Neural Network Task1 Analysis Report

**Team Number: 79**

|  |  |
| --- | --- |
| ID | Name |
| 20201701002 | يوسف إسحق وصفي واصف |
| 20201701077 | ربا وصفي الشماط |
| 20201701234 | ساندى يوثام نزهى نصيف |
| 20201701223 | محمد فوزي ابراهيم |
| 20201701269 | يوسف وليد محمد إسماعيل |
| 20201701078 | رفيق نصر توفيق حنا |

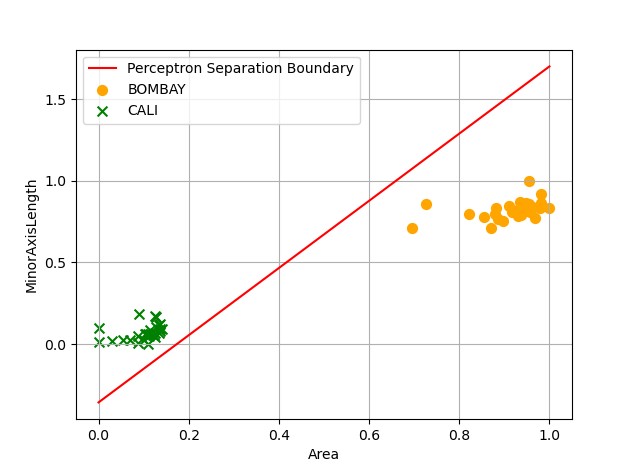
# **Perceptron:**

With Learning rate: 0.02, Epochs number: 300

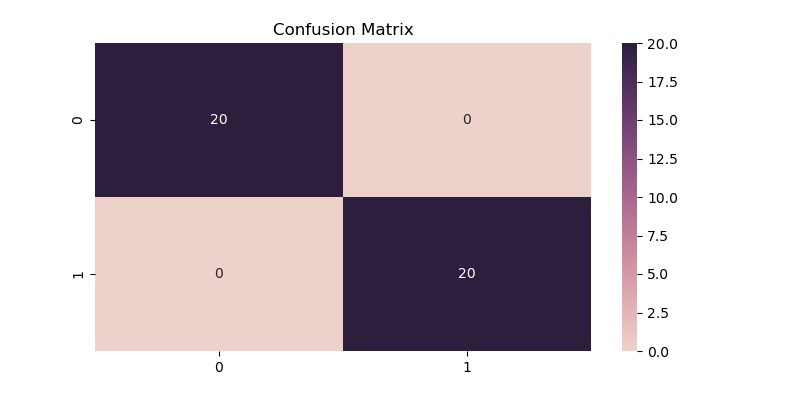
First combination:

Feature one: Area, Feature two: MinorAxisLength

Class one: Bombay, Class two: Cali



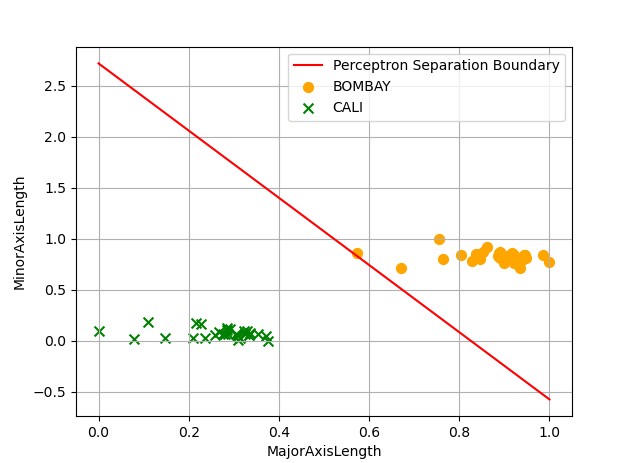
With Bias

Training accuracy and Testing accuracy=100%

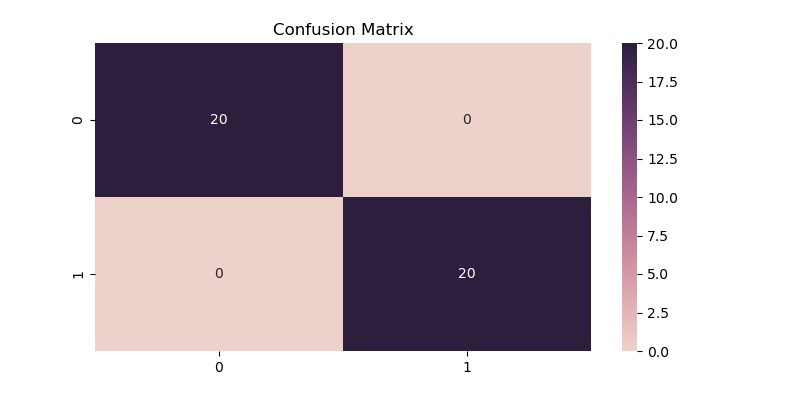
Second combination:

Feature one: MajorAxisLength, Feature two: MinorAxisLength

Class one: Bombay, Class two: Cali



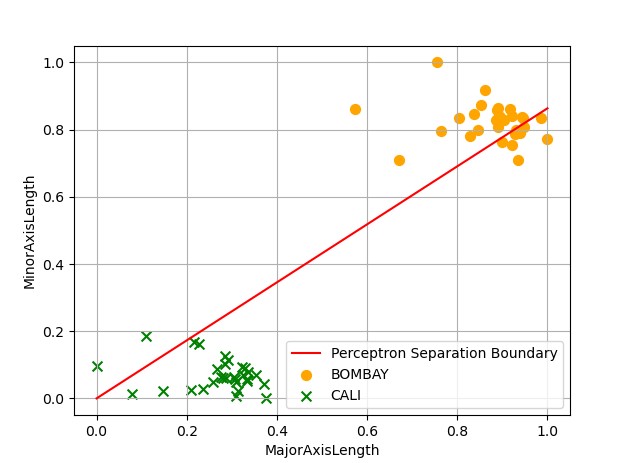
With Bias



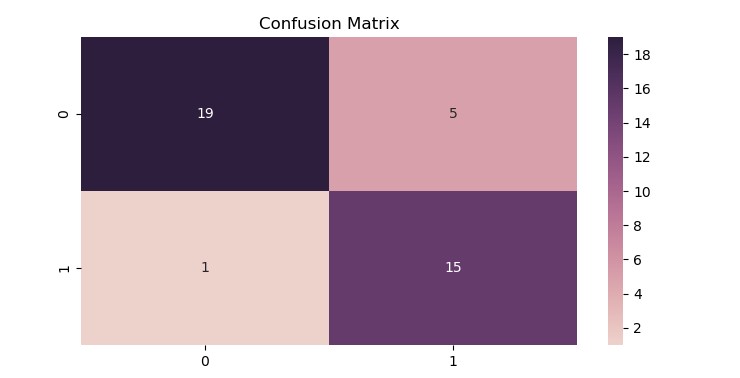
Testing Confusion Matrix

Training accuracy and Testing accuracy=100%

* With Bias we found line that discriminates data perfectly.



Without Bias

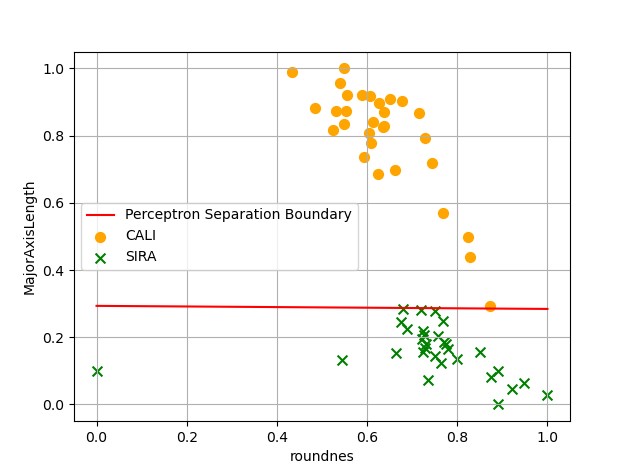


Training accuracy = 81.7% and Testing accuracy = 85%

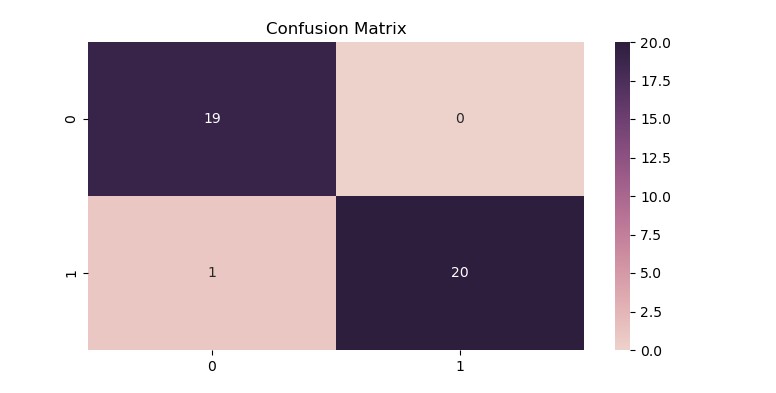
* This indicates that bias helps algorithms to classify better.
* If there is a line that classify the data perfectly not adding a bias enforce it to use a line that passes through origin despite its accuracy.

Third combination:

Feature one: Roundnes, Feature two: MajorAxisLength

Class one: Cali, Class two: Sira 

With Bias



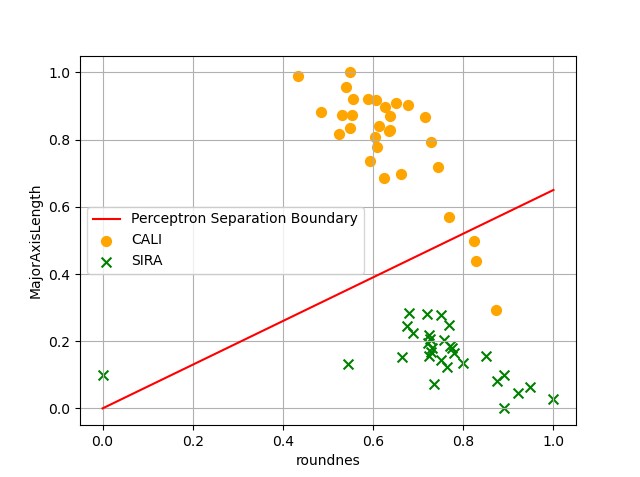
Training accuracy = 100% and Testing accuracy = 97.5%

* The difference in accuracies may indicate that there is an overfitting.

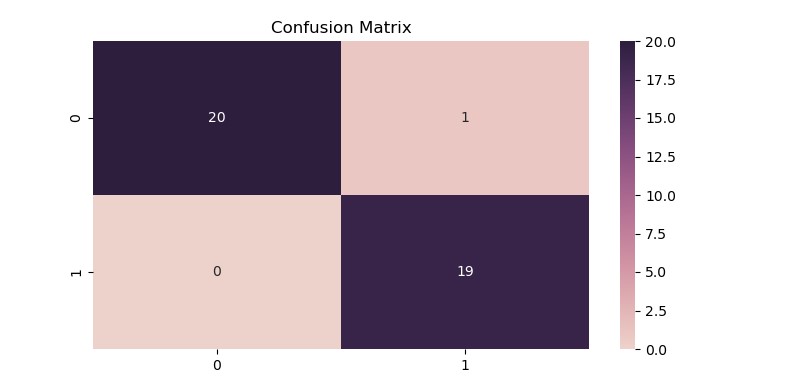
Fourth combination:

Feature one: Roundnes, Feature two: MajorAxisLength

Class one: Cali, Class two: Sira



Without Bias



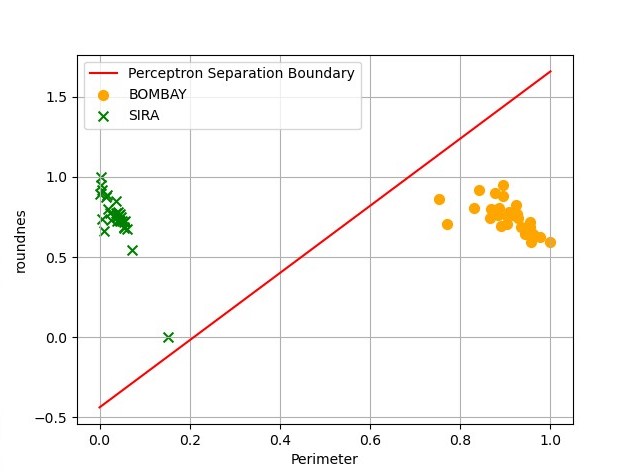
Training accuracy = 93.3% and Testing accuracy = 97.5%

* Here removing bias also caused the accuracy to drop.

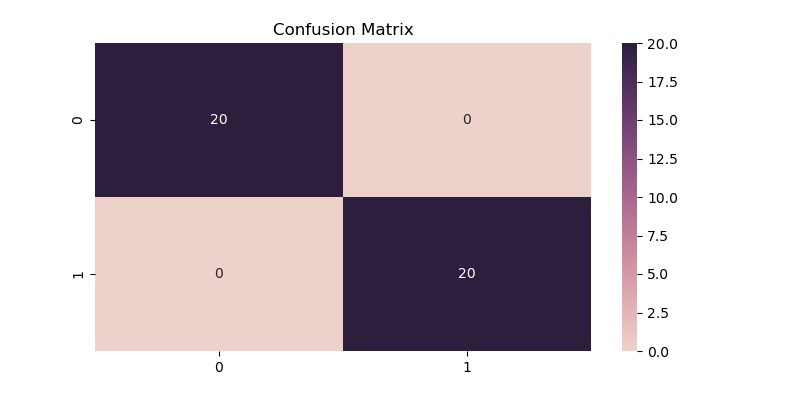
Fifth combination:

Feature one: Perimeter, Feature two: Roundnes

Class one: Bombay, Class two: Sira



With Bias



Training accuracy and Testing accuracy=100%

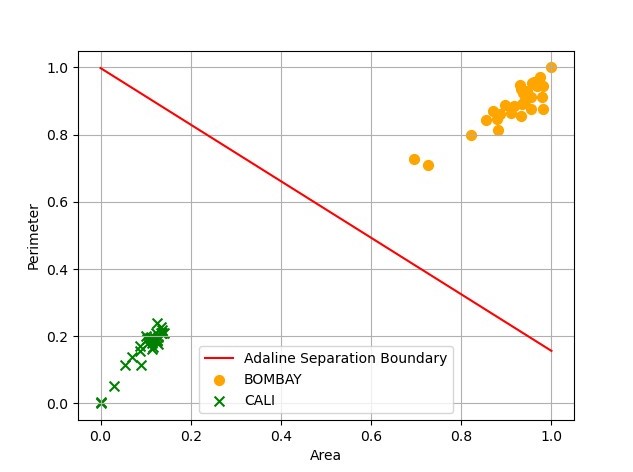
**Adaline:**

Learning rate: 0.02, Epochs number: 300, MSE: 0

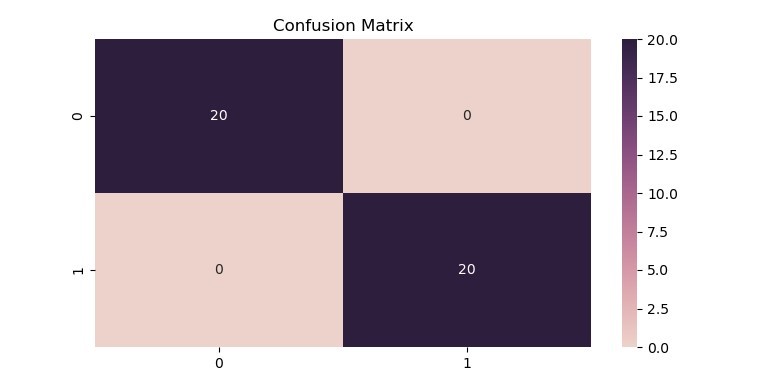
First combination:

Feature one: Area, Feature two: Perimeter

Class one: Bombay, Class two: Cali



With Bias

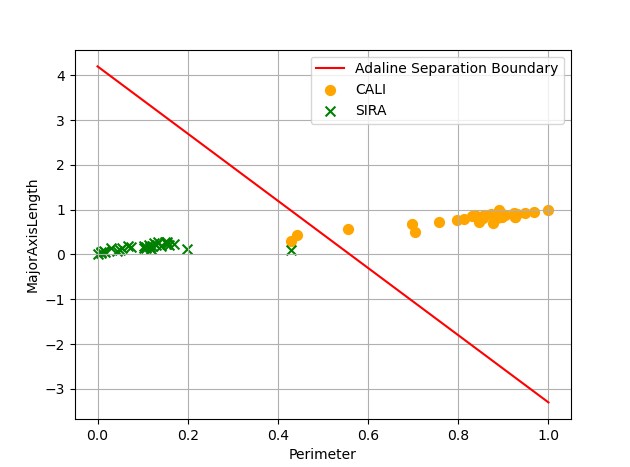


Training accuracy and Testing accuracy=100%

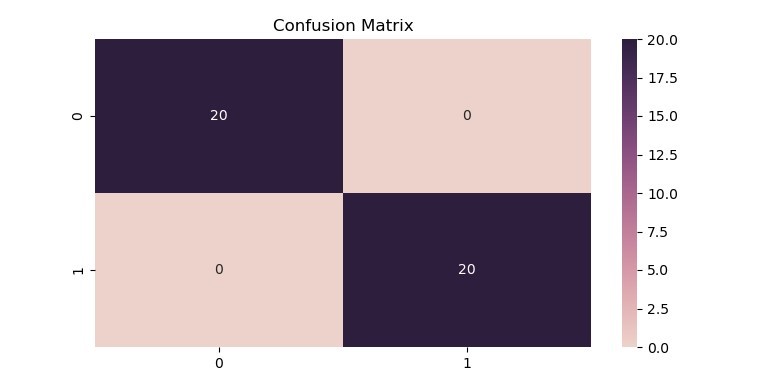
Second combination:

Feature one: perimeter, Feature two: MajorAxisLength

Class one: Sira, Class two: Cali



With Bias



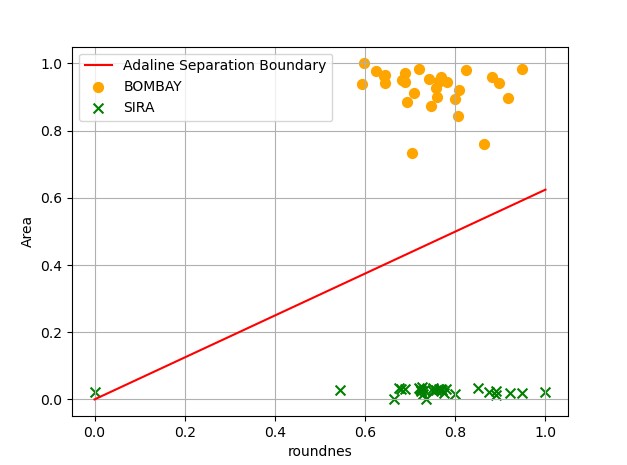
Training accuracy = 96.7% and Testing accuracy = 100%

* The difference in accuracies may indicate that there is a generalization.

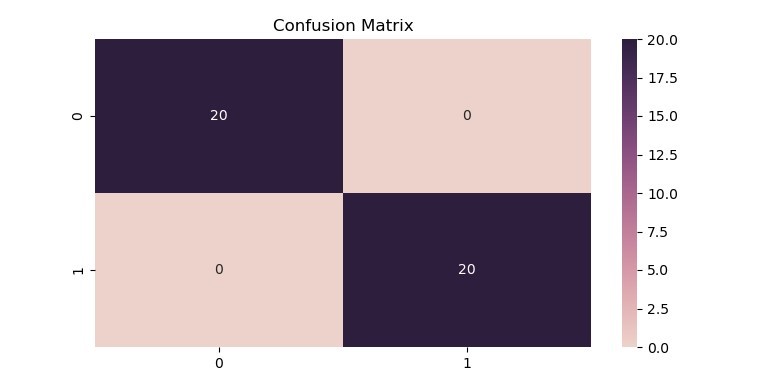
Third combination:

Feature one: Roundnes, Feature two: Area

Class one: Bombay, Class two: Sira



Without Bias



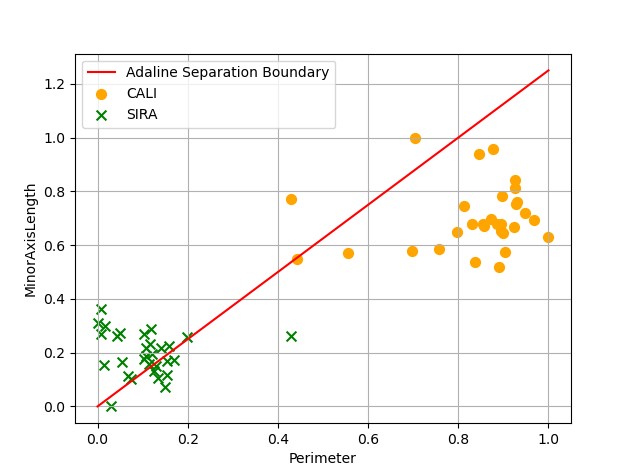
Training accuracy =98.3% and Testing accuracy=100%

* The difference in accuracies may indicate that there is a generalization.

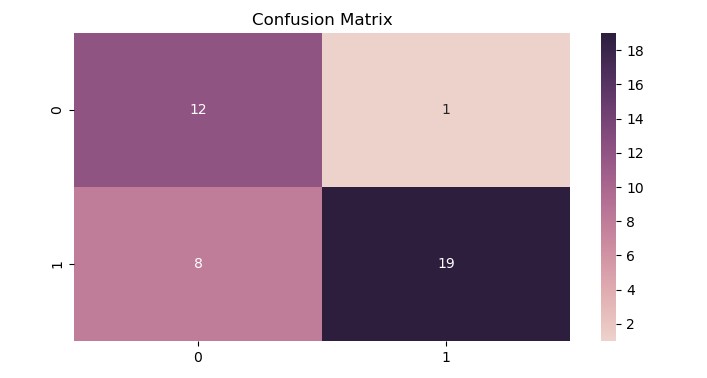
Fourth combination:

Feature one: Perimeter, Feature two: MinorAxisLength

Class one: Cali, Class two: Sira

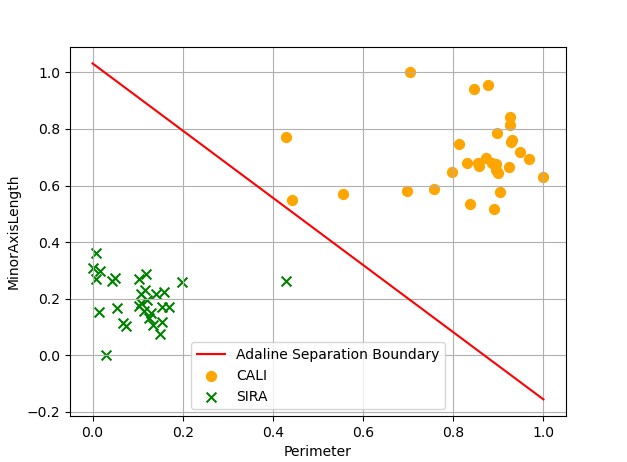


Without Bias

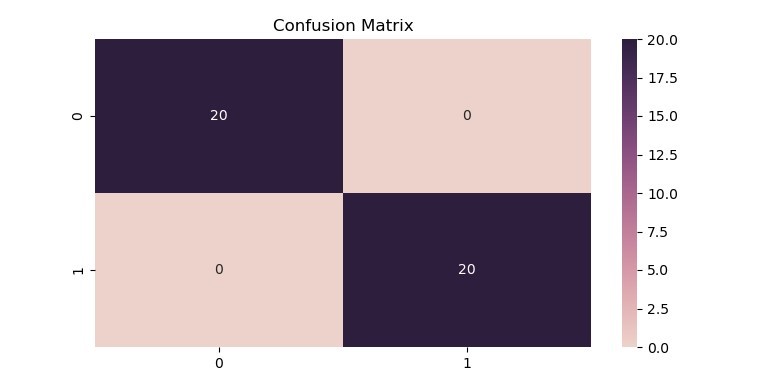


Training accuracy = 81.7% and Testing accuracy = 77.5%

* This indicates bias is important in improving accuracies.



With Bias

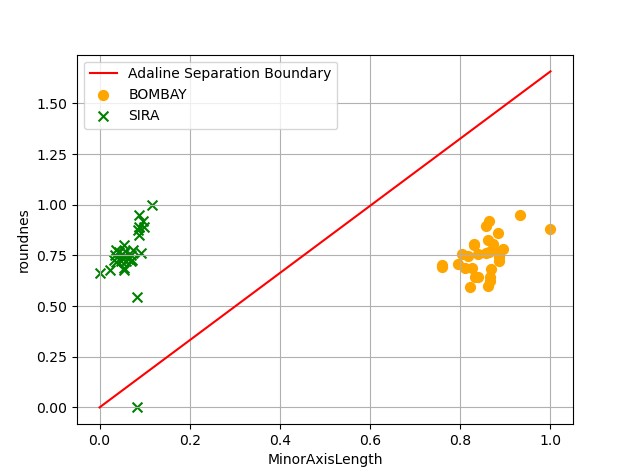


Training accuracy and Testing accuracy=100%

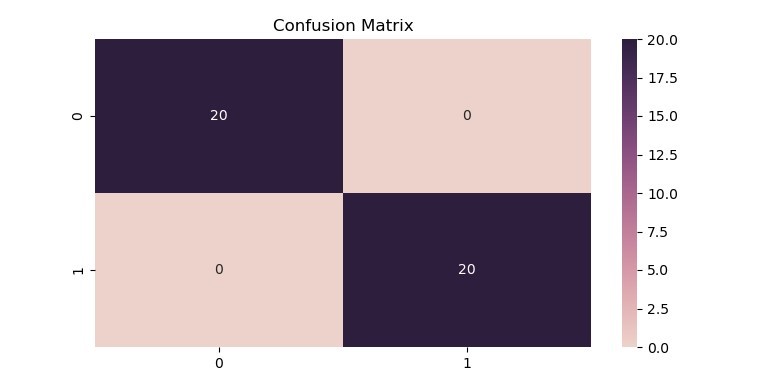
Fifth combination:

Feature one: MinorAxisLength, Feature two: Roundnes

Class one: Bombay, Class two: Sira



Without Bias



Training accuracy = 98.3% and Testing accuracy=100%

In conclusion, the best combinations of features for Perceptron are Area & MinorAxisLength or MajorAxisLength & MinorAxisLength or Perimeter & Roundness ,and the best combinations of features for Adaline are Perimeter & MinorAixsLength or Area & Perimeter. We have to use bias with all of them as well.