

# The Indianapolis Pacers 2025 Play-off Run: Discovering Trends to Predict Future Game Outcomes

1<sup>st</sup> Timothy Gilmore

CSCI-B 365

Luddy School of Informatics, Computing, and Engineering  
Bloomington IN, United States  
timgilm@iu.edu

2<sup>nd</sup> Matt Damsma

CSCI-B 365

Luddy School of Informatics, Computing, and Engineering  
Bloomington IN, United States  
mdamsma@iu.edu

3<sup>rd</sup> Fritz Kussow

CSCI-B 365

Luddy School of Informatics, Computing, and Engineering  
Bloomington IN, United States  
fdkussow@iu.edu

4<sup>th</sup> Alonso Carrasco

CSCI-B 365

Luddy School of Informatics, Computing, and Engineering  
Bloomington IN, United States  
alscarr@iu.edu

**Abstract**—This project organizes statistics and features from the 2024-2025 Indiana Pacers’ season and found what similarities exist between wins and losses for games played in that time frame. We pulled basic statistics from each game, including all 82 regular season games and all 23 playoff games. The statistics we looked at include who played in each game, how long they played in each game, and each player’s individual stats (points, rebounds, assists, turnovers, etc.) for each game they played in. Using this data, we organized players into certain ‘archetypes.’ For instance, if there’s a game where a player shoots 80% from three, we could say that the pacers had a sharpshooter for that game, which would likely improve their chance of winning. Additionally, we’ve taken into account the makeup of whatever team the Pacers played against. For instance, it’s possible that the Pacers had a much worse win percentage against teams with a sharpshooting big that could stretch the floor. By organizing them into certain player archetypes, we have created a reasonably accurate model to predict the outcomes of pacers games based on the top five player archetypes.

**Index Terms**—prediction, clustering, basketball, pattern recognition

## I. INTRODUCTION

Professional basketball has entered an era where data driven decision making shapes everything from game strategies to team roster construction. The NBA’s statistical tracking system captures hundreds of metrics per game. In return, this creates many datasets that offer great opportunities for analytical exploration. Predicting the outcome of NBA games is not an easy task. This is because every NBA game has great variability where a single shot or defensive stop could change the result of the game. For the Indiana Pacers, the leveraging of the vast amount of data collected by the NBA can prove to be very beneficial, providing a competitive advantage. The core problem this project aims to address is: Can we predict Indiana

Pacers game outcomes more accurately by finding patterns in similar historical games?

## II. METHODOLOGY

### A. Data Acquisition and Preprocessing

The dataset for this project is constructed from publicly available NBA game data, sourced from the nba\_api. The dataset will encompass all 105 games (82 regular season, 23 playoff) played by the Indiana Pacers during the 2024-2025 season.

The following data were collected for each of these games:

- Game-level data: Date, opponent, location (home/away), and final score.
- Player-level data: Detailed box scores for every player who participated in each game (for both the Pacers and their opponent). This includes minutes played (MP) and standard counting stats (points, rebounds, assists, steals, blocks, turnovers, fouls, FGA, FGM, 3PA, 3PM, FTA, FTM).

The raw data was preprocessed into a structured format where each row represents a single game ( $N = 105$ ). The primary target variable will be a binary outcome: Pacers\_Win (1 for a win, 0 for a loss). All other collected data was used to engineer predictive features. Preprocessing also involved handling missing values, such as removing players who were on the roster but did not play (DNP).

### B. Feature Engineering: Player Archetypes

A central component of our methodology is to move beyond individual player statistics and model the composition of the lineups. As proposed in the abstract, we operationalized this by creating “player archetypes.”

- Archetype Clustering: We applied a K-Means clustering algorithm to define the player archetypes. The clustering was performed on a normalized dataset of all players (from both the Pacers and their opponents) who played significant minutes during the season. The features used for clustering included per-36-minute standardized statistics and advanced metrics (e.g., True Shooting Percentage (TS%), Usage Rate (USG%), Assist Rate, Rebound Rate) to capture a player's style and on-court role.
- Archetype Labeling: After clustering, we manually inspected the statistical profile of each cluster to assign a descriptive label.
- Game-level Feature Vector: For each of the 105 games, we created a feature vector representing the lineup composition. This vector is a count of how many players belonging to each archetype were among the top five minute-getters for that game. This was done for both the Pacers and their opponent.

The final feature set for each game will include:

- Pacers\_Archetype\_1\_Count
- Pacers\_Archetype\_2\_Count
- ...
- Opponent\_Archetype\_1\_Count
- Opponent\_Archetype\_2\_Count
- ...

This approach directly models the interaction of lineup archetypes, addressing the hypothesis that the type of opponent impacts the Pacers' win probability.

### C. Model Development and Evaluation

Given the binary nature of our target variable (Pacers\_Win), this project formulates the problem as a binary classification task. Due to the small dataset size ( $N = 105$ ), we prioritized models that are less prone to overfitting and offer high interpretability.

1) *Selected Models*: We implemented and compared several classification algorithms:

- **Logistic Regression**: A baseline model that is highly interpretable, allowing us to quantify how the presence of specific Pacers or opponent archetypes directly impacts the log-odds of winning.
- **K-Nearest Neighbors (K-NN)**: This model works by finding the ' $k$ ' most similar games from the past (the "nearest neighbors") and then predicting the outcome based on how those similar games turned out.
- **Naive Bayes**: The Naive Bayes probabilistic classifier calculates the likelihood of a win or loss given the lineup archetypes.

2) *Model Validation*: We employed a 80/20 train-test split where the games were selected at random so that 80% were used to train and 20% were used to test the model's ability to predict the outcome of the game.

3) *Evaluation Metrics*: Model performance is assessed using a suite of standard classification metrics:

- **Accuracy**: The overall percentage of correctly predicted game outcomes.
- **Precision, Recall, and F1-Score**: These metrics are especially crucial if the win-loss record is imbalanced (e.g., if the Pacers had a 70-win season, a model that always predicts "Win" would have high accuracy but zero utility).

The deliverable is a model that is reasonably accurate and may provide actionable insights into which lineup compositions and opponent archetypes led to wins and losses for the 2024-2025 Pacers. Although the model is not highly accurate—as you will see in the results—the model shows does show strong potential for meaningfully predicting the outcomes based on the player archetypes if it is supplied with more game data.

## III. RESULTS

### A. Player Archetype Analysis

We applied K-Means clustering to identify different player archetypes based on playing style and player statistics. The analysis included all players, Pacers and opponents, who logged more than 10 minutes during the 2024-2025 season in regular and postseason games. The stats of the five player archetypes identified through clustering are shown in figure 1.

- **Archetype 0**: These players contribute minimally across the statistical categories not assuming a primary offensive or defensive role. These players have a low average scoring output and do not have a large number of field goals attempted per 36 minutes. They provide moderate rebounding and assist support, while having a low turnover rate. These players typically function as role players within a team's structure, providing support when needed.
- **Archetype 1**: These players are the disruptive defenders. With an average of 4 steals per 36 minutes these players thrive on forcing turnovers and turning defense into offense. The rest of their 36-minute stats are relatively high, making them a versatile guards or wings who are able to contribute to the team with hustle and defensive intensity.
- **Archetype 2**: These players are characterized by elite defensive metrics in the paint. Averaging 10 rebounds are nearly 3 blocks per 36 minutes, these players anchor a team defense and control the paint. These players have a limited offensive role, most likely finishing plays rather than initiating plays, through assists. These players align with the traditional defensive big men, whose primary value lies in rim protection and being able to secure possessions through rebounding.
- **Archetype 3**: These players combine high scoring with elite play making ability. Averaging nearly 20 points and over 8 assists per 36 minutes, these players function as a team's offensive engine. The elevated turnover rate for this archetype reflects that these players have a central role in ball handling and decision making for a team's

| Cluster Centers (Average Per 36 Stats): |              |                     |               |              |
|---|--------------|---------------------|---------------|--------------|
| Cluster                                 | points_Per36 | reboundsTotal_Per36 | assists_Per36 | steals_Per36 |
| 0                                       | 10.460058    | 5.381284            | 2.901230      | 0.738448     |
| 1                                       | 14.868522    | 6.101273            | 4.001247      | 4.029634     |
| 2                                       | 14.920619    | 10.584298           | 2.351456      | 0.854133     |
| 3                                       | 19.913025    | 4.906411            | 8.554681      | 1.134894     |
| 4                                       | 25.640368    | 6.843131            | 3.060059      | 0.879420     |

| Cluster | blocks_Per36 | turnovers_Per36 | threePointersMade_Per36 |
|---------|--------------|-----------------|-------------------------|
| 0       | 0.318550     | 1.513121        | 1.067895                |
| 1       | 0.537819     | 1.629817        | 1.380758                |
| 2       | 2.780147     | 1.966000        | 1.100271                |
| 3       | 0.457295     | 3.641459        | 1.626336                |
| 4       | 0.454469     | 1.565769        | 3.849917                |

| Cluster | fieldGoalsAttempted_Per36 |
|---------|---------------------------|
| 0       | 9.366166                  |
| 1       | 12.510348                 |
| 2       | 11.534521                 |
| 3       | 16.021801                 |
| 4       | 17.087690                 |

Fig. 1. Player Archetype per 36 Minutes Stats

offense. These players best reflect a star point guard or primary ball handler that is essential to a team's success.

- **Archetype 4:** These players are scoring machines. Averaging 25 points and nearly 4 three pointers made per 36 minutes, these players are the primary scorers for a team's offense. This is also shown in the fact that these players have the highest field goal attempts per 36 minutes. These players contribute fine in rebounding and assists but have a primary function of offensive scoring. These players align with scoring specialists whose value to a team comes from their ability to generate points at high efficiency and volume.

## B. Model Evaluation

Three classification algorithms, K-Nearest Neighbors (KNN), Naïve Bayes, and Logistic Regression, were evaluated on the dataset. Their performance is shown in Figure 2

- **KNN:** achieved the highest overall accuracy 0.62 but failed to correctly classify any losses. KNN had a strong performance for classifying wins with precision being 0.68, recall being 0.87, and F1-score being 0.76.
- **Naïve Bayes:** obtained an accuracy of 0.57. The Naïve Bayes algorithm did demonstrate a slightly better time at classifying losses with precision being 0.20, recall being 0.17, and F1-score being 0.18. The Naïve Bayes algorithm like the KNN algorithm stayed strong at classifying wins with precision being 0.69, recall being 0.73 and F1-score being 0.71.
- **Logistic Regression:** had the lowest accuracy of 0.52 and also did a slightly better time at classifying losses compared to KNN with precision being 0.17, recall being 0.17, and F1-score being 0.17. Logistic Regression like the other algorithms performed well at classifying wins with precision being 0.67, recall being 0.67, and F1-score being 0.67.

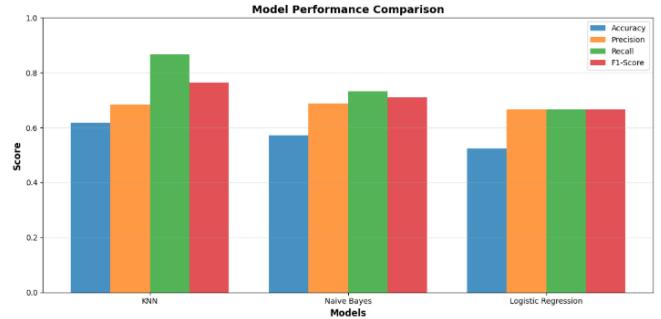


Fig. 2. Model Performance Comparison

|                  | games_played | wins | losses | win_pct   | disparity_vs_baseline |
|------------------|--------------|------|--------|-----------|-----------------------|
| Pacers_Archetype |              |      |        |           |                       |
| Pacers_Arch_4    | 84           | 54   | 30     | 0.642857  | 0.023810              |
| Pacers_Arch_1    | 103          | 65   | 38     | 0.631068  | 0.012020              |
| Pacers_Arch_0    | 99           | 62   | 37     | 0.626263  | 0.007215              |
| Pacers_Arch_2    | 103          | 63   | 40     | 0.6111650 | -0.007397             |
| Pacers_Arch_3    | 69           | 41   | 28     | 0.594203  | -0.024845             |

Fig. 3. Pacers Archetypes's Performance

## IV. DISCUSSION

### A. Pacers Archetypes

By looking at the win percentages for the Pacers when they do or do not have each archetype, we can find what archetype seems to tip the scale in the Pacer's favor the most. From the data we collected, we can see that Archetype 4 (25 pts, 4 3PM) seems to boost the Pacers' win percentage the most, boasting a win rate about 2.4% higher than the baseline. Of course, this percentage isn't very large, so we don't think we can assuredly say that having a player of the fourth archetype would automatically make the team better. A reason I believe this is because the Pacers' best player (in my opinion), Tyrese Haliburton, fits best into player Archetype 3 (20+ pts, 8+ ast), which shows the worst win rate when on the Pacers' side. However, a player like Haliburton could also fit into Archetype 4, displaying again how the nature of basketball makes games hard or even impossible to predict.

### B. Opponent Archetypes

Conversely, we can look at what archetypes, when playing for the Pacers' opponent, tend to cause the Pacers the most trouble. Using our data (from nba\_api, the 20245-25 Pacers' 109 regular season and post season games), we can see that the Pacers struggle the most when an opposing player fits Archetype 3 (20 pts, 8 ast). The win disparity of about negative 5.5% indicates that when the Pacers face a player like this, they'll surely have a worse chance of winning. One thing we noticed is that Archetype 3 actually appears for the Pacers the least out of the five archetypes, so it's possible that they struggle against a player of such kind since they wouldn't

```

(
    games_played  wins  losses  win_pct
Opponent_Archetype
Opp_Arch_2          104   65     39  0.625000
Opp_Arch_1          99    61     38  0.616162
Opp_Arch_0          85    53     32  0.623529
Opp_Arch_4          86    52     34  0.604651
Opp_Arch_3          55    31     24  0.563636

    disparity_vs_baseline
Opponent_Archetype
Opp_Arch_2           0.005952
Opp_Arch_1          -0.002886
Opp_Arch_0           0.004482
Opp_Arch_4          -0.014396
Opp_Arch_3          -0.055411 ,

```

Fig. 4. Opponents Archetypes's Performance

be practicing as much against such a player. It's also worth mentioning that a statline of 20-plus points and 8-plus assists is quite a good statline, so it makes sense those kinds of players could skew the data.

#### REFERENCES

In terms of resources used, we based all of our code off of what we have done in class and the documentation surrounding nba\_api.

#### REFERENCES

- [1] Patel, Swar. nba\_api, [https://github.com/swar/nba\\_api](https://github.com/swar/nba_api)