NBA Player Performance Prediction

Project Overview

This project predicts the top 10 NBA players for the next season based on historical performance data. The prediction is based on key performance metrics such as points scored, rebounds, assists, steals, and blocks.

Project Structure

- `scrape_nba_data.py`: Script to scrape NBA player data from the NBA stats website.
- `clean_nba_data.py`: Script to clean and preprocess the scraped data.
- `visualize_nba_data.py`: Script to visualize the cleaned data.
- `predict_top_players.py`: Script to train a model and predict the top 10 players for the next season based on historical data.

Setup

1. Clone the repository:

```
git clone https://github.com/yourusername/nba-player-performance-prediction.git
cd nba-player-performance-prediction
```

2. Install the required packages:

```
pip install pandas scikit-learn requests
```

Usage

Step 1: Scrape NBA Data

Run the `scrape_nba_data.py` script to scrape the NBA player data. This script collects player statistics from the NBA stats website.

```
python scrape_nba_data.py
```

Step 2: Clean NBA Data

Run the `clean_nba_data.py` script to clean and preprocess the scraped data. This script filters out players with insufficient playing time and combines regular season and playoff stats.

```
python clean_nba_data.py
```

Step 3: Visualize NBA Data

Run the `visualize_nba_data.py` script to visualize the cleaned data. This script generates various plots and charts to provide insights into the data.

```
python visualize_nba_data.py
```

Step 4: Predict Top Players

Run the `predict_top_players.py` script to train a model and predict the top 10 players for the next season. This script uses historical data to create lag features, trains a regression model, and ranks players based on a composite score.

```
python predict_top_players.py
```

Explanation

Data Preparation

1. **Filter Out Players:** The data is filtered to exclude players with fewer than 200 minutes in the regular season and fewer than 50 minutes in the playoffs.
2. **Combine Stats:** Regular season and playoff stats for each player are combined to get a comprehensive view of their performance.
3. **Feature Engineering:** Lag features are created for the past year's performance metrics to predict the next season's performance.

Model Training

1. **Target and Features:** The target variable is points scored (PTS), and the features are the lag features created from past performance.
2. **Train/Test Split:** The data is split into training and testing sets.
3. **Model Training:** A linear regression model is trained to predict the next season's performance.

Prediction and Ranking

1. **Predict Performance:** The most recent season's data is used to predict the next season's performance for each player.
2. **Composite Score:** A composite score is calculated for each player based on predicted points, rebounds, assists, steals, and blocks.
3. **Rank Players:** Players are ranked based on the composite score to identify the top 10 performers for the upcoming season.

Composite Score Calculation

The composite score is calculated as a weighted sum of several key performance metrics for each player. The formula used for calculating the composite score is as follows:

```
composite_score = (PTS_predicted) +
                  (REB * 0.7) +
                  (AST * 0.5) +
                  (STL * 0.3) +
                  (BLK * 0.3)
```

- **PTS_predicted:** The predicted points scored by the player for the next season, as determined by the regression model.
- **REB:** The total rebounds by the player.
- **AST:** The total assists by the player.
- **STL:** The total steals by the player.
- **BLK:** The total blocks by the player.

Each of these components contributes to the composite score, with higher weights assigned to metrics that are generally considered more indicative of a player's impact on the game. This composite score is then used to rank the players, identifying the top performers for the upcoming season.