Elijah Chou

CS 334

Dr. Li Xiong

9/29/2021

HW 2

1. (a) – (b): See dt.py for Q1  
   (c): The following graphs were created by the q1\_c.py file in the directory. It generates the two graphs below as “maxDepth\_test\_train\_acc.png” and “minSample\_test\_train\_acc.png”, respectively, in the same directory.  
    Chart, line chart

   Description automatically generated  
   (The above graph arbitrarily set the minimum number of samples to 10)  
   Chart, line chart

   Description automatically generated  
   (The above graph arbitrarily set maximum depth of the tree to 15)  
     
   (d): The computational complexity for my train function is about O(p\*(n^2)\*d). This is because, in the worst case, every single test point (n) will be tested as a possible split point for every given feature (d) at every level of the decision tree and be compared to every other test point (n). The maximum number of levels that the train function will go down to is equal to the maximum depth (p). All complexities multiplied together will give a combined computational complexity of O(p\*(n^2)\*d).  
     
   The computational complexity for my predict function, on the other hand, is equal to O(p). This is because the predictor will only need to go down a maximum of p nodes to reach a terminal node.
2. (a) – (c): See q2.py for Q2  
   (d): Table

   Description automatically generated with medium confidence  
   All model selection techniques ended up not changing the training AUC much compared to the true test. Even though the techniques overall seemed to decrease the value estimation AUC, the difference is not that much, which suggests that the techniques help keep the estimation robust. This can be seen through the consistent ValAUCs shown in the table above. This indicates that, with the validation techniques, the decision tree model is quite robust in estimation. In terms of time, all techniques were extremely fast and required little to no additional time. A trend in an increase of time, however, can be seen when more folds or more samples are taken. Essentially, the more complicated and the more trials that are done, the slower the technique will take.
3. (a): The optimal parameters for the k-nn algorithm is 20 for number of neighbors. The optimal parameters for the decision tree classifier is 1 for minimum sample size for leaf, 3 for the maximum depth, and the Gini index for the criterion. I chose 5 for the k-fold cross validation for this since k = 5 seemed to improve the ValAUC the most out of the 3 possibilities shown above from the previous question.  
   (b) – (c): See q3.py for Q3.  
   (d): The following table was created by running q3.py.  
   Text, table

   Description automatically generated

With this data, it seems that both the K-NN algorithm and the Decision Tree classifiers are quite insensitive to the reduction of the training set size. K-NN seems to be slightly affected by it as its AUC gradually decreases with more proportion of the dataset being left out, but the Decision Tree classifier seems quite robust in this aspect in both AUC and accuracy.