

BANA 273 LEC A: MACHINE LEARNING ANALYTICS

Predicting the 2026 MLB All-Star Team

Project Report



Jaya Sruthi Raj Perikala, Carson Pimental

Table of Contents

1. Executive Summary.....	2
2. Introduction	2
3. Data	2
3.1. Data Pre-Processing	4
3.2. Data for Modeling	4
3.3. Data for Predictions	6
4. Analysis	6
4.1. Evaluation	6
4.2. Modeling Technique	7
4.3. Logistic Regression	8
4.3.1. Benchmark	8
4.3.2. Regularization with Class Weight	8
4.3.3. Regularization with SMOTE (Optimal Ratio)	9
4.4. Decision Tree.....	10
4.4.1. Benchmark	10
4.4.2. Pre-Pruning	11
4.4.3. Post-Pruning.....	11
4.5. Random Forest.....	12
4.5.1. Benchmark	12
4.5.2. Random Forest w/ Tuning	13
4.6. Naive Bayes	14
4.6.1. Benchmark	14
4.6.2. Gaussian Naive Bayes (Calculated Priors)	14
4.6.3. Gaussian Naive Bayes with SMOTE.....	15
5. Findings	16
6. Data Mining and Key Takeaways	17
6.1. Data Mining	17
6.2. Key Takeaways.....	18
7. Predictions	19
8. Business Implications	20
8.1. Budget for Expected Increases in Player Salary.....	20
8.2. Marketing Campaigns	20
9. Noteworthy Data Considerations	21
9.1. Data Assumptions	21
9.2. Data Limitations	22

1. Executive Summary

As the 2025 Major League Baseball season draws to a close, anticipation begins to build for the sport’s return in the spring. In the meantime, our attention turned to a question beyond typical box scores: can future performance and/or recognition be predicted? Looking ahead to the 2026 season, we wanted to see if we could predict next year’s MLB All-Star lineup, and ultimately sought to build a model that would consider each player’s offensive statistics and produce their probability of being rewarded with an All-Star selection.

2. Introduction

By using All-Star selections as our target variable, we don’t just measure performance, we also estimate the level of attention and recognition a player will receive. Rather than previously set qualifications, All-Stars are voted on by fans, players, and coaches, indicating that selections can reflect perception and popularity just as much as performance.

Understanding how players are perceived around the game in advance can help the team plan for future payroll increases. One example of this appears during arbitration hearings, where players and their agents leverage accolades such as All-Star selections to argue for higher salaries. These accomplishments are used as evidence of a player’s rising value, often putting teams in a difficult position when a strong season has already elevated their market perception. This results in the team having to increase the player’s salary, thus causing an unexpected spike in payroll that could have been anticipated prior. Recognizing perception early can also help the team decide which players to promote in marketing campaigns, in order to build excitement around players who are likely to become stars and capitalize on their predicted success.

3. Data

The dataset, sourced from Kaggle, contains Major League Baseball batting statistics spanning the 2015–2024 seasons. It includes 4,502 rows and 36 columns, where each row represents a player-season record with performance metrics such as Batting Average (BA), On-Base Percentage (OBP), and Slugging Percentage (SLG). The dataset also includes an ‘Awards’ column that lists any awards a player received in that particular season. The following are the description of variables available in the dataset:

AS	Binary indicator of whether the player was selected as an All-Star (1 = selected, 0 = not selected)
----	---

Offensive Performance Metrics	
WAR	A comprehensive metric estimating the number of additional wins a player contributes compared to a replacement-level player. Higher WAR values indicate greater overall value
G	Total number of games in which the player appeared during the season
PA	Total trips to the plate, including at-bats, walks, hit-by-pitch, and sacrifice plays
AB	Number of official batting attempts excluding walks, hit-by-pitch, sacrifices, and catcher interference
R	Number of times the player safely crosses home plate
H	Total number of successful hits, including singles, doubles, triples, and home runs
2B	Number of hits where the batter safely reaches second base
3B	Number of hits where the batter safely reaches third base
HR	Total number of home runs hit by the player
RBI	Number of runs the player drives in through their hitting actions (excluding runs scored from errors or double plays)
SB	Number of bases the player successfully steals
CS	Number of times the player gets out while attempting to steal a base
BB	Number of times the player reaches first base after receiving four balls
SO	Number of times the player is retired via three strikes
Rate Statistics	
BA	Percentage of at-bats resulting in a hit
OBP	Frequency a player reaches base, via hits, walks, or hit-by-pitches
SLG	Measures power by accounting for extra-base hits

OPS	Combined measure of OBP and SLG
Additional Counting Stats	
TB	Cumulative number of bases gained from hits
HBP	Number of times the batter reaches base after being struck by a pitched ball.
SH	Number of bunts that advance a runner while the batter is intentionally put out.
SF	Number of fly balls that result in a run scored, with the batter being out
IBB	Number of times the player is intentionally walked by the pitcher
Contextual Variables	
Lg	Indicates the league the player plays in: American League (AL) or National League (NL)
Team	MLB team for which the player competed during the season
Year	The season/year corresponding to the player statistics
Rk	Player's ranking based on the statistical source's ordering

3.1. Data Pre-Processing

We created a binary target variable, 'AS', which takes the value 1 if a player received an All-Star selection and 0 otherwise. The dataset contains no missing values, and we retained all outliers because each extreme value reflects a valid real-world performance scenario.

3.2. Data for Modeling

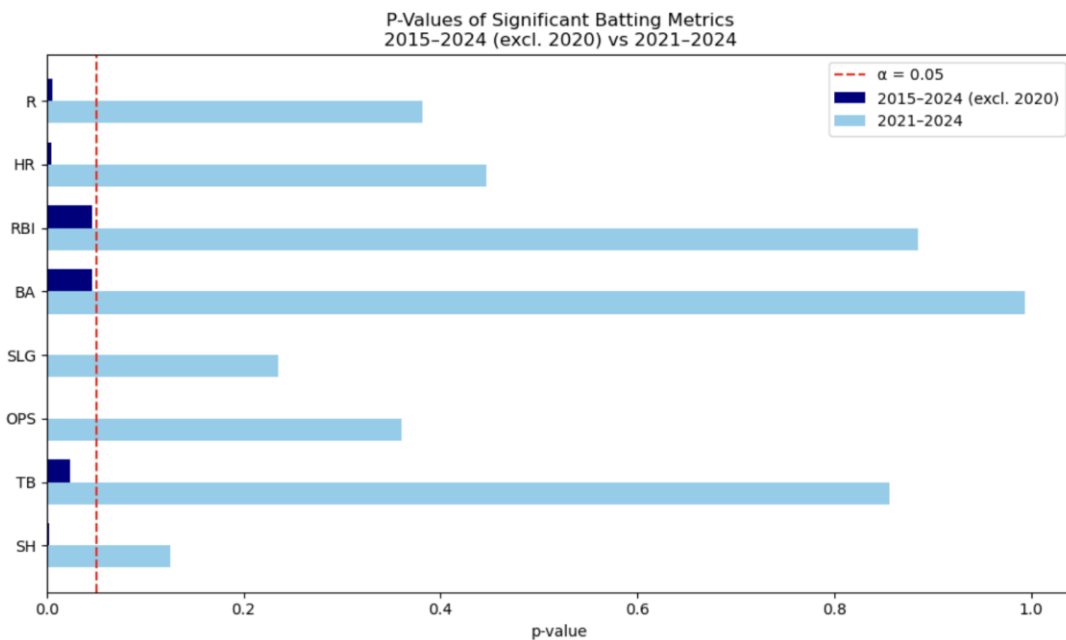
Since our dataset spans the past ten seasons, we recognized that the metrics valued by voters every year could have shifted with time. Since we ultimately want to predict All-Stars for 2026, it was essential for our model to form its predictions on current league expectations. To account for this, we needed to ensure that the level of performance needed to become an All-Star did not evolve over the course of our data. We compared mean offensive statistics across two distinct time periods (while excluding the 2020

season, which was shortened to 60 games by the COVID-19 pandemic and did not feature an All-Star game):

- 1.) Full range: 2015–2024 (excluding 2020)
- 2.) Post-pandemic period: 2021–2024

Our statistical tests showed that eight offensive metrics differed significantly across the full 2015–2024 period, indicating instability in performance trends. Alternatively, no significant differences were observed during the 2021–2024 period, implying greater consistency of standards across the recent, post-pandemic seasons. Based on this stability, we selected the 2021–2024 period for our All-Star performance analysis and model development.

The following chart shows the p-values for the eight metrics that have significant differences for the time period of 2015–2024 (excluding 2020), but insignificant differences for the more recent 2021–2024 time period.¹



Additionally, each season is treated as an independent observation of a player's performance, so repeated appearances of the same player across seasons do not

¹ We set our significance level (α) to 0.05, meaning that metrics with p-values less than 0.05 were considered statistically significant, while those with p-values greater than 0.05 were deemed not significant.

introduce correlation into the dataset. The ratio of All-Stars to non-All-Stars in the 2021–2024 seasons is approximately 1:9, indicating that the dataset is highly imbalanced.

3.3. Data for Predictions

In order to make predictions, we needed a dataset containing performances that have not occurred yet. We downloaded a second dataset from Steamer Projections on Fangraphs that showed each players' projected stats for the upcoming season. We will ultimately plug this data into our final model and have it return each players' probability of being selected to the All-Star team.

4. Analysis

4.1. Evaluation

Our goal was to identify as many All-Stars as possible while avoiding excessive misclassification of non-All-Stars. We must note the severe imbalance of our target variable, with All-Stars comprising only 10% of the observations. Due to this imbalance, accuracy could not be used as an evaluation metric, since a model that predicts every player as a non-All-Star would still achieve an accuracy score of roughly 95%. This makes accuracy misleading and uninformative for our objective of identifying true All-Stars. To achieve our objective, we designed a custom evaluation metric that incorporates both recall and precision, with greater emphasis on recall to reflect the higher cost of missing true All-Stars.

The evaluation metric used in this study is the F_β score, which provides a weighted harmonic mean of precision and recall. When $\beta > 1$, the metric places greater emphasis on recall. We selected $\beta = 1.24$, corresponding to approximately 60% weight on recall and 40% on precision, to reflect the higher importance of correctly identifying true All-Stars in our imbalanced dataset. In this analysis, precision and recall are computed with respect to the All-Star class, which is treated as the positive (1) class. The metric is calculated using the following formula:

$$F_\beta = (1 + \beta^2) \cdot \frac{\text{Precision} \cdot \text{Recall}}{(\beta^2 \cdot \text{Precision}) + \text{Recall}}$$

This ensures that the model:

- Prioritizes identifying as many true All-Stars as possible
- Maintains a reasonable level of precision to avoid excessive false positives

- Results in a balanced and organization aligned evaluation of model performance

To account for our binary target variable, we elected to run the following Supervised Learning algorithms:

- 1.) Logistic Regression
- 2.) Decision Tree
- 3.) Random Forest
- 4.) Naive Bayes

4.2. Modeling Technique

All models were implemented using the Scikit-learn library in Python. Models were trained on a random 70% subset of the data, with the remaining 30% reserved for validation. Once we had split our data, we began training our models. Across each model, we followed a consistent workflow to ultimately evaluate models that have been trained in comparable structures. We created a machine learning pipeline for each model to standardize preprocessing and ensure a correct training workflow. By placing the StandardScaler inside the pipeline and running it through GridSearchCV with cross validation, all preprocessing steps are fit only on the training portion of each fold. This prevents information from leaking into the validation folds and ensures that model performance is evaluated fairly and without bias.

Next we defined a set of hyperparameters appropriate for the respective model, and used GridSearchCV to run stratified 5-fold cross-validation to evaluate each of the potential parameter combinations. The stratification ensures an equal distribution of the target class (All-Star) across each fold, and we used 5 folds to best compensate for our smaller dataset, after 10-fold tests surprisingly proved to perform slightly worse. Within GridSearchCV we specified to optimize F- β score, causing it to train multiple versions with distinct combinations and select the values that produced the highest F- β score from the validation folds.

The model was then evaluated across threshold values ranging from 0.25 to 1.0 in increments of 0.05. We began at 0.25 because lower thresholds tend to produce excessive false positives in imbalanced datasets, making them unsuitable for our objective. The threshold that achieved the highest F- β score on the test set was selected as the model's final decision cutoff.

Finally, we displayed a classification report, to observe the model's recall and precision performances for the true All-Star class (AS = 1), along with a confusion matrix to

examine the distribution of true positives, true negatives, false positives, and false negatives.

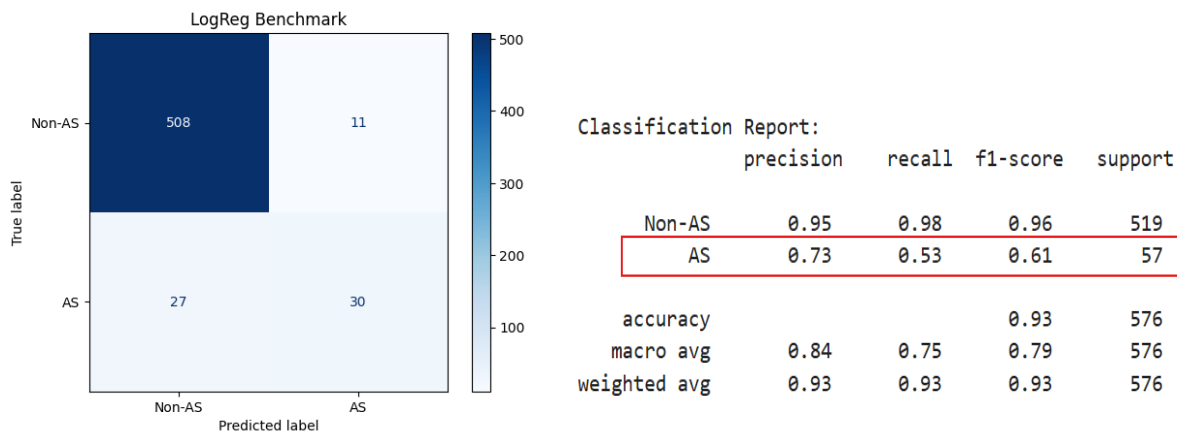
4.3. Logistic Regression

Logistic regression is well-suited for binary classification and serves as a strong baseline for predicting whether a player is an All-Star. Its simplicity, interpretability, and effectiveness with smaller datasets make it a natural starting point before evaluating more complex models. The following logistic function maps the linear combination of player statistics to a probability between 0 and 1, providing clear and meaningful insights into All-Star likelihood.

$$p = \frac{1}{1 + \exp(-\alpha - \beta X)}$$

4.3.1. Benchmark

A baseline benchmark was established using a simple logistic regression with default parameters and no feature scaling. This baseline helps us determine whether more complex models provide meaningful performance improvements. The model achieved a recall of 0.53 and a precision of 0.73, indicating that it misses a substantial number of true All-Stars despite maintaining reasonable precision.



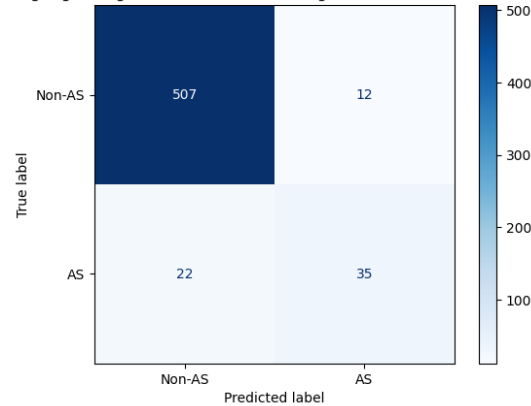
4.3.2. Regularization with Class Weight

Using the modeling framework in Section 4.1, the hyperparameters tuned included the regularization type, regularization strength, and class weight. GridSearchCV selected the optimal configuration, which selected a regularization strength of 100, class weight as none, and L1 (lasso) regularization. The Final predictions were made using the optimized threshold of 0.42.

The model achieved an F- β score of 0.59, with precision of 0.74 and recall of 0.61. This is a clear improvement over the benchmark model's recall of 0.53, meaning it identifies

more true All-Stars. Although precision decreased slightly, the gain in recall makes this model more suitable for our objective, as missing true All-Stars is more costly than additional false positives.

LogReg + Regularization w/ Class Weight (best threshold = 0.42)



Classification report at best threshold:

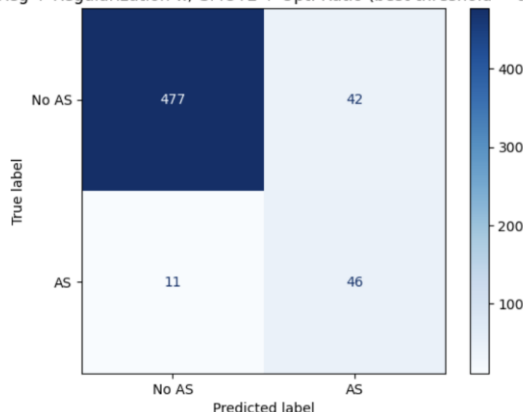
	precision	recall	f1-score	support
Non-AS	0.9584	0.9769	0.9676	519
AS	0.7447	0.6140	0.6731	57
accuracy				0.9410
macro avg	0.8515	0.7955	0.8203	576
weighted avg	0.9373	0.9410	0.9384	576

4.3.3. Regularization with SMOTE (Optimal Ratio)

Rather than testing for balanced and non-balanced class weights, we applied SMOTE to artificially generate synthetic samples of our minority-class (All-Stars), and ensure the model was trained on a balanced dataset. We tuned the SMOTE sampling ratio through GridSearchCV, testing values from 0.1 to 1, varying in increments of 0.1. The best hyperparameters were identified to be a regularization strength of 1, L2 (Lasso) Regularization, and a SMOTE ratio of 0.3. The final predictions were generated with an optimized threshold of 0.33.

This model produced an F_β score of 0.65, and a recall score of 0.81, eclipsing the respective values from the previous model that tested for class weight (0.61, 0.74). Similar to before, we accept this new model even despite the new, slightly lower precision score (0.52) in an effort to raise recall and reward the reduction of false negatives.

LogReg + Regularization w/ SMOTE + Opt. Ratio (best threshold = 0.33)



Classification Report at Best Threshold:

	precision	recall	f1-score	support
No AS	0.98	0.92	0.95	519
AS	0.52	0.81	0.63	57
accuracy				0.91
macro avg	0.75	0.86	0.79	576
weighted avg	0.93	0.91	0.92	576

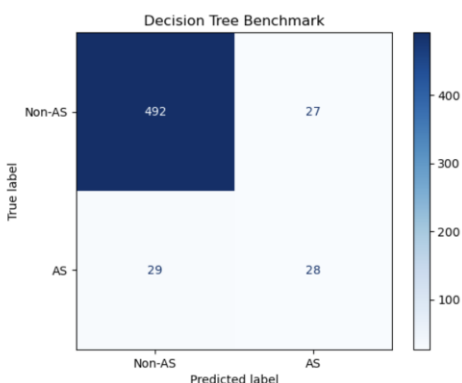
4.4. Decision Tree

Decision trees naturally capture nonlinear relationships and interaction effects between player performance metrics, which can be valuable for identifying All-Star players.

Unlike linear models, they split the feature space into intuitive decision rules, making them easy to interpret. These properties make decision trees a strong candidate for exploring complex patterns complementing insights from more linear or probabilistic models.

4.4.1. Benchmark

A baseline benchmark was established using a simple decision tree with default parameters and no pruning. The unpruned tree reflects the model's raw ability to separate All-Star and non-All-Star players without any optimization or complexity control, and it may overfit the data as a result. The model achieved a recall of 0.49 and a precision of 0.51, indicating that it struggles to correctly identify a large portion of true All-Stars while also offering only moderate precision when predicting them.



Classification Report for Decision Tree Benchmark:

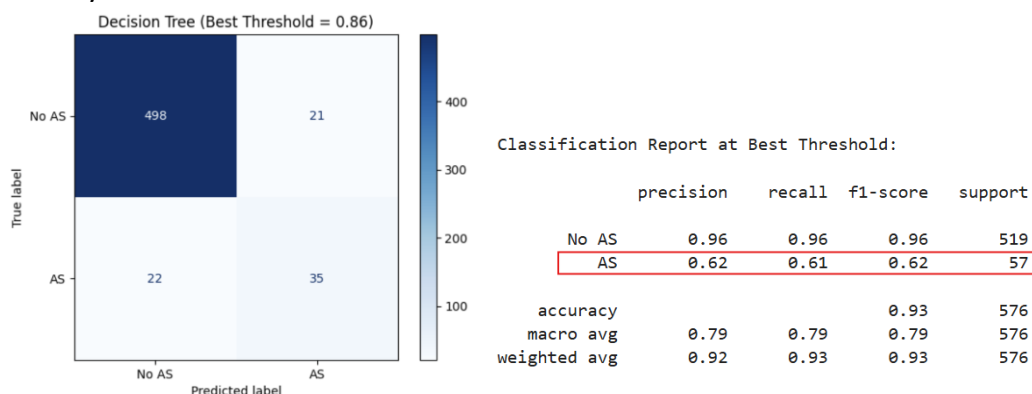
	precision	recall	f1-score	support
Non-AS	0.94	0.95	0.95	519
AS	0.51	0.49	0.50	57
accuracy			0.90	576
macro avg	0.73	0.72	0.72	576
weighted avg	0.90	0.90	0.90	576

4.4.2. Pre-Pruning

Using the modeling framework in Section 4.1, the hyperparameters tuned included the maximum tree depth, minimum samples to split a node, minimum samples at a leaf node, splitting criterion, class weight strategy. GridSearchCV identified the optimal configuration as Gini splitting criterion, no maximum depth, a minimum of four samples per leaf, two samples required to split a node, and class weight of balanced. The final model predictions were generated using an optimized classification threshold of 0.86.

The model achieved an F- β score of 0.49, with precision of 0.62 and recall of 0.61. Compared to the benchmark model's recall of 0.49, the pre-pruned decision tree improves recall to 0.61, identifying more true All-Stars. Despite this gain, the overall F- β score remains modest, indicating that the model (while better than baseline), still does

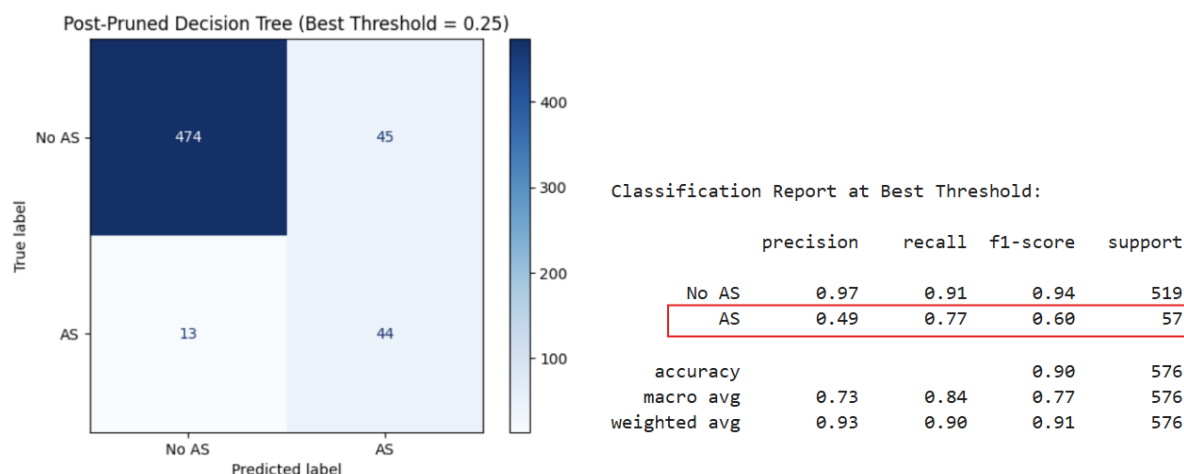
not deliver a strong enough balance between precision and recall to be considered a reliably effective classifier.



4.4.3. Post-Pruning

Still using the same modeling framework, we tuned maximum depth, minimum samples to split, minimum samples per leaf, splitting criterion, class weight, and the cost-complexity pruning parameter (`ccp_alpha`). GridSearchCV selected the optimal configuration as entropy criterion, no maximum depth, two samples per leaf, two samples required to split a node, no class weighting, and `ccp_alpha` = 0.0076. The final model predictions were generated using an optimized classification threshold of 0.25.

The model achieved an F_β score of 0.54, with a recall of 0.77 and a precision of 0.49. Compared to both the benchmark model and the pre-pruned tree, the post-pruned tree model significantly improved recall, capturing a larger share of true All-Stars. However, despite this improvement, the precision remains moderate, and the F_β score suggests that the model still does not achieve a strong balance between precision and recall. This suggests that although pruning reduces overfitting and improves generalization, a single decision tree remains limited in its ability to consistently classify All-Stars in this imbalanced setting.

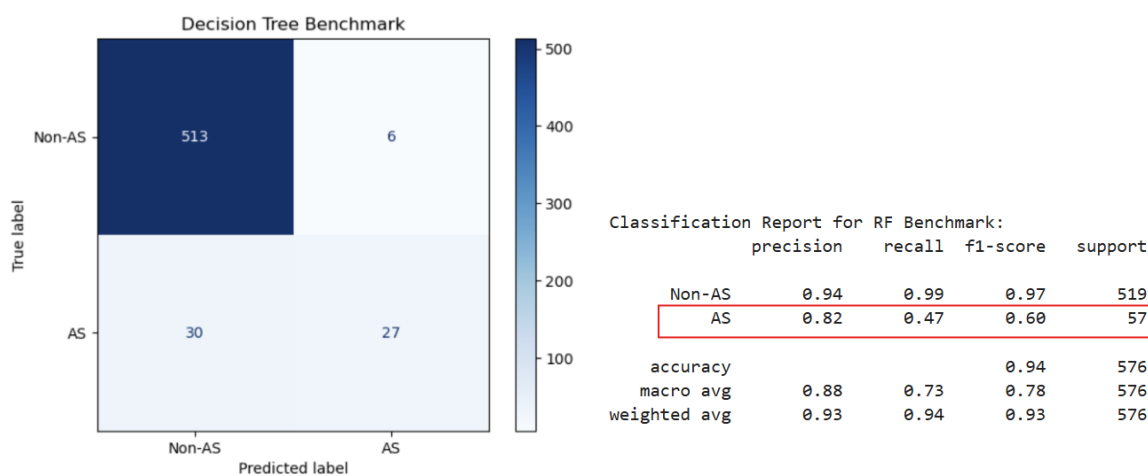


4.5. Random Forest

The limitations of both pre-pruned and post-pruned decision trees motivated the transition to random forest. This model reduces instability and overfitting by aggregating predictions from many trees built on different data subsets. This ensemble structure captures more complex patterns and interactions among player performance metrics, making it better suited for identifying All-Stars in an imbalanced classification setting.

4.5.1. Benchmark

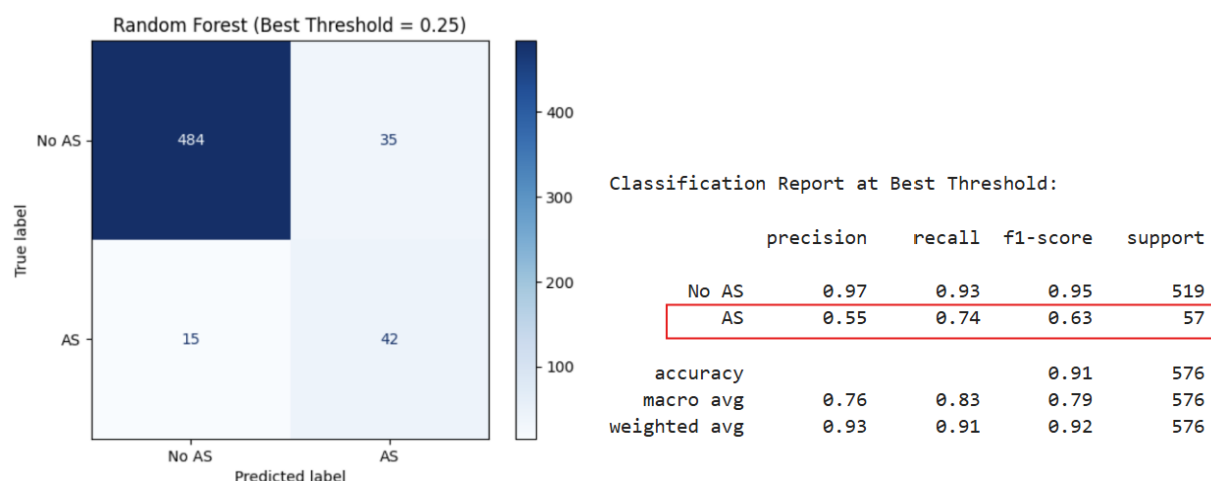
A baseline benchmark was established using a simple decision tree with default parameters and no pruning, reflecting the tree's raw ability to separate All-Star and non-All-Star players. The benchmark achieved a recall of 0.47 and a precision of 0.82, indicating that while it predicts All-Stars with high precision, it misses more than half of them.



4.5.2. Random Forest w/ Tuning

Using the modeling framework in Section 4.1, we tuned the number of trees, maximum depth, minimum samples to split, minimum samples per leaf, maximum features per split, and bootstrap strategy. GridSearchCV selected the optimal configuration of 100 estimators, no maximum depth, 10 samples to split, one sample per leaf, all features considered at each split, and bootstrap enabled.

The model achieved an F- β score of 0.55, with a recall of 0.74 and a precision of 0.55. Compared to the decision tree models, random forest offers a better balance between recall and precision, while still prioritizing recall. Its ensemble structure, which aggregates multiple decision trees, improves generalization and captures more complex patterns, making it a more reliable classifier for identifying All-Stars in this imbalanced setting.

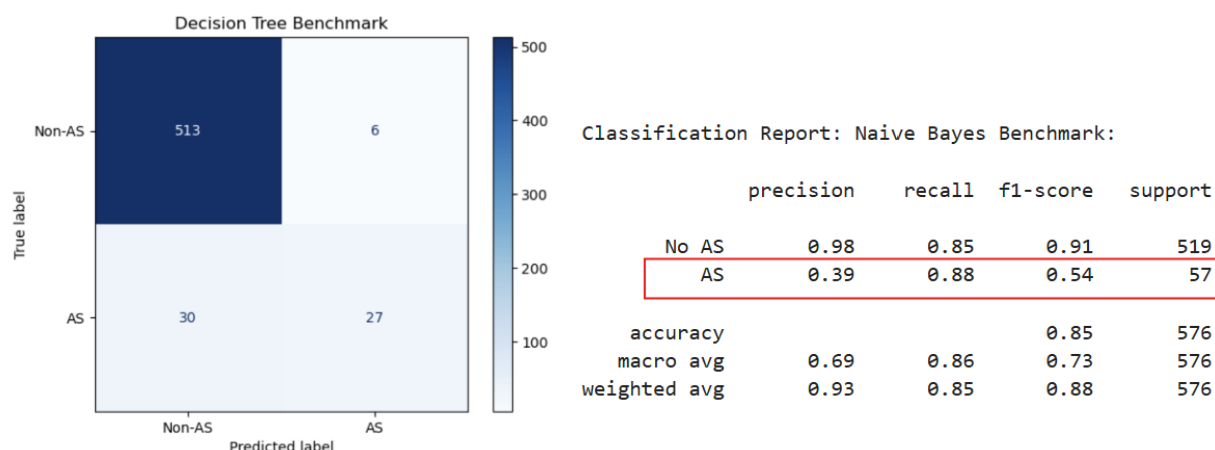


4.6. Naive Bayes

Naive bayes was included as a fast, interpretable probabilistic classifier and a useful baseline for comparison. It incorporates class priors, which helps in imbalanced settings. The Gaussian variant was selected because our dataset consists mainly of continuous performance metrics, making its normality assumption appropriate for modeling player statistics.

4.6.1. Benchmark

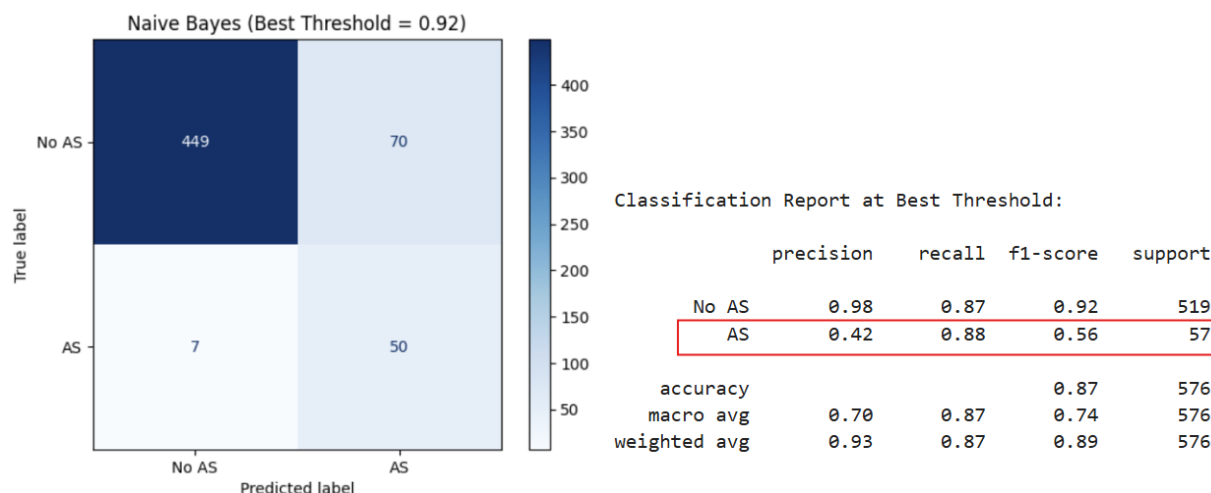
A baseline benchmark was established using simple gaussian naive bayes with default parameters and no tuning. The benchmark achieved a recall of 0.88 and a precision of 0.39, indicating that while it successfully identifies most true All-Stars, it does so with low precision, generating many false positives.



4.6.2. Gaussian Naive Bayes (Calculated Priors)

Keeping consistent with the modeling framework, we fit a gaussian naive bayes model with class priors computed from the training data to account for class imbalance. The tuned hyperparameter was the variance-smoothing parameter, which stabilizes probability estimates when feature variances are small. GridSearchCV selected a variance-smoothing value of $1e-8$, and final predictions were generated using an optimized threshold of 0.92.

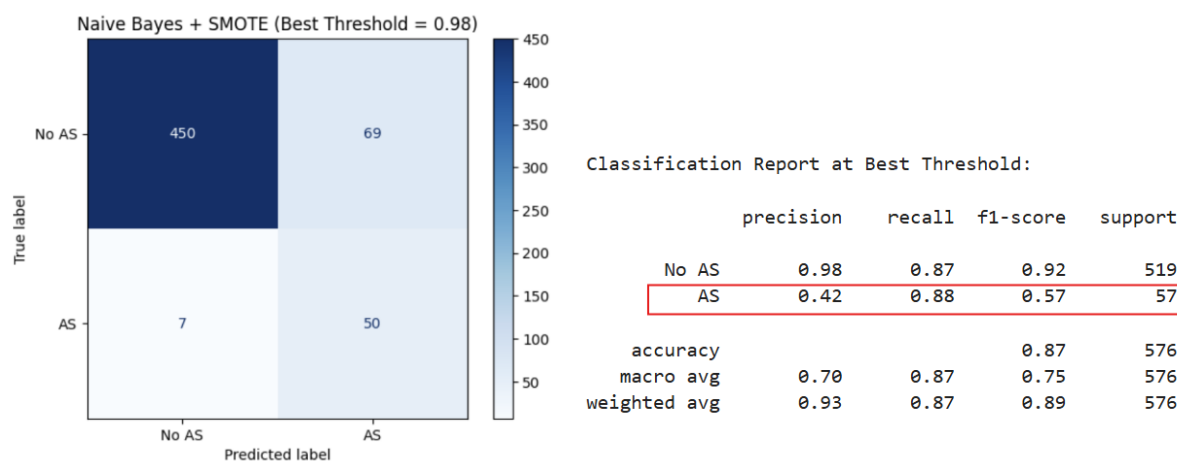
The model achieved an F- β score of 0.54, with a recall of 0.88 and a precision of 0.42. Among all models, naive bayes provides the highest recall, capturing most true All-Stars. However, this comes at the cost of lower precision, meaning the model produces more false positives. While the strong recall aligns with our objective of minimizing missed All-Stars, the moderate F- β score indicates that the overall balance between recall and precision is still limited. Thus, naive bayes is effective for identifying All-Stars but less reliable when precision must also be maintained.



4.6.3. Gaussian Naive Bayes with SMOTE

This model used the same gaussian naive bayes framework as before but incorporated SMOTE to oversample the minority All-Star class and more directly address class imbalance. The tuned hyperparameters included the variance-smoothing parameter and the SMOTE sampling ratio. GridSearchCV selected a variance-smoothing value of $1e-12$ and a SMOTE ratio of 0.5, and final predictions were generated using an optimized threshold of 0.98.

The model achieved an F- β score of 0.54, with a recall of 0.88 and a precision of 0.42. Compared to naive bayes with calculated priors, the recall remains equally high, indicating that the model continues to identify a large proportion of true All-Stars. However, precision does not improve, showing that oversampling the minority class increases sensitivity without reducing false positives. This suggests that while SMOTE helps reinforce recall, it does not create a more balanced trade-off between precision and recall for naive bayes, limiting the model's overall effectiveness for our objective.



5. Findings

After training our model, we gathered their respective scores to evaluate their performance.

Model	F- β	Recall	Precision
LogReg with Class Weight	0.59	0.61	0.74
LogReg with SMOTE + Opt. Ratio	0.65	0.81	0.52
DecTree w/ Pre-Pruning	0.49	0.61	0.62
DecTree w/ Post-Pruning	0.54	0.77	0.49
Random Forest	0.55	0.74	0.55
Gaussian Naive Bayes with Calculated Priors	0.54	0.88	0.42
Gaussian Naive Bayes with SMOTE + Opt. Ratio	0.54	0.88	0.42

It became clear that the model that (given our weighting of 60% recall and 40% precision) best minimized unidentified All-Stars while still avoiding excessive false designations, was logistic regression with SMOTE and optimal ratio, delivering the strongest predictive value for our goal of predicting All-Stars for organization.

The recall and precision values from the models perfectly reflect why we could not simply evaluate the models by their recall scores. Despite both naive bayes models having higher recall, they have precision values that are simply too low for us to accept. These models predicted far too many All-Stars in effort to boost correctly identified All-Stars, ultimately sacrificing precision too much for our comforts. This supports our decision to evaluate model performances in terms of F- β score, a properly weighted balance of the two metrics that we tailored to match the context of our objective.

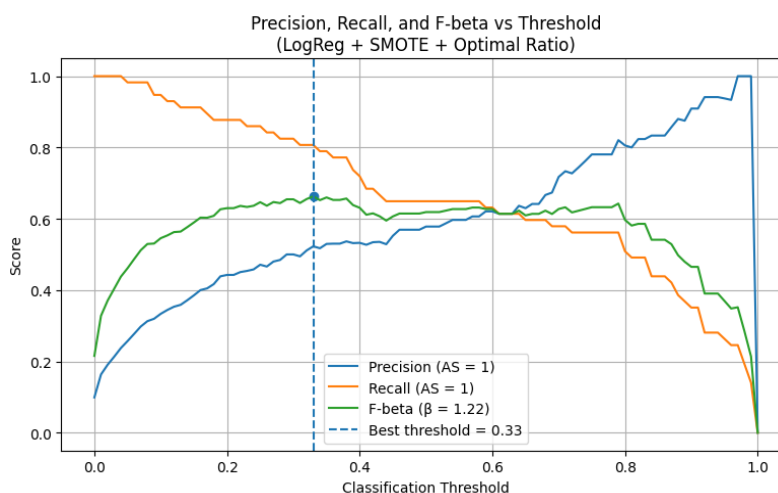
6. Data Mining and Key Takeaways

6.1. Data Mining

As part of the data mining process, several preprocessing and modeling strategies were applied to improve performance over the benchmark logistic regression model. We experimented with

multiple SMOTE sampling ratios, feature scaling, and multiple regularization types and strengths. GridSearchCV played a particularly important role by systematically identifying the optimal configuration, including a SMOTE ratio of 0.3, which shifted the class proportions from 10:90 in the benchmark model to 23:77 in the final model and significantly improved recall while maintaining reasonable precision. Threshold tuning further strengthened performance by selecting a decision boundary aligned with our objective of maximizing true All-Star identification.

The threshold-selection plot (precision, recall, and F- β vs. threshold) shows that as the threshold increases, precision rises while recall declines, and the F- β curve highlights the point at which the best balance is achieved. The chosen threshold of 0.33 corresponds to the peak F- β score, providing a clear visual justification for the final decision boundary. Together, these preprocessing steps, especially GridSearchCV optimization and threshold adjustment, proved highly effective in outperforming the benchmark and producing a more balanced and goal aligned classifier.



The following are the values of our logistic regression benchmark model as well as the model with optimal SMOTE ratio:

Metrics	Benchmark (for AS=1)	Regularization + Optimal SMOTE (for AS=1)
Recall	0.54	0.81
Precision	0.74	0.52

6.2. Key Takeaways

- 1.) It is important to assess the statistical stability of the target variable before modeling. In our case, the criteria and performance patterns of All-Star players varied noticeably between 2015 and 2024, but remained consistent from 2021 to 2024. This stability justified using the 2021–2024 window for reliable model training and evaluation.
- 2.) Class imbalance is common in real business datasets, and recognizing it early in the data mining process is crucial, as it directly affects how models should be evaluated. In our case, the 10% All-Star rate made accuracy misleading, guiding us to choose metrics that better capture minority-class performance.
- 3.) Adjusting the decision threshold significantly improved our model's usefulness. Because our objective places greater importance on identifying true All-Stars, tuning the threshold based on F_β allowed us to increase recall while keeping precision at an acceptable level, rather than relying on the default 0.5 cutoff that would have missed many All-Star players.

7. Predictions

Confident in logistic regression with SMOTE and Opt. ratio as our finalized model, we proceeded to apply it to the 2026 Steamer projections of player statistics to predict which players are likely to receive All-Star selections. We began by cleaning the projection data to include only the required feature columns. We then refit the model on the complete 2021-2024 dataset to provide it with all available observations before making its predictions. With the model fully trained and the new dataset prepared, we ran the model on the Steamer projections. Next, we used the model to produce each player's probability of becoming an All-Star, as well as a binary value depicting the model's ultimate prediction. We adjusted the threshold to match the model's previously optimized value (0.33), allowing the model to be more aggressive in identifying potential All-Stars than it would have with the default 0.5.

Final Model's 2026 MLB All-Star Predictions

Name	AS_prob	Name	AS_prob	Name	AS_prob
Aaron Judge	0.999819	Bryce Harper	0.765791	Alex Bregman	0.499088
Shohei Ohtani	0.992562	Gunnar Henderson	0.735672	Cody Bellinger	0.490618
Yordan Alvarez	0.985605	Fernando Tatis Jr.	0.730375	Vinnie Pasquantino	0.472697
Vladimir Guerrero Jr.	0.982964	Yandy Díaz	0.700022	Francisco Lindor	0.469113
José Ramírez	0.976648	Pete Alonso	0.683680	Elly De La Cruz	0.467058
Cal Raleigh	0.924767	Manny Machado	0.669665	Kyle Schwarber	0.453158
Juan Soto	0.923202	Salvador Perez	0.615471	Austin Riley	0.431575
Corey Seager	0.919219	Kyle Tucker	0.612005	Jackson Merrill	0.431026
Bobby Witt Jr.	0.889760	Matt Olson	0.599660	Jose Altuve	0.413034
Freddie Freeman	0.882781	William Contreras	0.581765	Riley Greene	0.411896
Julio Rodríguez	0.847030	Bo Bichette	0.578516	Jeremy Peña	0.396280
Ketel Marte	0.836592	Mookie Betts	0.547794	Luis Arraez	0.394306
Ronald Acuña Jr.	0.817517	Nick Kurtz	0.532805	Alejandro Kirk	0.389368
Junior Caminero	0.807347	Josh Naylor	0.512621	Corbin Carroll	0.347504
Rafael Devers	0.768751	Brent Rooker	0.500144	George Springer	0.341912

8. Business Implications

8.1. Budget for Expected Increases in Player Salary

Several of the players identified in our projections would be first-time All-Stars, including Nick Kurtz (SAC) and Vinnie Pasquantino (KC). Organizations with potential first-time selections should anticipate these players and their representatives to pursue salary increases in the following year. As a result, front offices should factor these likely payroll implications into their 2026 budgeting and roster decisions.

8.2. Marketing Campaigns

In addition, a number of projected All-Stars have recently joined new teams and will be entering their first season with those organizations in 2026, such as Rafael Devers (SFG). This group may expand, as five players on our list (Luis Arraez, Bo Bichette, Kyle Tucker, Kyle Schwarber, and Pete Alonso) remain free agents at the time of this analysis. Teams acquiring these players should incorporate them into upcoming marketing initiatives including promotional campaigns, merchandise, and giveaways to leverage both fan excitement and the players' projected on-field success.

9. Noteworthy Data Considerations

9.1. Data Assumptions

In order to derive our findings, we needed to make various assumptions regarding our data along the way:

1.) $F-\beta$

To best interpret the performances of our models, we had to create the $F-\beta$ metric as a customized balance between recall and precision for the minority class that fits our business context better than F1 score. Since our business analysis involves hypothetical scenarios, we are unable to precisely quantify the value of a true All-Star prediction, or the cost of a misidentification. We are only able to discern that the cost of failing to identify an All-Star (false negative) is greater than the cost of failing to identify a non-All-Star (false positives). Thus, we must assume that our 60:40 recall-precision ratio is appropriate and more aligned with our objective than the equal balance of importance calculated by F1.

2.) Consistent player performance

All of the data we provided our models with consisted of season total statistics for players. However, since the All-Star game is played halfway through each season, only player statistics from the first half of the season are taken into consideration for nominations. As we are not able to filter the statistics from this window with the dataset, we had to assume there were no drastic differences in players' 1st or 2nd statistics that would mislead the model. We acknowledge that being able to filter the data would have likely slightly improved our predictions.

3.) Players' performance from the training dataset was worthy of a selection

Major League Baseball requires each organization to have a representative in the All-Star game, potentially allowing for less-qualified players whose performance does not warrant a selection to receive the honor. These players' statistics would negatively impact our model's performance, but we proceeded under the assumption that negative outliers were not meaningful, or at least minimal. It is also worth noting that since fan voting is involved, we must assume that the fans voted for players responsibly and withheld biases toward their favorite player(s).

9.2. Data Limitations

As alluded to under the 'Consistent player performance' assumption, we were limited to data of season total statistics, and were unable to filter by the preferred sample lengths. Furthermore, our dataset is limited to offensive statistics, and fails to encapsulate a players' defensive performance. While defensive statistics would provide an additional metric to evaluate a player by, we also understand that offensive statistics are always considered to carry significantly more weight, especially amongst fans, thus not making this a significant concern for our model.