Comparing Models for Forecasting NBA Game Result: KNN and Decision Tree

Yiming Zhang, Mingqi Zhang, Ruohan You

***Abstract*—This project focuses on developing a predictive model to forecast NBA game outcomes using machine learning. By leveraging data of seasons from 2019 to 2023, two models—K-Nearest Neighbors (KNN) and Decision Tree—were employed to classify game results based on metrics like win percentage, offensive/defensive ratings, and player statistics. Both models are trained and tested on normalized data of recent matchups between each pair of teams. This project explores the methods used, the results, and discusses further findings related to the models used and the results.**

Keywords—Data processing, KNN, Decision Tree, Predictions, NBA, Basketball, Classification

I. Introduction

The National Basketball Association (NBA) is one of the most popular sports leagues globally, attracting millions of fans. In modern days, people have shown great interest in knowing the game results before the game finishes. In this project, we focus on finding a better model that predicts better game results between K-Nearest Neighbors (KNN) and Decision Tree. By analyzing historical game statistics, each model can train and test the dataset to predict what happens the next time these two teams have a matchup.

This project uses the data of previous NBA plays from the 2019-2023 seasons. These season datasets could represent most of the recent plays that have a crucial point for understanding further matchups in the 2024-2025 seasons. The goal is to find the most suitable model for analyzing and predicting future games by comparing KNN and Decision tree.

II. Methods

Both models, KNN and Decision Tree, were implemented using the scikit-learn library:

A. K-nearest neighbors

The original team statistics dataset [1] includes seasons from 2000 to 2023. To reduce the chance of getting overfitting, we filter out the 2019-2023 season stats and calculate some team statistics that are important when evaluating a team's record, including average points per game, win percentage, average rebounds, average assists, etc. And then we use the dataset that includes previous matchups and records [2].

We then clean the dataset for testing and training by dropping the columns with more than 30% missing values, and for the rest of the missing values, we fill in the column median values. We use rolling averages to calculate 5 games for each team's stats: points, rebounds, assists, and wins to sort the data by team and season. We also calculate the pace, offensive and defensive rating, and net rating based on shooting efficiency and points scored. Comparing the differences between two teams is one of the most important steps in KNN modeling; if there are statistics for both teams, we subtract two teams' features to create new feature differences that represent the difference between two teams. This allows the model to make final decisions on the game result in a more accurate and efficient way. And then we use binary labels for team matchups based on the differences between win percentages of two teams that are calculated in the previous step to create a new dataset where label 1 is team 1 win, and label 0 is team 2 win. To specify the seasons, we convert the game_date to datetime to filter out specific matchup times, from 2019-02-14 00:00:00 to 2023-02-19 00:00:00.

To perform better accuracy, we split the data into the training set and testing set. We use sklearn.discriminant_analysis to import StandardScaler to scale the data to normalize values for KNN. For the KNN model, we set n = 5 to train the model to study 5 nearest neighbors and then eventually predict the outcomes for the test set.

B. Decision Tree

We use the same datasets as we used for the KNN model. We clean the dataset [1] to make sure it is pulling the data from the 2019-2023 season, and we drop columns with more than 30% missing values, and for the rest of the columns with missing values, we fill them out with their column median. We use the previous matchup record dataset [2], limiting the time from 2019-02-14 00:00:00 to 2023-02-19 00:00:00 to calculate more important statistics to know each team better.

XXX-X-XXXX-XXXX-X/XX/$XX.00 ©20XX IEEE

We use rolling averages to calculate 5 games for each team's stats: points, rebounds, assists, and wins to sort the data by team and season. If there are not enough historical plays, the average is computed by available games. We then create the pairs of teams and calculate the differences in the win percentage and other matchup features. We also set up the binary result for this model for the final prediction that 1 is home wins (team 1 wins) and 0 is away wins (team 2 wins). We split the whole dataset into training and testing subsets for better performance. The scale feature is applied for the model to ensure that all features have equivalent contributions to the model, preventing any feature possibly dominating the whole training set.

For the Decision Tree model, we used sklearn.tree: max_depth = 5 to limit the tree depth to reduce overfitting, and min_samples_split = 10 sets The minimum splits to 10 samples; min_samples_leaf = 5 sets the minimum leaf to 5 samples.

III. Results

For comparison purposes, each model is evaluated using accuracy, precision, recall, f1-score, and ROC-AUC. Since each model pulls recent matchups randomly, the classification reports could show slightly different results. The K-nearest neighbors' model has a 56% overall accuracy (Table 1), and the Decision Tree model has a 63% overall accuracy (Table 2). This indicates that the decision tree model is better at learning from the patterns in historical matchups and predicting future results more effectively. The decision tree shows its ability to handle imbalanced data with an 80% recall (class 1), highlighting its ability to identify the true positives, which is particularly helpful when 2 teams present an imbalanced matchup history. The K-nearest neighbors' model has a 63% recall (class 1), emphasizing that the KNN model has certain ability to identify some imbalanced data.

Table 1: KNN classification report

Accuracy: 0.56

Classification Report:

	precision	recall	f1-score	support
0	0.48	0.46	0.47	436
1	0.61	0.63	0.62	592
accuracy			0.56	1028
macro avg	0.54	0.54	0.54	1028
weighted avg	0.56	0.56	0.56	1028

Predicted winner (1=home win): 1

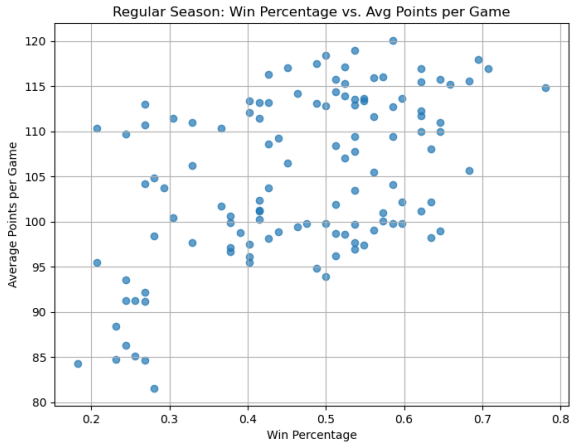
Table 2: Decision Tree classification

Accuracy: 0.63

Classification Report:

	precision	recall	f1-score	support
0	0.59	0.40	0.48	436
1	0.64	0.80	0.71	592
accuracy			0.63	1028
macro avg	0.62	0.60	0.59	1028
weighted avg	0.62	0.63	0.61	1028

Graph 1: Regular season: Win percentage vs. average points per game



Now we can make predictions of a specific matchup using KNN and Decision Tree.

In our example, we selected Team 1 = "Boston Celtics" and Team 2 = "Denver Nuggets" as the matchup for prediction. Both the KNN and Decision Tree models predicted that the Boston Celtics would win the matchup.

This result highlights both models' ability in analyzing past game events and determining the likely wins. The prediction is made based on the patterns learned by models using team statistics, especially the win percentage. In most cases, win

percentage and average points per game are in a positive correlation (graph 1); the higher the average points per game, the more possible it is for the team to win a matchup. For instance, teams like the Boston Celtics — known for their high-scoring capabilities—often benefit from this correlation, as their ability to score more points increases the probability of winning.

IV. Discussion

Our KNN model has an accuracy of 56%, which is almost 10% lower than Bryant University's study with an accuracy of 65.1% [3] due to different time ranges, which limit their data to 3 seasons, from 10/19/2018 to 3/20/21. The difference could happen due to the changes within teams. In the future we could adjust the dataset we use to predict any specific game results for better performance.

Our decision tree achieves an accuracy of 63% and a recall rate of 80% when predicting home wins. We see each team calculating the statistics for a team, ignoring any change of players in the season 2019 to 2023. Washington University's study focuses more on players' performances [4]. To determine a synergy for each player with each cluster, they examine how playing a game with specific clusters impacts each player's performance.

We found another article comparing the performances between models, and they also found that decision trees perform better [5], and extra trees perform the best; however, they also applied the model with players instead of teams. They pick these top-ranked players out based on their usual performance during regular seasons, which makes their model more selective in a way. In the future, we could develop our models based on each team's usual starting lineups to have more control with player statistics other than overall team statistics.

V. Author contribution statement

Yiming: Final report and data collection.

Mingqi: Data preprocessing and decision tree model.

Ruohan: KNN model assisted by ChatGPT for data exploration.

References

[1] H, Michael. "NBA Team Stats." *Kaggle*, 16 Apr. 2024, www.kaggle.com/datasets/mharvnek/nba-team-stats-00-to-18.

[2] Walsh, Wyatt. "NBA Database." *Kaggle*, 6 July 2023, www.kaggle.com/datasets/wyattowalsh/basketball.

[3] *Predicting the Outcome of NBA Games*, digitalcommons.bryant.edu/cgi/viewcontent.cgi?article=1000&context=honors_data_science. Accessed 17 Dec. 2024.

[4] *NBA Player Performance Prediction Based on ... - Washington*, courses.cs.washington.edu/courses/cse547/23wi/old_projects/23wi/NBA_Performance.pdf. Accessed 17 Dec. 2024.

[5] *Evaluating the Effectiveness of Machine Learning Models for Performance Forecasting in Basketball: A Comparative Study*, www.researchgate.net/publication/379242081_Evaluating_the_effectiveness_of_machine_learning_models_for_performance_forecasting_in_basketball_a_comparative_study. Accessed 17 Dec. 2024.