Hockey Odds Challenge

Sean Hellingman

2023-07-31

Ice Hockey

Introduction

After some preliminary examination and the exclusion of some observations in the data five total models were estimated for each of the tasks using all of the remaining observations. The models were validated using ten-fold cross validation. Most of the code is left in this report as this is a coding assessment but normally reports like this would be much shorter.

```
Hockey <- read_csv("Icehockey_OU_data_3000.csv")
Hockey$Goals <- Hockey$TotScore_T1 + Hockey$TotScore_T2
Hockey$O4_5 <- ifelse(Hockey$Goals > 4.5,1,0)
Hockey$O6_5 <- ifelse(Hockey$Goals > 6.5,1,0)
```

A

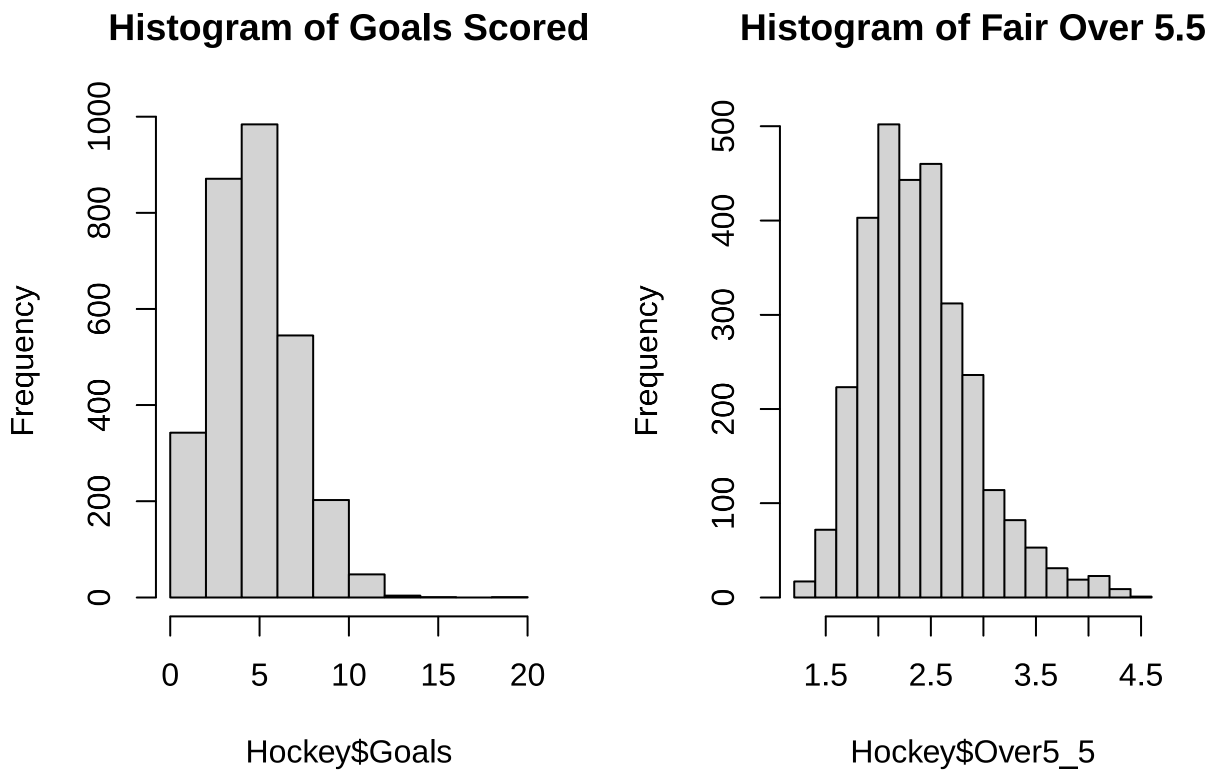
Where could you imagine errors in the data? So what would you check before using it?

It is immediately apparent that the O/U 4.5 odds are missing many observations and will be re-estimated separately from the O/U 4.5 odds given in the dataset. Furthermore, there may be collection errors in the actual scores of the matches. This shouldn't be the case due to the availability of match scores. There are also some extreme values in the the O/U 5.5 odds, number of goals, and Tipp 1X2 values. There may also be some issues in the calculation of the O/U 5.5 odds as there is little information as to how they were obtained.

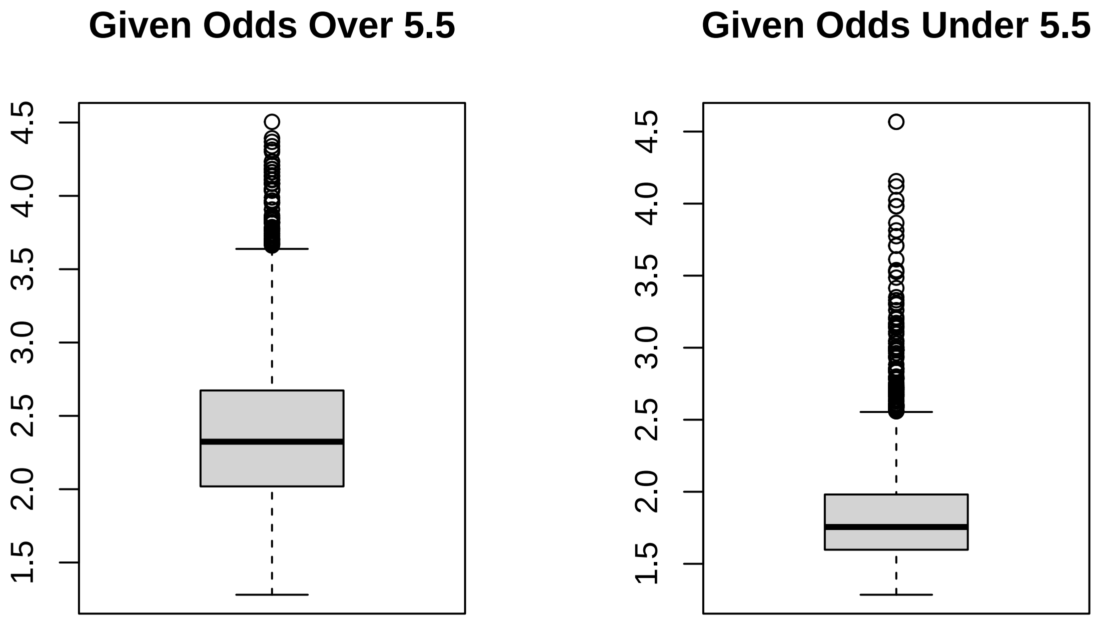
Some other problems may arise from the sparsity of data for specific leagues. A more in-depth analysis into which leagues could be combined and the variability of the number of goals within specific leagues. In this analysis, the *Category* variable will be used to identify which nation the match was played in. Further issues arise in the availability of information about the teams themselves. The estimated fair odds for the result of each match provides information about which team is favoured but does not give any information about the historical scoring and defending abilities of either team. Assuming the O/U 5.5 odds are estimated using more detailed information this variable will have the most predictive power of all the available information. I do not anticipate very accurate predictions based on the available information.

```
Hockey$P5_5O <- 1/Hockey$Over5_5
Hockey$P5_5U <- 1/Hockey$Under5_5

Hockey$T5_5 <- Hockey$P5_5O + Hockey$P5_5U #They are all approximately 1
```

```r
par(mfrow=c(1,2))

hist(Hockey$Goals,main = "Histogram of Goals Scored")
hist(Hockey$Over5_5,main = "Histogram of Fair Over 5.5")
```



```r
boxplot(Hockey$Over5_5,main = "Given Odds Over 5.5")
boxplot(Hockey$Under5_5,main = "Given Odds Under 5.5")
```



Extreme values were removed from the analysis. Other possible areas of exclusion could be women's matches, youth matches, or friendlies. These games were left in the analysis due to lack of informative variables but with more detailed information these competitions should be looked at separately.

```
#Matches over 11 goals
Hockey$TopGoals <- ifelse(Hockey$Goals > quantile(Hockey$Goals,.99), "Yes" , "No")

#Over 4.036668
Hockey$TopOver5_5 <- ifelse(Hockey$Over5_5 > quantile(Hockey$Over5_5,.99), "Yes" , "No")

#Over 7.727448
Hockey$TopTipp1 <- ifelse(Hockey$Tipp1 > quantile(Hockey$Tipp1,.99), "Yes" , "No")

#2917 observations remaining
Hockey <- subset(Hockey,Hockey$TopGoals == "No" & Hockey$TopOver5_5 == "No"
                 & Hockey$TopTipp1 == "No")
```

B

How would you derive Over/Under 4.5 and 6.5 odds out of this data?

The following models are focused on the binary outcome '1' if the total match goals are over 4.5 and over 6.5 and a '0' if they are not over either of those totals. Five total methods are used in this stage of the analysis. Logisitic regression, logistic regression with random intercepts, linear discriminant analysis (LDA), quadratic discriminant analysis (QDA), and a binary neural network. The estimates for the logistic regression models are shown below. It should be noted that no attempt was made to optimize the size or decay rates of the neural network models and that there may be more optimal models in existence.

Category, Tipp1, Tipp2, and Over5_5 are chosen as the explanatory variables. TippX and Under5_5 are excluded as their inclusion would create very high levels of colinearity. Furthermore, colinearity may exist between Over5_5 and the other included explanatory variables.

```
##Logistic Regression##

logit4_5 <- glm(O4_5 ~ Category + Tipp1 + Tipp2 + Over5_5,
                family = binomial,
               data = Hockey)
summary(logit4_5)
```

```
##
## Call:
## glm(formula = O4_5 ~ Category + Tipp1 + Tipp2 + Over5_5, family = binomial,
##     data = Hockey)
##
## Coefficients:
##                         Estimate Std. Error z value Pr(>|z|)
## (Intercept)            1.6140173  0.3658744   4.411 1.03e-05 ***
## CategoryDeutschland   -0.1983474  0.2404195  -0.825 0.409369
## CategoryFinnland      -0.2771928  0.2322178  -1.194 0.232605
## CategoryInternational -0.5517123  0.2483522  -2.221 0.026318 *
## CategoryNorwegen       0.1744415  0.2709842   0.644 0.519749
## CategoryÖsterreich     0.2469804  0.2787439   0.886 0.375592
## CategoryPolen         -0.3317980  0.4965899  -0.668 0.504036
## CategoryRussland      -0.1996459  0.2246616  -0.889 0.374190
## CategorySchweden      -0.1901836  0.2345744  -0.811 0.417503
## CategorySchweiz       -0.0351475  0.3084326  -0.114 0.909273
## CategorySlowakei       0.0522351  0.3882411   0.135 0.892973
## CategoryTschechien    -0.2369063  0.2379852  -0.995 0.319510
## CategoryUSA            0.0786172  0.2740753   0.287 0.774231
## CategoryWeißrussland  -0.3282565  0.4182406  -0.785 0.432541
## Tipp1                 -0.0529880  0.0485066  -1.092 0.274663
## Tipp2                  0.0009545  0.0201220   0.047 0.962165
## Over5_5               -0.3924817  0.1023581  -3.834 0.000126 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 3940.6  on 2916  degrees of freedom
## Residual deviance: 3873.3  on 2900  degrees of freedom
## AIC: 3907.3
##
## Number of Fisher Scoring iterations: 4
```

```
logit6_5 <- glm(O6_5 ~ Tipp1 + Tipp2 + Category +  Over5_5,
                family = binomial,
               data = Hockey)
summary(logit6_5)
```

```
##
## Call:
## glm(formula = O6_5 ~ Tipp1 + Tipp2 + Category + Over5_5, family = binomial,
##     data = Hockey)
##
## Coefficients:
##                         Estimate Std. Error z value Pr(>|z|)
## (Intercept)            0.4217599  0.4071338   1.036    0.300
## Tipp1                 -0.0007561  0.0520981  -0.015    0.988
## Tipp2                  0.0077344  0.0202282   0.382    0.702
## CategoryDeutschland    0.0367777  0.2524908   0.146    0.884
## CategoryFinnland      -0.2034415  0.2541134  -0.801    0.423
## CategoryInternational -0.0654719  0.2696014  -0.243    0.808
## CategoryNorwegen       0.0888630  0.2718253   0.327    0.744
## CategoryÖsterreich     0.1522375  0.2772798   0.549    0.583
## CategoryPolen          0.1564190  0.5032348   0.311    0.756
## CategoryRussland      -0.0029084  0.2401887  -0.012    0.990
## CategorySchweden       0.0303754  0.2538473   0.120    0.905
## CategorySchweiz        0.1169144  0.3203311   0.365    0.715
## CategorySlowakei      -0.2930254  0.4351732  -0.673    0.501
## CategoryTschechien    -0.0899003  0.2582311  -0.348    0.728
## CategoryUSA            0.1449624  0.2834706   0.511    0.609
## CategoryWeißrussland  -0.0181854  0.4513465  -0.040    0.968
## Over5_5               -0.6320963  0.1256362  -5.031 4.88e-07 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 3350.7  on 2916  degrees of freedom
## Residual deviance: 3282.1  on 2900  degrees of freedom
## AIC: 3316.1
##
## Number of Fisher Scoring iterations: 4
```

```
##ICC ESTIMATES Binary## #Not neccessary
#iccbin(Category, O6_5, data = Hockey, method = "aov",
#       ci.type = "aov", alpha = 0.05,
#       kappa = 0.45, nAGQ = 1, M = 1000)

#iccbin(Category, O4_5, data = Hockey, method = "fc",
#       ci.type = "fc", alpha = 0.05,
#       kappa = 0.45, nAGQ = 1, M = 1000)


logitRS4_5 <- glmer(O4_5 ~ Tipp1 + Tipp2 + Over5_5
              + (1| Category) ,
              family = binomial,  data=Hockey)
summary(logitRS4_5)
```

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: binomial  ( logit )
## Formula: O4_5 ~ Tipp1 + Tipp2 + Over5_5 + (1 | Category)
##    Data: Hockey
##
##      AIC      BIC   logLik deviance df.resid
##   3902.4   3932.3  -1946.2   3892.4     2912
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -1.6522 -1.1543  0.7161  0.8254  1.3209
##
## Random effects:
##  Groups   Name        Variance Std.Dev.
##  Category (Intercept) 0.002291 0.04786
## Number of obs: 2917, groups:  Category, 14
##
## Fixed effects:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  1.843405   0.307756   5.990 2.10e-09 ***
## Tipp1       -0.067155   0.047454  -1.415    0.157
## Tipp2       -0.009124   0.019810  -0.461    0.645
## Over5_5     -0.529900   0.097459  -5.437 5.41e-08 ***
## ---
## Signif. codes:  0 ’***’ 0.001 ’**’ 0.01 ’*’ 0.05 ’.’ 0.1 ’ ’ 1
##
## Correlation of Fixed Effects:
##         (Intr) Tipp1  Tipp2
## Tipp1   -0.612
## Tipp2   -0.638  0.574
## Over5_5 -0.836  0.149  0.261
## optimizer (Nelder_Mead) convergence code: 0 (OK)
## Model failed to converge with max|grad| = 0.00225798 (tol = 0.002, component 1)
```

```
logitRS6_5 <- glmer(O6_5 ~ Tipp1 + Tipp2 + Over5_5
               + (1| Category) ,
               family = binomial,  data=Hockey)
```

```
## boundary (singular) fit: see help(’isSingular’)
```

```
summary(logitRS6_5)
```

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: binomial  ( logit )
## Formula: O6_5 ~ Tipp1 + Tipp2 + Over5_5 + (1 | Category)
##    Data: Hockey
##
##      AIC      BIC   logLik deviance df.resid
##   3297.2   3327.1  -1643.6   3287.2     2912
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -0.8916 -0.6280 -0.5403  1.2832  3.1362
##
## Random effects:
```

```
##  Groups   Name        Variance  Std.Dev.
##  Category (Intercept) 3.648e-17 6.04e-09
## Number of obs: 2917, groups:  Category, 14
##
## Fixed effects:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.651060   0.301590   2.159   0.0309 *
## Tipp1       -0.002679   0.050714  -0.053   0.9579
## Tipp2        0.004363   0.019421   0.225   0.8222
## Over5_5     -0.727586   0.096469  -7.542 4.62e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##         (Intr) Tipp1  Tipp2
## Tipp1   -0.591
## Tipp2   -0.594  0.540
## Over5_5 -0.809  0.076  0.174
## optimizer (Nelder_Mead) convergence code: 0 (OK)
## boundary (singular) fit: see help('isSingular')
```

```r
#Random Intercepts are not useful as variance is very small

##Simple Binary Classifiers LDA/QDA##

lda4_5 <- lda(O4_5 ~ Category + Tipp1 + Tipp2 + Over5_5,
              data = Hockey)

lda6_5 <- lda(O6_5 ~ Category + Tipp1 + Tipp2 + Over5_5,
              data = Hockey)

qda4_5 <- qda(O4_5 ~ Category + Tipp1 + Tipp2 + Over5_5,
              data = Hockey)

qda6_5 <- qda(O6_5 ~ Category + Tipp1 + Tipp2 + Over5_5,
              data = Hockey)


##Neural Network##

#size = Number of hidden units
#decay is tuning parameter

#Not optimized!!

nnet4_5 <- nnet(O4_5 ~ Category + Tipp1 + Tipp2 + Over5_5,
                data = Hockey,
                size=2,decay=1.0e-2,maxit=1000,trace = FALSE)

nnet6_5 <- nnet(O6_5 ~ Category + Tipp1 + Tipp2 + Over5_5,
                data = Hockey,
                size=2,decay=1.0e-2,maxit=1000,trace = FALSE)
```

As expected, the O/U 5.5 goals variable accounts for most of the variability in the models. The international category has a small amount of significance in the logistic regression model focusing on O/U 4.5. The second source of variability assumption caused by the category variable does not appear to hold. All models will be tested for the predictive capabilities.

C

Please demonstrate the quality of your results.

In order to demonstrate the quality of the results ten-fold cross validation was conducted on each of the models focusing on predicted probabilities. Four measures of accuracy were used. The percentage of overall accuracy (Accuracy), the Kappa value, the mean absolute error (MAE), and root mean squared error (RMSE) were used. Higher accuracy and Kappa values are desired while smaller values for mean absolute error (MAE) and root mean squared error (RMSE) are desired.

```
set.seed(14)


CVHockey <- Hockey
#shuffle
CVHockey<-CVHockey[sample(nrow(CVHockey)),]
#Create 10 equally size folds
folds <- cut(seq(1,nrow(CVHockey)),breaks=10,labels=FALSE)

Prediction.Capability <- data.frame(matrix(ncol = 6, nrow = 0 )) #Dataframe for results
x <- c("Model","Test_Number", "Accuracy","Kappa","MAE","RMSE") #col names
colnames(Prediction.Capability) <- x
rm(x)
PCLogit <- Prediction.Capability
PCRS <- Prediction.Capability
PCLDA <- Prediction.Capability
PCQDA <- Prediction.Capability
PCNN <- Prediction.Capability

#Perform 10 fold cross validation
for(i in 1:10){
  #Segement your data by fold using the which() function
  testIndexes <- which(folds==i,arr.ind=TRUE)
  testData <- CVHockey[testIndexes, ]
  trainData <- na.omit(CVHockey[-testIndexes, ])



  logit4_5 <- glm(O4_5 ~ Category + Tipp1 + Tipp2 + Over5_5,
                family = binomial,
               data = trainData)

  RSlogit4_5 <- glmer(O4_5 ~ Tipp1 + Tipp2 + Over5_5
               + (1| Category) ,nAGQ=0,
               family = binomial,  data=trainData)

  lda4_5 <- lda(O4_5 ~ Category + Tipp1 + Tipp2 + Over5_5,
              data = trainData)

  qda4_5 <- qda(O4_5 ~ Category + Tipp1 + Tipp2 + Over5_5,
              data = trainData)

  nnet4_5 <- nnet(O4_5 ~ Category + Tipp1 + Tipp2 + Over5_5,
                data = trainData,
                size=2,decay=1.0e-2,maxit=1000,trace = FALSE)




  #LOGIT#
  # Make predictions and compute Accuracy, Kappa, MAE, and RMSE #
  predictions <- logit4_5 %>% predict(testData, type = "response")
  predictions <- as.data.frame(predictions)
  names(predictions)[1] <- 'Pred'
  mae <- MAE(predictions$Pred, as.numeric(testData$O4_5))
  rmse <- RMSE(predictions$Pred, as.numeric(testData$O4_5))

  predictions <- as.data.frame(ifelse(predictions > 0.5,1,0))
  predictions$Pred <- factor(predictions$Pred)
  testData$O4_5 <- factor(testData$O4_5)
  CON <- confusionMatrix(predictions$Pred, testData$O4_5)
  CO <- as.data.frame(CON$overall)


  #assign(paste("logit4_5",i, sep=""), logit4_5) #Save each model
  PCLogit[1,] <- c("logit4_5",i,CO[1,],CO[2,],mae,rmse)


  #LOGITRS#
  # Make predictions and compute Accuracy, Kappa, MAE, and RMSE #
  predictions <- RSlogit4_5 %>% predict(testData, type = "response")
  predictions <- as.data.frame(predictions)
  names(predictions)[1] <- 'Pred'
  mae <- MAE(predictions$Pred, as.numeric(testData$O4_5))
  rmse <- RMSE(predictions$Pred, as.numeric(testData$O4_5))

  predictions <- as.data.frame(ifelse(predictions > 0.5,1,0))
  predictions$Pred <- factor(predictions$Pred)
  testData$O4_5 <- factor(testData$O4_5)
  CON <- confusionMatrix(predictions$Pred, testData$O4_5)
  CO <- as.data.frame(CON$overall)


  #assign(paste("logit4_5",i, sep=""), logit4_5) #Save each model
  PCRS[1,] <- c("RSlogit4_5",i,CO[1,],CO[2,],mae,rmse)




  #LDA#
  # Make predictions and compute Accuracy, Kappa, MAE, and RMSE #
  predictions <- lda4_5 %>% predict(testData, type = "response")
  predictions <- as.data.frame(predictions)
```

```r
names(predictions)[1] <- 'Pred'
mae <- MAE(predictions$posterior.1, as.numeric(testData$04_5))
rmse <- RMSE(predictions$posterior.1, as.numeric(testData$04_5))

#predictions <- as.data.frame(ifelse(predictions > 0.5,1,0))
#predictions$Pred <- factor(predictions$Pred)
testData$04_5 <- factor(testData$04_5)
CON <- confusionMatrix(predictions$Pred, testData$04_5)
CO <- as.data.frame(CON$overall)


#assign(paste("logit4_5",i, sep=""), logit4_5) #Save each model
PCLDA[1,] <- c("lda4_5",i,CO[1,],CO[2,],mae,rmse)


#QDA#
# Make predictions and compute Accuracy, Kappa, MAE, and RMSE #
predictions <- qda4_5 %>% predict(testData, type = "response")
predictions <- as.data.frame(predictions)
names(predictions)[1] <- 'Pred'
mae <- MAE(predictions$posterior.1, as.numeric(testData$04_5))
rmse <- RMSE(predictions$posterior.1, as.numeric(testData$04_5))

#predictions <- as.data.frame(ifelse(predictions > 0.5,1,0))
#predictions$Pred <- factor(predictions$Pred)
testData$04_5 <- factor(testData$04_5)
CON <- confusionMatrix(predictions$Pred, testData$04_5)
CO <- as.data.frame(CON$overall)


#assign(paste("logit4_5",i, sep=""), logit4_5) #Save each model
PCQDA[1,] <- c("qda4_5",i,CO[1,],CO[2,],mae,rmse)



#NN#
# Make predictions and compute Accuracy, Kappa, MAE, and RMSE #
predictions <- nnet4_5 %>% predict(testData, type = "raw")
predictions <- as.data.frame(predictions)
names(predictions)[1] <- 'Pred'
mae <- MAE(predictions$Pred, as.numeric(testData$04_5))
rmse <- RMSE(predictions$Pred, as.numeric(testData$04_5))

predictions <- as.data.frame(ifelse(predictions > 0.5,1,0))
predictions$Pred <- factor(predictions$Pred)
testData$04_5 <- factor(testData$04_5)
CON <- confusionMatrix(predictions$Pred, testData$04_5)
CO <- as.data.frame(CON$overall)


#assign(paste("logit4_5",i, sep=""), logit4_5) #Save each model
PCNN[1,] <- c("nnet4_5",i,CO[1,],CO[2,],mae,rmse)
```

```
  Prediction.Capability <- rbind(Prediction.Capability,PCLogit,PCRS,PCLDA,PCQDA,PCNN)
  #df with all the results

}

Prediction.Capability <- Prediction.Capability[order(Prediction.Capability$Model),]
```

Based on the averages of the scores applied to the ten-fold cross validation none of the models are particularly accurate with the mixed effects logistic regression being the most accurate. The QDA model has the best Kappa score and the logistic regression model has the best MAE and RMSE scores.

```
Prediction.Capability <-  Prediction.Capability %>%
  mutate_at(vars(Accuracy, Kappa, MAE, RMSE), as.numeric)

aggregate(Prediction.Capability[, 3:6], list(Prediction.Capability$Model), mean)
```

```
##     Group.1  Accuracy      Kappa       MAE      RMSE
## 1    lda4_5 0.5828061 0.01530649 0.9985148 1.1118132
## 2  logit4_5 0.5824625 0.01344103 0.4751937 0.4896981
## 3   nnet4_5 0.5800652 0.02706127 0.9988125 1.1136142
## 4    qda4_5 0.5111613 0.05410113 1.0942416 1.2240465
## 5 RSlogit4_5 0.5924128 0.02572497 0.9973471 1.1101579
```

```
set.seed(14)


CVHockey <- Hockey
#shuffle
CVHockey<-CVHockey[sample(nrow(CVHockey)),]
#Create 10 equally size folds
folds <- cut(seq(1,nrow(CVHockey)),breaks=10,labels=FALSE)

Prediction.Capability <- data.frame(matrix(ncol = 6, nrow = 0 )) #Dataframe for results
x <- c("Model","Test_Number", "Accuracy","Kappa","MAE","RMSE") #col names
colnames(Prediction.Capability) <- x
rm(x)
PCLogit <- Prediction.Capability
PCRS <- Prediction.Capability
PCLDA <- Prediction.Capability
PCQDA <- Prediction.Capability
PCNN <- Prediction.Capability

#Perform 10 fold cross validation
for(i in 1:10){
  #Segement your data by fold using the which() function
  testIndexes <- which(folds==i,arr.ind=TRUE)
  testData <- CVHockey[testIndexes, ]
  trainData <- na.omit(CVHockey[-testIndexes, ])



  logit6_5 <- glm(O6_5 ~ Category + Tipp1 + Tipp2 + Over5_5,
              family = binomial,
             data = trainData)

RSlogit6_5 <- glmer(O6_5 ~ Tipp1 + Tipp2 + Over5_5
             + (1| Category) ,nAGQ=0,
             family = binomial,  data=trainData)

lda6_5 <- lda(O6_5 ~ Category + Tipp1 + Tipp2 + Over5_5,
            data = trainData)

qda6_5 <- qda(O6_5 ~ Category + Tipp1 + Tipp2 + Over5_5,
            data = trainData)

nnet6_5 <- nnet(O6_5 ~ Category + Tipp1 + Tipp2 + Over5_5,
              data = trainData,
              size=2,decay=1.0e-2,maxit=1000,trace = FALSE)




#LOGIT#
# Make predictions and compute Accuracy, Kappa, MAE, and RMSE #
predictions <- logit6_5 %>% predict(testData, type = "response")
predictions <- as.data.frame(predictions)
names(predictions)[1] <- 'Pred'
mae <- MAE(predictions$Pred, as.numeric(testData$O6_5))
rmse <- RMSE(predictions$Pred, as.numeric(testData$O6_5))

predictions <- as.data.frame(ifelse(predictions > 0.5,1,0))
predictions$Pred <- factor(predictions$Pred)
testData$O6_5 <- factor(testData$O6_5)
CON <- confusionMatrix(predictions$Pred, testData$O6_5)
CO <- as.data.frame(CON$overall)


#assign(paste("logit6_5",i, sep=""), logit6_5) #Save each model
PCLogit[1,] <- c("logit6_5",i,CO[1,],CO[2,],mae,rmse)


#LOGITRS#
# Make predictions and compute Accuracy, Kappa, MAE, and RMSE #
predictions <- RSlogit6_5 %>% predict(testData, type = "response")
predictions <- as.data.frame(predictions)
names(predictions)[1] <- 'Pred'
mae <- MAE(predictions$Pred, as.numeric(testData$O6_5))
rmse <- RMSE(predictions$Pred, as.numeric(testData$O6_5))

predictions <- as.data.frame(ifelse(predictions > 0.5,1,0))
predictions$Pred <- factor(predictions$Pred)
testData$O6_5 <- factor(testData$O6_5)
CON <- confusionMatrix(predictions$Pred, testData$O6_5)
CO <- as.data.frame(CON$overall)
```

```
#assign(paste("logit6_5",i, sep=""), logit6_5) #Save each model
PCRS[1,] <- c("RSlogit6_5",i,CO[1,],CO[2,],mae,rmse)




#LDA#
# Make predictions and compute Accuracy, Kappa, MAE, and RMSE #
predictions <- lda6_5 %>% predict(testData, type = "response")
predictions <- as.data.frame(predictions)
names(predictions)[1] <- 'Pred'
mae <- MAE(predictions$posterior.1, as.numeric(testData$O6_5))
rmse <- RMSE(predictions$posterior.1, as.numeric(testData$O6_5))

#predictions <- as.data.frame(ifelse(predictions > 0.5,1,0))
#predictions$Pred <- factor(predictions$Pred)
testData$O6_5 <- factor(testData$O6_5)
CON <- confusionMatrix(predictions$Pred, testData$O6_5)
CO <- as.data.frame(CON$overall)


#assign(paste("logit6_5",i, sep=""), logit6_5) #Save each model
PCLDA[1,] <- c("lda6_5",i,CO[1,],CO[2,],mae,rmse)


#QDA#
# Make predictions and compute Accuracy, Kappa, MAE, and RMSE #
predictions <- qda6_5 %>% predict(testData, type = "response")
predictions <- as.data.frame(predictions)
names(predictions)[1] <- 'Pred'
mae <- MAE(predictions$posterior.1, as.numeric(testData$O6_5))
rmse <- RMSE(predictions$posterior.1, as.numeric(testData$O6_5))

#predictions <- as.data.frame(ifelse(predictions > 0.5,1,0))
#predictions$Pred <- factor(predictions$Pred)
testData$O6_5 <- factor(testData$O6_5)
CON <- confusionMatrix(predictions$Pred, testData$O6_5)
CO <- as.data.frame(CON$overall)


#assign(paste("logit6_5",i, sep=""), logit6_5) #Save each model
PCQDA[1,] <- c("qda6_5",i,CO[1,],CO[2,],mae,rmse)



#NN#
# Make predictions and compute Accuracy, Kappa, MAE, and RMSE #
predictions <- nnet6_5 %>% predict(testData, type = "raw")
predictions <- as.data.frame(predictions)
names(predictions)[1] <- 'Pred'
mae <- MAE(predictions$Pred, as.numeric(testData$O6_5))
rmse <- RMSE(predictions$Pred, as.numeric(testData$O6_5))
```

```
  predictions <- as.data.frame(ifelse(predictions > 0.5,1,0))
  predictions$Pred <- factor(predictions$Pred)
  testData$O6_5 <- factor(testData$O6_5)
  CON <- confusionMatrix(predictions$Pred, testData$O6_5)
  CO <- as.data.frame(CON$overall)


  #assign(paste("logit6_5",i, sep=""), logit6_5) #Save each model
  PCNN[1,] <- c("nnet6_5",i,CO[1,],CO[2,],mae,rmse)


  Prediction.Capability <- rbind(Prediction.Capability,PCLogit,PCRS,PCLDA,PCQDA,PCNN)
  #df with all the results

}

Prediction.Capability <- Prediction.Capability[order(Prediction.Capability$Model),]
```

Overall, the accuracy is better, this is probably due to the fact that fewer matches have over 6.5 goals. The very low Kappa values support this assumption. Again, the mixed effects logistic regression is the most accurate. The QDA model has the best Kappa score and the logistic regression model has the best MAE and RMSE scores.

```
Prediction.Capability <-  Prediction.Capability %>%
  mutate_at(vars(Accuracy, Kappa, MAE, RMSE), as.numeric)

aggregate(Prediction.Capability[, 3:6], list(Prediction.Capability$Model), mean)
```

```
##      Group.1  Accuracy        Kappa       MAE      RMSE
## 1     lda6_5 0.7363696 -0.004740006 1.0052062 1.0963720
## 2   logit6_5 0.7367133 -0.004056784 0.3782153 0.4376207
## 3    nnet6_5 0.7319164 -0.002181501 1.0050465 1.0973378
## 4     qda6_5 0.6688439  0.070113762 0.9854934 1.1136906
## 5 RSlogit6_5 0.7380843 -0.001361613 1.0041412 1.0943088
```

Poisson Odds

Under the assumption that the O/U 5.5 odds are accurately estimated and that scoring intensities in ice hockey follow a Poisson process this information may be used to estimate the probabilities of over/under 4.5 goals. This provides an alternative approach to the methods found above and does not require training and testing models.

```
Hockey$Goals <- Hockey$TotScore_T1 + Hockey$TotScore_T2
Hockey$O4_5 <- ifelse(Hockey$Goals > 4.5,1,0)
Hockey$O6_5 <- ifelse(Hockey$Goals > 6.5,1,0)
Hockey$P5_5O <- 1/Hockey$Over5_5
Hockey$P5_5U <- 1/Hockey$Under5_5
```

The assumption that the number of goals follows a Poisson process was tested. Overall, there is not enough evidence to reject the hypothesis that scoring in ice hockey follows a Poisson distribution.

```
mean(Hockey$Goals)
```

```
## [1] 5.158382
```

```
var(Hockey$Goals)
```

```
## [1] 4.987938
```

```
P1 <- glm(Goals ~ Over5_5, family = poisson(link="log"), data=Hockey)
#summary(P1)

#Cameron & Trivedi (1990) Dispersion test
dispersiontest(P1,trafo=1) #trafo = transformation function - linear specification
```

```
##
##  Overdispersion test
##
## data:  P1
## z = -2.6801, p-value = 0.9963
## alternative hypothesis: true alpha is greater than 0
## sample estimates:
##       alpha
## -0.06087474
```

```
dispersiontest(P1,trafo=2) #trafo = transformation function - quadratic specification
```

```
##
##  Overdispersion test
##
## data:  P1
## z = -2.6676, p-value = 0.9962
## alternative hypothesis: true alpha is greater than 0
## sample estimates:
##       alpha
## -0.01171614
```

Numerical analysis was used to in order to estimate the rate parameters assumed for each match. A function was created based on the probability estimates that there are fewer than 5.5 goals coming from the Poisson process.

```
f <- function(x){abs(ppois(5.5, print(x), lower.tail = TRUE, log.p = FALSE)-0.46)}

xmin<-optimize(f, interval=c(4.5,7.5), tol=0.0001)
```

```
## [1] 5.645898
## [1] 6.354102
## [1] 5.208204
## [1] 5.901555
## [1] 5.959944
## [1] 5.869244
## [1] 5.912818
## [1] 5.930819
## [1] 5.913529
## [1] 5.90978
## [1] 5.911657
## [1] 5.911401
## [1] 5.911151
## [1] 5.911464
## [1] 5.911504
## [1] 5.911538
## [1] 5.911504
```

```
#Function to estimate lambda numerically from under5.5 probabilities

Lambda <- function(Probability){

f <- function(x){abs(ppois(5.5, x, lower.tail = TRUE, log.p = FALSE)-Probability)}

xmin<-optimize(f, interval=c(3.5,8.5), tol=0.0001)

xmin$minimum

}
```

Finally, the estimated rate parameters were used to assign probabilities of the over/under 4.5 goals outcomes. The Kappa values (estimated on the entire dataset) are slightly higher than the regression approach.

```
#Rate Parameter for each match
Hockey$Rate <- as.numeric(lapply(Hockey$P5_5U,Lambda))

#Probability of over/under 4.5 Goals
Hockey$PoissonU4_5 <- ppois(4.5, Hockey$Rate, lower.tail = TRUE, log.p = FALSE)
Hockey$PoissonO4_5 <- 1 - Hockey$PoissonU4_5

Hockey$PredictedO4_5Poisson <- ifelse(Hockey$PoissonO4_5 > 0.5,1,0)

#Accuracy Check
confusionMatrix(factor(Hockey$PredictedO4_5Poisson), factor(Hockey$O4_5))
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0  136  118
##          1 1049 1614
##
##                Accuracy : 0.5999
##                  95% CI : (0.5819, 0.6178)
##     No Information Rate : 0.5938
##     P-Value [Acc > NIR] : 0.2549
##
##                   Kappa : 0.0532
##
```

```
## Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.11477
##             Specificity : 0.93187
##          Pos Pred Value : 0.53543
##          Neg Pred Value : 0.60608
##              Prevalence : 0.40624
##          Detection Rate : 0.04662
##    Detection Prevalence : 0.08708
##       Balanced Accuracy : 0.52332
##
##        'Positive' Class : 0
##
```

Conclusions

The models estimated in this task do not perform very well. They are missing relevant variables about the scoring and defending abilities of the teams. Assuming the estimates are correct, the O/U 5.5 goals odds serve as the most informative variable in the analysis as they are presumably estimated using additional information. Assuming a Poisson distribution and then estimating the Over/Under probabilities allows for a simplification of this modelling task.

Another approach to solving this problem could be to combine different techniques through ensemble learning. Furthermore, more could be done to focus on specific leagues and competitions as their scoring intensities may differ. The playing surfaces in North America for example, are smaller than the European competitions.

Resources

lme4 package: https://cran.r-project.org/web/packages/lme4/index.html

nnet package: https://cran.r-project.org/web/packages/nnet/index.html

optimize function: https://www.rdocumentation.org/packages/stats/versions/3.6.2/topics/optimize