LOTI.05.046 PATTERN RECOGNITION

HOMEWORK 2 - DECISION STUMP

March 18, 2019

Quazi Saimoon Islam
Msc. Robotics and Computer Engineering

1 INTRODUCTION

Decision stump is a machine learning model that consists of a single layered decision tree. A prediction is made based on the value of a single input feature. Thus decision stumps are known as weak classifier. A more sophisticated classifier is the decision tree. In direct comparison, a decision stump is a tree with a single node, i.e. the root only.

2 VISUALIZING THE DATA

The given data contains 150 data samples on 4 features, with 3 unique labels. The given labels are the three variations of the flower Iris; namely Iris Setosa, Iris Virginica and Iris Versicolor. The features included are the Petal Width, Petal Length, Sepal Width and Sepal Length. The unit for the dataset is not specified.

To effectively visualize the data, the data given was read into a list of rows in python. This list was then used to create arrays of the different features and a combined array of all the features was also extracted to generate the plots in the following section.

2.1 Combined Scatter Plot of all Features

The first plot visualized was the overall plot of all the data points taking into account all the features. The scatter plot was used to perform the visualization and gain a better understanding about the distribution of the data.

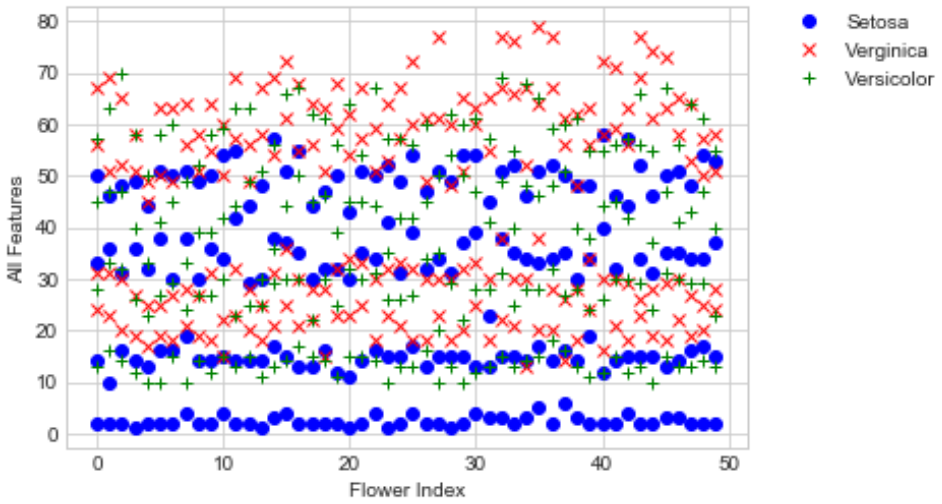

Figure 1: Scatter Plot of all Features

The above plot was generated by looping through each of the data points and labeling the points accordingly. The plot makes it quite hard to easily discriminate between the different data points. There are multiple data points where a lot of the features are bundled together and are overlapping in some cases as well. In terms of generating a good decision stump, this plot does not prove handy enough. However, the there are certain patterns that become apparent even in this plot, in particular, the fact that some data points are quite consistent over the whole flower index. This will become more apparent on the Feature plots in the next section.

2.2 Scatter Plot Based on each Feature

The next step was to visualize the data using individual features as data points. This was done by separating the features corresponding to individual classes (types of flower in this case). The subplot feature of the pyplot library was then used to make 4 plots where the Y-axes were the different features, and the X-axis was once again reserved for the flowr index.

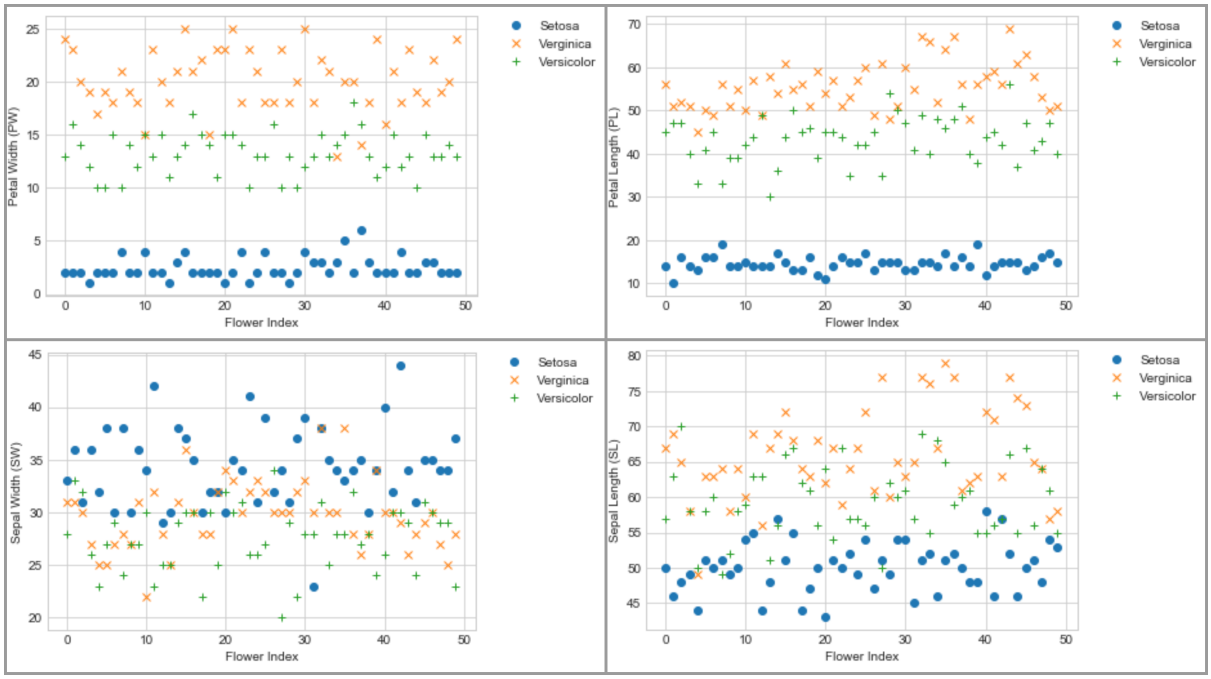

Figure 2: Scatter Plots of the data based on features

The above plots clarify a lot about the individual features themselves. In particular, how some features present some opportunities of linearly partitioning the data, giving rise to the selected values for the thresholds. The next section makes use of these plots to develop the Decision stump for the individual features.

3 Decision Stump

To understand better about patterns in features and their relative accuracy in predicting labels, 4 decision stumps were implemented and subsequently their accuracy in predicting the right labels tested to gain a better understanding of whether it is possible to segregate good features from poorer once. Once again, decision stumps are very poor classifiers, so the accuracies in this case do not mean much, but do provide a simple guideline to understanding the nature of features.

3.1 Decision Stump for the Petal Width Feature

For the petal width feature, the data is pretty well partitioned as is apparent from the scatter plot. Using this plot, it is possible to linearly separate the data along the horizontal plane using threshold values of 8 and 16 to predict the output labels.

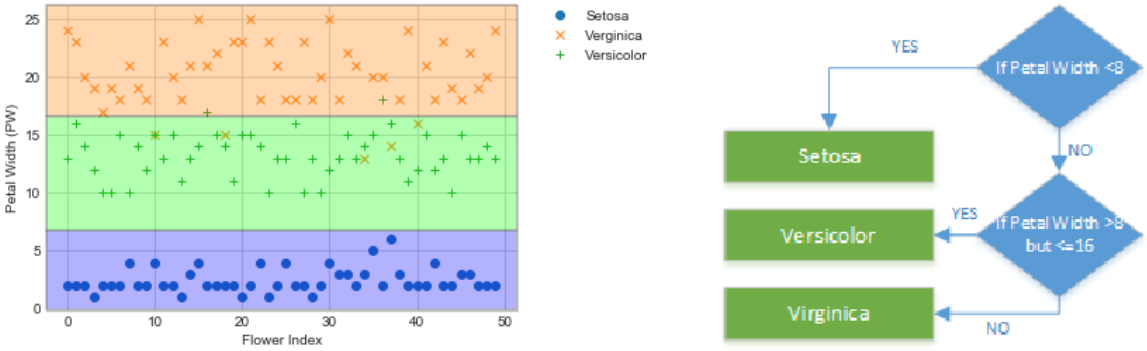

Figure 3: Data Partitioning and Decision Stump Flowchart for PW

3.2 Decision Stump for the Petal Length Feature

For the petal width feature, the data is also pretty well partitioned as is apparent from the scatter plot. Using this plot, it is possible to linearly separate the data along the horizontal plane using threshold values of 25 and 50 to predict the output labels. There a few datapoints here that are clustered together so the expected accuracy should be a bit low with this feature.

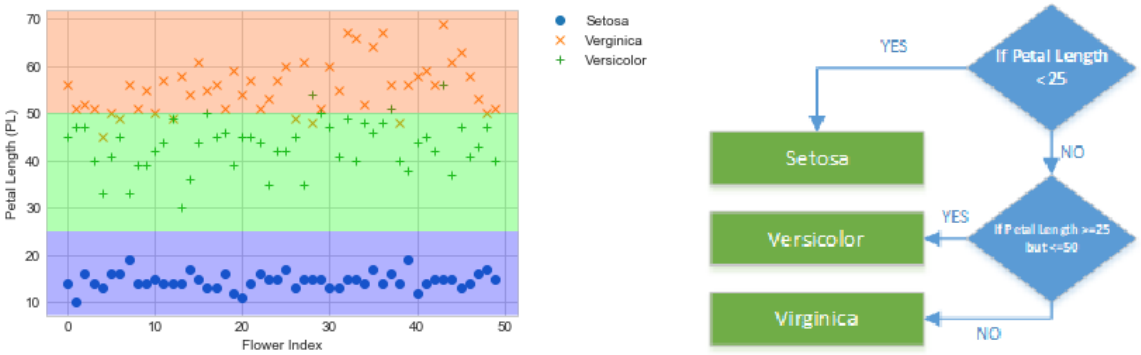

Figure 4: Data Partitioning and Decision Stump Flowchart for PL

3.3 Decision Stump for the Sepal Width Feature

On the contrary to the above, the Sepal Width feature was found to be very difficult to accurately partition. This was because a lot of the data has high variance. This means that it is difficult to draw a linear line to separate the classes accurately. Based on the plot, the best partitioning that could be done was at threshold values of 28 and 33 to predict the output labels. Just from the visualization of the data, it is quite apparent that the decision stump will not be able to provide a very accurate model to classify the given data.

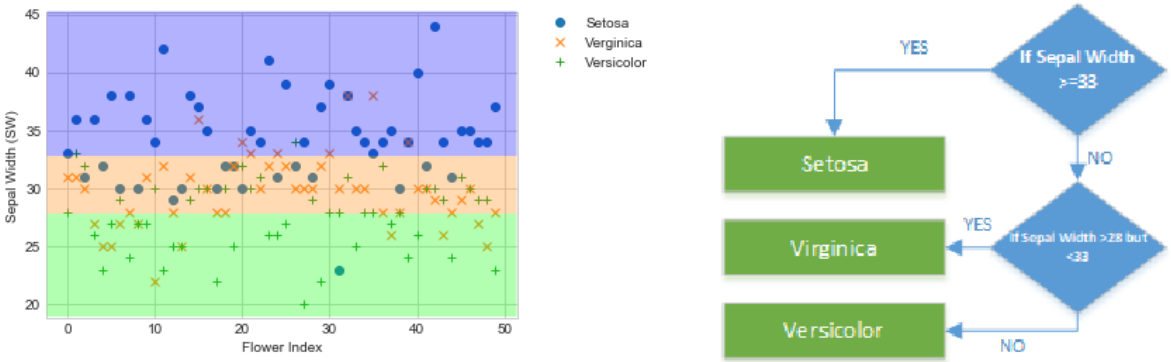

Figure 5: Data Partitioning and Decision Stump Flowchart for PL

3.4 Decision Stump for the Sepal Length Feature

The final decision stump developed was for the sepal length feature. Compared to the sepal width data, it is slightly easier and more intuitive to partition the data. However, the partitioning was still not as smooth as was for the Petal width and Petal length features. Based

on the plot, the best partitioning that could be achieved was by using the threshold values of 54 and 63.

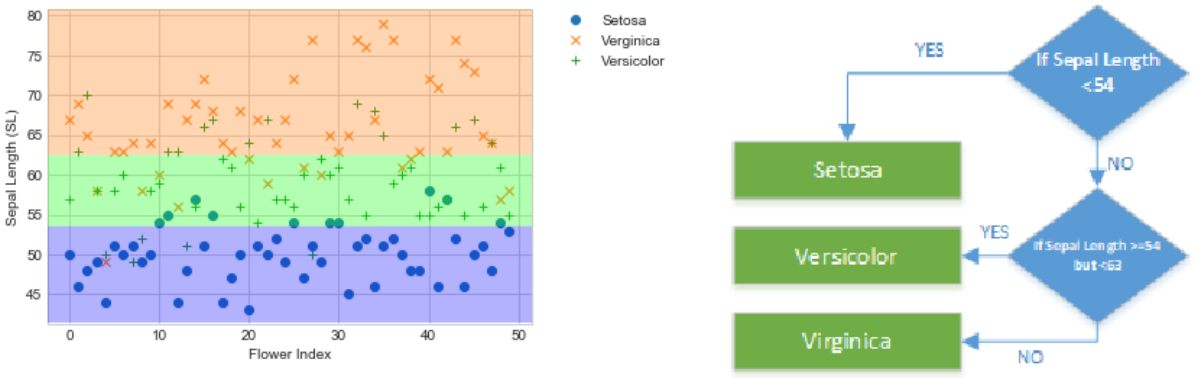

Figure 6: Data Partitioning and Decision Stump Flowchart for PL

4 Evaluating the Decision Stump

In order to evaluate the performance of the Decision Stumps described above, a simple binary test was used. In short, once the predictions are compiled into a vector, it was compared with the labels assigned in the original text file (column 1 of the text file). If the values matched, a 1 was appended to an accuracy array, and if the values did not match, then a 0 was appended. Finally, the mean of this accuracy array was then computed to find the overall accuracy of prediction of the classifier. The following image is the output of the accuracies of the decision stumps computed using this method.

```
Accuracy of Decision stump using Petal Width: 0.953333
Accuracy of Decision stump using Petal Length: 0.920000
Accuracy of Decision stump using Sepal Width: 0.560000
Accuracy of Decision stump using Sepal Length: 0.720000
```

Figure 7: Accuracies of the Decision Stump on different features

From these accuracy values, it can be said that the Petal Width and Petal Lengths are the strongest features. This also agrees with the visualization from the scatter plots and the thresholding values used. Most of the data were discriminate and thusly provided the most useful information towards predicting the right values. The Sepal Length seems to be

a moderately acceptable feature in this case as it provides an accuracy of 72%. The Sepal Width indeed seems to be the worst feature of the batch which is understandable also from the visual representation of the data in the scatter plot.

5 CONCLUSION

From the above, one can conclude that the use of decision stump is fairly limiting towards making accurate predictions, but when used in combination as done, it can shed some light into the validity and strength of the feature. This technique may become cumbersome for larger dimensions of data and thus, is only feasible for lower numbers of features as present in this example.

REFERENCES

[1] Tom M. Mitchell. *Machine Learning.* (Published) 2017.

[2] Steven Cooper. *Machine Learning for Beginers.* (Published) 2018.

[3] Matplotlib. *Pyplot Tutorials* [online]. Available at: <https://matplotlib.org/tutorials/introductory/pyplot.html>
[Accessed 10 March 2019].