Problem 1. Decision Tree Classifier

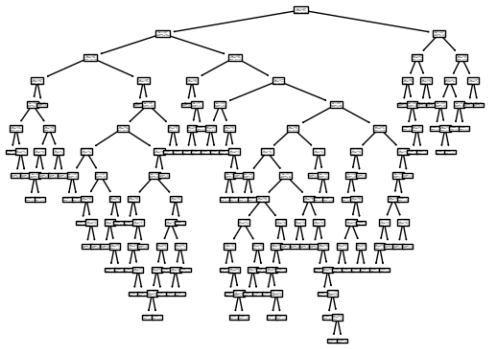
Part a

Decision Tree Classifier	LogReg
***** Train data stats *****	***** Train data stats *****
Train score: 1.00	Train score: 0.76
[[340 0]	[[294 46]
[0 197]]	[82 115]]
***** Test data stats *****	***** Test data stats *****
Test score: 0.73	Test score: 0.79
[[121 39]	[[134 26]
[24 47]]	[23 48]]
AUROC score: 0.71	AUROC score: 0.84

Overall, the logistic regression model has a better performance because its mean accuracy on test data is higher (0.79 vs. 0.73) and its AUROC score is higher (0.84 vs. 0.71). For the decision tree classifier here, we need to worry about the overfitting problem. The accuracy of the train data is 1 which denotes a perfect fitting, while the accuracy of the test data is 0.73. We fit the train data too well and too tightly.

Part b

number of nodes: 193



Part c

(1) max_depth = 5

***** Train data stats *****

Train score: 0.82

[[317 23]

 [74 123]]

***** Test data stats *****

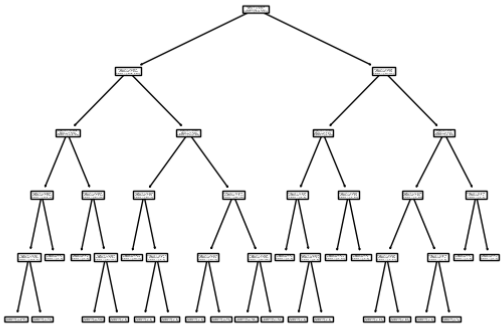
Test score: 0.79

[[143 17]

 [32 39]]

AUROC score: 0.77

number of nodes: 47



In this model, by changing max_depth to 5, we improved the mean accuracy (from 0.73 to 0.79) and AUROC score (from 0.71 to 0.77). Also, we get mean accuracy of train data as 0.82, and we don't have overfitting problem in this model. We didn't fit the model too tight and the discrepancy between train and test errors is small.

(2) min_samples_leaf = 5

***** Train data stats *****

Train score: 0.88

[[311 29]

 [37 160]]

***** Test data stats *****

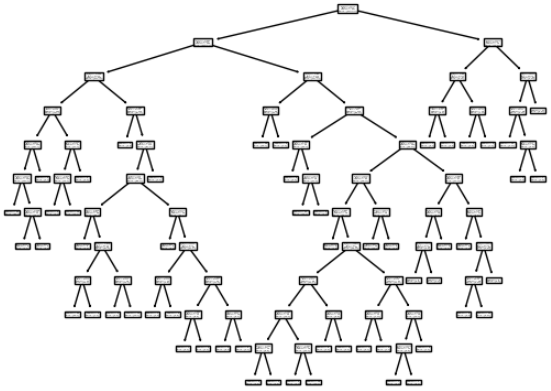
Test score: 0.74

[[121 39]

 [21 50]]

AUROC score: 0.76

number of nodes: 109



Here we improved the accuracy for test data (from 0.73 to 0.74) and AUROC score (from 0.71 to 0.76). There is no significant discrepancy between train and test errors to indicate overfitting problems: the accuracy for train data is 0.88 and for test data is 0.74.

(3) max_depth = 4

Train score: 0.80

[[274 66]

 [43 154]]

***** Test data stats *****

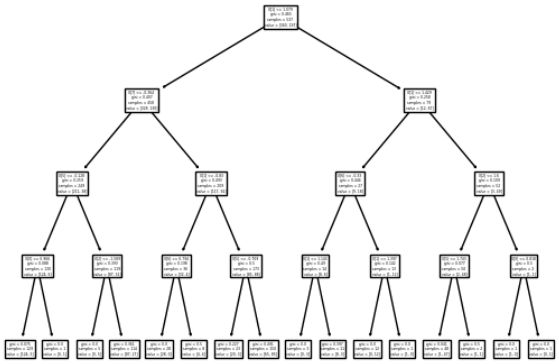
Test score: 0.71

[[111 49]

 [18 53]]

AUROC score: 0.78

number of nodes: 31



In this model, we don't have overfitting problems (which is an improvement from model in part a) because the discrepancy between train and test errors is small. There is no obvious improvement in the test accuracy score, but the AUROC score increases from 0.71 to 0.78. In the (1) model we trained in part b where the max_depth is 5, we got test accuracy score of 0.79. Compared with that model, we can see that it doesn't hold that smaller max depth leads to better results.

Problem 2. kNN classifier

Part a

KNN Classifier

***** Train data stats *****

Train score: 0.80

[[302 38]

 [69 128]]

***** Test data stats *****

Test score: 0.70

[[119 41]

 [28 43]]

AUROC score: 0.76

Based on the train and test accuracies (0.80 and 0.70), there is no significant evidence of overfitting problems.

Compared with previous models:

Models	Train score	Test score	AUROC
Logistic regression	0.76	0.79	0.84
Decision tree (part a)	1	0.73	0.71
Decision tree (max_depth = 5)	0.82	0.79	0.77
Decision tree (min_samples_leaf=5)	0.88	0.74	0.76
Decision tree (max_depth = 4)	0.80	0.71	0.78
KNN	0.80	0.70	0.76

The KNN model has similar performance to the decision tree model with max_depth = 4. Based on test score and AUROC, all of the logistic regression model, decision tree modes with max_depth and min_samples_leaf of 5 perform better than the KNN model. The KNN model performs a little better than the decision tree model from part a because the KNN one doesn't have overfitting problems and has higher AUROC score.

Part b

(1)

n_neighbors = 2

***** Train data stats *****

Train score: 0.83

[[340 0]

 [93 104]]

***** Test data stats *****

Test score: 0.72

[[139 21]

 [44 27]]

AUROC score: 0.75

n_neighbors = 3

***** Train data stats *****

Train score: 0.84

[[304 36]

 [51 146]]

***** Test data stats *****

Test score: 0.74

[[125 35]

 [25 46]]

AUROC score: 0.78

These 2 models have improvements in train and test accuracies and the AUROC score compared with the default one.

n_neighbors = 7

***** Train data stats *****

Train score: 0.79

[[301 39]

 [73 124]]

***** Test data stats *****

Test score: 0.70

[[119 41]

 [28 43]]

AUROC score: 0.77

This model performs similar to the default one, with a little decrease in train accuracy (from 0.80 to 0.79) and a little increase in AUROC (from 0.76 to 0.77).

For all of the 3 models with different n_neighbors, there is no significant evidence of overfitting problem.

By changing the number of neighbors, we can see that the model's performance changes with the number of neighbors are not monodirectional. There must be some optimal point where the model performs the best, and after that point, the performance goes down.

(2) Manhattan distance (p=1)

***** Train data stats *****

Train score: 0.80

[[303 37]

 [73 124]]

***** Test data stats *****

Test score: 0.74

[[126 34]

 [27 44]]

AUROC score: 0.77

Models	Train score	Test score	AUROC
KNN with p=2	0.80	0.70	0.76
KNN with p=1	0.80	0.74	0.77

Using Manhattan distance, we improved the performance where test score changes from 0.70 to 0.74 and the AUROC score changes from 0.76 to 0.77. Also, there is no overfitting problems in this model.

(3) algorithm = ball_tree or kd_tree

***** Train data stats *****

Train score: 0.80

[[302 38]

 [69 128]]

***** Test data stats *****

Test score: 0.70

[[119 41]

 [28 43]]

AUROC score: 0.76

There are no improvements by changing the algorithms and the ball_tree algorithm and kd_tree algorithm give the same results. There is no significant evidence of overfitting.