Homework Two Report

University of Victoria
Instructor: Xuekui Zhang
Zhaocheng Li (V00832770)

November 24, 2018

Abstract

The report demostrates the prediction performance without stacking procedure. Thus the goal is very similar to frist assignment's goal. But differently, we apply classification method (cutoff) this time.

In order to assess the performance of each model, we construct the miss classification table and calculate the miss classification rate. Be aware that the lower the rate is, the better the model prediction gets.

Notice that although this assignment is much simplified, it takes significant runtime. I separate the *glmnet*() functions from the rest.

1 Instruciton

Please following the *README* file, which explains in detail how my program works.

Basically, for grading process, the tutor does not have to make any modification except, the working directory. Whoever running the program should adjust the working directory accordingly.

The working directory is listed at the top of code.

2 Code Skeleton

This section I will brief the basic skeleton of my entire code file to ease the workload of the grader, given the massive data needed to process.

2.1 dataset loading

In the first, of course, we are going to load the data into program and modify them into desirable format. To achieve this goal, we here apply the professor's $Helperfunctions.R$ to do so.

In the end of this part, we will have a matrix that contains the outcomes of five drugs (continuous response) and corresponding mutual gene status (discrete variables). We also have the our needed libraries ready to use later.

2.2 function preparation

I have created three main strucutures I need to use throughout the code. The $cutoff$ function is used in classification process, making a matrix of continuous y-values (5 drugs outcomes) as input and transform it into a matrix of discrete values (0's and 1's);

Also, the *reorder* function is used in converting unorder matrix into one by the order of *columnnames*.

At the end, there is a 8-by-20 matrix defined at first, called $Matrix$, which is a storage to save the miss classification rates for eight models, in 20 seeds (a ramdom index). This is basically the result we want to have and study for this assignment.

In addtion, there are also alot minor temporary data structures I use in the middle of computation, and I will demonstrate their uses in comments as I use them.

2.3 method one & analyzing first four models

I did not include set of *glmnet* functions here, but separate them into the following individual section. This is due to the consideration of massive runtime for running all of them together. Although the fact suggests that in the way, it still costs significantly long runtime.

As required, in this part, before comparing the methods in 20 loops with 5-fold cross validation, we need to go through the $cutoff$ with matrix first, to make it a $discrete - valued$ matrix. This is method one. And the models we assess are logistic regression, LDA, KNN, and classification tree.

2.4 method two & analyzing another two models

Similarly, we leave out the family of *glmnet* functions. The models we uses here are linear regression and regression tree.

Method two simply means that we predict first using the continuous variables, then convert them into discrete variable using $cutoff$ function.

2.5 method 1,2 & analyzing penalized regression

Then we will study the glmnet functions here using the two methods, repectively.

Notice that this is a huge workload computationally because there are 11 models to process each time, including Lasso, Ridge, and other elastic net functions. And in comparison we pick the one with best peroformance, i.e. the one with lowest miss classification rate, to take into consideration.

3 Outcome Display

The table $Matrix$ shown below is recording the miss classification rate for each model in each random seed out of 20. It represents how the model behaves and it gives us better chance to know more by analyzing on it.

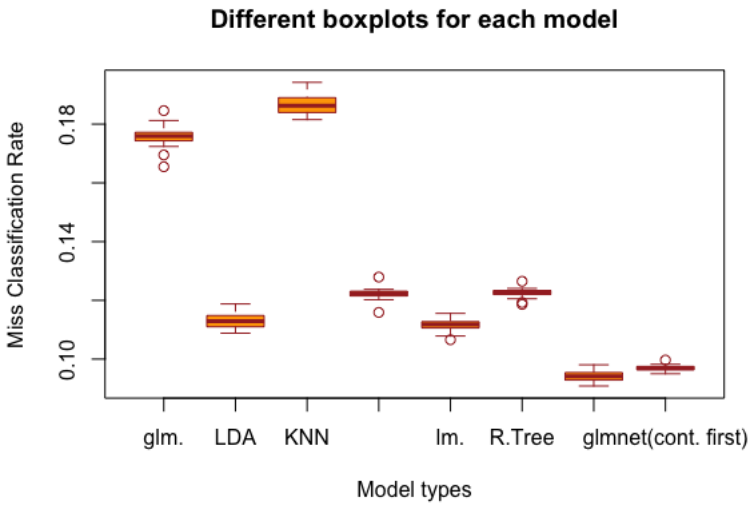
Of course, we can easily see, for each model, different distribution of sets in

cross-validation (the change of seeds) does not incluence its behaviour. Each model performs steadily among all seeds. Which will ensure a convincing outcome of analysis.

1	0.1770465	0.1139647	0.1828250	0.1224719	0.1147673	0.1232745	0.09550562	0.09775281
2	0.1747994	0.1150883	0.1844302	0.1229535	0.1107544	0.1192616	0.09149278	0.09630819
3	0.1741573	0.1097913	0.1889246	0.1224719	0.1128411	0.1234350	0.09534510	0.09695024
4	0.1731942	0.1133226	0.1924559	0.1234350	0.1141252	0.1186196	0.09807384	0.09759230
5	0.1812199	0.1187801	0.1884430	0.1223114	0.1109149	0.1223114	0.09245586	0.09502408
6	0.1764045	0.1168539	0.1855538	0.1210273	0.1113965	0.1240770	0.09406100	0.09743178
7	0.1797753	0.112 0.1168539	4799	0.1205457	0.1121990	0.1231140	0.09325843	0.09727127
8	0.1747994	0.1144462	0.1890851	0.1218299	0.1078652	0.1229535	0.09614767	0.09502408
9	0.1723917	0.1088283	0.1834671	0.1224719	0.1115570	0.1223114	0.09293740	0.09823435
10	0.1794543	0.1117175	0.1942215	0.1213483	0.1118780	0.1239165	0.09390048	0.09646870
11	0.1759230	0.1163724	0.1865169	0.1231140	0.1117175	0.1231140	0.09197432	0.09646870
12	0.1845907	0.1110754	0.1852327	0.1232745	0.1155698	0.1226324	0.09438202	0.09678973
13	0.1773676	0.1162119	0.1834671	0.1279294	0.1126806	0.1219904	0.09614767	0.09518459
14	0.1746388	0.1146067	0.1815409	0.1219904	0.1093098	0.1264848	0.09566613	0.09614767
15	0.1764045	0.1128411	0.1908507	0.1237560	0.1149278	0.1237560	0.09390048	0.09695024
16	0.1695024	0.1109149	0.1860353	0.1218299	0.1065811	0.1219904	0.09470305	0.09967897
17	0.1744783	0.1128411	0.1818620	0.1216693	0.1120385	0.1213483	0.09486356	0.09759230
18	0.1759230	0.1107544	0.1897271	0.1202247	0.1104334	0.1226324	0.09277689	0.09646870
19	0.1762440	0.1130016	0.1886035	0.1158909	0.1097913	0.1205457	0.09085072	0.09727127
20	0.1654896	0.1104334	0.1850722	0.1231140	0.1126806	0.1221509	0.09422151	0.09646870

This is the type of solution matrix we can get, 8 models, 20 seeds, miss classification rate in each table

4 Findings



By the solution matrix, we have constructed a system of boxplots for each model, we can see that the highest miss classification rate is KNN, and maybe glm as well. And it may indicates a relative poor performance of prediction. However, the glmnet functions are the most accurate function to make prediction, and the glmnet using cutoff first is lowest. It performs very well.