Baseball OBP Prediction for 2021 Season

Author: David Shableski

Email: dbshableski@gmail.com

Date: October 17, 2024

Project Overview

This project predicts the On-base Percentage (OBP) of MLB players for the 2021 season based on their historical OBP and Plate Appearances (PA) from 2016 to 2020. The program also factors in player age to adjust the predictions. The data is fetched from a CSV file and parsed using PapaParse, with predictions and actual OBP for 2021 displayed in the browser.

Files in the Project

- **index.html**: Contains the structure of the webpage, including placeholders for the predicted OBPs and total error.
- **obp.csv**: CSV file containing player data such as OBP, plate appearances from 2016 to 2020, their actual OBP for 2021, and birthdates.
- **script.js**: Main JavaScript file that processes the data, makes predictions, and displays results.

Features

1. **Weighted OBP Prediction**:
    - Predicts players' 2021 OBP by weighing recent seasons more heavily, with adjustments for missing or incomplete data.
    - More recent seasons and those with more plate appearances are given higher weights.
2. **Age Adjustment**:
    - If a player is older than 30, their predicted OBP is reduced by 5%.
3. **Error Calculation**:
    - The program calculates the total absolute error between the predicted OBP and the actual OBP for all players, providing a measure of prediction accuracy.
4. **Error Display**:
    - Players whose predicted OBP exactly matches the actual OBP are highlighted with a star *.

Dependencies

- **PapaParse**: Used to parse the CSV file containing player data.

Installation and Setup

1. Ensure you have the **obp.csv** file in the same directory as the HTML and JavaScript files.
2. Include the **PapaParse** library in your HTML file, either by downloading it or using a CDN:

<script src="https://cdnjs.cloudflare.com/ajax/libs/PapaParse/5.3.0/papaparse.min.js"></script>

3. Open **index.html** in a web browser.

Usage Instructions

1. The application automatically fetches and parses the **obp.csv** file containing player stats.
2. The JavaScript file processes the data to compute OBP predictions for each player based on their historical data.
3. Predicted OBPs, actual OBPs, and prediction errors are displayed in an ordered list on the webpage.
4. The total absolute error across all predictions is displayed at the bottom.

Methodology

To predict the 2021 on-base percentage (OBP) for MLB players, I used the provided data from the 2016 to 2020 seasons that included OBP and plate appearances for each year. The idea behind my approach was to place greater emphasis on more recent years and seasons where players had more plate appearances, as these tend to be more indicative of a player's current skill level. I assigned arbitrary weights to each season, increasing the weight for more recent years to reflect the recency of performance. Specifically, I multiplied the OBP for each season by its corresponding weight, which was determined by the player's plate appearances for that year, scaled so that more plate appearances carried more weight (e.g., 5x for 2020, 4x for 2019). This produced a weighted average OBP across the five years, which was then used as the predicted OBP for 2021.

Additionally, I included an age factor to account for player aging. If a player was over 30 years old, I reduced their predicted OBP by 5%, reasoning that older players generally experience a decline in performance. This reduction was arbitrary but based on general baseball knowledge. The model did not increase OBP for younger players, though that could be explored in future work. The predictions were then compared to the actual 2021 OBP, and the absolute error for each player was calculated and summed to give a total error for the model's performance.

Throughout the process I ensured that missing data (such as missing OBP or PA for certain years) did not skew the results by assigning a weight of zero to years without valid data. This allowed the model to still make a prediction even if some historical data was unavailable. Finally, the results were displayed in a user-friendly format on a web page, along with the total absolute error to assess the model's accuracy. Throughout the process I calculated the total error sum by taking the absolute value of my predicted OBP subtracted by the actual OBP and adding these numbers up for each player. I displayed that number at the top of the screen and monitored it during this process (wanting the number to be as close to 0 as possible).

Future Enhancements and Shortcomings

In this project, I calculated the weights for the on-base percentage (OBP) prediction model based on an arbitrary approach, assigning higher weights to more recent seasons and seasons with more plate appearances. While this approach is logical for prioritizing the most recent performances, it is somewhat simplistic and could very likely be improved with more sophisticated methods. For instance, the decision to reduce the predicted OBP by 5% for players over the age of 30 was an arbitrary rule based on a general understanding that players tend to decline as they get older. However, this could be refined using more data driven techniques. A machine learning model could, for example, learn the optimal reduction factor based on a broader dataset of player performances and ages, rather than relying on a fixed 5% reduction. This would allow the model to make more precise predictions for players across a variety of ages, considering individual variability in performance decline or improvement.

Additionally, the current model could be significantly enhanced by incorporating more advanced statistics such as exit velocity and launch angle. These metrics are strong predictors of hitting success, as they directly measure the quality of a player's contact with the ball. By including these variables in the model, the predictions could capture more nuance in a player's hitting ability beyond just past OBP and plate appearances. Furthermore, exit velocity and launch angle tend to show less year-to-year variability than traditional stats, which would make them reliable inputs for improving the accuracy of predictions.

Another shortcoming is that the model only uses data from five years (2016-2020). More data from additional seasons could improve the model's robustness, allowing for better trend analysis and the ability to detect longer-term patterns in player performance. Moreover, the model does not account for factors such as injuries, which could drastically impact a player’s performance but might not be reflected in their prior statistics. A more complex model could account for these kinds of anomalies, leading to better predictions overall.

Another thought for predicting the batting averages could be to include the batting average for the whole league over the season in each of the weights for each season.

Finally, machine learning algorithms like linear regression, decision trees, or even neural networks could be explored as more sophisticated methods for weighting previous performance and factoring in additional variables such as age, playing time, exit velocity, and launch angle. These models would be able to adjust weights dynamically based on historical data, optimizing the balance between the importance of recent performance and long term consistency. By training on a larger dataset, such models could significantly reduce prediction errors and provide more insightful evaluations of future player performance.