NBA 2013 Dataset - Feature Explanation

Player Information:

- player: Player's full name.
- pos: Position (PG, SG, SF, PF, C).
- age: Player age during the season.
- bref_team_id: 3-letter team code (from Basketball-Reference).
- season: Season label (e.g., 2013).
- season_end: Year when the season ended (e.g., 2014).

Games Played:

- g: Games played.
- gs: Games started.
- mp: Minutes played per game.

Shooting - Overall:

- fg: Field goals made per game.
- fga: Field goals attempted per game.
- fg.: Field goal percentage = FG / FGA.

Three-Point Shooting:

- x3p: 3-pointers made per game.
- x3pa: 3-pointers attempted per game.
- x3p.: 3-point percentage = 3P / 3PA.

Two-Point Shooting:

- x2p: 2-point field goals made per game.
- x2pa: 2-point attempts.
- x2p.: 2-point percentage.

Effective Field Goal %:

- efg.: Effective field-goal percentage = (FG + 0.5*3P) / FGA.

Free Throws:

- ft: Free throws made per game.
- fta: Free throws attempted per game.
- ft.: Free throw percentage = FT / FTA.

Rebounding:

- orb: Offensive rebounds per game.
- drb: Defensive rebounds per game.
- trb: Total rebounds per game.

Playmaking & Defense:

- ast: Assists per game.
- stl: Steals per game.
- blk: Blocks per game.
- tov: Turnovers per game.
- pf: Personal fouls per game.

Scoring:

- pts: Points per game.

Exercise: NBA Player Analysis with k-Nearest Neighbors

Objective

Use the k-Nearest Neighbors (k-NN) algorithm to explore similarities between NBA players based on their statistics. You will apply k-NN for both classification and regression tasks.

Part 1: Data Exploration

1. Load the nba_2013.csv dataset.
2. Examine the first few rows.
3. Check for missing values and handle them appropriately.

4. Explore the distributions of key statistics (pts, ast, trb, mp) using histograms or boxplots.

Part 2: k-NN Classification

Task: Predict the **player position (pos)** based on numeric statistics.

1. Select numeric features that may help predict position. Suggested features: mp, fg, fga, fg., x3p, x3pa, x3p., x2p, x2pa, x2p., efg., ft, fta, ft., orb, drb, trb, ast, stl, blk, tov, pf, pts.
2. Split your dataset into **training (80%)** and **testing (20%)** sets.
3. Scale the numeric features (important for k-NN).
4. Train a k-NN classifier (try k=5 as a start) to predict pos.
5. Evaluate your model using **accuracy** and a **confusion matrix**.
6. Experiment with different values of k and discuss how it affects the results.

Questions:

- Which positions are easiest to predict? Which are hardest?
- How do the shooting stats (fg., x3p.) contribute to predicting position?

Part 3: k-NN Regression

Task: Predict **points per game (pts)** based on other numeric statistics.

1. Select predictor features (exclude pts and pos).
2. Split the dataset into training and testing sets.
3. Scale numeric features.
4. Train a k-NN regressor (start with k=5) to predict pts.
5. Evaluate your model using **Mean Squared Error (MSE)** and **R^2 score**.
6. Experiment with different values of k and discuss the results.

Questions:

- Which features seem most important for predicting points?
- Does increasing k improve or worsen your predictions?

Part 4: Player Similarity Exploration

1. Pick any player in the dataset.
2. Use k-NN to find the **5 most similar players** based on numeric stats.
3. Compare the players’ stats and positions. Do the nearest neighbors make sense?
4. Visualize the neighbors using a scatter plot of two key stats (e.g., pts vs ast).