Machine Learning & Data Mining
(SOFE 4620U)

Project: NBA Lineup Prediction

Professor: Dr. Masoud Makrehchi

Name	ID	Initial
John Howe	100785128	J.H
Bilal Khalil	100824635	B.K
Ashwin Prem	100805031	A.P

Project Group 12

Due Date: Wednesday, Mar 19, 2025

GitHub Link: https://github.com/johnh-otu/NBA-machine-learning

Methodology

We made a machine learning model that predicts the fifth player for a home team in an NBA game. The methodology was to use key steps like, data preprocessing, feature selection, model training with Sci Kit Learn (SK Learn) library , and evaluation. We used datasets containing the NBA game lineups from 2007 to 2015, to make sure that only the features we specified in the metadata were used. The training data consisted of the historical matchups, while the test data was structured to have four home team players and five away team players, making it so that the model would have to predict the missing home player.

Feature Engineering

Feature engineering was a very important part of this project. We were asked to only use specific features from the dataset. So due to these restrictions we were required to extract and process key game-related attributes like team compositions, player stats, and the performance outcomes. We selected the features based on their relevance to the team's success and the individual player contributions. We also made sure that categorical variables were encoded using label encoding. This encoding method is more efficient than one-hot encoding given the size of our data. In future iterations, we may choose to use one-hot encoding for more accurate results. Feature engineering like this was important in improving the models ability to generalize and make accurate predictions.

Selected Model

We chose to use SciKit learn as our model framework because according to our research we found that they are the best suited to handle structured data, and require minimal parameter tuning, we also liked that it provides interpretability in the predictions. Decision forests can handle missing values, non-linearity, and feature important evaluations effectively. We specifically used Random Forests, since they offer robust performance without needing super extensive deep learning techniques. Choosing to use this model allowed us to benefit from the Sci Kit Learn library and use a powerful decision tree learning model.

Training Process

We used the training dataset for training and the test dataset for evaluation to make sure that the model was learning from the historical matches while being tested on the unseen data. We used a supervised learning approach, training 5 models to each predict the “i”th player based on the past matches. For example, model 1 predicts a missing player 1, model 2 predicts player 2, etc. Each model was trained on the entire training dataset to allow our model to properly predict the missing player, regardless of which player was missing. The training phase required us to optimize some hyperparameters, tune the decision trees and validate the results to prevent possible overfitting. We had to experiment with different tree depths and sizes to balance performance and generalization.

Pre and Post Processing

Pre-processing involved how we handled missing data, removed irrelevant columns and made sure data was consistent across the years. Player statistics have to be normalized where necessary and categorical data has to be encoded for the machine learning compatibility.

For pre-processing the given training data, we combined all of the given CSV files into one united dataframe, and stored it as *games_df*. The next task is to build a function that would evaluate each player and assign them a score based on specific performance metrics (3-pointers, 2-pointers, free throws, rebounds, assists, blocks, and turnovers.) The calculation is a bit different depending on whether the player is on the home team or the away team. Then, with the help of a dictionary, the team, score, and total games are tracked and stored for each player. This is done dynamically, and it gets updated each time a player is found in the dataset, and the games count is incremented. The final dictionary is then converted into a structured data frame format, which is the final CSV file that is now ready for post-processing.

To select the best match we used a combination of both the pre-processed player scores and the prediction of the model. The evaluation score was calculated as 0.7 * prediction score + 0.3 * preprocessed player score. In future iterations, we may choose to test different combinations of coefficients to decide on the best function to use.

Post-processing consisted of us analyzing the models predictions to make sure they align with historical patterns. This was done using Sci Kit Learn's built in evaluation functions.

Evaluation

To make sure that the model is effective we evaluated for accuracy and mean square error. The primary metric was how much the predicted player contributed to the teams success historically which we measured by the impact of their performance. We also did qualitative analysis by reviewing predictions to make sure that we had logical consistency. We also compared our models predictions against our baseline.

Performance

The performance of our model was measured at having 30.9% accuracy and a mean squared error of 4260.106.