# Predicting Runs

Derek Owens-Oas, Fan Bu, Federico Ferrari, Megan Robertson

September 24, 2017

## 1 Introduction

We chose to investigate the fourth prompt during the NBA Hackathon - create a model to predict exciting runs during games. One way to define a run is when a team scores on multiple consecutive possessions without the opponent scoring any points in between. This definition captures many runs, but it does not capture the scenario in which a team makes three-pointers on three consecutive possessions, and the other team scores a quiet lay-up in between. We propose a novel and flexible definition of runs which captures scenarios like this.

With runs well defined, we set out with a goal to learn which covariates, or features of the current gameplay, correlate strongly with runs. A few specific hypotheses motivate our analysis. We think there are scenarios in which a streak is more likely. A motivating halftime speech, a large skill differential between the offense?s and defense?s lineups, or an opponent with a propensity to turnover the ball may increase the likelihood of a streak. On the other hand, some features may decrease the likelihood of a run. Playing the second game of a back to back is one example.

## 2 Defining an "exciting" run

In order to make the project approachable during a 24-hour time period, a simplified definition of run was used to classify every possession in the 2016-2017 season. The algorithm below was used to create the labels for the possession log data.

**Algorithm 1** Streak Yes or No?

Set $t = 2$ (minutes)
Call $t_1, t_2, ....t_T$ the end times of each possesion
Find $K$ such that $\lfloor (t_T - t_K) \rfloor = t$

**for** $k$ in $1 : T$ **do**
  Compute R=range($[t_k, t_k + 2]$)
  **if** $R > 6$ points **then**
    Set m=min($[t_{k-1}, t_k]$)
    Set M=max($[t_{k-1}, t_k]$)
    Label points in $[m, M]$ as part of the *exciting* streak
  **end if**
**end for**

Repeat the procedure with starting points $t_0' = t_0 + \frac{1}{3}t$ and $t_0'' = t_0 + \frac{2}{3}t$

Figure 1: Algorithm for Defining Run

The algorithm first finds the interval of length $t$ having as left endpoint the $t_0$, that is the time when the first possession ends. Then, it computes the maximum and the minimum that the function $H()$ attains in the interval, i.e. the highest point difference that we observe in $[t_0, t_0 + 2]$. We have a run if this point difference exceeds a definite threshold, and if this is the case then we label the points between the Maximum and the minimum as part of the *exiting* streak. Then, we translate the interval and we check $[t_1, t_1 + 2]$,$[t_2, t_2 + 2]$, etc. We chose $t$ to be 2 minutes because we wanted to consider exciting, fast-happening, runs. However, it is clearly possible to $t$ according to different goals. Finally, this choice of $t$ generates around $2 - 3$ runs per game, so that around 20-30% of points are labelled as part of a run, making them even harder to detect.

# 3 Feature Generation

Multiple variables were added to the possession log data that we thought might be indicative of whether a run was occurring at a certain possession in the game. Two components that define a run are the pace of the game as well as the points being scored. Typically runs are the result of many rapid plays. Therefore, variables were added for the number of shots taken during a possession as well as the number of rebounds. More shots and more rebounds would be indicative of a team getting many offensive boards and having to work for their basket. The type of shot is also very important to consider when exploring runs. Discussions with our coach and other league mentors led us to add an indicator variable defining whether a shot was a "good" shot. This was arbitrarily defined as any shot that was within six feet of the hoop. This is roughly the areas in and around the key closest to the hoop.

In addition to capturing the variability of runs through the above variables, we also wanted to capture the skill level of the players on the court at the time of the run. Using the play by play data, it was possible to determine the players on the court at the time of each possession. From this, we were able to use external data [1] to determine the number of all stars on the court at the time. This is a very basic way to evaluate skill, but other methods could be used to define quality players. However, given the brief work period we only incorporated the all-star information for the time being.

# 4 Modeling Runs

We approached predicting exciting runs as a classification problem, each possession in a game can either be classified as being part of a run or not.

From a certain team's perspective, each possession has one of 3 potential labels: -1 (an existing run against them), 0 (no existing run), 1 (an existing run favoring them). Therefore we can treat the prediction task as a 3-class classification problem. We propose a multinomial classification model with the label of the next possession as the response and the following variables of the current possession as predictors:

- Period: which period the possession is in (included in the model as a factor)
- Team of possession: which team possesses the ball during that possession
- Net points: the cumulative net point of one of the teams (eg. if that team is losing, this number is negative)
- Rolling net points in the last 3 possessions: the rolling sum of net points won during the past 3 possessions
- Team streak indicator: whether the team with possession is on a run (-1: against; 0: no run; 1: favor)
- Touches
- Dribbles
- Passes

We randomly sample 500 games from the 2016-17 season and partition them into 9/1 portions as trainning set and testing set. The model yields a 0.58 top-1 accuracy and 0.86 top-2 accuracy on the test set.

The figures below demonstrate the model prediction results on the game between Golden State Warriors and New Orleans Pelicans on November 7, 2016, when Stephen Curry set a new career record of 3 point shots in one game.

The left plot shows the net point curve of Warriors, with the lighter part as the "runs" detected by our algorithm explained in Section 2. The vertical lines mark the timeouts, with the red ones as timeouts called by the Warriors coach. The dots are the predicted high probability "run" warning points generated by our model, where the orange dots suggest a run favoring the Warriors and the green dots otherwise. It's safe to say that the model is capable of picking up "runs" as defined, since the dots almost perfectly coincide with the highlighted lines.

The right one plots the probability curve assigned to "UP" runs and "DOWN" runs with respect to the Warriors. When the model decides that the Pelicans are going to have a run, the second curve has a peak; otherwise the third curve peaks. We can see very clearly that the peaks of the bottom two curves correspond to the highlighted "runs" in the first plot, where the net points of the Warriors are plotted over the game time.

[1] "NBA All Star", `http://www.nba-allstar.com/players/lists/players-by-draft-pick.html,9/23/17`

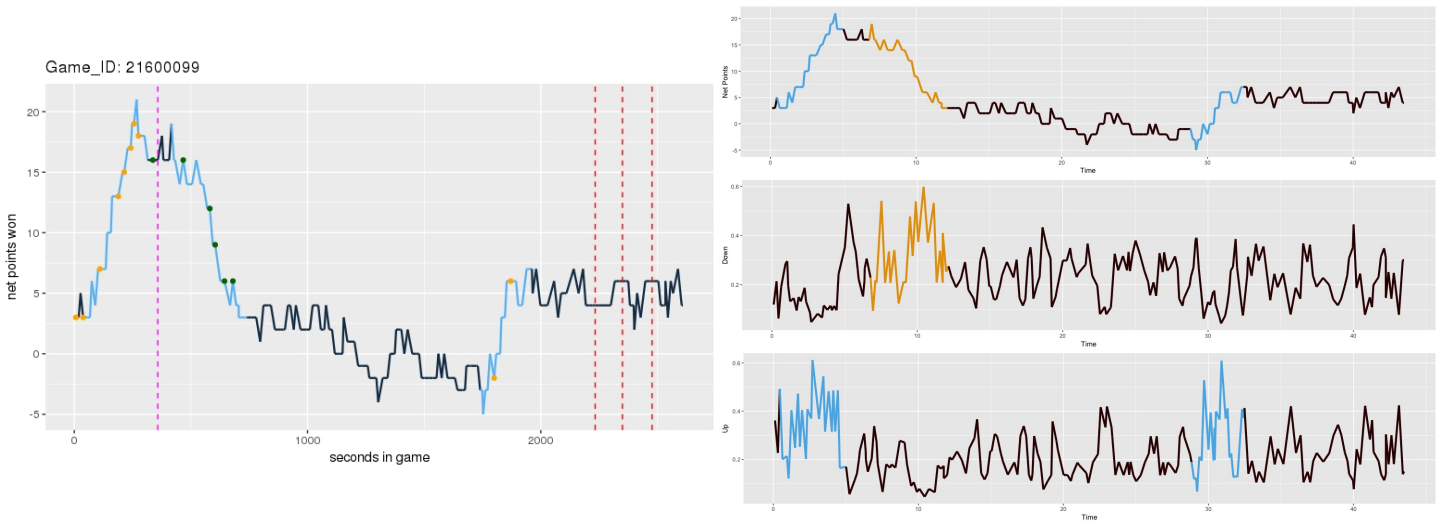


Figure 2: Left - Net points curve with timeouts and predictions, Right - UP run and DOWN run probabilities

## 5 Interpreting Coefficients

We fit two models, one to predict the probability of going on a run and one for the probability of the opponent going on a run. The plot on the left shows changes in the odds (technically log odds) of going on a run. For example the points coefficient has a positive value. This means for every point scored, we expect an increase in the log odds of approximately .7. This agrees with our intuition that higher scoring possessions are more likely to accumulate to a run. In the other direction, we observe a negative effect for a defensive run. There is also a slight significant positive effect for the presence of all stars on the court, confirming intuition that increase in the number of all stars on the court increases the probability of going on a run. On the see predicted changes in the odds of allowing a run. As expected, scoring more points on a possession decreases the odds of allowing a run.

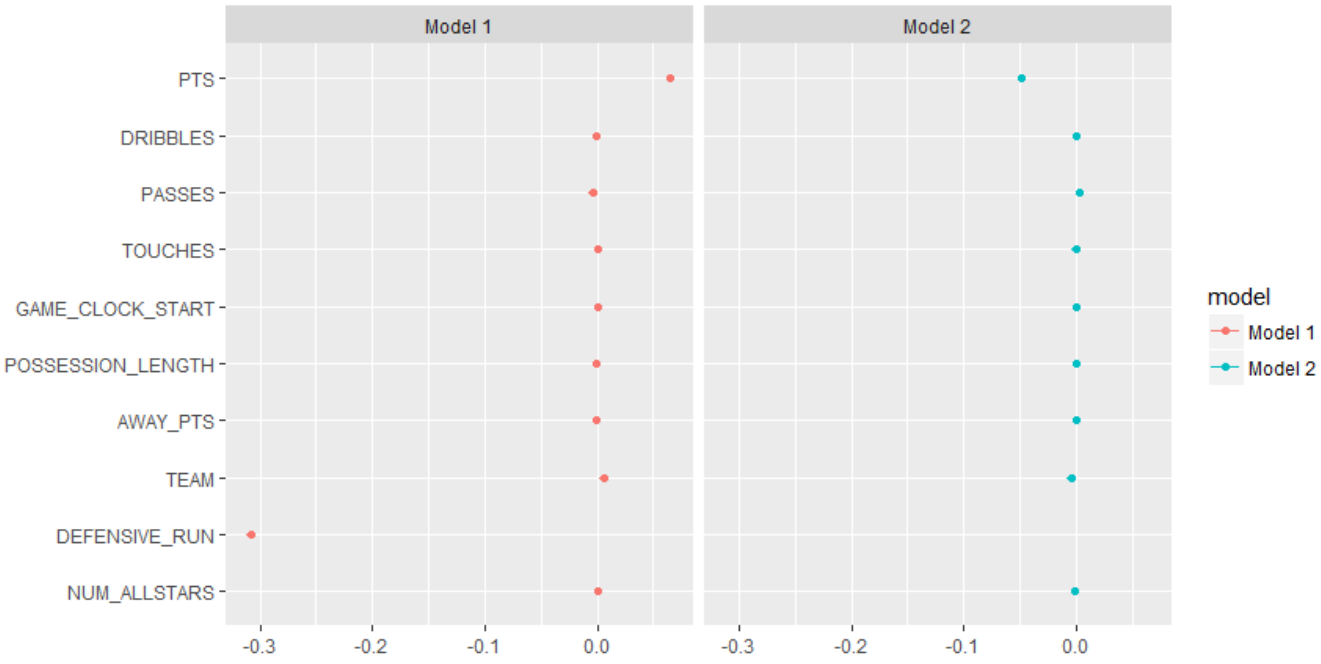


Figure 3: Modeling Results. Model 1 - Odds of going on a run, Model 2 - Odds of allowing a run

Coefficients:

| | Estimate | Std. Error | t value | Pr(>\|t\|) | |
|---|---|---|---|---|---|
| (Intercept) | 2.765e-01 | 3.217e-03 | 85.965 | < 2e-16 | *** |
| PTS | 6.502e-02 | 7.434e-04 | 87.473 | < 2e-16 | *** |
| DRIBBLES | -4.890e-04 | 2.272e-04 | -2.152 | 0.03139 | * |
| PASSES | -3.694e-03 | 1.818e-03 | -2.032 | 0.04212 | * |
| TOUCHES | 1.356e-03 | 1.665e-03 | 0.814 | 0.41557 | |
| GAME_CLOCK_START | -9.395e-07 | 4.076e-07 | -2.305 | 0.02116 | * |
| POSSESSION_LENGTH | -3.722e-05 | 1.928e-05 | -1.931 | 0.05353 | . |
| AWAY_PTS | -9.758e-04 | 7.349e-04 | -1.328 | 0.18427 | |
| TEAM | 6.292e-03 | 2.258e-03 | 2.787 | 0.00532 | ** |
| DEFENSIVE_RUN | -3.074e-01 | 2.143e-03 | -143.453 | < 2e-16 | *** |
| NUM_ALLSTARS | 9.120e-04 | 5.216e-04 | 1.749 | 0.08037 | . |

Coefficients:

| | Estimate | Std. Error | t value | Pr(>\|t\|) | |
|---|---|---|---|---|---|
| (Intercept) | 2.356e-01 | 3.028e-03 | 77.808 | < 2e-16 | *** |
| PTS | -4.894e-02 | 7.014e-04 | -69.764 | < 2e-16 | *** |
| DRIBBLES | 7.173e-04 | 2.166e-04 | 3.312 | 0.000925 | *** |
| PASSES | 3.633e-03 | 1.733e-03 | 2.097 | 0.036000 | * |
| TOUCHES | -2.810e-04 | 1.587e-03 | -0.177 | 0.859487 | |
| GAME_CLOCK_START | -2.865e-07 | 3.885e-07 | -0.737 | 0.460888 | |
| POSSESSION_LENGTH | 1.041e-05 | 1.838e-05 | 0.567 | 0.570998 | |
| AWAY_PTS | -1.451e-04 | 7.005e-04 | -0.207 | 0.835862 | |
| TEAM | -4.050e-03 | 2.152e-03 | -1.882 | 0.059821 | . |
| NUM_ALLSTARS | -5.136e-04 | 4.972e-04 | -1.033 | 0.301567 | |

Figure 4: Left - Model 1 Coefficients, Right - Model 2 Coefficients

# 6 Next Steps

This project was an attempt to answer a very complex question in a sort period of time. There are a few routes we would consider if we had more time to continue the project. To begin with, there is more data that will most likely capture the variability of whether a possession could be classified as a run. At the moment, the skill of the players is captured by the number of all stars on the court during a possession. However, analyzing different combinations of players, particularly when there are starting players vs. bench players, etc, would be interesting to examine in regards to runs.

Another next step would be to examine different types of classification models. There are many different types of machine learning algorithms that could be implemented in such a project. A more abstract model, such as a decision tree or a support vector machine might capture some of the complicated relationships that are present in the data.