**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’ data:**

For most time, players’ and teams’ data is quite hard to understand if all those data are provided in numbers. It’s beneficial to build a user-friendly interface to generate clear map of plays’ 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**
2. **Impact and Limitation**
3. **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.
4. **Work Plan Evaluation**
5. **Testing**
6. **Collaboration**