Faculty of Engineering and Applied Science

SOFE 4620U: Machine Learning & Data Mining

Project- NBA Lineup Prediction

March 17th, 2025

Group ML5:

Hamzi Farhat	100831450
Tahmid Chowdhury	100822671
Jason Stuckless	100248154

GitHub Repository Link:

https://github.com/JasonStuckless/NBA_Prediction

Introduction

The NBA lineup prediction model is designed to identify missing players in historical game matchups using machine learning techniques. In order to generate precise predictions, the model uses XGBoost to evaluate a number of variables, including player win rates, team strengths, and synergy scores. There are two modes of operation for the model: test data mode, which involves testing on a curated dataset after the model is trained on all season data from 2007 to 2015, and training/testing year pairs mode, which involves training on one season and testing on the next. (i.e. training on 2007 and testing on 2008, and so on.) The model's implementation, prediction generating approach, data preprocessing, feature selection and engineering, and overall performance evaluation are all included in the report that follows.

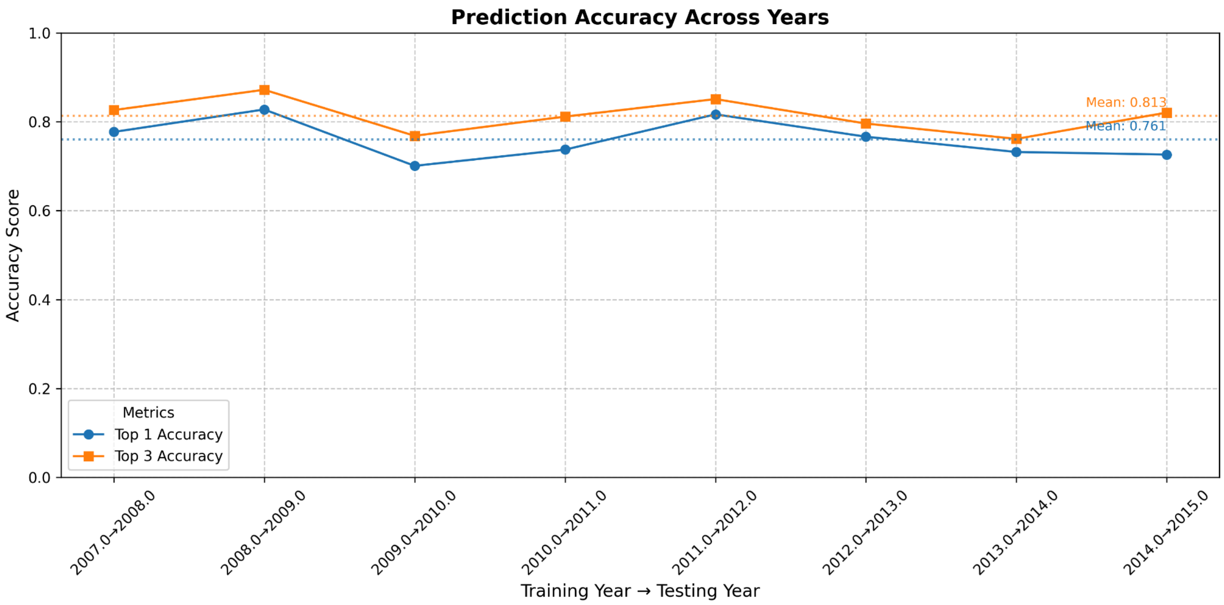
Model Implementation

The model is built using XGBoost, a powerful decision-tree-based algorithm designed for structured data. It analyzes past NBA matchups to predict which player is missing from a lineup by identifying patterns in historical performance. The process begins by loading training and test datasets, converting categorical information like player and team names into numerical values, and extracting key statistical features. To enhance efficiency, the model incorporates GPU acceleration through CUDA and fine-tunes hyperparameters for better performance. With two operational modes, the model can either test its predictions across different seasons or evaluate its accuracy on a specific test dataset, providing a flexible way to measure its accuracy, precision, recall and f1-score.

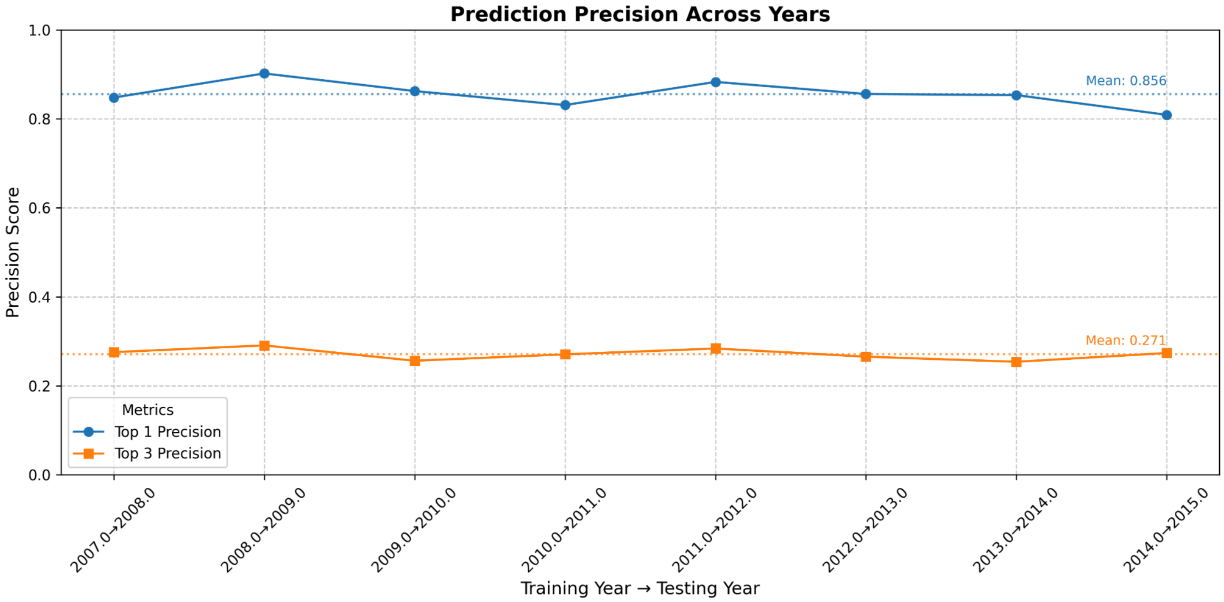
Prediction Outputs & Evaluation

Training/Testing Year Pair Mode:

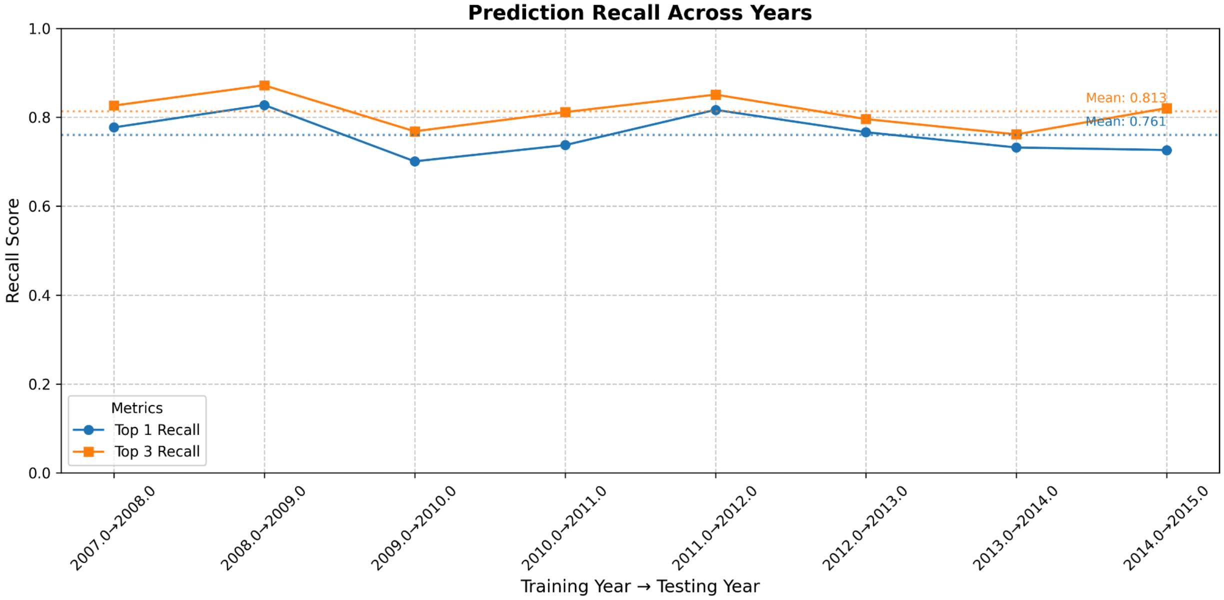
This was the original method developed for the project before we received the test data. For this mode, the model generates two types of predictions: Top 1 Prediction, which identifies the single most likely missing player, and Top 3 Predictions, which provides a shortlist of the three most probable candidates. When tested in the training/testing year pairs mode, the model demonstrates strong performance, achieving an average Top 1 accuracy of 76.05% and a Top 3 accuracy of 81.35%, showing that it effectively recognizes the correct player in most cases. The Top 1 F1-score of 74.17% highlights the model's ability to balance precision and recall. However, the Top 3 F1-score is lower at 40.67%, indicating that while the model can identify multiple potential players, its confidence in ranking them correctly could be improved. These results suggest that while the Top 1 predictions are highly reliable, additional refinements could enhance the precision of Top 3 predictions to make them more accurate and meaningful. The following graphs are a pair-by-pair accounting of the results of this mode for accuracy, precision, recall, f1-score, and then two graphs that show all the metrics combined for top 1 and top 3 predictions:



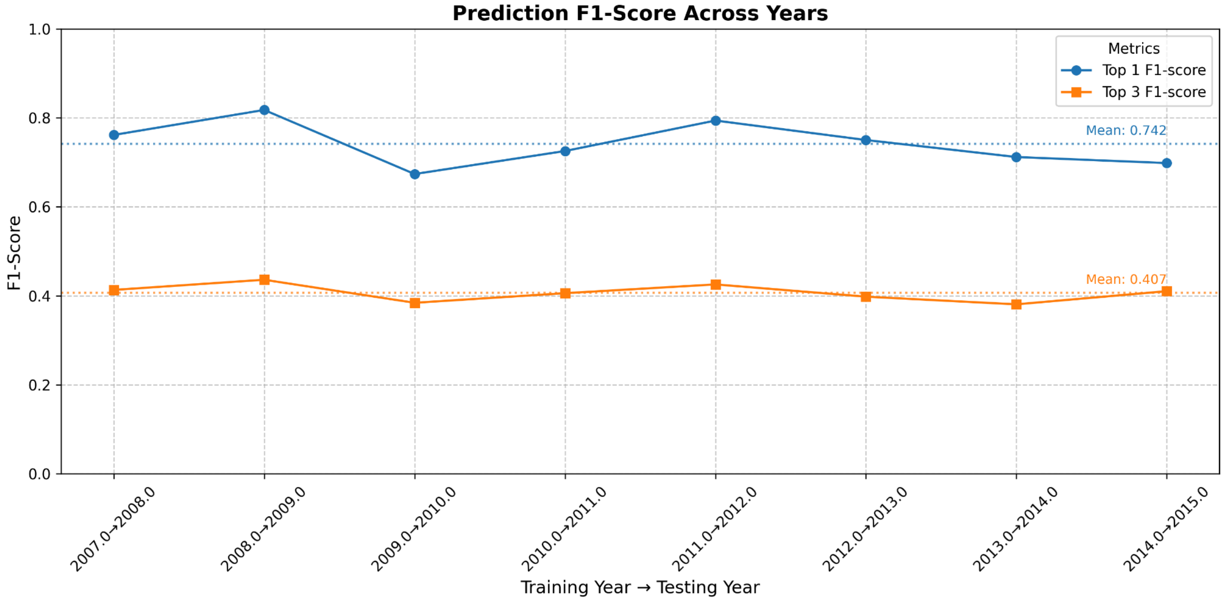
This graph shows Top 1 and Top 3 accuracy for different training/testing year pairs. The Top 3 accuracy is consistently higher than Top 1 accuracy, indicating that while the model may not always get the exact missing player right, the correct player is frequently included in the top three predictions. The dotted lines represent the mean accuracy values, with Top 1 accuracy averaging 76.1% and Top 3 accuracy at 81.4%.



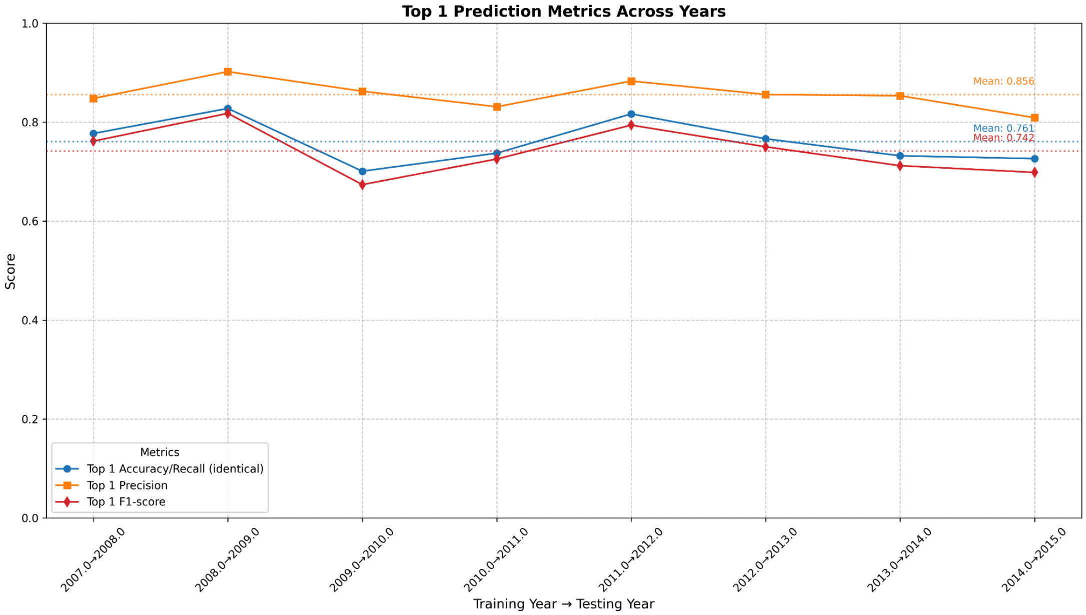
This graph compares precision scores for Top 1 and Top 3 predictions, showing that Top 1 precision is much higher (mean: 85.6%) than Top 3 precision (mean: 27.1%). The lower Top 3 precision reflects that making three predictions per test instance introduces more incorrect guesses, reducing the precision metric.



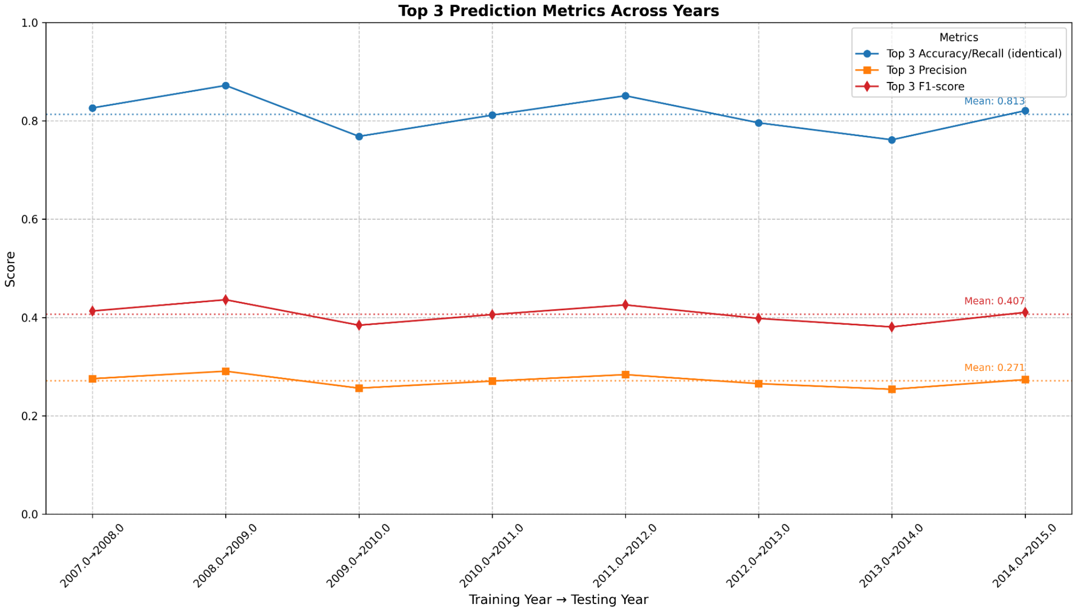
Since recall measures how often the actual missing player appears in the model's predictions, Top 3 recall is consistently higher than Top 1 recall, reinforcing that the correct player is more frequently included in the top three guesses. The recall values mirror the accuracy graph, since every test instance has exactly one missing player, making recall and accuracy identical.



This graph tracks the F1-score for Top 1 and Top 3 predictions over different training/testing years. The Top 1 F1-score remains high (mean: 74.2%), indicating a good balance between precision and recall, whereas the Top 3 F1-score is significantly lower (mean: 40.7%) due to the model making three predictions per test instance, lowering precision.



This graph overlays Top 1 accuracy, precision, recall, and F1-score, highlighting that accuracy and recall are identical, while precision is slightly higher. The F1-score is lower than precision due to its dependence on recall, but overall, the model maintains stable performance across different years.



This graph visualizes Top 3 accuracy, precision, recall, and F1-score, with recall and accuracy being identical as in the Top 1 graph. The Top 3 F1-score is low (40.7%) due to precision being much lower (27.1%), a direct result of the model making three predictions per instance, increasing incorrect guesses.

Test Data Mode:

In test data mode, the model is trained on NBA matchup data from 2007 to 2015 and evaluated on a separate curated test dataset. The results in this mode show a Top 1 accuracy of 52.1%, which is notably lower than in the training/testing year pairs mode. This drop in accuracy is largely due to conflicts within the test data, where some instances contain players who were not active in the training seasons or where lineup changes create situations with multiple possible correct answers. The Top 3 accuracy reaches 62.6%, meaning that in most cases, the actual missing player appears in the model's top three choices. However, the Top 3 F1-score is lower at 31.3%, with a precision of 20.87%. This is because the model makes three predictions per test instance, and only one can be correct, meaning at least two predictions are incorrect for all predictions, three when there is no correct prediction, which lowers precision. These variations in accuracy and F1-score highlight the challenges posed by inconsistent or overlapping player rosters across different seasons.. Below is a table containing all the discussed evaluation metrics and their values for this mode:

Prediction Method	Evaluation Metric	Value
Top 1 Prediction	Accuracy	0.5210
	Precision	0.7298 (0.5210) *
	Recall	0.6984 (0.5210) *
	F1-Score	0.6901 (0.5210) *
Top 3 Prediction	Accuracy	0.6901
	Precision	0.2087
	Recall	0.6260
	F1-Score	0.3130

Metrics marked with an asterisk are inflated in the program output due to the nature of scikit-learn having trouble calculating inconsistent label mapping. The four metrics—accuracy, precision, recall, and F1-score—are identical for the Top 1 prediction in test data mode because the model makes exactly one prediction per test instance, and there is only one correct answer for each case. This means that the total number of predictions is equal to the total number of actual labels, ensuring that true positives, false positives, and false negatives are counted in the same way. As a result, precision and recall become equal, making the F1-score the same as well. Since accuracy also measures the proportion of correct predictions, all four metrics share the same value.

Data Preprocessing

To maintain high-quality and consistent input data, the model goes through several preprocessing steps. It first loads NBA matchup data from multiple CSV files covering different seasons. In test data mode, missing player labels are filled in using a separate dataset to ensure completeness. Player and team names, which are categorical data, are converted into numerical values using Label Encoding so that they can be processed by the machine learning model. Continuous variables, like player win rates and synergy scores, are scaled to prevent any one feature from dominating the predictions. Extra care is taken when handling unknown players in the test set, filtering out those who were not active during the relevant season to maintain prediction accuracy. These steps help the model identify patterns in past data and apply them effectively to new matchups.

Feature Selection & Engineering

Feature selection plays a crucial role in the NBA lineup prediction model, ensuring that only the most relevant data is used to make accurate predictions. One of the most important features is the outcome column, which helps the model evaluate lineup effectiveness. In training/testing year pairs mode, the presence of this column allows the model to compute player win rates, team win rates, and synergy scores, leading to a Top 1 accuracy of 76.05% by identifying which players contribute most to winning lineups. It relies on player frequency and team history,, contributing to a lower Top 1 accuracy of 52.1%. This highlights the impact of feature selection, as the availability or absence of key features can significantly affect predictive accuracy.

Feature engineering enhances the model's predictive power by creating new metrics that highlight meaningful relationships within the data. The model derives player win rates to quantify individual contributions to winning teams and team win rates to measure overall team strength. Synergy scores are introduced to evaluate the effectiveness of player combinations, while player frequency tracks how often a player appears in matchups, helping to assess their role. Additionally, the model calculates win rate differences to compare home and away team strengths and uses statistical measures like minimum, maximum, and standard deviation of win rates to understand variations in team composition. These engineered features provide the model with deeper insights into team dynamics, player effectiveness, and lineup synergy, ultimately improving prediction accuracy.

Conclusions

The NBA lineup prediction model successfully identifies missing players in historical matchups by leveraging XGBoost and well-engineered features. It captures patterns in team compositions, player interactions, and game outcomes, achieving strong accuracy and recall scores across different evaluation methods. However, its performance varies between training/testing year pairs mode and test data mode, highlighting how the model generalizes under different conditions.

In training/testing year pairs mode, where the model is trained on one season and tested on the next, it achieves a Top 1 accuracy of 76.05% and a Top 1 F1-score of 74.17%, demonstrating strong predictive performance. The Top 3 accuracy of 81.35% shows that even when the first prediction is incorrect, the actual missing player is still included among the model's top three choices most of the time. However, the Top 3 F1-score is lower at 40.67%, not because of ranking issues but because the model makes three predictions per instance, and at least two of them will usually be incorrect, reducing precision.

In test data mode, where the model is trained on multiple seasons (2007-2015) and tested on a fixed dataset, the Top 1 accuracy drops to 52.1%. This lower accuracy is largely due to inconsistencies in the test data, such as players appearing in seasons they were not active in, making correct predictions more difficult. Unlike in training/testing mode, where precision differs from recall, in test data mode, all four metrics—accuracy, precision, recall, and F1-score—are identical at 52.1% because the number of predictions exactly matches the number of actual labels, meaning false positives and false negatives are counted the same way. Additionally, Top 3 accuracy reaches 62.6%, showing that while the model doesn't always select the correct player first, it still includes them in its top three choices in most cases.

These results show that while the model generalizes well when tested on sequential seasons, it faces more challenges when applied to a fixed test dataset with inconsistencies in player rosters. Despite this, its ability to consistently identify likely candidates, even in more difficult scenarios, reinforces its strength in analyzing player contributions, team dynamics, and historical performance trends. Ultimately, the model serves as a valuable tool for predicting players based on past NBA data.