Comparison of various classifiers on hockey in-game prediction

Ender Erimhan

5/04/2021

Introduction

Consider a two-player time dependent game with two outcomes, Win or Loss. It can be shown that the probability distribution of the outcome changes as the game progresses. Depending on the circumstances, the outcome of the game becomes obvious near the end. As a game progresses, we can use various classification learning machines to predict the outcome before the end of the game. This paper will interest anyone who has a passion for data science with some level of interest in Game Theory. In the simple game of Tic-Tac-Toe, one can easily determine who has a better chance of winning after just a couple of moves. In a complex game like chess, one can develop a machine that inputs the board at the nth move and output a prediction of whether black or white might win. The accuracy of this machine being a function of n is obvious.

Consider the game of hockey. A basic understanding of the game is required from the reader and will not be explained here. We assume here that the game of hockey has 3 periods regardless of any overtime play. Another assumption this paper holds is that a team either wins or loses. A tie will indicate that both teams have lost. We will use the RIT men's hockey team as a point of reference. The RIT men's hockey website displays data of all past games in an unstructured format across multiple pages. The data shows the game location, the date and the number of goals scored in each period by each team. The data was collected into an Excel CSV file. We then imported the data into R. The first 5 lines is shown below:

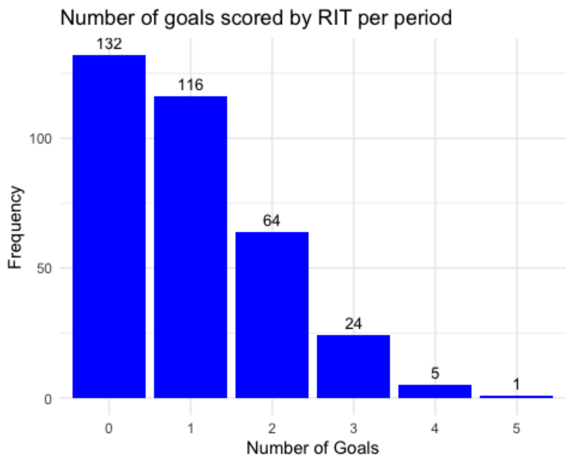
```
##       Date Location p1T p2T p3T p1G p2G p3G
## 1  10/1/16        H   1   1   1   0   0   0
## 2  10/7/16        H   1   3   2   0   2   1
## 3  10/8/16        H   3   2   0   1   4   1
## 4 10/15/16        H   0   0   1   0   0   1
## 5 10/21/16        A   0   1   1   0   2   3
```

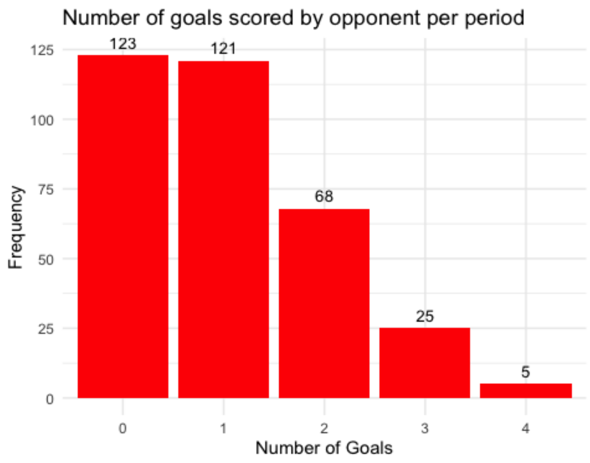
The Date column provides a unique identifier for each data point. Location is a binary variable that records H if the game was at home, A otherwise. The last 6 columns record the number of goals scored in each period by each team. The data was collected from the past 4 RIT hockey seasons. The first 3 seasons will be used to train our classification machines. The last season will be used to test our machine accuracy using ROC analysis.

Four classic classifiers will be used to make predictions. Logistic Regression, K-Nearest Neighbors, Linear Discriminant Analysis and Quadratic Discriminant Analysis will all be trained on three features. The classifiers will attempt to use Location, point-difference at period 1 and point-difference at period 2 to make predictions of the outcome at the end of the game.

Exploratory Data Analysis

We begin by exploring the different properties and main characteristics of our data. Let's look at the number of goals scored in any given period for each team. The results are in bar chart format below:





For any given period, a hockey team will most frequently score zero goals. About 38.6% of all periods in our training sample, RIT has scored zero points. Looking at the bar charts above, the number of goals scored per period seems to follow a Poisson Distribution. We test this using the chi-squared goodness of fit test with 95% confidence:

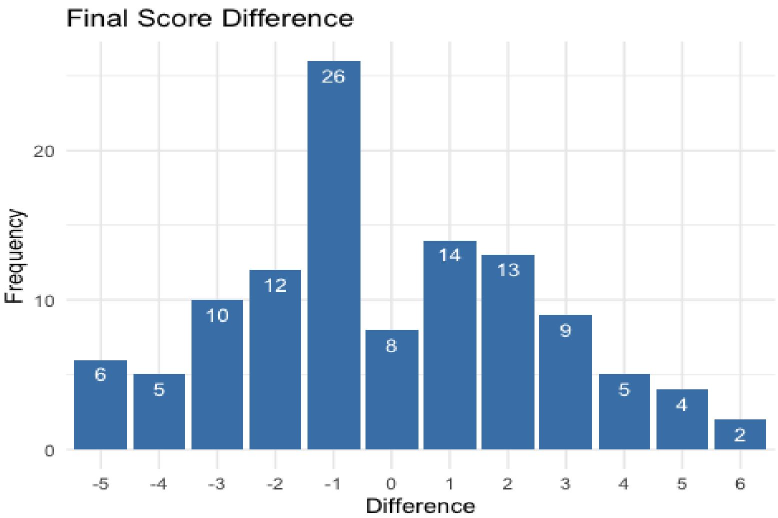
```
##  Chi-squared test for given probabilities with simulated p-value (based
##  on 2000 replicates)
##
## data:  c(tables1.df$Freq, 0)
## X-squared = 1.7456, df = NA, p-value = 0.9415


##  Chi-squared test for given probabilities with simulated p-value (based
##  on 2000 replicates)
##
## data:  c(tables2.df$Freq, 0)
## X-squared = 2.2049, df = NA, p-value = 0.8196
```

Since both p-values are greater than 0.05, we fail to reject the null hypothesis. That is, the number of goals scored per period do indeed follow a Poisson Distribution.

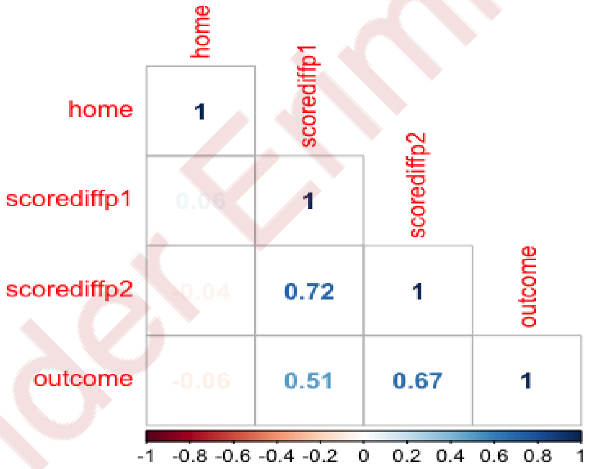
Using R, we cleaned up our data that we imported above in order to prepare it for our classifiers. The first five lines are shown below:

```
##   home scorediffp1 scorediffp2 scorediffFin outcome
## 1    1           1           2            3       1
## 2    1           1           2            3       1
## 3    1           2           0           -1       0
## 4    1           0           0            0       0
## 5    0           0          -1           -3       0
```

We transformed our home variable from a factor to a column in our data frame. 1 represents a home game, 0 otherwise. Scorediffp1, scorediffp2 and scorediffFin measure the number of points RIT is ahead by at period 1 ,2 and endgame respectively. Outcome is 1 for an RIT-win, 0 otherwise. Let’s take a look at the distribution of point-differences at the end of the game.

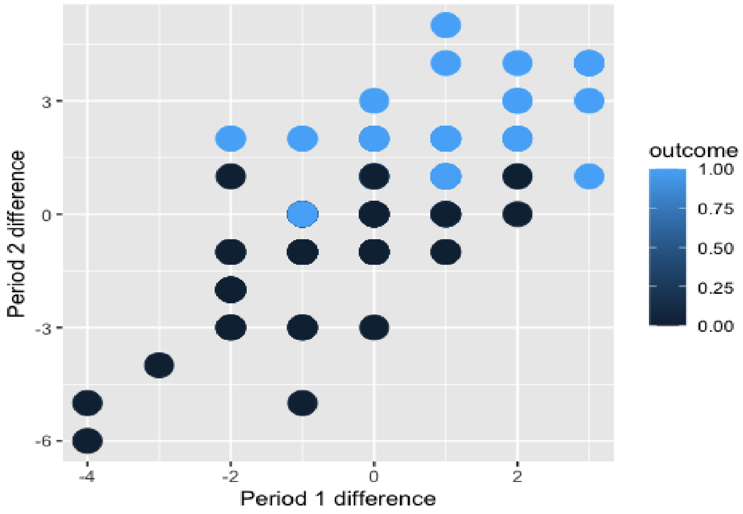


A few conclusions can be drawn from the chart above. RIT lost 67 games and won 47 games in our training sample therefore we have a relatively balanced data set. Final score difference is a difference of two Poisson distributions, thus it can be shown that it follows a Skellam Distribution. Since scorediffp1 and scorediffp2 have the same probabilistic structure as scorediffFin, these also theoretically follow the Skellam Distribution. Let's look at the correlations of the variables we will be training our models with:



We seem to have moderate correlations with outcome, scorediffp1 and scorediffp2. However there does not seem to indicate any high multi-collinearity. It is interesting to note that the point-difference in period 2 is correlated with the outcome higher than in period 1. This result is to be expected. It is surprising to note that home does not seem to correlate with any of the other variables. This will surprise any hockey enthusiast who will gladly tell you that the home team has a higher chance of winning in their own turf. The data doesn't seem to support this hypothesis.

Finally, let's look at a scatterplot of our moderately correlated variables:



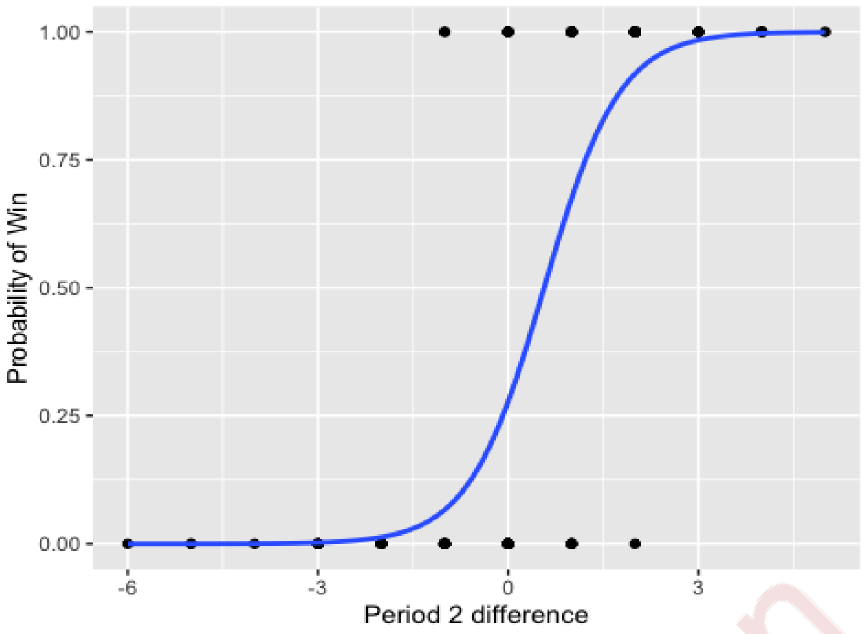
There seems to be a positive relationship between period 1 and period 2 point differences. The lower left section of the plot has a unanimous outcome of an RIT loss while the upper right section has the opposite effect. The middle has a mix of both wins and losses. The data does not seem to be well-separated however there does seem to be some degree of separation. With this knowledge we now explore our different classifiers.

Logistic regression

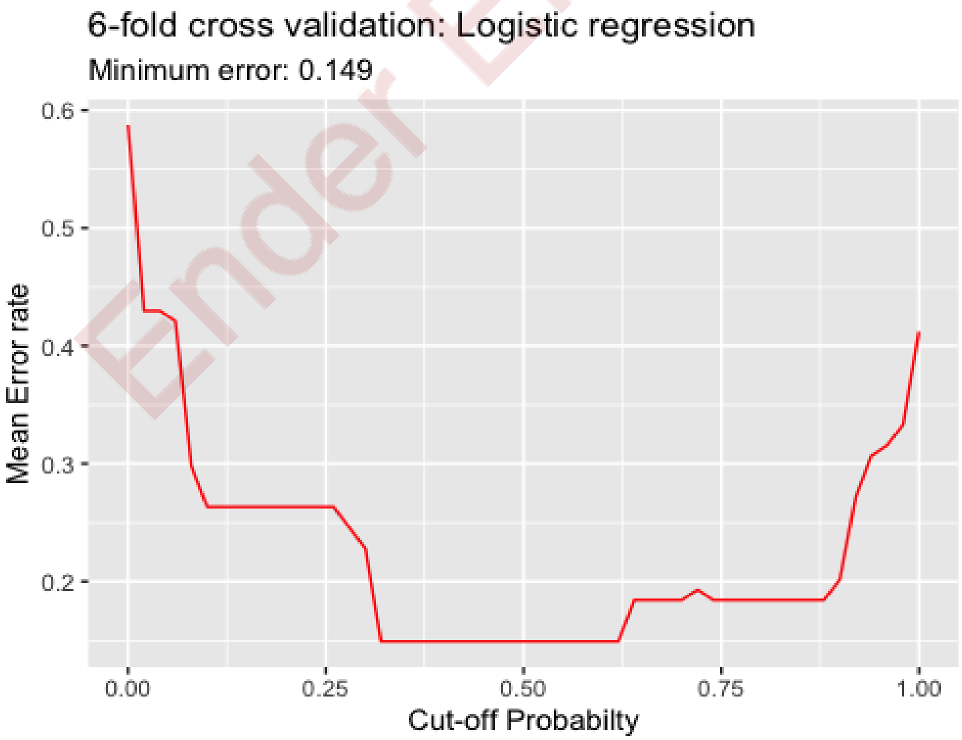
We first consider the simplest classifier, Logistic Regression. Let's use stepwise logistic regression to choose the best subset of features to train with:

```
##
## Call:  glm(formula = outcome ~ scorediffp2, family = binomial, data = myData2)
##
## Coefficients:
## (Intercept)  scorediffp2
##      -0.958        1.689
##
## Degrees of Freedom: 113 Total (i.e. Null);  112 Residual
## Null Deviance:       154.5
## Residual Deviance: 77.37     AIC: 81.37

## `geom_smooth()` using formula 'y ~ x'

##Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
##(Intercept)  -0.9580     0.3117  -3.074  0.00211 **
##scorediffp2   1.6892     0.3221   5.244 1.57e-07 ***
```

The only feature that survives our selection process is scorediffp2. This model seems to be significant with the feature being significant as well. A graph of the model is shown below:



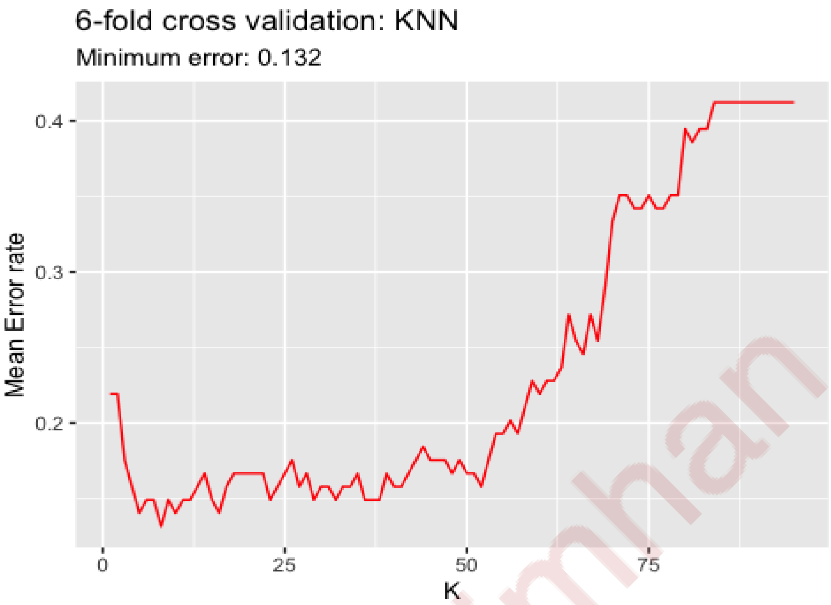
The model says that for every point-difference increase in period 2 , RIT has a 441.41% increase in odds of winning the game. If the game is tied at the end of period 2, RIT has a probability of 0.277 of winning the game. We now ask the question, what is the best cut-off probability value to predict an RIT win with a minimum error rate? We turn to cross-validation to answer this question. For all cross-validations in this paper, we randomly split our training sample into 6 folds. This means that we train on 95 data points and validate on 19 of them. We do this 6 times for each probability value and find the mean estimated error rate. The results are below:



The best cut-off value is 0.5 with a minimum estimated error rate of 0.149. We find that the best model here is a Logistic Regression model trained on scorediffp2 with a probability cutoff value of 0.50.

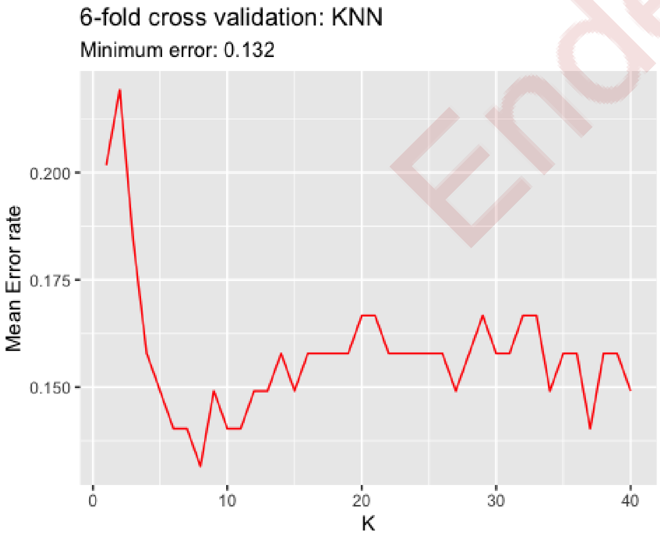
K-Nearest Neighbors

With some loss of interpretability, let's turn to our K-Nearest Neighbors classifier. Since KNN is a lazy learner, we choose to put all three features into our model. All three features have a similar range of discrete values so there is no need to standardize our data. In order to choose the best nearest neighbor K, we use cross-validation. Our candidates for K are all integers from 1 to 95 inclusive.

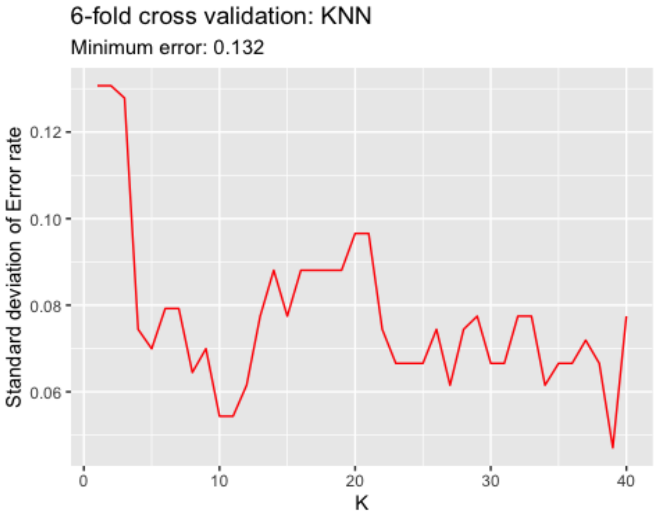

6-fold cross validation: KNN

Minimum error: 0.132

The mean error rate has a slight drop off from 1 to 6 nearest neighbors as bias of the machine decreases. The mean error rates stay relatively flat between 7 and 40. After around 40 nearest neighbors, the mean error rates slowly increase back up as the variance of the machine begins to increase. We cross-validate again for 1 to 40 as potential candidates. This time we take into account the standard deviation of the error rate as well.


6-fold cross validation: KNN

Minimum error: 0.132


6-fold cross validation: KNN

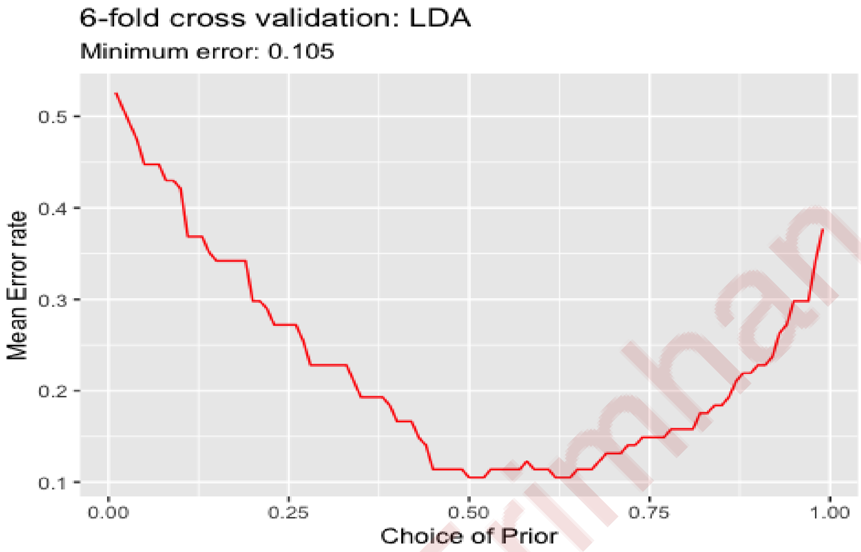
Minimum error: 0.132

We will use K = 10 as a nearest neighbor since both the mean estimated error rate and the standard deviation of the estimated error rates are low compared to the others. With a minimum training error of 0.132, we choose 10-Nearest Neighbors as our 2nd classifier.

Discriminant Analysis

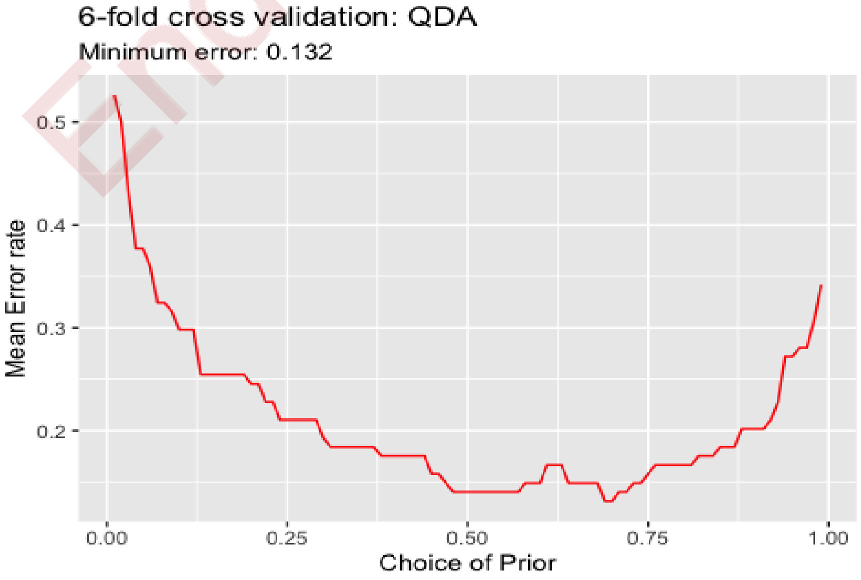
We now turn to another classic family of classifiers under the umbrella of Discriminant Analysis. Since we are dealing with discrete data, some of the assumptions of Discriminant Analysis are violated. However, Discriminant analysis can still perform well since it is quite robust to some violations of its assumptions. This paper gives both Linear and Quadratic discriminant analysis a try with home deleted as a feature.

We begin with Linear Discriminant analysis. This machine seeks to establish a linear decision boundary between the two classes. Our choice of prior class probabilities will depend on our cross-validation results:



Our 3rd classifier, with a minimum estimated training error of 0.105, is LDA with prior class probabilities of 0.57 and 0.43.

Quadratic discriminant analysis on the other hand seeks to establish a non-linear decision boundary between the two classes. Again, our choice of prior class probabilities will depend on our cross-validation results:

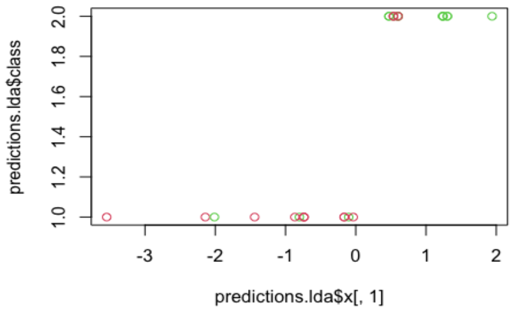


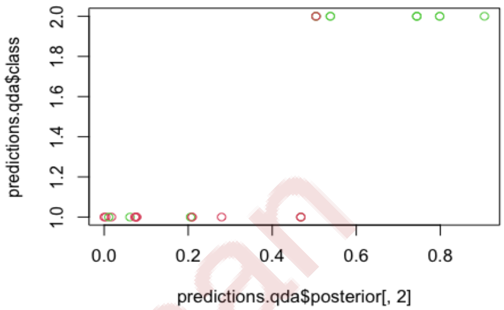
Our 4th classifier, with a minimum estimated training error of 0.132, is QDA with prior class probabilities of 0.67 and 0.33 respectively

ROC Analysis

We will now consider the test set which consistents of games from the 2019 - 2020 hockey season. Using our four optimal classifiers, we will compare and contrast some of the metrics from the confusion matrices produced by the classifiers.

Consider LDA and QDA, below are visuals of the confusion matrices produced when applying it to our training data

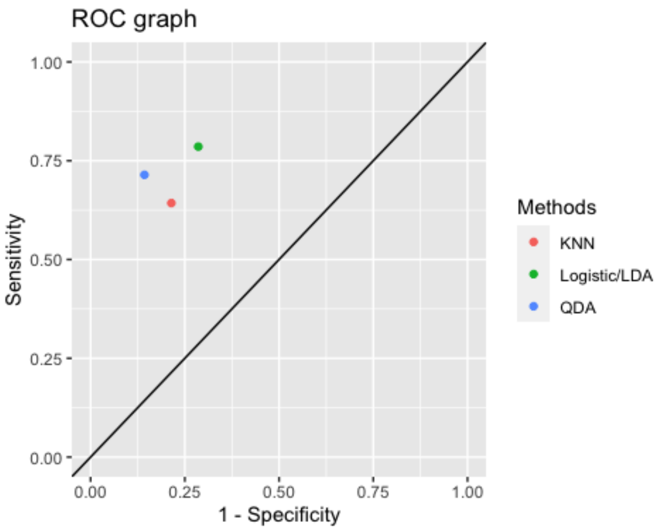




The predictions have been colored with the actual classification. The mixing of the red and green colors show that there is some level of incorrect classification. Lets compare the Accuracy, Sensisitivity and Specificty of our methods applied to the test data:

```
##    Methods  Accuracy Sensitivity Specificity
## 1 Logistic 0.7500000   0.7857143   0.7142857
## 2      KNN 0.7142857   0.6428571   0.7857143
## 3      LDA 0.7500000   0.7857143   0.7142857
## 4      QDA 0.7857143   0.7142857   0.8571429
```

LDA and Logistic Regression have the same confusion matrix and thus have the same prediction metrics. Surprisingly, QDA is our most accurate machine but has a lower true win rate than LDA. QDA does really well with true loss rates. 10-nearest neighbors tends to underperform and has the lowest accuracy of the classifiers. Let's look at the following ROC graph to visualize the sensitivity and specificity better.



All 4 classifiers do much better than random guessing which is on the black diagonal line. The closer towards the upper left corner, the better. 10-nearest neighbors is the closest towards the black line. Logistic regression, Linear discriminant analysis and quadratic discriminant analysis seem to have similar distances away from the black line. For accuracy, it seems that QDA is the way to go but if one wants to predict wins better, LDA and logistic regression are the way to go. If one wants a simple interpretable model with some loss of predictability, Logistic Regression is king.

Conclusion

Any good statistician that practices the art of statistical machine learning needs to be aware and be skillful in a few things. Being skillful in the programming language of R is a must since it provides enough flexibility to play around with the data and the models. The statistician can simply do more compared to more rigid computer programs like Minitab. However, R takes a long time to master and some of the code in this project took a few tries to get right. Especially the use of the prediction function, which is very picky on what the test data frame being fed in should look like. Knowing the difference between factors and vectors is also a must.

It is also important to be aware of one important lesson. That is the principle of Occam`s razor. Sometimes simpler models are better. When given a task, any good practitioner should always start with the simplest of models. If need be, then they can move to more complex models. Finally, a model is only as good as the data. One of the pitfalls this project had was that early on, the models were being fed data that was not well understood. This is the primary reason why power plays were left out from all the models. As a wise professor once said, “Make sense of your data before anything else”.