1.
Done.

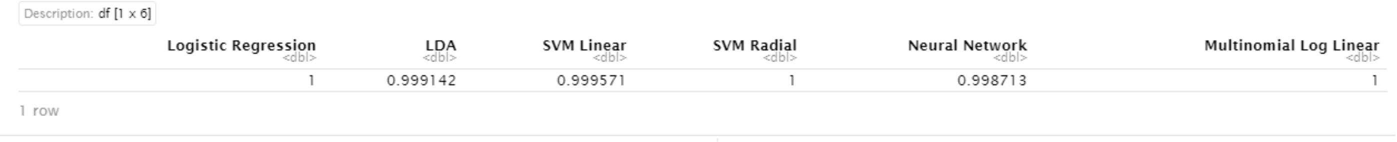
2.
https://github.com/EdwardS2027/biostat626_midterm1

3.
Done. It is in the github repository. It is labeled under binary classifier instructions and multiclass classifier instructions.

4.

For the binary classifier, we first do a lasso regression on our training data to select our features. Then, we splitted the training data into 70 percent training and 30 percent validation. We fitted the logistic regression, linear discriminant analysis, SVM with linear kernel, SVM with radial kernel, neural network, and multinomial log linear model. The neural network has three hidden layers, and both of the SVM models have the cost of 10.

Below is the accuracy results of the baseline algorithms for the binary classifier:



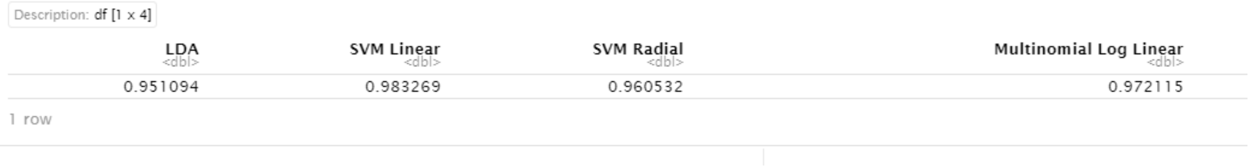
Description: df [1 x 6]

Logistic Regression <dbl>	LDA <dbl>	SVM Linear <dbl>	SVM Radial <dbl>	Neural Network <dbl>	Multinomial Log Linear <dbl>
1	0.999142	0.999571	1	0.998713	1

1 row

For the multiclass classifier, we utilized the cross validation with 5 fold in our feature selection. The feature selection was composed by the Boruta Package, which is random forest classification. We also splitted the training data into 70 percent training and 30 percent validation. For model training, we choose SVM with linear kernel, SVM with radial kernel, linear discriminant analysis, and multinomial log linear model. We also hypertune the parameters for SVM, which both resulted in 15 costs.

Below is the accuracy results of the baseline algorithms for the multiclass classifier:



Description: df [1 x 4]

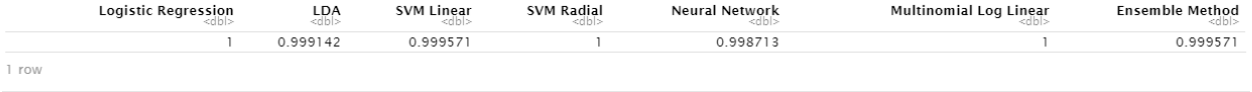
LDA <dbl>	SVM Linear <dbl>	SVM Radial <dbl>	Multinomial Log Linear <dbl>
0.951094	0.983269	0.960532	0.972115

1 row

5.
For the binary classifier, to improve the accuracy, I used the ensemble method, which composed of all algorithms used in the baseline algorithm: logistic regression, linear discriminant analysis,

SVM with linear kernel, SVM with radial kernel, neural network, and multinomial log linear model. I also hypertune the parameters for the SVM models.

Below is the accuracy results of the ensemble method and the baseline algorithms for binary:

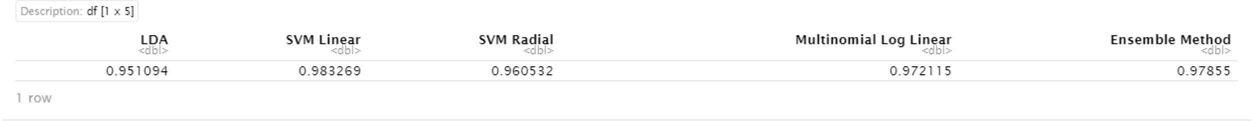


Logistic Regression	LDA	SVM Linear	SVM Radial	Neural Network	Multinomial Log Linear	Ensemble Method
<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
1	0.999142	0.999571	1	0.998713	1	0.999571

1 row

For the multiclass classifier, we simply used the ensemble method like in the binary classifier to improve our accuracy. We also used cross validation in feature selection and hypertung for more optimization.

Below is the accuracy results of the ensemble method and the baseline algorithms for multiclass:



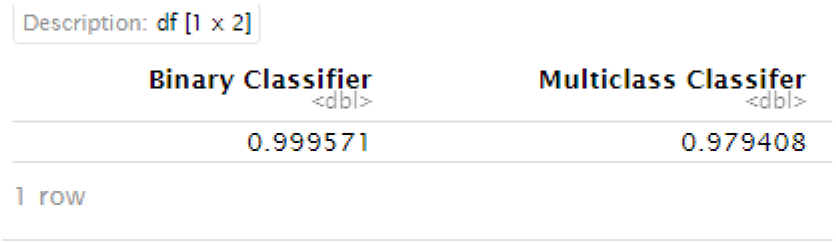
Description: df [1 x 5]

LDA	SVM Linear	SVM Radial	Multinomial Log Linear	Ensemble Method
<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
0.951094	0.983269	0.960532	0.972115	0.97855

1 row

6.

Here is the accuracy result of the models being used on the training data:

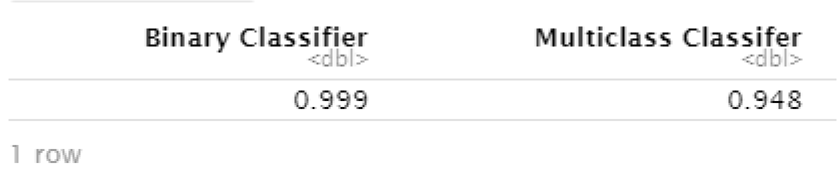


Description: df [1 x 2]

Binary Classifier	Multiclass Classifer
<dbl>	<dbl>
0.999571	0.979408

1 row

Here is the accuracy result of the models being used on the testing data:



Binary Classifier	Multiclass Classifer
<dbl>	<dbl>
0.999	0.948

1 row

While the performances of both binary classifiers are the same, we tried to improve the performance of the multiclass classifier by utilizing cross validation in feature selection and using the Boruta package. This is because the testing performance was lower than the training performance, which was due to overfitting. We tried to overcome this issue through cross validation. Furthermore, due to complexity caused by the multiclass data, we used the Boruta package to select our features.

7.
The multiclass model significantly required a large amount of time to select features for model training. We can possibly look at PCA over the Boruta package used. The PCA will reduce a large amount of dimensionality represented by the data. This will improve our accuracy.

Additionally, for model training, insteading of using hold one leave one out set, we can look into 5 or 10 fold cross validation. This can help give us a true accuracy of our model and prevent overfitting.

Lastly, we should also do more hypertuning of the parameters in our models. This will tailor our models to the data to get better accuracy.