two Class Types on Versicolor and Virginica. Classifying the dataset using histograms will cause the model to have inaccurate predictions.

Each dataset may need a different classification model.
Each Pattern on each Dataset at times it can be categorized differently. Either with numbers or with words.
Though the Naïve Bayesian Theorem will only process numbers, so we need to somehow convert the patterns of the dataset to processable data for the model.
When the patterns of the dataset is already categorized with a certain value as a number

(including decimal numbers) we generally need to use the Gaussian Naïve Bayes model.
If the patterns of the dataset includes strings, then we need to classify our data using histograms.
For the given assignment, I had to think of the appropriate model to classify the patterns in to satisfy the problem.

1.2 Programming Language & the Dataset of the Assignment

In the given Assignment I used the C++ programming language.
The reason why I selected C++ is generally because I wanted to develop an easy to compile and use algorithm for anybody to embed into their computers and execute it for themselves.
Another reason why I selected C++ is because I can keep everything organized Object Oriented wise when it comes to the programmer's side of things.
A better choice would be Python due to its ease of use and how easy it is to develop Machine Learning and Data Science algorithms.
Another reason why someone would want to pick Python is because of the multiple tools that are available online to help the programmer to develop a good model.
But in my case I did not select Python because Data Visualization was not necessary for this assignment.

The Dataset of the Assignment involves the Iris Dataset.
The dataset contains a total of 150 patterns, a total of 4 Features and a total of 3 Class Types.
The Class Types for the Iris Flower are:
Iris-Setosa, Iris-Versicolor, Iris-Virginica
The values are categorized as Strings (words)

The Features of each pattern for the Iris Flower are:
Sepal Length, Sepal Width, Petal Length, Petal Width
The values are categorized as numbers (Decimals)

1.3 Methods used to calculate the Probabilities

For this assignment I implemented two different classification methods.
The first method is classifying the dataset using Histograms.
The other method is classifying the dataset by calculating the Mean and Standard Deviation of each Feature and Class Type

At first, I implemented the method that classifies the dataset using Histograms, which it will not give a 100% Accurate Result.

Afterwards, I realized that this method was not a good approach, so I implemented a different method that classifies the dataset using the Gaussian Naïve Bayes Theorem which it will give better and more accurate results.

The reason why the first method did not work is because the dataset is being categorized into smaller more processable values and later on used to calculate the probability. This will cause dataset to shrink. This is called Data Shrinkage a phenomenon seen in Data Science and analytics.

2 Management & Explanation of the Iris Dataset

The dataset contains a total of 150 patterns (lines)
4 Features and 3 Class Types.
It is important to state the fact that in the dataset the class types are equally distributed.
We have a total of 3 Class Types which means the probability of a flower being one of those 3 types is 1/3 for 150 patterns. In other words, the real probability is 33.33~%

To manage our dataset and to be able to see whether it works on patterns it hasn't seen before we split the Dataset into two different parts.
The train Set and the Test Set.
There are 3 executables that implement both the methods in different data set ranges.
1) Gaussian Naïve Bayes with a Train Set of 80% and a Test Set of 20%.
2) Gaussian Naïve Bayes with a Train Set of 96% and a Test Set of 4%.
3) Naïve Bayes using Histograms with a 90% Train Set and a 10% Test Set.
In the main file, NaiveBayesianClassifier_Iris.cpp, you can freely change those values by carefully changing the parameters.

2.1 The Characteristics of the Dataset

The Feature Types are:
Sepal Length, Sepal Width, Petal Length, Petal Width
Characteristics of the Features: Decimal values (e.g. 2.3, 5.8, 0.2, 4.5)

2.2 The Class types of the Dataset

The Class Types of the Dataset for each flower type are:
Iris Setosa, Iris Versicolor, Iris Virginica
Characteristics of the Class Type: Class, Type/Naming or Title(string).

3. Classification using the Gaussian Naïve Bayes method

Training the model using the Gaussian Naïve Bayes method will give as an accurate result of each probability of each pattern.

The training of the Gaussian model has 4 steps

3.1 Calculating the Mean for each Pattern

The first step for the Gaussian Classifier, we need to calculate the Mean of every Class Type and every Feature using all the patterns in the dataset. The formula to calculate the Mean is as follows.

$$\mu = \frac{1}{n}\sum_{i=1}^{n} x_i$$

n being the number of patterns (For our dataset 150)
i = 1,2, …, 150 being the iteration
Xi being the data that is being processed in the i_{th} pattern. (Aka the value of the current feature being processed)

3.2 Calculating the Standard Deviation for each Pattern

The second step after calculating the Mean we need to calculate the Standard Deviation of every Class Type and every Feature. The formula to calculate the Standard Deviation is as follows.

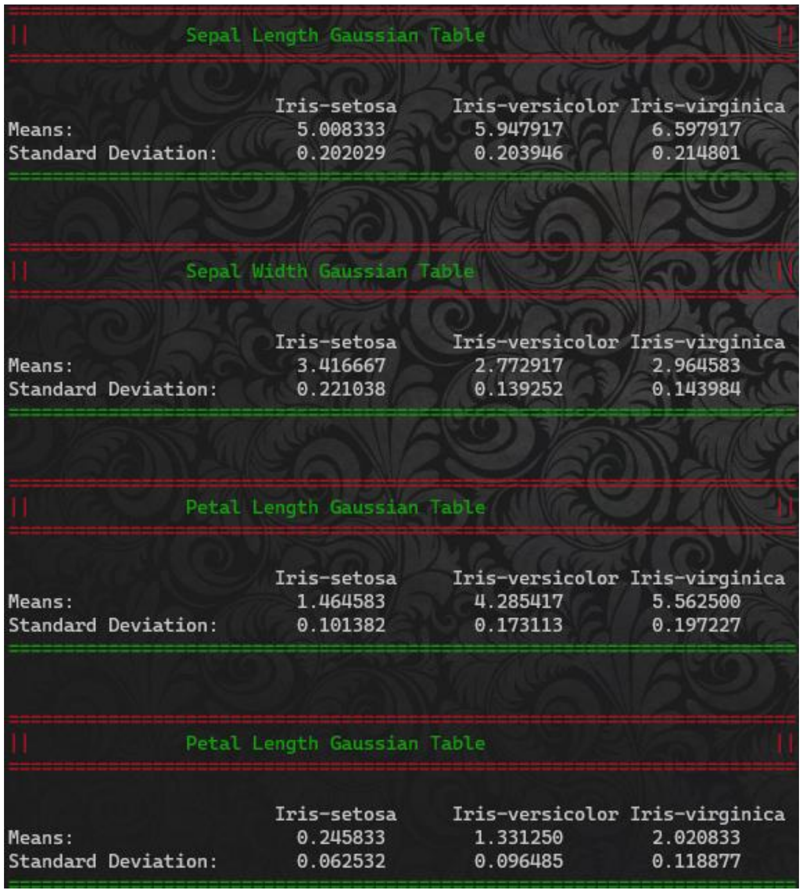
$$\sigma = \left[\frac{1}{n-1}\sum_{i=1}^{n}(x_i - \mu)^2\right]^{0.5}$$

μ being the Mean that we calculated before.

3.3 Frequency Table of the Gaussian Naïve Bayes Classification method

It’s important to be able to see what happened after training the model.

We can see all the values that have changed. In our case the Means and the Standard Deviation for each Class Type.
Train Set: 96%
Test Set: 4%



```
||            Sepal Length Gaussian Table                              ||

                          Iris-setosa      Iris-versicolor Iris-virginica
Means:                      5.008333         5.947917        6.597917
Standard Deviation:         0.202029         0.203946        0.214801

||            Sepal Width Gaussian Table                               ||

                          Iris-setosa      Iris-versicolor Iris-virginica
Means:                      3.416667         2.772917        2.964583
Standard Deviation:         0.221038         0.139252        0.143984

||            Petal Length Gaussian Table                              ||

                          Iris-setosa      Iris-versicolor Iris-virginica
Means:                      1.464583         4.285417        5.562500
Standard Deviation:         0.101382         0.173113        0.197227

||            Petal Length Gaussian Table                              ||

                          Iris-setosa      Iris-versicolor Iris-virginica
Means:                      0.245833         1.331250        2.020833
Standard Deviation:         0.062532         0.096485        0.118877
```

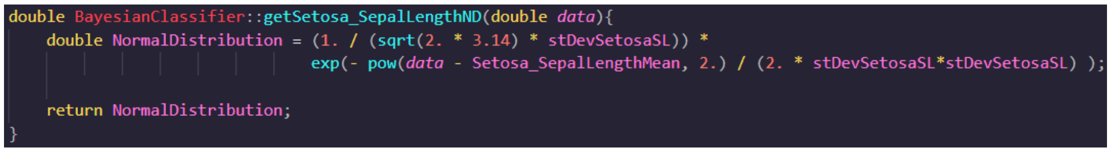
3.4 Calculating Probabilities using the Normal Distribution formula

After we have successfully trained our model and have all of our Mean and SD values trained, on the third step we can extract the likelihood of each flower type of a certain pattern from the Test Set.
The formula of the Normal Distribution is as follows.

$$f(x) = \frac{1}{\sqrt{2\pi\sigma}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

π is pi 3.14
σ is the Standard Deviation
e being the exponential
x being our data that is currently being processed for each Feature.
μ is the Mean

The calculation of the formula for the Sepal Length of the Setosa is as follows.



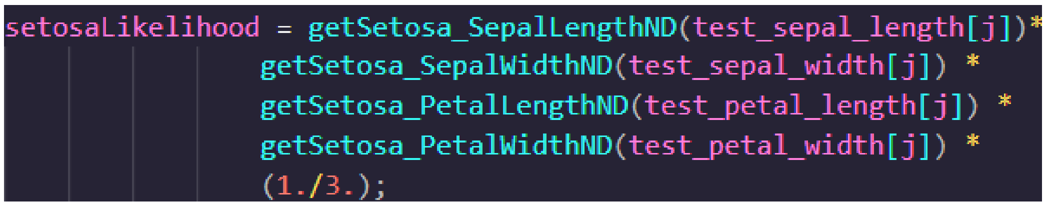
```
double BayesianClassifier::getSetosa_SepalLengthND(double data){
    double NormalDistribution = (1. / (sqrt(2. * 3.14) * stDevSetosaSL)) *
                                exp(- pow(data - Setosa_SepalLengthMean, 2.) / (2. * stDevSetosaSL*stDevSetosaSL) );

    return NormalDistribution;
}
```

The fourth most important step is using all of the things we have gathered and finally calculating the probability.
To do that we need to use the Naïve Bayes Theorem Formula given on the Introduction

$$P(y|X) = \frac{P(X|y)P(y)}{P(X)}$$

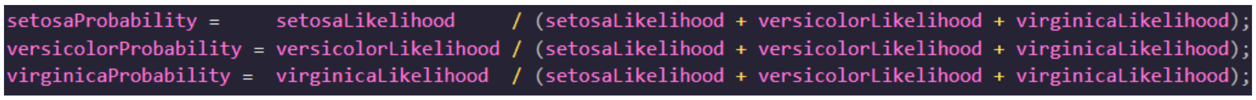
P(X|y) being the Normal Distribution of all of the features using the Test Set’s pattern.
P(y) being the distribution for each Class Type in our case it’s equally Distributed so 1/3 and P(X) can be ignored because we want to “relatively” the data to the Class Type.

The formula to calculate the Likelihood for the Setosa is as follows.



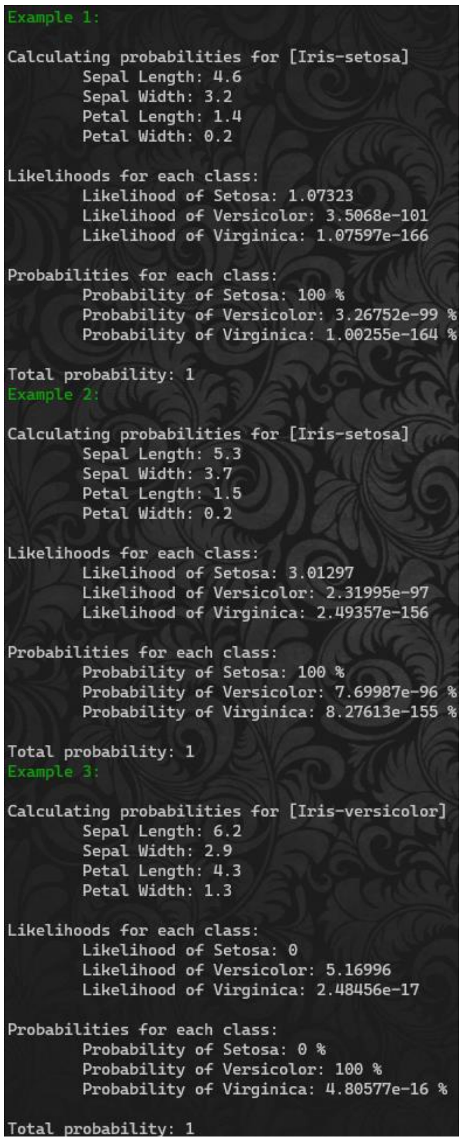
```
setosaLikelihood = getSetosa_SepalLengthND(test_sepal_length[j])*
                   getSetosa_SepalWidthND(test_sepal_width[j]) *
                   getSetosa_PetalLengthND(test_petal_length[j]) *
                   getSetosa_PetalWidthND(test_petal_width[j]) *
                   (1./3.);
```

The final result will be a likelihood, not a real probability. So we need to convert it into a real probability.
This is done as follows

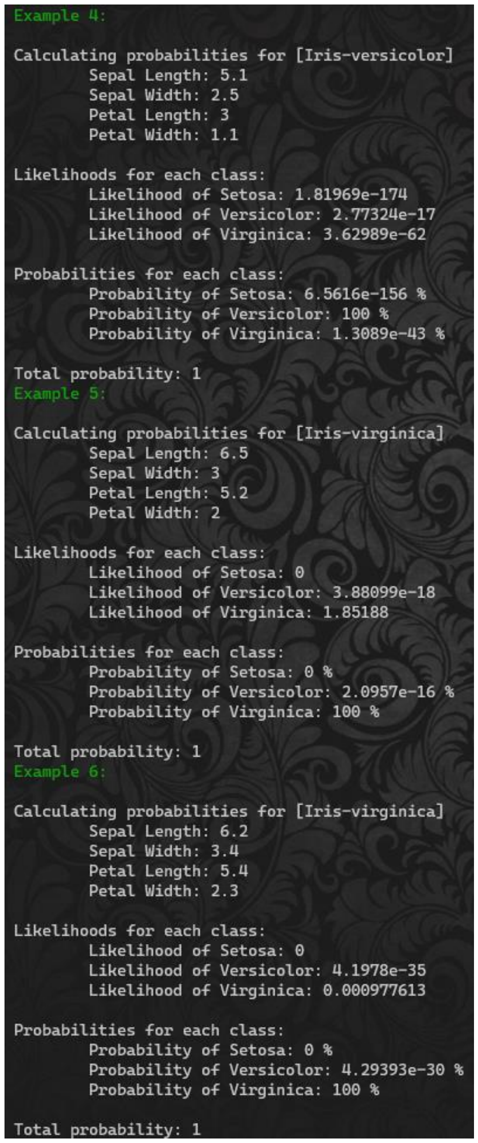


```
setosaProbability =     setosaLikelihood     / (setosaLikelihood + versicolorLikelihood + virginicaLikelihood);
versicolorProbability = versicolorLikelihood / (setosaLikelihood + versicolorLikelihood + virginicaLikelihood);
virginicaProbability =  virginicaLikelihood  / (setosaLikelihood + versicolorLikelihood + virginicaLikelihood);
```

The results from the Test Set is as follows.



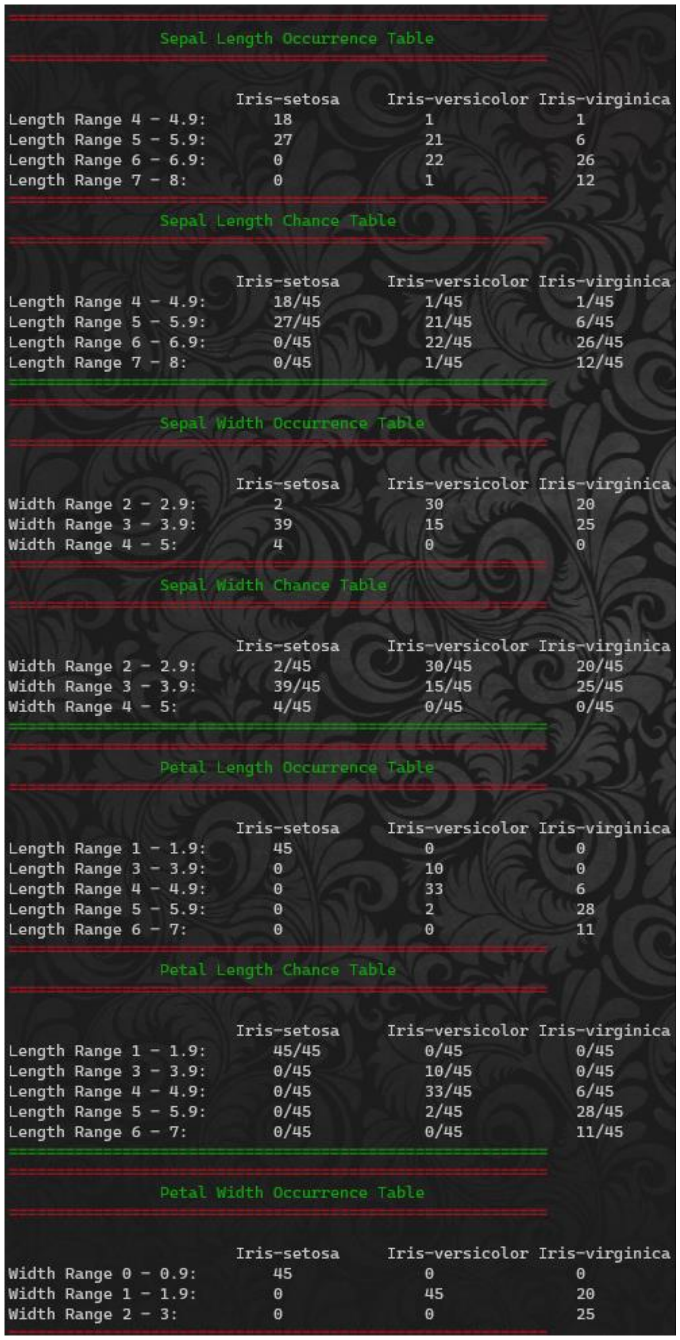
```
Example 1:

Calculating probabilities for [Iris-setosa]
        Sepal Length: 4.6
        Sepal Width: 3.2
        Petal Length: 1.4
        Petal Width: 0.2

Likelihoods for each class:
        Likelihood of Setosa: 1.07323
        Likelihood of Versicolor: 3.5068e-101
        Likelihood of Virginica: 1.07597e-166

Probabilities for each class:
        Probability of Setosa: 100 %
        Probability of Versicolor: 3.26752e-99 %
        Probability of Virginica: 1.00255e-164 %

Total probability: 1
Example 2:

Calculating probabilities for [Iris-setosa]
        Sepal Length: 5.3
        Sepal Width: 3.7
        Petal Length: 1.5
        Petal Width: 0.2

Likelihoods for each class:
        Likelihood of Setosa: 3.01297
        Likelihood of Versicolor: 2.31995e-97
        Likelihood of Virginica: 2.49357e-156

Probabilities for each class:
        Probability of Setosa: 100 %
        Probability of Versicolor: 7.69987e-96 %
        Probability of Virginica: 8.27613e-155 %

Total probability: 1
Example 3:

Calculating probabilities for [Iris-versicolor]
        Sepal Length: 6.2
        Sepal Width: 2.9
        Petal Length: 4.3
        Petal Width: 1.3

Likelihoods for each class:
        Likelihood of Setosa: 0
        Likelihood of Versicolor: 5.16996
        Likelihood of Virginica: 2.48456e-17

Probabilities for each class:
        Probability of Setosa: 0 %
        Probability of Versicolor: 100 %
        Probability of Virginica: 4.80577e-16 %

Total probability: 1
```



```
Example 4:

Calculating probabilities for [Iris-versicolor]
        Sepal Length: 5.1
        Sepal Width: 2.5
        Petal Length: 3
        Petal Width: 1.1

Likelihoods for each class:
        Likelihood of Setosa: 1.81969e-174
        Likelihood of Versicolor: 2.77324e-17
        Likelihood of Virginica: 3.62989e-62

Probabilities for each class:
        Probability of Setosa: 6.5616e-156 %
        Probability of Versicolor: 100 %
        Probability of Virginica: 1.3089e-43 %

Total probability: 1
Example 5:

Calculating probabilities for [Iris-virginica]
        Sepal Length: 6.5
        Sepal Width: 3
        Petal Length: 5.2
        Petal Width: 2

Likelihoods for each class:
        Likelihood of Setosa: 0
        Likelihood of Versicolor: 3.88099e-18
        Likelihood of Virginica: 1.85188

Probabilities for each class:
        Probability of Setosa: 0 %
        Probability of Versicolor: 2.0957e-16 %
        Probability of Virginica: 100 %

Total probability: 1
Example 6:

Calculating probabilities for [Iris-virginica]
        Sepal Length: 6.2
        Sepal Width: 3.4
        Petal Length: 5.4
        Petal Width: 2.3

Likelihoods for each class:
        Likelihood of Setosa: 0
        Likelihood of Versicolor: 4.1978e-35
        Likelihood of Virginica: 0.000977613

Probabilities for each class:
        Probability of Setosa: 0 %
        Probability of Versicolor: 4.29393e-30 %
        Probability of Virginica: 100 %

Total probability: 1
```

4. Classification using Histograms

In the case of the Iris Dataset where all of the feature’s values are decimal values, with this method, the result will not be as accurate as the Gaussian Naïve Bayes method due to Data Shrinkage.
With this method we can build a Frequency Table that will help us find the probabilities



```
========================================================
                Sepal Length Occurrence Table
========================================================

                        Iris-setosa     Iris-versicolor Iris-virginica
Length Range 4 - 4.9:       18              1               1
Length Range 5 - 5.9:       27              21              6
Length Range 6 - 6.9:       0               22              26
Length Range 7 - 8:         0               1               12
========================================================
                Sepal Length Chance Table
========================================================

                        Iris-setosa     Iris-versicolor Iris-virginica
Length Range 4 - 4.9:       18/45           1/45            1/45
Length Range 5 - 5.9:       27/45           21/45           6/45
Length Range 6 - 6.9:       0/45            22/45           26/45
Length Range 7 - 8:         0/45            1/45            12/45
========================================================
========================================================
                Sepal Width Occurrence Table
========================================================

                        Iris-setosa     Iris-versicolor Iris-virginica
Width Range 2 - 2.9:        2               30              20
Width Range 3 - 3.9:        39              15              25
Width Range 4 - 5:          4               0               0
========================================================
                Sepal Width Chance Table
========================================================

                        Iris-setosa     Iris-versicolor Iris-virginica
Width Range 2 - 2.9:        2/45            30/45           20/45
Width Range 3 - 3.9:        39/45           15/45           25/45
Width Range 4 - 5:          4/45            0/45            0/45
========================================================
========================================================
                Petal Length Occurrence Table
========================================================

                        Iris-setosa     Iris-versicolor Iris-virginica
Length Range 1 - 1.9:       45              0               0
Length Range 3 - 3.9:       0               10              0
Length Range 4 - 4.9:       0               33              6
Length Range 5 - 5.9:       0               2               28
Length Range 6 - 7:         0               0               11
========================================================
                Petal Length Chance Table
========================================================

                        Iris-setosa     Iris-versicolor Iris-virginica
Length Range 1 - 1.9:       45/45           0/45            0/45
Length Range 3 - 3.9:       0/45            10/45           0/45
Length Range 4 - 4.9:       0/45            33/45           6/45
Length Range 5 - 5.9:       0/45            2/45            28/45
Length Range 6 - 7:         0/45            0/45            11/45
========================================================
========================================================
                Petal Width Occurrence Table
========================================================

                        Iris-setosa     Iris-versicolor Iris-virginica
Width Range 0 - 0.9:        45              0               0
Width Range 1 - 1.9:        0               45              20
Width Range 2 - 3:          0               0               25
========================================================
```

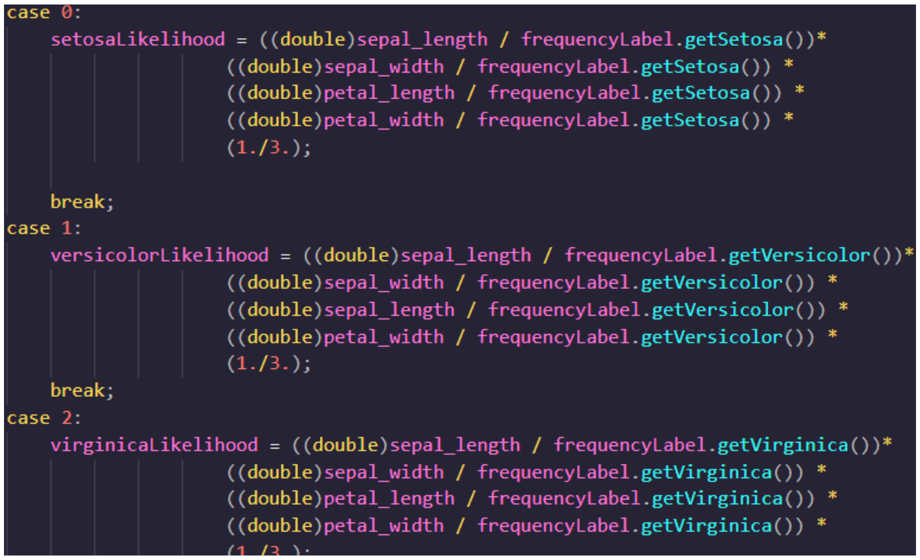
4.1 The Frequency Table by Classifying with Histograms

As long as the frequency table has been created using the Train set, we can use the Test Set's patterns to extract probabilities.

To calculate the probabilities, I used the Naïve Bayes Theorem formula which is as follows.

$$P(y|X) = \frac{P(X|y)P(y)}{P(X)}$$

The calculations are as follows

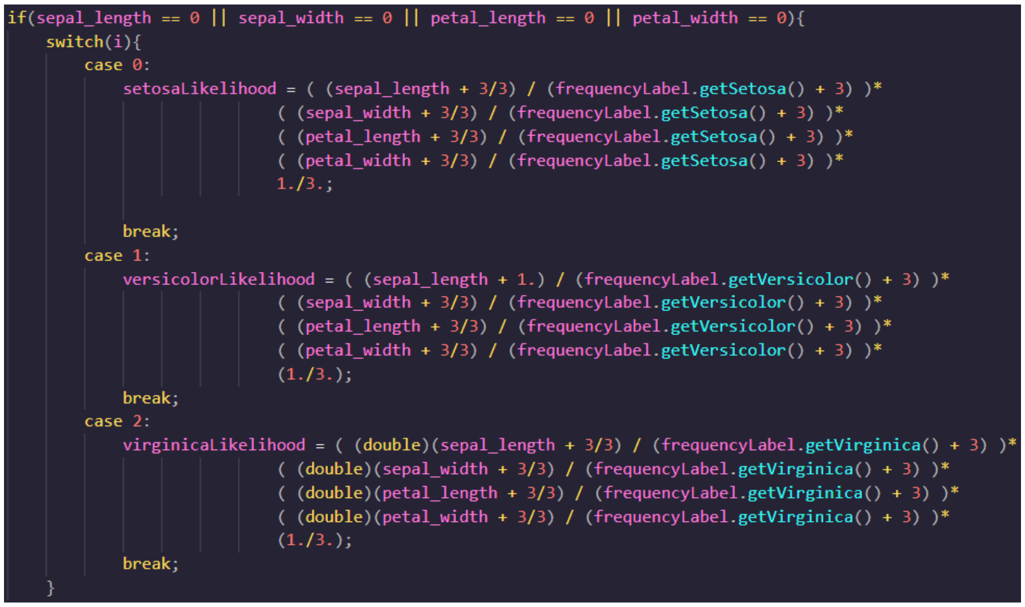


```
case 0:
    setosaLikelihood = ((double)sepal_length / frequencyLabel.getSetosa())*
                    ((double)sepal_width / frequencyLabel.getSetosa()) *
                    ((double)petal_length / frequencyLabel.getSetosa()) *
                    ((double)petal_width / frequencyLabel.getSetosa()) *
                    (1./3.);

    break;
case 1:
    versicolorLikelihood = ((double)sepal_length / frequencyLabel.getVersicolor())*
                    ((double)sepal_width / frequencyLabel.getVersicolor()) *
                    ((double)sepal_length / frequencyLabel.getVersicolor()) *
                    ((double)petal_width / frequencyLabel.getVersicolor()) *
                    (1./3.);
    break;
case 2:
    virginicaLikelihood = ((double)sepal_length / frequencyLabel.getVirginica())*
                    ((double)sepal_width / frequencyLabel.getVirginica()) *
                    ((double)petal_length / frequencyLabel.getVirginica()) *
                    ((double)petal_width / frequencyLabel.getVirginica()) *
                    (1./3.);
```

The sepal length, sepal width, petal length, petal width, values are the counts that we have on the frequency table that are also appropriate to the current pattern we are processing.

It's important to note that the Zero-Frequency Problem here is very apparent.
Because of that, if there's a 0 value in any of those counts in the frequency table, we want to avoid making our likelihoods 0 because of that.
To stop that from happening if there's a 0 in the model's frequency table using that certain pattern we need to add 1 to every single count from the frequency table to avoid getting 0 in our likelihoods.

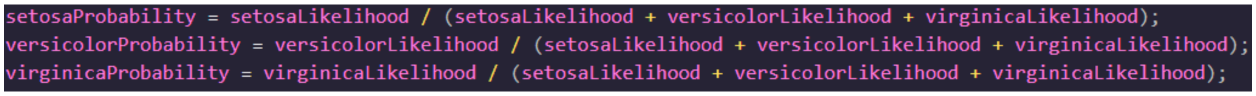
The calculation is as follows.



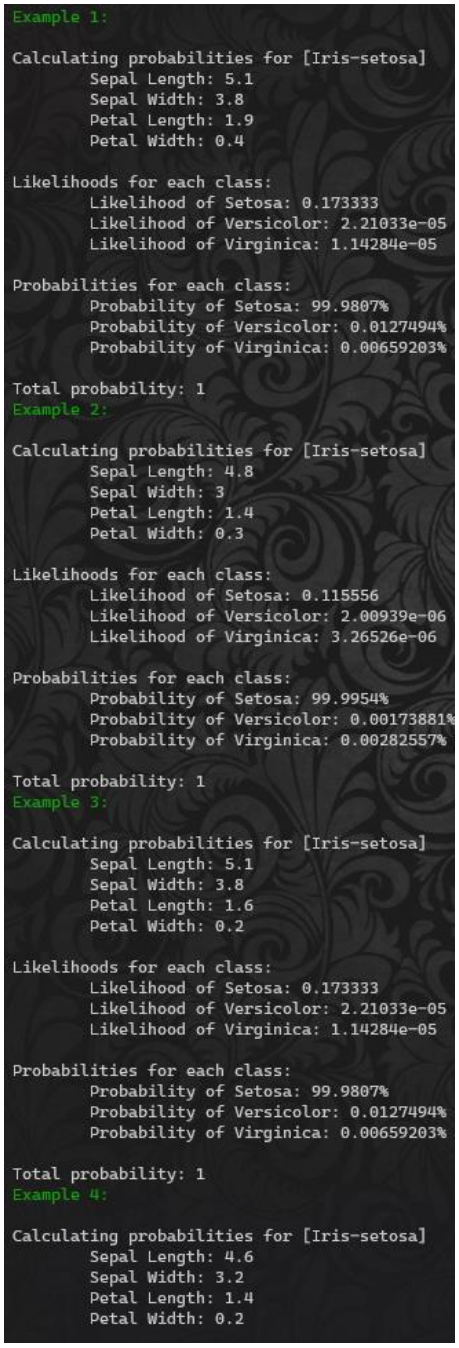
```
if(sepal_length == 0 || sepal_width == 0 || petal_length == 0 || petal_width == 0){
    switch(i){
        case 0:
            setosaLikelihood = ( (sepal_length + 3/3) / (frequencyLabel.getSetosa() + 3) )*
                            ( (sepal_width + 3/3) / (frequencyLabel.getSetosa() + 3) )*
                            ( (petal_length + 3/3) / (frequencyLabel.getSetosa() + 3) )*
                            ( (petal_width + 3/3) / (frequencyLabel.getSetosa() + 3) )*
                            1./3.;

            break;
        case 1:
            versicolorLikelihood = ( (sepal_length + 1.) / (frequencyLabel.getVersicolor() + 3) )*
                            ( (sepal_width + 3/3) / (frequencyLabel.getVersicolor() + 3) )*
                            ( (petal_length + 3/3) / (frequencyLabel.getVersicolor() + 3) )*
                            ( (petal_width + 3/3) / (frequencyLabel.getVersicolor() + 3) )*
                            (1./3.);
            break;
        case 2:
            virginicaLikelihood = ( (double)(sepal_length + 3/3) / (frequencyLabel.getVirginica() + 3) )*
                            ( (double)(sepal_width + 3/3) / (frequencyLabel.getVirginica() + 3) )*
                            ( (double)(petal_length + 3/3) / (frequencyLabel.getVirginica() + 3) )*
                            ( (double)(petal_width + 3/3) / (frequencyLabel.getVirginica() + 3) )*
                            (1./3.);
            break;
    }
```

4.2 Calculating the Probabilities by Classifying with Histograms

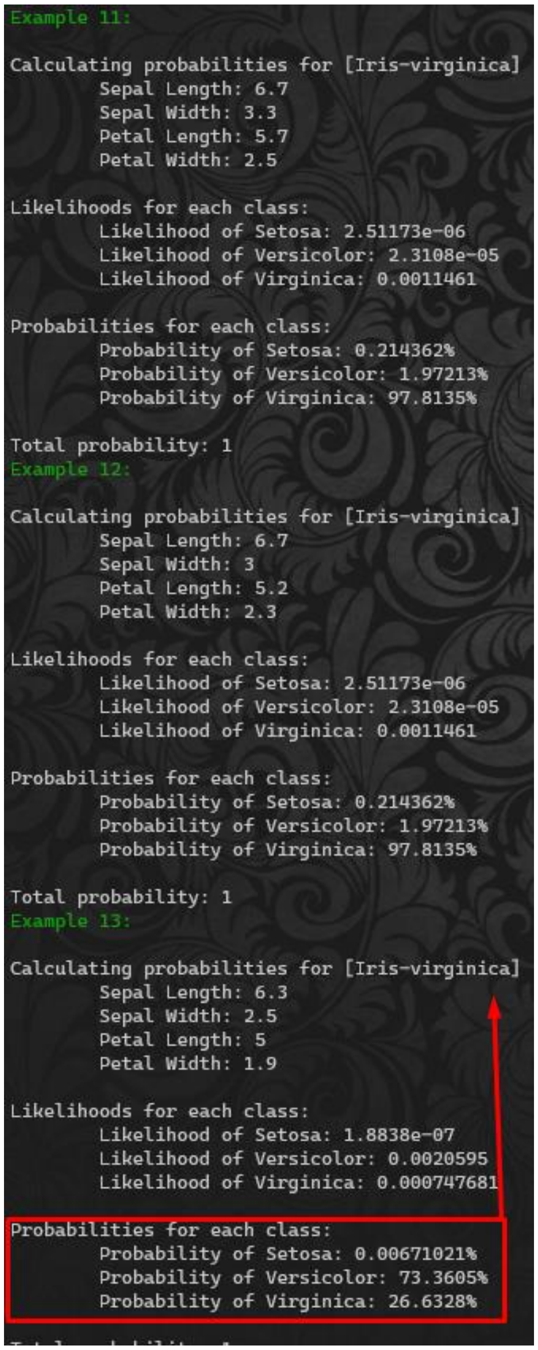
As I mentioned before, classifying the Iris dataset using Histograms will not give accurate predictions due to the Data Shrinkage problem.
After calculating the likelihoods we need to convert them into real probabilities.



```
setosaProbability = setosaLikelihood / (setosaLikelihood + versicolorLikelihood + virginicaLikelihood);
versicolorProbability = versicolorLikelihood / (setosaLikelihood + versicolorLikelihood + virginicaLikelihood);
virginicaProbability = virginicaLikelihood / (setosaLikelihood + versicolorLikelihood + virginicaLikelihood);
```



```
Example 1:

Calculating probabilities for [Iris-setosa]
        Sepal Length: 5.1
        Sepal Width: 3.8
        Petal Length: 1.9
        Petal Width: 0.4

Likelihoods for each class:
        Likelihood of Setosa: 0.173333
        Likelihood of Versicolor: 2.21033e-05
        Likelihood of Virginica: 1.14284e-05

Probabilities for each class:
        Probability of Setosa: 99.9807%
        Probability of Versicolor: 0.0127494%
        Probability of Virginica: 0.00659203%

Total probability: 1
Example 2:

Calculating probabilities for [Iris-setosa]
        Sepal Length: 4.8
        Sepal Width: 3
        Petal Length: 1.4
        Petal Width: 0.3

Likelihoods for each class:
        Likelihood of Setosa: 0.115556
        Likelihood of Versicolor: 2.00939e-06
        Likelihood of Virginica: 3.26526e-06

Probabilities for each class:
        Probability of Setosa: 99.9954%
        Probability of Versicolor: 0.00173881%
        Probability of Virginica: 0.00282557%

Total probability: 1
Example 3:

Calculating probabilities for [Iris-setosa]
        Sepal Length: 5.1
        Sepal Width: 3.8
        Petal Length: 1.6
        Petal Width: 0.2

Likelihoods for each class:
        Likelihood of Setosa: 0.173333
        Likelihood of Versicolor: 2.21033e-05
        Likelihood of Virginica: 1.14284e-05

Probabilities for each class:
        Probability of Setosa: 99.9807%
        Probability of Versicolor: 0.0127494%
        Probability of Virginica: 0.00659203%

Total probability: 1
Example 4:

Calculating probabilities for [Iris-setosa]
        Sepal Length: 4.6
        Sepal Width: 3.2
        Petal Length: 1.4
        Petal Width: 0.2
```



```
Example 11:

Calculating probabilities for [Iris-virginica]
        Sepal Length: 6.7
        Sepal Width: 3.3
        Petal Length: 5.7
        Petal Width: 2.5

Likelihoods for each class:
        Likelihood of Setosa: 2.51173e-06
        Likelihood of Versicolor: 2.3108e-05
        Likelihood of Virginica: 0.0011461

Probabilities for each class:
        Probability of Setosa: 0.214362%
        Probability of Versicolor: 1.97213%
        Probability of Virginica: 97.8135%

Total probability: 1
Example 12:

Calculating probabilities for [Iris-virginica]
        Sepal Length: 6.7
        Sepal Width: 3
        Petal Length: 5.2
        Petal Width: 2.3

Likelihoods for each class:
        Likelihood of Setosa: 2.51173e-06
        Likelihood of Versicolor: 2.3108e-05
        Likelihood of Virginica: 0.0011461

Probabilities for each class:
        Probability of Setosa: 0.214362%
        Probability of Versicolor: 1.97213%
        Probability of Virginica: 97.8135%

Total probability: 1
Example 13:

Calculating probabilities for [Iris-virginica]
        Sepal Length: 6.3
        Sepal Width: 2.5
        Petal Length: 5
        Petal Width: 1.9

Likelihoods for each class:
        Likelihood of Setosa: 1.8838e-07
        Likelihood of Versicolor: 0.0020595
        Likelihood of Virginica: 0.000747681

Probabilities for each class:
        Probability of Setosa: 0.00671021%
        Probability of Versicolor: 73.3605%
        Probability of Virginica: 26.6328%
```

5 Bibliographies

One of the most important sources I used to implement the Gaussian Naïve Bayes is the following.
This source explains in detail on how to implement a Naïve Bayes classifier.
Introduction To The Naïve Bayes Algorithm From Nagesh Singh Chauhan

A very useful source I used to implement the Naïve Bayes and understand it better is the following.
It also contains an explanation to the Zero-Frequency Problem.
A Tutorial on Naive Bayes Classification From Choochart Haruechaiyasak