Blog 4: Measurements of a Set

Michael O'Donnell

October 20, 2020

In Blog Three I explored the ANOVA analysis. In this Blog I will dive into Measurements of a Set including accuracy, precision, recall, and F1 score.

Now, start by loading a dataset This dataset contains data for all NBA teams from 2014-2018

```
nbaData <- read.csv("data/nba_data.csv")
colnames(nbaData)[1] <- "Team"
nbaData$WinningTeam <- nbaData$WinPercentage
nbaData$WinningTeam[nbaData$WinningTeam > .5] <- 1
nbaData$WinningTeam[nbaData$WinningTeam <= .5] <- 0
nbaData$PositivePtsDiff <- nbaData$Pts - nbaData$OppPts
nbaData$PositivePtsDiff[nbaData$PositivePtsDiff > 0] <- 1
nbaData$PositivePtsDiff[nbaData$PositivePtsDiff <= 0] <- 0

#head(nbaData, 1)
str(nbaData)
```

```
## 'data.frame':    214 obs. of  41 variables:
##  $ Team                     : Factor w/ 30 levels "Atlanta Hawks",..: 1 2 3 4 5 6 7 8 9 10 ...
##  $ Season                   : int  2018 2018 2018 2018 2018 2018 2018 2018 2018 2018 ...
##  $ SeasonType               : Factor w/ 2 levels "POFF","REG": 2 2 2 2 2 2 2 2 2 2 ...
##  $ Win                      : int  28 49 42 39 22 19 33 53 41 57 ...
##  $ Loss                     : int  53 33 40 43 60 63 48 28 40 25 ...
##  $ MatchCount               : int  81 82 82 82 82 82 81 81 81 82 ...
##  $ WinPercentage            : num  0.346 0.598 0.512 0.476 0.268 ...
##  $ Pts                      : num  113 112 112 111 105 ...
##  $ OppPts                   : num  119 108 112 112 113 ...
##  $ Pace                     : num  103.5 99 100.3 97.8 98.2 ...
##  $ OffEff                   : num  108 113 110 112 106 ...
##  $ DefEff                   : num  115 108 110 114 114 ...
##  $ EFgPercentage            : num  0.521 0.534 0.52 0.514 0.505 0.503 0.517 0.528 0.51 0.564 ...
##  $ OppEFgPercentage         : num  0.541 0.514 0.512 0.538 0.541 0.564 0.521 0.522 0.527 0.508 ...
##  $ TsPercentage             : num  0.555 0.567 0.556 0.554 0.541 0.54 0.554 0.558 0.545 0.596 ...
##  $ OppTsPercentage          : num  0.58 0.55 0.548 0.57 0.573 0.593 0.556 0.557 0.563 0.546 ...
##  $ RebRate                  : num  50.1 49.2 50.2 48.9 48 ...
##  $ EffPts                   : num  125 132 123 124 115 ...
##  $ OppEffPts                : num  138 120 127 132 133 ...
##  $ FastBreakPts             : num  15.3 16.2 11.6 11.7 12.1 ...
##  $ OppFBPts                 : num  16.5 13.2 11.8 13.3 13 ...
##  $ PointsInPaint            : num  51.2 44.8 48.8 46.8 50.8 ...
##  $ OppPointsInPaint         : num  49.4 45.9 51.2 49 49.1 ...
##  $ PointsOffTO              : num  21.1 14.8 17.4 13.6 16.6 ...
##  $ OppPointsOffTO           : num  16.9 18.1 15.4 16.1 15.2 ...
##  $ SecondChancePTS          : num  14.1 12.5 13.8 13 10.9 ...
##  $ OppSecondChancePTS       : num  14.5 13.5 14.4 13.4 13.4 ...
##  $ PersonalFoulsPG          : num  23.5 21.5 20.4 18.9 20.3 ...
##  $ OppPersonalFoulsPG       : num  22.1 22 19.5 20.6 18.7 ...
```

```
##  $ ShootingFoulsPG              : num  14.9 12.3 12.1 10.9 12 ...
##  $ ShootingFoulsDrawnPG         : num  12.6 13.4 10.5 12.3 11.2 ...
##  $ LessThnEightFeedUsage        : num  43.5 43.5 36.2 41.9 46.2 ...
##  $ EightToSixteenFeedUsage      : num  11.5 11.5 14.8 12 14.3 ...
##  $ SixteenToTwentyFourFeetUsage: num  4.8 4.89 10.9 8.39 10.05 ...
##  $ TwentyFourPlusFeetUsage      : num  39.9 40 38 37.2 29.2 ...
##  $ AvgShotDistance              : num  13.1 13.2 14 13.4 11.9 ...
##  $ OppAvgShotDistance           : num  13.3 12.9 13.5 13.2 13.2 ...
##  $ AvgMadeShotDistance          : num  10.34 10.7 11.64 10.96 9.58 ...
##  $ OppMadeAvgShotDis            : num  10.8 10.4 10.8 10.6 10.6 ...
##  $ WinningTeam                  : num  0 1 1 0 0 0 0 1 1 1 ...
##  $ PositivePtsDiff              : num  0 1 0 0 0 0 0 1 0 1 ...
```

To look at measurements of a set, we will let our model assume that a positive points differential implies the team will be a winning team (>50% win percentage)

```
# Loading the data
table(nbaData$PositivePtsDiff, nbaData$WinningTeam)
```

```
##
##      0  1
##   0 98 14
##   1 20 82
```

First, a function to look at the accuracy of our model. Accuracy is the % of correct predictions out of all predictions.

```
# write a function to calculate accuracy
accuracy <- function(df) {
  TruePositive <- nrow(df[df$PositivePtsDiff == 1 & df$WinningTeam == 1,])
  TrueNegative <- nrow(df[df$PositivePtsDiff == 0 & df$WinningTeam == 0,])
  FalsePositive <- nrow(df[df$PositivePtsDiff == 0 & df$WinningTeam == 1,])
  FalseNegative <- nrow(df[df$PositivePtsDiff == 1 & df$WinningTeam == 0,])

  acc <- round((TruePositive+TrueNegative)/
                 (TruePositive+TrueNegative+FalsePositive+FalseNegative), 3)

  return(acc)
}

accuracy(nbaData)
```

```
## [1] 0.841
```

Second, a function to look at the precision of our model. Precision is the % of True Positives out of all Positive predictions.

```
# write a function to calculate precision
precision <- function(df) {
  TruePositive <- nrow(df[df$PositivePtsDiff == 1 & df$WinningTeam == 1,])
  TrueNegative <- nrow(df[df$PositivePtsDiff == 0 & df$WinningTeam == 0,])
  FalsePositive <- nrow(df[df$PositivePtsDiff == 0 & df$WinningTeam == 1,])
```

```
  FalseNegative <- nrow(df[df$PositivePtsDiff == 1 & df$WinningTeam == 0,])

  prec <- round((TruePositive)/(TruePositive+FalsePositive), 3)

  return(prec)
}

precision(nbaData)
```

```
## [1] 0.854
```

Third, a function to look at the recall of our model. Recall is the % of True Positives out of all True Positives and False Negative predictions.

```
# write a function to calculate sensitivity
recall <- function(df) {
  TruePositive <- nrow(df[df$PositivePtsDiff == 1 & df$WinningTeam == 1,])
  TrueNegative <- nrow(df[df$PositivePtsDiff == 0 & df$WinningTeam == 0,])
  FalsePositive <- nrow(df[df$PositivePtsDiff == 0 & df$WinningTeam == 1,])
  FalseNegative <- nrow(df[df$PositivePtsDiff == 1 & df$WinningTeam == 0,])

  sens <- round((TruePositive)/(TruePositive+FalseNegative), 3)

  return(sens)
}

recall(nbaData)
```

```
## [1] 0.804
```

Last, a function to look at the F1 Score of our model. F1 Score is is blended metric of precision and recall

```
# write a function to calculate F1 Score
f_one_score <- function(df) {
  TruePositive <- nrow(df[df$PositivePtsDiff == 1 & df$WinningTeam == 1,])
  TrueNegative <- nrow(df[df$PositivePtsDiff == 0 & df$WinningTeam == 0,])
  FalsePositive <- nrow(df[df$PositivePtsDiff == 0 & df$WinningTeam == 1,])
  FalseNegative <- nrow(df[df$PositivePtsDiff == 1 & df$WinningTeam == 0,])

  f_one <- round((2*precision(df)*recall(df))/
                   (precision(df)+recall(df)), 3)

  return(f_one)
}

f_one_score(nbaData)
```

```
## [1] 0.828
```

Overall, the accuracy, precision, recall, and F1 Score were all over 80%.