CMPE 462

Assignment 3

Mahir Efe KAYA – 2016400195

Description:

Implementing a decision tree for a dataset and a SVM model for another dataset.

PART I:

The dataset given to us is in csv format with headers and is about predicting class of iris from its' characteristics by implementing a Decision Tree.

Step I:

In our Decision Tree with information gain, it's been observed that 3rd and 4th columns are enough all by themselves. The Tree can successfully determine the class of the iris in the first depth(Fig. 1). To check if the program works correctly, I removed those columns from consideration and built the tree with only first two columns. It works with about %75 accuracy(Fig. 2) . Result:

DT petal-length 2.5999999999999996

Step II:

In our Decision Tree with information gain ratio, the same relation between the columns and the tree has been observed. The Tree can successfully determine the class of the iris in the first depth(Fig. 1). Result:

DT petal-length 2.5999999999999996

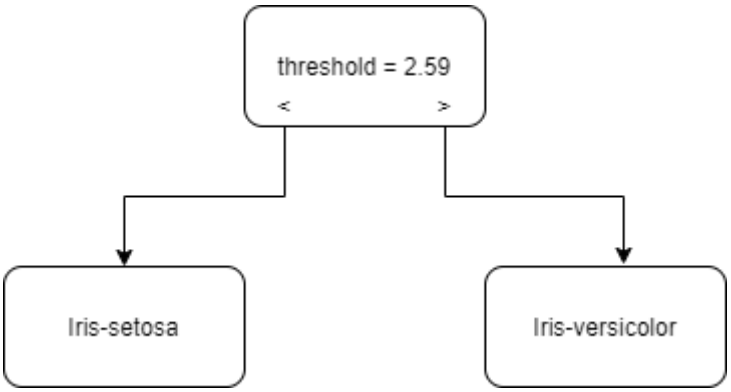

Fig. 1

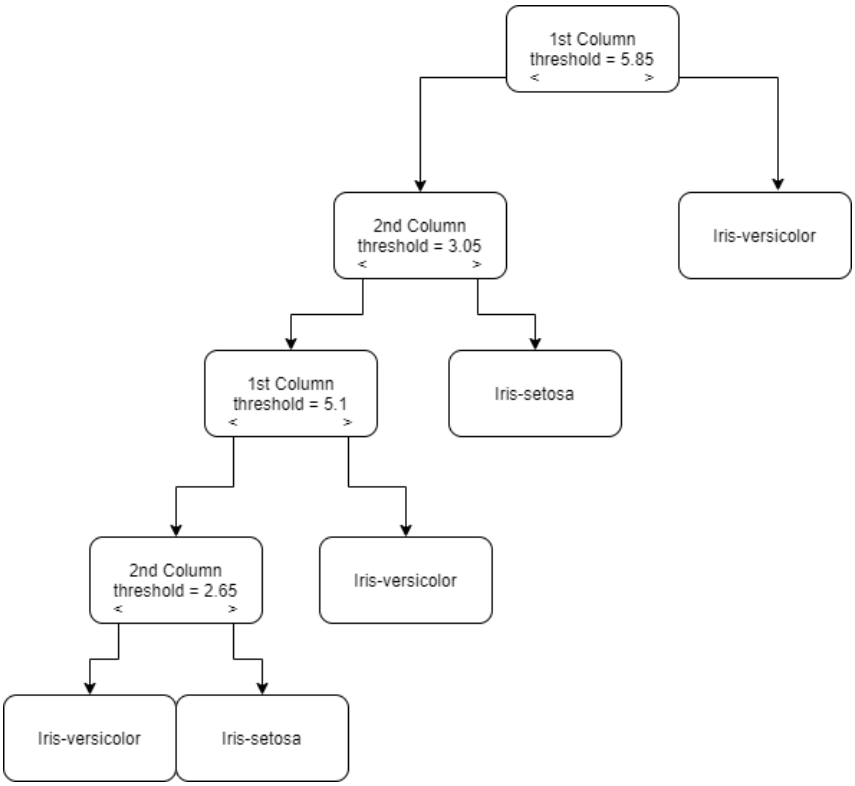

Fig. 2

PART II:

In this part, the dataset given to us in csv format with headers and is about predicting if the cancer exists from its' characteristics by implementing Support Vector Machine, with different C's and kernels. In this part, to remove the warnings and prevent program from reaching the maximum number of iterations, we have normalized the data with Min-Max Normalization.

Step I:

In this step, for a fixed kernel, 5 different C values will be applied and their accuracies and number of vectors will be compared and discussed. Results:

```
(base) C:\Users\mhrfk\dev\462\Assignment 3>python 2nd.py part2 step1
SVM kernel=sigmoid C=0.1 acc=0.7811 n=346
SVM kernel=sigmoid C=1 acc=0.9822 n=214
SVM kernel=sigmoid C=50 acc=0.9763 n=64
SVM kernel=sigmoid C=500 acc=0.9645 n=39
SVM kernel=sigmoid C=1000 acc=0.9586 n=35
```

From the results, we can see as we increase C, number of vectors decrease. In other words, as we choose larger-margin separating hyperplane, number of vectors increase, vice-versa. In short, the C parameter tells the SVM optimization how much one want to avoid misclassifying each training sample unit.

Till C = 1, the accuracy increases, however after that the accuracy decreases. It can concluded as the model starts overfitting after a certain value of C, which results in a decrease in accuracy.

The maximum accuracy for C seems to be about 1 for this kernel.

Step 2:

In this step, for a fixed C, 100, 5 different kernels will be applied and their accuracies and number of vectors will be compared and discussed. Results:

```
(base) C:\Users\mhrfk\dev\462\Assignment 3>python 2nd.py part2 step2
SVM kernel=linear C=1 acc=0.9704 n=73
SVM kernel=polynomial C=1 acc=0.7811 n=346
SVM kernel=radial C=1 acc=0.9763 n=174
SVM kernel=sigmoid C=1 acc=0.9822 n=214
```

From the results, it can be seen that sigmoid kernel outperforms all of them. And it is with the greater number of vectors for this result.

Step 2 Extended:

```
SVM kernel=linear C=0.1 acc=0.9763 n=150
SVM kernel=linear C=1 acc=0.9704 n=73
SVM kernel=linear C=50 acc=0.9645 n=34
SVM kernel=linear C=500 acc=0.9645 n=29
SVM kernel=linear C=1000 acc=0.9645 n=29

SVM kernel=polynomial C=0.1 acc=0.7692 n=346
SVM kernel=polynomial C=1 acc=0.7811 n=346
SVM kernel=polynomial C=50 acc=0.9763 n=172
SVM kernel=polynomial C=500 acc=0.9822 n=87
SVM kernel=polynomial C=1000 acc=0.9763 n=75

SVM kernel=radial C=0.1 acc=0.9112 n=340
SVM kernel=radial C=1 acc=0.9763 n=174
SVM kernel=radial C=50 acc=0.9822 n=53
SVM kernel=radial C=500 acc=0.9763 n=36
SVM kernel=radial C=1000 acc=0.9763 n=35

SVM kernel=sigmoid C=0.1 acc=0.7811 n=346
SVM kernel=sigmoid C=1 acc=0.9822 n=214
SVM kernel=sigmoid C=50 acc=0.9763 n=64
SVM kernel=sigmoid C=500 acc=0.9645 n=39
SVM kernel=sigmoid C=1000 acc=0.9586 n=35
```

From here we can see, that every kernel can get the maximum accuracy of 0.9822 with a different C value. However the maximum margin of error is only possible with sigmoid kernel with the value of C as 1.

In conclusion, it seems that the best result is actually a sigmoid kernel with C value as 1.