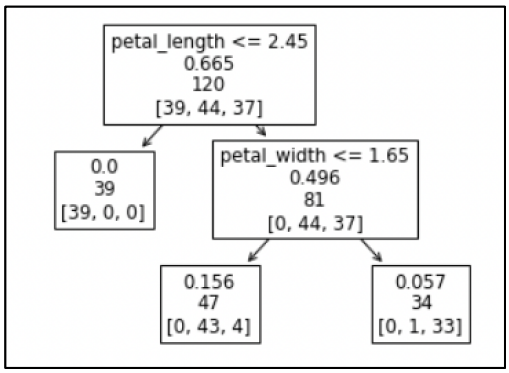
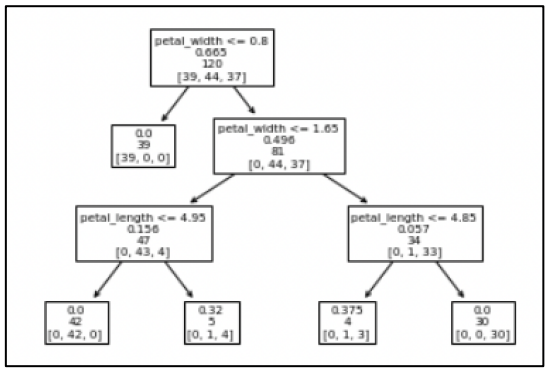
Homework 2 – Programming

Support Vector Machine (SVM) and Decision Tree classifier

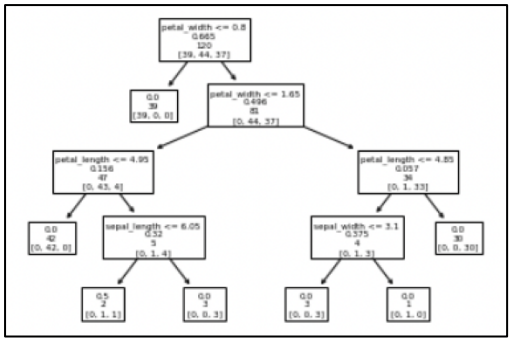
1.



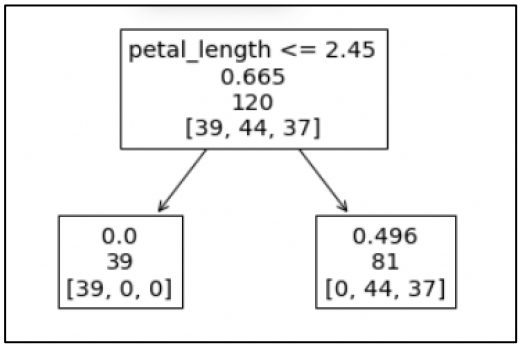
Depth = 2, Acc = 96.7%



Depth = 3, Acc = 96.7%



Depth = 4, Acc = 96.7%

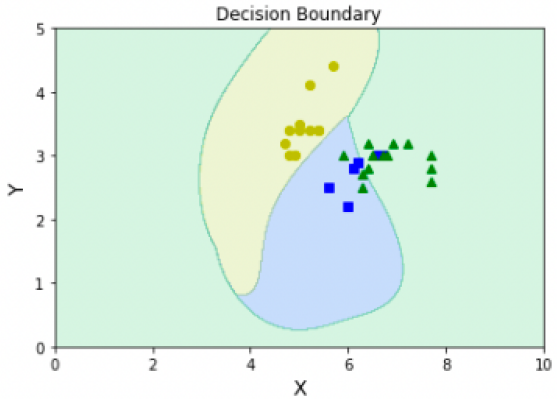


Depth = 1, Acc = 56.7%

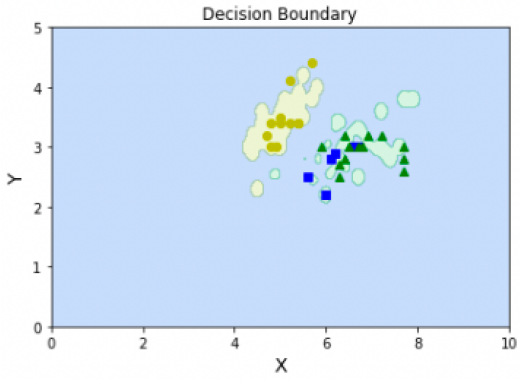
<Comment>

Based on the training results of decision tree classifier shown above, we can see that there is almost no difference in terms of accuracy with **depth** parameter from 2 to 4. To find out the reason more clearly, I test depth parameter as 1 additionally, and obtain the accuracy with 56.7%. From the tree diagram with depth = 1 fisrt, we can observe that there is only one condition (petal_length <= 2.45) to split the data into two classes. However, in this homework, we totally have three classes, so it's indeed to get poor performance (56.7%). Next, focus on depth parameter from 2 to 4, the conditions to split the data is increasing, which is enough to split the data into three classes. Hence, **I think that the larger depth parameter, the more classes can be splitted, thus to get better accuracy.** But in this case, we only have three classes. It is sufficient to split the data effectively with depth = 2.

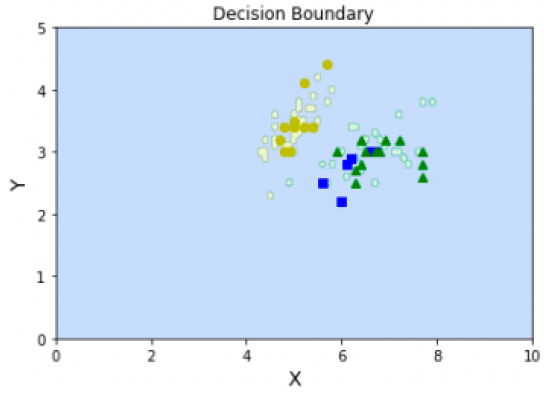
2. & 3.



Gamma = 1, Acc = 96.7%



Gamma = 100, Acc = 36.7%

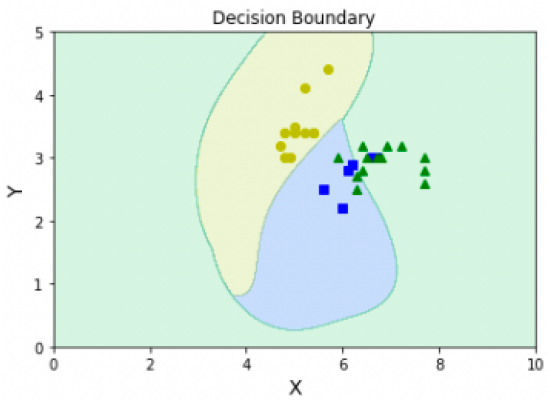


Gamma = 500, Acc = 20%

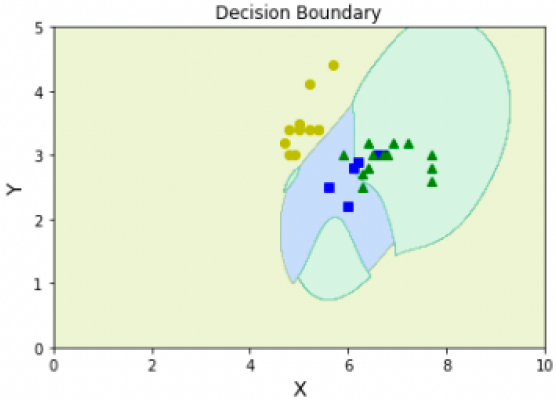
<Comment>

According to the training results of SVM classifier shown above, we can observe that there is a decreasing accuracy with larger **Gamma** parameter. From the plot of decision boundary, we can observe that the curvature is less smooth with larger Gamma parameter, causing **overfitting** situation. With [1], [2] as reference, it can be seen that **Gamma parameter defines how far the influences of a training example reaches**. It means that the larger Gamma parameter, the larger influences of the closer point in decision boundary, causing less smooth curvature. Basically, larger gamma parameter implies more complex model, but smaller gamma value might cause the model is too easy to decide a better behavior of testing model (underfitting situation). In this case, I think smaller gamma parameter have a better performance, since Gamma = 1 is sufficient to have better accuracy.

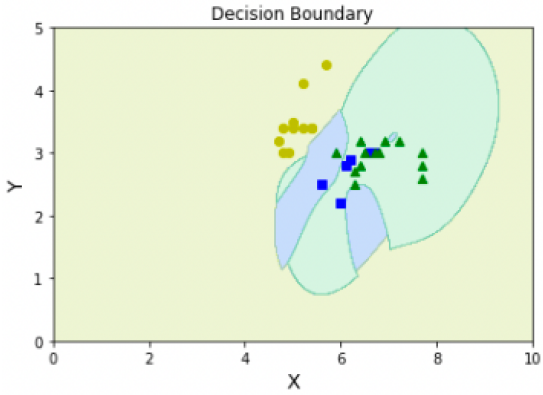
4. & 5.



C = 1, Acc = 96.7%



C = 100, Acc = 93.3%



C = 500, Acc = 93.3%

<Comment>

According to the training results of SVM classifier shown above, we can observe that there is some decreasing tread of accuracy with larger **C** parameter. From the plot of decision boundary, we can observe that the curvature is less smooth with larger C parameter. It can be speculated that there is an overfitting situation when the value of C parameter is larger. With [1] as reference, **the meaning of the C parameter is how much you want to avoid misclassifying each training example in SVM optimization.** For large values of C, the tolerance of SVM is smaller, having a tendency of overfitting. However, for small values of C, the tolerance of SVM is bigger, inclining to lead to underfitting situation. In this case, I think smaller C parameter have a better performance, since C = 1 is sufficient to have better accuracy.

6.

From the above results of decision tree classifier and support vector machine, if we tune the parameter well, nearly the same accuracy performs in two classifiers. The below shows the pros and cons between two classifiers:

	Pros	Cons
Decision Tree	● Normalization or scaling of data not needed. ● Model is very intuitive and easy to explain to technical teams (easy visualization).	● often involves higher time to train the model. ● is inadequate for applying regression and predicting continuous values.
SVM	● Performs well in Higher dimension. (not just 2D, 3D) ● Not biased by outliers	● For larger dataset, it requires a large amount of time to process. ● Selecting the appropriate kernel function can be tricky.

7. Reference:

[1]: https://rpubs.com/skydome20/R-Note14-SVM-SVR

[2]: https://www.youtube.com/watch?v=m2a2K4lprQw&ab_channel=Udacity