

Stats 100: Final Project

Drew Silva

March 8, 2024

Acknowledgements

```
suppressPackageStartupMessages(library(BradleyTerry2))
suppressPackageStartupMessages(library(dplyr))
suppressPackageStartupMessages(library(glmnet))
suppressPackageStartupMessages(library(ggplot2))
suppressPackageStartupMessages(library(predtools))
suppressPackageStartupMessages(library(psych))

careerData = read.csv('NFLQBCareerStats.csv')
collegeData = read.csv('NFLQBCollegeStats.csv')

mergedPlayerData = merge(careerData, collegeData, by = "Player")

normalize_data = function(col) {
  df_mean = mean(col)
  df_sd = sd(col)
  return((col - df_mean) / df_sd)
}

filtered_df <- mergedPlayerData[mergedPlayerData$GS >= 10 & !is.na(mergedPlayerData$GS),
  ↪ ]

filtered_df$NormPA = normalize_data(filtered_df$Pass.Attempts)
filtered_df$NormCompPct = normalize_data(filtered_df$Comp..)
filtered_df$NormYds = normalize_data(filtered_df$Total.Yards)
filtered_df$NormY.A = normalize_data(filtered_df$Passing.Yards.Attempt)
filtered_df$Ratio = filtered_df$Passing.Touchdowns / filtered_df$Passing.Interceptions
filtered_df$NormRatio = normalize_data(filtered_df$Ratio)
filtered_df$NormPER = normalize_data(filtered_df$Passer.Efficiency.Rating)
filtered_df$NormConfRank = normalize_data(filtered_df$Conference.Rank)

model = lm(Rate ~ NormPA + NormCompPct + NormRatio + NormY.A + NormPER + NormConfRank,
  ↪ data = filtered_df)
#model = lm(Rate ~ NormPA + NormCompPct + NormYds + NormY.A + NormRatio + NormPER +
  ↪ NormConfRank, data = filtered_df)
```

```
summary(model)
```

```
##
## Call:
## lm(formula = Rate ~ NormPA + NormCompPct + NormRatio + NormY.A +
##     NormPER + NormConfRank, data = filtered_df)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -19.057  -5.445   2.393   6.056  14.417
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept)    86.595      2.384   36.324 2.96e-15 ***
## NormPA           5.133      2.751    1.866  0.0832 .
## NormCompPct    -8.608      7.460   -1.154  0.2679
## NormRatio       2.412      6.845    0.352  0.7298
## NormY.A       -23.869     16.814   -1.420  0.1776
## NormPER        33.987     24.994    1.360  0.1954
## NormConfRank    4.812      2.845    1.692  0.1128
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 10.92 on 14 degrees of freedom
## Multiple R-squared:  0.4181, Adjusted R-squared:  0.1687
## F-statistic: 1.677 on 6 and 14 DF,  p-value: 0.1992
```

```
test_data = read.csv('NFLQBProspectStats.csv')
```

```
test_data$NormPA = normalize_data(test_data$Pass.Attempts)
test_data$NormCompPct = normalize_data(test_data$Comp..)
test_data$NormYds = normalize_data(test_data$Total.Yards)
test_data$NormY.A = normalize_data(test_data$Passing.Yards.Attempt)
test_data$Ratio = test_data$Passing.Touchdowns / test_data$Passing.Interceptions
test_data$NormRatio = normalize_data(test_data$Ratio)
test_data$NormPER = normalize_data(test_data$Passer.Rating)
```

```
test_data$NormConfRank = normalize_data(test_data$Conference.Rank)
```

```
training_predictions = predict(model, filtered_df)
filtered_df$Predictions = training_predictions
training_data_qbs = filtered_df[order(-filtered_df$Predictions), ]
sorted_training_data = training_data_qbs[, c("Player", "Predictions")]
sorted_training_data
```

```
##              Player Predictions
## 29   Justin Herbert    101.14531
## 45   Tua Tagovailoa    99.18264
## 39   Patrick Mahomes    97.42913
## 2    Baker Mayfield    95.37520
## 17   Gardner Minshew    90.74524
## 9    Daniel Jones     90.57121
```

```
## 15    Dwayne Haskins    90.09106
## 25      Jordan Love    89.10970
## 31      Lamar Jackson  88.04427
## 24      Joe Burrow     87.37810
## 14      Drew Lock     86.53911
## 12    Deshaun Watson   86.38308
## 26      Josh Allen     86.14370
## 34      Mason Rudolph  83.77745
## 41      Sam Darnold    81.11431
## 21      Jalen Hurts    80.66939
## 27      Josh Rosen     80.15692
## 36 Mitchell Trubisky  79.79203
## 5       C.J. Beathard  77.60709
## 28      Joshua Dobbs   74.90679
## 13      DeShone Kizer  72.33828
```

```
predictions = predict(model, test_data)
test_data$Predictions = predictions

test_data = test_data[order(-test_data$Predictions), ]
test_data$Predictions = round(test_data$Predictions, 1)
sorted_test_data = test_data[, c("Player", "Predictions")]
new_df = test_data[, c("Player", "Predictions"), drop = FALSE]

sorted_test_data
```

```
##           Player Predictions
## 1    Caleb Williams    115.0
## 8    Michael Pratt     103.2
## 10   Austin Reed       100.0
## 5      Bo Nix           98.2
## 2    Drake Maye        91.9
## 4    JJ McCarthy       90.1
## 9    Jordan Travis     83.1
## 3    Jayden Daniels    79.5
## 7    Spencer Rattler   73.2
## 6    Michael Penix Jr.  73.0
## 12 Taulia Tagovailoa   70.9
## 11   Joe Milton III    61.1
```

```
#ggplot(data = test_data, aes(x= NormCompPct, y= Predictions, label=Player)) +
  ↪ geom_point() +
  #geom_text(vjust = 1, hjust = 0.5) +
  # xlim(-2.5, 1.5) +
  #ylim(60, 105)

ggplot(data = test_data, aes(x= Ratio, y= Weighted.Prediction)) + geom_point() +
  ↪ xlab("TD/Int Ratio") +
  ylab("Weighted Predicted Passer Rating")
```

