Milestone Report

Creating a Rock Climbing Recommendation System

Capstone 1 Project

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 one of the largest rock climbing websites, and using it to predict what the users will want to climb next.

Audience

Although the climbing community is still a small subset of the population, 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 able to provide a rock route prediction system for their users.

Data Source

For the data, I choose a Kaggle data set scraped from the website: 8a.nu, one the world's largest database of rock routes with particular attention to the international climbing community. With over 4 million entries of climbs and ratings, this is a sufficient size to develop a good predictor model. To view the original Kaggle data set, click the link below:

Kaggle Dataset

In order to upload this large dataset, I utilized the Kaggle API in a google colab notebook in order to combat the RAM issue of my personal computer. To view my report on converting the data from an SQLite Kaggle dataset to a pandas dataframe click below:

Data Import Report

Data Wrangling

All of this data is online user-entered and prone to discrepancies which made it a particularly challenging dataset to clean. Also, I ended up cleaning the data in two different ways to test out what would work best with the recommendation system:

- Main Cleaning Report
- 1x_route_name_filter

There were 4 different tables in this dataset. Below are general descriptions of the four tables:

- Ascent Table
    - 4 million user entered rock climbing entries

- User Table
    - 67,000 user profiles of climbers that log their climbs
- Grade Table
    - A reference table that translates the rock climbing grade to French or American rock ratings
- Methods Table
    - Shows a reference of the way in which the climber finished the climb (i.e. did they fall, take, or climb is cleanly)
    - I was able to eliminate all together because this was completely irrelevant to the recommendation system.

In order to properly clean these tables, I needed to delete a lot of extraneous information that would not serve the recommendation system. Essentially, I only needed three columns to feed to recommendation system. However, I needed to clean several more to serve as a "reference table" so my recommendation system can be filtered based on area and type of climb. Below is an overview of my wrangling of the three main columns:

- User_id
    - Straight forward, no duplicates, a perfect match up to the user table
- Climb_name
    - User entered individual rock route names
    - **Problem:**
        - Routes were spelled incorrectly, or spelled multiple different ways (EXAMPLE: "red rocks canyon", "redrocks canyon", "redrock canyon", "redrock canyons")
        - This is what my recommendation system will return to the user so it's very important to clean this column correctly
    - **Solution:**
        - <u>Normalizing the route name column:</u>

- Make all characters lowercase
- Exchange “&” sign for and
- Get rid of all spaces in the names, and special characters (, . - ! ‘ ?)
- Take away all accent marks (because there are lots of foreign language names)
- Create a REGEX expression that only accepts letters and numbers
- Filter out phrases such as (“I dont know”, “noname”, “none”)

- Tackling the spelling problem:
  - Phonetic algorithms: I tried two different types of phonetic grouping algorithms that would group the names based on if they sound similar
    - The double metaphone algorithm was the more successful avenue, however both the soundex and the double metaphone grouped the names too aggressively. These algorithms were sometimes grouping up to 10 different route names together as the same location. Thus, I was not able to use either of these.
  - Filtering out names: Since the phonetic grouping didn’t work, I looked at the distribution & frequency of unique names, and based on this, was able to see that around 10 mentions resulted in an accurate rock climb location.
    - Thus, I filtered out any rock route name that occurred less than 10 times

        - **I also created a second cleaned DF that only filtered out names that occurred < 1 time to compare and contrast the effectiveness of the recommendation system
    - Encoding the names: finally, I used Hash64 to encode the names to use with the surprise library recommendation system, and will subsequently use this to decode them
- Rating (1-3 stars)
    - There were a few 4 star ratings that I had to filter out because according to their website the highest rating a user can rate a climb is to give it 3 stars

Exploratory Data Analysis

In the EDA, I was able to identify that the dataset will be sufficient for the recommendation system.

Click here for the detailed EDA report of the main dataframe
Click here for the EDA report of the 1x_name_filter dataframe

Below are a couple pertinent findings:

- Star ratings:
    - The ratings of routes had an expected distribution of 1-3 stars. Where there were less one star ratings than two or three star (because people tend to negatively rate things less).
- Users providing ratings:
    - The most ratings a single user provided was 2,088 ratings
    - The majority of users (mode) only provided 1 rating
    - The average number of ratings per user is 50

    - 243 users gave more than 500 ratings
- Routes that were given ratings
    - The most ratings a single route received was 687 ratings
    - The lowest amount was only 1 rating for a route
    - The average number of ratings per route was 30
    - The mode was 10 ratings per route with a total of 3,622 routes with exactly ten ratings
    - Only 353 routes have more than 200 ratings (thus this was excluded from the graph)

Hypothesis Testing

Even though it does not pertain the recommendation system itself, I wanted to explore two intriguing questions about the climbing data:

1. ***Do females climb the same grades as males now?***
    a. They have been pushing the envelope in climbing since the 90's, and even though this is traditionally a male dominated and developed sport in the past, I was curious if this has changed.
    b. Female versus Male Climbing Grades Hypothesis Test
    c. Results: According to this dataset, females are climbing on average two grades lower than males. The hypothesis test showed this difference is not due to chance.
2. ***Do taller people climb higher grades than shorter people?***
    a. There is a rock climbing stereotype that if you are taller, you are naturally better at climbing. With the height data, I wanted to test this presumption.
    b. Tall vs. Short Climbing Grades Hypothesis Testing

c. **Results:** According to this dataset, “short” people are climbing on average two grades higher than “tall” climbers and one grade better than “average” height climbers. The hypothesis test showed this difference is not due to chance. ***Thus, according to this dataset, shorter climbers are better at climbing!***

Method- Recommendation System

1. The biggest decision was choosing between the three main types of recommendation systems: collaborative, content, or a hybrid system. I will be using a user-based collaborative rating system because it will be the best option given my dataset.
2. Steps:
    a. Fit the model
        i. This training model will take more RAM than my computer currently has so I will use Google Colaboratory,
        ii. I will be using the surprise library recommendation system and start with the SVD algorithm to get recommendation system working
        iii. After that, I will play around with different algorithms to see which gives me the best output
    b. Dealing with the “cold start problem”
        i. i.e. a new user with no background information

Deliverables

- Colab notebook recommendation system
- Report
- You tube video or blog post
- Code on github