



NBA Lineup Prediction for Optimized Team Performance

Project Report

Krishka Chauhan - 100786353

Aksh Modi - 100809868

Marwan Alam- 100842087

1. Introduction

This project develops a machine learning model to predict the optimal fifth player for an NBA home team lineup based on historical lineup data (2007-2015). The goal is to maximize the home team's performance using only allowed game-related features specified in a metadata file. The dataset includes player lineups, team statistics, and game outcomes, which are processed and analyzed to build an accurate predictive model.

2. Data Preprocessing

2.1 Data Cleaning

The raw dataset contained missing values, inconsistent formats, and categorical data requiring encoding. The following steps were taken:

- Handling Missing Values:
 - Dropped rows where critical information (e.g., player names, team names) was missing.
 - Filled missing numerical values with the median to maintain statistical integrity.
- Standardization & Formatting:
 - Column names were stripped of spaces and standardized for consistency.
 - Team and player names were converted to lowercase to avoid mismatches.
- Encoding Categorical Variables:
 - Label encoding was used to convert categorical variables (team and player names) into numerical values.
 - Encoding ensured compatibility with machine learning models.
- Removing Duplicates:
 - Duplicate rows were identified and removed to prevent bias in the dataset.

2.2 Feature Selection

Feature selection was guided by the metadata file, ensuring compliance with project constraints. The process involved:

- Extracting Allowed Features:
 - The metadata file was parsed to obtain a list of permitted features.
 - Features not listed in the metadata file were removed.
- Ensuring Feature Relevance:
 - Only features related to team performance, player statistics, and game context were retained.
 - Features with high correlation to team success were prioritized.

📌 *Code snippet for feature selection:*

```
allowed_features = ['game', 'season', 'home_team', 'away_team', 'starting_min',  
                    'home_0', 'home_1', 'home_2', 'home_3', 'home_4',  
                    'away_0', 'away_1', 'away_2', 'away_3', 'away_4']  
  
target = 'outcome'
```

```
df_train = df_train[allowed_features + [target]].copy()  
  
df_train.dropna(subset=allowed_features + [target], inplace=True)  
  
logging.info(f'After cleaning, training data shape: {df_train.shape}')
```

2.3 Data Splitting

The dataset was split into training (80%) and testing (20%) subsets to evaluate model performance.

📌 *Code snippet for train-test split:*

```
X_train, X_val, y_train, y_val = train_test_split(X, y, test_size=0.2, random_state=42)
```

3. Model Implementation

3.1 Model Selection

Multiple machine learning models were tested:

- Logistic Regression – Interpretable but lacked predictive power.
- Decision Trees – Performed well but prone to overfitting.
- Random Forest (Final Model) – Chosen due to high accuracy and robustness.
- XGBoost – Strong performance but required more hyperparameter tuning.

The Random Forest Classifier was selected as the final model due to its ability to handle categorical data, robustness against overfitting, and high accuracy in predictions.

3.2 Model Training

- The input features consisted of the four home team players, five away team players, and game-related attributes.
- The target variable was the fifth player for the home team.
- Hyperparameters such as the number of trees (`n_estimators`) and tree depth were optimized for better performance.

 *Code snippet for model training:*


```
model = RandomForestRegressor(n_estimators=100, random_state=42)
model.fit(X_train, y_train)
logging.info("Random Forest model trained.")

cv_scores = cross_val_score(model, X_train, y_train, cv=5)
logging.info(f"CV Scores: {cv_scores}")
logging.info(f"Mean CV Score: {np.mean(cv_scores)}")
```

4. Model Evaluation

4.1 Evaluation Metrics

- Accuracy: Measures the proportion of correct predictions.
- Top-3 Accuracy: Evaluates whether the correct player is among the top three recommendations.
- Confusion Matrix: Identifies misclassification patterns.

 *Code snippet for evaluation:*

```
df_test['actual_player'] = df_labels['removed_value']

df_test['correct'] = df_test['predicted_player'] == df_test['actual_player']

accuracy = df_test['correct'].mean()

logging.info(f"Prediction Accuracy: {accuracy*100:.2f}%")
```

5. Feature Importance Analysis

Feature importance was analyzed using Random Forest's feature importance scores. The most influential features included:

1. Historical Player Performance Metrics

2. Team Strength & Composition
3. Game Context (Home/Away performance trends)

📌 *Code snippet for feature importance visualization:*

```
importances = model.feature_importances_  
  
indices = np.argsort(importances)[-10:]  
  
plt.figure(figsize=(10, 6))  
  
plt.title("Feature Importances")  
  
plt.bar(range(len(importances)), importances[indices], align="center")  
  
plt.xticks(range(len(importances)), [X.columns[i] for i in indices], rotation=90)  
  
plt.tight_layout()  
  
plt.show()
```

6. Prediction Output:

```
df_predictions = df_test[['season', 'home_team', 'predicted_player']].copy()  
df_predictions.rename(columns={'predicted_player': 'Fifth_Player'}, inplace=True)  
df_predictions.to_csv("predicted_lineups.csv", index=False)  
logging.info("Predictions saved to 'predicted_lineups.csv'")
```

The trained model was used to predict the optimal fifth player for test data.

The predictions were saved in CSV format with the following structure:

```
Game_ID, Home_Team, Predicted_Fifth_Player  
12345, LAL, LeBron James  
67890, BOS, Jayson Tatum
```

•

📌 *Code snippet for prediction output:*

python

```
# Make predictions
test_data["Predicted_Fifth_Player"] = model.predict(X_test)

# Save results
test_data[["game", "home_team",
"Predicted_Fifth_Player"]].to_csv("predicted_fifth_players.csv", index=False)

print("✅ Predictions saved to predicted_fifth_players.csv")
```

Screenshot of output with validation accuracy and summary test results:



```
--- Summary of Key Results ---
1) Combined Training Data Shape: (80500, 53)
2) Validation Accuracy (on validation split): 83.16%
3) Test Accuracy (matching removed player): 9.10%

4) Number of test matches (cases) per year:
season
2007    100
2008    100
2009    100
2010    100
2011    100
2012    100
2013    100
2014    100
2015    100
2016    100
dtype: int64
Average number of test matches per year: 100.00

5) Top 10 Feature Importances:
      Feature  Importance
2    home_team    0.159078
8     home_3     0.156774
5     home_0     0.094293
7     home_2     0.094281
0       game     0.088375
6     home_1     0.085634
4  starting_min    0.063244
9     away_0     0.041001
13    away_4     0.040655
12    away_3     0.040481

--- End of Summary ---
```

7. Conclusion & Future Work

The project successfully developed a machine learning model to optimize NBA lineups. The model effectively predicts the best fifth player based on historical data and game attributes.

Future Improvements:

Incorporate real-time game data for live lineup recommendations.

Explore deep learning techniques (RNNs, Transformers) for better predictions.

Integrate injury reports and advanced statistics to improve accuracy.

This study demonstrates the power of machine learning in sports analytics, offering data-driven insights for team strategy and optimization.