Milestone 2:
Pre-Analysis Plan

Shawn Malik [*1] **Huyen Huynh** [*1] **Wenwan Xu** [*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 ($\sim$ \$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 [1]University of Virginia, Charlottesville, Virginia, USA.

Proceedings of the 42^{nd} International Conference on Machine Learning, Vancouver, Canada. PMLR 267, 2025.

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.

2. Methods and Results

Method Overview

Our goal is to predict each player's base salary for the 2025–26 NBA season using performance metrics from the 2024–25 season. We frame this as a supervised regression problem in which the target variable is salary (a numeric value in USD) and the predictors are player-level statistics such as minutes, efficiency, usage, and composite performance metrics.

We will begin with an exploratory data analysis (EDA) to examine relationships between the predictors and the target. Visualizations—including histograms, scatter plots, and box plots—will help us assess variable distributions, identify potential transformations, and detect outliers or influential observations. We will also conduct a correlation analysis to identify strong linear relationships and consider removing predictors with very weak correlations to salary, while remaining mindful of possible non-linear effects. In addition, we will explore potential interaction terms, as many basketball statistics are intrinsically correlated (e.g., WS and BPM). Investigating these interactions can reveal combined effects among predictors and inform the use of regularization techniques such as Ridge and Lasso regression to handle multicollinearity and reduce overfitting.

Our modeling strategy begins with simple, interpretable approaches before moving to more flexible models. This staged process allows us to build intuition about the data, understand which variables drive salary differences, and minimize the risk of overfitting.

Ultimately, this approach helps us balance predictive accuracy with interpretability. Beyond producing accurate predictions, we aim to understand how different aspects of player performance contribute to contract value. For example, players with strong efficiency metrics or high playmaking impact may command higher salaries even when their raw counting stats (e.g., points or rebounds) are modest. Insights from early models will guide feature selection, transformations, and model refinement in later stages.

Models

Our analysis aims to identify which player performance metrics most strongly predict NBA salaries while balancing model interpretability and predictive accuracy. To do so, we adopt a staged modeling approach that begins with transparent, interpretable regression methods and progresses toward more flexible machine-learning models capable of capturing nonlinear relationships.

1. **Linear Regression.** We begin with a standard linear regression model as a baseline to establish a direct and interpretable relationship between performance metrics and player salary. This model allows us to estimate marginal effects—how much salary is expected to change, on average, with a one-unit increase in metrics such as minutes played, win shares, or true shooting percentage, holding all other variables constant. Although simple, this baseline model provides a clear benchmark and helps us identify issues such as skewness, outliers, or multicollinearity before progressing to more complex modeling approaches.

2. **Ridge and Lasso Regression.** Basketball statistics often exhibit high collinearity—for example, overall impact metrics such as BPM, VORP, and WS are strongly correlated. To address this, we employ Ridge and Lasso regression, which introduce regularization penalties to shrink coefficients and

reduce variance. Ridge regression (L2 penalty) stabilizes estimates when predictors are correlated, while Lasso regression (L1 penalty) performs variable selection by shrinking some coefficients exactly to zero. By comparing the results of these two methods, we can both improve predictive performance and identify which features are consistently important across models.

3. **Random Forest.** To capture nonlinear relationships and higher-order interactions that linear models might overlook, we incorporate a Random Forest regressor. This ensemble approach aggregates multiple decision trees trained on bootstrapped samples of the data, using random feature selection at each split to enhance robustness. Random Forests handle outliers well and can model complex dependencies—such as the interaction between usage rate and efficiency—without explicit specification. Feature importance measures from this model will also be used to complement coefficient-based interpretations from the linear models.

Implementation and Preprocessing. All models are implemented in Python using the `scikit-learn` library. Numeric predictors are standardized to zero mean and unit variance to ensure comparability, and categorical variables (such as player position) are one-hot encoded. Salary, the dependent variable, is log-transformed to reduce right-skewness and stabilize variance, improving model fit. This unified modeling pipeline allows for reproducibility, consistent evaluation, and transparent comparison across modeling approaches.

Model Validation Plan

We evaluate model performance through a combination of quantitative metrics, visual diagnostics, and cross-validation to ensure that our findings are both accurate and generalizable.

Data Splitting and Cross-Validation. The dataset is partitioned into an 80% training set and a 20% held-out test set. Within the training data, we perform 5-fold cross-validation to tune hyperparameters—such as the regularization strength (α) in Ridge and Lasso regression—based on predictive performance. This approach balances bias and variance, providing a fair assessment of how well each model generalizes to unseen players.

Evaluation Metrics. Each model is assessed using three complementary performance metrics:

- **Root Mean Squared Error (RMSE)** – measures the magnitude of prediction errors, penalizing large deviations heavily.
- **Mean Absolute Error (MAE)** – captures the average absolute difference between predicted and actual salaries, providing an interpretable measure of prediction accuracy.
- R^2 **(Coefficient of Determination)** – quantifies the proportion of variance in salaries explained by the model.

Together, these metrics allow us to balance interpretability and predictive precision across linear and ensemble models.

Model Diagnostics. For linear models, we will perform residual diagnostics to verify key assumptions:

- **Linearity and Homoscedasticity:** Residuals versus fitted values plots will help assess whether variance is constant and relationships are approximately linear.
- **Normality:** Q–Q plots of residuals will indicate whether the error terms are normally distributed.
- **Independence:** An autocorrelation function (ACF) plot will ensure residuals are uncorrelated across observations.

If strong violations appear, transformations or alternative model structures (e.g., log-salary, interaction terms) will be considered.

Model Comparison and Interpretation. After validation, models will be compared on both test-set performance and interpretability. Visual tools such as predicted-versus-actual scatter plots will reveal systematic bias (e.g., underprediction of high earners). Coefficient magnitudes (from linear and regularized models) and feature importance scores (from Random Forest) will be jointly analyzed to identify which performance metrics consistently predict salary. We expect that indicators of playing time (`MP`), efficiency (`TS%`), and overall impact (`BPM`, `VORP`, `WS`) will emerge as the strongest predictors.

Finally, we will reflect on the uncertainty of our assessments—particularly for outlier cases such as rookies or players on unusual contract structures—and discuss how model limitations affect the confidence of our conclusions.

Next Steps

Once models are trained and validated, we will compare them and interpret which performance metrics best explain or predict player salary.

We expect that metrics capturing playing time (`MP`), overall impact (`WS`, `BPM`, `VORP`), and efficiency (`TS%`) will emerge as the strongest predictors.

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.