Data Mining final project

Lauren Stover

5/3/2021

###Report

#Introduction For my final project, I was interested in studying the NCAA March Madness tournament. ESPN provides various offensive and defensive statistics for each team for the tournament. I was curious to better understand the importance of these statistics in predictor the winner of a single game in the tournament. If there is a combination of statistics that is able to predict the likely winner with almost 100% accuracy. Below I have broken down some of the statistics that were included.

GP: Games played PTS: Points per game FGM-FGA: Field Goals Made-Attempted Per Game FG%: Field Goal Percentage 3PM-3PA: 3-Point Field Goals Made-Attempted Per Game 3P%: 3-Point Field Goal Percentage FTM-FTA: Free Throws Made-Attempted Per Game FT%: Free Throws Percentage OFF: Offensive Rebounds ORPG: Offensive Rebounds Per Game DEF: Defensive Rebounds DRPG: Defensive Rebounds Per Game REB: Rebounds RPG: Rebounds Per Game PPG: Points Per Game FGM: Field Goals Made Per Game FGA: Field Goals Attempted Per Game FGM: Field Goals Made FGA: Field Goals Attempted FG%: Field Goal Percentage 2PM: 2-Point Field Goals Made 2PA: 2-Point Field Goals Attempted 2P%: 2-Point Field Goal Percentage PPS: Points Per Shot FG%: Adjusted Field Goal Percentage FTM: Free Throws Made Per Game FTA: Free Throws Attempted Per Game FTM: Total Free Throws Made FTA: Total Free Throws Attempted FT%: Free Throw Percentage 3PM: 3-Point Field Goals Made Per Game 3PA: 3-Point Field Goals Attempted Per Game 3PM: Total 3-Point Field Goals Made 3PA: Total 3-Point Field Goals Attempted 3P%: 3-Point Field Goal Percentage 2PM: 2-Point Field Goals Made 2PA: 2-Point Field Goals Attempted 2P%: 2-Point Field Goal Percentage PPS: Points Per Shot FG%: Adjusted Field Goal Percentage AST: Assists APG: Assists Per Game TO: Turnovers TOPG: Turnovers Per Game AST/TO: Assists Per Turnover STL: Steals STPG: Steals Per Game TO: Turnovers TOPG: Turnovers Per Game PF: Personal Fouls ST/TO: Steals Per Turnovers ST/PF: Steals Per Personal Fouls BLK: Blocks PF: Personal Fouls BLKPG: Blocks Per Game BLK/PF: Blocks Per Fouls

#Data Analysis I have already cleaned all the data and gotten in ready for use in a python script. I will now join the data frames.

Now I will attempt to do an inner join for ncaa and threepoints. I will go through this process with all the data sets until there is one large data set of all the observations

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --

## v ggplot2 3.3.3     v purrr   0.3.4
## v tibble  3.1.0     v dplyr   1.0.5
## v tidyr   1.1.3     v stringr 1.4.0
## v readr   1.4.0     v forcats 0.5.1

## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

All of the data is now merged into one large file called ncaa.The file contains the various games played in the NCAA March Madness tournament from 2002-2019.

I would now like to do some comparative statistics on this. I know from some of the preliminary data cleaning that in the first round of the tournament, the higher seeded team (for example a 1 vs 16, the 1 is the higher seed) has over a 99% chance of winning. I am curious if all the other statistics for these games indicate that the higher seeded team is favored to win.

First thing I will do is split the data into a training and testing data set. I will set the split at 0.8.

Now I will fit the training data to a linear model. I will start with a very basic model that includes points per game and how accurate that statistic is at predicting the winner. I will use the PPGDIFF variable I created as well that indicates the differences in points per game as either positive or negative. Positive indicates that the home team had the higher points per game statistics, negative indicates the away team had the higher points per game statistic.

```
## (Intercept)       PPG.x       PPG.y     PPGDIFF
## -0.01156013  0.04286876 -0.03380405          NA
```

```
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type == :
## prediction from a rank-deficient fit may be misleading
```

```
## [1] 0.4992309
```

The RMSE for this model is around 0.6. I believe we can do much better than that.

The second model I will build will include all of the original model variables plus rebounds per game for each team.

```
##  (Intercept)        PPG.x        PPG.y      PPGDIFF        RPG.x        RPG.y
## -0.379904280  0.034484795 -0.033218468           NA  0.028137797 -0.002217857
```

```
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type == :
## prediction from a rank-deficient fit may be misleading
```

```
## [1] 0.5025033
```

Unfortunatley, the RMSE increased. This means that rebounds per game for each team is not a good predictor of the winner. I will remove this variable from future models.

For the third model, I will include variables form the first model as well as offensive rebounds per game for each team.

```
## (Intercept)       PPG.x       PPG.y     PPGDIFF      ORPG.x      ORPG.y
## -0.20122257  0.03313448 -0.03165801          NA  0.08925841 -0.02512675
```

```
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type == :
## prediction from a rank-deficient fit may be misleading
```

```
## [1] 0.5187139
```

Again, the RMSE has increased. This would indicate offensive rebounds per game is not a good indicator of winner.

For the fourth model, I will try something new. I hypothesize now the points per game and the difference is too good of an indicator of the winner for a model. I will now include the variables I did not think were good predictors up to this point: offensive rebounds per game, rebounds per game, as well as a new variable, turnovers per game

```
## (Intercept)      TOPG.x      TOPG.y      ORPG.x      ORPG.y       RPG.x
##  0.53597348 -0.25144024  0.12444802  0.14614373 -0.02155627  0.05002698
##       RPG.y
## -0.04355872
```

```
## [1] 0.5780046
```

While the RMSE is higher than the original models that included points per game, it is around 0.65. It did not increase much which would indicate to me that these are good predictors of the winner of a game. I will now work on further improving the model by including some other per game variables. I will include field goals attempted, defensive rebounds, free throws made, three points made per game, two points made per game, and assists per game.

```
##  (Intercept)       TOPG.x       TOPG.y       ORPG.x       ORPG.y        RPG.x
##  1.311271529 -0.287356786  0.161759132  2.316742685  1.808999373 -1.922768654
##        RPG.y        FGA.x        FGA.y       DRPG.x       DRPG.y     FTM.PG.x
## -1.921691939 -0.190782071  0.081303543  1.940348265  1.940798076  0.017152266
##     FTM.PG.y       X3PM.x       X3PM.y       X2PM.x       X2PM.y        APG.x
##  0.004149689  0.421836767 -0.188982551  0.007527454 -0.006583408 -0.003374367
##        APG.y
##  0.021767914
```

```
## [1] 0.7242301
```

The RMSE increased again from the fourth model. I will now include interactions to see if the RMSE can be decreased further.

For this model I will now take out the offensive and defensive rebounds per game as it seems repetitive of the total.

```
##   (Intercept)        TOPG.x        TOPG.y         RPG.x         RPG.y
##  -2.340827045  -0.082371875   0.316353194   0.129107425  -0.036092596
##         FGA.x         FGA.y      FTM.PG.x      FTM.PG.y      X3PM.1.x
##  -0.055132616   0.036875500   0.021316303   0.021774694   0.004749254
##      X3PM.1.y      X2PM.x.x      X2PM.y.y         APG.x         APG.y
##  -0.004433864   0.003115234  -0.004934916  -0.011771830   0.020455399
## TOPG.x:TOPG.y
##  -0.013452402
```

```
## [1] 0.6232739
```

The RMSE decreased significantly, which is great. This means the offensive and defensive rebounds were harming the model.

Below I will create the confusion matrix for out of sample performance.

```
##    yhat
## y    0  1
##   0  5 39
##   1  6 96
```

The false discovery rate is about 34%. The false positive rate is about 100%. This seems extremely inaccurate The true positive rate is about 92%.

#Results As we can see, the RMSE decreased each model until it resulted with an RMSE of around 0.68. This means that this linear model is the best prediction of the winner of a game in the NCAA March Madness tournament. for the different apartment buildings.

If we calculate the marginal effects of our model, one can see that the change in the rent and leasing rate is relatively the same across the model.

```
##   dydx_TOPG.x dydx_TOPG.y dydx_RPG.x   dydx_RPG.y   dydx_FGA.x  dydx_FGA.y
## 1 -0.04749247  0.02147662 0.02188913 -0.006119210 -0.009347293 0.006251945
## 2 -0.05428198  0.02425297 0.02420472 -0.006766544 -0.010336117 0.006913321
## 3 -0.04898966  0.03113073 0.02594303 -0.007252499 -0.011078428 0.007409817
## 4 -0.05151461  0.02737827 0.02571911 -0.007189900 -0.010982806 0.007345860
## 5 -0.06483023  0.03427677 0.03219954 -0.009001537 -0.013750140 0.009196794
## 6 -0.04303089  0.02228193 0.02072872 -0.005794811 -0.008851763 0.005920509
##   dydx_FTM.PG.x dydx_FTM.PG.y dydx_X3PM.1.x dydx_X3PM.1.y dydx_X2PM.x.x
## 1   0.003614008   0.003691725  0.0008051979 -0.0007517261  0.0005281630
## 2   0.003996324   0.004082262  0.0008903776 -0.0008312491  0.0005840359
## 3   0.004283329   0.004375439  0.0009543220 -0.0008909472  0.0006259797
## 4   0.004246358   0.004337673  0.0009460849 -0.0008832571  0.0006205766
## 5   0.005316312   0.005430635  0.0011844697 -0.0011058111  0.0007769431
## 6   0.003422418   0.003496014  0.0007625118 -0.0007118747  0.0005001633
##   dydx_X2PM.y.y   dydx_APG.x  dydx_APG.y
## 1 -0.0008366754 -0.001995819 0.003468049
## 2 -0.0009251849 -0.002206952 0.003834924
## 3 -0.0009916291 -0.002365449 0.004110338
## 4 -0.0009830700 -0.002345032 0.004074860
## 5 -0.0012307738 -0.002935909 0.005101601
## 6 -0.0007923205 -0.001890015 0.003284196
```

#Conclusion Overall, I would say that these statistics do a good job of predicting a winner of the NCAA tournament. I believe that there are better combinations possibly to predict the winner, but more data would be required. I tested separately interacting several terms and to my shock it increased the RMSE. I would assume this increase is due to the interaction counteracting the good statistic of the winner and slightly improving the statistic for the loser.

I think moving forward I would be interested in testing separate techniques. Perhaps simulating a tournament itself to see how many games it can accurately predict would be interesting.