**Reading the Court: A Comparative Study of Machine Learning Models for Basketball Shot Prediction**

The LogReg Lakers

Finn Beckmann, Vignesh Kumar

Linköping University

# Abstract

TBD

# Introduction

# The National Basketball Association (NBA) generates detailed data for every field goal attempt: location on the court, time in the game, score situation, and information about the players involved. This makes it an ideal setting for studying which factors drive shot success and how accurately we can predict whether a given shot will go in.

# In this project, we build a large-scale shot-level dataset and use it to predict the probability that a field goal attempt is made. Our main goals are (1) to quantify the predictive value of contextual, player, and schedule features, and (2) to compare simple, interpretable models with more flexible non-linear approaches.

# The work combines two environments. All data preparation, feature engineering, and missing-data handling were carried out in SAS® Viya® Workbench, which is well suited for working with large tables and building reproducible pipelines. The predictive models were then implemented in Python using scikit-learn, following the same feature definitions that were developed in SAS Viya Workbench. This setup reflects a practical constraint: SAS Viya Workbench handled the heavy lifting on the data side, while the external environment provided more flexibility for large-scale model training.

# DATA

Our starting point is a shot-level dataset covering multiple NBA regular seasons, where each row in the final modeling table corresponds to a single field goal attempt. For each shot, we record detailed game context (quarter, remaining time in seconds, score margin, and a home/away indicator, although the latter is not used directly in the final models), shot geometry (court coordinates in centimeters, derived shot distance from the basket, and simple spatial indicators such as side of the court), player profile information (anthropometric measurements including height, weight, wingspan, standing reach, hand length, and hand width, as well as physical test metrics such as lane agility time, three-quarter-court sprint time, and standing and maximum vertical leap, plus the nominal playing position PG, SG, SF, PF, or C), and schedule and travel variables (travel distance to the current game, number of rest days since the previous game, time-zone shifts between games, and indicators for dense schedule situations such as back-to-back games).

Using Jupyter Notebooks in SAS® Viya® Workbench, we derived two main versions of this master dataset:

* a complete-case version, master\_data\_dropped, in which all shots with missing values in any modeling predictor were removed, yielding approximately 2.8 million shot attempts and including one-hot encoded position indicators (POS\_PG, POS\_SG, POS\_SF, POS\_PF, POS\_C)
* an imputed version, master\_data\_imputed, in which missing numeric predictors were filled using a multivariate iterative imputation procedure and explicit position dummies were omitted so that player roles are instead captured indirectly through continuous profile variables, resulting in roughly 4.5 million shots.

These two datasets allow us to compare modeling results under a conservative complete-case strategy and an imputation-based approach that preserves nearly all observations.

# Data Cleaning and Validation

# All data cleaning and validation were performed in SAS® Viya® Workbench before exporting the modeling tables. We first addressed missing and invalid values. Records with missing or undefined values for the target variable MADE\_SHOT were removed, and shots with clearly invalid or out-of-bounds court coordinates were filtered out. Observations with missing key timing variables, such as the remaining time in the quarter, were also excluded from both modeling datasets. For the complete-case dataset, master\_data\_dropped, any observation with a missing predictor used in modeling was dropped, resulting in a clean dataset of about 2.8 million shots. For the imputed dataset, master\_data\_imputed, missing numeric predictors were filled using a multivariate iterative imputation procedure based on the joint distribution of the observed covariates. Categorical position dummies were not imputed and are omitted in this version.

# In addition to handling missingness, we constructed several derived features and performed a series of quality checks. Shot distance was computed from the court coordinates and the basket location and then checked against NBA court dimensions to ensure plausibility. Remaining time in the quarter was stored as a continuous variable (in seconds), while the quarter itself was kept as a categorical variable. Score margin at the time of the shot was derived from the scoring history for each game, and travel distance, rest days, and time-zone shifts were calculated from game dates and locations.

# To further ensure data integrity, we checked for duplicated records using combinations of game ID, player ID, and timestamps and removed any confirmed duplicates. Basic summary statistics and simple visual inspections, such as histograms and boxplots, were used to identify implausible values in both raw and derived features; when necessary, these values were filtered or capped. Taken together, these steps produced two clean and internally consistent datasets that share the same feature definitions but differ in their approach to handling missing data.

# PROBLEM

The predictive task is to estimate whether a given field goal attempt will be successful. We define a binary response variable MADE\_SHOT that takes the value 1 if the shot is made and 0 otherwise. Each observation corresponds to one shot attempt and includes its associated context, player, and schedule information.

Let X denote the feature vector for a shot (including distance, coordinates, quarter, remaining time, score margin, anthropometric and physical test measures, and schedule metrics), and let Y ∈ {0, 1} denote MADE\_SHOT. Our goal is to learn a function f(X) that approximates P(Y = 1 | X). This is a standard supervised binary classification problem.

We focus on two broad questions:

* How well can we predict shot success using only contextual, player, and schedule features?
* Which predictors consistently emerge as important across different model classes?

Because both classes (made and missed shots) are of interest, we emphasize the F1 score as the main performance metric, alongside overall accuracy and confusion matrices.

# Analysis

All data preparation and feature engineering took place in SAS® Viya® Workbench. For model fitting, we exported the prepared tables and built the predictive models in Python using scikit-learn. The preprocessing in the Python pipelines mirrors the feature definitions and transformations developed in SAS.

## Feature set and preprocessing

To avoid information leakage and prevent the models from relying on identifiers without real predictive content, we removed all purely identifying variables (GAME\_ID, PLAYER\_ID, PLAYER\_NAME, TEAM\_ID, TEAM\_NAME). We also excluded latitude and longitude coordinates related to travel (LAT, LON, D\_LAT, D\_LON), the estimated flight time (FLIGHT\_TIME\_MIN), and explicit home and away labels (HOME\_TEAM, AWAY\_TEAM). Finally, we dropped the three-point indicator (IS\_3PT) so that the models have to infer the effect of three-point attempts from shot location and distance instead of relying on a dedicated flag.

The remaining predictors include shot geometry, quarter and remaining time, score margin, anthropometric and physical test measurements, schedule metrics, and (in the complete-case dataset) position dummies. The quarter of the game is encoded using one-hot indicators. All continuous variables are standardized using z-score scaling, implemented with a ColumnTransformer that is combined with each classifier in a unified pipeline. This ensures that cross-validation and final model fitting always apply exactly the same preprocessing steps.

## Missing-data strategies in modeling

We fit each model class (regularized logistic regression, decision trees, and, on subsamples, SVMs) on both master\_data\_dropped and master\_data\_imputed. On master\_data\_dropped, we gain explicit position information but lose observations with missing values. On master\_data\_imputed, we retain nearly all shots but rely on imputation and continuous player features to encode roles. Comparing results across these two versions helps us gauge the impact of the missing-data strategy on both performance and interpretation.

## Regularized logistic regression

As a linear baseline, we used logistic regression with regularization, implemented via scikit-learn’s SGDClassifier with the loss set to "log\_loss". This corresponds to logistic regression fit by stochastic gradient descent, which scales well to our multi-million-record datasets.

We considered two regularization schemes:

* L1 (LASSO): encourages sparsity and can shrink some coefficients exactly to zero.
* L2 (Ridge): discourages large coefficients but typically retains all features.

We tuned the regularization parameter α over a logarithmic grid {10⁻², 10⁻³, 10⁻⁴, 10⁻⁵} using two-fold cross-validation and the F1 score as the objective. For each candidate α, we built a pipeline containing the preprocessing and the classifier, selected the best configuration, refit it on the full dataset, and obtained cross-validated predictions using cross\_val\_predict. From these predictions, we computed classification reports and confusion matrices.

Across all four combinations (LASSO vs. Ridge on dropped vs. imputed data), the results are quite stable. On both datasets, overall accuracy is around 0.60, and the macro-averaged F1 score is about 0.59. The F1 score for made shots is around 0.54–0.55, and the models are somewhat better at identifying missed shots than made shots. This asymmetry reflects the intrinsic randomness of shot outcomes: even with detailed context, a large share of shot-to-shot variation remains unpredictable.

The choice between L1 and L2 regularization has only a modest effect on performance. Ridge models are slightly better in terms of weighted F1, while LASSO yields sparser, more interpretable coefficient vectors. In all variants, shot distance stands out as the dominant predictor, with a large negative coefficient: longer shots strongly reduce the log-odds of success. Game-context variables, such as quarter and remaining time, also have consistent effects, with earlier-clock and earlier-quarter shots being slightly more efficient. Player size and position contribute smaller but intuitive adjustments (for example, centers and taller players have higher success probabilities near the basket). A positive coefficient for the season indicator points toward a gradual improvement in league-wide shooting efficiency over time.

## Exploratory PCA

To better understand the structure of the feature space, we performed an exploratory principal component analysis (PCA) on the standardized predictors, separately for master\_data\_dropped and master\_data\_imputed. We removed identifiers and non-predictors, then applied PCA with n\_components set to 0.9, which instructs the algorithm to retain the smallest number of components that explain at least 90% of the total variance.

In both datasets, a relatively small number of components captures most of the variance, which is consistent with groups of correlated variables (for example, anthropometric measurements that jointly describe player size, or schedule variables that capture related aspects of travel load). However, because principal components are harder to interpret than the original variables, we did not use PCA for dimensionality reduction in the main models. Instead, PCA serves as a diagnostic that supports the use of regularization: penalized models can control effective complexity while keeping a directly interpretable feature representation.

## Decision tree classifiers

To capture non-linear effects and interactions in a more interpretable way, we trained classification trees on the full master\_data\_dropped and master\_data\_imputed datasets using scikit-learn’s DecisionTreeClassifier. The target was MADE\_SHOT, and the predictor set matched the one used for the logistic models.

We tuned the main complexity parameters (max\_depth, min\_samples\_split, min\_samples\_leaf, and class\_weight) using cross-validation with F1 score as the objective. The selected trees achieve accuracy and macro F1 scores that are very similar to those of the logistic regression models. In other words, moving from a linear to a tree-based model does not produce a consistent performance gain given the current feature set.

The trees are useful for their interpretability. Inspecting the learned structures, we see that the top splits almost always involve shot distance and related spatial features. Lower-level splits bring in game-time variables such as remaining seconds and quarter, and occasionally player profile metrics. This confirms the narrative suggested by the logistic models: distance dominates, context provides meaningful corrections, and the additional player and schedule information fine-tunes predictions rather than fundamentally changing them.

## Support vector machines on subsamples

Support vector machines (SVMs) with non-linear kernels can represent more complex decision boundaries but are computationally demanding on datasets of this size. To still assess their behavior, we trained SVMs with RBF and polynomial kernels on 1% random subsamples of both master\_data\_dropped and master\_data\_imputed.

We used the same preprocessing as for the other models and tuned C ∈ {0.1, 1, 10} and gamma ∈ {"scale", "auto"} via cross-validated grid search, again with F1 as the objective. On these subsamples, the best SVM models achieve accuracy and macro F1 scores that are very close to those of the logistic and tree-based models. In a few cases, polynomial SVMs slightly improve recall for missed shots at the cost of worse performance on made shots, but overall gains are small and not robust across datasets.

Given their high computational cost and limited improvements, we treat SVMs as a robustness check rather than a primary modeling tool. The fact that they do not clearly outperform simpler approaches suggests that most of the exploitable structure in this feature set is already captured by regularized logistic regression and decision trees.

## Visualization (Appendix)

Several visualizations were important during the analysis and would be included in an appendix in a full paper:

* Shot charts showing made and missed shots overlaid on the court, by distance band or player group.
* Histograms and boxplots for key features such as distance, remaining time, and score margin.
* Heatmaps of confusion matrices for the main models, highlighting the balance between correctly classified makes and misses.
* Simple tree diagrams or partial views of the learned decision trees, illustrating how splits on distance and time create regions of higher or lower success probability.
* Scree plots of explained variance in the PCA.

These plots help to communicate the data structure and model behavior in a way that complements the numerical results.

# Suggestions for Future Studies

This work can be extended in several directions. A natural next step is to incorporate richer features, especially player-tracking information such as defender distance, player speeds, and off-ball movement, in order to capture shot quality more directly. It would also be useful to distinguish between different shot types (for example, catch-and-shoot versus pull-up attempts) and offensive actions leading to the shot, rather than treating all field goal attempts as a single category.

On the modeling side, future studies could explore additional model classes, in particular ensemble methods such as random forests and gradient boosted trees implemented within SAS® Viya®, which may provide modest performance gains while remaining relatively interpretable. Hierarchical or mixed-effects models that explicitly account for player-level and team-level differences are another promising direction.

Finally, it would be valuable to place more emphasis on calibration and downstream use. Studying probability calibration would help ensure that predicted probabilities can be used directly for decision support and simulation, and evaluating models in terms of their impact on downstream tasks—such as shot selection optimization or lineup evaluation—would move the analysis closer to practical decision-making in a basketball context.

# Conclusion

# This project presents an end-to-end pipeline for predicting NBA shot success using large-scale shot-level data. Data preparation, feature construction, and missing-data handling were carried out in SAS® Viya® Workbench, resulting in two modeling datasets: a 2.8 million shot complete-case table and a 4.5 million shot imputed table. On top of these datasets, we built regularized logistic regression models, decision tree classifiers, and SVMs on subsamples using scikit-learn.

# All model classes tell a consistent story. Shot distance is by far the most important predictor of whether a shot will go in, with game context (quarter and remaining time) providing meaningful but smaller adjustments. Player size and role matter in intuitive ways, and travel-related variables appear to play at most a secondary role. In terms of predictive performance, regularized logistic regression and decision trees on the full datasets reach similar accuracy and F1 scores, and more flexible SVMs on subsamples do not yield substantial improvements.

# Overall, the results show that even detailed contextual and player-level information can only partially resolve the inherent randomness of shooting outcomes. At the same time, the models capture intuitive and stable patterns in shot success and provide a solid baseline for more advanced SAS-based sports analytics work in the future.

# References

Book <Author name: last name, first name (or initials)>. <Publication date>. <*Book title*>. <City, State (abbrev) of publication> : <Publisher name>.

Journal article <Author name: last name, first name>. <Publication date>. “<Article title>.” <*Journal title*>, <volume no.:page numbers>.

Article in conference proceedings <Author name: last name, first name>. <Publication date>. “<Article title>.” <*Title of proceedings such as Proceedings of the SAS Global 2010 Conference*>, <City, State (abbrev) of publication> : <Publisher name>. Optional: You can add a URL to access available online proceedings. For example: Available at <http://support.sas.com/resources/papers/proceedings09/TOC.html>.

Website <Author name: last name, first name>. “<Title>.” <*Source*>. <Date>. Available at <URL>.

Reference examples:

Book Agresti, A. 2013. *Categorical Data Analysis*. 3rd ed. Hoboken, NJ: John Wiley & Sons.

Journal article Akaike, H. 1979. “A Bayesian Extension of the Minimum AIC Procedure of Autoregressive Model Fitting.” *Biometrika,* 66:237–242.

Article in conference proceedings Dorfman, A. H. and R. Valliant. 1993. “Quantile Variance Estimators in Complex Surveys.” *Proceedings of the Survey Research Methods Section*, 866–871. Alexandria, VA: American Statistical Association.

Website Federal Reserve Bank of St. Louis. 2012. “Economic Research.” Accessed November 7, 2012. <http://research.stlouisfed.org>.

# Acknowledgments

This is the text for the acknowledgments. This paragraph uses the PaperBody style, which uses the Verdana font, not the Arial font.

# Contact Information <heading 1>

Your comments and questions are valued and encouraged. Contact the author at:

<Name>

<Enterprise (optional)>

<Phone (optional)>

<E-mail>

<Web (optional)

SAS and all other SAS Institute Inc. product or service names are registered trademarks or trademarks of SAS Institute Inc. in the USA and other countries. ® indicates USA registration.

Other brand and product names are trademarks of their respective companies.

**Delete the following instructions**.

Basic Instructions

## Writing Guidelines

### Trademarks and product names

To find correct SAS product names (including use of trademark symbols), if you are a SAS employee, see the [Master Name List](http://sww.sas.com/sasnaming/). Otherwise, see [SAS Trademarks](http://www.sas.com/en_us/legal/trademarks.html).

* Use superscripted trademark symbols in the first use in title, first use in abstract, and in graphics, charts, figures, and slides.
* Do not abbreviate product names. For example, you cannot use “EM” for SAS® Enterprise Miner™. After having introduced a SAS product name, you can occasionally omit “SAS” for certain products, provided that your editor agrees. For example, after you have introduced SAS® Simulation Studio, you can occasionally use “Simulation Studio.”

### Writing style

* Use active voice. (Use passive voice only if the recipient of the action needs to be emphasized.) For example:

The product creates reports. (active)  
Reports are created by the product. (passive)

* Use second person and present tense as much as possible. For example:

You get accurate results from this product. (second person, present tense)  
The user will get accurate results from this product. (future tense)

* Run spellcheck, and fix errors in grammar and punctuation.

### Citing references

All published work that is cited in your paper must be listed in the REFERENCES section.

If you include text or visuals that were written or developed by someone other than yourself, you must use the following guidelines to cite the sources:

* If you use material that is copyrighted, you must mention that you have permission from the copyright holder or the publisher, who might also require you to include a copyright notice. For example: “Reprinted with permission of SAS Institute Inc. from *SAS® Risk Dimensions®: Examples and Exercises*. Copyright 2004. SAS Institute Inc.”
* If you use information from a previously printed source from which you haven’t requested copyright permission, you must cite the source in parentheses after the paraphrased text. For example: “The minimum variance defines the distance between cluster (Ward 1984, p. 23)

## Tips for using Word

### To select a paragraph style

1. Click the HOME tab. The most common styles in your document are displayed in the top right area of the Microsoft ribbon. If you don’t see a style that you want, click the slanted down arrow at the bottom right corner of the Styles area, and scroll through the list. The main styles for this template are headings 1 through 4, PaperBody, and Caption. Avoid using other styles.
2. To change a paragraph style, click the paragraph to which you want to apply a style, and then click the style that you want in the ribbon.
3. PaperBody (used for most text) is automatically applied when you press Enter at the end of any heading style or the Caption style.

### To insert a caption

1. Click **REFERENCES** on the main Word menu.
2. Click **Insert Caption**.
3. Select the **Label** type that you want.
4. Click **OK**.

### To insert a graphic from a file

1. Click **INSERT** on the main Word menu.
2. Click **Picture**.
3. In the Insert Picture dialog box, navigate to the file that you want to insert.
4. When the name of the file that you want to insert is displayed in the **File name** box, click **Insert**.