Team 187 **Final Report**
Adam Warner, Calvin Leather, Robert Curran

Introduction - Motivation:

The motivation behind our project is to provide a way for users to interact with a model that predicts the likelihood of an average NBA player making a shot. There is plenty of literature on prediction and modeling of NBA games, and a number of static visualizations of the likelihood of a shot success or the effects of a decision to pass on the outcome of a possession, but these solutions all lack one thing: user interaction.

Problem Definition:

We want to allow a user to interact visually with a predictive model trained on features of more than 270,000 shots from an NBA season in a highly accessible way. Allowing a user to interact with the model in real time by moving relevant players around on an NBA court will facilitate intuition building. Compared to static visualizations, the user can move players around the court, update values of non-spatial model features, and receive a probabilistic output as a measure of shot quality. This functionality is of interest to many groups including basketball coaches trying to design and implement new plays, managers making player roster decisions, basketball trainers trying to determine which player skills are most important to improve, and statistically infatuated basketball fans that want an entertaining and rigorous predictive basketball experience. Accessibility is another issue with current approaches - we want to provide a solution that does not require a background in programming or statistics, the installation of additional software or exploration of scientific papers to build a probabilistic intuition of the game of basketball.

Survey:

Sports data analysis has proven a powerful approach to learn from past games and optimize strategies for future games. Many applications of sports analytics have been researched in recent years, especially for NBA games. A recent study was done using play-by-play NBA data to optimize shot selection strategy, including how often to make 3-point vs. 2-point shots (Fichman & O'Brien, 2019)(Chang S-C, 2018). Another used gameplay data to choose between taking or passing on a shot, given time remaining on the shot clock (Goldman & Rao, 2017). These are relevant studies because they highlight shot type and shot clock as potentially important features in predicting shot success. An additional study centered around basketball shot factors that chose to consider include a study that used shot tracking data from the 2013-2014 NBA season to propose and illustrate "Effective Shot Quality", or the rate at which an average NBA player would make a shot given distance from the basket and distance of defender (Chang et al., 2014). This study provides evidence that distances are relevant for predicting shot likelihood, allowing us to design a spatial visualization around those features.

The ability of an individual to make a successful shot is a key component to winning a game. "A Shot Recommender System for NBA Coaches" used a recommender system to predict shot success with shot log data and underscores some of the challenges of using supervised learning, the main approach in our research. We also considered another study that analyzes the optimality of the shot distribution and which players on a team are contributing most to the lost shots (Sandholtz et al., 2019). As our project approach seeks to analyze shot quality, shot location is likely to be a factor we need to take into consideration. Players must also determine which shot opportunities are likely to lead to success, and which should be passed as considered in "The Problem of Shot Selection in Basketball", which analyzed which shot opportunities were most likely to result in points scored and should be taken. Lastly, the fourth study to consider analyzed how team formations influenced getting shot takers open and its effect on shot quality (Lucey et al., 2014). Each of these papers brings to light factors we will consider when determining the features our model needs to consider. Our project seeks to improve upon these studies by creating a real-time, interactive visualization based on player identity and location in the field.

Spatial variables also play a key role when considering analytics basketball models. One such study used player data to build an economic model of how a player's actions will change the Expected

Value of Possession. The model assigns a probability and value to each decision (pass, shoot, drive) and the resulting EPV after the decision using Markov chains (Cervone et al., 2014). This article provides perspective on creating a framework to determine how changes in spatial attributes affect outcomes in an NBA game. Another study identified the "best" shooters by identifying the range of court area a player effectively shoots from (Goldsberry, 2012). This addresses "who is the best shooter?", rather than "how does the average NBA shooter perform?", making it less useful to the project. Two additional papers we reviewed include one that measures shot success using multinomial regression using spatial and shot type variables which provides a specific example of how to create a model to fit our data (Erčulj & Štrumbelj, 2015) and another that attempts to answer the "hot-hand" fallacy, if a player is more likely to make a shot given a previous miss or a previous make (Chang, 2018). These two studies can be integrated into our research because they examine how shot type and previous shot attempts affect the probability of a successful attempt. A fourth paper provides an interesting example of how environment and time affects the score outcome throughout the game. Fitting a power law model attempts to find the scoring rate for a game (Ribeiro, 2016). All three studies fail to recognize that there is more than a binomial outcome to a shot attempt.

Lastly, we assessed some additional papers that surveyed other disciplines for spatial analysis approaches. Chi and Zhu (2007) assessed modeling techniques including conventional ordinary least squares (OLS), OLS with adjacency-based autoregression, and spatial autoregressive moving average (SARMA), and found SARMA had the best AIC/BIC with the least spatial error autocorrelation. This study also provides factors to measure spatial model fit. A second study we reviewed applied several advanced methods, including wavelet regression, and Bayesian approaches, finding generalized linear model (GLM) convergence issues, but that the Bayesian autoregressive approach yielded the best data fit (C. M. Beale et al., 2010). This paper is relevant as it suggests Bayesian approaches. Lastly, we reviewed a methods paper for a Bayesian spatial modeling system (Lee, 2013). They identify mechanisms to speed up spatial markov chain monte carlo (MCMC) sampling, and benchmark results against OLS methods. This paper provides a specific method to use to analyze our data. Each of these papers have the shortcoming that they analyze data with pre-existing spatial discretizations (e.g. county boundaries), whereas basketball only has the overly-broad 1, 2 and 3 point zones, meaning we will have to identify a novel strategy to discretize the space.

Proposed Method:

Intuition: We believed our visualization would be better than the state of the art because we knew that static, non-interactive visualizations could impede learning. Using an interactive visualization, the end user can easily run experiments of different basketball situations, fostering an intuition of the impact of various features on a shot's outcome. We believed our model would provide value over modern shot prediction models for two reasons. First, our model would not only use cutting edge player location data but would also be joined with features of players such as height and wingspan, and categorizations of shot types into jump shot, lay-up and others. Secondly, we knew we had a number of modeling approaches in our tool bet that we could experiment with to provide a cutting-edge prediction.

Description of Approaches: The original data set contains 21 features of approximately 120,000 shots taken during the 2014-2015 NBA season [D1]. This data was formatted as a csv for download. There are two popular open source packages that pull in NBA data, ballR [P5] and nba_api [P6] for R and Python, respectively. ballR allowed for the pulling of the shot location for the 2014-2015 by player id, which is a unique key that is standardized across many NBA data sets. To join the seasons worth of x, y shot location data generated by the ballR package we had to create our own unique shot key id. To create a unique shot id both datasets were grouped by the player id and game id and then sorted by the game clock in descending order. We then created a running tally of the shot number allowing us to join the datasets on the combination of game id, player id and shot number. Player attributes such as player

height and weight were gathered using the nba_api package in Python, but not all player attribute data was successfully pulled from the API and needed to be manually added. To increase the size of our dataset, we added shots from the 2015-2016 season to make our final data set 270,000 rows and 49 columns.

To build a model to predict the likelihood of a shot going in we experimented with a few models including logistic regression, decision trees, a random forest, and boosting algorithms. The idea was to experiment with various models, sets of features and hyperparameters to try to arrive at a highly accurate model, as well as an accurate model that is also simple. Logistic regression was initially proposed due to the probabilistic output of a binary predictor. We also thought trees would capture the complex interactions of basketball, while also providing a probabilistic output.

To create an accessible visualization, we decided to build the visualization using JavaScript, HTML and CSS, along with the D3 and JQuery libraries. The foundation of the user interface is the lines of a basketball court expressed as GeoJSON sourced from a publicly available GeoJSON repository [P7]. This GeoJSON set contains the courts sidelines, baseline, the key area, half court line, the location of the basket over the court and various hash marks. On top of the map of the court are two moveable circles that represent an offensive player and a defensive player. There are lines between the two players and the offender player and the basket that are redrawn as the circles are moved. We theorized that the distances between players and the basket would be relevant in our model, so we decided that they would need to be visualized. We also wanted to create points of interaction with non-spatial features of the model, such as time remaining in the possession and shot type and proposed using HTML input fields to achieve this.

To allow a user to interact with our model with just a web browser, we experimented with a few solutions, but decided to use the Pyodide package [P8]. This package makes a Python interpreter available in the browser by compiling key scientific and statistical packages such as numpy, scipy, statsmodels and sklearn [P1, P2, P3, P4] into WebAssembly. We used the Pickle package to serialize trained models and load them into the browser when the page is opened. As a user moves either player on the page, or when changing one of the non-spatial feature values, the model's parameters are passed to our model's predict function, returning the measure of shot quality predicted by the model in real time.

Experiments/Evaluation:

Testbed: We used packages from both R and Python for our analytics and modeling experimentation including scikit-learn, statsmodels, xgboost and glm. For data manipulation and cleaning, we used Pandas, NumPy dplyr, and data.table. The subset of features includes shot distance from basket, closest defender distance, shot clock or game clock remaining, shot count of shot by player and game, amount of time a player held the ball before shooting, if the shot was a two or three point attempt, the type of shot (e.g. jump shot, lay-up, dunk), lateral orientation of shot location (i.e. center, right and left side of court). We leveraged glm, sklearn and statsmodels to build models and measured model accuracy using 10-fold cross validation as used in the state-of-the-art models discussed in "Quantifying Shot Quality in the NBA". Visualizations were tested with an http server, web browser and JS/HTML/CSS stack.
Experimental Questions: How can we facilitate a user's interaction with a trained model in R or Python? Which models should we experiment with and which models will provide the most accurate models? Which set of features and interactions of features will result in a simple but accurate predictive model? How can we effectively visualize the findings of our spatial based model development? How can we improve on the 63.3% state of the art model accuracy report in Quantifying Shot Quality in the NBA?
Details of Experiments: To find a solution to allowing a user to interact with a trained model from a web browser, we experimented with two packages - PlumbR and Pyodide. PlumbR allows you to expose methods in R as http endpoints. For example, we could expose a method that takes the model

parameter and calls predict on a trained model, returning the model's prediction in an http response. Using the PlumbR solution would require the end user to run to install R on their machine but would allow us to access the functionality of an R development environment. Pyodide is a package that makes a Python interpreter available in the web browser using WebAssembly and exposes important scientific and statistical packages available, such as numpy, scipy, sklearn and statsmodels. We experimented by loading relevant statistical packages into our web browser embedded Python environment at page start up, loading a Pickled model into the environment as a global variable, and then calling predict on that model with parameters arranged in JavaScript as the user interacted with the web page. This proved to be accessible, practical, and efficient; it took less than a minute for the page to load the Python environment, packages and model and provided low-latency model interaction.

In developing an accurate predictive model of shot outcomes, we experimented with a few types of models: logistic regression, decision tree, random forest and xgboost. First, we began experimenting with logistic regression. Using the features outlined in the previous section, we used sklearn's StandardScalar to scale the continuous features and Pandas' get_dummies method to create dummy variables from the categorical features such as type of shot and lateral orientation of the offensive player at the time of the shot. Using sklearn's LogisticRegressionCV, we were able to train, and cross validate a model in the same method call. We used 10-fold cross validation in order to compare model performance against our target of 63.3% accuracy outlined in "Quantifying Shot Quality in the NBA", which also used 10 folds. We were able to decrease the execution time of the cross validation by using the n_jobs parameter of -1, which uses all logical processors to run validation in parallel. This method call returned accuracies scores across the 10 folds for 10 different values of C, the inverse of regularization strength. We then extracted the C value with the highest mean accuracy of the 10 folds, and reported an accuracy of 65.9% and C value of 0.006, 2.6% more accurate than the state of the art. The low C value implies a high level of regularization and simpler model compared to a high C value. This makes sense for the out of sample accuracy of the model, a regularized model reduces overfitting and results in a model that generalizes well. We then extracted the model coefficients and identified features with the largest beta values, which resulted in a difficult to interpret model given the number of dummy variables and generalized results for continuous variables.

In addition to running a simple logistic regression, we used more computational complex algorithms such as the xgboost algorithm. Xgboost [A4] is a scalable tree boosting algorithm that was created in 2016. When using this algorithm we got an accuracy rate of 63% using a 5-fold CV. Boosting algorithms learn in an intuitive way, in each iteration that attempts to minimize the error measure of choice the algorithms weights the previous rounds residuals more heavily in order to create an ensemble of learners which generates a final model that has high predictive accuracy. We still needed to answer how useful the data is in predicting the success of the shot. In another attempt to prove that our model's performance was better than just chance we created a null "model". The null model is created by randomly shuffling the labels such that the data should just provide noise. The null model had an accuracy rate of 54% showing that our xgboost model was substantially outperforming the null model.

Next, we experimented with a decision tree. Our intuition was that a decision tree would provide an interpretable and highly predictive model because it would capture the complex interactions of the game of basketball. We used sklearn's GridSearchCV method to tune the DecisionTreeClassifier's hyperparameter of max_depth. We tested values from a max_depth of 2 to a max_depth of 24. Our experiments yielded the result that a max_depth of 8 provided the most accurate model using 10 folds to cross-validate, with an accuracy of 65.3%, exceeding the 63.3% target by 2%. This was an accurate model, but when we investigated the structure of the tree using sklearn's plot_tree, the result was much more complex than anticipated. Eight nodes of depth resulted in 256 leaf nodes, which would be too

complex to communicate to an end user, and it's accuracy would likely decrease when applied to future NBA seasons, as the style of play changes, compared to more simple models. We experimented with a tree of depth 3, which resulted in a 10-fold accuracy of 63.7%, which is still above our target of 63.3%. Unfortunately, we didn't have the time or data to compare how well models performed outside of the NBA season they were trained in.

Finally, we examined the use of random forests. The hyperparameters we tuned when building these models were the number of estimators, or number of trees in the forest, and the maximum depth of the trees. Initially, we experimented with a subset of the features (excluded shot type and shot location categorical features) we used for other models, and found that the most accurate random forest contained 250 trees, had a maximum depth of 8 nodes and produced an accuracy of 62%. Then we added the excluded categorical variables and re-ran the hyperparameter tuning using GridSearchCV and found that the most accurate model had 400 trees in the forest, and a maximum depth of 17 for each tree. This model produced an accuracy of 65.8% but was far more complex than the previous model.

To evaluate end user experience, we planned to use a convenience sample of basketball fans (n=5) from the project team's friends and colleagues and provided a link to the visualization system on github. We would have provided a simple questionnaire - Were you able to load and interact with the visualizations? Were there any technical issues? Did you learn anything new from the visualization? How do you think fantasy sports fans or team managers could use this information to alter decisions? What questions would you be interested in answering about shot success that you were unable to answer? We were not able to conduct this experiment due to time constraints, but it would be a next step for evaluating the user interface.

Conclusion and Discussion:

We were able to train multiple types of models that achieved accuracy higher than the state-of-the-art model discussed in "Quantifying Shot Quality in the NBA", which achieved an accuracy of 63.3%. We achieved similar 10-fold cross-validated accuracy across logistic regression, classification trees and random forests, with our most accurate model being a logistic regression with 65.9% accuracy. Our simplest model, a three-node deep classification tree, achieved an accuracy of 63.7%. Our experiments showed that feature selection contributed more to model accuracy than the type of model or choice of hyperparameter values, and features of shot type of jump shot, distance of the shooter from the basket, distance of the closest defender, number of dribbles taken before the shot, and amount of time remaining in the possession were relevant across models. The next model we would experiment with is a regression tree; this model would capture the complex interaction we believe exist between the features of a basketball shot and would allow for feature coefficients to be estimated to allow a user to examine how marginal changes in continuous variables affect the quality of a shot.

We were also able to achieve our goal of providing an interactive and accessible visualization with Javascript, HTML, CSS and Pyodide, allowing for a user to interact with a trained and persisted model through a web browser. This approach requires no additional software to be installed by an end user and offers a better experience than viewing static visualizations or reading through several published papers. Unfortunately, we did not have time to test the effectiveness of our visualization through end user surveys, which would be the next step for us to improve our user interface.

Overall, we believe we improved on the state of the art from both a modeling and visualization perspective. We believe our visualization can inspire others to incorporate interactive visualizations into their research report, and we have provided a template that can be open sourced to contribute to the quality of basketball visualizations.

Adam Warner, Calvin Leather, Robert Curran

Distribution of Team Effort:

All team members have contributed a similar amount of effort.

Bibliography:

[E1] Fichman, M., & O'Brien, J. R. (2019). Optimal shot selection strategies for the NBA. *Journal of Quantitative Analysis in Sports*, *15*(3), 203–211. doi: 10.1515/jqas-2017-0113
[E2] Goldman, M., & Rao, J. M. (2017). Optimal stopping in the NBA: Sequential search and the shot clock. *Journal of Economic Behavior & Organization*, *136*, 107–124. doi: 10.1016/j.jebo.2017.02.012
[E3] Wright, R., Silva, J., & Kaynar-Kabul, I. (2016). Shot recommender system for nba coaches. *KDD Workshop on Large Scale Sports Analytics*. doi: 10.475/123 4
[C1] G. Chi and J. Zhu, "Spatial Regression Models for Demographic Analysis," Population Research and Policy Review, vol. 27, no. 1, pp. 17–42, 2007.
[C2] C. M. Beale, J. J. Lennon, J. M. Yearsley, M. J. Brewer, and D. A. Elston, "Regression analysis of spatial data," Ecology Letters, vol. 13, no. 2, pp. 246–264, 2010.
[C3] Lee, D. (2013) CARBayes: an R package for Bayesian spatial modeling with conditional autoregressive priors. Journal of Statistical Software, 55(13), pp. 1-24.
[R1] Chang, Y.-H., Maheswaran, R., Kwok, S., Levy, T., Wexler, A., Squire, K., & Su, J. (2014). Quantifying Shot Quality in the NBA. *MIT Sloan Sports Analytics Conference 2014.*
[R2] Goldsberry, K. (2012). CourtVision : New Visual and Spatial Analytics for the NBA. *MIT Sloan Sports Analytics Conference 2012.*
[R3] Cervone D, D'Amour A, Bornn L, Goldsberry K. (2014). POINTWISE: Predicting Points and Valuing Decisions in Real Time with NBA Optical Tracking Data. *MIT Sloan Sports Analytics Conference 2014.*
[A1] Erčulj, F., & Štrumbelj, E. (2015). Basketball Shot Types and Shot Success in Different Levels of Competitive Basketball. *PLoS One*.
[A2] Chang S-C (2018). Capability and opportunity in hot shooting performance: Evidence from top-scoring NBA leaders. *PLoS ONE 13(2): e0179154*. https://doi.org/10.1371/journal.pone.0179154
[A3] Ribeiro HV, Mukherjee S, Zeng XHT (2016). The Advantage of Playing Home in NBA: Microscopic, Team-Specific and Evolving Features. *PLoS ONE 11(3): e0152440*.
https://doi.org/10.1371/journal.pone.0152440
[A4] KDD '16: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining August 2016 Pages 785–794 https://doi.org/10.1145/2939672.2939785
[S1] Sandholtz, Nathan, et al.(2019). "Chuckers: Measuring Lineup Shot Distribution Optimality Using Spatial Allocative Efficiency Models." *MIT Sloan Sports Analytics Conference*,
http://www.lukebornn.com/papers/sandholtz_sloan_2019.pdf.
[S2] Skinner, Brian. (2012). "The Problem of Shot Selection in Basketball." *PLoS ONE*, vol. 7, no. 1, 2012, doi:10.1371/journal.pone.0030776.
[S3] Lucey, Patrick, et al. (2014). "'How to Get an Open Shot': Analyzing Team Movement in Basketball Using Tracking Data." MIT Sloan Sport Analytics Conference,
http://www.iainm.com/publications/Lucey2014-How/paper.pdf.
[P1]Pandas: McKinney, W., & others. (2010). Data structures for statistical computing in python. In *Proceedings of the 9th Python in Science Conference* (Vol. 445, pp. 51–56).
[P2]Numpy: Oliphant, T. E. (2006). *A guide to NumPy* (Vol. 1). Trelgol Publishing USA.
[P3] Sklearn: Scikit-learn: Machine Learning in Python, Pedregosa *et al.*, JMLR 12, pp. 2825-2830, 2011.
[P4] Statsmodel: Seabold, Skipper, and Josef Perktold. "Statsmodels: Econometric and statistical modeling with python." Proceedings of the 9th Python in Science Conference.
[P5] Ballr: Elmore, R. (2020). *Ballr* (Vol. 0.2.6). Retrieved from https://cran.r-project.org/web/packages/ballr/ballr.pdf

Adam Warner, Calvin Leather, Robert Curran

[P6] Nba_api: Patel, S. (n.d.). *Nba_api*. Retrieved from https://pypi.org/project/nba-api/
[P7] Geojson: Carto. Basketball Court. Retrieved From https://team.carto.com/u/aromeu/tables/basketball_court/public
[P8] Pyodide: iodide. Pyodide. Pyodide, pyodide.readthedocs.io/en/latest/.
[D1] DanB. NBA shot logs. Version 1. Retrieved 2020-04-15 from https://www.kaggle.com/dansbecker/nba-shot-logs

Images of Visualization: