Rock Climbing Data Collection, Analysis, and Recommendation System
Craig Fox

Introduction

As the COVID pandemic has raged on, more and more people have turned to outdoor activities to occupy their free time. One activity that has been gaining steam is rock climbing. Even before the pandemic, factors such as the popularity of the Oscar-winning documentary "Free Solo" about Alex Honnold's historic ascent of El Capitan and the introduction of climbing to the 2020 Olympics have helped to propel climbing into the spotlight. In 2018, 7.7 million people participated in climbing contributing over 12 billion dollars to the economy [1]. Despite this growth, the climbing community still predominantly relies on small, expensive guidebooks and word of mouth to discover outdoor climbing locations that best suit them. Many newcomers can struggle to find places to climb without knowing the right people or combing through a large number of expensive guide books.

However Mountain Project [2], a free website of user posted climbs, has been a rapidly growing exception to this. Mountain Project provides free route data that users can find new climbing locations from. The problem is that there is no way to tell how well a user might like a specific route or area. Since these locations can be difficult to access and require time to set up climbing gear, it would be helpful for someone to have an estimate of how much they would enjoy the route. Other attempts have been made to make a rock climbing recommender but they are either outdated [3], limited to a small area [4], only provide recommendations based on one similar route [5], or do not take into account user's enjoyment of other routes at all [6]. Additionally many of these projects rely on a now deprecated API making them functionally useless. One website provides a publicly accessible API as well a recommendation system, but their recommendation system is also limited to a small region, they are facing DMCA issues, and their goal seems to be predominantly computer science education instead of climbing assistance [7]. Additionally few attempts have been made to analyze this data to understand more about outdoor rock climbers and routes.

I will be developing a web scraper to collect the climbing data from Mountain Project. I will then use this data to discover some basic insights into climbing routes. I will create a general rating prediction system using linear regression. Finally I will develop a recommendation system for specific individuals to provide them with personalized recommendations for routes.

Approach

My idea would be to develop a new climbing recommendation system. Based on a user's past ratings of routes, ratings for all other routes would be generated. To get my initial data, I will write a web scraper that will gather route data and user rating data from Mountain Project. The data I will collect for each route is name, description, difficulty, star rating, number of ratings, page views, whether it is trad climbing, whether it is sport climbing, whether it is top rope climbing, whether it is alpine climbing, whether it is aid climbing, whether it is bouldering, whether it is snow climbing, whether it is ice climbing, whether it is mixed climbing, length, pitches (number of sections), and commitment grade (how much time needed). I will perform some exploratory analysis of this data to find useful general information about it. I will then normalize the data. Then I will perform linear and ridge regression to try to predict the general route rating using only the attributes of the climb. The target would be to match the actual rating that Mountain Project uses to rank climbs. Afterwards I will analyze the data to see which attributes of the data impacted the recommendation system the most. I will then develop a

recommendation system to predict how much users will like a specific climb based on their climbing history.

Results

For the first part of my project, I needed to collect the route data. Since Mountain Project does not maintain a public API anymore, I decided to develop a web scraper. The primary data I needed was route information. All of the routes on Mountain Project follow a URL format of

https://www.mountainproject.com/route/[**ROUTE NUMBER]/[ROUTE NAME]**

Luckily I discovered this could be shortened to just

https://www.mountainproject.com/route/[**ROUTE NUMBER]**

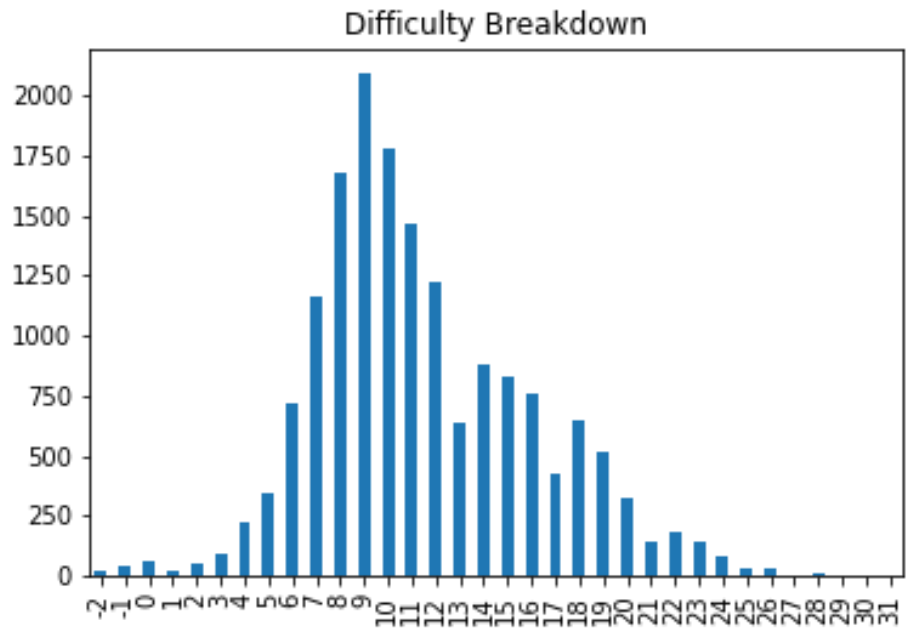
and you would get redirected to the full length url. The route number is a 9 digit number that seems to be based on when the route was added, although many numbers are missing for unknown reasons.

My initial plan was to bruteforce go to every URL possible and see if it contained a route. After trying this out, I realized how slow and inefficient it would be. I decided to look for another way. I ended up finding that the sitemap for the website contained a list of 100 sitemaps which could be downloaded and which contained links to all the routes available. So I downloaded the sitemaps and wrote a web scraper that scans through them, finds the route ids, and then goes to those sites. Without doing this method, it would have taken roughly 4,000 times longer to collect the data. Even so, there were so many routes I had to collect only 25,000 out of over 250,000 due to bandwidth limitations. I decided to go with the first 25,000 ids because these are the climbs that have been on the site the longest so they probably have the most complete and stable data.

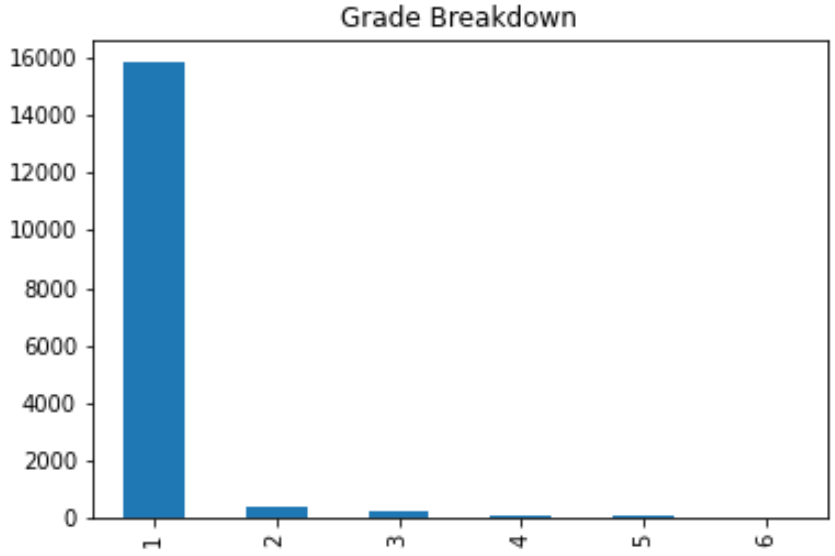
Next I converted the data to a more usable form. The main change was trying to standardize the difficulty ratings. Yosemite Decimal System (YDS) is a difficulty rating system commonly used in the United States to rate the difficulty of paths. YDS rates everything from hiking trails to rock climbing with 1 being a flat trail and 5 being a climbing route. Within 5 it goes 5.1-5.9, 5.10a-d, 5.11a-d, …, 5.15a-d. Additionally a climb can have a plus or minus or could be between two ratings. I changed this to a linear integer scale. 5.0 became 0, 5.1 became 1, …, 5.10a became 10, 5.10b because 11, … There were a few 4s and 3s, so 4 became -1 and 3 became -2. If it was in between two ratings, the lower rating was always used. Pluses and minuses were dropped for anything below 5.10a. For anything above, pluses became 5.xc and minuses became 5.xb. Ultimately this created a linear scale from -2 to 31 (the highest climb was a 5.15b). Additionally I filtered out all non YDS scale routes. However multiple other systems exist and there is no clear conversion between them. Therefore I decided to focus on YDS climbs for this project. This left 16,641 routes. I also cut out some data that I had initially planned on collecting such as comments, FA (first ascent), photos, shared by date, access issues, protection, and safety because they were too inconsistent in formating, did not seem to be of any use, or were rarely present in the data.

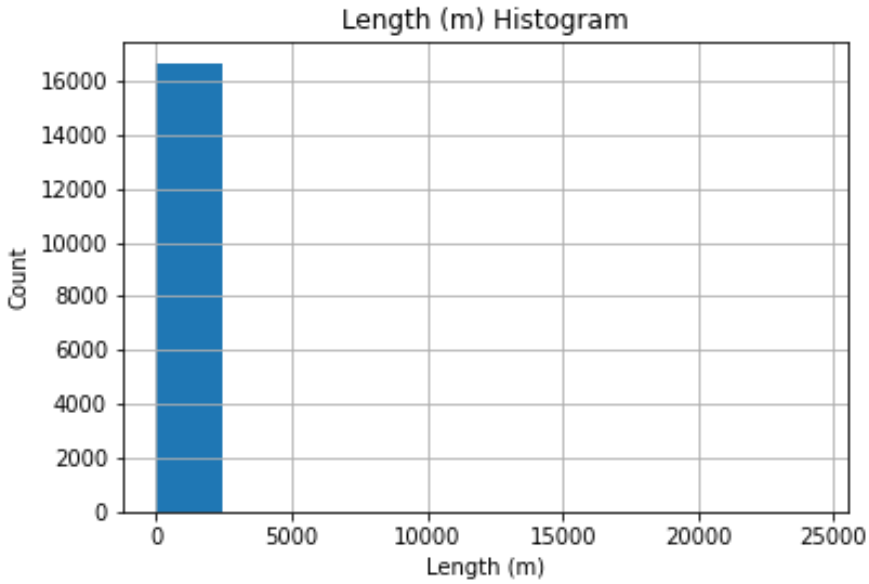
Next I performed some exploratory data analysis. All of it was similar to what I expected. First I analyzed the difficulty breakdown and found that it closely followed a bell curve. This

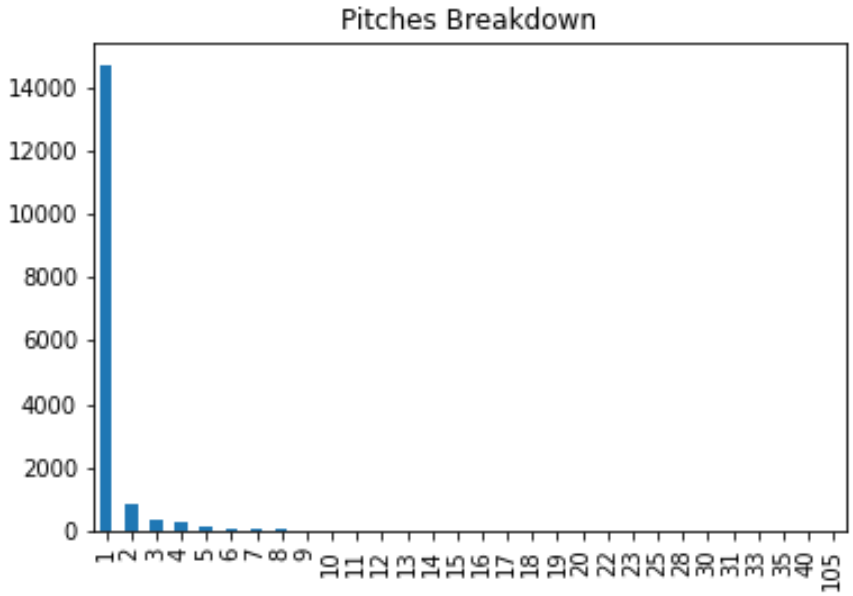
makes sense because climbers will tend to try to find climbs close to their ability and the ability levels of people tend to follow a bell curve.



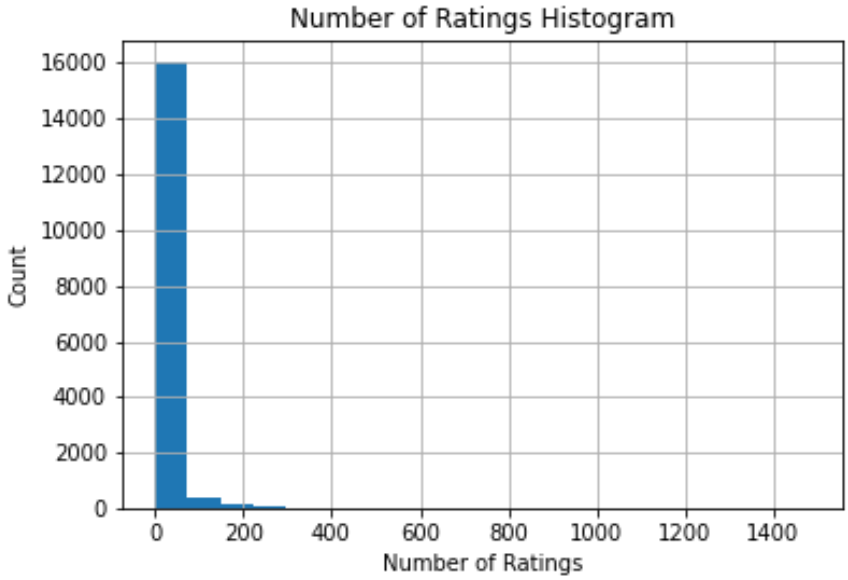
Almost all of the grades were I which means they take a couple of hours. Most people are not doing multi-day trips, and even those that are do not do them as frequently as shorter climbs, so again this is not surprising. The length and pitches graphs matched this same trend for the same reason.

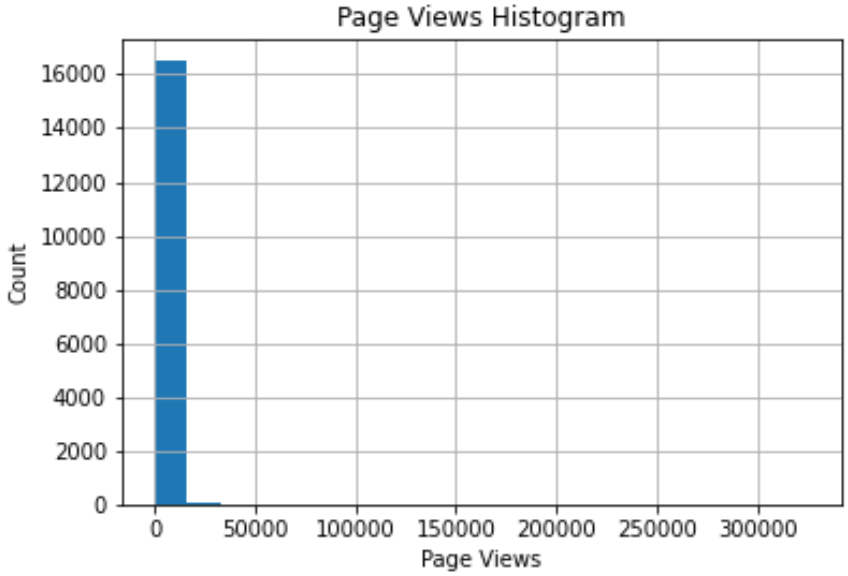




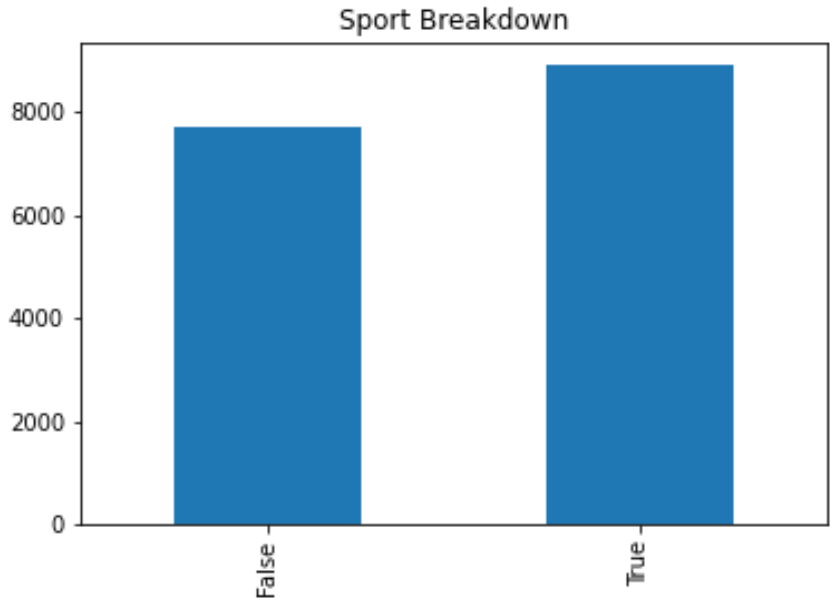


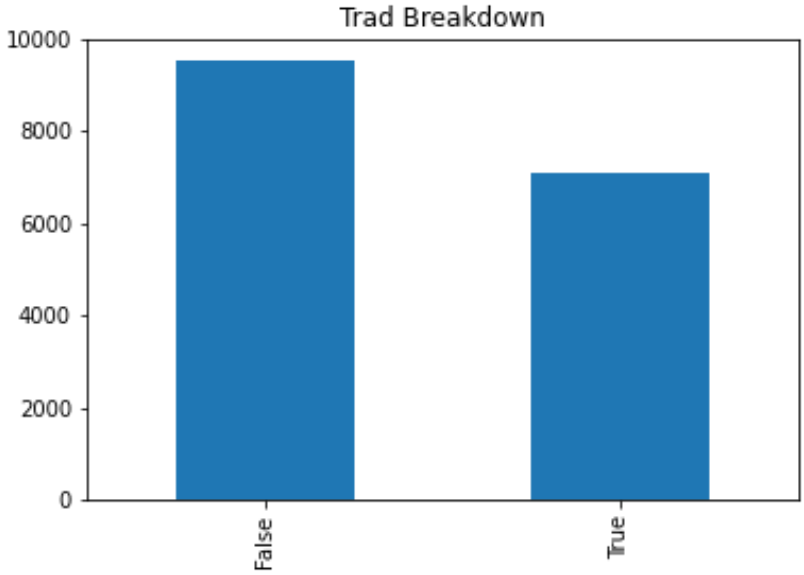
Like the three previous graphs, the numbers of ratings and page views graphs both exhibit extreme long tail distributions. These incredibly popular routes at the end of the trail get at least 100 times more attention then the average route by both metric.

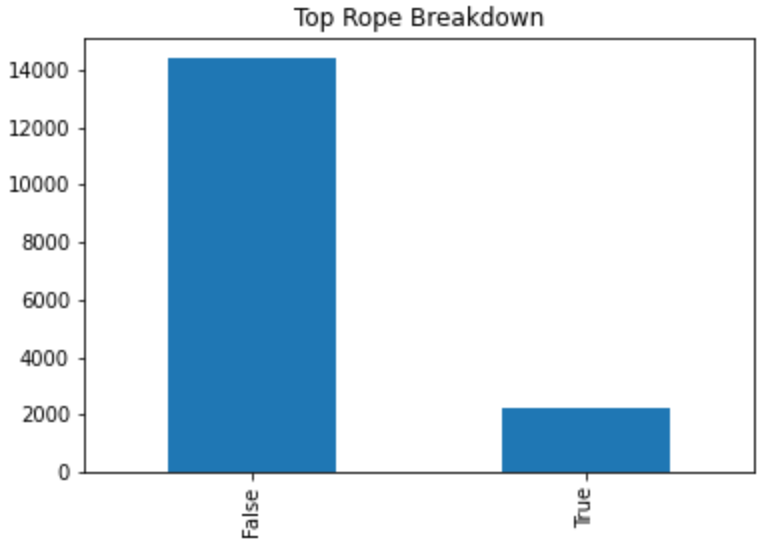




Within types of climbing, the results were again unsurprising to anyone familiar with climbing. Roughly half of climbs are trad and roughly half are sport. Just over ten percent are top rope. The rest of the categories make up less than one percent. While these categories are not mutually exclusive, they very rarely overlap.

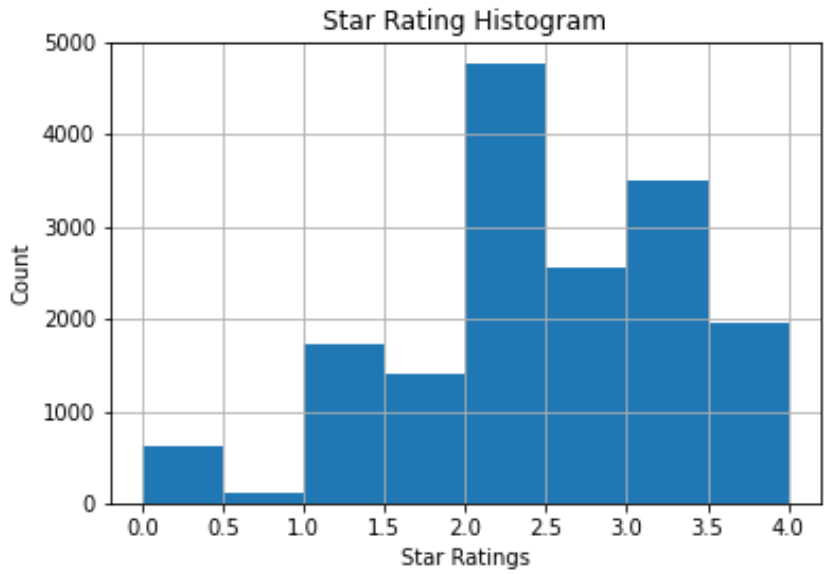

Sport Breakdown
8000
6000
4000
2000
0
False
True


Trad Breakdown
10000
8000
6000
4000
2000
0
False
True




```
False    16022
True       619
Name: Alpine, dtype: int64
False    16509
True       132
Name: Aid, dtype: int64
False    16539
True       102
Name: Boulder, dtype: int64
False    16605
True        36
Name: Snow, dtype: int64
False    16621
True        20
Name: Ice, dtype: int64
False    16629
True        12
Name: Mixed, dtype: int64
```

The star rating distribution is fairly balanced, although there are more ranking in the middle and less at the bottom



I looked at the correlation between these variables. Most of it was to be expected such as sport, trad, and top rope are all inversely correlated with each other, number of ratings and page views are strongly positively correlated. The interesting result was a correlation between star rating and difficulty of .35. This suggests that climbers like harder routes more.

	Difficulty	Star Rating	Number of Ratings	Page Views	Trad	Sport	Top Rope	Alpine	Aid	Boulder	Snow	Ice	Mixed	Length (m)	Pitches	Grade
Difficulty	1.00	0.35	-0.06	-0.02	-0.33	0.38	-0.17	-0.09	-0.03	-0.01	-0.09	-0.04	-0.02	-0.01	-0.04	-0.06
Star Rating	0.35	1.00	0.14	0.15	-0.06	0.11	-0.11	0.03	-0.00	0.02	0.02	0.02	0.01	0.03	0.09	0.08
Number of Ratings	-0.06	0.14	1.00	0.69	0.01	0.01	-0.05	-0.02	0.01	-0.00	-0.01	-0.01	-0.01	0.01	0.06	0.02
Page Views	-0.02	0.15	0.69	1.00	0.08	-0.06	-0.04	0.05	0.09	0.00	0.00	0.01	0.00	0.05	0.23	0.16
Trad	-0.33	-0.06	0.01	0.08	1.00	-0.85	-0.02	0.12	0.08	-0.02	0.04	0.03	0.02	0.05	0.15	0.14
Sport	0.38	0.11	0.01	-0.06	-0.85	1.00	-0.26	-0.11	-0.09	-0.07	-0.05	-0.04	-0.02	-0.04	-0.13	-0.13
Top Rope	-0.17	-0.11	-0.05	-0.04	-0.02	-0.26	1.00	-0.05	-0.02	0.08	-0.02	-0.01	-0.00	-0.03	-0.08	-0.05
Alpine	-0.09	0.03	-0.02	0.05	0.12	-0.11	-0.05	1.00	0.03	-0.01	0.20	0.10	0.08	0.10	0.24	0.28
Aid	-0.03	-0.00	0.01	0.09	0.08	-0.09	-0.02	0.03	1.00	0.00	0.02	0.04	0.05	0.05	0.20	0.27
Boulder	-0.01	0.02	-0.00	0.00	-0.02	-0.07	0.08	-0.01	0.00	1.00	0.01	0.02	0.03	-0.01	-0.02	-0.01
Snow	-0.09	0.02	-0.01	0.00	0.04	-0.05	-0.02	0.20	0.02	0.01	1.00	0.33	0.19	0.07	0.11	0.12
Ice	-0.04	0.02	-0.01	0.01	0.03	-0.04	-0.01	0.10	0.04	0.02	0.33	1.00	0.65	0.05	0.08	0.07
Mixed	-0.02	0.01	-0.01	0.00	0.02	-0.02	-0.00	0.08	0.05	0.03	0.19	0.65	1.00	0.02	0.04	0.04
Length (m)	-0.01	0.03	0.01	0.05	0.05	-0.04	-0.03	0.10	0.05	-0.01	0.07	0.05	0.02	1.00	0.18	0.15
Pitches	-0.04	0.09	0.06	0.23	0.15	-0.13	-0.08	0.24	0.20	-0.02	0.11	0.08	0.04	0.18	1.00	0.53
Grade	-0.06	0.08	0.02	0.16	0.14	-0.13	-0.05	0.28	0.27	-0.01	0.12	0.07	0.04	0.15	0.53	1.00

Next I normalized the data and performed a linear regression and a ridge regression with a λ of 1. The results were similar but linear was slightly better. Since there is so much data, there is not as much of an issue of overfitting. The results were good. On a 0 to 4 scale, on average the rating deviated from the actual by 0.65.

Linear MAE: 0.6473237744568865
Linear MSE: 0.7205095090640216
Linear RMSE: 0.8488283154231022

Ridge MAE: 0.6473250969520485

Ridge MSE: 0.7205103534348337
Ridge RMSE: 0.8488288127972764

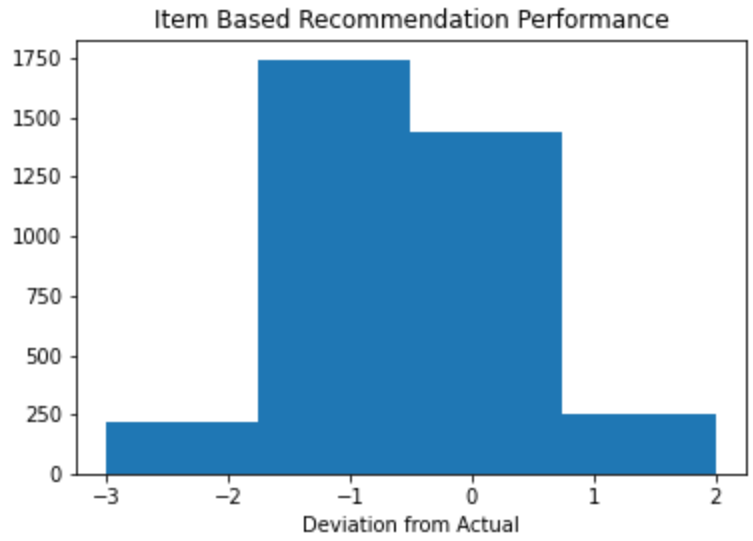
Next I collected the individual star ratings to begin work on a recommender system. They are contained at a different URL called

https://www.mountainproject.com/route/stats/[**ROUTE NUMBER]/[ROUTE NAME]**

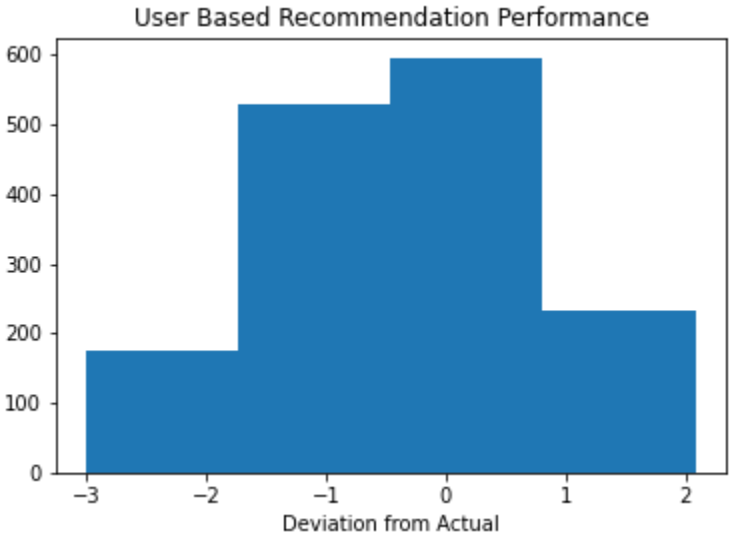
that has the clickable list of all the star ratings, suggested difficulty ratings, ticks, and on-to lists. The star ratings are in an attribute called “scoreStars” that contains 1-4 images of stars or an image of a bomb (to represent zero) based on a user’s rating right next to an attribute containing the user’s name. I went back and scraped these without an issue. I decided to use 1000 rows. This time the issue was not bandwidth, but memory and processing. The first 100 routes had 3129 users, the first 215 routes had 5293 users, and the first 1000 routes had 9436 users. Since users are generally tied to a specific geographic region, I decided this would not be an issue because most of the users would be visiting similar climbs. However the matrix turned out to be sparser than I predicted. There were just under 20,000 ratings out of 9436000 possible ones which means a density of less than 0.00212.

I decided to use cosine correlation and both an item-based and a user-based recommendation system using the 10 nearest neighbors. The item-based recommendation system had good results, but the user-based did not. My prediction of the geographic nature of climbing and the sparse network making it hard to do user based prediction proved correct. Over half of the test set, the user-based recommendation could not even provide a recommendation because none of the ten closest neighbors had done the route. Even when it did have a recommendation it is noticeably worse then the item-based prediction. For this reason, I decided to remove the user-based prediction. As can be seen in the graphs below, when the prediction was incorrect, it tended to underestimate the actual rating. I am not sure what could cause this but it definitely warrants further investigation.

Item MAE: 0.6703095624355488
Item MSE: 0.8000268358343339
Item RMSE: 0.8944421925615618
Item Number Could Estimate: 3655
Item Number Could Not Estimate: 191



User MAE: 0.7454792973882082
User MSE: 1.0089451387681025
User RMSE: 1.004462611931426
User Number Could Estimate: 1531
User Number Could Not Estimate: 2361



Conclusion

Overall the project was a success. I was able to collect the data from Mountain Project and found some interesting insights in it as well as build a general rating prediction and a user specific rating predictor.

There is still lots of room for improvement. First, although I do not expect hugely different results, I would like to try various λ for the ridge regression in a more systematic and scientific way. Next I would like to try to make the nearest neighbors of the routes based on the general

data collected and see how that impacts results. I would also like to expand to other difficulty ratings systems. Additionally I could work on improving the performance of the program to allow more routes to be included. I could also collect location data and analyze that as well as make recommendations for good areas for someone to visit based on how they would rate its routes. This would probably be more accurate as long as the user had a few ratings in various locations to start since the ratings are so geographically tied. The owners of Mountain Project maintain four other similar websites for hiking, trail running, mountain biking, and skiing that I could also analyze. As auxiliary tasks, it would be nice to automate the sitemap download and to use k fold split cross validation on the linear regression.

Acknowledgements

Thank you to Mountain Project for having this data publicly available and accessible. Thank you to Stack Overflow for assistance on many questions. Thank you to the numerous web guides I referenced. Thank you to python, requests, BeautifulSoup, pandas, and sklearn for their wonderful documentation. Thank you to Jupyter Notebook for the development platform.

References

[1] The American Alpine Club, “State of Climbing Report”, 2019
[2] Mountain Project, https://www.mountainproject.com/
[3] David Blaszka, “RedPointer”, https://github.com/davidblaszka/redpointer
[4] Viet Nguyen, “Building a Climbing Route Recommendation Engine”, https://openbeta.substack.com/p/building-a-climbing-route-recommendation
[5] Gabriel Seemann, “Rock Climbing Recommendation Engine”, https://medium.com/@gabrielseemann/rock-climbing-recommendation-engine-9aada4c5f3c6
[6] Andrew Kelleher, “Climbing Suggestions”, https://climbingsuggestions.com/
[7] Viet Nguyen, “Open Beta”, https://openbeta.io/