Classification

Jered Hightower & Haniyyah Hamid

9/26/2022

https://www.kaggle.com/datasets/datasnaek/chess

This notebook explores Chess data from Kaggle

How Linear Models for Classification Work: Strengths and Weaknesses

Linear models for classification find a decision boundary between classes. As with any other linear algorithm, it will perform poorly where there are non-linear relationships, it's biased. However, logistic regression is strong when the data is linear and are fairly easy to interpret.

Import our Data Set

```r
df <- read.csv("games.csv")
str(df)
```

```
## 'data.frame':    20058 obs. of  16 variables:
##  $ id            : chr  "TZJHLljE" "l1NXvwaE" "mIICvQHh" "kWKvrqYL" ...
##  $ rated         : chr  "FALSE" "TRUE" "TRUE" "TRUE" ...
##  $ created_at    : num  1.5e+12 1.5e+12 1.5e+12 1.5e+12 1.5e+12 ...
##  $ last_move_at  : num  1.5e+12 1.5e+12 1.5e+12 1.5e+12 1.5e+12 ...
##  $ turns         : int  13 16 61 61 95 5 33 9 66 119 ...
##  $ victory_status: chr  "outoftime" "resign" "mate" "mate" ...
##  $ winner        : chr  "white" "black" "white" "white" ...
##  $ increment_code: chr  "15+2" "5+10" "5+10" "20+0" ...
##  $ white_id      : chr  "bourgris" "a-00" "ischia" "daniamurashov" ...
##  $ white_rating  : int  1500 1322 1496 1439 1523 1250 1520 1413 1439 1381 ...
##  $ black_id      : chr  "a-00" "skinnerua" "a-00" "adivanov2009" ...
##  $ black_rating  : int  1191 1261 1500 1454 1469 1002 1423 2108 1392 1209 ...
##  $ moves         : chr  "d4 d5 c4 c6 cxd5 e6 dxe6 fxe6 Nf3 Bb4+ Nc3 Ba5 Bf4" "d4 Nc6 e4 e5 f4 f6 dxe
##  $ opening_eco   : chr  "D10" "B00" "C20" "D02" ...
##  $ opening_name  : chr  "Slav Defense: Exchange Variation" "Nimzowitsch Defense: Kennedy Variation"
##  $ opening_ply   : int  5 4 3 3 5 4 10 5 6 4 ...
```

Data Cleanup

Subset data frame and make factors into factors. Make target binomial

```r
df <- df[,c(2, 5, 6, 7, 10, 12, 16)]
df$rated <- factor(tolower(df$rated))
df$victory_status <- factor(df$victory_status)
df$winner <- factor(df$winner)

# New Rating Difference Factor (Positive: Black Favored, Negative: White Favored)
df$rating_difference <- df$black_rating- df$white_rating
```

```
# Remove White, Black Ratings and Victory Status
df <- df[,c(1, 2, 4, 7, 8)]

# Combine White and Draw Factors
levels(df$winner) <- c("black", "not black", "not black")

str(df)
```

```
## 'data.frame':    20058 obs. of  5 variables:
##  $ rated            : Factor w/ 2 levels "false","true": 1 2 2 2 2 1 2 1 2 2 ...
##  $ turns            : int  13 16 61 61 95 5 33 9 66 119 ...
##  $ winner           : Factor w/ 2 levels "black","not black": 2 1 2 2 2 2 2 1 1 2 ...
##  $ opening_ply      : int  5 4 3 3 5 4 10 5 6 4 ...
##  $ rating_difference: int  -309 -61 4 15 -54 -248 -97 695 -47 -172 ...
```

Divide into Train and Test

```
set.seed(1234)
i <- sample(1:nrow(df), nrow(df)*.8, replace = FALSE)

train <- df[i, ]
test <- df[-i, ]
```

Data Exploration

A look at some of the data

```
head(df)
```

```
##   rated turns    winner opening_ply rating_difference
## 1 false    13 not black           5              -309
## 2  true    16     black           4               -61
## 3  true    61 not black           3                 4
## 4  true    61 not black           3                15
## 5  true    95 not black           5               -54
## 6 false     5 not black           4              -248
```

```
counts <- table(df$winner)
counts <- counts / sum(counts)
counts[1]
```

Proportion of How Often Black Won

```
##     black
## 0.4540333
```

This is what we hope our model beats in accuracy.

```
mean(df$rating_difference)
```

Average Rating Difference

```
## [1] -7.79988
```

Slightly in favor of white.

```
median(df$rating_difference)
```

Median Rating Difference

```
## [1] -3
```

Seems matchmaking is distributed fairly evenly (which would make sense, avg and median are similar).

```
range(df$rating_difference)
```

The Biggest Difference in Ratings

```
## [1] -1499  1605
```

This represents the biggest skill gap there was in some games (Yikes…).

```
cor(abs(df$rating_difference), df$turns)
```

Correlation between Rating Difference and Turns

```
## [1] -0.1265309
```

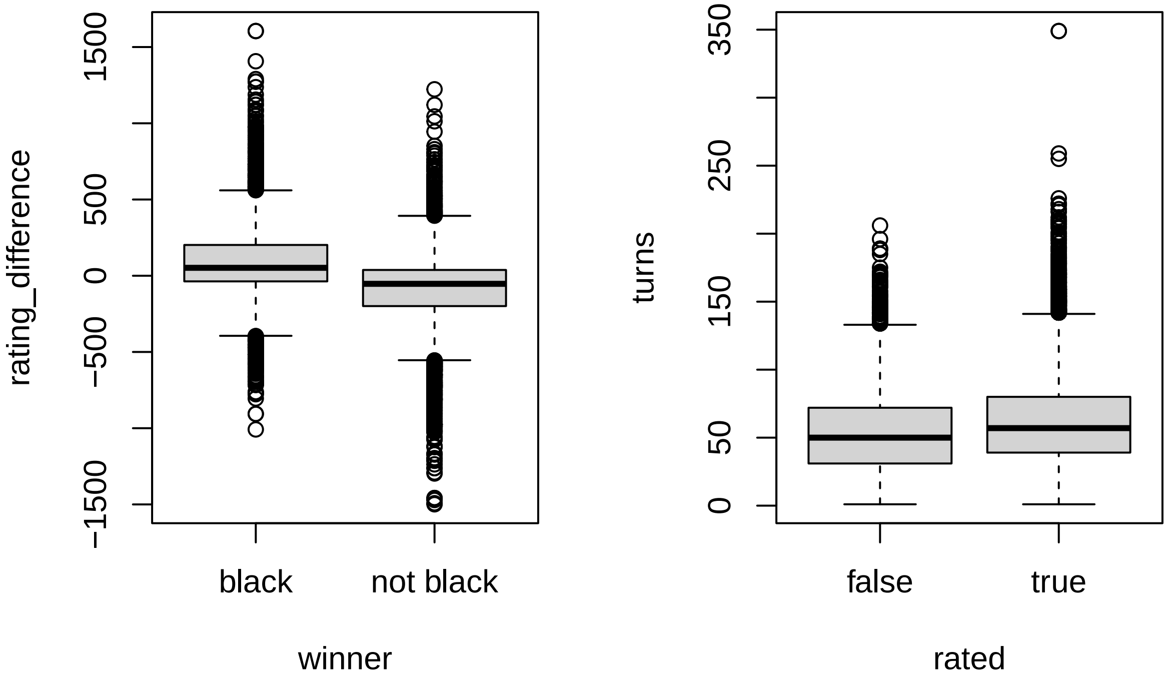
Do bigger skill gaps lead to quicker games? There's a little correlation.

```
rate <- ifelse(df$rating_difference > 0, 1, -1)
win <- ifelse(df$winner=="black", 1, 0)
rightful <- sum(rate==win) / sum(win)
print(paste("Proportion of games black wins when higher rated", rightful))
```

How Often Black Wins (if higher rated)

```
## [1] "Proportion of games black wins when higher rated 0.639398265070825"
```

```
par(mfrow=c(1, 2))
plot(df$winner, df$rating_difference, xlab = "winner", ylab = "rating_difference")
plot(df$rated, df$turns, xlab = "rated", ylab = "turns")
```



Plots

A rating difference that favors black seems to imply that black will win more often. A rated game also looks like it tends to last just a bit longer than non-rated games.

Logistic Regression Model

```
library(caret)
```

```
## Loading required package: ggplot2

## Loading required package: lattice
```

```
# Predict if Black Wins or Not
glm1 <- glm(winner~., data = train, family = "binomial")
summary(glm1)
```

```
##
## Call:
## glm(formula = winner ~ ., family = "binomial", data = train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.7107  -1.1424   0.5806   1.0427   2.8036
##
## Coefficients:
##                       Estimate Std. Error z value Pr(>|z|)
## (Intercept)          1.500e-01  5.698e-02   2.633  0.00846 **
## ratedtrue           -5.169e-02  4.499e-02  -1.149  0.25061
## turns                1.383e-04  5.053e-04   0.274  0.78438
## opening_ply          1.763e-02  6.130e-03   2.877  0.00402 **
## rating_difference -3.682e-03  9.052e-05 -40.671  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
```

```
## 
##     Null deviance: 22083  on 16045  degrees of freedom
## Residual deviance: 19808  on 16041  degrees of freedom
## AIC: 19818
## 
## Number of Fisher Scoring iterations: 4
```

Evaluate on the test set

```
probs <- predict(glm1, newdata=test, type="response")
pred <- ifelse(probs>0.5, "not black", "black")
acc <- mean(pred==test$winner)
print(paste("accuracy = ", acc))
```

```
## [1] "accuracy =  0.644067796610169"
```

Sensitivity, Specificity & Kappa

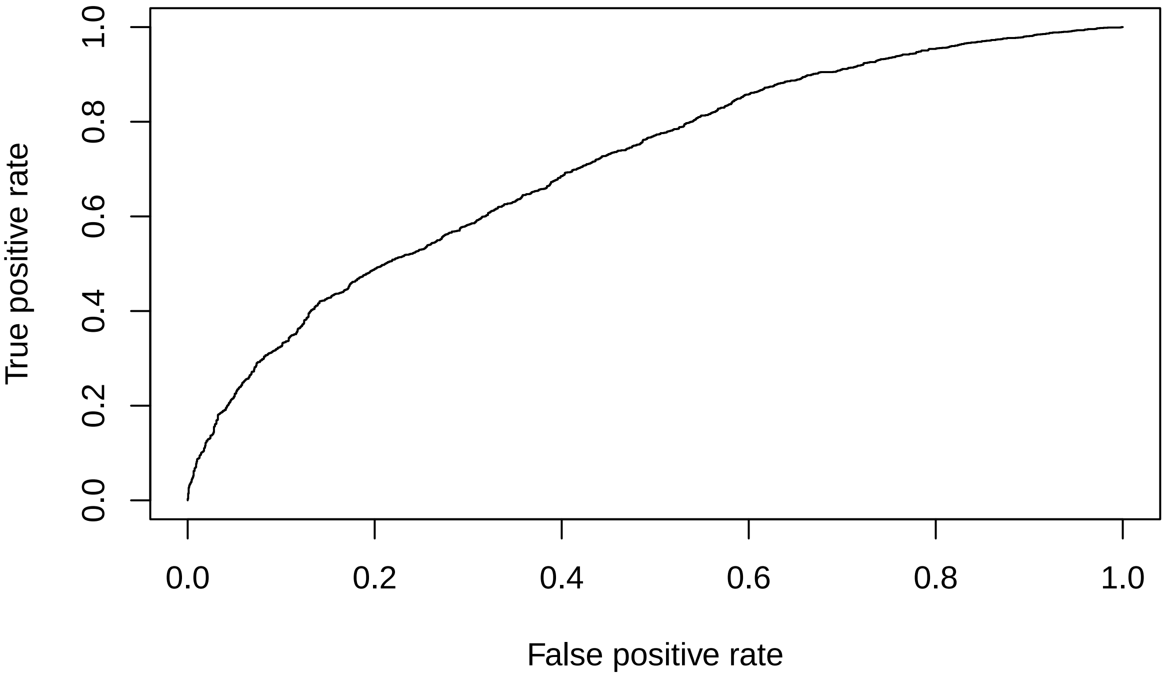
```
caret:: confusionMatrix(as.factor(pred), reference=test$winner)
```

```
## Confusion Matrix and Statistics
## 
##            Reference
## Prediction  black not black
##   black       941       482
##   not black   946      1643
## 
##                Accuracy : 0.6441
##                  95% CI : (0.629, 0.6589)
##     No Information Rate : 0.5297
##     P-Value [Acc > NIR] : < 2.2e-16
## 
##                   Kappa : 0.2756
## 
##  Mcnemar's Test P-Value : < 2.2e-16
## 
##             Sensitivity : 0.4987
##             Specificity : 0.7732
##          Pos Pred Value : 0.6613
##          Neg Pred Value : 0.6346
##              Prevalence : 0.4703
##          Detection Rate : 0.2345
##    Detection Prevalence : 0.3547
##       Balanced Accuracy : 0.6359
## 
##        'Positive' Class : black
## 
```

ROC Curve and AUC

```
library(ROCR)
pr <- prediction(probs, test$winner)
# TPR = Sensitivity, FPR=Specificity
```

```
prf <- performance(pr, measure = "tpr", x.measure ="fpr")
plot(prf)
```



```
auc <- performance(pr, measure ="auc")
auc <- auc@y.values[[1]]
auc
```

```
## [1] 0.7115798
```

Matthew's Correlation Coefficient

```
library(ModelMetrics)
```

```
##
## Attaching package: 'ModelMetrics'

## The following objects are masked from 'package:caret':
##
##     confusionMatrix, precision, recall, sensitivity, specificity

## The following object is masked from 'package:base':
##
##     kappa
```

```
predbin <- ifelse(pred=="black", 1, 0)
realbin <- ifelse(test$winner=="black", 1, 0)

mcc(realbin, predbin, .5)
```

```
## [1] 0.2836148
```

Model Summary

The summary shows us that R believes rating_difference is our best predictor. It also thinks opening_ply (number of moves in the opening phase) is a good predictor. The Null Deviance and Residual Deviance are measures of how well our model fits the data with only the intercept and with all predictors. Lower Deviance is better. Using them we can calculate that the p-value is close to zero meaning this model may be useful for prediction. This is shown by our decent accuracy. AIC also measures how well our model fits the data but its relative to other models. Lower AIC is better.

Naive Bayes

```
library(e1071)

# Predict if Black Wins or Not
nb1 <- naiveBayes(winner~., data=train)
nb1
```

```
##
## Naive Bayes Classifier for Discrete Predictors
##
## Call:
## naiveBayes.default(x = X, y = Y, laplace = laplace)
##
## A-priori probabilities:
## Y
##     black not black
## 0.4499564 0.5500436
##
## Conditional probabilities:
##            rated
## Y               false      true
##   black     0.1875346 0.8124654
##   not black 0.1989576 0.8010424
##
##            turns
## Y               [,1]     [,2]
##   black     60.91717 32.34590
##   not black 60.33843 34.67622
##
##            opening_ply
## Y               [,1]     [,2]
##   black     4.740305 2.806994
##   not black 4.906639 2.795202
##
##            rating_difference
## Y                 [,1]     [,2]
##   black      87.96427 230.5815
##   not black -88.04725 234.4408
```

Evaluate on the test set

```
p1 <- predict(nb1, newdata=test, type="class")
acc <- mean(p1==test$winner)
print(paste("accuracy = ", acc))
```

```
## [1] "accuracy =  0.643320039880359"
```

Sensitivity, Specificity & Kappa

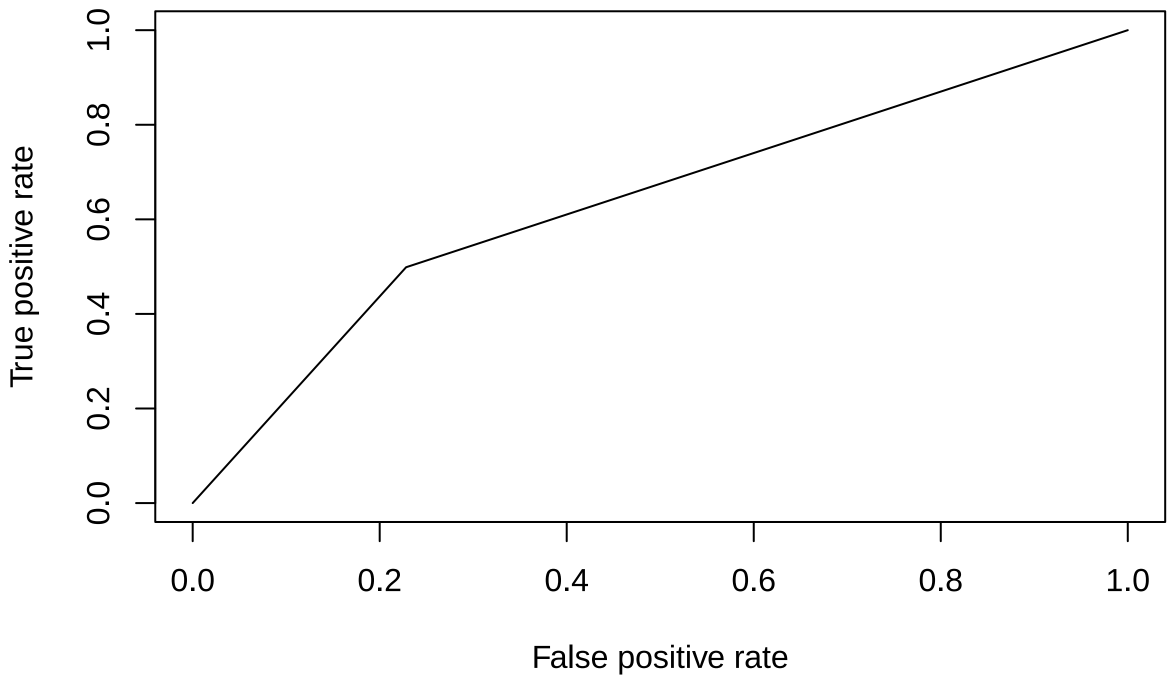
```
caret:: confusionMatrix(as.factor(p1), reference=test$winner)
```

```
## Confusion Matrix and Statistics
##
##            Reference
## Prediction  black not black
##   black       941       485
##   not black   946      1640
##
##                Accuracy : 0.6433
##                  95% CI : (0.6283, 0.6582)
##     No Information Rate : 0.5297
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.2742
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.4987
##             Specificity : 0.7718
##          Pos Pred Value : 0.6599
##          Neg Pred Value : 0.6342
##              Prevalence : 0.4703
##          Detection Rate : 0.2345
##    Detection Prevalence : 0.3554
##       Balanced Accuracy : 0.6352
##
##        'Positive' Class : black
##
```

ROC Curve and AUC

```
predvec <- ifelse(p1=="black", 1, 0)
realvec <- ifelse(test$winner=="black", 1, 0)

pr <- prediction(predvec, realvec)
# TPR = Sensitivity, FPR=Specificity
prf <- performance(pr, measure = "tpr", x.measure ="fpr")
plot(prf)
```



```
auc <- performance(pr, measure ="auc")
auc <- auc@y.values[[1]]
auc
```

```
## [1] 0.6352199
```

Matthew's Correlation Coefficient

```
mcc(realvec, predvec, .5)
```

```
## [1] 0.2820084
```

Model Summary

The A-priori probabilities are purely based on the distribution of black vs. not black (winning). This matches up with the proportion of black wins shown previously. There are conditional probabilities for each feature (TP, FP, FN, TN). We had fairly good accuracy for this model.

Comparing Results and Metrics of the Models

Accuracy The Naive Bayes model had similar accuracy to the logistic regression model. However, their accuracy is almost exactly the same as the proportion of black wins when they were higher rated. This implies the other predictors had little effect on the model.

Sensitivity and Specificity Both models has similar sensitivity and specificity. This implies they both captured a similar proportion of relevant items and were similarly precise.

Kappa Both models had a Kappa value of around .275 meaning little to some data was predicted correctly and not just by chance. After all our accuracy was only about .20 greater than the proportion of black wins.

ROC Curve and AUC The ROC Curve looks pretty similar between the models excluding the linearity of the Naive Bayes model curve. Both had a similar AUC (log reg = .71, bayes = .64) and we would prefer them to be closer to 1. Since they're working with the same data, it's not surprising their curves are similar (and the models as a whole).

Matthew's Correlation Coefficient Both of their Matthew's Correlation Coefficients are also very similar around .28. Mcc accounts for class distribution and similar to their Kappa above, this implies that our models are a little better than a random prediction based on class distribution.

Strengths and Weaknesses of Logistic Regression

Strengths of logistic regression includes how it can separate classes nicely if the classes themselves are linearly separable, it can be computationally inexpensive, and it can also have good probabilistic output. A weakness of logistic regression is that it is likely that it can under fit the data, meaning it may not be able to capture outliers and boundaries.

Strengths and Weaknesses of Naive Bayes

Strengths of Naive Bayes includes how it can work well with small data sets, it is easy to implement and understand, and it can handle high dimensions. Weaknesses of Naive Bayes includes how it can be outperformed by other classifying models when it comes to large data sets, it makes guesses for values in the test set that may not have occurred in the training set, and if predictors are not independent then assumption that they are instead may reduce the overall performance of the algorithm.

Benefits and Drawbacks of

Accuracy

Accuracy is a simple way of calculating the number of correct predictions out of the total number of examples. A drawback of this would be that it does not account for the differences in class distribution.

Sensitivity and Specificity

The benefits of sensitivity and specificity is that they both measure the true positive and true negative rates simply respectively. They both help to quantify how much a class was misclassified during evaluation. A drawback of these is that they can be difficult to interpret and defining a good sensitivity and specificity is dependent on the data being used.

Kappa

Kappa is beneficial when you need to quantify the agreement between two annotators of data, which is done by trying to adjust accuracy to account for the likelihood of a correct prediction by only chance. A drawback of Kappa is that it does not perform well with very skewed data sets. Even if observed agreement is relatively high, a skewed data set will make Kappa very low.

ROC Curve and AUC

The ROC (Receiver Operating Characteristics) curve shows the trade off between predicting true positives while avoiding false positives. It helps us visualize the trade off we are making. The AUC (area under the curve) is a measure of how well we've classified our data. A drawback of ROC Curve and AUC is that they're dependent on the order of probabilities not the probabilities themselves. So even if probabilities are changed, if the order remained unchanged, the ROC curve and AUC should stay the same. This means it can't be used to compare models.

Matthew's Correlation Coefficient

The MCC is beneficial since it accounts for the differences in class distribution, unlike accuracy. A drawback of this metric would be that its specific to binary classification only. MCC is also not suitable for imbalanced data sets.