Construction of an IBD Classifier Using Machine Learning Methods

Peiyu Zhan

28 April 2022

Abstract

Inflammatory bowel disease (IBD) is a group of intestinal disorders which can cause chronic and incurable inflammation of the digestive tract (CDC, 2018). Although IBD has no specific biomarker, it has been found that the genetic susceptibility plays a key role in the occurrence of IBD (Loddo &amp; Romano, 2015). In this paper we propose to construct an IBD classifier based on a gene expression data set using machine learning methods. Several algorithms in machine learning have shown great success in performing the task of binary classification. The three methods considered in this paper include logistic regression, k-nearest neighbours, and decision tree. However, the small sample size makes it non-trivial to validate each model for choosing the top performer, and the large number of features makes it necessary to find proper regularization approaches. For the first concern, we applied split set approach and run 10 epochs for each model to compare their average performance in different evaluation measures. For the second concern, we implemented ridge regression and least absolute shrinkage and selection operator for the logistic model and pruning method for the decision tree. The results show that logistic regression with LASSO regularization is comparatively the best classifier, hence we trained it using the whole data set as the final classifier that can be used for future applications.

Keyword: Inflammatory bowel disease, LASSO, KNN, Ridge, Classification Tree

Contents

*Code and data are available at: https://github.com/peiyuzhan/sta304-final-paper

1. Introduction

Machine learning is the subject composed of computer science and statistics that studies how to make algorithms learn patterns and features from big data and improve their performance over time without being explicitly programmed to do so (, n.d.) (IBM Cloud Education). Although the origin of machine learning can only date back to 1950s, it has become one of the most popular research areas and the driving force of many technologies such as stock prediction, face recognition, language translation, and self-driving cars in the recent decades. After years of vigorous development, many segments of machine learning have grown up. Classification, which is the process of predicting the categories (class) of given data points based on a set of features (predictors), has been regarded as the most classical and representative subsection of machine learning. A lot of particular algorithms have been invented to perform the task of classification. Once people have a datapoint whose class and several features are given, it can be fed to the algorithms for exploring the patterns between the predictors and classes. Such data points are called training examples. Each trained algorithm could be regarded as a function$f: X \to Y$, where X is the set of all possible values of the predictors and Y represents the collection of all categories. Every time the trained algorithm is given a new datapoint with a set of features $(x_1, x_2, \dots, x_p)$, it can generate a predicted category $y = f(x_1, x_2, \dots, x_p)$. Theoretically speaking, as more and more training examples are fed to the algorithms, the predicted class has a greater chance of being equal to the true class. These algorithms are playing a critical role in the fields of medical science and public health. One representative application is disease prediction.

Inflammatory bowel disease (IBD), is a general term that describes a group of disorders that can cause prolonged chronic inflammation in human's digestive tract. IBD is usually classified into two types which are ulcerative colitis and Cohn's disease. Patients with either type of IBD have to suffer from symptoms including severe diarrhea, abdominal pain, fatigue, bloody stools and unintended weight loss (CDC, 2018). Approximately 1.5 million people have IBD in North America where the number is among the highest in the world (CDC, 2020). Although IBD is not a rare disease, it is still incurable and more importantly its specific cause still remains unknown. However, it was recently mentioned by Genome-wide Association Studies (GWAS) that genetic susceptibility is playing a crucial role in the occurrence of IBD, which means genetic data can be potentially used to construct a biomarker for IBD, which is not only useful for exploring better treatments and customized patient care but also helpful for promoting the development of clinical trials involving new medications (SSC, 2017). In addition, if an effective classifier can be constructed, we will be able to identify the population with high incidence risk of IBD so that some preventive actions such as lifestyle changes and medications can be implemented ahead. The earlier the patients treat them, the greater the chance they will live a longer and healthier life. Therefore, the work is worth doing.

The objective of this paper contains two parts. First, we will assess and compare the performance of three types of classifiers including logistic regression with regularization, k-nearest neighbours, and decision tree. Second, based the performance result, we will train a model that can be used for classification of future observations.

The comparison of the three methods and their methodology will be discussed in the Model section. In Result section, we summarize the performance of each method and make the final decision. Finally in the section 4, we discuss the limitations of this study and conclude the relationship between gene and IBD. Appendix will contain the box plots that demonstrate the distributions of the performance of the methods we referred in the result section.

2. Data

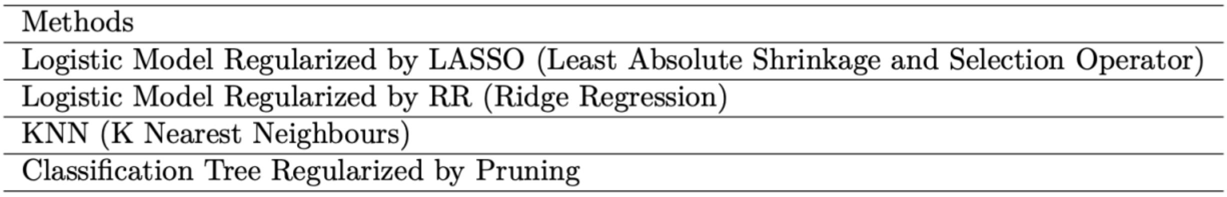
This paper using the statistical programming language R and the data of this study comes from the 2017 case study on SSC website. It contains genome-wide gene expression profiles of 41 healthy people and 85 IBD patients. Among the observations with IBD, 26 of them were recorded as ulcerative colitis patients and 59 of them were recorded as Cohn's disease patients. For each observation, some personal information including illness status, age, ethnicity, and sex are recorded. Also, the expression levels of 309 probe sets are quantified and contained in the data for each subject. According to the research objective of this paper, we will consider the illness status, namely 'Group', as the response variable and the rest features as the predictors. Although the response variable has three levels, we will dichotomize it as a binary variable where each 0 represents a healthy individual and each 1 represents an IBD patient. Several classifiers will be trained using this data and their performance will be assessed and compared using resampling method.

3. Model

3.1 Methods comparison

Our work will consist of four specific approaches supported by three classification models:

Table 1: Summary of the machine learning methods



Methods
Logistic Model Regularized by LASSO (Least Absolute Shrinkage and Selection Operator)
Logistic Model Regularized by RR (Ridge Regression)
KNN (K Nearest Neighbours)
Classification Tree Regularized by Pruning

The methodology of each approach is one-by-one explained in details in the following sections.

3.1.1 Logistic Regression with Regularization

As logistic regression is technically created to model categorical response variables, it has been the most commonly used and well-studied prediction method for classification. Let Y be the binary response variable which takes values of 0 (for health individual) and 1(for IBD patient), and $X_1, X_2, \ldots, X_p$ be the predictors (age, ethnicity, sex, and the quantified probe set levels). Then the relationship between $Y$and X's is modelled by logistic regression as:

$$\log\left(\frac{P\left(Y=1 \mid X_1, \ldots, X_p\right)}{1-P\left(Y=1 \mid X_1, \ldots, X_p\right)}\right)=w_0+w_1 X_1+\ldots+w_p X_p$$

It is shown in the equation that this model does not directly classify the response variable but instead it assigns each observation with a probability of belonging to a class, given the values of the predictors. This probability will act as a decision maker for classification. By assuming the response variable Y follows a Bernoulli distribution with parameter $P(Y = 1)$, we can derive the likelihood function of the weights $w_0, w_1, \ldots, w_p$:

$$L\left(w_0, \ldots, w_p\right)=\prod_{i=1}^N\left(\frac{\exp \left(w_0+w_1 x_{i 1}+\ldots+w_p x_{i p}\right)}{1+\exp \left(w_0+w_1 x_{i 1}+\ldots+w_p x_{i p}\right)}\right)^{y_i}\left(1-\frac{\exp \left(w_0+w_1 x_{i 1}+\ldots+w_p x_{i p}\right)}{1+\exp \left(w_0+w_1 x_{i 1}+\ldots+w_p x_{i p}\right)}\right)^{1-y_i}$$

The training purpose is to find weight estimates $\hat{w}_0, \hat{w}_1, \ldots, \hat{w}_p$which can maximize the log likelihood function. Although there is no closed form solution of the estimates, many built-in functions in various programming languages can help us to solve them numerically given any training data. Once we obtain the estimates, we will be able to classify a new observation with known feature values into a category based on the predicted probabilities:

$$\hat{P}\left(Y=1 \mid x_1, \ldots, x_p\right)=\frac{\exp \left(\hat{w}_0+\hat{w}_1 x_1+\ldots+\hat{w}_p x_p\right)}{1+\exp \left(\hat{w}_0+\hat{w}_1 x_1+\ldots+\hat{w}_p x_p\right)}$$

$$\hat{P}\left(Y=0 \mid x_1, \ldots, x_p\right)=1-\frac{\exp \left(\hat{w}_0+\hat{w}_1 x_1+\ldots+\hat{w}_p x_p\right)}{1+\exp \left(\hat{w}_0+\hat{w}_1 x_1+\ldots+\hat{w}_p x_p\right)}$$

If $\hat{P}(Y = 1|x_1, \ldots, x_p) > 0.5$, we will classify the new observation into the group of IBD patients (i.e. $\hat{Y} = 1$). Otherwise, we will classify it as a healthy individual.

However, the presence of a large number of predictors can introduce noise in the model which can lead to undesirable overfitting and non-ideal prediction performance. In our data the number of predictors is 312 which is even two-times larger than the sample size (126), therefore the concern of overflexibility needs to be taken into consideration. One solution to this concern is regularization in which the size of $\hat{w}_0, \hat{w}_1, \ldots, \hat{w}_p$are shrunk by adding constraints or penalties in the process of maximizing the likelihood function. We will focus on two regularization methods: LASSO (Least Absolute Shrinkage and Selection Operator) and RR (Ridge Regression). Note that the training purpose of logistic model is to find weight estimates that maximize $logL(w_0, \ldots, w_p)$ or equivalently minimize $-logL(w_0, \ldots, w_p)$. Hence, the negative likelihood function could be considered as a loss function. In both RR and LASSO, an additional term, which is proportional to the weights' size, is added to the loss function $-logL(w_0, \ldots, w_p)$:

$$RR: -\log\left(L\left(w_0, \ldots, w_p\right)\right) + \lambda \sum_{j=1}^{p} w_j^2$$

$$LASSO: -\log\left(L\left(w_0, \ldots, w_p\right)\right) + \lambda \sum_{j=1}^{p} \left|w_j\right|$$

By using the updated loss functions, the loss value becomes proportional to the size of the weight estimates, which means larger weights will apply more penalty to the loss function. Thus the weights are controlled from being too large to cause overfitting. In both equations, the parameter λ is a positive number that controls the magnitude of the penalty. As λ increases, we will have larger penalty, smaller weight estimates, and less model flexibility. A smaller λ implies larger flexibility but smaller training bias. The choice of λ is important due to the bias-variance trade off. We will use cross validation to find the optimal λ value in the training process.

Since the penalty terms added to the loss function are different for RR and LASSO, the two methods actually perform different forms of regularization. Ridge regression shrinks the weights but does not perform feature selection. However, LASSO works to select only a subset of predictors by forcing some of the weights to become 0. RR usually have better performance when most of the features are associated with the response while LASSO performs better when only some are strong predictors and the rest add noise. People often cannot know in advance which one is better, hence we will train two logistic models that are respectively regularized by RR and LASSO, and then compare their prediction capability.

3.1.2 K-Nearest Neighbours

Although Logistic regression is a mature model for binary classification, it still needs several statistical assumptions like that the responses should identically and independently follow a Bernoulli distribution. K-Nearest Neighbours is a sort of Bayes classifier that is particularly designed to perform classification with no need of assumption. This non-parametric model estimates the conditional probabilities of belonging to a class by inspecting the classes of the neighbours.

Since we started to talk about neighbours, it is necessary to introduce the concept of distance in the feature space before explaining the details of the algorithm. Given any two data points m_1 and m_2 in the feature space, they can be represented by two sets of predictor values:

$$m_1 = \{X_1 = s_1, X_2 = s_2, \ldots, X_p = s_p\} \& \, m_2 = \{X_1 = t_1, X_2 = t_2, \ldots, X_p = t_p\}$$

Then the distance between them is calculated using the Euclidean distance formula:

$$d\left(m_1, m_2\right) = d\left(\left(s_1, s_2, \ldots, s_p\right), \left(t_1, t_2, \ldots, t_p\right)\right) = \sqrt{\sum_{i=1}^{p}\left(s_i - t_i\right)^2}$$

Given any new observation with features $\{X_1 = x_1, X_2 = x_2, \ldots, X_p = x_p\}$, we will classify its corresponding Y as 0 (absence of IBD) or 1 (presence of IBD) based on a given training data set in the following way. First, we select K data points from the training set which are closet to the new observation in the feature space. We put these K nearest data points in a set denoted by T. Then we estimate the conditional probability of each class using sample proportions in set T:

$$\hat{P}\left(Y = 0 \mid x_1, \ldots, x_p\right) = \frac{1}{K} \sum_{i \in T} I\left(y_i = 0\right); \hat{P}\left(Y = 1 \mid x_1, \ldots, x_p\right) = \frac{1}{K} \sum_{i \in T} I\left(y_i = 1\right)$$

Then we will assign Y with the class that has the highest estimated conditional probability. This is called the majority vote rule. Although the mechanism of KNN is simple, it usually does pretty well in utilizing patterns and associations in the data. One critical point is how to choose the optimal value of K since K determines the bias-variance trade-off of the model. Small values of K may lead to overfitting while large values of K may cause undesirable bias. Hence finding an optimal K is the major training purpose in KNN. We will obtain the optimal value of K using a resampling method called leave-one-out which is actually one-fold cross validation.

3.1.3 Decision Tree

The last predictive model we will take into consideration is decision trees in which the feature space is split in a recursive way using one feature at a time. The approach's name comes from the segmentation of the feature space which can be summarized and visualized by a tree-like structure.

Once the model is fed with a set of observations, the data will be partitioned in an iterative way using one predictor at a time. First, the algorithm will decide one feature to be used for data partition. If the chosen feature is categorical, the data will be simply split based on the levels of the feature. If the chosen predictor is continuous, the algorithm will determine a numeric threshold first and then partition the data based on the threshold. After the initial partitioning is finished, each one of the two subdivided sets of data can be further partitioned. This will be implemented again after selecting features and thresholds for each one of the partitions. Therefore, the resulted model can have a tree-like structure. Each split in the tree is called a node, and the terminal nodes are referred as leaves. Each leaf of the tree is finally assigned with the class that has the highest proportion in that node.

The criterion used by the algorithm to choose feature and threshold at each time is called purity. This term describes how homogeneous the subdivided data regions are in terms of having data from the same class as much as possible. If we let m be a node with K split regions, then the purity of node can be quantified through two mathematical functions:

$$1.\ \text{Gini Index: } G = \sum_{k=1}^{K} \hat{p}_{mk} \left(1 - \hat{p}_{mk}\right)$$

$$2.\ \text{Cross-entropy: } D = -\sum_{k=1}^{K} \hat{p}_{mk} \log\left(\hat{p}_{mk}\right)$$

In the above equations, $\hat{p}_{mk}$ stands for the proportion of data points with level k. Both of these measures take values close to 0 if $\hat{p}_{mk}$ is either close to 0 or close to 1 which represents a low level of homogeneity. The training objective is to ensure the data regions created at each node to be homogeneous as much as possible. The splitting will stop if no further split can improve the purity.

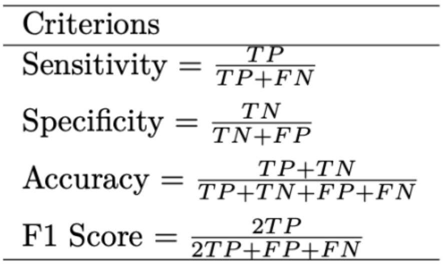
Once we have a trained tree model and a new observation with known feature values, the algorithm can take it down the tree and ends in one of the leaves. The estimated class of the new data point will be the class that has been assigned to the leaf where it ends at.

According to the mechanism introduced above, the complexity of a tree model can be significantly related to the number of features. Hence the large number of predictors in our case can lead to an over-complicated model which can cause overfitting as well as non-ideal running time. Therefore, once the tree is trained, we will prune it by eliminating some of the terminal branches. Suppose T is a tree model, we can calculate its corresponding loss value using the following loss function $LOSS(T) = N_m + \alpha|T|$, where N_m represents the number of misclassifications, $|T|$represents the number of leaves, and α is a complexity parameter that decides the magnitude of penalty. Let T_{full} represent the fully developed tree and $S = \{T \subset T_{full}\}$ be the set of all pruned subtrees of T_{full}. Then the chosen T_c should be the element in S that gives the minimum $LOSS(T)$. The working principle of pruning is similar to the regularization of logistic model, thus it is also necessary to find an optimal value of α which can balance between bias and variance. The optimal will be obtained using cross validation like before.

3.2 Validation Method

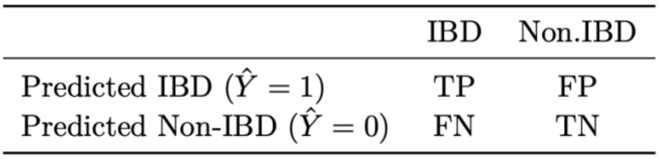
We will assess and compare the performance of these classification methods using split set approach. Each time we will randomly select 2/3 of the observations for training and use the rest 1/3 for validating. Since there is randomness involved, we will respectively repeat the process ten times for each model and record the performance in each epoch. Finally we will compare the average performance for each model over the ten epochs. There are multiple measures that can be used to evaluate the classification performance. Our criterions will be the following four quantities that can be derived from a confusion matrix:

Table 2: Criterions



Criterions
Sensitivity $= \frac{TP}{TP+FN}$
Specificity $= \frac{TN}{TN+FP}$
Accuracy $= \frac{TP+TN}{TP+TN+FP+FN}$
F1 Score $= \frac{2TP}{2TP+FP+FN}$

Table 3: Confusion Matrix

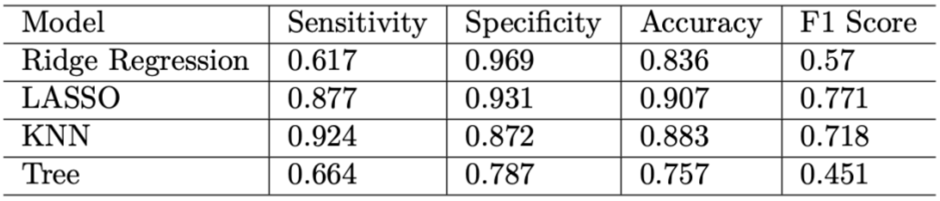


	IBD	Non.IBD
Predicted IBD ($\hat{Y} = 1$)	TP	FP
Predicted Non-IBD ($\hat{Y} = 0$)	FN	TN

Since different measures have different practical meanings and good performance on one measure does not imply good performance on the other measures, we will assess and compare the classifiers' performance using all of the four evaluation scores. The different application scenarios of these four scores will be further discussed in the conclusion.

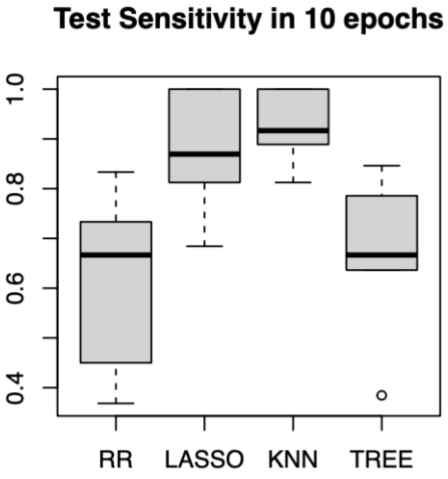
4. Results

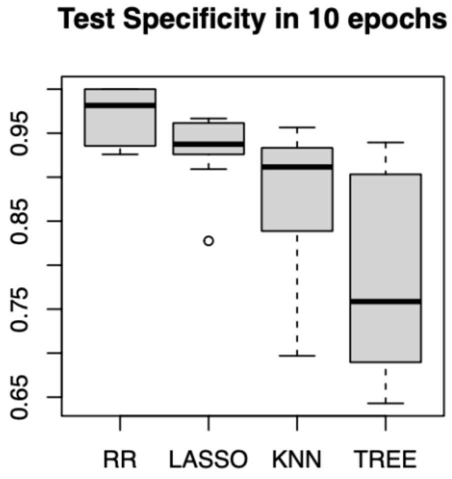
After we repeatedly tested each model ten times using split set approach, their average test sensitivity is summarized in the following table:

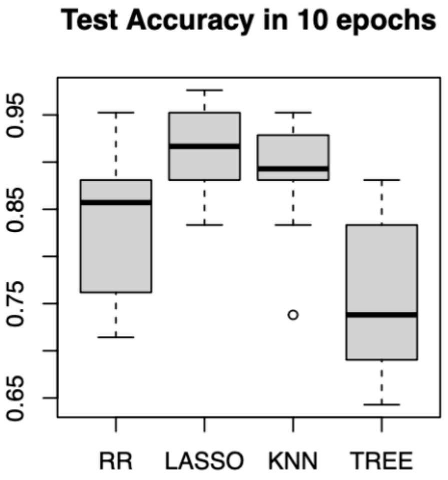


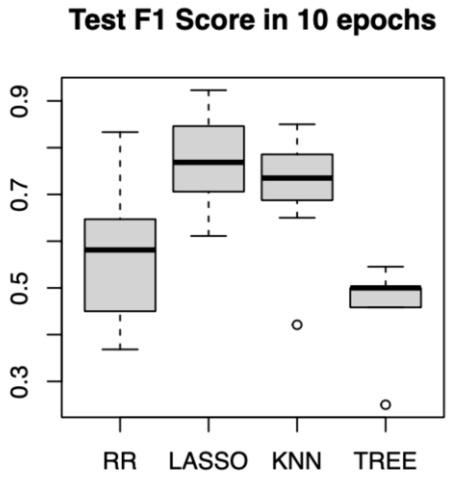
Model	Sensitivity	Specificity	Accuracy	F1 Score
Ridge Regression	0.617	0.969	0.836	0.57
LASSO	0.877	0.931	0.907	0.771
KNN	0.924	0.872	0.883	0.718
Tree	0.664	0.787	0.757	0.451

The first thing to note is that each method is playing a role since all the scores are greater than 0.5. Then it can be seen from the table that logistic model with RR is the top performer on specificity, logistic model with LASSO shows the best performance on both accuracy and F1 score, and KNN gets the highest score on sensitivity. Also, it is worth noting that decision tree has the poorest overall performance. The tree's score is always at the bottom except for specificity. The following box plots in appendix further demonstrate the distributions of the performance of the proposed methods among the 10 epochs for each prediction score.









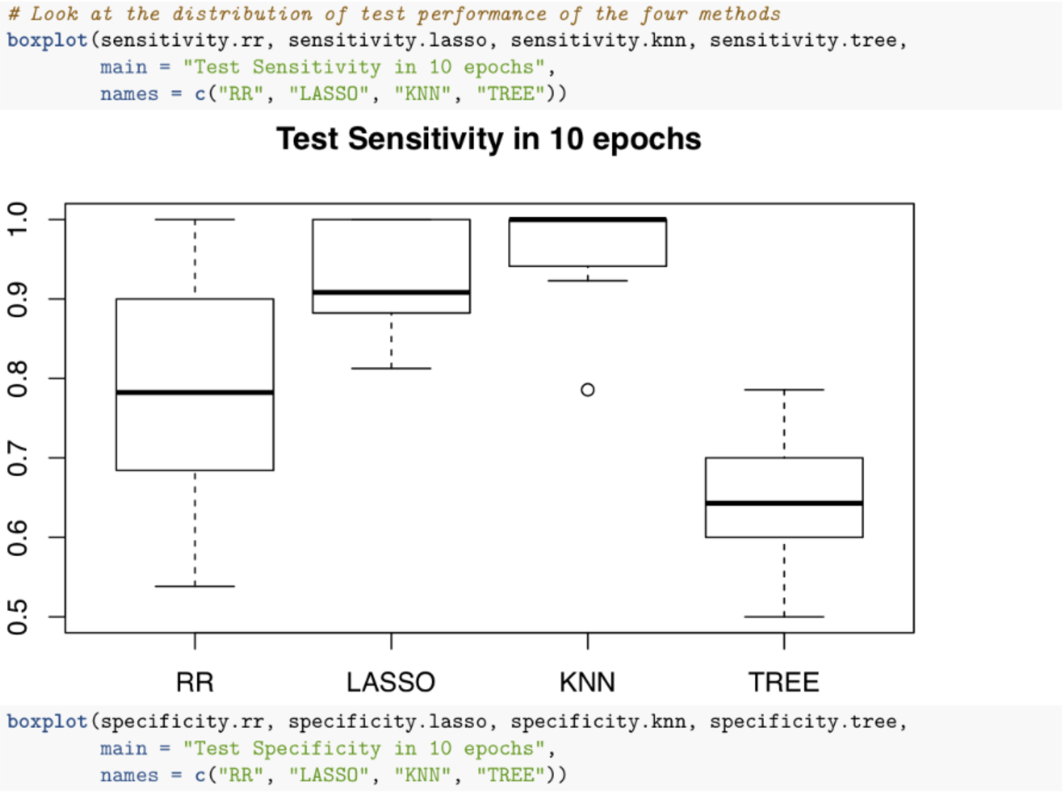
It is indicated from the box plots that the top performer of each measure also owns the most consistent distribution. For sensitivity, the performance of KNN is the best as well as the most consistent among all the methods. The median performance of KNN on sensitivity is 1, which means that KNN produced perfect validation sensitivities half of the time. For specificity, RR is the best performer with the most consistent performance. Only two of the ten validation specificities of RR were less than one. With respect to accuracy and F1 score, LASSO's performance is the best and most consistent. Since the logistic model with LASSO has been the winner on two of the four evaluation measures and it has no obvious shortfalls on the other measures, we claim the winner is logistic regression with LASSO regularization. Finally, we trained a logistic model with LASSO regularization using the whole data set as the ultimate classifier for future IBD classification.

5. Conclusion & Discussion

We assessed and compared four classification models including logistic regression with RR regularization, logistic regression with LASSO regularization, K-nearest neighbours, and decision tree based on an IBD diagnosis data set found on the SSC website. The classification performance was respectively evaluated using four measures including sensitivity, specificity, accuracy, and F1 score. The validation results show that every classification model is instrumental, which mediately confirms the relationship between gene and IBD. Besides that the tree showed the poorest overall performance, the other methods all have their own merits and demerits with respect to each measure. Finally, we decided to train a logistic model with LASSO regularization using the whole data set as the final classifier because it was the top performer on both accuracy and F1 score without showing shortage on the rest of the measures. Using this model, we will be capable of doing IBD diagnosis or providing an estimated chance of getting IBD for future observations.

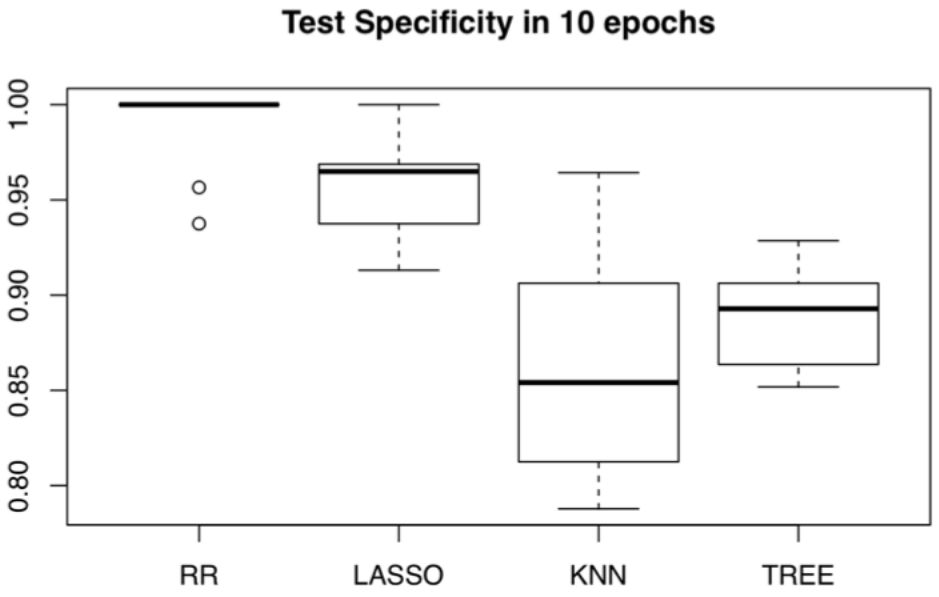
There are some non-negligible limitations existing in our study. First, the sample size is insufficient. In our case, although we have more than one thousand predictors, the number of observations is only 126. As a consequence, we do not have enough training data to tune each classifier as effective as possible. Also, the small size of training data can limit the performance of some models. For example, the tree model becomes very complex when the number of features is too large due to its mechanism. Therefore, the advantage of tree cannot be shown without excessive training. This is why the tree model had the poorest validation result. Second, the criterion of a good IBD classifier can vary from situation to situation. For example, if we know that a false negative cost much more than a false positive, we are supposed to use the model whose sensitivity score is the highest. In that case we will prefer KNN rather than logistic regression according to the validation results.

Appendix

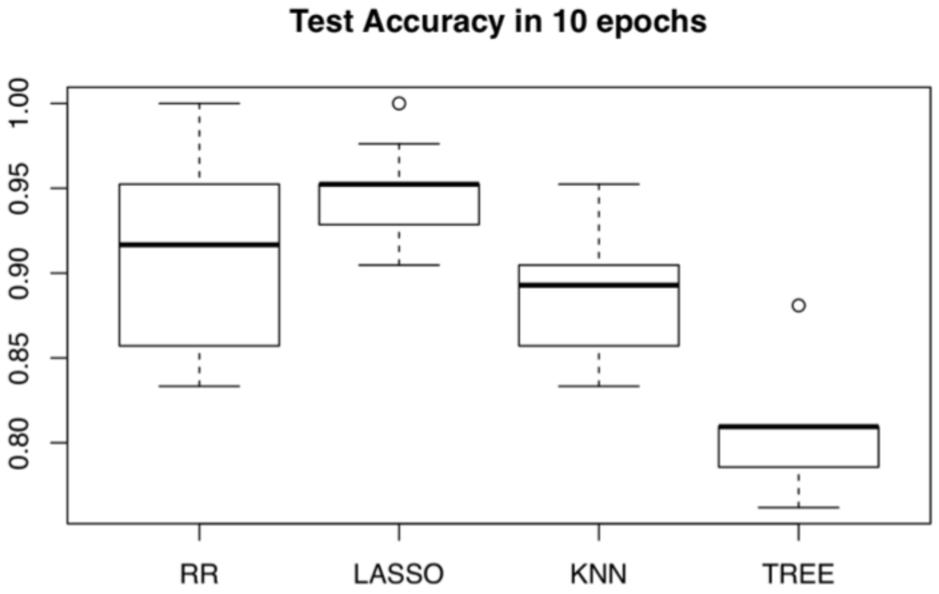


```r
# Look at the distribution of test performance of the four methods
boxplot(sensitivity.rr, sensitivity.lasso, sensitivity.knn, sensitivity.tree,
        main = "Test Sensitivity in 10 epochs",
        names = c("RR", "LASSO", "KNN", "TREE"))
```

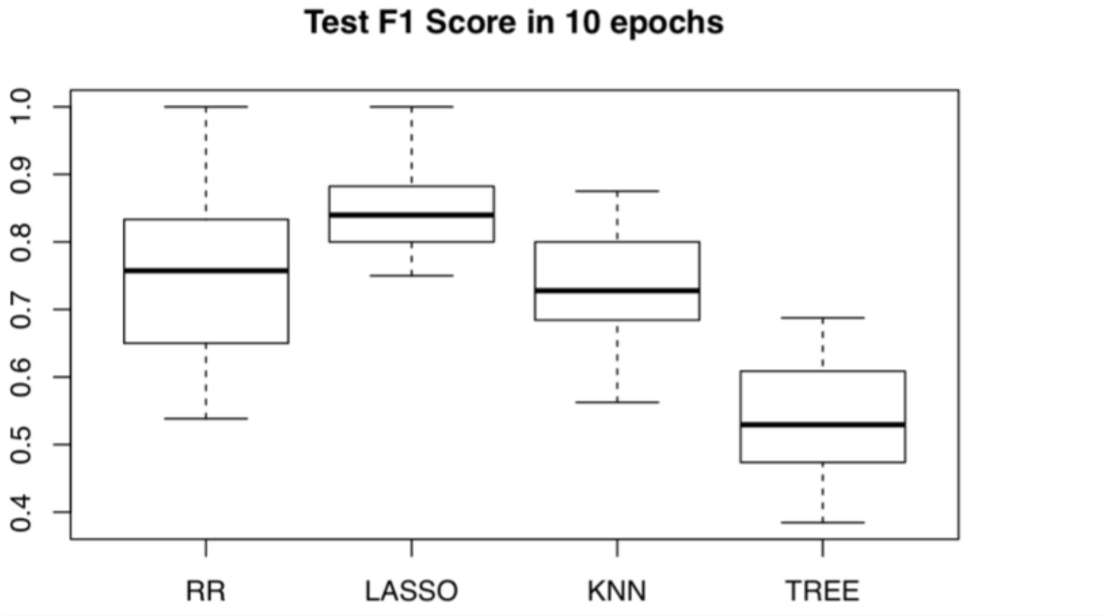
```r
boxplot(specificity.rr, specificity.lasso, specificity.knn, specificity.tree,
        main = "Test Specificity in 10 epochs",
        names = c("RR", "LASSO", "KNN", "TREE"))
```



```
boxplot(accuracy.rr, accuracy.lasso, accuracy.knn, accuracy.tree,
        main = "Test Accuracy in 10 epochs",
        names = c("RR", "LASSO", "KNN", "TREE"))
```



```
boxplot(f1.rr, f1.lasso, f1.knn, f1.tree,
        main = "Test F1 Score in 10 epochs",
        names = c("RR", "LASSO", "KNN", "TREE"))
```


Test F1 Score in 10 epochs
0.4
0.5
0.6
0.7
0.8
0.9
1.0
RR
LASSO
KNN
TREE

Reference

By: IBM Cloud Education. (n.d.). What is Machine Learning? Retrieved January 12, 2021, from https://www.ibm.com/cloud/learn/machine-learning

CDC -What is inflammatory bowel disease (IBD)? - Inflammatory Bowel Disease - Division of Population Health. (2018, March 22). Retrieved January 12, 2021, from https://www.cdc.gov/ibd/what-is-IBD.htm

Loddo, I., &amp; Romano, C. (2015, November 2). Inflammatory Bowel Disease: Genetics, Epigenetics, and Pathogenesis. Retrieved January 12, 2021, from https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4629465/

Machine Learning. (n.d.). Retrieved January 12, 2021, from https://www.coursera.org/learn/machine-learning

Statistical Society of Canada. (n.d.). Retrieved January 12, 2021, from https://ssc.ca/en/meeting/annual/2017/case-study-2

R reference

R Core Team. 2020. *R: A Language and Environment for Statistical Computing*. Vienna, Austria: RFoundation for Statistical Computing. https://www.R-project.org/.

Hadley Wickham, Jennifer Bryan, RStudio, Marcin Kalicinski, Komarov Valery, Christophe Leitienne, Bob Colbert, David Hoerl, Evan Miller. 2022. readxl: Read Excel Files. https://CRAN.R-project.org/package=readxl

Jerome Friedman, Trevor Hastie, Rob Tibshirani, Balasubramanian Narasimhan, Kenneth Tay, Noah Simon, Junyang Qian, James Yang. 2021. glmnet: Lasso and Elastic-Net Regularized Generalized Linear Models. https://CRAN.R-project.org/package=glmnet

Xavier Robin, Natacha Turck, Alexandre Hainard, Natalia Tiberti, Frédérique Lisacek, Jean-Charles Sanchez, Markus Müller, Stefan Siegert, Matthias Doering, Zane Billings. 2021. pROC: Display and Analyze ROC Curves. https://CRAN.R-project.org/package=pROC

Max Kuhn, Jed Wing, Steve Weston, Andre Williams, Chris Keefer, Allan Engelhardt, Tony Cooper, Zachary Mayer, Brenton Kenkel, R Core Team, Michael Benesty, Reynald Lescarbeau, Andrew Ziem, Luca Scrucca, Yuan Tang, Can Candan, Tyler Hunt. 2022. caret: Classification and Regression Training. https://CRAN.R-project.org/package=caret

Brian Ripley, William Venables. 2022. class: Functions for Classification. https://CRAN.R-project.org/package=class

Mark Stevenson and Evan Sergeant with contributions from Telmo Nunes, Cord Heuer, Jonathon Marshall, Javier Sanchez, Ron Thornton, Jeno Reiczigel, Jim Robison-Cox, Paola Sebastiani, Peter Solymos, Kazuki Yoshida, Geoff Jones, Sarah Pirikahu, Simon Firestone, Ryan Kyle, Johann Popp, Mathew Jay, Charles Reynard, Allison Cheung, Nagendra Singanallur, Aniko Szabo and Ahmad

Rabiee. 2022. epiR: Tools for the Analysis of Epidemiological Data. https://CRAN.R-project.org/package=epiR
Brian Ripley. 2021. tree: Classification and Regression Trees. https://CRAN.R-project.org/package=tree

Terry Therneau, Beth Atkinson, Brian Ripley. 2022. rpart: Recursive Partitioning and Regression Trees. https://CRAN.R-project.org/package=rpart