Project Progress I - Predicting the outcomes of NBA games

Introduction :

The goal of this project is to develop a predictive model that forecasts the outcomes of NBA games using a comprehensive dataset of game and player statistics. Specifically, the model aims to predict which team will win each game, and further, it seeks to estimate the margin of victory.

The dataset consists of NBA game data from the 2004 to 2020 seasons, including both team-level and player-level statistics. Key features such as points scored, assists, rebounds, and turnovers are analyzed to identify the most significant predictors of game outcomes. Additionally, contextual factors like home-court advantage and player injuries are considered for their potential impact on game results.

To achieve this, we are employing a supervised learning approach, using classification methods to predict game outcomes (win/loss) and regression techniques to estimate the point margin of victory. This progress report focuses on the data preprocessing and exploratory data analysis (EDA) stages, which are critical for understanding the dataset and preparing it for model development.

Data Overview :

The dataset for this project was sourced from Kaggle, which provides a comprehensive collection of NBA game data spanning the 2004 to 2020 seasons. The original dataset included several files, but I have carefully selected the ones most relevant to the project to focus on the most impactful data while reducing unnecessary processing time and resource usage. Specifically, I excluded the teams.csv file, which only contains a list of teams in the league, as it does not add value to the predictive modeling process.

The final dataset comprises four main files that include game-level, player-level, and team-ranking data. A summary of the included files is as follows:

File	Key fields	Description
games.csv	GAME_DATE_EST, HOME_TEAM_ID, VISITOR_TEAM_ID, PTS_home, PTS_away, HOME_TEAM_WINS	Game-level data including dates, teams, points scored, and the outcome (win/loss).
games_details.csv	GAME_ID, PLAYER_ID, PLAYER_NAME, MIN, PTS, REB, AST, PLUS_MINUS	Player-level statistics for each game, including points, minutes played, rebounds, and assists.
players.csv	PLAYER_NAME, TEAM_ID, PLAYER_ID, SEASON	Metadata about the players, including player names, team associations, and seasons they played.
ranking.csv	TEAM_ID, SEASON_ID, W, L, W_PCT, HOME_RECORD, ROAD_RECORD	Team standings data, tracking wins, losses, and performance at home and on the road.

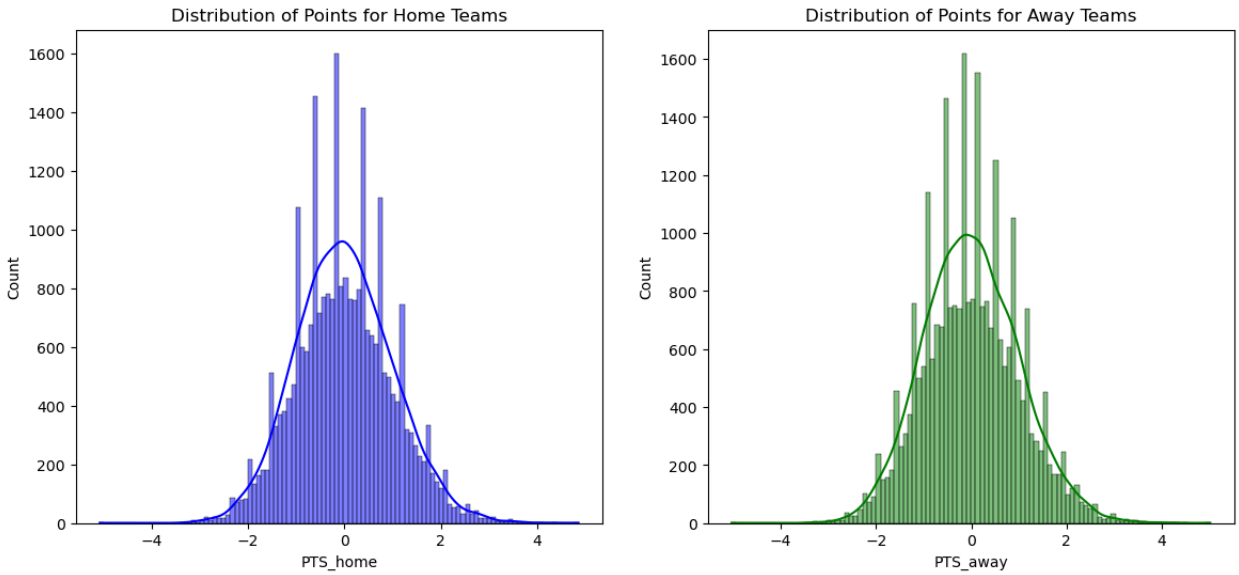
These selected files contain essential information for analyzing and predicting game outcomes. By excluding irrelevant data, such as the teams.csv file and other fields from the selected files, I aimed to streamline the analysis and ensure a more manageable scope, which helps reduce computation time and the complexity.

Data Preprocessing and Exploratory Data Analysis :

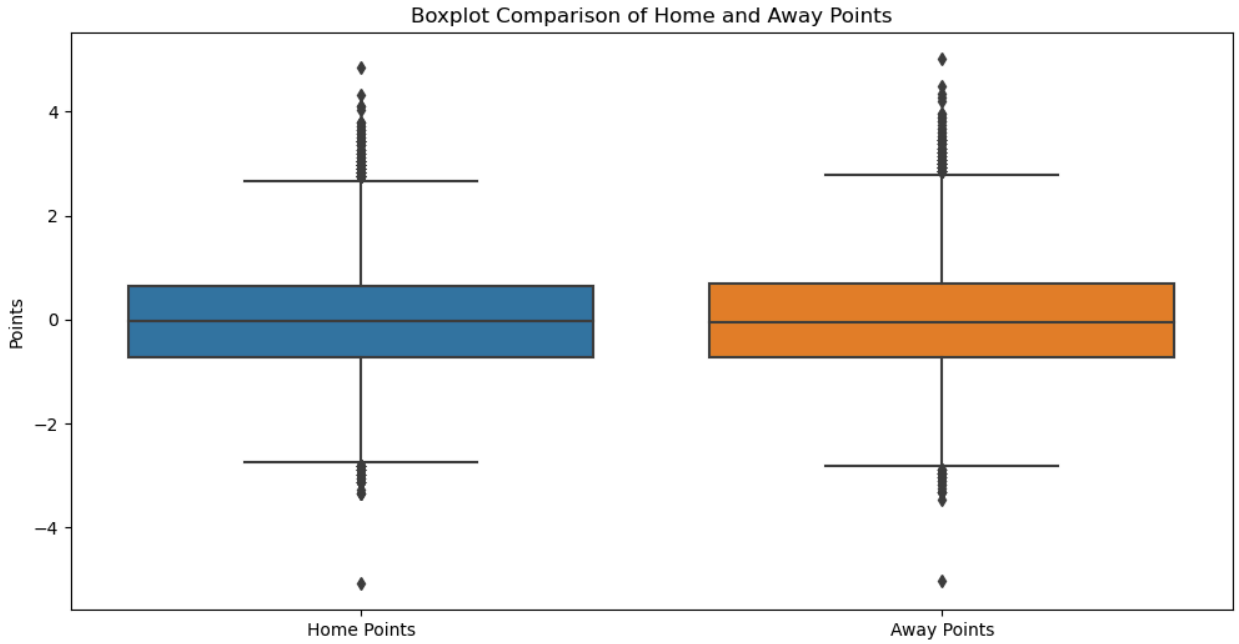
In the data preprocessing stage, we begin by handling missing values in both the game and player statistics datasets. Rows with significant missing data will either be removed or have their values imputed based on the context and availability of related information. After that, the focus is on feature selection, keeping only the most relevant variables for predicting game outcomes. Categorical variables, such as team and player names, will be encoded into numerical formats to ensure compatibility with machine learning models. Additionally, I will apply scaling to numeric features like points, assists, and rebounds to standardize them for analysis.

For the exploratory data analysis, I generated summary statistics to understand overall trends in team and player performance. This includes reviewing key statistics like points scored, assists, and shooting percentages. I also use data visualizations to explore the relationships between variables, creating histograms, box plots, and heatmaps to uncover patterns in game outcomes. Finally, correlation analysis will help identify the most important predictors of game results, focusing on features such as shooting accuracy, assists, and rebounds.

Data Visualizations :

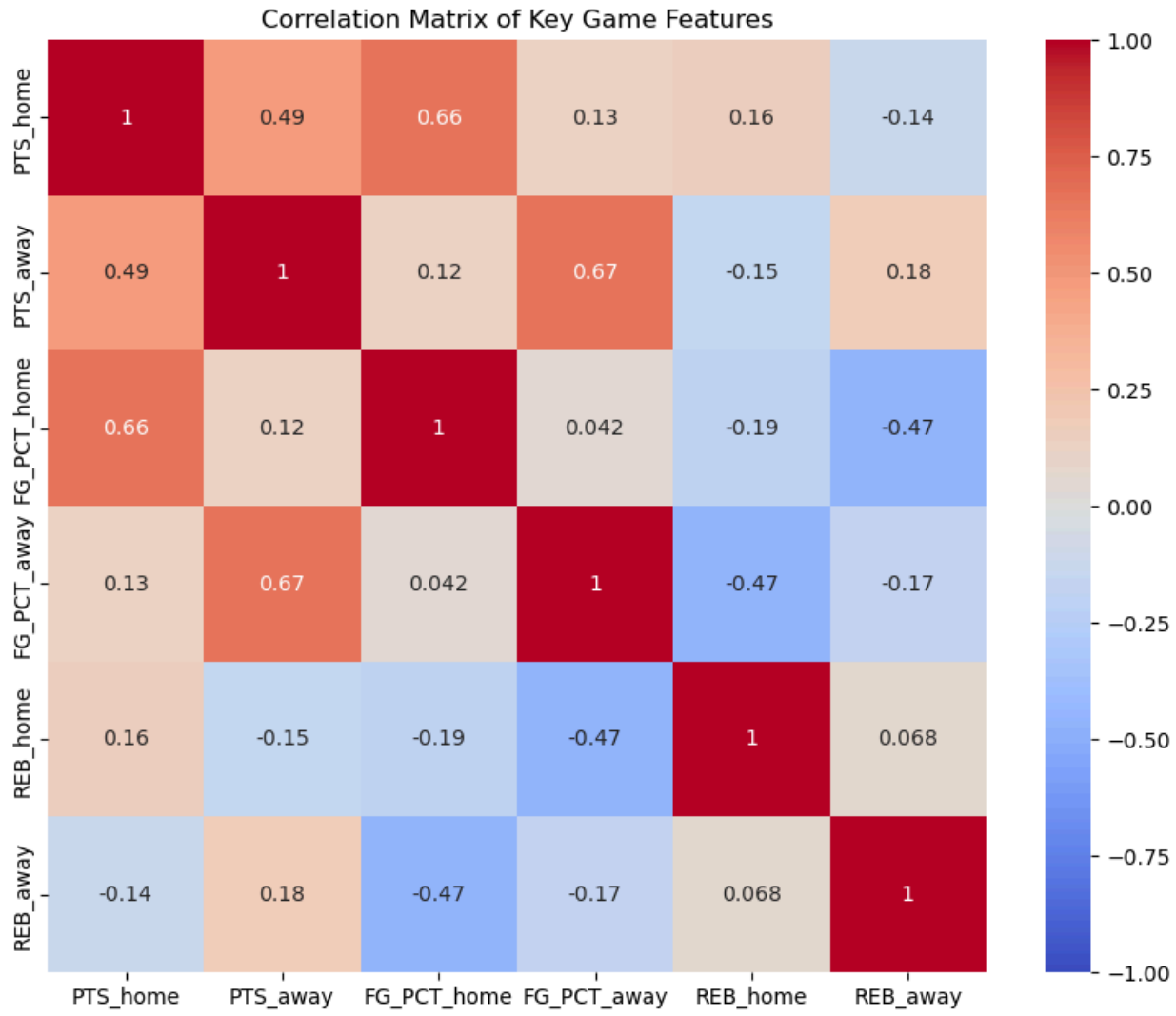


The distributions of points for both home and away teams appear to follow a normal-like pattern, with most games clustering around a central score. Both distributions are symmetrical, suggesting that most teams tend to score within a consistent range. There's no significant skew in either distribution, indicating that both home and away teams perform predictably in terms of scoring. This visualization helps confirm the overall balance in team performance when it comes to points scored.

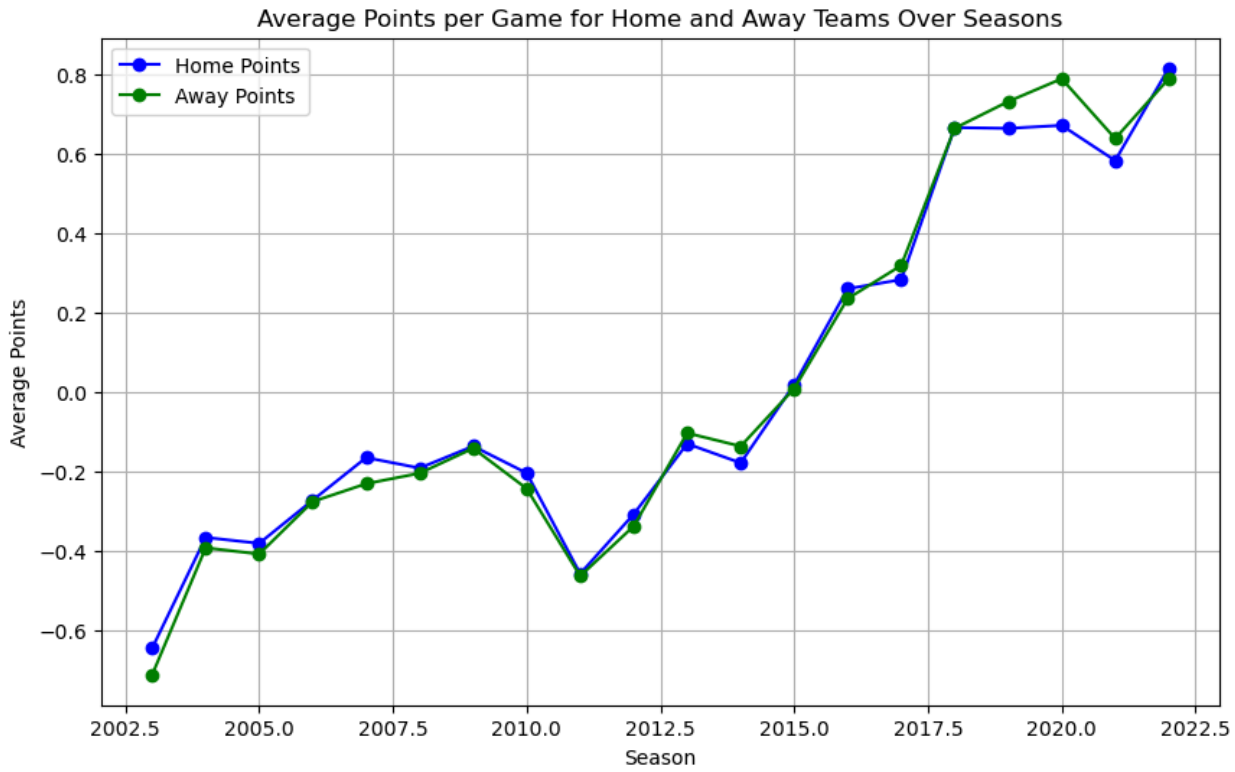


The boxplot comparison of home and away points shows that both distributions are quite similar in terms of their spread. Both sides show some outliers, mostly in the higher range, showing

games where teams overperformed. The home teams may have a slightly wider spread in their scoring patterns, indicating more variability in their performance.



The correlation matrix shows the relationships between key game features such as points, field goal percentages, and rebounds for both home and away teams. There is a strong positive correlation between home team points and field goal percentage, as well as between away team points and field goal percentage, indicating that better shooting accuracy is closely linked to higher scores. Rebounds show weaker correlations with points and field goal percentages, meaning that it is important, but they may not be as directly tied to scoring outcomes as shooting efficiency.



This line plot shows the average points per game for both home and away teams across seasons. From 2010 to 2020, there is an upward trend in scoring for both home and away teams. The points for home and away teams remain quite close throughout the seasons, with away teams occasionally outscoring home teams in more recent years. This suggests that home-court advantage may not mean as much anymore, or even that away teams have improved their performance over time. The most recent seasons exhibit the highest average points, reflecting a high-scoring trend in modern NBA games.

Code :

```
# Importing the necessary libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.preprocessing import StandardScaler

# Loading the datasets
games = pd.read_csv('C:/Users/patron/Desktop/MATH 748/NBA Data/games.csv')
games_details = pd.read_csv('C:/Users/patron/Desktop/MATH 748/NBA
Data/games_details.csv')
players = pd.read_csv('C:/Users/patron/Desktop/MATH 748/NBA
Data/players.csv')
ranking = pd.read_csv('C:/Users/patron/Desktop/MATH 748/NBA
Data/ranking.csv')

# Basic exploration to check the shape and first few rows of each dataset
print("Games Dataset:")
print(games.shape)
print(games.head())

print("\nGames Details Dataset:")
print(games_details.shape)
print(games_details.head())

print("\nPlayers Dataset:")
print(players.shape)
print(players.head())

print("\nRanking Dataset:")
print(ranking.shape)
print(ranking.head())

# Checking for missing values in each dataset
print("\nMissing Values in Games Dataset:")
print(games.isnull().sum())

print("\nMissing Values in Games Details Dataset:")
```

```
print(games_details.isnull().sum())

print("\nMissing Values in Players Dataset:")
print(players.isnull().sum())

print("\nMissing Values in Ranking Dataset:")
print(ranking.isnull().sum())

# Handling any missing values
games_details_cleaned = games_details.dropna()
games_details_cleaned = games_details.fillna(games_details.mean())

# Feature selection
games_selected = games.drop(columns=['GAME_ID', 'GAME_STATUS_TEXT'])
games_details_selected = games_details.drop(columns=['PLAYER_ID',
'NICKNAME', 'COMMENT'])

# Encoding the categorical variables
games_details_selected['TEAM_ABBREVIATION'] =
games_details_selected['TEAM_ABBREVIATION'].astype('category').cat.codes

# Scaling some of the features
scaler = StandardScaler()
games_selected[['PTS_home', 'PTS_away', 'REB_home', 'REB_away']] =
scaler.fit_transform(
    games_selected[['PTS_home', 'PTS_away', 'REB_home', 'REB_away']])
```

```
# Generating summary statistics for key features in the games dataset
print("\nSummary statistics for Games dataset:")
print(games_selected[['PTS_home', 'PTS_away', 'FG_PCT_home',
'FG_PCT_away', 'REB_home', 'REB_away']].describe())

# Visualizing the distributions of key features using histograms
plt.figure(figsize=(14, 6))

# Histogram for points scored by home teams
plt.subplot(1, 2, 1)
sns.histplot(games_selected['PTS_home'], kde=True, color='blue')
plt.title('Distribution of Points for Home Teams')
```

```
# Histogram for points scored by away teams
plt.subplot(1, 2, 2)
sns.histplot(games_selected['PTS_away'], kde=True, color='green')
plt.title('Distribution of Points for Away Teams')

plt.show()

# Step 12: Boxplots for comparing performance of home vs away teams
plt.figure(figsize=(12, 6))

# Boxplot for home and away points
sns.boxplot(data=games_selected[['PTS_home', 'PTS_away']])
plt.title('Boxplot Comparison of Home and Away Points')
plt.ylabel('Points')
plt.xticks([0, 1], ['Home Points', 'Away Points'])

plt.show()

# Correlation heatmap to identify relationships between features
plt.figure(figsize=(10, 8))
corr_matrix = games_selected[['PTS_home', 'PTS_away', 'FG_PCT_home',
'FG_PCT_away', 'REB_home', 'REB_away']].corr()
sns.heatmap(corr_matrix, annot=True, cmap='coolwarm', vmin=-1, vmax=1)
plt.title('Correlation Matrix of Key Game Features')
plt.show()

# Analyzing win distributions (home vs away)
win_counts = games_selected['HOME_TEAM_WINS'].value_counts()

plt.figure(figsize=(6, 6))
sns.barplot(x=win_counts.index, y=win_counts.values, palette='Set2')
plt.title('Home vs Away Wins')
plt.xticks([0, 1], ['Away Wins', 'Home Wins'])
plt.ylabel('Number of Games')
plt.show()

# Trend analysis over season by season
seasonal_points = games_selected.groupby('SEASON')[['PTS_home',
'PTS_away']].mean()
```

```
# Plotting the trend of points over seasons
plt.figure(figsize=(10, 6))
plt.plot(seasonal_points.index, seasonal_points['PTS_home'], label='Home
Points', marker='o', color='blue')
plt.plot(seasonal_points.index, seasonal_points['PTS_away'], label='Away
Points', marker='o', color='green')
plt.title('Average Points per Game for Home and Away Teams Over Seasons')
plt.xlabel('Season')
plt.ylabel('Average Points')
plt.legend()
plt.grid(True)
plt.show()
```