Methods of Classification

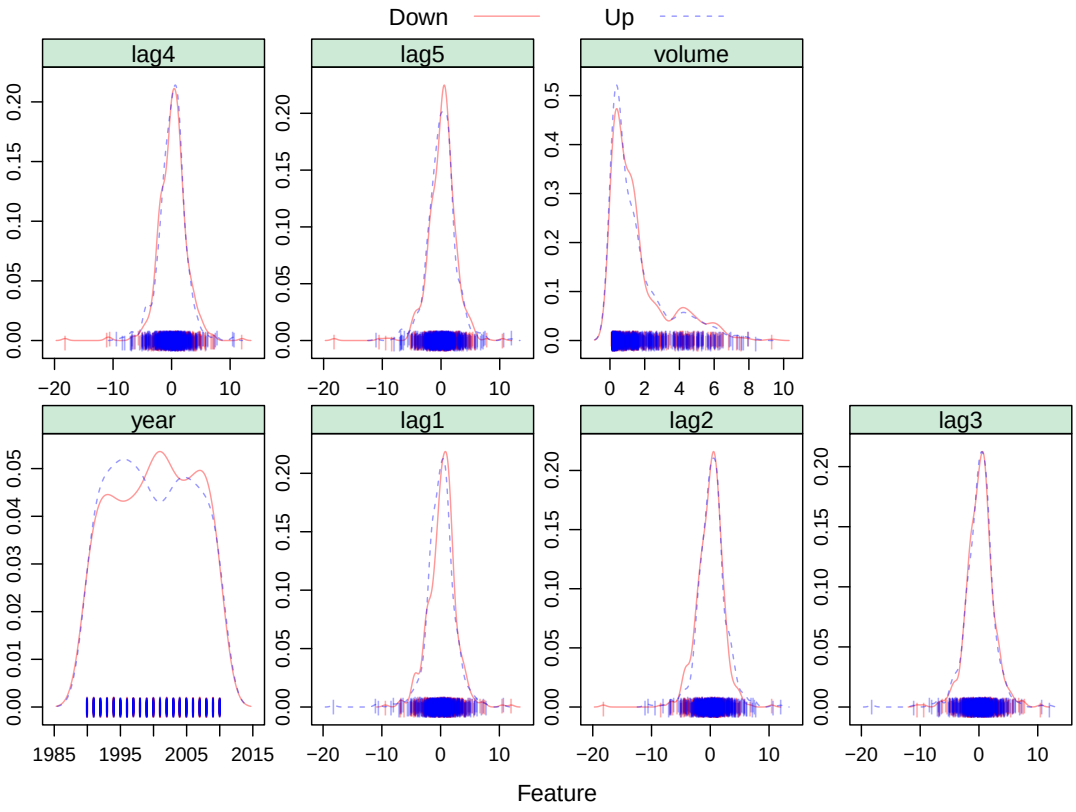
Hana Akbarnejad

4/10/2020

In this exercise, I have been working with *Weekly* dataset from *ISLR* package. The dataset contains data about Weekly percentage returns for the S&P 500 stock index between 1990 and 2010 and it has 1089 rowns and 8 columns. We consider a binary outcome (*direction*) with two levels: Up and Down, 5 *lag* variables (1 to 5), *year*, and *volume* as predictors.

Part (a)

In this part, I included a graphical summary of the *Weekly* data:



The above plots plot marginal density functions within each response class for each predictor. We can observe that Up and Down classes of response have similar denisty functions for each predictor and there are a lot of overlaps within blue and red curves. Also, we can see that lag variables are almost normally distributed and we can observe positive skewness in volume variable.

Part (b)

In this part, I used the full dataset to perform a logistic regression with *direction* as the response and the five *Lag* variables and *Volume* as predictors.

```
##
## Call:
## NULL
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.6949  -1.2565   0.9913   1.0849   1.4579
```

```
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.26686    0.08593   3.106   0.0019 **
## lag1        -0.04127    0.02641  -1.563   0.1181
## lag2         0.05844    0.02686   2.175   0.0296 *
## lag3        -0.01606    0.02666  -0.602   0.5469
## lag4        -0.02779    0.02646  -1.050   0.2937
## lag5        -0.01447    0.02638  -0.549   0.5833
## volume      -0.02274    0.03690  -0.616   0.5377
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1496.2  on 1088  degrees of freedom
## Residual deviance: 1486.4  on 1082  degrees of freedom
## AIC: 1500.4
##
## Number of Fisher Scoring iterations: 4
```

Using all datset, the summary of *Glm* model shows that *lag2* with p-value of 0.03 is the only significant variable at 95% significance level.

Part (c)

In this part, I will take a look at the confusion matrix and what it tells us about the model.

To do so, I have partitioned data into training and test (2/3-1/3), and refitted glm model on same variables as part (b) which is lag variables and volume.

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction Down  Up
##       Down    7  13
##       Up    154 188
##
##                Accuracy : 0.5387
##                  95% CI : (0.4858, 0.5909)
##     No Information Rate : 0.5552
##     P-Value [Acc > NIR] : 0.7544
##
##                   Kappa : -0.0232
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.93532
##             Specificity : 0.04348
##          Pos Pred Value : 0.54971
##          Neg Pred Value : 0.35000
##              Prevalence : 0.55525
##          Detection Rate : 0.51934
```

```
##    Detection Prevalence : 0.94475
##       Balanced Accuracy : 0.48940
##
##        'Positive' Class : Up
##
```

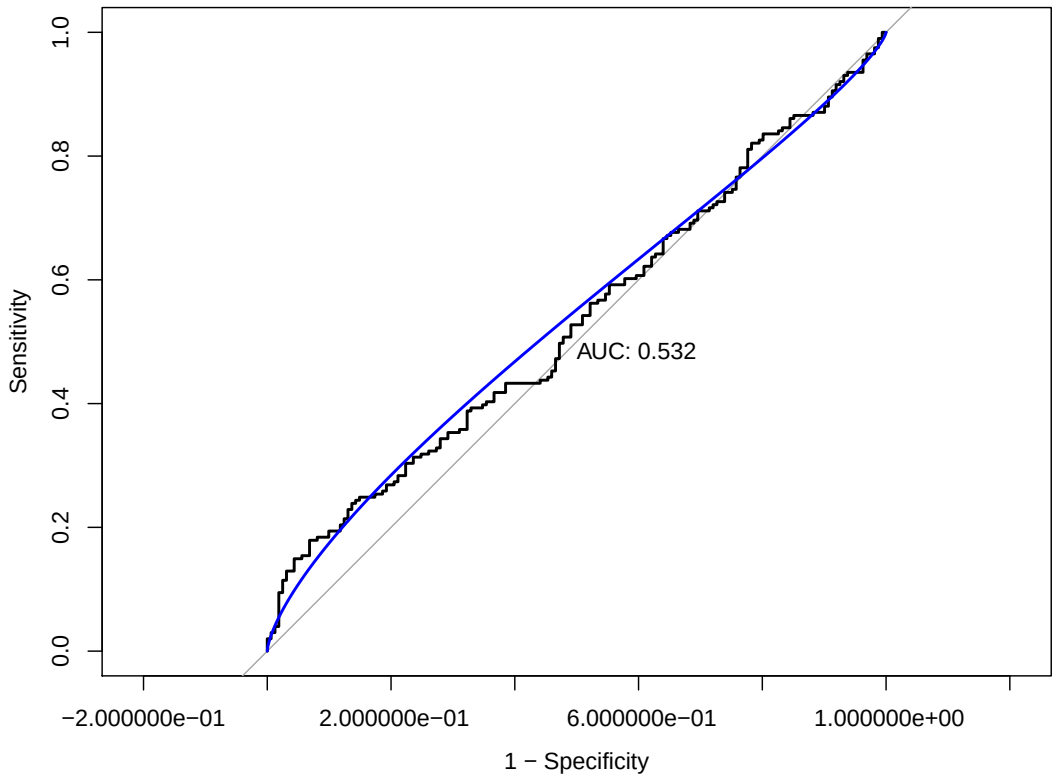
The confusion matrix provided above shows the accuracy of 0.54 which means that 54% of data (with 95% CI of (0.48, 0.59)) is correctly classified which is not very high and it shows that Glm model is probably not the best model to be used here. We also have No Information Rate of 0.55 which is a feature of data and shows the largest proportion (max) of each class. Here this number shows the the proportion of *Up* observations because most of the observations are in that class. We have a large p-value which shows that this prediction is not very meaningful and the accuracy is not significantly larger than the no information rate. Kappa value shows consistency of predicted and observed values. The Kappa value here is -0.023 which shows that these two values are not in agreement and that the agreement is even worse than random and the prediction is not meaningful. We observe a very high sensitivity(0.93) which means we do not have many false negatives and a very low specificity(0.04) which means we have many false-positives in prediction using this model. Also we have PPV of 0.55 and NPV of 0.35. These results show that the classification model is not performing perfectly and it might be useful if we can consider other methods of classification.

Part (d)

In this part, I have plotted the ROC curve using the predicted probability from logistic regression:

```
## Setting levels: control = Down, case = Up
```

```
## Setting direction: controls < cases
```



It can be observed that AUC is 0.5320911. We usually consider AUC of 0.8 or higher as good, so this value is pretty low and shows that the classification model is not performing very well.

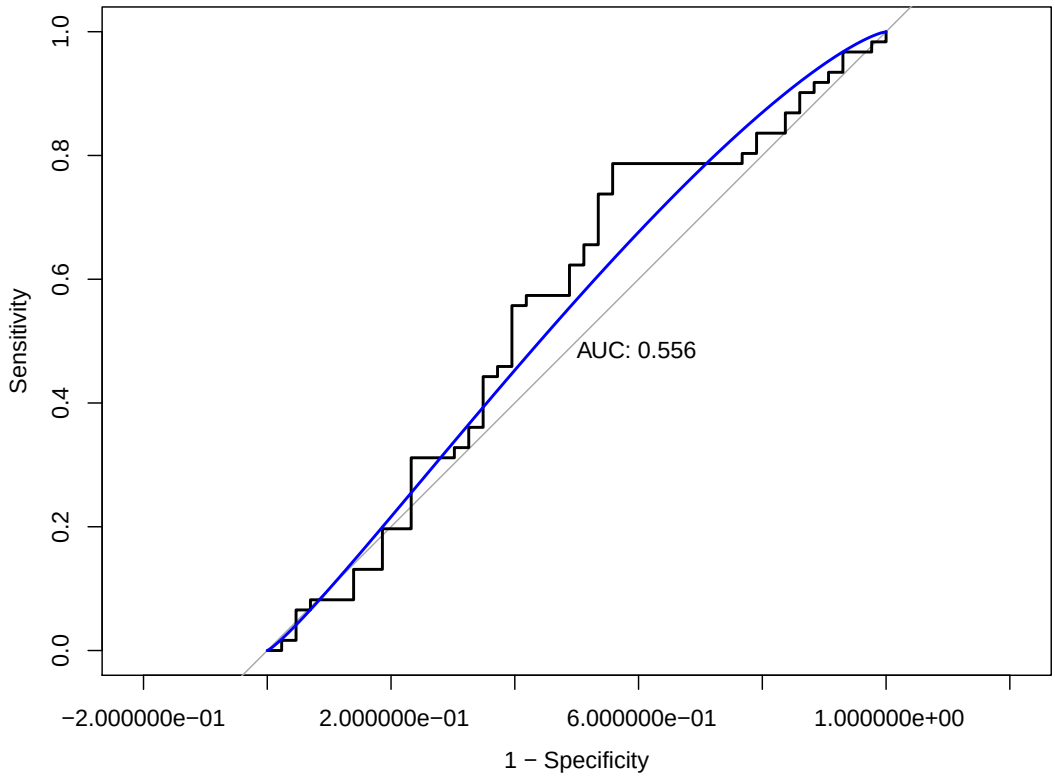
Part (e)

In this part, I have fitted logistic regression model again, but this time using a training data period from 1990 to 2008, with Lag1 and Lag2 as the predictors. Then I plotted the ROC curve using the held out data (that is, the data from 2009 and 2010).

```
##
## Call:
## glm(formula = direction ~ lag1 + lag2, family = binomial, data = train_year_data)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.6149  -1.2565   0.9989   1.0875   1.5330
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.21109    0.06456   3.269  0.00108 **
## lag1        -0.05421    0.02886  -1.878  0.06034 .
## lag2         0.05384    0.02905   1.854  0.06379 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1354.7  on 984  degrees of freedom
## Residual deviance: 1347.0  on 982  degrees of freedom
## AIC: 1353
##
## Number of Fisher Scoring iterations: 4

## Setting levels: control = Down, case = Up

## Setting direction: controls < cases
```

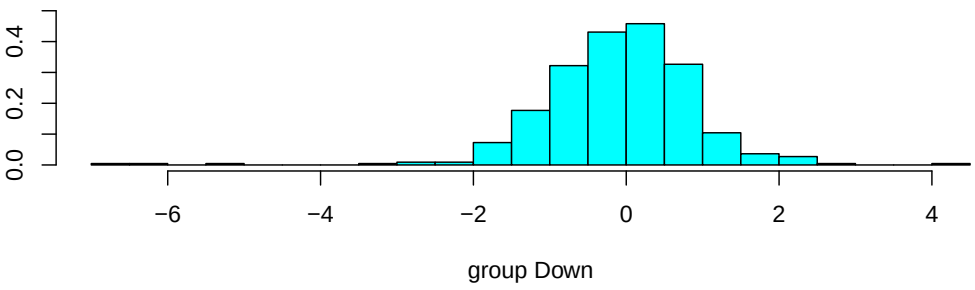


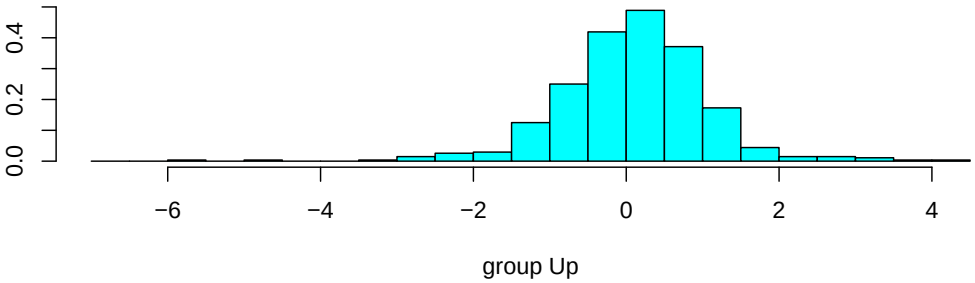
Doing so, it can be observed that neither lag1 or lag2 are significant anymore when fitting glm. Also the ROC curve above shows AUC value of 0.5558521, which is also not very good when predicting the direction od 2009 and 2010 using data from 1990 to 2008.

Part (f)

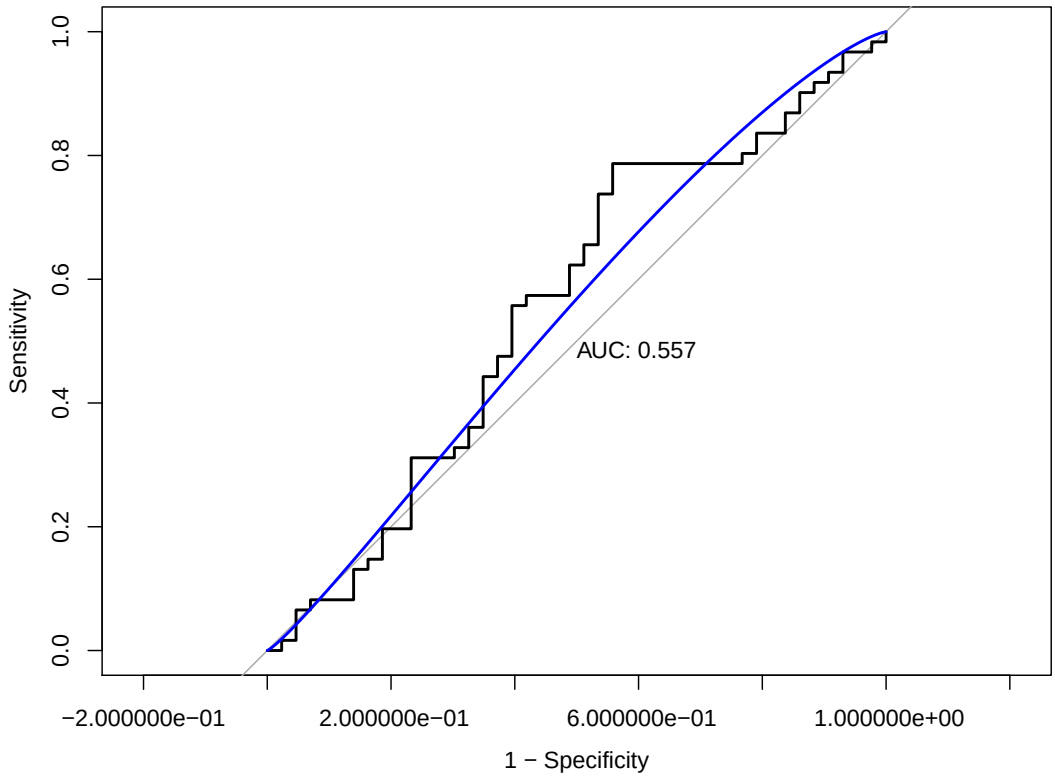
In this part, I repeat part (e) using LDA and QDA methods instead of Glm, still using lag1 and lah2 as predictors.

LDA





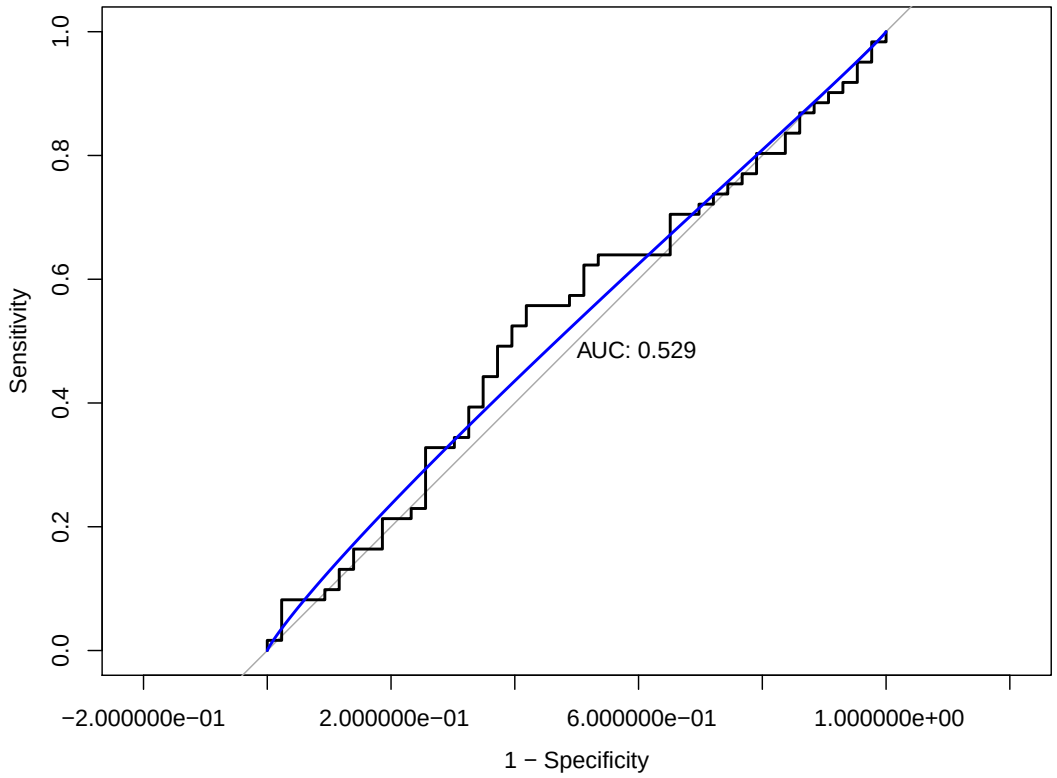

```
## Setting direction: controls < cases
```



As above ROC curve shows, we get AUC of 0.5566146 using LDA, which is pretty similar to AUC value using Glm method.

QDA

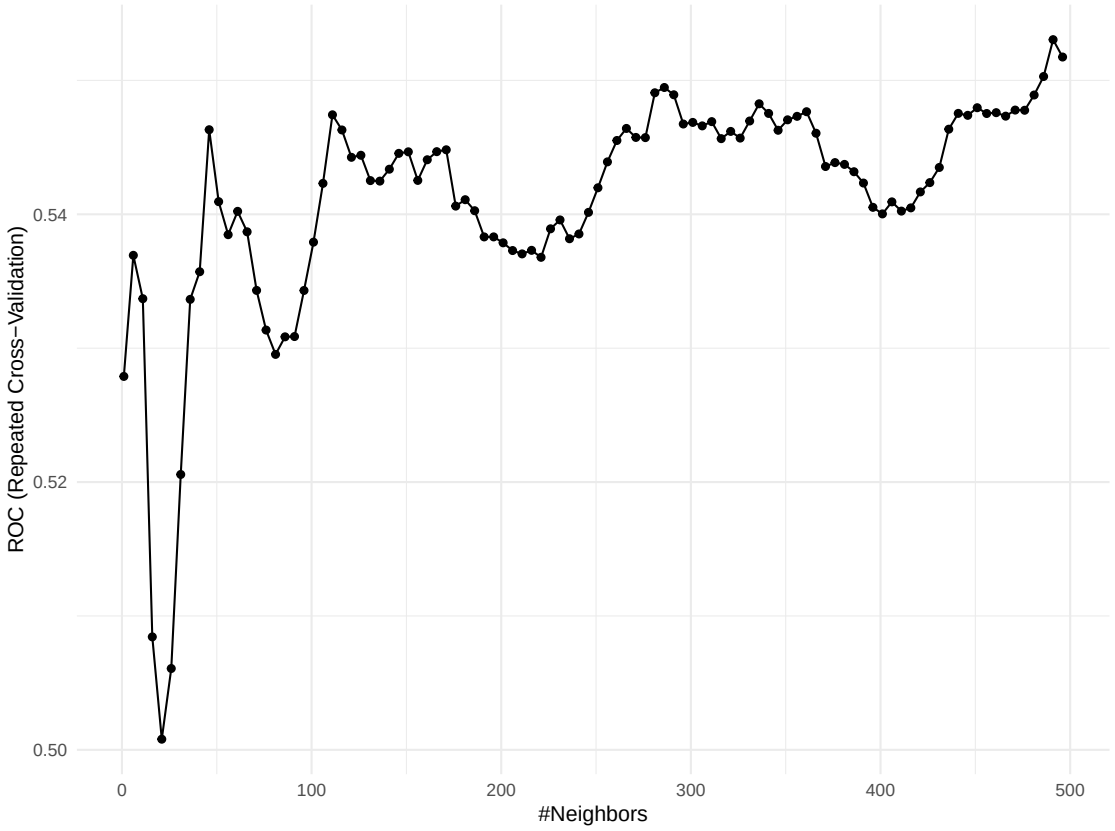
```
## Setting direction: controls > cases
```



As above ROC curve shows, we get the area under the curve of 0.5287838 using QDA, which is pretty similar to AUC value using LDA and Glm methods.

Part (g)

In this part, I repeated the procedure using KNN.

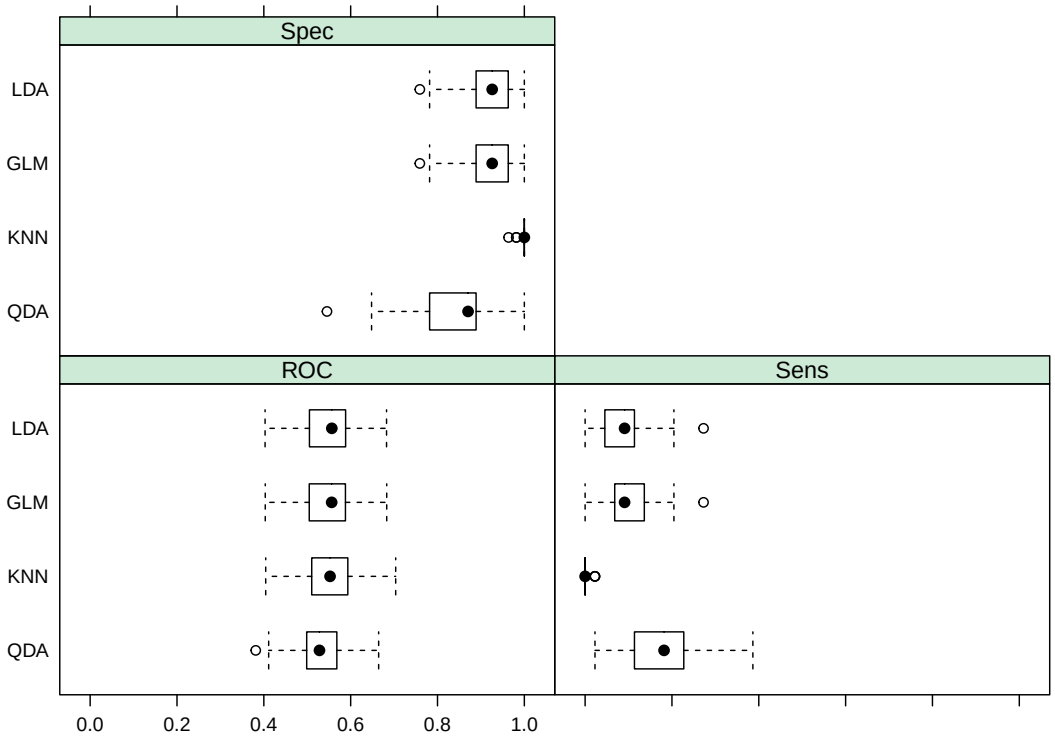


To find the best tuning parameter, I used the grid from 1 to 600, however, we can observe that the plot above does not reach a steady state or a decreasing effect. Theoretically, I should extend the grid till I get the desired shape in the plot but when extending the grid to beyond 500, I get the error regarding the presence of too many ties in knn which shows that the method cannot perform well for the values beyond 500. So, I have to stop at the maximum value that the method can run (500) which gives me the tuning parameter of 491.

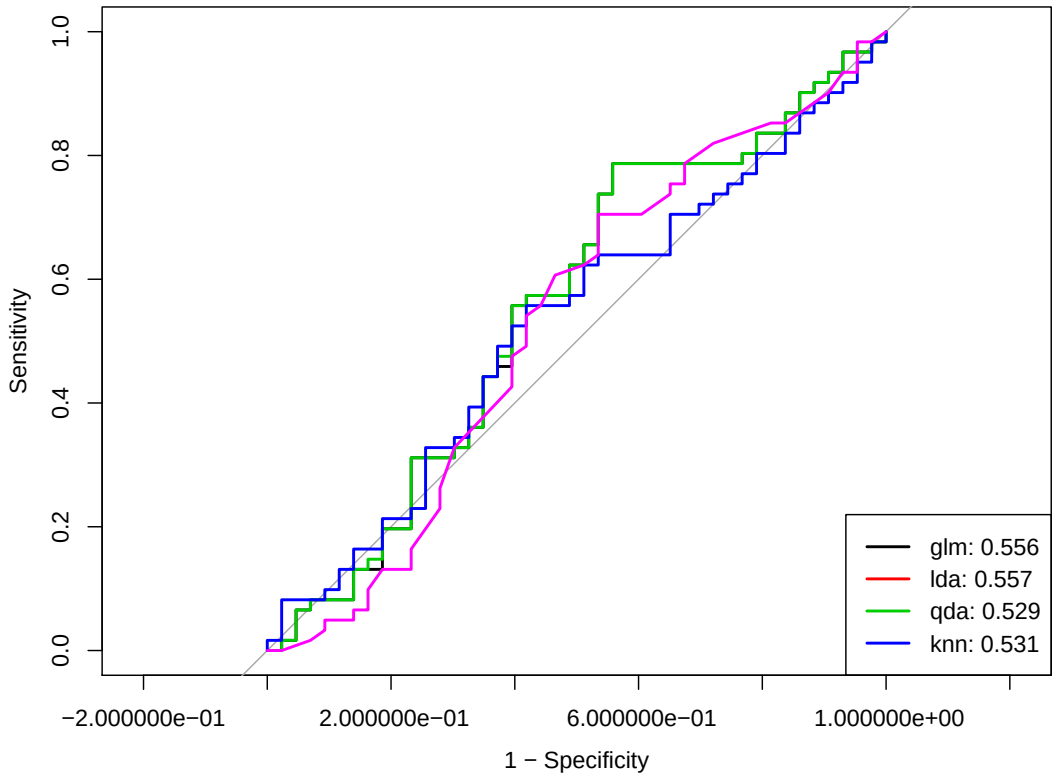
Model comparison and discussion

```
##
## Call:
## summary.resamples(object = res)
##
## Models: GLM, LDA, QDA, KNN
## Number of resamples: 50
##
## ROC
##          Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
## GLM 0.4032922 0.5048152 0.5562710 0.5451208 0.5873316 0.6830579    0
## LDA 0.4028807 0.5052342 0.5566077 0.5453792 0.5876473 0.6826446    0
## QDA 0.3813131 0.4993629 0.5281451 0.5274730 0.5676557 0.6644628    0
## KNN 0.4043210 0.5112058 0.5525253 0.5530453 0.5918216 0.7037190    0
##
## Sens
##           Min.    1st Qu.     Median        Mean   3rd Qu.       Max. NA's
## GLM 0.00000000 0.06818182 0.09090909 0.097505051 0.1306818 0.27272727    0
```

```
## LDA 0.00000000 0.05113636 0.09090909 0.095242424 0.1136364 0.27272727    0
## QDA 0.02272727 0.11931818 0.18181818 0.185525253 0.2272727 0.38636364    0
## KNN 0.00000000 0.00000000 0.00000000 0.001353535 0.0000000 0.02272727    0
##
## Spec
##          Min.   1st Qu.    Median      Mean   3rd Qu. Max. NA's
## GLM 0.7592593 0.8893939 0.9259259 0.9143704 0.9585859    1    0
## LDA 0.7592593 0.8893939 0.9259259 0.9154747 0.9585859    1    0
## QDA 0.5454545 0.7818182 0.8703704 0.8398653 0.8888889    1    0
## KNN 0.9636364 1.0000000 1.0000000 0.9985320 1.0000000    1    0
```



Comparing models built using *caret()* and training data, we can observe that **LDA** model has the best performance considering ROC value with the highest mean and median values. **GLM** model has ROC value which is very close to LDA value. However, the ROC values for other models do not have a considerable difference and are in a close range. We can conclude that the predictive ability of all these models are not very different from each other. The below ROC curve also compare performace of models using test data.


Sensitivity
1 − Specificity
0.0
0.2
0.4
0.6
0.8
1.0
−2.000000e−01
2.000000e−01
6.000000e−01
1.000000e+00
glm: 0.556
lda: 0.557
qda: 0.529
knn: 0.531