BASKETBALL PREDICTION ANALYSIS OF MARCH MADNESS GAMES

CHRIS TSENG

YIBO WANG

GOAL OF PROJECT

- The goal is to predict the winners between college men's basketball teams competing in the 2018 (NCAA) 's "March Madness" tournament.
- The creation of these brackets has largely been an informal competition between friends and. family.
- There has increasingly been a monetary component for this competition, namely a prize for being the one who finishes with the most accurate March Madness bracket.
- As a result, figuring out the best way to accurately predict the bracket for that year has attracted considerable interest due to the difficulty of correctly predicting the outcome of 64 teams playing a total of 63 games.
- The most significant prize being Warren Buffet's offer of $1 million dollars a year for the rest of one's life if one is able to perfect predict the outcome of every March Madness game in an year.

PREDICTION INFORMATION

- In basketball, it has been determined that the future performance of a team will depend on many factors:
  - past performance of team
  - momentum in terms of win streaks
  - team standings
  - external environmental circumstances(travel schedule, home-court advantage, off-season days, etc)
- Overall, these prediction models attempt to predict a strong team vs a weak team. These give a rough estimate of what teams are likely to appear on the top of the bracket and which are on the lower end.

DATASET INFORMATION

- We’ll be using datasets from **Kaggle’s “Google Cloud & NCAA ML Competition 2018-Men’s**” machine learning competition.
- These datasets have a range of statistics including the outcomes from past NCAA March Madness tournaments every year since 1984, team seedings, team points scored per game, team standings during regular season, and team.
- The model is evaluated based on accuracy of how many correct outcomes it can predict for a certain year’s NCAA March Madness tournament.
- Better performing methods should predict the probability of two teams playing each other in the tournament for that year.

STEPS IN PROJECT

- 1. Decide which variables have the highest predictive power in determining which team is most likely to win a matchup in a given year.
  - Given the wide variety of data provided, deducing these variables can be accomplished using a number of machine learning techniques including decision trees, linear regression, and clustering.
- 2. Construct a model with these variables such that, when given two teams in a specific year, one will be able to predict which team is most likely to win if they play against each other in the NCAA tournament.
- 3. Try to predict an NCAA March Madness tournament bracket for a given year using the model we created

Data Collection

- Obtained datasets from Kaggle
- Data preprocessing accomplished using Python
- Variety of statistics calculated for each team that participated in March Madness since 2003 (first year that detailed statistics were collected for teams that season)
    - Total number of home, away, neutral wins
    - Average regular season stats, including three pointers made and attempted, blocks, steals, etc.
    - Point differential

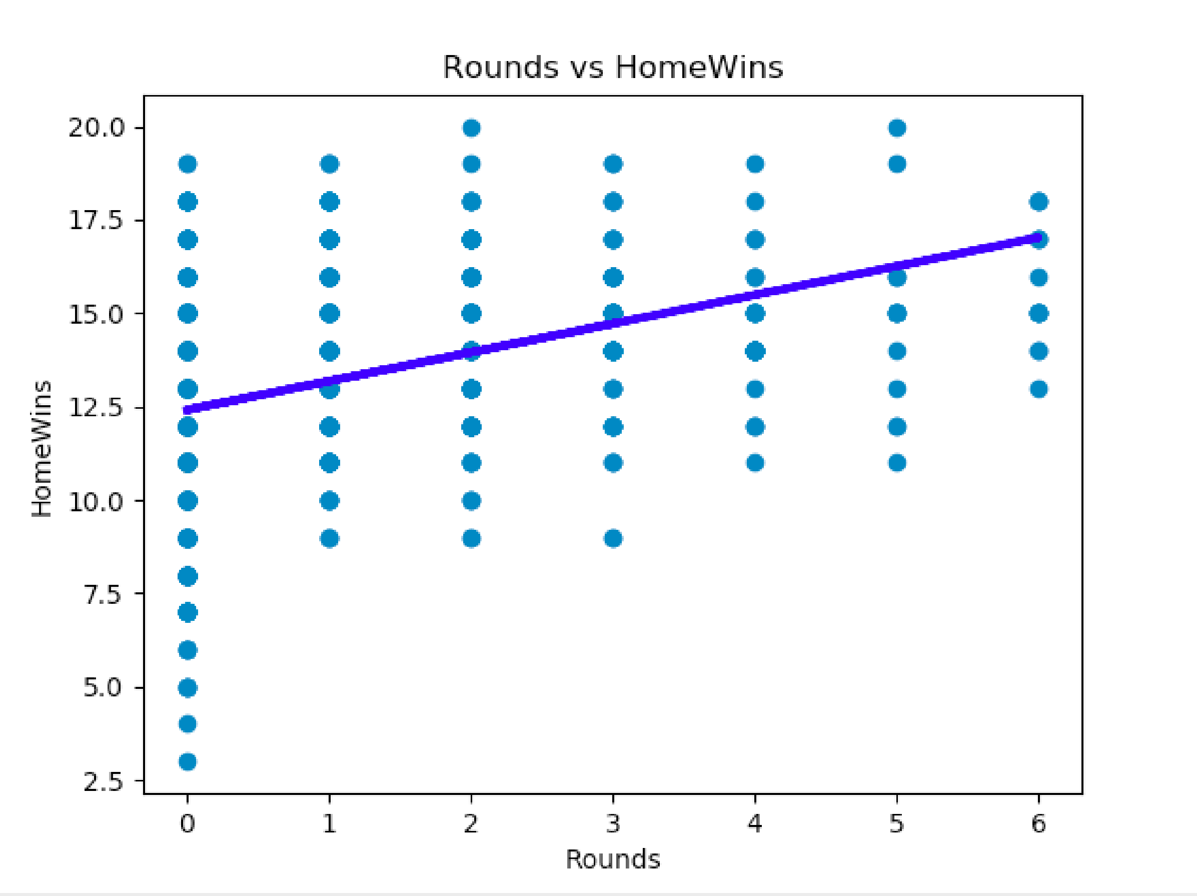
Linear Regression Results

ROUNDS VS HOME WINS

Coefficients: [0.76880701]

Mean squared error: 148.59

Variance score: -83.16

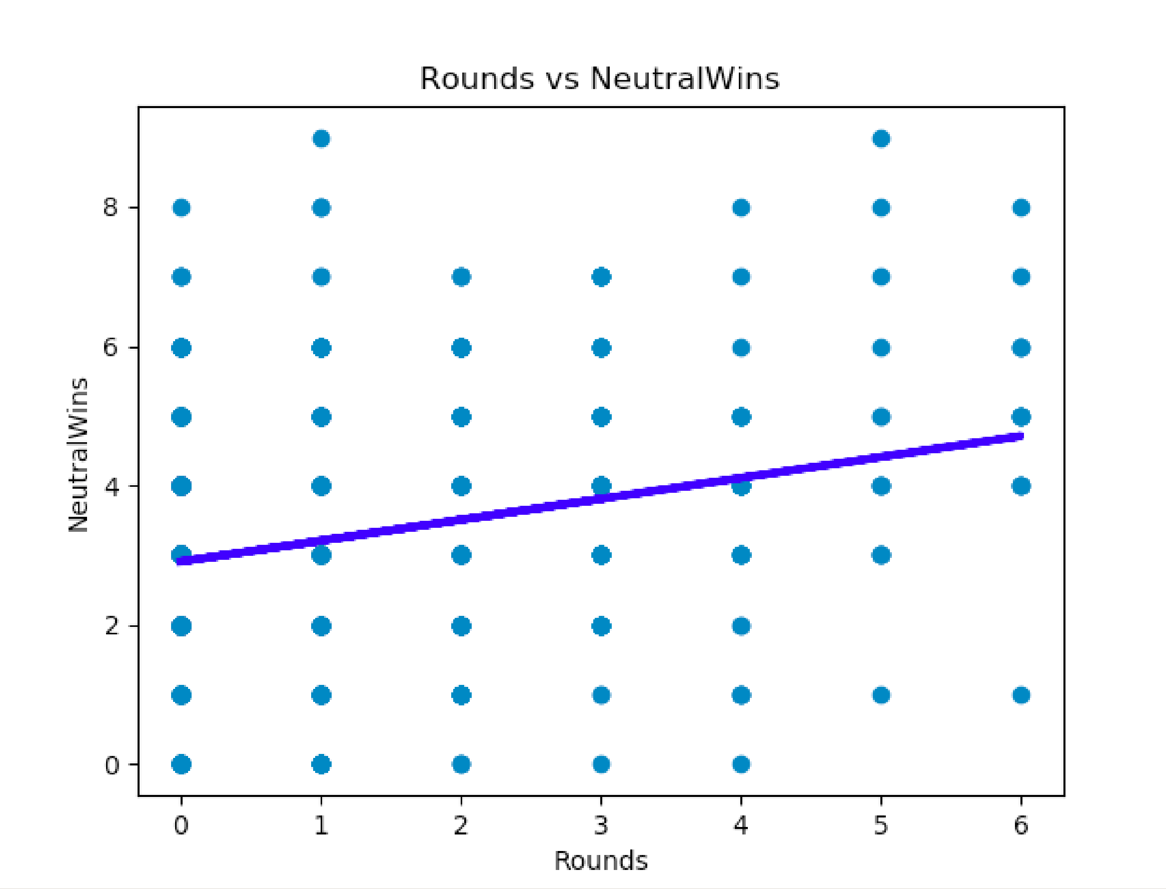


ROUNDS VS NEUTRAL WINS

Coefficients: [0.30054386]

Mean squared error: 5.87

Variance score: -2.32

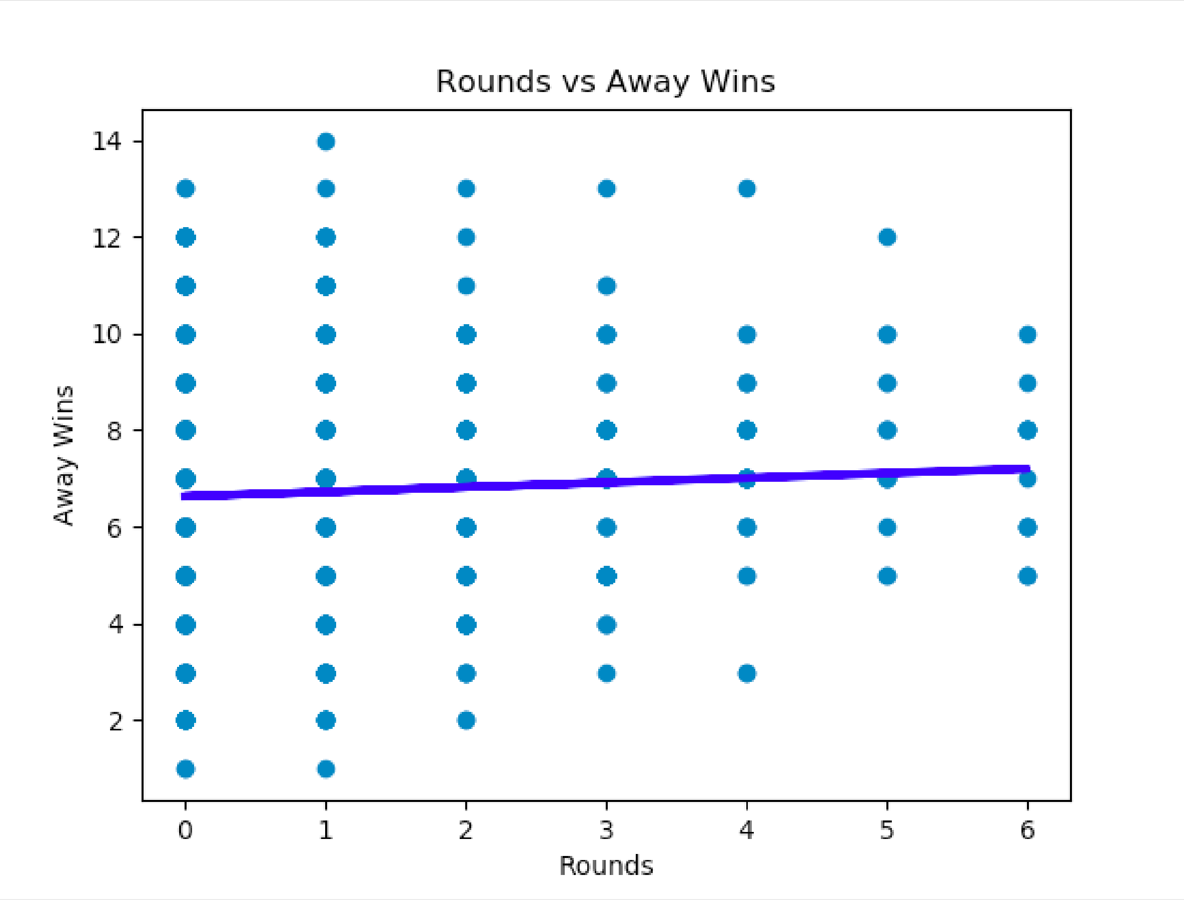


ROUNDS VS AWAY WINS

Coefficients: [0.09588975]

Mean squared error: 34.73

Variance score: -18.67

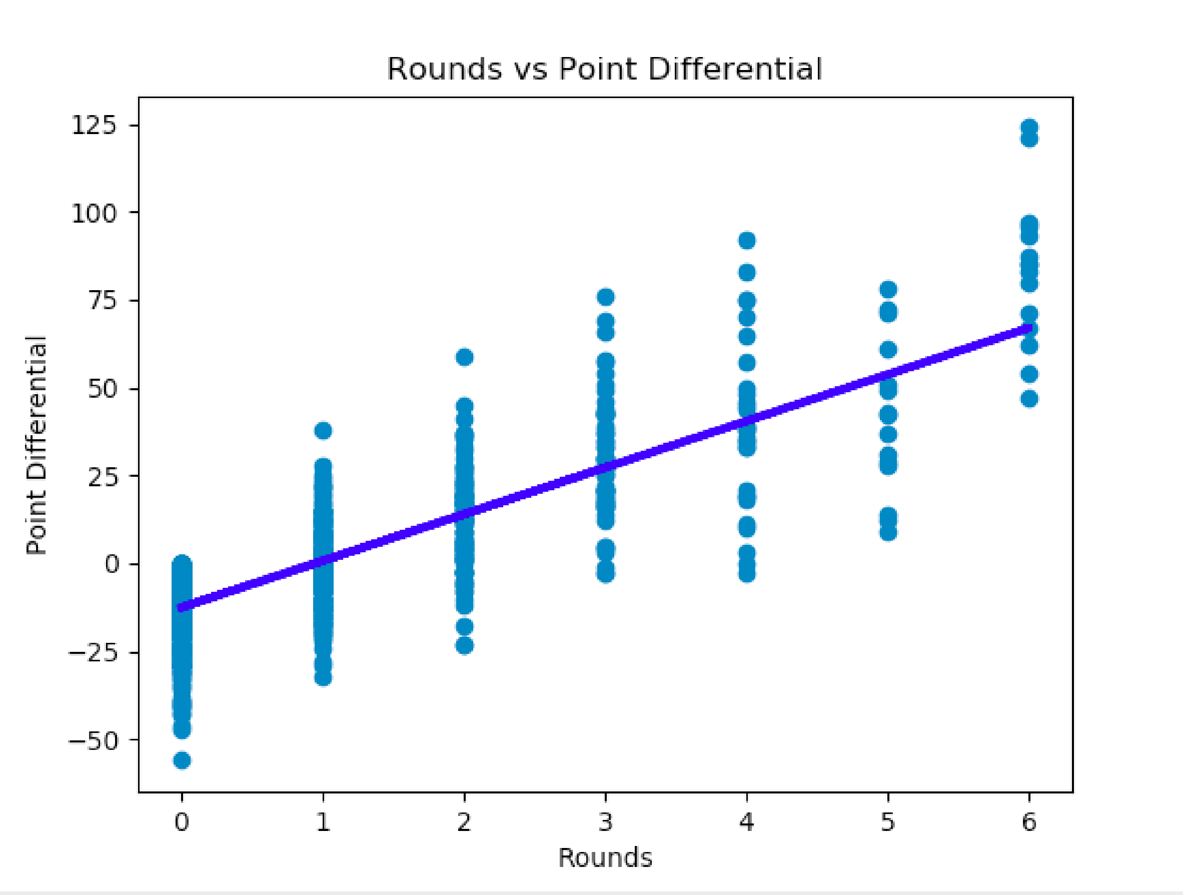


ROUNDS VS POINT DIFFERENTIAL

Coefficients: [13.24735648]

Mean squared error: 265.71

Variance score: -149.51

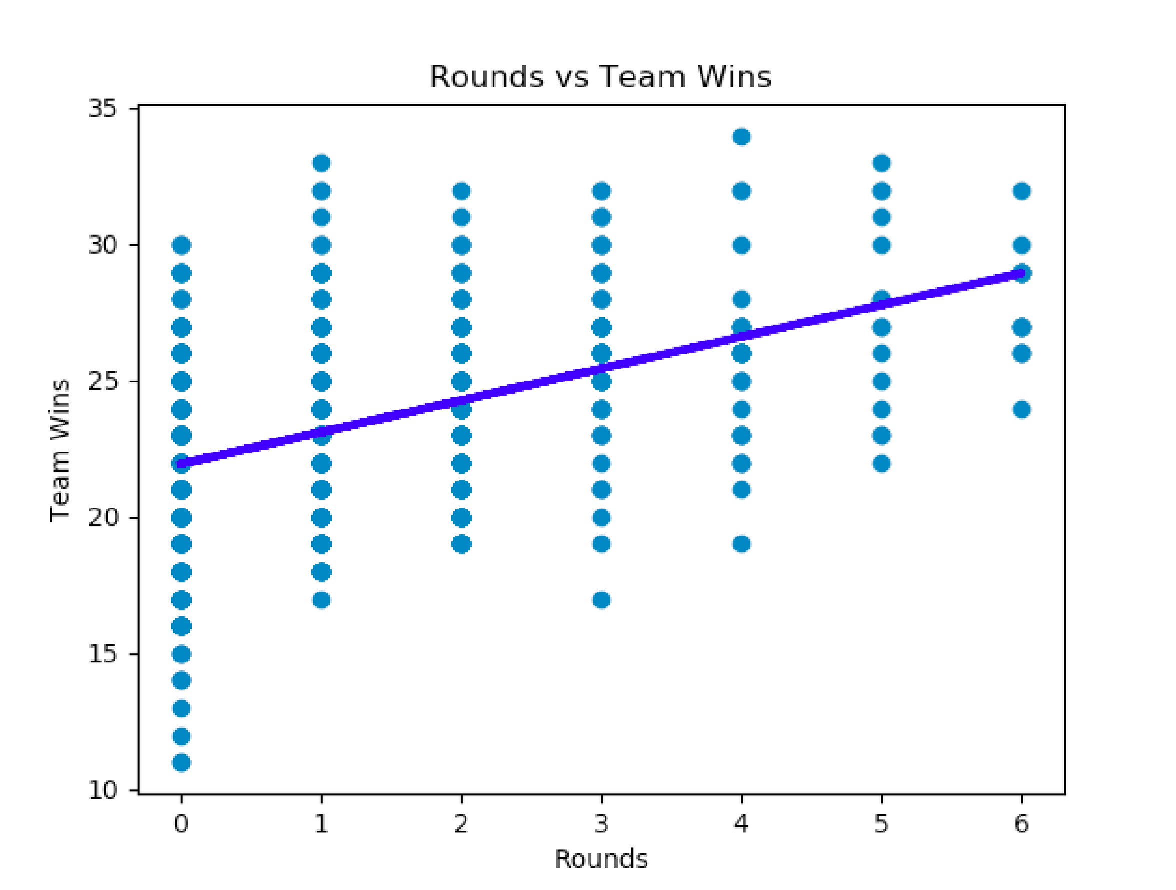


ROUNDS VS TEAM WINS

Coefficients: [1.16524063]

Mean squared error: 487.99

Variance score: -275.41

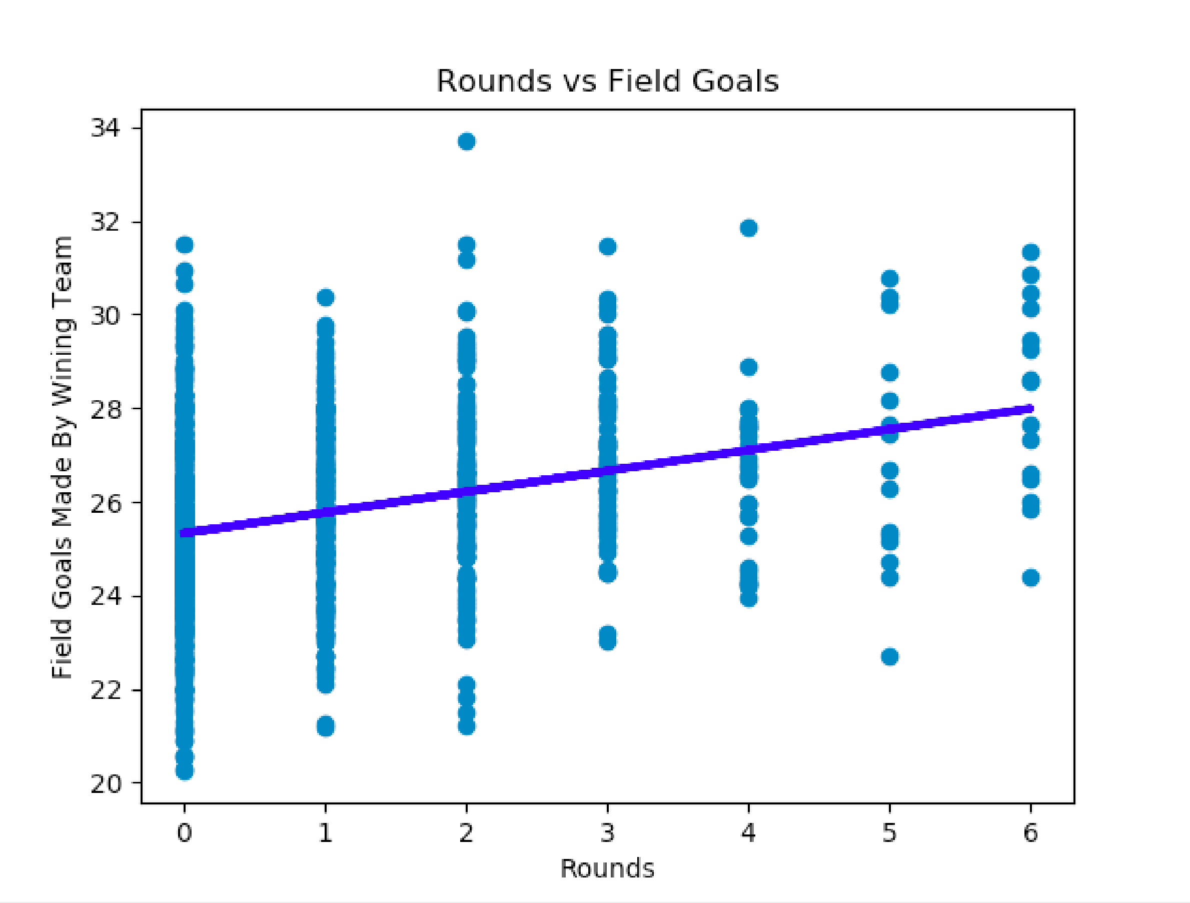


ROUNDS VS FIELD GOALS MADE

Coefficients: [0.44359187]

Mean squared error: 615.39

Variance score: -347.57

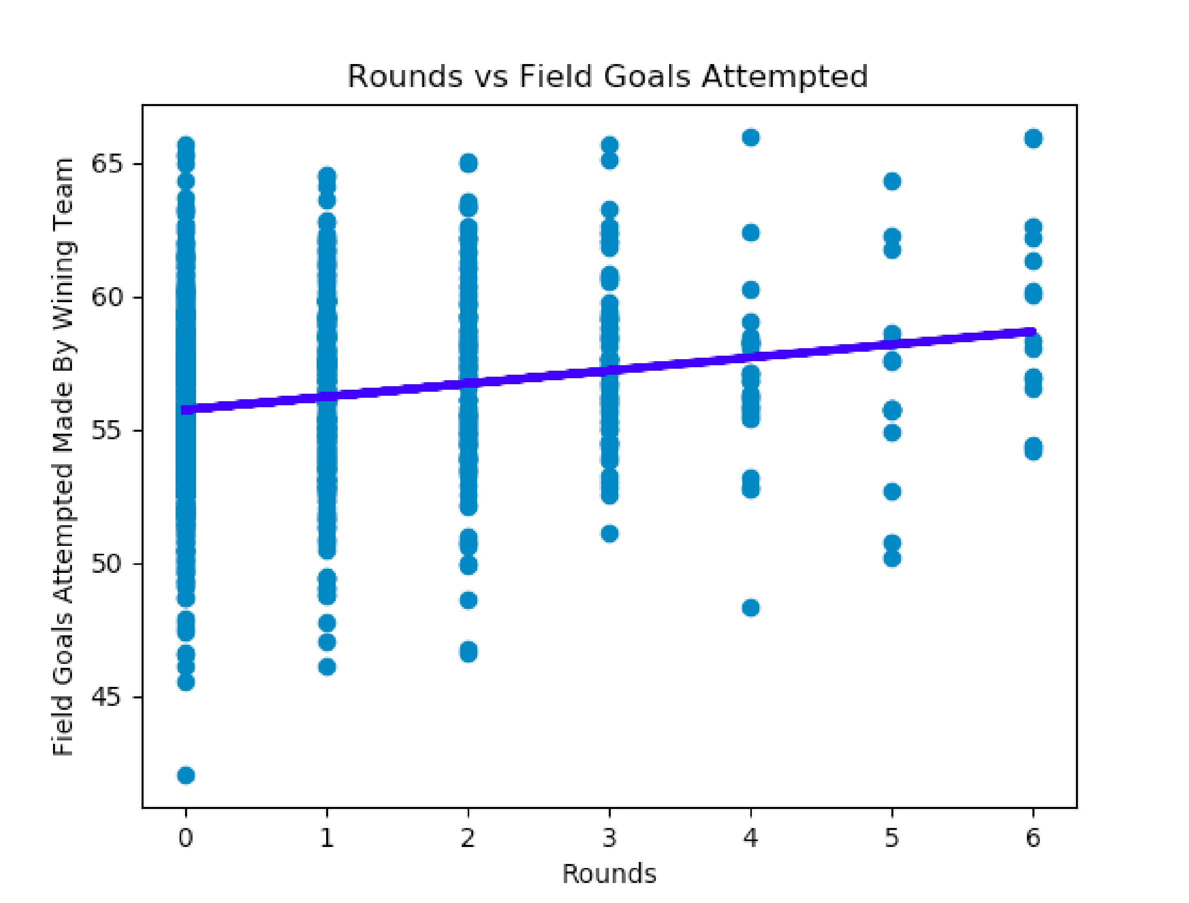


ROUNDS VS FIELD GOALS ATTEMPTED

Coefficients: [0.48955215]

Mean squared error: 3054.05

Variance score: -1728.90



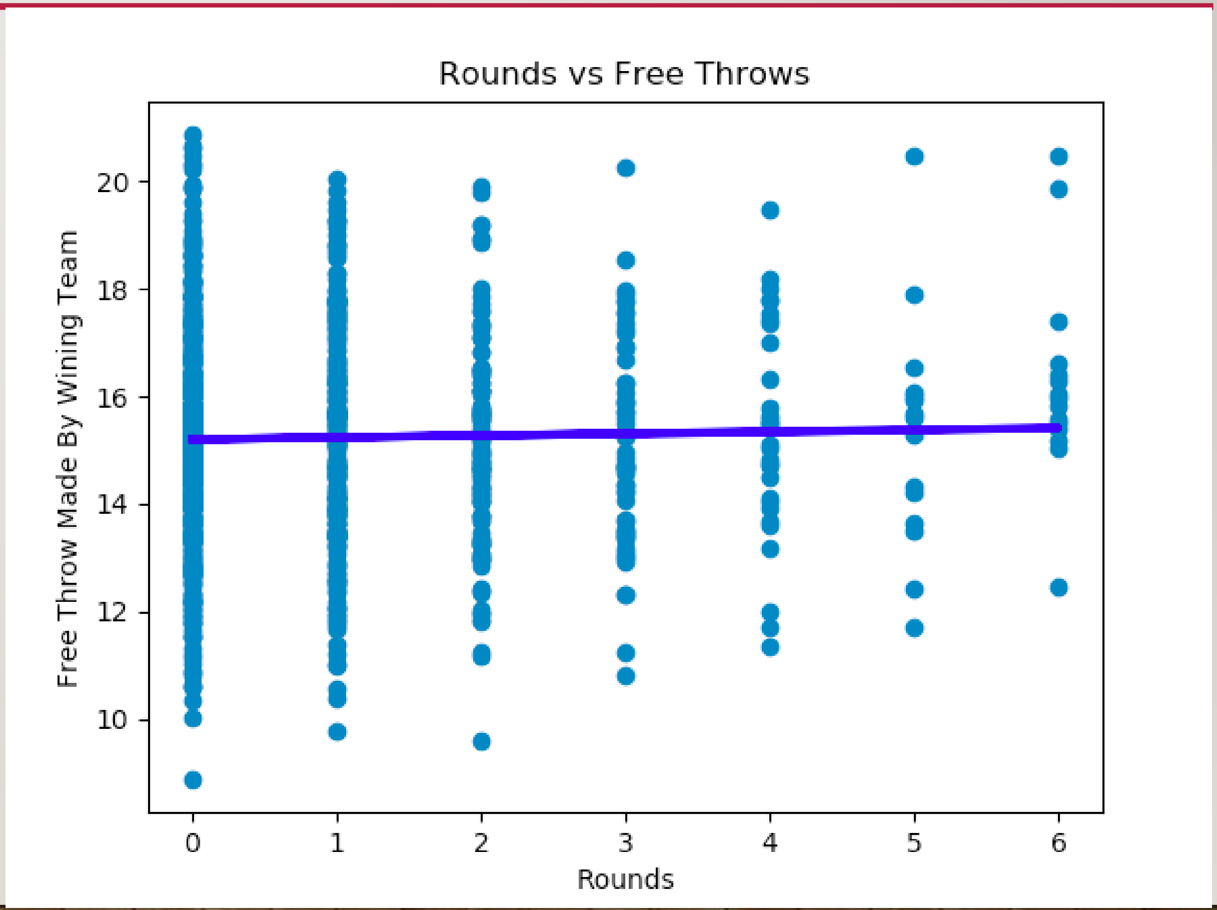
ROUNDS VS FREE THROWS MADE

Coefficients:

[[0.03517384]]

Mean squared error: 205.68

Variance score: -115.50



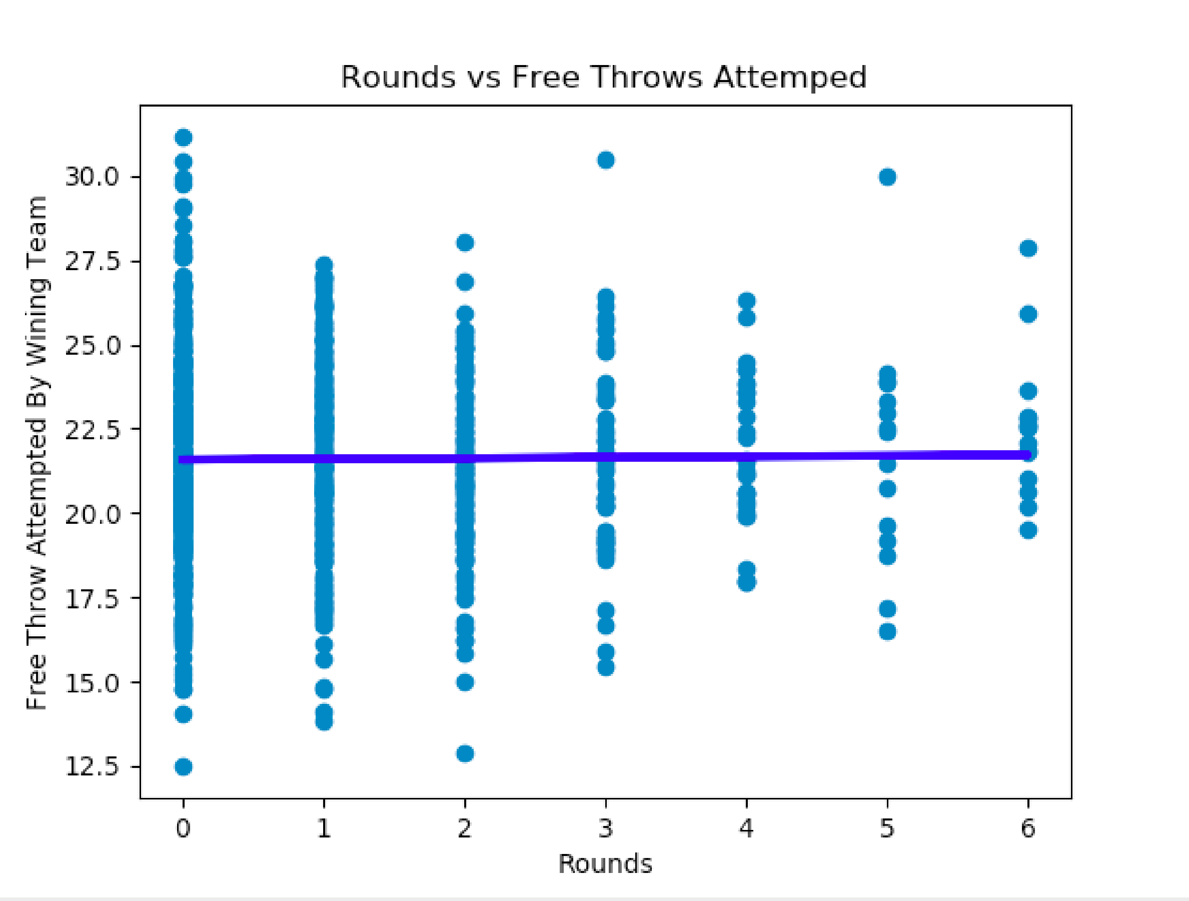
ROUNDS VS FREE THROWS ATTEMPTED

Coefficients:

[[0.02434854]]

Mean squared error: 428.21

Variance score: -241.55



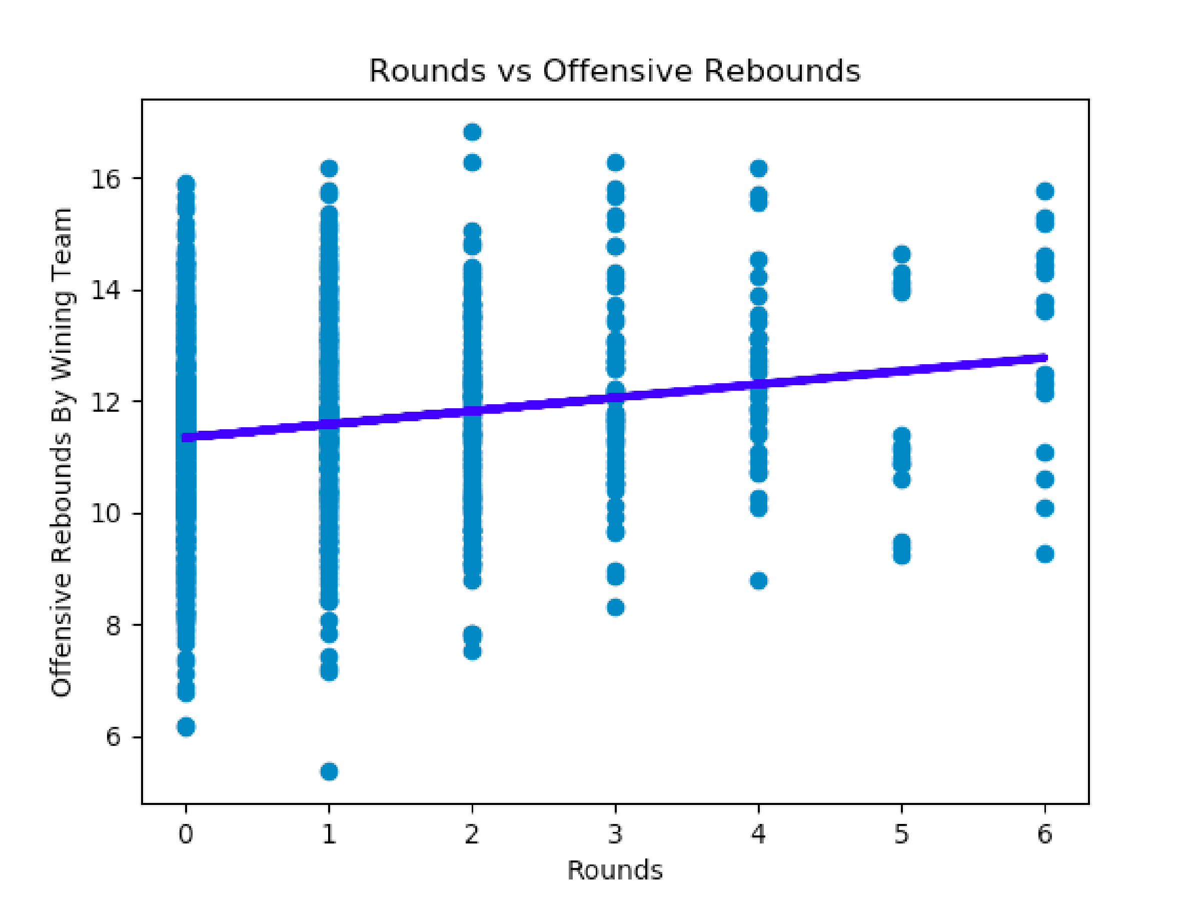
ROUNDS VS OFFENSIVE REBOUNDS

Coefficients:

[[0.23840846]]

Mean squared error: 113.75

Variance score: -63.43



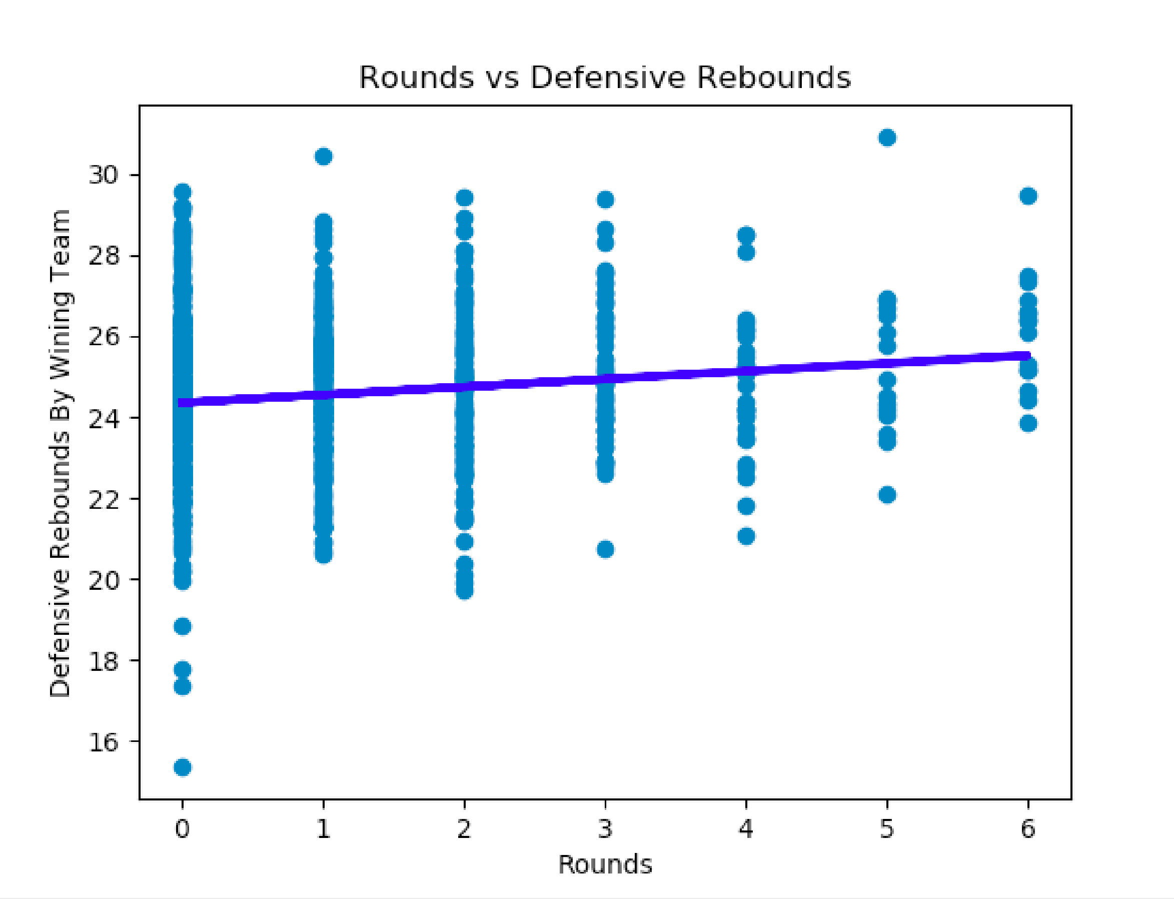
ROUNDS VS DEFENSIVE REBOUNDS

Coefficients:

[[0.19470072]]

Mean squared error: 557.56

Variance score: -314.82



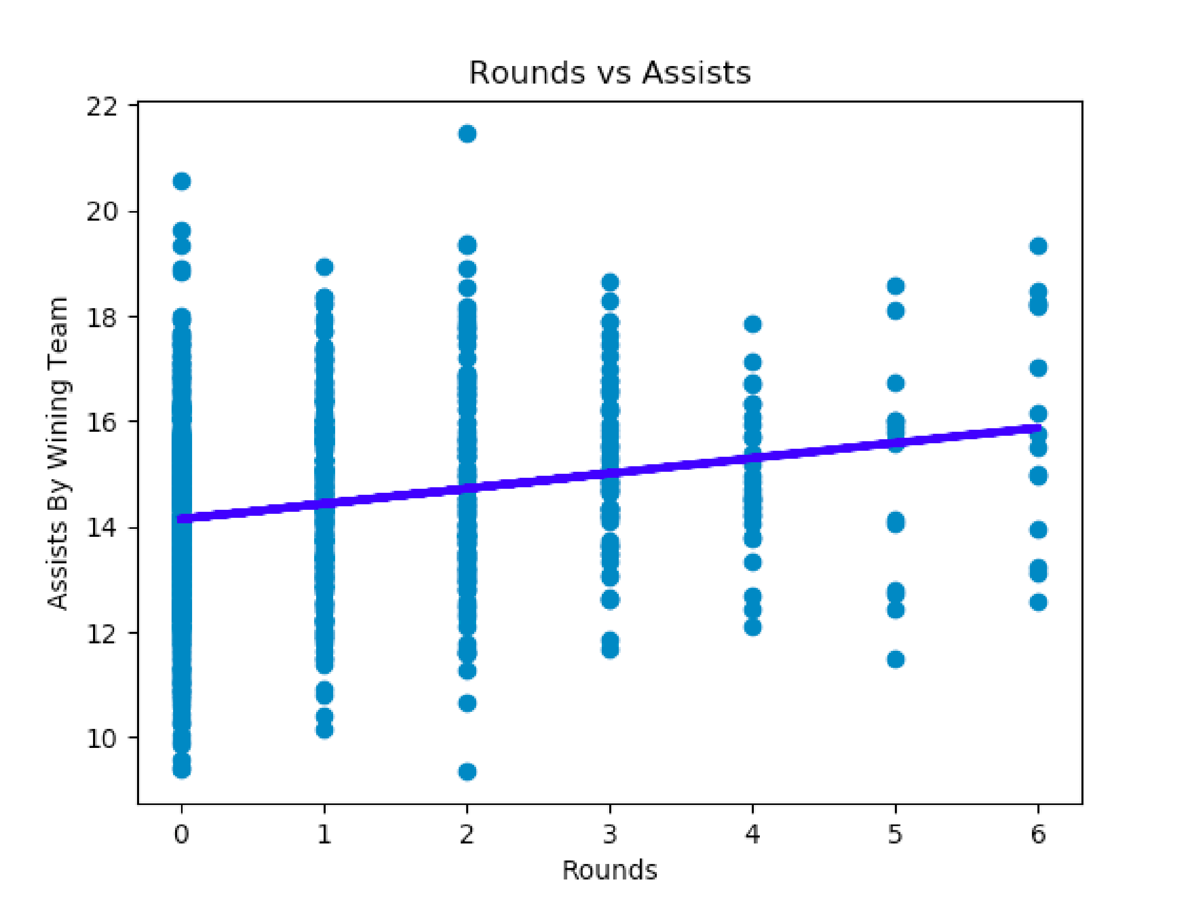
ROUNDS VS ASSISTS

Coefficients:

[[0.28843754]]

Mean squared error: 182.25

Variance score: -102.23



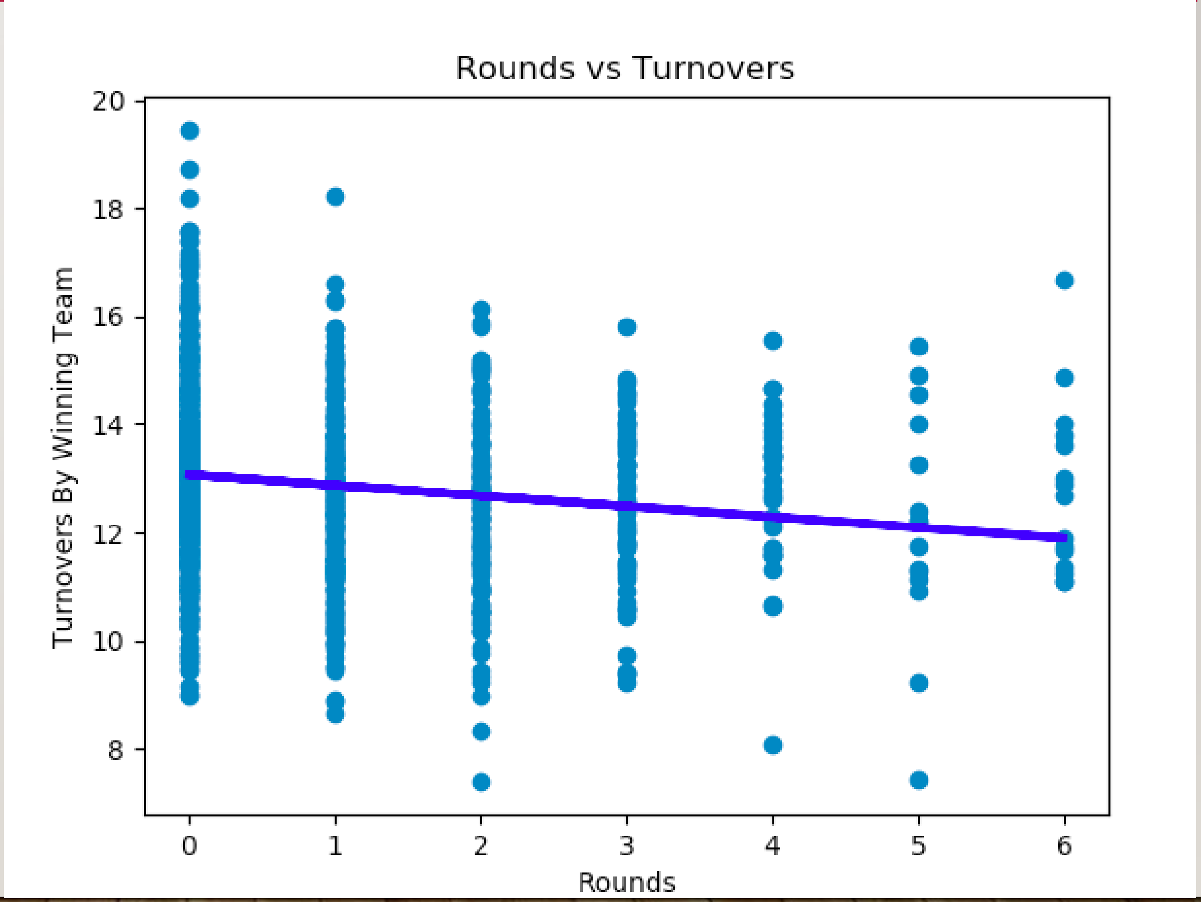
ROUNDS VS TURNOVERS

Coefficients:

[[-0.19625423]]

Mean squared error: 145.03

Variance score: -81.15



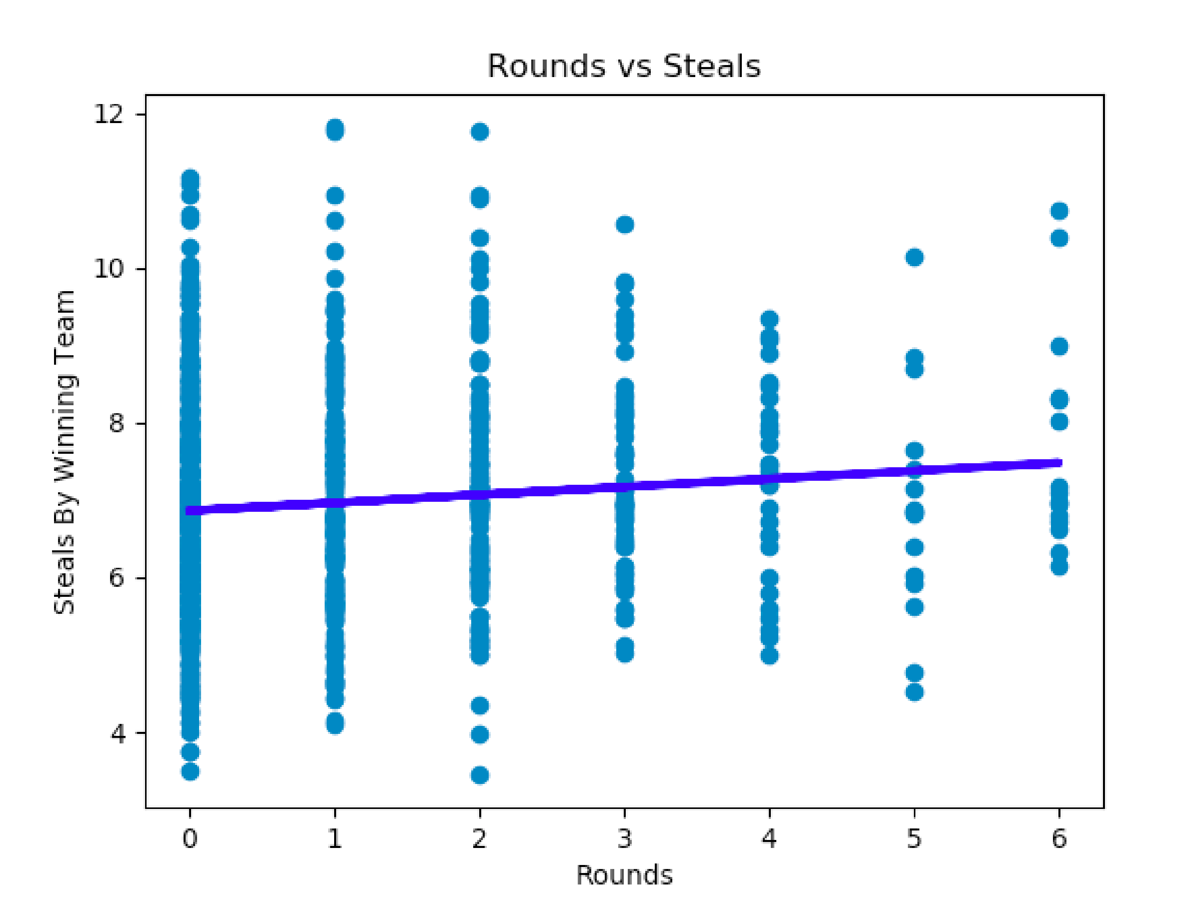
ROUNDS VS STEALS

Coefficients:

[[0.10295244]]

Mean squared error: 37.52

Variance score: -20.25



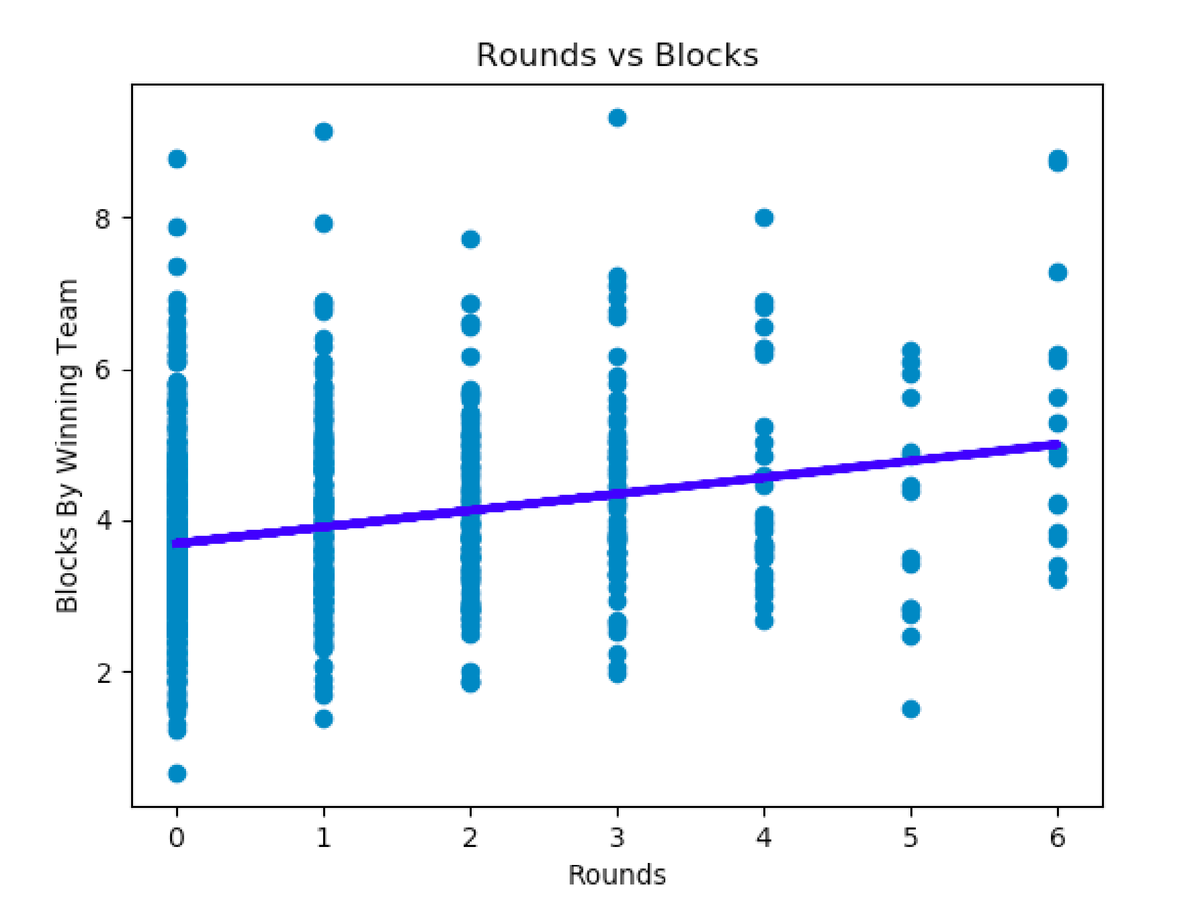
ROUNDS VS BLOCKS

Coefficients:

[[0.21824861]]

Mean squared error: 9.75

Variance score: -4.52



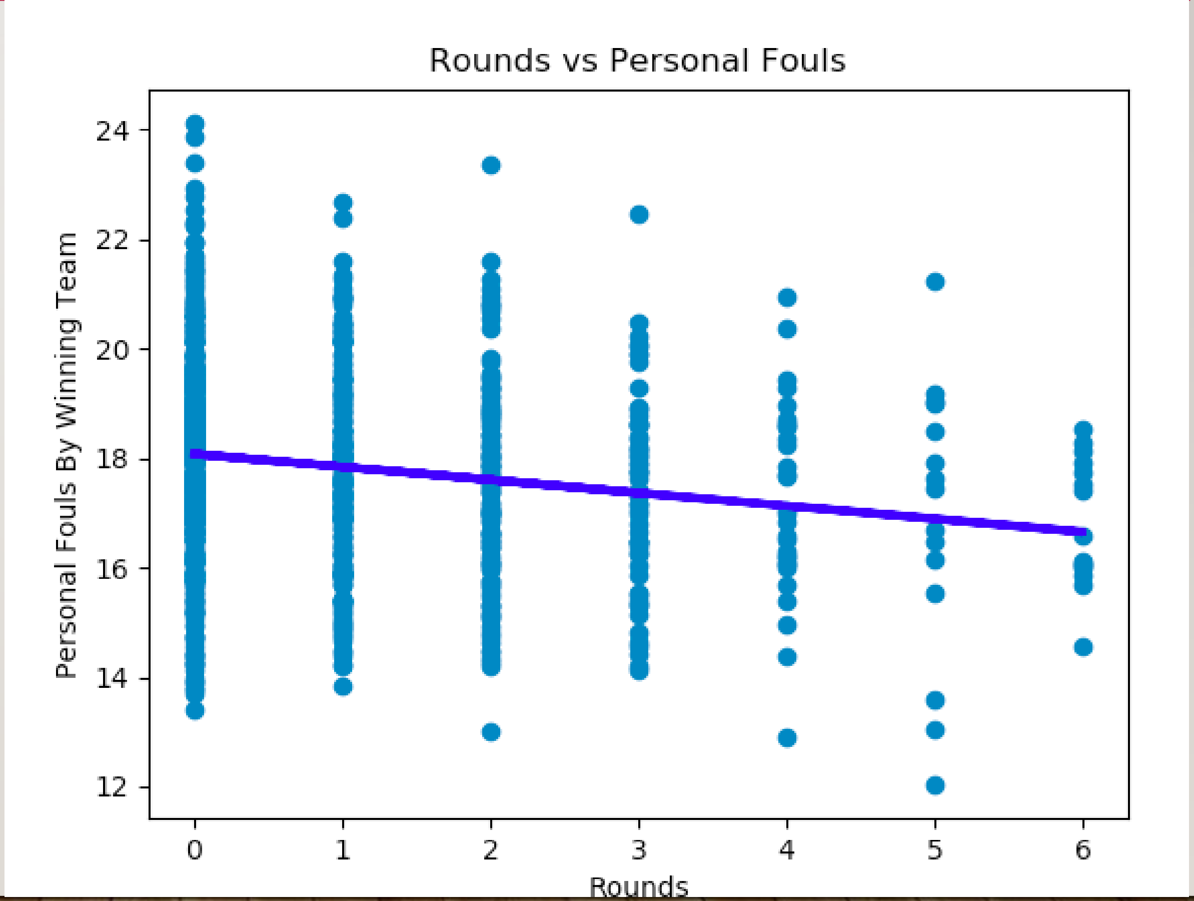
ROUNDS VS PERSONAL FOULS

Coefficients:

[[-0.23720717]]

Mean squared error: 288.20

Variance score: -162.24



Binary Classification

- Label teams based on whether or not they reached the Final Four stage of the March Madness tournament
- Trained classifiers using regular season statistics for each team
  - Excluded March Madness statistics for each team as there is a known correlation between features like seeding and likelihood of the likelihood of reaching the Final Four
- Utilized Scikit-Learn implementation of classifier with default parameters
  - neural network (MLPClassifier)
  - decision tree (CART algorithm) (DecisionTreeClassifier)
  - K Nearest Neighbor (KNeighborsClassifier)

Binary Classification Results

- Classifier Accuracy based off of testing data (2017 season stats)
  - Neural Network: 0.94 (64/68)
  - Decision Tree: 0.93 (63/68)
  - K Nearest Neighbor: 0.96 (65/68)

Future Direction

- Investigate March Madness games where a lower seeded teams defeats an higher seeded team, leading to an upset
  - Use supervised machine learning techniques like support vector machines (SVM) to see if upset teams have specific advantages in certain stats/features over the teams they upset
  - Apply kmeans clustering to see if upset teams cluster together based on a certain subset of features compared to other teams