Predictive Modeling in College Basketball

By: Sarvesh Gopalakrishnan

MAIN GOALS

Predict Win Percentages

Which predictors are most useful in predicting win percentage?

Cross Validate Models

Which predictive model is most effective in predicting win percentage?

Predict T25 Teams for the Next Decade

How can the predicted win percentages from 2013-2024 be applied to a ranking system?

Predict the T25 Teams for the 2025 Season

How can we utilize the performance of teams in 2024 to predict the top teams in 2025?

DATASET INFORMATION

CBB Information

- Historical data for all CBB teams from 2013-2024
- Postseason, seeding, GP, and win data included

Outside Data

- Imported strength of schedule ratings (SOS_Ratings) from TeamRankings.com

Predictors:

ADJOE/ADJDE → Adjusted Off/Def Efficiency

EFG_O/EFG_D → Effective Off/Def Field Goal Percentage

TOR/TORD → Off/Def Turnover Rate

ORB/DRB → Off/Def Rebound Rate

FTR/FTRD → Off/Def Free Throw Rate

TWOPT/DTWOPT → Off/Def 2PT Percentage

OFFTH/DEFTH → Off/Def 3PT Percentage

BARTHAG/WAB → Power Rating/Wins Above Bubble

SOS_RATING → Strength of Schedule Rating

MODELS TESTED

MODEL 1

PREDICTOR: WAB

MODEL 2

PREDICTORS: WAB & TWOPT

MODEL 3

PREDICTORS: FTR, OFFTHR, BARTHAG

MODEL 4

PREDICTORS: SOS_RATING, ADJDE, EFG_O, DEFTH

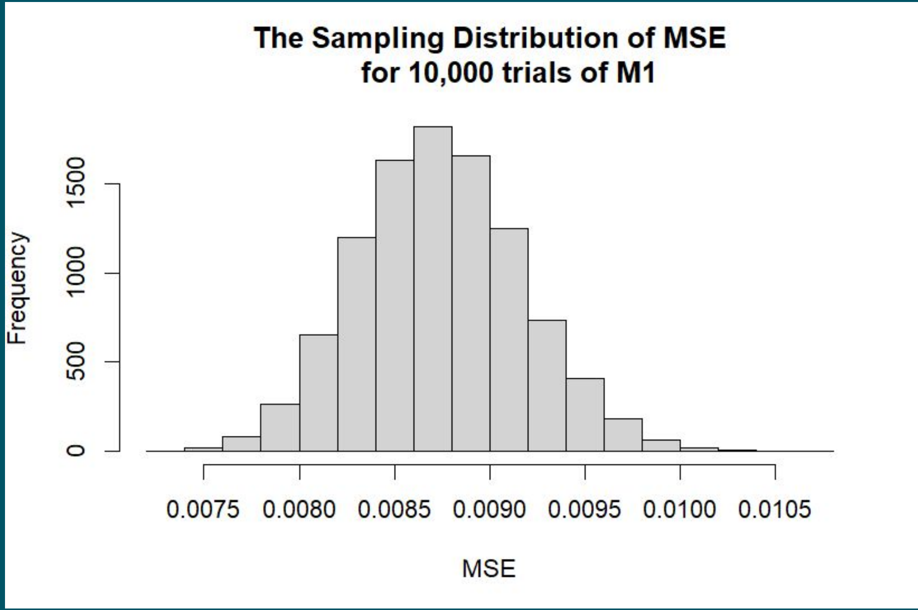
MODEL 5

PREDICTORS: ALL MINUS FTR/ADJOE/ADJDE

CORRELATIONS

```
> storeCor
     ADJOE      ADJDE      EFG_O      EFG_D        TOR       TORD 
 0.6879788 -0.6226684  0.6195404 -0.5787832 -0.4279623  0.1506454 
       ORB        DRB        FTR       FTRD      TWOPT     DTWOPT 
 0.2686537 -0.3821165  0.1157755 -0.2676139  0.5885508 -0.4923840 
     OFFTH      DEFTH SOS_RATING    BARTHAG        WAB 
 0.4215599 -0.4736091  0.4500161  0.7498545  0.8561277 
> storeRSqd
     ADJOE      ADJDE      EFG_O      EFG_D        TOR       TORD 
0.47331481 0.38771598 0.38383027 0.33499004 0.18315171 0.02269402 
       ORB        DRB        FTR       FTRD      TWOPT     DTWOPT 
0.07217479 0.14601300 0.01340397 0.07161719 0.34639205 0.24244201 
     OFFTH      DEFTH SOS_RATING    BARTHAG        WAB 
0.17771274 0.22430556 0.20251446 0.56228178 0.73295464 
> 
```

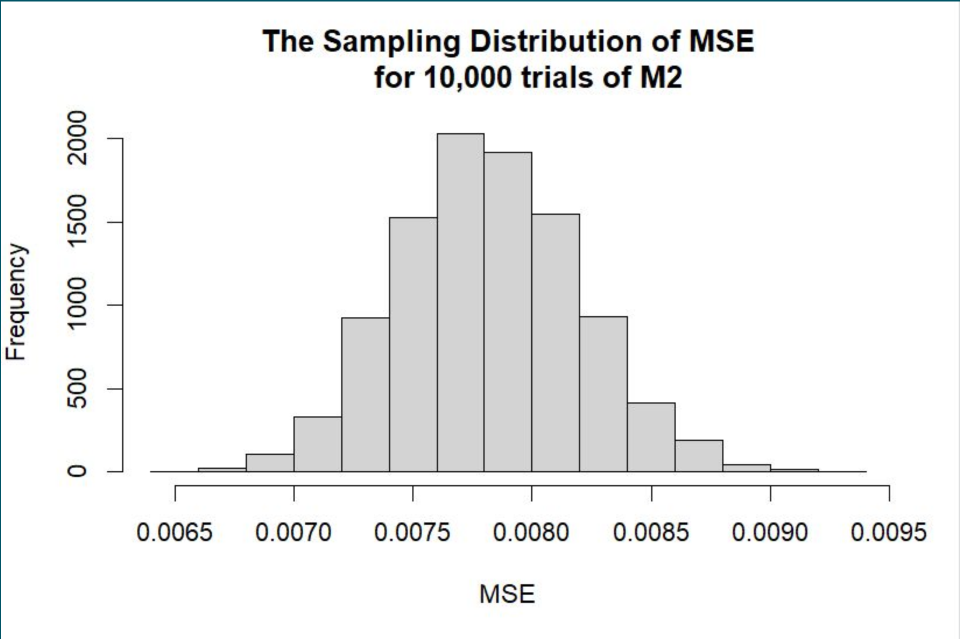
MODEL 1




```
> lmCheck1 <- lm(WINPER ~ WAB, data = cbb_data)
> summary(lmCheck1)$coefficients
             Estimate   Std. Error  t value Pr(>|t|)
(Intercept) 0.6905427 0.0022686966 304.3786        0
WAB         0.0226406 0.0002236726 101.2220        0
> mean(storecalcMSE1)
[1] 0.008739453
>
```

```
# MODEL 1 (PREDICTOR(S): WAB)

# Reason: WAB is selected due to its strong correlation with WINPER (0.8561277).
#   WAB also has the highest R^2 value (0.73295464)

lmCheck1 <- lm(WINPER ~ WAB, data = cbb_data)
summary(lmCheck1)$coefficients

# p-value for WAB = 0 < 0.05 (statistically significant model)

storecalcMSE1 <- c()

for (i in 1:10000)
{
  n1 <- sample(1:nrow(cbb_data), floor(0.8 * nrow(cbb_data)))
  train_data1 <- cbb_data[n1, ]
  test_data1 <- cbb_data[-n1, ]

  lm1 <- lm(WINPER ~ WAB, data = train_data1)
  storePred <- c(predict(lm1, test_data1[ , ]))

  indcalcMSE1 <- mean((test_data1[ , "WINPER"] - predict(lm1, test_data1[ , ]))^2)
  storecalcMSE1 <- c(storecalcMSE1, indcalcMSE1)
}
```

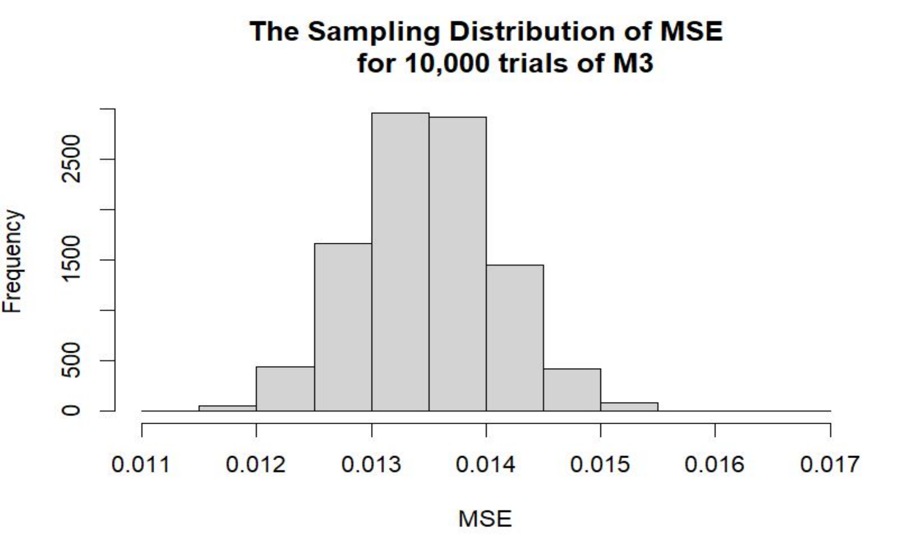
MODEL 2




```
> summary(lmCheck2)$coefficients
              Estimate   Std. Error   t value     Pr(>|t|)
(Intercept) 0.15009831 0.0255021764  5.885706 4.312891e-09
WAB         0.01992210 0.0002469267 80.680206 0.000000e+00
TWOPT       0.01055814 0.0004964495 21.267302 7.822854e-95
> mean(storecalcMSE2)
[1] 0.00781668
>
```

```
# MODEL 2 (PREDICTOR(S): WAB & TWOPT)

# WAB is highly correlated with WINPER and has a high R^2 value
# TWOPT is moderately correlated with WINPER
# WAB and TWOPT are only moderately correlated to one another
# Let's see the results

lmCheck2 <- lm(WINPER ~ WAB + TWOPT, data = cbb_data)
summary(lmCheck2)$coefficients

# p-value for WAB = 0 < 0.05 (statistically significant)
# p-value for TWOPT = 7.8e^-95 (statistically significant)

storecalcMSE2 <- c()

for (i in 1:10000)
{
  n2 <- sample(1:nrow(cbb_data), floor(0.80 * nrow(cbb_data)))
  train_data2 <- cbb_data[n2, ]
  test_data2 <- cbb_data[-n2, ]

  lm2 <- lm(WINPER ~ WAB + TWOPT, data = train_data2)
  storePred <- c(predict(lm2, test_data2[ , ]))

  indcalcMSE2 <- mean((test_data2[ , "WINPER"] - predict(lm2, test_data2[ , ]))^2)
  storecalcMSE2 <- c(storecalcMSE2, indcalcMSE2)
}
```

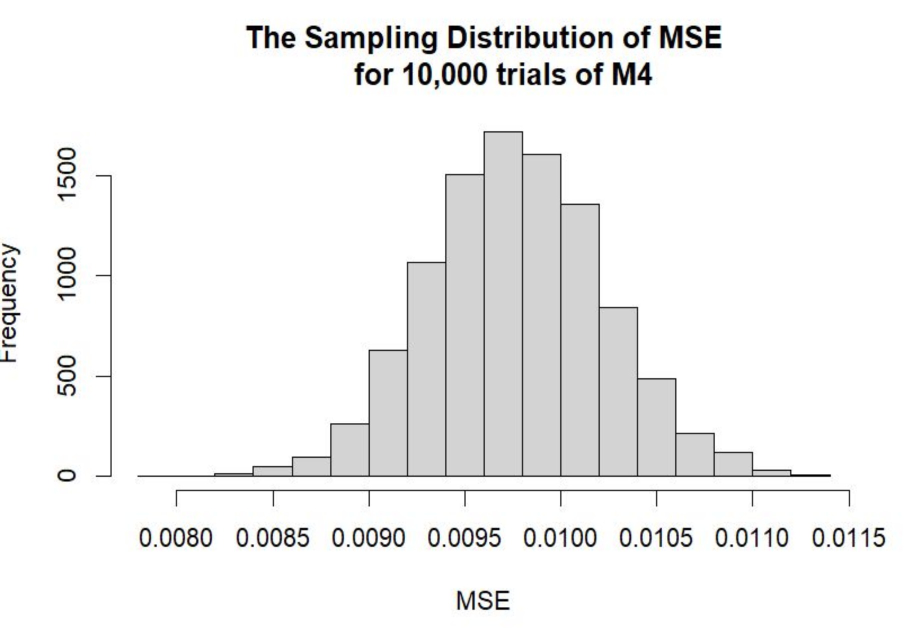
MODEL 3




```
> summary(lmCheck3)$coefficients
               Estimate   Std. Error   t value     Pr(>|t|)
(Intercept) -0.165737734 0.0278829802 -5.944047 3.035827e-09
BARTHAG      0.483798150 0.0080673041 59.970238 0.000000e+00
OFFTH        0.011422466 0.0007557374 15.114332 3.864904e-50
FTR          0.001571448 0.0003484754  4.509494 6.698028e-06
> mean(storecalcMSE3)
[1] 0.013486
>
```

```
# MODEL 3 (PREDICTOR(S): FTR, OFFTHREE, BARTHAG)

# OFFTH is moderately correlated to WINPER
# FTR is weakly correlated to WINPER (let's discover what happens)
# BARTHAG is strongly correlated to WINPER

lmCheck3 <- lm(WINPER ~ BARTHAG + OFFTH + FTR, data = cbb_data)
summary(lmCheck3)$coefficients

# p-value for BARTHAG = 0 < 0.05 (statistically significant)
# p-value for TWOPT = 3.86e-50 (statistically significant)
# p-value for FTR = 6.69e-06 (statistically significant)

storecalcMSE3 <- c()

for (i in 1:10000)
{
  n3 <- sample(1:nrow(cbb_data), floor(0.80 * nrow(cbb_data)))
  train_data3 <- cbb_data[n3, ]
  test_data3 <- cbb_data[-n3, ]

  lm3 <- lm(WINPER ~ FTR + OFFTH + BARTHAG, data = train_data3)
  storePred <- c(predict(lm3, test_data3[ , ]))

  indcalcMSE3 <- mean((test_data3[ , "WINPER"] - predict(lm3, test_data3[ , ]))^2)
  storecalcMSE3 <- c(storecalcMSE3, indcalcMSE3)
}
```

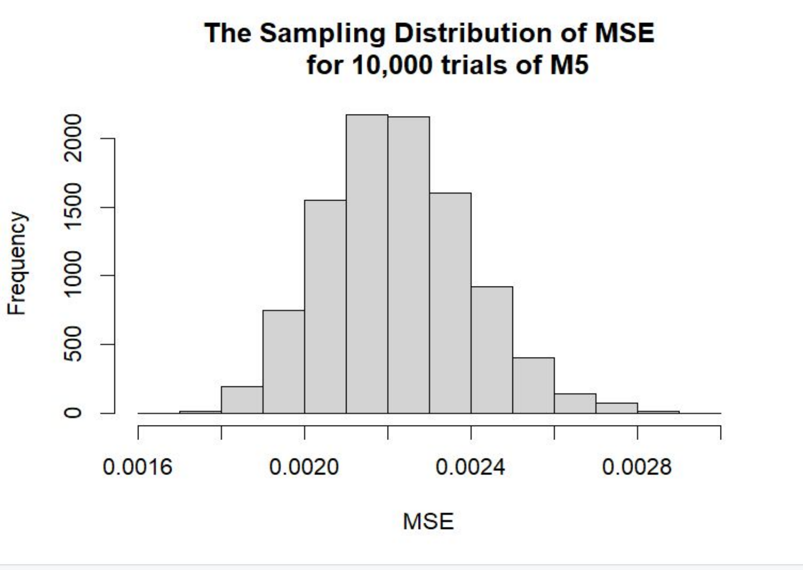
MODEL 4




```
> summary(lmCheck4)$coefficients
                Estimate   Std. Error   t value      Pr(>|t|)
(Intercept)  1.064072166 0.0465381887  22.86449 2.144926e-108
SOS_RATING  -0.010266308 0.0004976079 -20.63132  1.215777e-89
ADJDE       -0.019102891 0.0004728144 -40.40252 2.299400e-296
EFG_O        0.034577429 0.0005709929  60.55667  0.000000e+00
DEFTH       -0.008784501 0.0008475953 -10.36403  7.828413e-25
> mean(storecalcMSE4)
[1] 0.009768698
>
```

```
# MODEL 4 (PREDICTOR(S): SOS_RATING, ADJDE, EFG_O, DEFTH)

lmCheck4 <- lm(WINPER ~ SOS_RATING + ADJDE + EFG_O + DEFTH, data = cbb_data)
summary(lmCheck4)$coefficients

# p-value for SOS_RATING = 2.14e-108 < 0.05 (statistically significant)
# p-value for ADJDE = 2.3e-296 (statistically significant)
# p-value for EFG_O = 0 (statistically significant)
# p-value for DEFTH = 7.82e-25 (statistically significant)

storecalcMSE4 <- c()

for (i in 1:10000)
{
  n4 <- sample(1:nrow(cbb_data), floor(0.80 * nrow(cbb_data)))
  train_data4 <- cbb_data[n4, ]
  test_data4 <- cbb_data[-n4, ]

  lm4 <- lm(WINPER ~ SOS_RATING + ADJDE + EFG_O + DEFTH, data = train_data4)
  storePred <- c(predict(lm4, test_data4[ , ]))

  indcalcMSE4 <- mean((test_data4[ , "WINPER"] - predict(lm4, test_data4[ , ]))^2)
  storecalcMSE4 <- c(storecalcMSE4, indcalcMSE4)
}
```

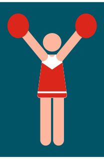
MODEL 5




```
> summary(lmCheck5)$coefficients
                Estimate   Std. Error    t value      Pr(>|t|)
(Intercept)  0.634429203 0.0296612758  21.389141  7.930374e-96
WAB          0.021565887 0.0003405105  63.333993  0.000000e+00
TWOPT        0.008077463 0.0003291990  24.536719 2.322944e-123
OFFTH        0.005431809 0.0003646511  14.895908  8.682735e-49
DEFTH       -0.006231346 0.0003991681 -15.610832  2.900348e-53
DTWOPT      -0.004758334 0.0003409663 -13.955437  3.507359e-43
TOR         -0.011023288 0.0004877425 -22.600628 4.341779e-106
TORD         0.009587046 0.0004512487  21.245595  1.219648e-94
BARTHAG      0.132626764 0.0143968060   9.212235  5.201870e-20
ORB          0.004298570 0.0002418355  17.774768  6.595975e-68
DRB         -0.006709478 0.0002886789 -23.242010 1.077060e-111
FTRD        -0.001652265 0.0001512980 -10.920598  2.378593e-27
SOS_RATING  -0.019674288 0.0004007806 -49.089923  0.000000e+00
> mean(storecalcMSE5)
[1] 0.002227097
> 
```

```
# MODEL 5 (PREDICTOR(S): ALL except FTR AND ADJOE/ADJDE

# FTR and ADJOE/ADJDE are not significant to the model

lmCheck5 <- lm(WINPER ~ WAB + TWOPT + OFFTH + DEFTH + DTWOPT + TOR + TORD + BARTHAG
                 + ORB + DRB + FTRD + SOS_RATING, data = cbb_data)
summary(lmCheck5)$coefficients

storecalcMSE5 <- c()

for (i in 1:10000)
{
  n5 <- sample(1:nrow(cbb_data), floor(0.80 * nrow(cbb_data)))
  train_data5 <- cbb_data[n5, ]
  test_data5 <- cbb_data[-n5, ]

  lm5 <- lm(WINPER ~ WAB + TWOPT + DEFTH + OFFTH + DTWOPT + TOR + TORD + BARTHAG +
              ORB + DRB + FTRD + SOS_RATING, data = train_data5)
  storePred <- c(predict(lm5, test_data5[ , ]))

  indcalcMSE5 <- mean((test_data5[ , "WINPER"] - predict(lm5, test_data5[ , ]))^2)
  storecalcMSE5 <- c(storecalcMSE5, indcalcMSE5)
}
```

CROSS VALIDATING MODELS

MODEL 1	MODEL 2	MODEL 3	MODEL 4	MODEL 5	BEST
					
PRED: WAB	PRED: WAB + TWOPT	PRED: FTR + OFFTH + BARTHAG	PRED: SOS_RATING + ADJDE + EFG_O + DEFTH	PRED: ALL EXCEPT ADJOE + ADJDE + FTR	MODEL 5
AVG MSE: 0.0087	AVG MSE: 0.0078	AVG MSE: 0.013	AVG MSE: 0.0098	AVG MSE: 0.0022	LOWEST AVG MSE

PREDICTING T25 TEAMS FOR THE NEXT DECADE

COMPONENT 1: AVG PRED WINPER

WEIGHT: 0.25

COMPONENT 2: AVG MM SCORE

WEIGHT: 0.50

COMPONENT 3: AVG SOS_RATING

WEIGHT: 0.25

DETAILS ON PREDICTION/RANKING

AVG PRED WINPER

The model which was the most effective in predicting win percentage was Model 5 which had the lowest MSE.

The win percentages for every team from 2013-2024 is averaged to create an average predicted win percentage.

AVG MM SCORE

Every team from 2013-2024 is given an individual score based on their March Madness Performance. This score is then averaged for every team.

CHAMPION → 10
2ND → 8
F4 → 7
E8 → 6
S16 → 4
R32 → 2
R64 → 1

AVG SOS_RATING

Strength of schedule ratings taken from TeamRankings.com.

Teams that face more difficult teams in the regular season have higher SOS_Ratings.

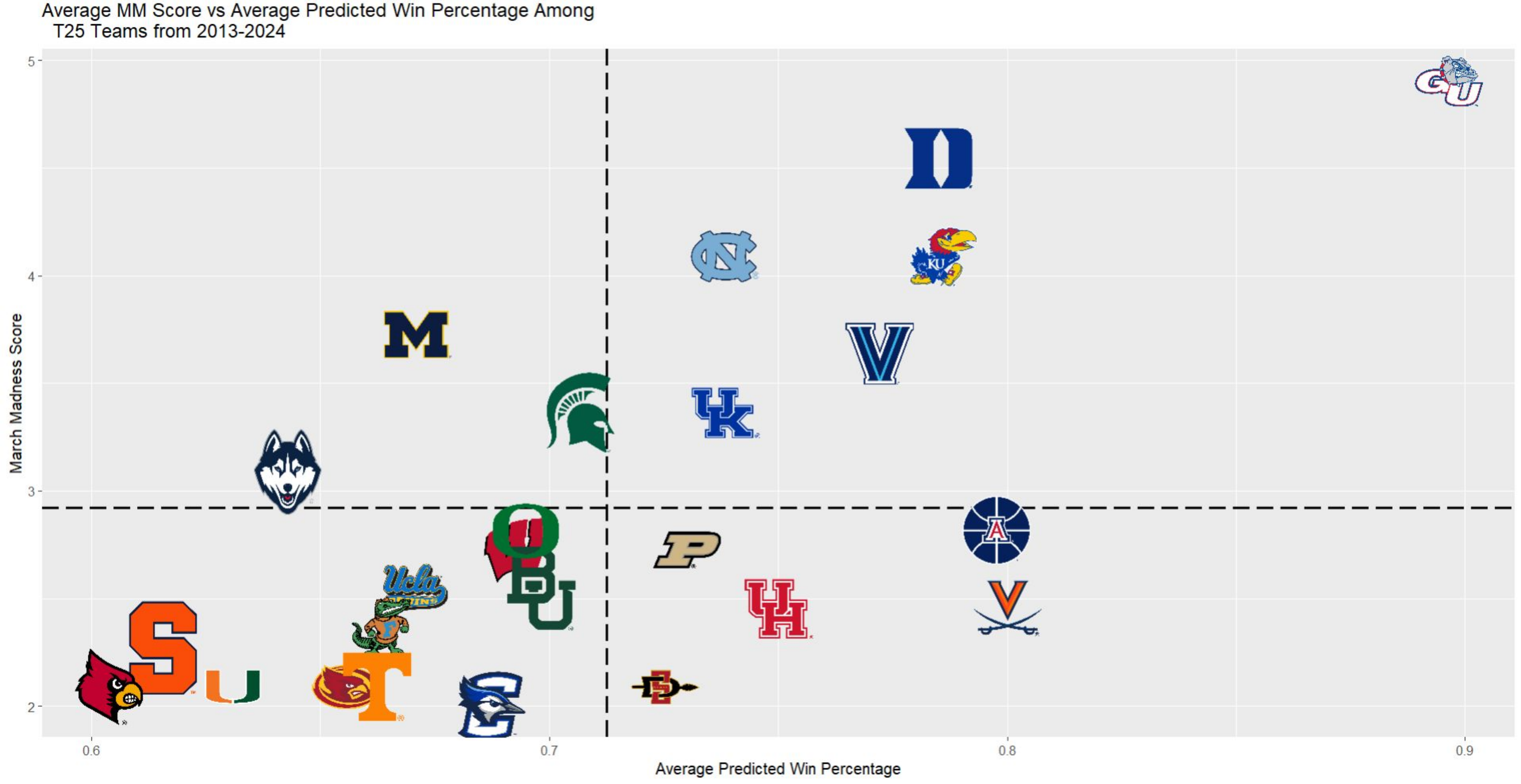
Teams that face less difficult teams have lower SOS_Ratings.

IMPLEMENTATION IN R

```r
# Store Pred Win Percentages
storeBestPred <- predict(lm5, cbb_data[, ])

# Assign Pred Win Percentages to cbb_data
for (i in 1:nrow(cbb_data))
{
  cbb_data$PREDWINPER[i] <- storeBestPred[i]
}

# Store every team uniquely from 2013-2024
uniqueTeams <- unique(c(cbb_data$TEAM))

# New data frame to store scores
pred_data <- data.frame(uniqueTeams)

# Calculate averages for win percentages
for (i in 1:nrow(pred_data))
{
  storePredWinPer <- c()
  teamIndex <- which(cbb_data$TEAM == uniqueTeams[i])
  storePredWinPer <- c(cbb_data[teamIndex, "PREDWINPER"])
  avgPredWinPer <- mean(storePredWinPer)
  pred_data$AVGWINPER[i] <- avgPredWinPer
}

# Add MM scores
for (i in 1:nrow(pred_data))
{
  storeMMSuc <- c()
  teamIndex <- which(cbb_data$TEAM == uniqueTeams[i])
  storeMMSuc <- c(cbb_data[teamIndex, "POSTSEASON_SCORE"])
  avgMMScore <- mean(storeMMSuc)
  pred_data$MMSCORE[i] <- avgMMScore
}
```

```r
# ADD SOS_RATING
for (i in 1:nrow(pred_data))
{
  storeSOS_RATING <- c()
  teamIndex <- which(cbb_data$TEAM == uniqueTeams[i])
  storeSOS_RATING <- c(cbb_data[teamIndex, "SOS_RATING"])
  avgSOS_RATING <- mean(storeSOS_RATING)
  pred_data$AVGSOSRATING[i] <- avgSOS_RATING
}

names(pred_data) <- c("TEAM", "AVG_PRED_WINPER", "MM_SCORE", "AVG_SOS_RTG")

pred_data$NORM_PRED_WINPER <- scale(pred_data$AVG_PRED_WINPER,
                                    center = min(pred_data$AVG_PRED_WINPER),
                                    scale = max(pred_data$AVG_PRED_WINPER) - min(pred_data$AVG_PRED_WINPER))

pred_data$NORM_MM_SCORE <- scale(pred_data$MM_SCORE,
                                 center = min(pred_data$MM_SCORE),
                                 scale = max(pred_data$MM_SCORE) - min(pred_data$MM_SCORE))

pred_data$NORM_SOS_RTG <- scale(pred_data$AVG_SOS_RTG,
                                center = min(pred_data$AVG_SOS_RTG),
                                scale = max(pred_data$AVG_SOS_RTG) - min(pred_data$AVG_SOS_RTG))

pred_data$WEIGHTED_SCORE <- (pred_data$NORM_PRED_WINPER * 0.25) + (pred_data$NORM_MM_SCORE * 0.5) +
  (pred_data$NORM_SOS_RTG * 0.25)

pred_data$WEIGHTED_SCORE <- pred_data$WEIGHTED_SCORE * 100

pred_data <- pred_data[order(pred_data$WEIGHTED_SCORE, decreasing = TRUE), ]

print(pred_data[1:25, ])
```

AVG MM SCORE VS AVG PRED WIN PER FOR T25 TEAMS FROM 2013-2024



RESULTS

```
> print(pred_data[1:25, ])
              TEAM AVG_PRED_WINPER MM_SCORE AVG_SOS_RTG NORM_PRED_WINPER NORM_MM_SCORE NORM_SOS_RTG WEIGHTED_SCORE
5          Gonzaga       0.8973164 4.909091    8.127273        1.0000000     1.0000000    0.8400057       96.00014
7             Duke       0.7837894 4.545455   10.827273        0.8392937     0.9259259    0.9660489       91.42986
15          Kansas       0.7854383 4.090909   11.554545        0.8416277     0.8333333    1.0000000       87.70736
1   North Carolina       0.7381881 4.090909   10.827273        0.7747413     0.8333333    0.9660489       85.18642
9         Villanova      0.7713590 3.636364    9.781818        0.8216973     0.7407407    0.9172443       80.51058
3         Michigan       0.6694208 3.727273   10.345455        0.6773959     0.7592593    0.9435564       78.48677
6         Kentucky       0.7378095 3.363636   10.163636        0.7742054     0.6851852    0.9350686       76.99111
21     Michigan St.      0.7056640 3.363636   10.718182        0.7287009     0.6851852    0.9609563       76.50069
18          Arizona      0.7978932 2.818182    8.627273        0.8592586     0.5740741    0.8633470       71.76884
14           Purdue      0.7291724 2.727273    9.845455        0.7619789     0.5555556    0.9202150       69.83263
2        Wisconsin       0.6926384 2.727273   10.645455        0.7102621     0.5555556    0.9575612       69.47336
8         Virginia       0.8000007 2.454545    9.463636        0.8622420     0.5000000    0.9023907       69.11582
10     Connecticut       0.6424587 3.090909    7.800000        0.6392288     0.6296296    0.8247277       68.08039
92           Baylor      0.6970983 2.545455   10.372727        0.7165754     0.5185185    0.9448295       67.46105
19           Oregon      0.6939245 2.818182    7.581818        0.7120828     0.5740741    0.8145424       66.86933
258            UCLA      0.6699897 2.545455    8.118182        0.6782011     0.5185185    0.8395813       63.87048
20          Florida      0.6618851 2.363636    9.627273        0.6667284     0.4814815    0.9100297       63.49303
62          Houston      0.7508535 2.454545    5.181818        0.7926702     0.5000000    0.7025039       62.37935
93        Iowa St.       0.6532115 2.090909   10.036364        0.6544503     0.4259259    0.9291272       60.88573
293      Tennessee       0.6618837 2.090909    9.081818        0.6667265     0.4259259    0.8845664       60.07862
95       Creighton       0.6854761 2.000000    8.554545        0.7001234     0.4074074    0.8599519       59.37225
25        Syracuse       0.6137034 2.272727    8.190909        0.5985235     0.4629630    0.8429764       59.18564
226  San Diego St.       0.7247838 2.090909    5.163636        0.7557665     0.4259259    0.7016551       57.73184
11      Louisville       0.6022213 2.090909    8.800000        0.5822695     0.4259259    0.8714104       57.63829
47        Miami FL       0.6291418 2.090909    7.981818        0.6203777     0.4259259    0.8332154       57.63613
>
```

WAY-TOO-EARLY TOP 25 FOR 2025 (INSPIRATION: ESPN)

2024-25 men's NCAA basketball rankings: Way-Too-Early Top 25


Take a look at the numbers and storylines behind ESPN's way-too-early men's basketball rankings for the 2024-25 season. (2:11)

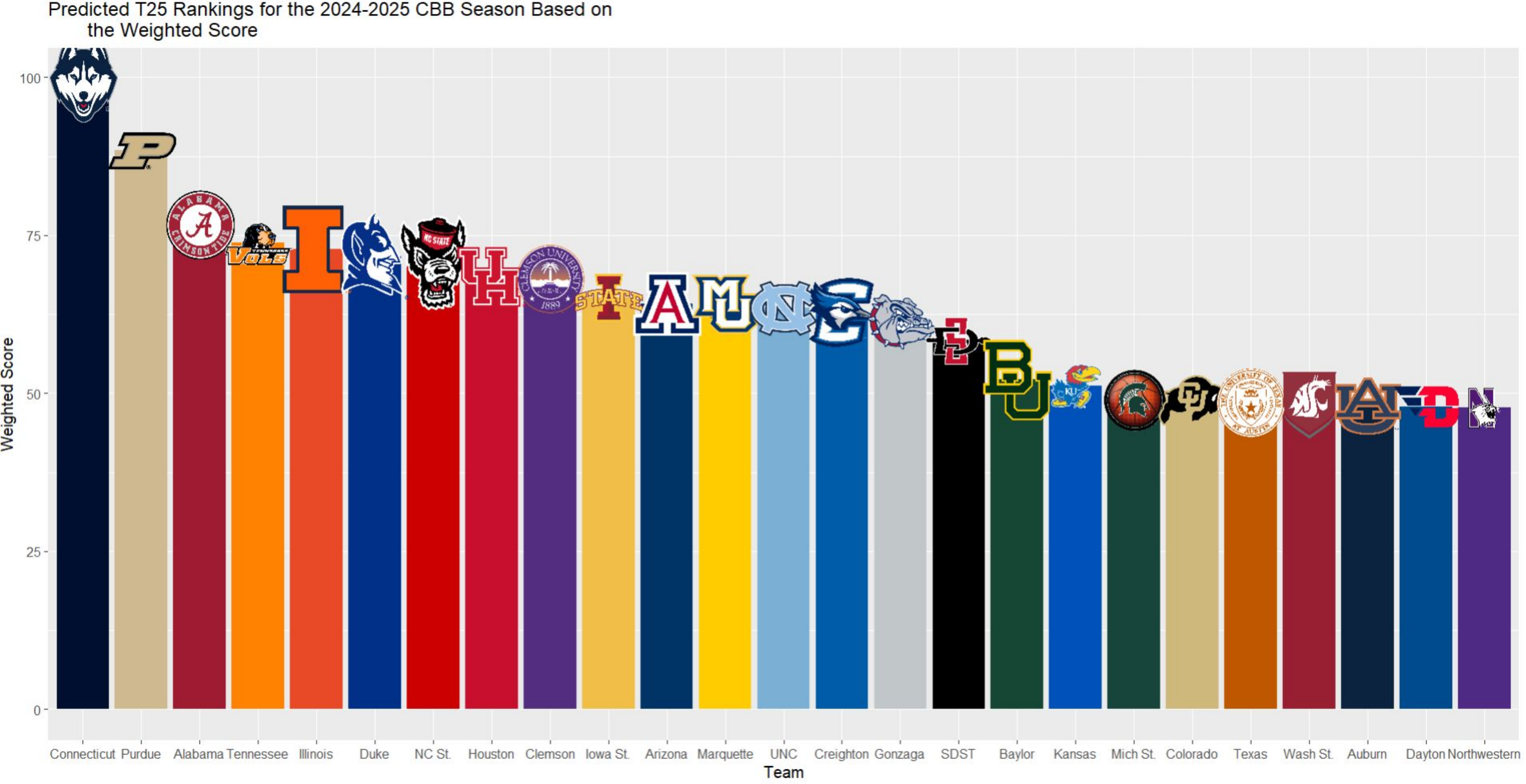
Jeff Borzello, ESPN Staff Writer
Apr 8, 2024, 11:30 PM ET

```r
# WAY TOO EARLY TOP 25 FOR 2025 (INSPIRATION: ESPN)

cbb_24 <- subset(cbb_data, YEAR == "2024")
cbb_24

predTop25 <- data.frame(cbb_24$TEAM, cbb_24$PREDWINPER, cbb_24$POSTSEASON_SCORE,
                        cbb_24$SOS_RATING)

names(predTop25) <- c("TEAM", "PRED_WINPER", "MM_SCORE", "SOS_RATING")

predTop25$NORM_PRED_WINPER <- scale(predTop25$PRED_WINPER,
                                    center = min(predTop25$PRED_WINPER),
                                    scale = max(predTop25$PRED_WINPER) - min(predTop25$PRED_WINPER))

predTop25$NORM_MM_SCORE <- scale(predTop25$MM_SCORE,
                                 center = min(predTop25$MM_SCORE),
                                 scale = max(predTop25$MM_SCORE) - min(predTop25$MM_SCORE))

predTop25$NORM_SOS_RATING <- scale(predTop25$SOS_RATING,
                                   center = min(predTop25$SOS_RATING),
                                   scale = max(predTop25$SOS_RATING) - min(predTop25$SOS_RATING))

predTop25$WEIGHTED_SCORE <- (predTop25$NORM_PRED_WINPER * 0.25) + (predTop25$NORM_MM_SCORE * 0.5) +
  (predTop25$NORM_SOS_RATING * 0.25)

predTop25$WEIGHTED_SCORE <- predTop25$WEIGHTED_SCORE * 100

predTop25 <- predTop25[order(predTop25$WEIGHTED_SCORE, decreasing = TRUE), ]

print(predTop25[1:25, ])
```

RESULTS

```
> print(predTop25[1:25, ])
                TEAM PRED_WINPER MM_SCORE SOS_RATING NORM_PRED_WINPER NORM_MM_SCORE NORM_SOS_RATING WEIGHTED_SCORE
2        Connecticut   0.9322644       10       13.0        1.0000000           1.0       0.9872340       99.68085
3             Purdue   0.8727027        8       13.3        0.9347052           0.8       1.0000000       88.36763
15           Alabama   0.6626188        7       12.4        0.7043999           0.7       0.9617021       76.65255
7          Tennessee   0.7477474        6       12.2        0.7977225           0.6       0.9531915       73.77285
12          Illinois   0.7473148        6       11.3        0.7972482           0.6       0.9148936       72.80355
11              Duke   0.7606984        6        9.9        0.8119201           0.6       0.8553191       71.68098
63 North Carolina St.  0.5871070        7        8.8        0.6216197           0.7       0.8085106       70.75326
1            Houston   0.9124082        4       12.4        0.9782325           0.4       0.9617021       68.49837
35           Clemson   0.6349624        6        9.8        0.6740814           0.6       0.8510638       68.12863
4           Iowa St.   0.8242889        4       11.5        0.8816314           0.4       0.9234043       65.12589
6            Arizona   0.8093084        4       11.0        0.8652089           0.4       0.9021277       64.18341
8          Marquette   0.7702595        4       11.9        0.8224014           0.4       0.9404255       64.07067
9     North Carolina   0.7891647        4       10.9        0.8431263           0.4       0.8978723       63.52497
10         Creighton   0.7593138        4       11.0        0.8104021           0.4       0.9021277       62.81324
14           Gonzaga   0.8008701        4        8.5        0.8559584           0.4       0.7957447       61.29258
29     San Diego St.   0.7050612        4        8.2        0.7509276           0.4       0.7829787       58.34766
17            Baylor   0.7139027        2       11.5        0.7606200           0.2       0.9234043       52.10061
16            Kansas   0.6872280        2       11.3        0.7313778           0.2       0.9148936       51.15678
23      Michigan St.   0.6084540        2       11.2        0.6450214           0.2       0.9106383       48.89149
27          Colorado   0.7021486        2        8.7        0.7477345           0.2       0.8042553       48.79975
20             Texas   0.6266845        2       10.4        0.6650067           0.2       0.8765957       48.54006
39    Washington St.   0.7186023        2        7.7        0.7657721           0.2       0.7617021       48.18685
5             Auburn   0.8112916        1        9.9        0.8673830           0.1       0.8553191       48.06755
40            Dayton   0.7685362        2        6.1        0.8205122           0.2       0.6936170       47.85323
31      Northwestern   0.6420373        2        9.3        0.6818373           0.2       0.8297872       47.79061
>
```

PREDICTED T25 RANKINGS FOR THE 2025 CBB SEASON BASED ON WEIGHTED SCORE



IMPROVEMENTS/FURTHER RESEARCH

1

CONSIDER RECRUITING CLASSES

Which teams are disadvantaged due to incoming recruiting classes?

2

NCAA TRANSFER PORTAL

Which teams have significantly improved from the last season based on additions from the transfer portal?

3

INJURIES

Currently, the model does not take into account rare cases such as injuries.

4

COACHING CHANGES

How do coaching changes play a role in team performance?