**Neural Network Report**

|  |  |  |  |
| --- | --- | --- | --- |
| Name | ID | Section | Department |
| 1. محمد سامي عبد الكريم أحمد حسان | 20201700691 | **Section 8** | Computer Science |
| 1. يوسف عبدالحميد فرج الحميد الكمار | 20201701249 | **Section 12** | Computer Science |
| 1. كريم حسين عبده جويلي | 20201700605 | **Section 7** | Computer Science |
| 1. كمال كريم فؤاد رمضان | 20191700897 | **Section 7** | Computer Science |
| 1. ماريا سامح مكرم بولس | 20201700633 | **Section 7** | Computer Science |

* Data analysis:

1. **Mean Imputation:** any missing values were replaced with the mean of the corresponding feature. This step is important to ensure that the missing values do not unduly influence the analysis.
2. **Normalization:** the data was normalized to ensure that all features were on the scale from 0 to 1.
3. **Standardization:** transform features to have a mean of 0 and a standard deviation of 1, while keeping the distribution shape the same.

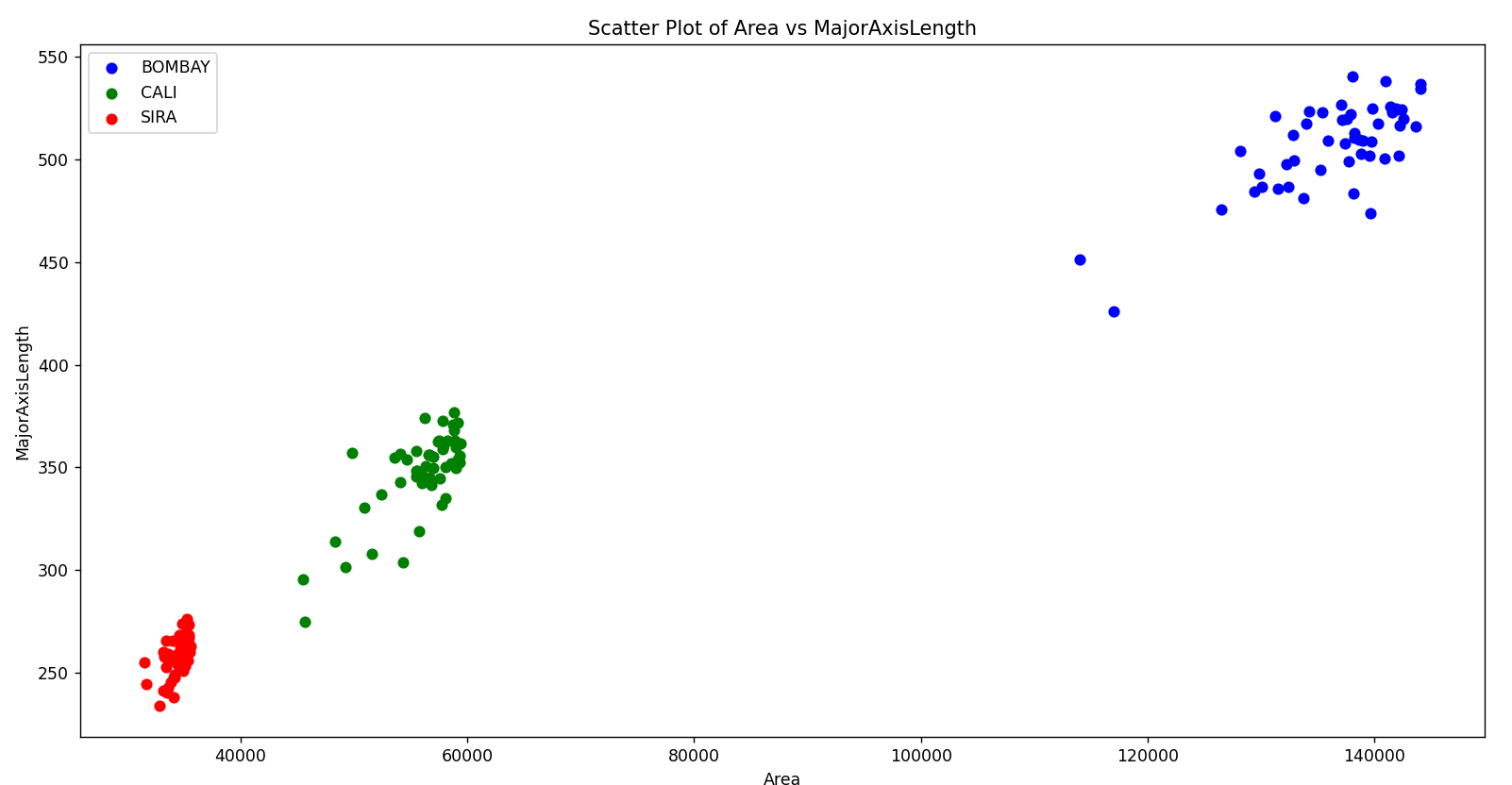
Data visualization

1. Area, Perimeter

A green and blue dots

Description automatically generated

1. Area, MajorAxisLength

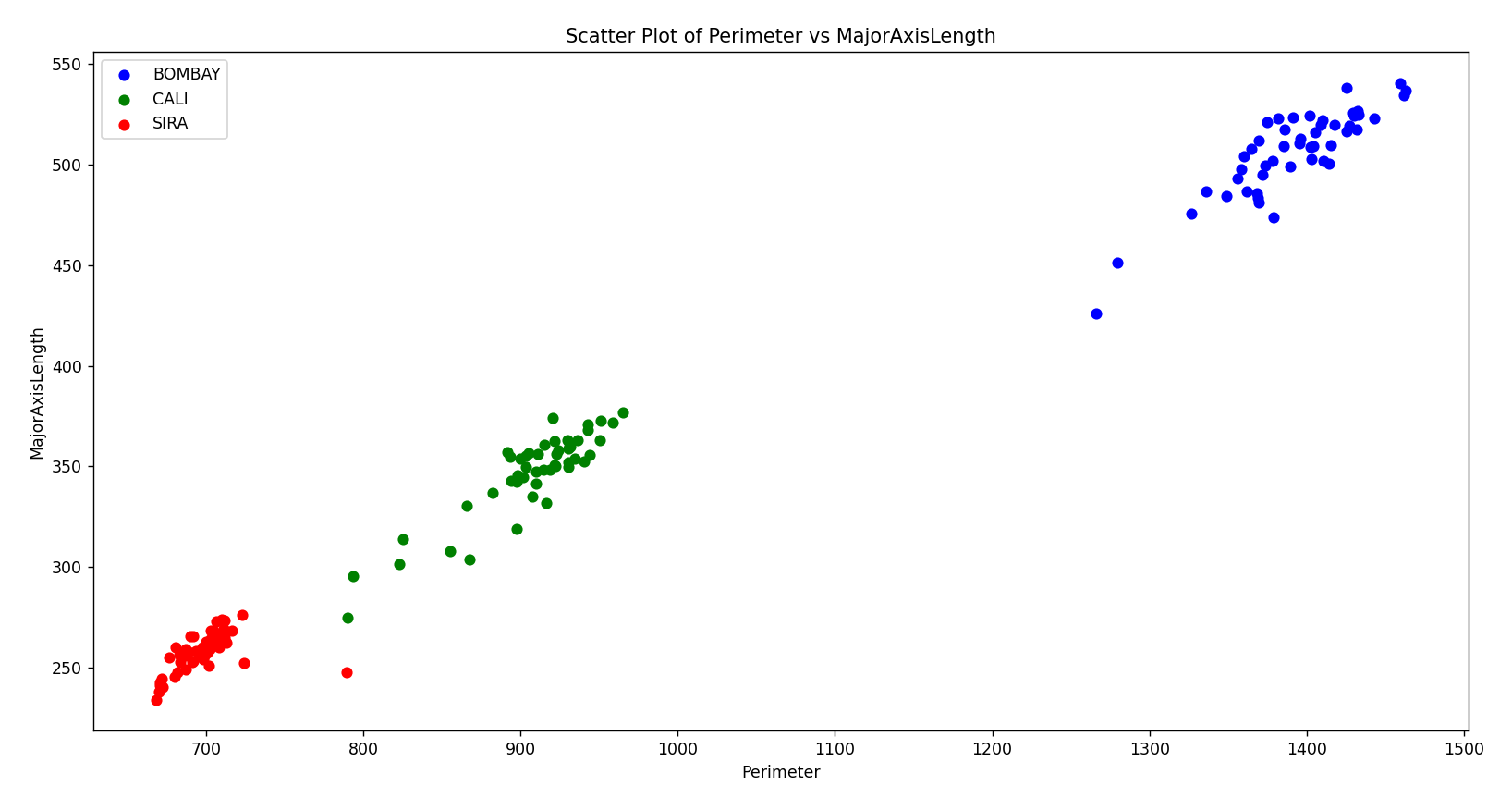
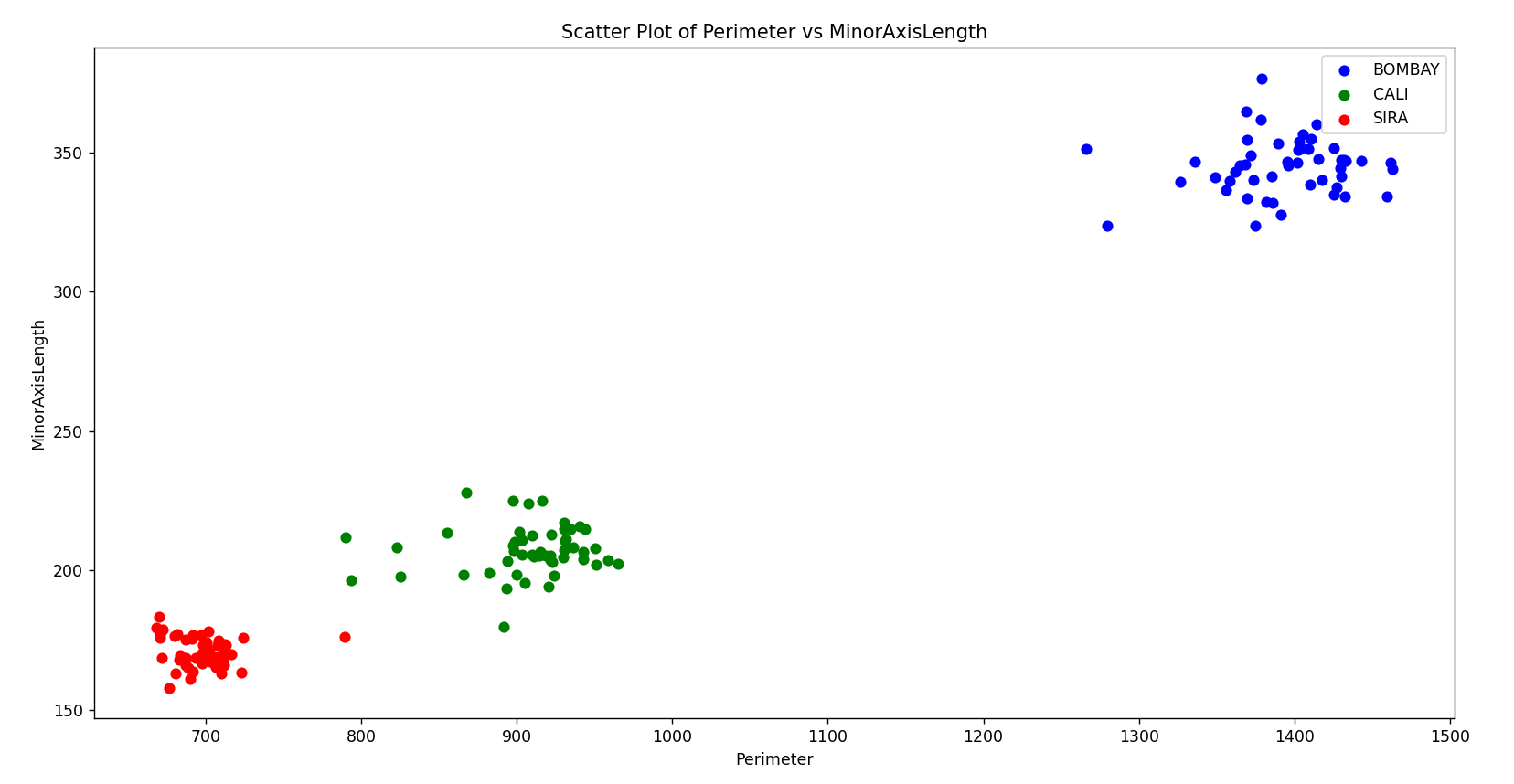


1. Area , MinorAxisLengthA screenshot of a computer screen

   Description automatically generated
2. Area, roundness

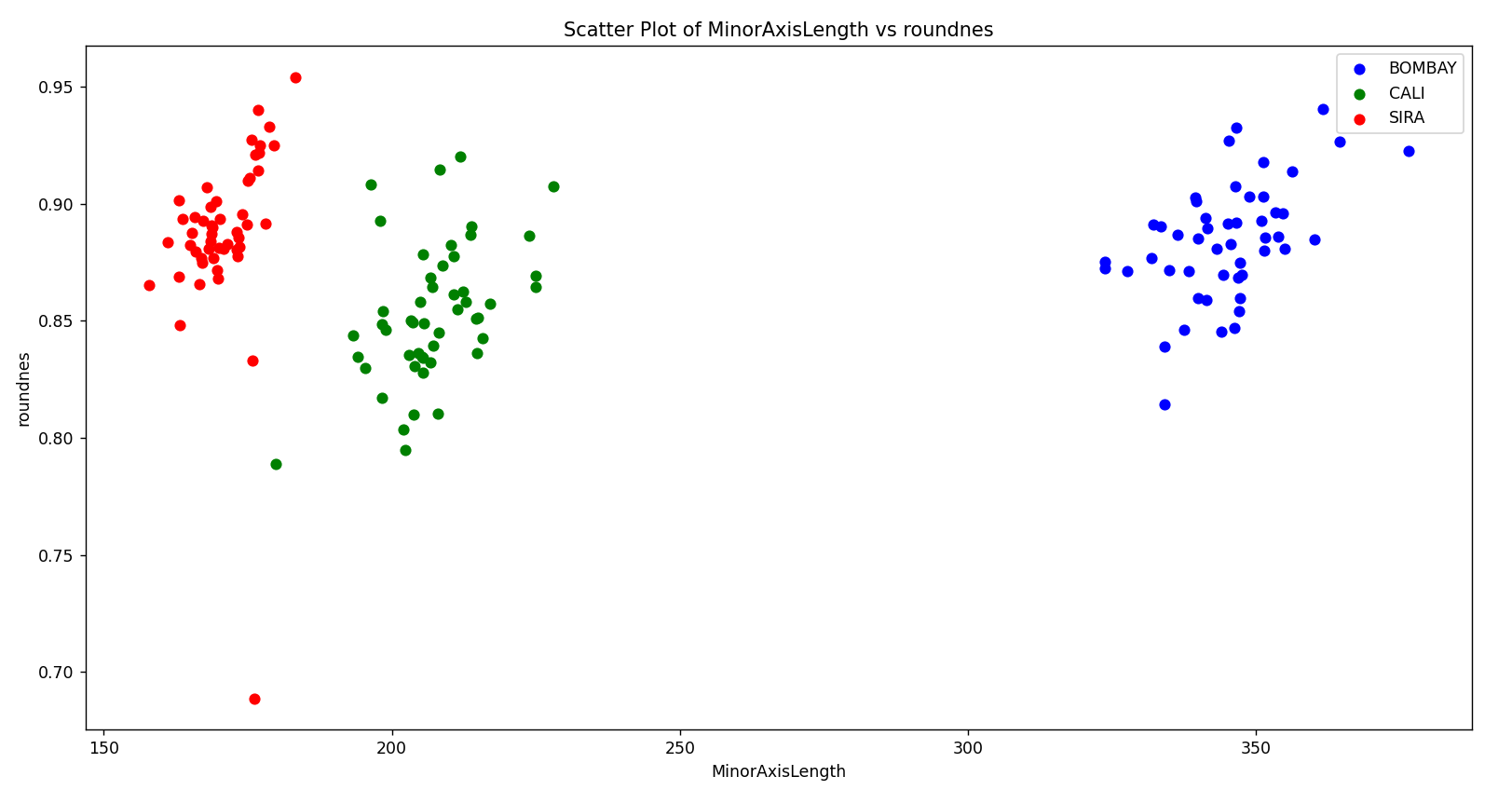
A screen shot of a computer

Description automatically generated

1. Perimeter, MajorAxisLength
2. Perimeter, MinorAxisLength
3. Perimeter, roundnesA screen shot of a computer screen

   Description automatically generated
4. MajorAxis Length , MinorAxisLengthA screen shot of a graph

   Description automatically generated
5. A screen shot of a graph

   Description automatically generatedMajorAxisLength, roundnes
6. MinorAxisLength, roundnes

Model results

* Adaline model
* Area, perimeter, and classes (BOMBAY ,CALI)

Must Use Bias to Separate Classes Linearly.A screenshot of a computer

Description automatically generated

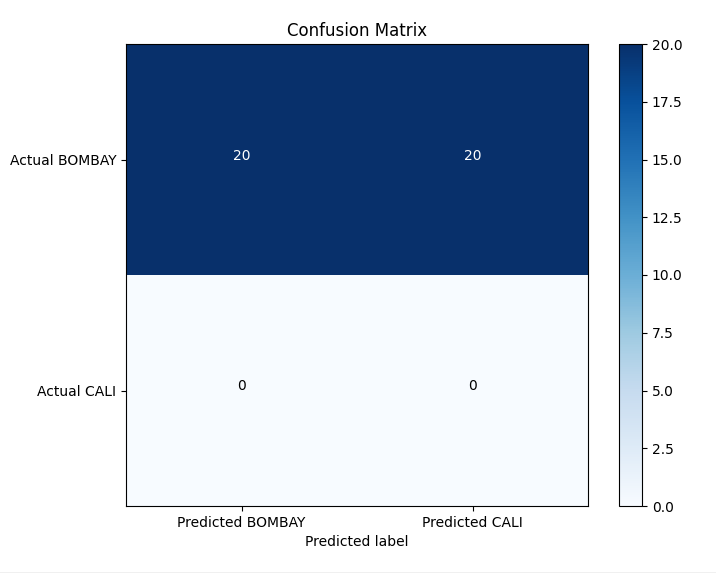
Adaline Classes with accuracy 100% with Bias

A diagram of a confusion matrix

Description automatically generated

**Without Using Bias** **and Give Accuracy 50%**

A screenshot of a computer

Description automatically generated

* Area, perimeter, and classes (CALI , SIRA)

A screenshot of a computer

Description automatically generated

* MajorAxisLength, ManiorAxisLength and classes (CALI , SIRA)

**Accuracy 100% with bias**

**A screenshot of a computer

Description automatically generated**

**A screenshot of a computer

Description automatically generated**

**Without Bias:**

A screenshot of a computer

Description automatically generated

* roundnes, Area

**Without using Bias is Linearly Separable with classes ( CALI , SIRA ) and (SIRA , BOMBAY ) With Accuracy 100%**

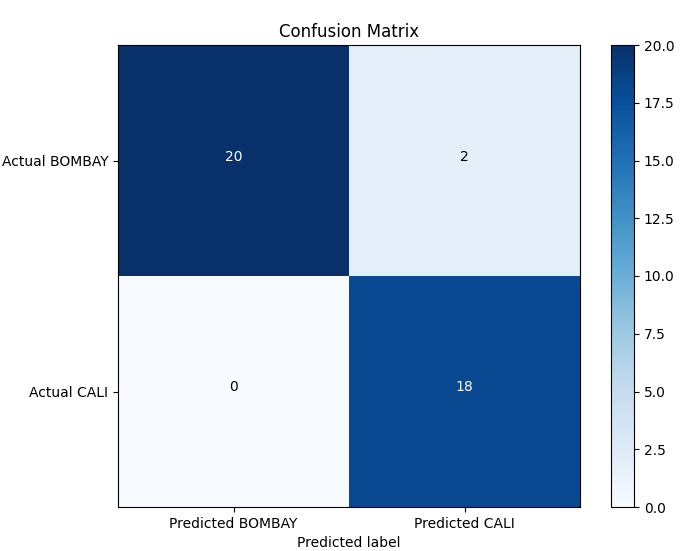
**A screenshot of a computer

Description automatically generated**

**classes (BOMBAY , CALI) Give Accuracy 90%**

**A screenshot of a computer

Description automatically generated**

****

* perceptron model
* Area , perimeter, and classes (BOMBAY ,CALI)

**With Using Bias is Linearly Separatable Classes with Accuracy 100%**

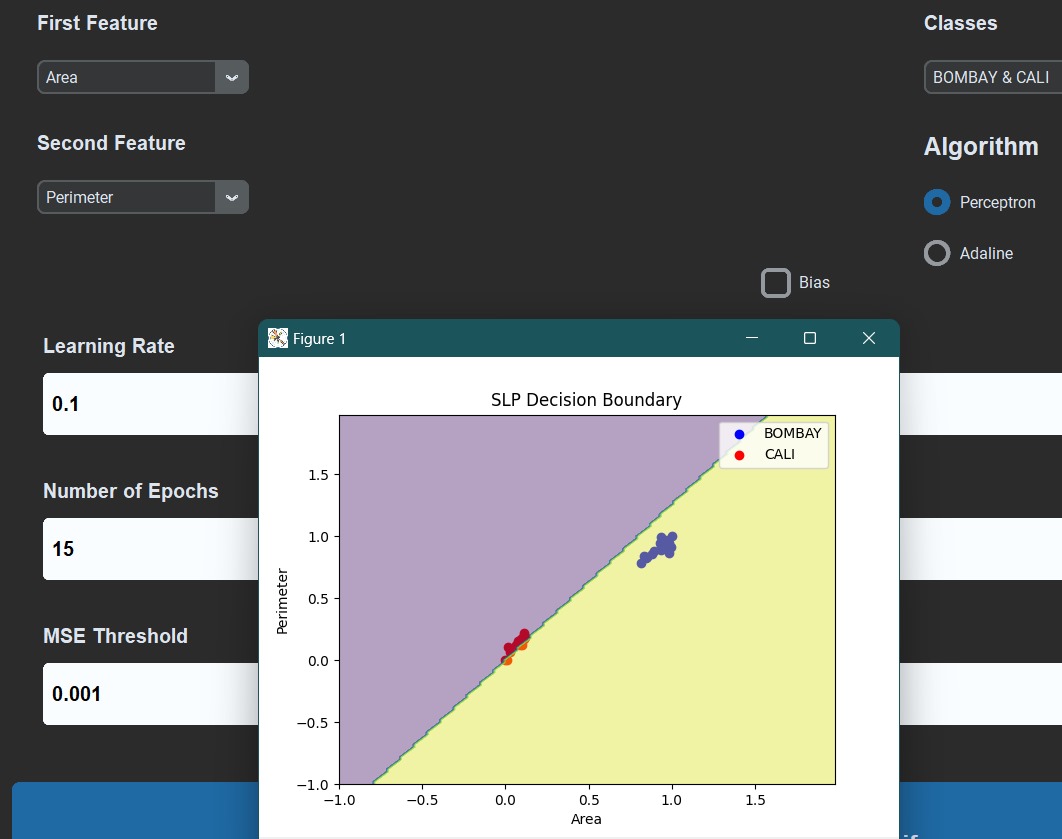
A screenshot of a computer

Description automatically generated

A diagram of a calibration matrix

Description automatically generated

**Without using bias**



* MajorAxisLength, ManiorAxisLength and classes (CALI , SIRA)

Same as area , perimeter. bias must be used to separate classes.

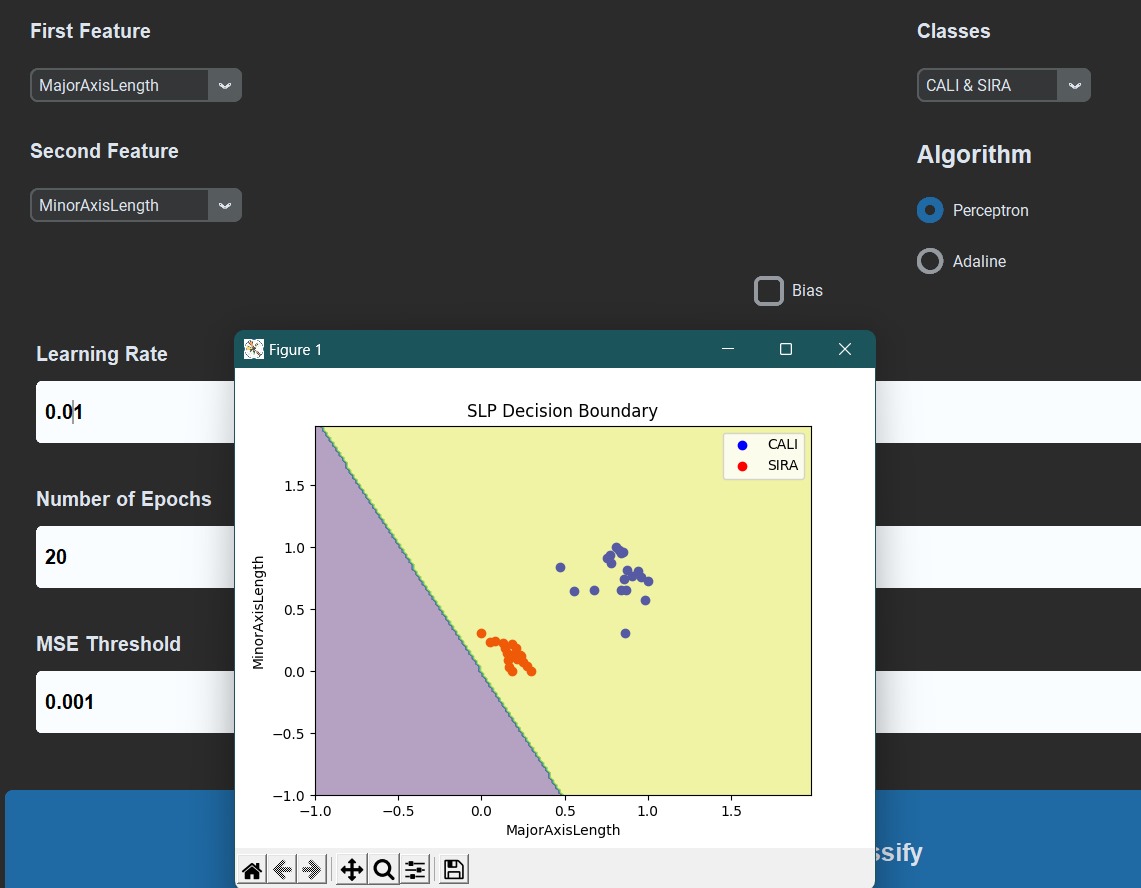
A screenshot of a computer

Description automatically generated

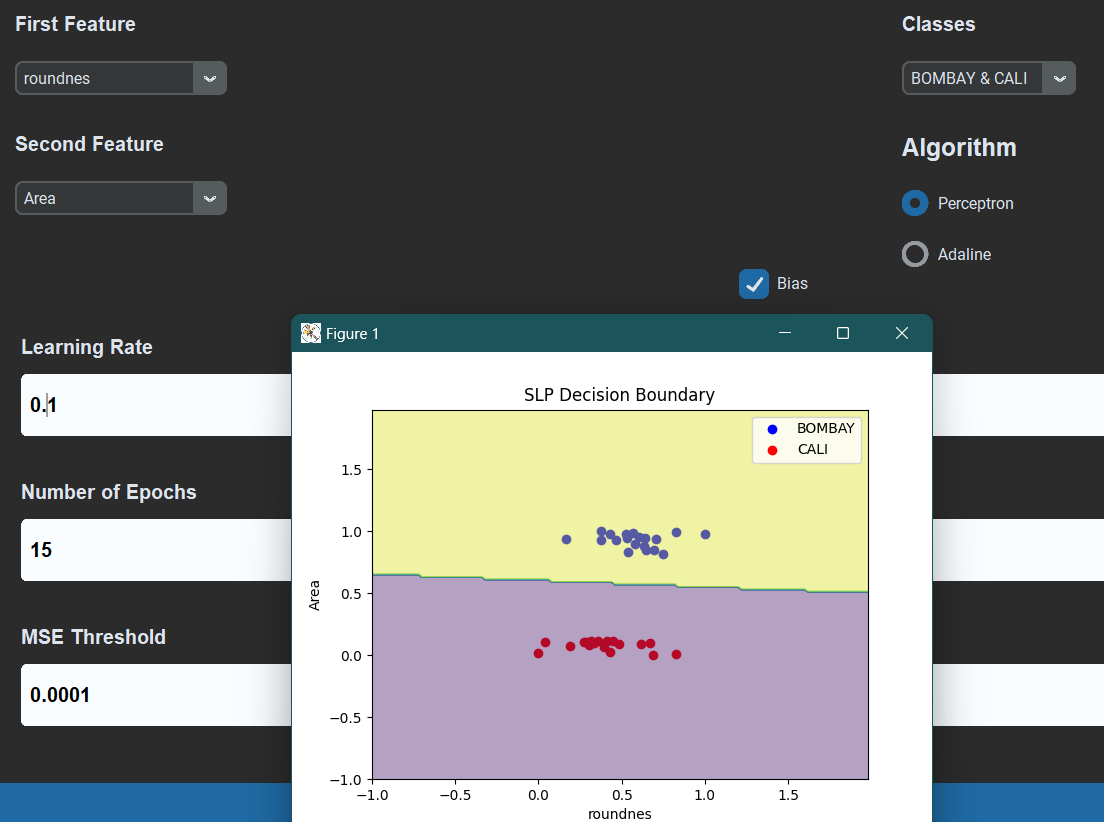
A diagram of a diagram

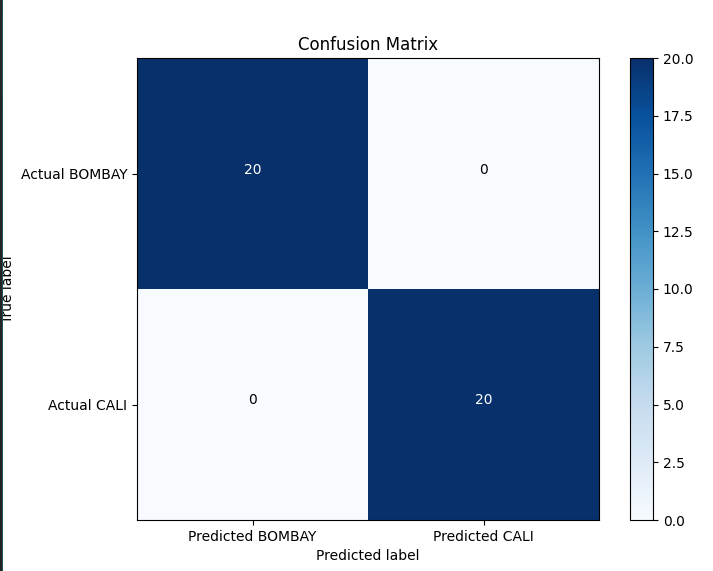
Description automatically generated with medium confidence

Without bias with Accuracy 0%



* roundnes, Area and classes (BOMBAY , CALI)





Conclusion

* All classes are linearly separable when using suitable features.
* Features: Area and MinorAxisLength can give very high accuracy as they tend to cluster data points for each class separately
* Bigger learning rates can be used with the single layer perceptron due to its discreate nature allowing faster convergence.
* Adaline works better with smaller learning rate as it works by minimizing continuous values.
* 100% accuracy is achievable for any pair of the provided classes given correct hyperparameters.
* Using bias usually gives better accuracy and faster convergence for Gradient descent.
* Some features cannot be separated with a proper decision boundary without using bias.