Data Analysis

Molly McNamara

7/1/2017

R code necessary to analyze NHL data to address questions for capstone project

Set trainControl, seed and preProcess. In this case, all models will be run with three separate 10-fold cross-validations as the resampling scheme.

```
control <- trainControl(method="repeatedcv", number=10, repeats=3)
seed <- 7
preProcess=c("center", "scale")

#Create the formula to evaluate the variables that may influence
Playoffs
formula <- Playoffs ~ Shooting_Hand + YearsExperience + BirthRegion +
Games_Played + Goals + Assists + Points + Penalty_Minutes + Plus_Minus
+ Shots + GoalsPerGame + ShotsPerGame + PointsPerGame + PercentGoals +
PercentGames + Draft_Pick + Draft_Round + Draft_Age
```

Logistic Regression Model

```
set.seed(seed)
fit.glm <- train(formula, data=dataset, method="glm",
trControl=control, na.action=na.pass)

print(fit.glm)

## Generalized Linear Model
##
## 10001 samples
##    18 predictor
##     2 classes: '0', '1'
##
## No pre-processing
## Resampling: Cross-Validated (10 fold, repeated 3 times)
## Summary of sample sizes: 9308, 9308, 9308, 9307, 9308, 9308, ...
## Resampling results:
##
##   Accuracy   Kappa
##   0.6846381  0.3694707
```

CART Model

```
set.seed(seed)
fit.cart <- train(formula, data=dataset, method="rpart",
trControl=control, na.action = na.pass)
print(fit.cart)
```

```
## CART
##
## 10001 samples
##    18 predictor
##     2 classes: '0', '1'
##
## No pre-processing
## Resampling: Cross-Validated (10 fold, repeated 3 times)
## Summary of sample sizes: 9308, 9308, 9308, 9307, 9308, 9308, ...
## Resampling results across tuning parameters:
##
##   cp          Accuracy   Kappa
##   0.01484561  0.6292790  0.2571150
##   0.02335709  0.6171917  0.2365055
##   0.20645289  0.5671424  0.1305106
##
## Accuracy was used to select the optimal model using  the largest
value.
## The final value used for the model was cp = 0.01484561.
```

C5.0

```
set.seed(seed)
fit.c50 <- train(formula, data=dataset, method="C5.0",
trControl=control, na.action = na.pass)
```

```
print(fit.c50)

## C5.0
##
## 10001 samples
##    18 predictor
##     2 classes: '0', '1'
##
## No pre-processing
## Resampling: Cross-Validated (10 fold, repeated 3 times)
## Summary of sample sizes: 9308, 9308, 9308, 9307, 9308, 9308, ...
## Resampling results across tuning parameters:
##
##   model  winnow  trials  Accuracy   Kappa
##   rules  FALSE    1      0.6440400  0.2883623
##   rules  FALSE   10      0.6664385  0.3313561
##   rules  FALSE   20      0.6627006  0.3242130
##   rules   TRUE    1      0.6437818  0.2877979
##   rules   TRUE   10      0.6639899  0.3266907
##   rules   TRUE   20      0.6570926  0.3130748
##   tree   FALSE    1      0.6421702  0.2847954
##   tree   FALSE   10      0.6746916  0.3486018
##   tree   FALSE   20      0.6694375  0.3380487
##   tree    TRUE    1      0.6418473  0.2839948
##   tree    TRUE   10      0.6652788  0.3294894
##   tree    TRUE   20      0.6616045  0.3229095
##
## Accuracy was used to select the optimal model using  the largest
value.
## The final values used for the model were trials = 10, model = tree
##  and winnow = FALSE.
```

Bagged CART

```
set.seed(seed)
fit.treebag <- train(formula, data=dataset, method="treebag",
trControl=control, na.action = na.pass)
print(fit.treebag)
```

```
## Bagged CART
##
## 10001 samples
##    18 predictor
##     2 classes: '0', '1'
##
## No pre-processing
## Resampling: Cross-Validated (10 fold, repeated 3 times)
## Summary of sample sizes: 9308, 9308, 9308, 9307, 9308, 9308, ...
## Resampling results:
##
##   Accuracy   Kappa
##   0.6586406  0.3168324
```

Random Forest

```r
set.seed(seed)
fit.rf <- train(formula, data=dataset, method="rf", trControl=control,
na.action = na.omit)
print(fit.rf)
```

```
## Random Forest
##
## 10001 samples
##    18 predictor
##     2 classes: '0', '1'
##
## No pre-processing
## Resampling: Cross-Validated (10 fold, repeated 3 times)
## Summary of sample sizes: 9001, 9001, 9001, 9000, 9001, 9001, ...
## Resampling results across tuning parameters:
##
##   mtry  Accuracy   Kappa
##    2    0.6723012  0.3447298
##   11    0.6762004  0.3517189
##   20    0.6701013  0.3395897
##
## Accuracy was used to select the optimal model using  the largest
```

```
value.
## The final value used for the model was mtry = 11.
```

Stochastic Gradient Boosting (Generalized Boosted Modeling)

```
set.seed(seed)
fit.gbm <- train(formula, data=dataset, method="gbm",
trControl=control, verbose=FALSE, na.action = na.pass)

print(fit.gbm)
```

```
## Stochastic Gradient Boosting
##
## 10001 samples
##    18 predictor
##     2 classes: '0', '1'
##
## No pre-processing
## Resampling: Cross-Validated (10 fold, repeated 3 times)
## Summary of sample sizes: 9308, 9308, 9308, 9307, 9308, 9308, ...
## Resampling results across tuning parameters:
##
##   interaction.depth  n.trees  Accuracy   Kappa
##   1                   50      0.6433627  0.2856874
##   1                  100      0.6534186  0.3068404
##   1                  150      0.6596715  0.3195901
##   2                   50      0.6568674  0.3137839
##   2                  100      0.6666985  0.3339102
##   2                  150      0.6725962  0.3456183
##   3                   50      0.6648612  0.3300943
##   3                  100      0.6744667  0.3493229
##   3                  150      0.6816217  0.3635975
##
## Tuning parameter 'shrinkage' was held constant at a value of 0.1
##
## Tuning parameter 'n.minobsinnode' was held constant at a value of
10
## Accuracy was used to select the optimal model using  the largest
value.
```

```
## The final values used for the model were n.trees = 150,
##  interaction.depth = 3, shrinkage = 0.1 and n.minobsinnode = 10.
```

Compile the resamples results from the models

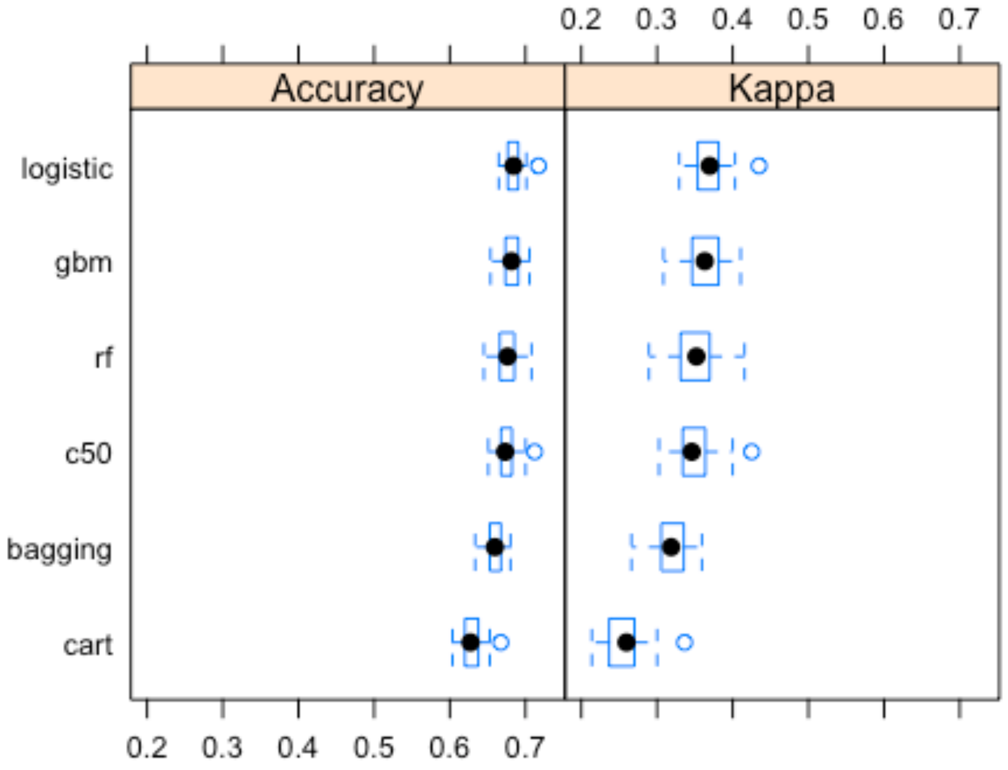
```
results <- resamples(list(logistic=fit.glm, cart=fit.cart,
c50=fit.c50, bagging=fit.treebag, rf=fit.rf, gbm=fit.gbm))
```

Compare method accuracy

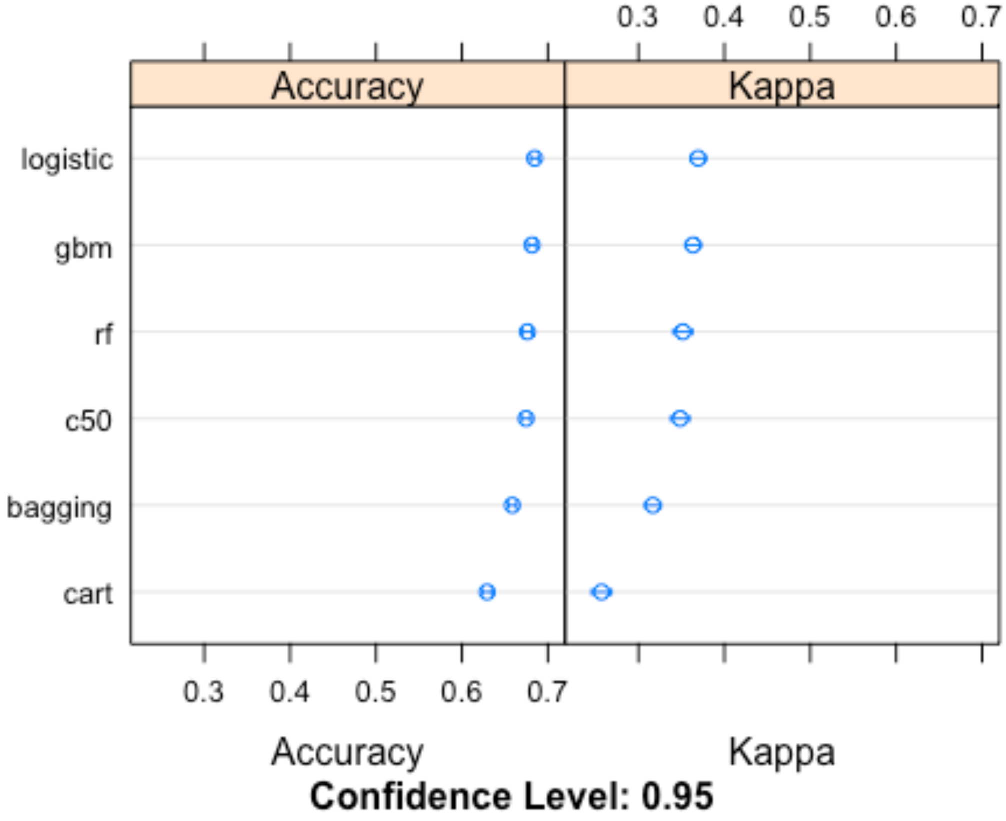
```
summary(results, metric="Accuracy")

##
## Call:
## summary.resamples(object = results, metric = "Accuracy")
##
## Models: logistic, cart, c50, bagging, rf, gbm
## Number of resamples: 30
##
## Accuracy
##               Min.   1st Qu.    Median      Mean   3rd Qu.
Max. NA's
## logistic 0.6646825 0.6767920 0.6842386 0.6846381 0.6904529
0.7171515    0
## cart     0.6034816 0.6201644 0.6273559 0.6292790 0.6374396
0.6673114    0
## c50      0.6508704 0.6677943 0.6731141 0.6746916 0.6815652
0.7120773    0
## bagging  0.6334623 0.6532882 0.6595745 0.6586406 0.6676329
0.6801932    0
## rf       0.6450000 0.6660000 0.6763388 0.6762004 0.6852360
0.7080000    0
## gbm      0.6537718 0.6740763 0.6814888 0.6816217 0.6902805
0.7053140    0

# Boxplot comparison of methods
bwplot(results)
```



```
# Dot-plot comparison of methods
dotplot(results)
```



Some further investigation of the linear regression model

```
summary(fit.glm)

## 
## Call:
## NULL
## 
## Deviance Residuals: 
##     Min       1Q   Median       3Q      Max  
## -3.7169  -1.0421   0.1247   1.0506   2.9056  
## 
## Coefficients: (1 not defined because of singularities)
##                         Estimate Std. Error z value Pr(>|z|)
## (Intercept)         -1.560e+00  7.847e-01  -1.988   0.0468 *
## Shooting_HandR      -5.153e-02  4.939e-02  -1.043   0.2968
## YearsExperience      3.735e-02  5.053e-03   7.392 1.45e-13 ***
## BirthRegionAmericas  1.298e+00  7.064e-01   1.838   0.0661 .
## BirthRegionAsia      1.092e+00  7.719e-01   1.415   0.1571
## BirthRegionEurope    1.284e+00  7.073e-01   1.815   0.0696 .
## Games_Played        -1.136e+04  1.514e+04  -0.750   0.4531
## Goals                6.879e-01  3.272e-02  21.020  < 2e-16 ***
## Assists             -3.354e-02  5.884e-03  -5.699 1.20e-08 ***
## Points                      NA         NA      NA       NA
## Penalty_Minutes     -2.157e-03  8.425e-04  -2.560   0.0105 *
## Plus_Minus           1.054e-01  3.753e-03  28.087  < 2e-16 ***
## Shots                7.353e-03  1.542e-03   4.768 1.86e-06 ***
## GoalsPerGame        -8.474e-03  4.800e-01  -0.018   0.9859
## ShotsPerGame        -1.462e-01  6.206e-02  -2.356   0.0185 *
## PointsPerGame        2.712e-01  2.448e-01   1.108   0.2679
## PercentGoals        -1.625e+00  7.244e-02 -22.433  < 2e-16 ***
## PercentGames         9.312e+03  1.241e+04   0.750   0.4531
## Draft_Pick          -2.532e-03  2.043e-03  -1.239   0.2152
## Draft_Round          9.958e-02  6.440e-02   1.546   0.1221
## Draft_Age            1.197e-03  1.798e-02   0.067   0.9469
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 13858  on 10000  degrees of freedom
## Residual deviance: 11443  on  9981  degrees of freedom
##   (341 observations deleted due to missingness)
## AIC: 11483
##
## Number of Fisher Scoring iterations: 5
```

Try to re-create the logistic regression model more simply to address potential overfitting

```r
set.seed(seed)
Logmodel <- glm(formula, family="binomial", dataset)
summary(Logmodel)
```

```
## 
## Call:
## glm(formula = formula, family = "binomial", data = dataset)
## 
## Deviance Residuals: 
##     Min       1Q   Median       3Q      Max  
## -3.7169  -1.0421   0.1247   1.0506   2.9056  
## 
## Coefficients: (1 not defined because of singularities)
##                       Estimate Std. Error z value Pr(>|z|)    
## (Intercept)         -1.560e+00  7.847e-01  -1.988   0.0468 *  
## Shooting_HandR      -5.153e-02  4.939e-02  -1.043   0.2968    
## YearsExperience      3.735e-02  5.053e-03   7.392 1.45e-13 ***
## BirthRegionAmericas  1.298e+00  7.064e-01   1.838   0.0661 .  
## BirthRegionAsia      1.092e+00  7.719e-01   1.415   0.1571    
## BirthRegionEurope    1.284e+00  7.073e-01   1.815   0.0696 .  
## Games_Played        -1.136e+04  1.514e+04  -0.750   0.4531    
## Goals                6.879e-01  3.272e-02  21.020  < 2e-16 ***
## Assists             -3.354e-02  5.884e-03  -5.699 1.20e-08 ***
## Points                      NA         NA      NA       NA    
## Penalty_Minutes     -2.157e-03  8.425e-04  -2.560   0.0105 *  
## Plus_Minus           1.054e-01  3.753e-03  28.087  < 2e-16 ***
## Shots                7.353e-03  1.542e-03   4.768 1.86e-06 ***
## GoalsPerGame        -8.474e-03  4.800e-01  -0.018   0.9859    
## ShotsPerGame        -1.462e-01  6.206e-02  -2.356   0.0185 *  
## PointsPerGame        2.712e-01  2.448e-01   1.108   0.2679    
## PercentGoals        -1.625e+00  7.244e-02 -22.433  < 2e-16 ***
## PercentGames         9.312e+03  1.241e+04   0.750   0.4531    
## Draft_Pick          -2.532e-03  2.043e-03  -1.239   0.2152    
## Draft_Round          9.958e-02  6.440e-02   1.546   0.1221    
## Draft_Age            1.197e-03  1.798e-02   0.067   0.9469    
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 13858  on 10000  degrees of freedom
## Residual deviance: 11443  on  9981  degrees of freedom
##   (341 observations deleted due to missingness)
## AIC: 11483
##
## Number of Fisher Scoring iterations: 5
```

```
anova(Logmodel, test ="Chisq")
```

```
## Analysis of Deviance Table
##
## Model: binomial, link: logit
##
## Response: Playoffs
##
## Terms added sequentially (first to last)
##
##
##                  Df Deviance Resid. Df Resid. Dev  Pr(>Chi)
## NULL                            10000      13858
## Shooting_Hand    1     1.75      9999      13856    0.1864
## YearsExperience  1    71.53      9998      13784 < 2.2e-16 ***
## BirthRegion      3     3.92      9995      13780    0.2699
## Games_Played     1    30.08      9994      13750 4.150e-08 ***
## Goals            1    21.34      9993      13729 3.843e-06 ***
## Assists          1    16.62      9992      13712 4.557e-05 ***
## Points           0     0.00      9992      13712
## Penalty_Minutes  1     2.10      9991      13710    0.1477
## Plus_Minus       1  1519.35      9990      12191 < 2.2e-16 ***
## Shots            1     0.98      9989      12190    0.3232
## GoalsPerGame     1     2.43      9988      12188    0.1188
## ShotsPerGame     1     4.98      9987      12183    0.0257 *
## PointsPerGame    1     0.14      9986      12182    0.7100
## PercentGoals     1   732.52      9985      11450 < 2.2e-16 ***
## PercentGoals     1     0.55      9984      11449    0.4575    
## Draft_Pick       1     3.64      9983      11446    0.0564 .  
## Draft_Round      1     2.39      9982      11443    0.1219    
## Draft_Age        1     0.00      9981      11443    0.9469    
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```
varImp(Logmodel)
```

```
##                         Overall
## Shooting_HandR       1.04330519
## YearsExperience      7.39173754
## BirthRegionAmericas  1.83787103
## BirthRegionAsia      1.41494957
## BirthRegionEurope    1.81475008
## Games_Played         0.75021078
## Goals               21.02021848
## Assists              5.69931717
## Penalty_Minutes      2.56002259
## Plus_Minus          28.08714323
## Shots                4.76760549
## GoalsPerGame         0.01765492
## ShotsPerGame         2.35643927
## PointsPerGame        1.10792895
## PercentGoals        22.43294486
## PercentGames         0.75021116
## Draft_Pick           1.23945102
## Draft_Round          1.54622395
## Draft_Age            0.06660011
```

Question 2: Differences between Over and Underperforming Draft Picks

```
#Determine average goals
summary(FullData$Goals)
```

```
##    Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
##   0.000   0.000   3.000   6.682  10.000  65.000
```

Set Over and Under Achievers: Players who were drafted early and score less than average goals & players who were drafted late who score more than average

```
Under  <- subset(FullData, Draft_Round <= 2 & Goals <=6.68)
Over <- subset(FullData, Draft_Round >= 7 & Goals >=6.68)
```

Create one data file with a variable to identify over or underachievers

```
Question2 <- rbind(Under, Over)
Question2 <- mutate(Question2, OverUnder =
as.numeric(Question2$Draft_Round <=2))
OverUnder <- c("OverUnder")
Question2[OverUnder][is.na(Question2[OverUnder])] <- 0
Question2$OverUnder <- as.factor(Question2$OverUnder)
```

Logistic regression model

```
PerfModel <- glm(OverUnder ~ Height + Weight + Position_Played +
BirthRegion + Draft_Team + Draft_Age + AmateurLeague,
family="binomial", Question2)

summary(PerfModel)

##
## Call:
## glm(formula = OverUnder ~ Height + Weight + Position_Played +
##     BirthRegion + Draft_Team + Draft_Age + AmateurLeague, family =
"binomial",
##     data = Question2)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.4502   0.0000   0.0656   0.2214   2.9599
##
```

```
## Coefficients:
##                                   Estimate Std. Error z value Pr(>|
z|)
## (Intercept)                       1.620e+01  4.543e+03   0.004
0.997155
## Height                            5.186e-01  8.885e-02   5.837
5.32e-09 ***
## Weight                           -5.826e-04  1.029e-02  -0.057
0.954849
## Position_PlayedD                  1.913e+00  3.214e-01   5.953
2.64e-09 ***
## Position_PlayedG                  1.997e+01  8.001e+02   0.025
0.980089
## Position_PlayedL                  1.422e+00  3.662e-01   3.883
0.000103 ***
## Position_PlayedR                  2.366e-01  3.269e-01   0.724
0.469168
## BirthRegionAmericas              -1.494e+01  4.543e+03  -0.003
0.997376
## BirthRegionAsia                  -1.894e+01  4.543e+03  -0.004
0.996674
## BirthRegionEurope                -1.566e+01  4.543e+03  -0.003
0.997250
## Draft_TeamAtlanta Thrashers       5.048e-01  1.334e+00   0.378
0.705069
## Draft_TeamBoston Bruins           2.864e+00  9.890e-01   2.896
0.003776 **
## Draft_TeamBuffalo Sabres          1.065e-01  6.853e-01   0.155
0.876495
## Draft_TeamCalgary Flames          2.051e+00  7.793e-01   2.632
0.008482 **
## Draft_TeamCarolina Hurricanes     2.082e+01  2.292e+03   0.009
0.992752
## Draft_TeamChicago Blackhawks     -1.227e-01  6.703e-01  -0.183
0.854765
## Draft_TeamColorado Avalanche      5.126e-01  7.405e-01   0.692
0.488833
## Draft_TeamColumbus Blue Jackets  1.600e+00  1.463e+00   1.094
0.274162
## Draft_TeamDallas Stars            2.286e+00  8.435e-01   2.710
```

```
0.006732 **
## Draft_TeamDetroit Red Wings      1.901e+01  1.783e+03   0.011
0.991494
## Draft_TeamEdmonton Oilers        1.374e+00  7.206e-01   1.907
0.056533 .
## Draft_TeamFlorida Panthers       1.777e+01  1.580e+03   0.011
0.991024
## Draft_TeamHartford Whalers       4.210e+00  1.266e+00   3.325
0.000885 ***
## Draft_TeamLos Angeles Kings      1.999e+01  1.598e+03   0.013
0.990024
## Draft_TeamMinnesota North Stars  1.688e+01  4.158e+03   0.004
0.996761
## Draft_TeamMinnesota Wild         2.076e+01  2.119e+03   0.010
0.992181
## Draft_TeamMontreal Canadiens     1.267e+00  7.022e-01   1.805
0.071113 .
## Draft_TeamNashville Predators    1.820e+01  2.392e+03   0.008
0.993931
## Draft_TeamNew Jersey Devils      1.474e+00  7.644e-01   1.928
0.053834 .
## Draft_TeamNew York Islanders     2.383e+00  9.715e-01   2.453
0.014173 *
## Draft_TeamNew York Rangers       3.594e+00  1.272e+00   2.825
0.004722 **
## Draft_TeamOttawa Senators        2.461e-02  7.372e-01   0.033
0.973367
## Draft_TeamPhiladelphia Flyers    4.283e+00  1.380e+00   3.103
0.001916 **
## Draft_TeamPhoenix Coyotes        3.819e-01  1.009e+00   0.378
0.705111
## Draft_TeamPittsburgh Penguins    2.969e-01  7.194e-01   0.413
0.679842
## Draft_TeamQuebec Nordiques       2.383e+00  1.487e+00   1.603
0.108911
## Draft_TeamSan Jose Sharks        2.166e+00  9.234e-01   2.345
0.019007 *
## Draft_TeamSt. Louis Blues        2.068e-01  7.449e-01   0.278
0.781316
## Draft_TeamTampa Bay Lightning    1.609e+00  1.033e+00   1.558
```

```
0.119260
## Draft_TeamToronto Maple Leafs    7.105e-01  8.380e-01   0.848
0.396513
## Draft_TeamVancouver Canucks      2.151e+00  9.572e-01   2.247
0.024649 *
## Draft_TeamWashington Capitals    2.632e+00  9.856e-01   2.671
0.007566 **
## Draft_TeamWinnipeg Jets          1.816e+01  2.191e+03   0.008
0.993384
## Draft_Age                       -2.066e+00  2.106e-01  -9.807  <
2e-16 ***
## AmateurLeagueAJHL                1.382e+01  1.254e+04   0.001
0.999121
## AmateurLeagueBCHL               -1.390e+00  1.272e+00  -1.093
0.274563
## AmateurLeagueCCHA                2.442e+00  1.610e+00   1.516
0.129491
## AmateurLeagueECAC               -6.630e-02  1.185e+00  -0.056
0.955370
## AmateurLeagueFinland            -2.409e-01  9.758e+03   0.000
0.999980
## AmateurLeagueH-East              2.633e+00  1.262e+00   2.086
0.036947 *
## AmateurLeagueHigh-CT            -1.816e+00  2.734e+00  -0.664
0.506510
## AmateurLeagueHigh-IN             1.211e+01  1.773e+04   0.001
0.999455
## AmateurLeagueHigh-MA            -2.094e+00  1.116e+00  -1.877
0.060539 .
## AmateurLeagueHigh-ME             1.536e+01  6.269e+03   0.002
0.998044
## AmateurLeagueHigh-MN            -3.685e-01  1.477e+00  -0.250
0.802904
## AmateurLeagueHigh-NY             1.693e+01  3.920e+03   0.004
0.996554
## AmateurLeagueHigh-VT             1.504e+01  1.773e+04   0.001
0.999323
## AmateurLeagueIHL                 2.351e+01  4.718e+03   0.005
0.996025
## AmateurLeagueNAHL                1.583e+01  7.856e+03   0.002
```

```
0.998393
## AmateurLeagueOHL                  9.882e-02  1.053e+00   0.094
0.925204
## AmateurLeagueOJHL                 1.937e+01  1.254e+04   0.002
0.998768
## AmateurLeagueQMJHL               -1.737e+00  1.057e+00  -1.643
0.100295
## AmateurLeagueRussia               9.248e+00  2.093e+00   4.418
9.98e-06 ***
## AmateurLeagueRussia-3             1.694e+01  8.865e+03   0.002
0.998475
## AmateurLeagueSlovakia             1.477e+00  1.782e+04   0.000
0.999934
## AmateurLeagueSweden              -8.561e-02  1.316e+00  -0.065
0.948146
## AmateurLeagueSweden-2             1.664e+01  1.773e+04   0.001
0.999251
## AmateurLeagueSweden-Jr.           1.578e+01  6.700e+03   0.002
0.998120
## AmateurLeagueUSHL                -2.334e+00  1.102e+00  -2.118
0.034153 *
## AmateurLeagueUSNTDP               1.791e+01  3.615e+03   0.005
0.996047
## AmateurLeagueWCHA                 1.349e+00  1.139e+00   1.185
0.236181
## AmateurLeagueWCHL                 1.693e+01  3.655e+03   0.005
0.996304
## AmateurLeagueWHL                 -1.121e+00  1.038e+00  -1.080
0.280227
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1364.61  on 2630  degrees of freedom
## Residual deviance:  677.49  on 2558  degrees of freedom
##   (776 observations deleted due to missingness)
## AIC: 823.49
```

```
##
## Number of Fisher Scoring iterations: 19
```

Further evaluate model components

```
anova(PerfModel, test="Chisq")
```

```
## Analysis of Deviance Table
##
## Model: binomial, link: logit
##
## Response: OverUnder
##
## Terms added sequentially (first to last)
##
##
##                 Df Deviance Resid. Df Resid. Dev Pr(>Chi)
## NULL                             2630    1364.61
## Height           1  149.140      2629    1215.47   <2e-16 ***
## Weight           1    0.001      2628    1215.47   0.9714
## Position_Played  4  134.610      2624    1080.86   <2e-16 ***
## BirthRegion      3    3.796      2621    1077.07   0.2844
## Draft_Team      33  157.066      2588     920.00   <2e-16 ***
## Draft_Age        1   92.537      2587     827.46   <2e-16 ***
## AmateurLeague   29  149.971      2558     677.49   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```