Elijah Chou

CS 334

Dr. Li Xiong

11/22/2021

HW 5

1. (a): See q1.py for Q1. The predictions are printed out in the terminal after running the file.  
     
   (b): For the dataset, at least 9 components were needed to capture at least 95% of the variance in the original data. The following describes the importance of original features on the first three components, which is deduced from the values printed by q1.py that indicate the different components.  
     
   For the first component, the features fixed acidity, citric acid, density, and pH are the ones that are important. This is because they have (absolute value) coefficients of 0.49063401, 0.46749766, 0.39452529, and 0.43160612, respectively. Their relatively high values indicate they are important for the first component.   
     
   For the second component, the free sulfur dioxide, total sulfur dioxide, and alcohol features are the most important, with absolute values of 0.50263871, 0.58480694, and 0.41649727, respectively.  
     
   For the third component, the volatile acidity, free sulfur dioxide, and alcohol features are the most important, with absolute values of 0.45341577, 0.46244316, and 0.457983, respectively.  
     
   (c): The graph of the ROC curves for the logistic regression results trained on normalized dataset and on PCA dataset (using 8 components) is shown below. It is also generated in a new matplotlib window after running q1.py. Overall, the PCA curve seems to almost always be positioned above the normalized dataset, indicating that the PCA transformation improves the unregularized logistic regression model’s performance more than normalizing the data. This can be further determined by examining the areas under the ROCs (AUROCs), where the AUROC for the normalized dataset was 0.880 while it was 0.902 for the PCA dataset.  
   Chart

   Description automatically generated
2. (a): See rf.py for Q2.  
   (b): The following graphs were produced by the q2bc.py script. According to the individual tests, the optimal parameters were as follows:   
     
   Nest: 10  
   maxFeat: 5  
   maxDepth: 10

minLeafSample: 4  
  
For criterion, the OOB was calculated for both “gini” and “entropy”, but were the same. For all of the optimal parameters, I chose the ones that gave the lowest OOB error and also weren’t too high or too low to avoid under/overfitting.

Chart, line chart

Description automatically generatedChart, line chart

Description automatically generated  
Chart, line chart

Description automatically generated  
Chart, line chart

Description automatically generated  
  
Chart, line chart

Description automatically generated  
  
(c): After training the model with the optimal parameters, the accuracy of the almost Random Forest model was 89.375% on the test data. Compared to the 0.04 error estimated by OOB, it seems that the model is quite accurate on the test data, but is not as accurate as OOB would suggest it to be. These results are printed after running q2bc.py in the zip file.