
Milestone 1: Data Description

Shawn Malik^{*1} Wenwan Xu^{*1} Huyen Huynh^{*1}

1. Data

Sources and Scope

All data, reports, and code will be at the link below:

https://github.com/shawnmalik1/DS3001_F25_Project

This project seeks to answer the question: Which performance metrics best predict NBA player salaries?

This project combines two season-level player files:

- **Player Performance** (`player_stats.csv`): per-player statistics of NBA players for season 2024–25 from Basketball-Reference, including counting metrics (e.g., G, MP), efficiency and usage rates (e.g., TS%, 3PAr, FTr, USG%), play-type rate stats (e.g., AST%, TRB%, STL%, BLK%, TOV%), and all-in metrics (e.g., PER, WS, WS/48, OBPM, DBPM, BPM, VORP), plus demographics/role (Age, Pos, Team).
- **Player Salary** (`player_salaries.csv`): per-player base-salary information from Basketball-Reference by future season (e.g., 2025–26, 2026–27, ...) and a total guaranteed money.

Both files are sourced from Basketball-Reference, which keeps naming and IDs consistent across statistics and contracts. The modeling target is *next season's* base salary (2025–26), and predictors are taken from the immediately preceding season's on-court performance. This aligns the temporal sequence of “performance → pay.”

Unit of Analysis and Coverage

Basketball-Reference reports one row per team stint and, for traded players, sometimes an aggregated multi-team line (e.g. “2TM / 3TM”). To keep things simple and consistent, we define one row per player-season by selecting the **single**

team stint with the most minutes (MP) for each player. This yields a representative stat line while avoiding double-counting and extra aggregation steps if a player played for a team and then got traded to another team in the same season. We will use the team that had the highest stint for that player.

After this filtering, we merge the stats and salary files and retain only players with both a prior-season stat line and a 2025–26 base salary. Not all players will meet both conditions, so some players will drop because they lack a next-season salary (unsigned, two-way/10-day conversions) or did not log meaningful minutes in the prior season.

Identifiers and Merge Strategy

When available, we use the Basketball-Reference player ID (e.g., lillada01) to create a common key `bbref_id` and perform an **inner join**. If an ID is missing for a small number of rows, we fall back to a name-based merge after light name cleaning (standardizing suffixes such as “Jr.”/“III”). The team variable that appears in both datasets (Team in `player_stats.csv` and `Tm` in `player_salaries.csv`) also helps us to double-check the merging process after we rename them to be the same. Using IDs where possible minimizes mismatches from suffixes, abbreviations, and nicknames.

Key Variables

Base salary (2025–26) (Y): parsed from currency strings into numeric values (USD). The distribution is strongly right-skewed: many players have salaries around the league minimum, and a long tail up to super max-level salaries (~ \$60M).

Candidate predictors (X)

- **Availability & playing time**: G (Games), MP (Minutes Played). Minutes both signal value and stabilize stats.
- **Efficiency & usage**: TS% (True Shooting Percentage), 3PAr (3-Point Attempt Rate), FTr (Free Throw Attempt Rate), USG% (Usage Percentage).
- **Role/rate indicators**: AST% (Assist Percentage), TRB% (Total Rebound Percentage, which could be

^{*}Equal contribution ¹University of Virginia, Charlottesville, Virginia, USA.

specified through ORB%, Offensive Rebound Percentage, and DRB%, Defensive Rebound Percentage), STL% (Steal Percentage), BLK% (Block Percentage), TOV% (Turnover Percentage).

- **Overall metrics:** PER (Player Efficiency Rating), WS (Win Shares), WS/48 (Win Shares Per 48 Minutes), OBPM (Offensive Box Plus/Minus), DBPM (Defensive Box Plus/Minus), BPM (Box Plus/Minus), VORP (Value over Replacement Player).
- **Demographics/role:** Age, Pos (Position). Team indicators are available if market/team effects are modeled.

Reading, Cleaning, and Preparation

1. **Read data from a single source.** Sourcing from Basketball-Reference (Basketball-Reference, 2025b;a), which is the single database for both of our data sets keeps identifiers consistent. Reading in as csv files allows for straightforward integration and ensures compatibility with subsequent data processing in Python.
2. **Consolidate multi-team seasons.** Retain the consolidated “xTM” line; otherwise, keep the highest-MP stint. This yields exactly one prior-season stat line per player and prevents double-counting.
3. **Parse money fields.** Strip “\$” and commas, coerce to numeric. Use 2025–26 base salary as the target; guaranteed totals span multiple years/options and are not modeled directly.
4. **Handle missing data.** Drop rows missing the target salary or essential stats (e.g., MP, Age). Leave rarely used or mostly-empty fields (e.g., awards) out of the model.
5. **Basic feature set.** Use straightforward predictors: MP, G, TS%, USG%, AST%, TRB%, STL%, BLK%, TOV%, and one or two overall metrics (e.g., BPM, WS). Include Age and Pos.

References

Basketball-Reference. 2024–25 nba player stats: Advanced, 2025a. URL https://www.basketball-reference.com/leagues/NBA_2025_advanced.html. Accessed: 2025-09-26.

Basketball-Reference. 2025–26 nba player contracts, 2025b. URL <https://www.basketball-reference.com/contracts/players.html>. Accessed: 2025-09-26.