MAIS 202 Deliverable 1

Antoine Dangeard
Thomas Inman

Datasets:

- MLB Team stats 1903-2021, web scraped
    - Lots of different stats, can filter stats on the website, official stats
    - https://www.mlb.com/stats/team
- Archive of baseball game results (team names, date, and score) from 1871 to 2020
    - Huge amount of games to train the model on
    - Already formatted so it is easy to clean the data
    - https://www.retrosheet.org/gamelogs/index.html

Methodology:

a)

- The web scraper will get the most relevant team stats for a certain year, format them into a matrix where the columns represent the regular season teams and the rows will be that team's stats in the same order
    - If we have the time we will also try to implement individual player stats into our model
    - The MLB stats website is also well designed so finding stats for a certain year and team will not be too difficult
- The retrosheet data is already formatted, so cleaning and removing unimportant data is necessary.
    - We will have to clean the data because the stats for a team are only available for every year, and so a game too early in the season won't necessarily reflect those teams yearly stats
    - We will also not be training our model on games that occurred too long ago because we believe that the sport of baseball has evolved, and thus we would like our model to predict games from the past 20 years

b)

- We would like our model to be able to predict the score difference in a game between two teams based on their current stats. Ideally this could be applied to sports betting if our model is reliable enough.
- This would be achieved by assigning each stat (and its polynomials) a weight which will represent how that stat affects the outcome of a game. For this we would use polynomial regression and gradient descent to find the best weights, training the model on previous games and stats at the time. We considered using a classification model instead which would just predict a winner/draw, but the classification model does not allow us to predict a score difference (instead predicting winner/loser).

c)

- For our evaluation metric we will use mean squared error. We would also like our model to be able to guess the outcome of games with more accuracy than a random guess so we would like our model to have a success rate of more than 50%. Under the best circumstances our model would outperform the sports betting models (which is very difficult)

Application:

- To demo our model we will build a simple webapp which can take as inputs the names of two teams, the webapp will then web scrape the teams' current stats (or starting lineup), and predict the outcome of a game. If we end up implementing player stats into our model, then we could also make it possible to set a starting lineup which does influence the outcome. If player by player stats are too long to implement, just adding pitcher and batter stats could help.