

# *Creating a Rock Climbing Recommendation System*

## **Capstone 1 Proposal**

Kristen Colley

The sport of rock climbing has been steadily increasing in popularity. From 2012-2017, the IBISWorld estimates that from average annual growth for the indoor climbing wall industry was [3.9% in the USA](#). In [2015, it ranked 17th out of 111](#) out of the most popular sports in the United States. ( [Physical Activity Council](#) and PHIT America).

Yet, even with this growth in popularity, there still lacks an official rock climbing recommendation system. As it stands, there exist a few main climbing mobile apps and websites that allow the user to look up a rock climbing route that provides various descriptive information such as location, user ratings, pictures, GPS, ect. However, none of these platforms have created a prediction of what the user would like to climb next. I will be taking data from the largest rock climbing website, and using it to predict the users next rock climbs.

### **Audience**

Although the climbing community is still a small subset of the population in general, they are a passionate group who invest a lot of their time climbing and researching future climbs. This recommendation system would be a good addition to the community to give climbers a quick reference of what they would like to climb next.

Also, any of the climbing apps and websites would reap the rewards of being the first to be able to provide a rock route prediction system for their users.

### **Data**

For the data, I choose a Kaggle data set scraped from the world's largest database of rock routes, the website: [8a.nu](https://www.kaggle.com/dcohen21/8anu-climbing-logbook). With over 4 million entries of climbs and ratings, I believe this will be a sufficient size to develop a good predictor model.

- a. <https://www.kaggle.com/dcohen21/8anu-climbing-logbook>

## Method

1. The biggest decision will be choosing between the three main types of recommendation systems: collaborative, content, or a hybrid system. Collaborative could be the best approach if my dataset proves to be sparse; However, a hybrid system tends to have the most accurate recommendations.
2. Steps:
  - a. Determine how sparse our data set is
  - b. Look at the distribution of user ratings
  - c. Fit the model
    - i. This training model will take more RAM than my computer currently has so I will look into using Google Colaboratory, or AWS EC2 instances
  - d. Dealing with the "cold start problem"
    - i. i.e. a new user with no background information

## Deliverables

- Blog post
- Flask website
- YouTube video
- Code on github