Project theme: NBA Trade Analysis

By: James Lim, Quang Nguyễn Đức, Neel Lal

1. **Introduction:**
    - Our project goals were to determine whether a trade between two players is beneficial for an inputted team. We used the team-record of a team after the trade as a measure for how beneficial the trade is for the team

2. **Data:**
    - Our dataset is presented in the csv file "2017-2018-statistic.csv" and was found on kaggle.com.
    - The data being displayed are in order [player name, player's team, play-minutes, points in game, assists in game, rebounds in game, blocks in game, steals in game, turnovers in game, field goal attempts, field goals made, player's height, player's position].
    - The csv contains all of these statistics for every single player in the NBA in every single regular season game of the 2017-2018 season.
    - The data are displayed in a table of lists. Every list is the player-record of one game.
    - The range of our data is 26,110 rows and 13 columns. This data includes every player and their stats in each regular season game.
    - The database is listed in alphabetical order.
    - For every player, the first occurrence of their name is the first game they played in and the last occurrence of their name is the last game they played in for the 2017-2018 NBA season.

3. **Algorithm:**

a. For the sake of consistency, we will define:

- 2 types of record:
    - a player-record: is a list contains:
        + "player's displayed Name" (string)
        + "team name abbreviation" (string)
        + "player's time" (integer)
        + "player's points" (integer)
        + "player's assists" (integer)
        + "player's rebound" (integer)
        + "player's blocks" (integer)
        + "player's steals" (integer)
        + "player's Turn Over" (integer)
        + "player's FGA" (integer)
        + "player's FGM" (integer)
        + "player's Height" (integer)
        + "player's position" (string)

- a team-record: is a list contains:
  + "team name abbreviation" (string)
  + "current players' total play time" (integer)
  + "total players' points" (integer)
  + "total players' assists" (integer)
  + "total players' rebound" (integer)
  + "total players' blocks" (integer)
  + "total players' steals" (integer)
  + "total players' Turn Over" (integer)
  + "total players' FGA" (integer)
  + "total players' FGM". (integer)
- when we mention just "record": it can mean team-record or player-record.
- An entry is an element in a record such as "player's displayed Name", "player's points", "total players' steals"...
- Database is a list contains records.
- **Important Acknowledgement:**
  - The database has the data of entire 2017-2018 season. Because there is trading in the middle of the season, some players might be classified in multiple teams in player-records.

    For example: Blake Griffin played for LAC before he went to DET.

    We would count his statistics of the entire season for his latest team, which is DET.

b. What our algorithm can do:

- Our algorithm can predict the team-records for the next season to see how the teams will perform with their current .
- From the input database, we come up with a decent ranking algorithm that gives a fairly accurate assessment of teams' performance in the 2017-2018 season and applicable to predict teams' performance in the 2018-2019 season.
- We also come up with an algorithm to simulate trading. We can input 2 players and the algorithm will output their teams' stat before and after the trade (using **"trade"** procedure), and we can see how their ranking points go up or down (using **"calculate-rank-trade"**). We can use this information to see if the trade is beneficial.

c. How do we do that:

- Overview of our code:
  + Our code consists of 6 big sectors:

Project theme: NBA Trade Analysis
By: James Lim, Quang Nguyễn Đức, Neel Lal

Players' statistic: This sector has procedures that work with the database of player-records and player-records. We can filter player-records belong to a specific player from a database containing player-records, sum up the stats of that player from a database containing that player's records.

Teams' statistic: This sector has procedures that solve the problems that relate to teams. We can identify if a player-record belongs to a specific team, filter from the database of player-records to get player-records of 1 specific team, add up stats of a team's players to achieve the team-record,

Diluted algorithm: This sector has procedures that are used to dilute some entries in a player-record **("player's points"-"player's assists"-"player's rebound"-"player's blocks"-"player's steals"-"player's Turn Over"-"player's FGA"-"player's FGM")**. The algorithm is taking his stat from one entry, for example his total points in 2017-2018 season. His diluted point would be:

(point * (number-of-games-he-played-this-season / 82) * point) / 2

This algorithm increases the stats of players who fail to play all the game in the season. Thus, we give him a fair assessment when we trade him.

Predicting algorithm: This sector has procedures that predict the stats of teams next season if they keep their current players **(categories applied "total players' points"-"total players' assists"-"total players' rebound"-"total players' blocks"-"total players' steals"-"total players' Turn Over"-"total players' FGA"-"total players' FGM")**. The procedures will work mainly with team-records and database of team-records.

Trading simulation: This sector has procedures that recalculated the stats of team-records if we trade 2 players of different teams to each other. The main procedure "trade" takes 2 parameters (name of 2 players) and outputs their team-records before and after trade.

Assessment and Ranking: This sector has procedures that calculate the ranking point of for each team, based on that team's **"total players' points"-"total players' assists"-"total players' rebound"- "total players' blocks"-"total players' steals"-"total players' Turn Over".** The idea is that for each of those categories, we give it a weight (define in "ranking-weight" list), then we multiply entries in team-record with its weight and sum them up. We then rank the whole season by applying ranking proc to each team-record.

+ Predefined databases and variables in our code:

Players-all-games-database: read the csv file "2017-2018 statistic.csv" file into players-all-games-database list (remove the title line).

Players-and-numb-of-games: a list of pairs. Each pair has the format (“player’s name” . “The number of game he played”)

Database-all-players-statistics-all-season: a database of player-records. Each player-record has the stats added up from the entire season. (ie total points of the entire season, etc…)

Teams-and-numb-of-players: a list of pairs. Each pair has the format (“team-name” . “its number of player”)

Database-all-teams-statistics-all-season: a database of team-records. Each team records has the stats added up from its players’ stats taken from **database-all-players-statistics-all-season**.

Database-all-players-statistics-all-season-diluted: this is **Database-all-players-statistics-all-season** with the stats being applied diluted algorithm.

Total-play-time-of-all-players = 5 (players per team on the field) * 48 (minutes of a game) * 82 (games of the entire season).

Database-all-teams-statistics-all-next-season: this is **Database-all-teams-statistics-all-season** with the stats being updated by predicting algorithm.

Ranking-weight: (list 0.05 3 0.05 25 10 -6). The order of elements: "total players' points"-"total players' assists"-"total players' rebound"-"total players' blocks"-"total players' steals"-"total players' Turn Over". Our ranking was determined by trial and error

Ranked-2017-2018-season: ranked teams with their ranking points for 2017-2018 season.

Ranked-2018-2019-season: predicted ranked teams with their ranking points for 2018-2019 season.

- Our step to achieve what we have:

  From the input database, we condense the data so that each player has 1 player-record in **database-all-players-statistics-all-season.** Their stats are added up from the entire season. From **database-all-players-statistics-all-season,** we

condense the data so that each team has 1 record in **database-all-teams-statistics-all-season.** Stats of each team are added up from its players' stats.

From there, we apply diluted algorithm for each entry (that is applicable) in each player-record in **database-all-players-statistics-all-season** to get **database-all-players-statistics-all-season-diluted.**

We apply diluted algorithm for each entry (that is applicable) in each team-record in **database-all-players-statistics-all-season** to get **database-all-players-statistics-all-next-season.**

To calculate trading, trade takes the input of 2 players' name. For each team of those 2 players, it will add the stats (**"play time"-"points"-"assists"-"rebound"-"blocks"- "steals"-"Turn Over"-"FGA"-"FGM"**) of the coming player and subtract the stats of the leaving player. Then multiply by time fraction (old total play time/ new total play time). The output would be: old team-record1, new team-record1, old team-record2, new team-record2.

To rank the teams, we just apply **"calculate-rank-pts"** to team-record. If we have a database, we will use "**rank-all-teams"** (this proc also sort the result too, from high rank to low rank).

Then we use "**calculate-rank-trade"** that removes the reference of team1 and team2 in "**rank-all-teams"** and their respective values by filtering using "get-records-team" using the predicate **"team-name?"**. Then, "**calculate-rank-trade"** appends the new values for team1 and team2 into the rank-all-teams. Then, it sorts the new list using **"compare-team-rank>?"**. The procedure "**trade-result"** compares the placement of team1in **"calculate-rank-trade"** to the placement of team1 in "**rank-all-teams".** If the ranking is better in **"calculate-rank-trade"**, then it reveals that the trade is good. If the ranking is worse in **"calculate-rank-trade"**, then it reveals that the trade is bad. If the rank does not change, then it reveals that the trade produces no significant change in ranking.

to output the ranking table after a trading happens, and "**trade-result"** to verbally tell if the trade is good for team1.

- Diluted algorithm:

When calculating a trade that includes a player who did not play the full 82 games, looking at the end of season statistics would under value that player.

For example: Steph Curry, who only played in 51 games throughout the season, would be placed at a lower value than expected. Just because he was injured for part of the season does not mean he would be traded for someone of lesser value.

On the other hand, if we look at each statistics in a per game basis (ex. Points per game) and multiplied that value by 82 to project season stats, we would overvalue players who were difference makers when they played, but who did not play enough to make a difference.

For example, Joel Embiid, who in the last two seasons has only played in 94 of 164 possible games but has performed well throughout those games, will be overvalued. Because Embiid is so injury prone, his value should be depreciated.

Our solution was the construction of a "diluted algorithm," which splits the difference between overvaluing often injured players and undervaluing often injured players; it finds the average between the sum of a player's actual statistics (of 1 entry) in the 2017- 2018 season and the sum of his projected statistics (of that same entry) if the player had played all 82 games at the same pace for the entirety of the season and divides the sum by two. The projected statistics if the player had played all 82 games is the players score per one game multiplied by 82 (the total number of games in the season)

(stat + (82/number-of-games-he-played-this-season) * stat) / 2

- Predicting algorithm:

The predicting algorithm was created to equalize the effects of in-season trades. Imagine that there is a trade between two teams in which one player has played more games and one player played fewer games. If we just added all of the stats of the players last total season, then the team who received the player who played more games would be overvalued

For example, if Joel Embiid was traded for LeBron James, then the sum of the stats for the 76ers would be unrealistically higher because LeBron James has scored what he did throughout the season by playing extra minutes

Our solution, therefore, was to find the total per minute efficiency of the team in the desired statistic and multiply that value by the total minutes played by a team over a season

((team's season-long minute totals)*(total end-of-season team stats)/(total minutes played by team)

(team's season-long minute totals) = (5 players on the field)*(48-minute games)*(82 games)

4. What your analysis show:

Our analysis showed that the best teams in the NBA (from best to worse) were the Golden State Warriors, the Toronto Raptors, the New Orleans Pelicans, the Miami Heat, the Milwaukee Bucks, the San Antonio Spurs, the Philadelphia 76ers, the Oklahoma City Thunders,

the Houston Rockets, the Utah Jazz, the Denver Nuggets, the Indiana Pacers, the Minnesota Timberwolves, the Boston Celtics, the Portland Trail Blazers, the Washington Wizards, the Charlotte Bobcats, the Detroit Pistons, the Los Angeles Clippers, the Orlando Magic, the Brooklyn Nets, the Memphis Grizzlies, the Phoenix Suns, the Sacramento Kings, the Chicago Bulls, and the Atlanta Hawks.

Our analysis also shows that the Detroit Piston should not trade Blake Griffin for Kyle Singler. Our analysis also shows that the Indiana Pacer should trade Tyreke Evans for Stephen Curry.

One bias that could have affected our analysis is the lack of specificity in the recording of time. The dataset we found did not keep track of the seconds each player played and rounded the statistics up and down which means that if a player played 5 minutes and 29 seconds, the play minutes for the player would only be 5 minutes. If this phenomena happens enough, it could result in the player being overvalued. While this problem is certainly significant, the probability that this underestimation occurs in all of games is rather strange. Assuming that there is an equal chance of overestimating and underestimating the data, it is quite likely that time overestimations will to some degree mitigate the underestimations. Another possible objection to our data could be the appearance that the ranking weight is random. First off, the ranking weight was able to predict with a surprising amount of accuracy the teams that made it into the playoffs and the only team it was unable to properly predict had extenuating factors (LeBron James's star power) to explain their placement. Second off, our metrics do make sense. The reason why points are deflated so significantly is because team points tend to be quite a bit larger than all of the other metrics. The reason why assists are valued moderately high is because assists are a measure of synergy and timing and more successful teams tend to do better in both. The reason why rebounds are lower is because rebounding is mainly a measure of height which is only somewhat important. Blocks are given such a large weight because blocks are relatively uncommon and when they happen, they can demoralize their opponent; this demoralization results in the other team playing worse and the opposing score being depressed. The reason steals are higher is because steals represent precise movements and are a measure of speediness. The reason turnovers are ranked the way they are is because turnovers represent the care one team takes with their possession; a team that is uncareful with the ball tends to struggle greatly. Another objection we would like to address is the Golden State Warriors being unable to make a "good trade" according to our algorithm and the Atlanta Hawks being unable to make a "bad trade." The reason we kept this in our code is because we reasoned that the best team cannot get better and the worse team cannot get worse if our metric is based on rankings. Besides, even if our algorithm does not explicitly report the trade good or bad, one can make that realization by looking at the change in its total score.

5. Instruction for running our code:

(trade "player-name1" "player-name2")

- Purpose: update 2 teams' stats when we swap player1 and player2. Output are listed in order: team1's stat before trade, team1's stat after trade, team2's stat before trade, team2's stat after trade.
  For example:
  > (trade "Stephen Curry" "LeBron James")
  '(("GS" 19740 9276 2397 3558 610 655 1228 6957 3498)
  ("GS" 21134 9127 2558 3671 627 624 1285 6920 3547)
  ("CLE" 20471 9384 1910 3460 281 615 1087 7156 3444)
  ("CLE" 19077 9537 1683 3314 236 646 1008 7194 3375))

(calculate-rank-trade "team-name1" "player-name1" "team-name2" "player-name2")

- Purpose: output the new ranked list with updated team-records after trade.
  For example:
  > (calculate-rank-trade "GS" "Stephen Curry" "CLE" "LeBron James")
  '(("GS" 22518.9)
  ("NO" 19149.8)
  ("TOR" 18809.5)
  ("MIL" 17792.9)
  ("SA" 17672.7)
  ("UTA" 17204.95)
  ("OKC" 16818.600000000002)
  ("MIA" 16488.5)
  ("HOU" 16229.5)
  ("DEN" 16153.05)
  ("PHI" 16076.500000000002)
  ("MIN" 15735.449999999999)
  ("NY" 15500.35)
  ("ORL" 15418.3)
  ("POR" 15366.900000000001)
  ("IND" 15340.75)
  ("ATL" 14881.8)
  ("WAS" 14875.949999999999)
  ("CHA" 14776.1)
  ("MEM" 14742.4)
  ("LAL" 14738.400000000001)
  ("BOS" 14729.95)
  ("DET" 14457.150000000001)
  ("SAC" 14415.250000000002)
  ("BKN" 14333.15)
  ("LAC" 13854.6)

("PHO" 13566.55)
("DAL" 13164.5)
("CHI" 12998.3)
("CLE" 12003.55))

(trade-result "team-name1" "player-name1" "team-name2" "player-name2")

- Purpose: verbally comment whether or not trading player 1 and player 2 is a good trade for **team 1.**

  For example:

  > (trade-result "GS" "Stephen Curry" "CLE" "LeBron James")

  "The change is not significant"

Pre-defined database of ranked team for 2017-2018 season:

> ranked-2017-2018-season

'(("GS" 22333.65)
("TOR" 19232.7)
("NO" 19070.05)
("MIA" 18113.850000000002)
("MIL" 17985.25)
("SA" 17716.350000000002)
("PHI" 17691.5)
("OKC" 17562.399999999998)
("HOU" 17000.6)
("UTA" 16885.35)
("DEN" 16353.7)
("IND" 16103.35)
("MIN" 15865.000000000002)
("BOS" 15246.849999999999)
("POR" 15146.599999999999)
("WAS" 15142.0)
("CHA" 15043.95)
("DET" 15030.45)
("NY" 14982.25)
("LAC" 14824.3)
("ORL" 14618.3)
("BKN" 14377.3)
("MEM" 13830.45)
("PHO" 13596.75)
("CLE" 13543.0)
("LAL" 13537.95)

("DAL" 13466.6)
("SAC" 13331.900000000001)
("CHI" 13283.650000000001)
("ATL" 12740.95))

Predefined database for ranked teams for the 2018-2019 season:
ranked-2018-2019-season
'(("GS" 22264.7)
("NO" 19149.8)
("TOR" 18809.5)
("MIL" 17792.9)
("SA" 17672.7)
("UTA" 17204.95)
("OKC" 16818.6)
("MIA" 16488.5)
("HOU" 16229.5)
("DEN" 16153.05)
("PHI" 16076.5)
("MIN" 15735.449999999999)
("NY" 15500.35)
("ORL" 15418.3)
("POR" 15366.9)
("IND" 15340.75)
("ATL" 14881.8)
("WAS" 14875.949999999999)
("CHA" 14776.1)
("MEM" 14742.4)
("LAL" 14738.400000000001)
("BOS" 14729.95)
("DET" 14457.150000000001)
("SAC" 14415.25)
("BKN" 14333.150000000001)
("LAC" 13854.6)
("PHO" 13566.55)
("DAL" 13164.5)

Project theme: NBA Trade Analysis
By: James Lim, Quang Nguyễn Đức, Neel Lal

("CLE" 13025.2)
("CHI" 12998.3))