NBA All-Star Player Assessment

Paul Z. Fox

Western Governors University

Table of Contents

A. Project Highlights 4

Research Question or Organizational Need: 4

Scope of the Project: 4

Overview of My Solution: 5

Tools and Methodologies: 5

B. Project Execution 6

C. Data Collection Process 8

C.1 Advantages and Limitations of Data Set 10

D. Data Extraction and Preparation 11

E. Data Analysis Process 13

E.1 Data Analysis Methods 13

E.2 Advantages and Limitations of Tools and Techniques 14

E.3 Application of Analytical Methods 15

F Data Analysis Results 17

F.1 Statistical Significance 17

F.2 Practical Significance 19

F.3 Overall Success 19

G. Conclusion 20

G.1 Summary of Conclusions 20

G.2 Effective Storytelling 20

G.3 Recommended Courses of Action 21

H Panopto Presentation 22

References 23

Appendix A 24

Project Code 24

# A. Project Highlights

## Research Question or Organizational Need:

The research question I attempted to answer with my project was: Do an NBA player’s season statistics of points, assists, rebounds, shooting stats (3-point, 2-point, and free throw makes and attempts) blocks, steals, and team considerations (team played for and team wins) predict whether that player will be named to the All-Star team? My intent was to create an objective means to determine a player’s All-Star worthiness—an approach that would be able to supplement the subjective and biased voting method that ultimately decides who is named to the All-Star team.

## Scope of the Project:

This project focused on building a machine learning model to predict NBA All-Star selections based on historical player and team performance data. The scope included data acquisition, cleaning, feature preparation, model training and tuning, model comparison, final model selection, model evaluation, and model performance visualization. Two models were developed and compared—Random Forest and Logistic Regression—with a Logistic Regression model ultimately being selected and saved The entire process was documented in Jupyter notebooks.

The project scope did not include deploying the model, any future updates, or attempting to influence or replicate the actual All-Star voting process. It was purely a data-driven exploration of who should be selected based on performance alone.

## Overview of My Solution:

My solution was to train and compare predictive models—both Random Forest and Logistic Regression—that were developed to classify NBA player-seasons into All-Star or non-All-Star seasons.

I obtained and explored a comprehensive dataset (linked in the Tools and Methodologies section below) that included all of the player, team and awards data I was interested in. I cleaned, merged, encoded, and scaled the data as necessary. I trained and tuned my candidate models based on classification metrics such as F1 score, recall, and precision. Once the models were trained, they were compared primarily on F1 score for the minority class (with a secondary preference for recall). I selected a base Logistic Regression model based on its performance and interpretability, and I summarized and created visualizations for the model’s performance.

## Tools and Methodologies:

I used Python as the project’s programming language. I created a conda environment with package requirements such as Jupyter for running code and documenting steps, pandas for handling data, scikit-learn for machine learning, and Matplotlib and Seaborn for data visualization. Other tools I used included GitHub for project tracking and Kaggle as a data source. I did not use existing third-party code except the Python packages listed above. I followed the CRISP-DM (Cross-Industry Standard Process for Data Mining) methodology to inform the stages of development. I used the [NBA Stats (1947-present)](https://www.kaggle.com/datasets/sumitrodatta/nba-aba-baa-stats) on Kaggle (Datta, 2025).

# B. Project Execution

I was able to stick to my project plan quite well, with a couple minor changes to better facilitate organization and flow (as detailed below).

**Project Plan:**

The project goal was to develop a machine learning model that could accurately predict whether an NBA player will be selected as an All-Star in a given season, based on season-level performance and team data. This goal was met by achieving the following three objectives:

* Objective 1: Acquire and prepare relevant datasets, including player season statistics, team records, and historical All-Star selections.
  + Planned Outputs (deliverables 1.1 - 1.4 from task 2): clean\_data\_reference.csv, X\_data.pkl, y\_data.pkl, data\_prep.ipynb
  + Changes to Plan: There were no changes between plan and execution for this objective. I successfully acquired and prepared data as expected. The data\_prep notebook ran successfully and produced the planned deliverable outputs.
* Objective 2a and 2b: (a) Train and evaluate multiple classification models to determine the most effective approach for predicting All-Star selections. (b) Select the final model based on performance and interpretability, and save it for future use.
  + Planned Outputs (deliverables 2.1 - 2.2 from task 2): model\_comparison.ipynb, all\_star\_model.pkl
  + Changes to Plan: While I did deliver the planned output files, it became clear to me during execution of this objective that my project flow would require three additional outputs from the model\_comparison notebook: X\_test.pkl, y\_test.pkl, and logreg\_scaler.pkl. These three unplanned outputs were an oversight from task 2, and I added them to facilitate the evaluation of my chosen model in task 3. X\_test.pkl and y\_test.pkl were required to make test set predictions in the model\_evaluation notebook (Objective 3), and logreg\_scaler.pkl was needed for proper use of the saved model.
* Objective 3: Evaluate the selected model and visually communicate its performance.
  + Planned Outputs (deliverables from task 2): model\_evaluation.ipynb
  + Changes to Plan: There were no changes between plan and execution for this objective. I evaluated and visualized the performance of my model in the successfully-implemented model\_evaluation notebook.

**Project Planning Methodology:**

No changes were made while executing the planned methodology. My project utilized CRISP-DM to inform all stages of development. *Business Understanding* was attained by defining the problem of subjectivity in NBA All-Star selection and exploring how a data-driven model can support more objective evaluations. *Data Understanding* was achieved through exploration of datasets containing player statistics, team records, and historical All-Star selections—and by identifying patterns, checking for inconsistencies, and assessing completeness. The *Data Preparation* process involved cleaning and merging the datasets, resolving naming conflicts and missing values, handling player-seasons with multiple teams, encoding categorical variables, and scaling numerical features as needed. *Modeling* was done by training and tuning Random Forest and Logistic Regression models using RandomizedSearchCV and classification metrics such as F1 score, recall, and precision. The *Evaluation* phase consisted of comparing models and selecting the one that performs best on the test data, with a focus on F1 score for the minority class (and a secondary preference for recall). The final model selection was justified based on both performance and interpretability, and a notebook was created to visualize and summarize model performance. Formal *Deployment* was out of scope, but the final model was saved in .pkl format for reproducibility and potential future use.

**Project Timeline and Milestones:**

Despite setting a seemingly aggressive timeline for project completion, I did end up finishing ahead of schedule. Data collection went as planned, taking one day to find and download a suitable, freely available dataset (linked in references section). Cleaning and saving data also took one day, as planned. The data I was working with was relatively clean and complete to begin with, so minimal adjustments were needed to prepare the data for modeling. Training and comparing models was also a quick process, finishing in half the time I had expected. It only took one day to train, compare and select my final model. Everything went smoothly, so I didn’t need the extra day. Evaluation, visualization, and summary of performance also took less time than allotted. I had tried to create a reasonably quick timeframe for the project, but with a bit of wiggle room to address issues that could potentially delay me. Luckily, each step was smooth and seamless, so I was able to complete it all in 4 days.

# C. Data Collection Process

**Data Selection and Collection:**

I chose a free, publicly available dataset that contained everything I was looking for. The data included season-level player statistics, team records, and actual All-Star selections—all of which were necessary to train and evaluate the model for predicting All-Star appearances. I downloaded the complete dataset as a ZIP folder of 17 CSV files from *<https://www.kaggle.com/datasets/sumitrodatta/nba-aba-baa-stats/data>*. I ended up using three files: “All-Star Selections.csv”, “Player Totals.csv”, and “Team Summaries.csv”. These files were added to my local project folder and tracked in GitHub. From there, I read the CSV files into a Jupyter notebook as DataFrames using pandas.read\_csv. All of this went exactly as planned. No changes were made from task 2.

**Obstacles Encountered:**

My chosen dataset was reasonably clean and complete, though some processing was required. After filtering the data for my desired timeframe (1977-2024), there were only four rows with actual missing values. Other rows that seemed to have missing values were because of things like total rows and other inconsistent records. NaN values in the “x3p” and “x3pa” columns represented zero attempted shots and zero makes, so those values were imputed with zero. Some combining and dropping was performed to engineer a temporary “primary team” column, which I used to select the single team a player played the most games for in a season in the case of mid-season player trades. The feature “player\_id” was not being used consistently across all files. That created a need to compare the list of player-seasons against All-Star selected player-seasons to check for potential seasons in which (a) multiple players had the same name and (b) one of those players was named to the All-Star team (luckily there was not any overlap on the All-Star list, so no further cleaning decision needed to be made in that regard).

**Data Governance Issues:**

I did not encounter any unplanned issues related to data governance, privacy, security, or compliance. I kept raw and cleaned data files in separate folders in my project structure (data/raw and data/clean) under version control. All changes to data preparation code and to data files were tracked in GitHub with descriptive commit messages. No security issues came up, as the project files all reside in a secured local directory, behind user credentials, and I am the sole user with contributor permissions to the project on GitHub. Privacy and compliance were not relevant issues for this project, as all data was publicly available player and team data for professional athletes, with no personally identifiable information included (other than players’ names), and because all third-party libraries used were open-source and included under their licensed terms.

## C.1 Advantages and Limitations of Data Set

**Advantages:**

One major advantage of the dataset I chose was the sheer volume of available features and multiple representations of those features. When initially exploring the data, I realized I could view players’ 3-point prowess (for instance) in many ways. Not only could I view makes, attempts, and percentages, but I could view all of those from varying distances, or I could view them broken up per game, per 100 possessions, per 36 minutes, or even by play.

Another big advantage to this dataset was the impressive level of historical completeness. I ultimately decided to focus on the post-merger era (beginning in 1977), but this dataset contained quality data for 30 years prior, across both the NBA and ABA.

**Limitations:**

While I was very happy with the data overall, a huge limitation I found was inconsistent usage of the “player\_id” column. While the main player data carried an associated identifier, this column was not available on important tables, such as the All-Star Selections data. This omission caused extra work for me, as I had to check for seasons where multiple players with the same name competed and potentially made the All-Star team.

# D. Data Extraction and Preparation

**Data Extraction:**

The data extraction process for this project began with finding the perfect dataset for the purposes of building All-Star classification models. I chose the [NBA Stats (1947-present)](https://www.kaggle.com/datasets/sumitrodatta/nba-aba-baa-stats) dataset, which is comprised of 17 CSV files, filled with detailed player and team statistics. I manually downloaded the dataset as a ZIP folder. I used a combination of the dataset’s documentation on [kaggle.com](http://kaggle.com), and directly inspecting the CSV files with Excel (because of it’s automatic formatting and ease of use for inspecting tabular data) to narrow my data selection to the three files ultimately included in the project. Once the files were chosen, I added them to my local project folder. Using Python, I created a data\_prep Jupyter notebook and loaded only my desired columns from the files as DataFrames using pandas. I chose Python because it allows me to use pandas and Jupyter. I chose pandas because it is a fantastic package for working with tabular data as DataFrames. I decided to house my code in Jupyter because I could easily document the process while exploring the data, and I could edit and re-run small pieces of code easily.

**Data Preparation:**

Once I had my data extracted and loaded as DataFrames into my data\_prep notebook, some processing was needed, all of which was done using pandas in my data\_prep notebook. I filtered the data to only show from 1977 to 2024, as this was the time range of interest. I removed a few rows with missing data because it was an insignificant number of rows, and there was no sensible way to impute the missing values. Other rows with missing values were imputed with zero, because those nulls corresponded with player-seasons with no 3-point shot attempts or makes. I engineered a temporary primary team column from existing team and games played data, and I used that to apply to players who had played for more than one team in a season. This seemed like the cleanest way to provide team data in these instances, assigning players the team they played the most games for that season. I checked and acted on a couple corner case possibilities, such as a Magic Johnson All-Star appearance in a year after he retired. This row was removed from the All-Star Selections list. Once I was happy with the data, I merged it into a single DataFrame, and I saved three files: one for feature data (X\_data.pkl), one for the target variable (y\_data.pkl), and one human-readable reference (clean\_data\_reference.csv). I chose pickle files for my X and y data because they’re efficient and they preserve the DataFrame objects and datatypes. I chose CSV for my reference data because it’s easily opened in a spreadsheet program or text editor for manual inspection.

# E. Data Analysis Process

## E.1 Data Analysis Methods

This project used supervised machine learning to classify NBA players as either All-Stars or non-All-Stars based on their season performance and team success. The two models I used were Random Forest and Logistic Regression, both of which were appropriate choices for supporting the project’s hypothesis.

I used Random Forest because it is a powerful ensemble model that can capture complex, nonlinear relationships in the data. It is particularly good at handling datasets with many features, which made it a strong choice for uncovering hidden patterns that influence All-Star selections. Random Forest was an appropriate choice because it provided a performance-focused way to model the actual selection results, which helped evaluate whether the hypothesis held true across a wide range of player types and team situations.

I also used Logistic Regression because it is a straightforward and interpretable model that estimates the probability of a binary outcome (whether or not a player is selected as an All-Star). Logistic Regression was an appropriate choice because it could show how much each individual variable contributed to the likelihood of selection. This aligned closely with the project’s goal of offering a transparent and explainable view of how performance relates to selection, and it helped answer the core research question by making the relationship between performance and outcome easier to interpret.

By comparing both models, I was able to evaluate how well performance-based data supported All-Star predictions in both predictive accuracy and interpretability, giving a well-rounded answer to the hypothesis.

## E.2 Advantages and Limitations of Tools and Techniques

The main techniques I used in this project were supervised classification models—specifically, Random Forest and Logistic Regression. Each had its own strengths and drawbacks, and using both helped me evaluate my hypothesis from different angles.

* Random Forest was a great choice for prediction power. It can handle lots of features and figure out complex relationships in the data without me having to do a lot of preprocessing. It also gives built-in feature importance, which is helpful for understanding what factors the model thinks matter most. That said, one downside is that it’s not as easily interpreted as a model like Logistic Regression, and that makes it less helpful if someone wants to understand the model’s reasoning.
* Logistic Regression was much easier to understand. It’s a simple model, and it gives coefficients that show how much each feature increases or decreases the odds of being an All-Star. This made it useful for transparency. The downside is that it assumes more of a straight-line relationship between features and outcomes, and that might not always be true. It also can be sensitive to multicollinearity, as I saw between my games played, minutes played, and points columns.
* Using supervised classification in general made sense for this project. I had labeled data, and I wanted to predict a binary outcome—whether a player made the All-Star team or not—so classification was the natural fit. The biggest limitation is that the model can only learn from what’s in the data. And since All-Star selections involve fan and media voting, the model may end up learning patterns that reflect popularity as much as performance.

## E.3 Application of Analytical Methods

I used two supervised classification models in this project: Random Forest and Logistic Regression. Below is a step-by-step explanation of how I applied each model to the data, along with how I checked that the data met the requirements for each method.

**Step 1: Prepare the Data**

After cleaning and merging the raw data, I split it into features (X) and a target label (y), where the label indicated whether the player was selected as an All-Star (1) or not (0). I then split this data into training and testing sets using train\_test\_split, stratifying on the target variable to preserve class distribution.

* For Random Forest, no additional preprocessing was required.
* For Logistic Regression, I scaled the numerical features using StandardScaler, since the model assumes that input features are on similar scales.

**Step 2: Train Baseline Models**

I first trained a basic version of each model using scikit-learn’s RandomForestClassifier and LogisticRegression. Both models were fit to the training data, and then predictions were made on the test set. I evaluated the results using a classification report, focusing on F1 score (especially for the minority class), along with precision and recall.

**Step 3: Tune Hyperparameters**

In an attempt to improve model performance, I ran RandomizedSearchCV for each model. This allowed me to test different combinations of parameters that might optimize F1 score.

* For Random Forest, I tuned things like number of trees, max depth, and class weights.
* For Logistic Regression, I tuned the regularization strength and tested class weights.

**Step 4: Compare Model Performance**

After tuning, I compared the baseline and tuned versions of each model using their F1 scores on the test data (with a secondary look at recall). I ultimately selected the base Logistic Regression model because it performed well, was easier to interpret, and closely aligned with the goals of the project.

**Step 5: Verify Assumptions and Requirements**

Each model has certain expectations about the input data:

* Random Forest does not require feature scaling and can handle nonlinear relationships, missing values (if present), and categorical variables once encoded. The only key requirement is that the input features be numeric, which I confirmed by using pd.get\_dummies() for team data and ensuring no missing values.
* Logistic Regression assumes:
  + The relationship between predictors and outcomes is linear. I didn’t formally test this, but I evaluated individual coefficients later for sanity checks.
  + Features should not be highly correlated. I noticed some mild multicollinearity, particularly with g (games played), and documented that in the final evaluation notebook.
  + Features should be scaled, which I ensured using StandardScaler.

Each step was done in Jupyter notebooks to keep the process modular and reproducible. All files were versioned in GitHub, and the selected model was saved as a .pkl file for use in evaluation and future applications.

# F Data Analysis Results

## F.1 Statistical Significance

**Type of Model:**

This project used a supervised classification model to predict a binary outcome: whether or not an NBA player would be selected as an All-Star in a given season. The chosen model was Logistic Regression, which estimates the probability of class membership—in this case, being selected as an All-Star—based on a linear combination of input features.

**Algorithm and Process:**

The Logistic Regression model was implemented using scikit-learn’s LogisticRegression class. Training was done on feature data that included season-level statistics and team information, with numerical features scaled using StandardScaler and team data one-hot encoded. I experimented with hyperparameter tuning using RandomizedSearchCV, evaluating different values for regularization strength and class weight, and I used 5-fold cross-validation to reduce overfitting. I selected F1 score as the scoring metric during this process to balance both recall and precision. Despite tuning efforts, the baseline Logistic Regression model (without regularization adjustments) performed best on the test data.

**Performance Metrics:**

To evaluate performance, I used the F1 score as the primary metric. This score balances precision (how many of the predicted All-Stars were correct) and recall (how many actual All-Stars were correctly identified), which is essential given the class imbalance in the dataset.

**Benchmark for Success:**

In Task 2, I established that the model would be considered successful if it achieved an F1 score of at least 0.70 on the test set for the minority class (All-Stars). This benchmark represented a meaningful level of predictive power for a task with inherently subjective outcomes and imbalance between positive and negative classes.

**Results:**

The baseline Logistic Regression model achieved an F1 score of 0.715, which meets and exceeds the defined threshold. It also achieved a precision of 0.776 and a recall of 0.662 for All-Star predictions. These values indicate the model was not only successful in finding true All-Stars but also in avoiding too many false positives.

**Conclusion:**

The Logistic Regression model successfully met the criteria for statistical significance. Its strong F1 score, combined with reliable precision and recall, provides clear support for the hypothesis that a player’s season performance and team success can be used to predict All-Star selection. Based on these results, the null hypothesis—that performance and team metrics do not predict All-Star selections—can be rejected.

## F.2 Practical Significance

The Logistic Regression model didn’t just meet the statistical benchmark—it produced results that are useful in a real-world context. With an F1 score of 0.715, it showed that season stats and team performance can meaningfully predict All-Star selections. This kind of model could help NBA analysts or fans identify overlooked players, especially those from smaller markets who might be undervalued in the current voting system. While it’s not meant to replace the existing process, it adds value as a fair, data-driven complement to subjective opinions.

## F.3 Overall Success

This project successfully answered the research question by building a model that can predict NBA All-Star selections using season-level stats and team performance. The final Logistic Regression model met the benchmark F1 score of 0.70, confirming that the data supports meaningful predictions.

Along the way, I cleaned and prepared a complex dataset, trained and compared models, selected the best one based on performance and interpretability, and visualized the results. Every planned deliverable was completed, and the project came in ahead of schedule.

In the end, the model achieved both statistical and practical significance. It not only performed well on test data but also demonstrated real-world value as a tool for supporting more objective evaluations of All-Star candidates.

# G. Conclusion

## G.1 Summary of Conclusions

The goal of this project was to determine whether NBA All-Star selections could be predicted using season-level performance data and team context. The hypothesis was that stats like points, assists, rebounds, shooting, and team wins could provide a reliable basis for predicting All-Star appearances. Based on the results of the final model, that hypothesis is supported.

The Logistic Regression model exceeded the success benchmark defined earlier in the project, achieving an F1 score of 0.715 on the test set. This indicates that All-Star selections can, in fact, be meaningfully predicted using performance data alone.

Beyond the model itself, the project also met all defined process goals. Every step of the CRISP-DM process was followed. Data was acquired and cleaned. Features and targets were prepared and saved. Models were trained, tuned, and compared, and the chosen model was evaluated and visualized. The entire process was completed according to plan and documented in a series of well-organized, reproducible Jupyter notebooks.

## G.2 Effective Storytelling

To effectively communicate the findings of this project, I relied on in-document markdown, as well as clear, targeted visualizations. I used Matplotlib and Seaborn—two powerful Python visualization libraries—within Jupyter notebooks to display model results in an accessible and visually intuitive way.

The confusion matrix was presented as a heatmap to highlight true positives, false positives, false negatives, and true negatives. I created a normalized version to make the proportions easier to interpret at a glance. This helped contextualize how well the model performed on the minority class.

For the selected Logistic Regression model, I visualized the top ten coefficients to show which features had the most impact—positive or negative—on All-Star prediction. This supported transparency and interpretability, aligning with the project’s goal of using data to shed light on what drives All-Star selections.

Together, the visuals and markdown supported a clear, data-driven narrative: performance data can predict All-Star selections, and the model’s behavior can be understood and explained.

## G.3 Recommended Courses of Action

Given both the success of our model in achieving the project goal, and some of the areas for improvement, I recommend the following two courses of action:

1. **Use the Model to Identify Overlooked All-Star Candidates.**NBA analysts, team staff, or media members could use this model to flag players whose performance metrics suggest they should be All-Stars—even if they aren’t receiving national attention or fan support. This would help promote fairness by giving more visibility to players in smaller markets or on underperforming teams. The model could serve as a check against bias in the voting process and be used to supplement—but not replace—human judgment.
2. **Expand and Refine the Model Using Additional Data.**  
   Future versions of this project could include more features such as per-minute stats, advanced metrics like player efficiency rating (PER), or even external factors like social media engagement or media coverage. Including multiple years of All-Star voting rules (as they’ve changed over time) could also improve predictive accuracy. Expanding the dataset would allow the model to evolve and potentially provide even more accurate and nuanced insights into what truly defines an All-Star season.

# H Panopto Presentation

My Panopto presentation can be found at:

* <https://wgu.hosted.panopto.com/Panopto/Pages/Viewer.aspx?id=e0b3c3e4-d688-4ae0-9f11-b2e100c691bf>

# References

Sumitro Datta. (2025, April 1) *NBA stats (1947-present)*. Kaggle. https://www.kaggle.com/datasets/sumitrodatta/nba-aba-baa-stats/data

# Appendix A

# Project Code

My project code and submissions can be found and viewed on GitHub at:

* <https://github.com/PaperBowser/Capstone>