**Project Title: Building prediction models and players analysis for NBA**

Author: Chun-Cheng Chang, Anthony Chu, Phyllis Chen

1. **Summary of Research Questions:**
   1. **How to efficiently visualize players’ shooting data:**

For most time, players’ shooting data is quite hard to understand since it is hard to plot x-y coordinate precisely by imagination. It’s beneficial to build a user-friendly program to generate clear plot of players’ shooting data.

* 1. **How salary of each player effects each team’s performance:**

Salary is an important indicator for the performance of a team; finding a way to see how salary can effect a team can help building up a model for predicting winning rate.

* 1. **Can we predict the winning rate of each team?:**

Most people are curious about each team’s winning rate, which include season and play-off winning rate. Using players’ data such as points, rebounds, assists rank within the league can allow us to predict the winning rate of each team.

1. **Motivation:**

Me and my friends are fascinated in NBA since we started to play basketball, and we often talk about which team might be the champion of the season. While there are lots of reporters that also make their own statistics, they only care about the news for some all-star players. Thus, we want to construct a program that can easily analyze and visualize all players’ data and predict their future performances.

1. **Dataset:**

We will mainly use three datasets:

* 1. **Nba-api | An API Client package to access the APIs for NBA.com:** [**https://github.com/swar/nba\_api**](https://github.com/swar/nba_api)
  2. **NBA Player Salaries | HoopsHype:** <https://hoopshype.com/salaries/players/>
  3. **bttmly/nba: Node.js client for nba.com API endpoints (github.com):**

json file for teams:

<https://raw.githubusercontent.com/bttmly/nba/master/data/teams.json>

Column ‘teamName’ refer to the teams’ names.

json file for players

<https://raw.githubusercontent.com/bttmly/nba/master/data/players.json>

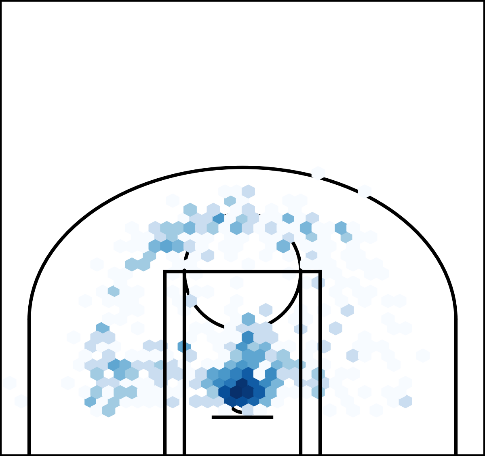
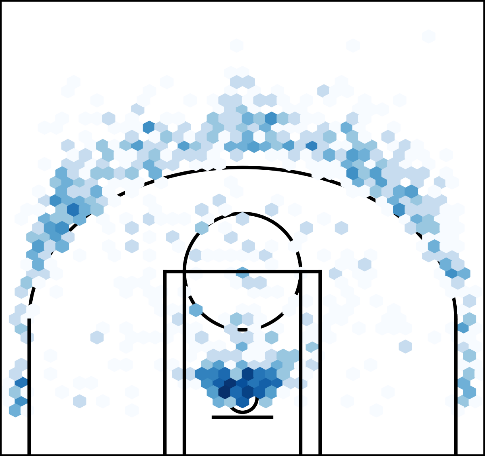
Column ‘firstName’, ‘lastName’ refer to the players’ names.

1. **Method:**
   1. **My environment:**

Using Anaconda to manage the package.

* + 1. Including NumPy, BeautifulSoup, pandas, matplotlib, requests, nba-api, SciPy, TensorFlow.
    2. Using git to manage version.
  1. **Data Preparation:**
     1. For each team’s past winning rate:
        1. Using endpoint LeagueDashTeamStats in nba\_api to access all teams’ record of winning percentage and other information.
        2. Using json and pandas modules to filter data until we get each team’s id, name, game played in a season, and winning rate.
        3. Save the filtered data into csv and category them by season.
     2. For each player’s data:
        1. Using endpoint LeagueDashPlayerStats in nba\_api to access each player’s data.
        2. Using json and pandas modules to clean the raw data.
        3. Save the prepared data into csv and category them by season.
     3. For each player’s salary:
        1. Using requests and BeautifulSoup modules to parse the HTML information on [NBA Player Salaries | HoopsHype](https://hoopshype.com/salaries/players/).
        2. Using pandas modules to transform dictionary-like data into csv file and category them by season.
  2. **Methods for Question A:**
     1. Use requests and json modules to obtain the data of each player and each team.
     2. Define a class of methods to quickly access data of player’s name, id and team’s name, id.
     3. Using endpoint ShotChartDetail in nba\_api to access target player’s shot location and the field goal made percentage.
     4. Generate easy-visualized map of players using matplotlib.
  3. **Methods for Question B:**
     1. Using all the data above and turn them to feature df and label df by pandas.
     2. Use keras.Sequential to build up the model.
     3. Save the best model based on the accuracy score of evaluating df.
  4. **Methods for Question C:**
     1. Using all the data above and turn them to feature df and label df by pandas and numpy.
     2. Use keras.Sequential to build up the model.
     3. Save the best model based on the accuracy score of evaluating df.

1. **Result**
   1. **Question A:**
      1. After using the plotting program we wrote, we now can easily recognize the precise position of where the target player makes a shot.
      2. We lessen the process of accessing target player’s shot chart by only required user to input player’s name, target season, and team name.
      3. The plot on left side is the field goal made chart for Stephen Curry, Golden State Warriors, 2021-22 season; the plot on right side is the field goal made chart for Tim Duncan, San Antonio Spurs, 2011-12 season. The darker color the mark be, the greater amount of field goal the player made.



* 1. **Question B:**
     1. After training, the neural network model we built has approximately greater than 50 percent of accuracy on predicting the salary level of each player.
     2. Feature we used for training: season, player age, team winning rate, total games played in season, real winning rate, minutes per game, points per game, rebounds per game, assists per game, steals per game, blocks per game, turnover per game, field goal percentage, 3-point field goal percentage, total plus minus
     3. Salary level: level 0 – annual salary below 1200000; level 1 – annual salary below 2000000; level 2 – annual salary below 5000000; level 3 – annual salary below 10000000; level 4 – annual salary below 20000000; level 5 – annual salary above 20000000
  2. **Question C:**
     1. After training, the neural network model we built has approximately greater than 33 percent of accuracy on predicting the winning rate of each team.
     2. For this question, we will focus on how to use players’ data to predict the result, but not using the overall team data to do so. With that not all players in a team have great influence on the standing, we only consider the top 12 influential players in the team based on the total minute played data.
     3. Since the players’ data value have great gaps between each other, we choose to use the ranking of each feature within the league to train our model.

1. **Impact and Limitation**

The target audience of our project is people who are interested in using players’ data to predict the game trajectory of entire season. People can use the shot chart visualizer we built to have a better idea on how each player makes shots on the real court; moreover, the models we trained can help people get a rough idea about the trend of a team performance and the salary level a player might get.

However, there are still lots of limitation about the result we got. First, for shot chart visualizing, we are only able to make a 2D plot that show the position of each shot, but we are not able to find data that show the real shooting height. Basketball is a sport that requires a good analysis of 3D space, while we are only eligible to visualize 2D plane.

Second, with that there are too many factors that can influence players’ salary and teams’ winning rate, it is really hard to choose the correct feature for training models for prediction. Also, there are some factors that we will never be able to include in our models, such as players’ injury. Although we still can get 50 percent accuracy on salary prediction, we could not know how each feature weight for evaluating a player ability. For winning rate prediction, the tactics of the coach and the team culture adaptability of each player may have great influences on team performance, but these data can’t be valued and predicted only by evaluating each player’s data.

For the usage of the result we obtained, I would only recommend people to use our result as a rough trend of possible future. Nevertheless, our data should not be used as a prediction machine for any commercial activity or gambling, while our data is not accurate enough to put in real application.

1. **Challenge Goal**
   1. **Multiple Datasets and Messy Data:**
      1. Using datasets from different sources to come out richer result; we use the api offer by the official website of NBA and some other datasets that cover the limits of nba\_api such as salary.
      2. Filtering out some useless data and combine different dataset’s information into appropriate form for faster and easier used in future.
   2. **Machine Learning:**
      1. Training models that allow people to predict each player’s salary by given information.
      2. Use players data to predict each teams’ winning rate.
   3. **New Library:**
      1. Using requests and BeautifulSoup module to get data from each website.
      2. Using TensorFlow to train models for salary and winning rate prediction.
   4. **Web Scraper:** 
      1. Learning the structure of HTML to obtain the correct data from website.
      2. Learning methods to obtain data from website, like BeautifulSoup module.
2. **Work Plan Evaluation**
3. **Testing**
   1. **Question B:**

We randomly split the data into 80% for training and 20% for evaluating the accuracy. Also, we have checked this model for 1000 times (approximately 15 hours) by keep feeding randomly split data and average the result accuracy.

* 1. **Question C:**

We randomly split the data into 80% for training and 20% for evaluating the accuracy. Also, we have checked this model for 500 times (approximately 19 hours) by keep feeding randomly split data and average the result accuracy.

1. **Collaboration**