



Building a Race Recommender for RaceRaves Users

DSI-WEST Capstone Project

Dwayne Jarrell

April 6, 2018

Building a Race Recommender - Overview

Problem Statement

- Runners are always looking for new challenges and setting new goals for themselves, but it's not always easy to find that next race that will help you make that goal
 - Information available online is scattered and inconsistent
- RaceRaves.com does a great job of collating race information and collecting reviews from racers, and they currently have a very useful Find a Race feature
 - Current tool gives you a list of races based on criteria you select, including Distance, Terrain, Geography and Date
 - Results can be sorted by Date, Overall Rating or Alphabetical
 - Racers don't currently have the option to input other factors (e.g., flat course, scenic route), and it would be a challenge for RaceRaves to determine all of the factors that matter to racers

Find a Race

[Reset](#)

▼ **NAME / KEYWORD**

☐ Upcoming races only

▼ **DISTANCE**

☐ 5K

☐ 10K

☐ 10 Miler

☐ Half Marathon

☐ Marathon

☐ 50K

[See more](#)

▶ **TERRAIN / TYPE**

▶ **COUNTRY / STATE**

▶ **ZIP CODE**

▶ **DATE**

FIND A RACE

Building a Race Recommender - Overview

Proposed Solution

- Use Natural Language Processing on reviews posted to [RaceRaves.com](https://www.raceraves.com) to match racers to races they are likely to enjoy running
 - Primary hypothesis is that users will mention aspects of races that matter most to them in their reviews
 - At the race level, reviews should combine to form patterns from common themes
- Once racers are matched to races, other data from the site can be used to filter results
 - Racer profiles provide a rich amount of data on past races, future races, goals and preferences
 - The recommender could be used to rank races that match the filter and would also have the highest appeal to the racer



Sample Review:

Lake Powell Half Marathon (Vacation Races)

awesome race

Jan 21, 2016 | Half Marathon | First-timer '15

This race is amazing. Road and trail running combined with beautiful views. Some challenging hills but you cross the dam and have some amazing views. Looking forward to doing it again.

DIFFICULTY		PRODUCTION	
SCENERY		SWAG	

Was this review helpful? **YES!**

Building a Race Recommender - Overview

Data

- All data was collected by scraping the RaceRaves website using Python and BeautifulSoup
 - Initially, race pages were scraped using monthly selections from the Find a Race tool, and racer IDs were collected from the reviews
 - User profiles and reviews were then scraped from the individual racer pages
 - Final counts were as follows:
 - Unique Racers = 2,009
 - Unique Races = 2,103
 - Total Reviews = 6,414

Methodology

- All modeling was done using Natural Language Processing and Machine Learning tools from the Python scikit-learn module
 - Primary methods include CountVectorizer, LatentDirichletAllocation and KMeans clustering, each explained in more detail throughout

Building a Race Recommender - The Data

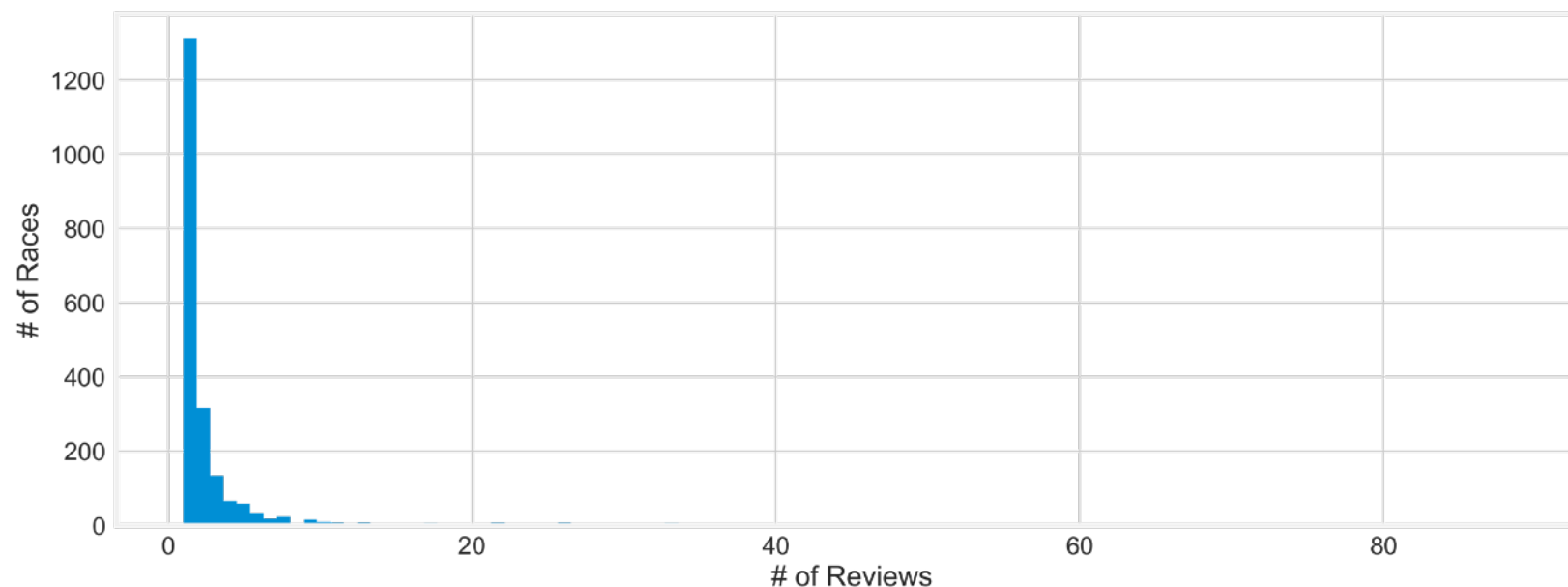
Reviews

The average # of reviews per user was 3.2, but the distribution is highly skewed towards single reviews at both the racer and race levels

Count of Racers by Number of Reviews Submitted



Count of Races by Number of Reviews Submitted

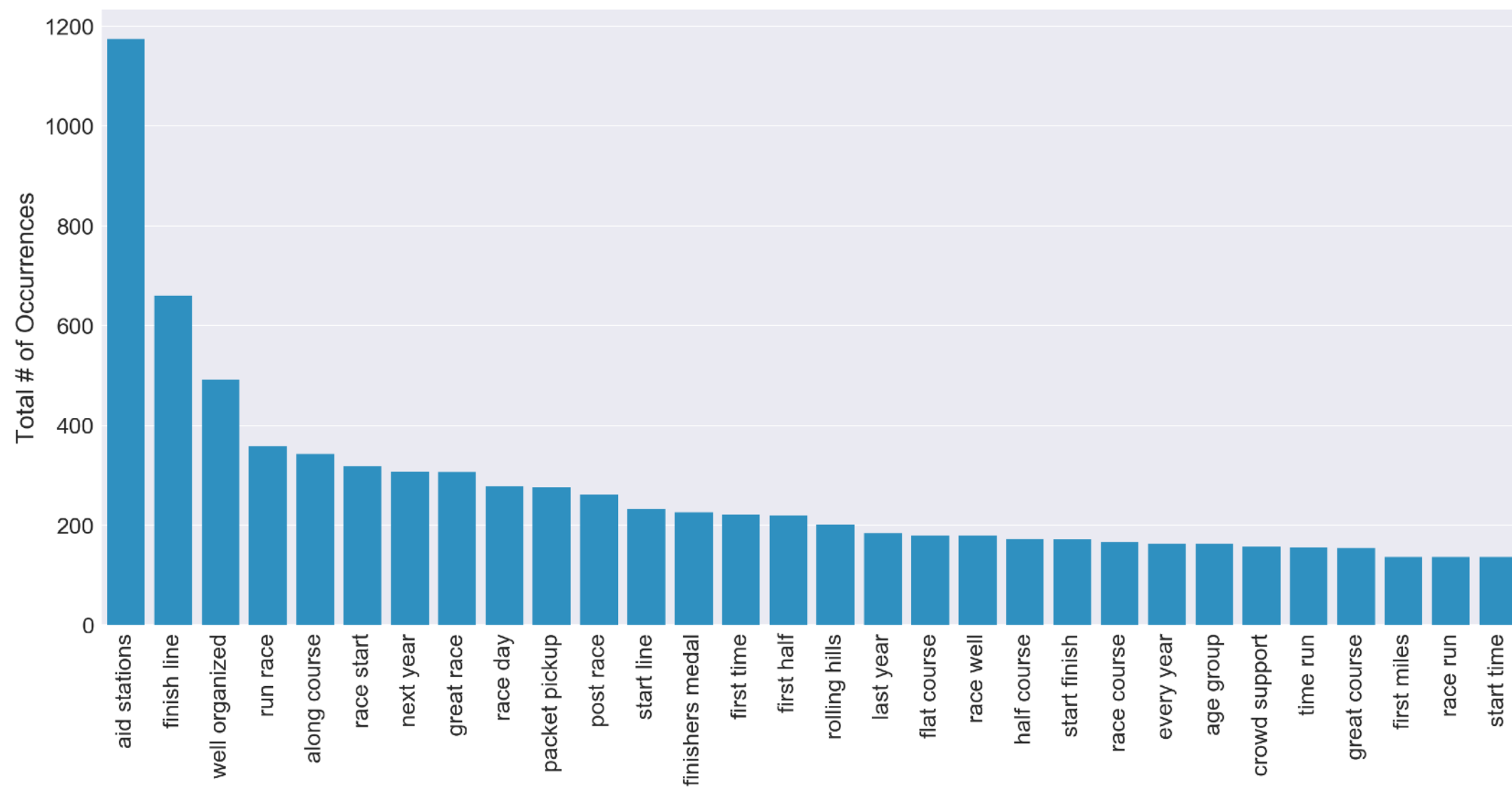


Building a Race Recommender - Word Counts

The first step in NLP is identifying and counting the most frequently occurring words and/or phrases occurring in the reviews

- Initial analysis indicated that two-word phrases (“bi-grams”) were the best at differentiating reviews
- The final selection involved several methods of removing non-critical words
 - Methods included eliminating stop words (e.g., ‘the’, ‘and’, ‘a’) and setting a minimum frequency and maximum occurrence across reviews (this last step eliminated common terms like ‘half marathon’)

Top 30 Bi-Grams from All Reviews



Building a Race Recommender - Finding Topics

Next, I used Latent Dirichlet Allocation (LDA) to assign topics across all reviews

- LDA uses frequency of occurrence to group similar terms and assign a probability that each of N topics is included in a given review
- The model was optimized at 30 topics - below are the most common bi-grams by topic*



* "Topic" is the standard term used to describe the grouping of words from LDA, but for our reviews we could also think of them as themes

Building a Race Recommender - Matching

Model Training

- To train the recommender, I randomly chose 80% of racers and ran the count vectorizer and LDA process on all of their reviews individually (regardless of racer or race)
- There were 1,607 racers in the training data, with a combined 5,057 reviews of 1,768 races
- Once the model was trained using all reviews, I applied the model to racers and races separately to facilitate matching

Racer Scoring

- The 5,057 reviews were rolled up at the RaceRaves User ID level to create one “bag” of words per user
- The fitted LDA model was used to assign topics to each user based on all of the words in their bag
- Each user was assigned probabilities for all 30 topics in the LDA model

Race Scoring

- The 5,057 reviews were also rolled up at the Race Name level to create one “bag” of words per race
- The fitted LDA model was used to assign topics to each race based on all of the words in its bag
- Each race was assigned probabilities for all 30 topics in the LDA model

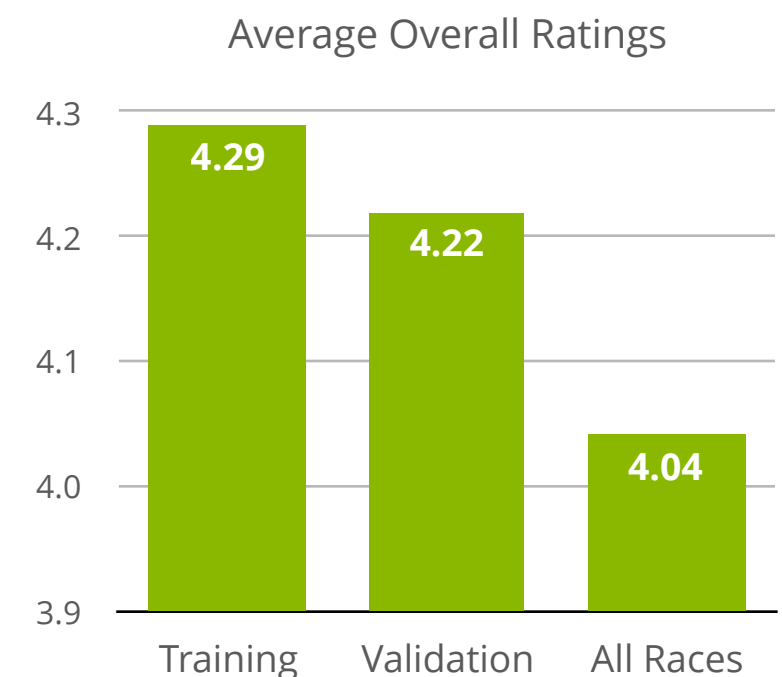
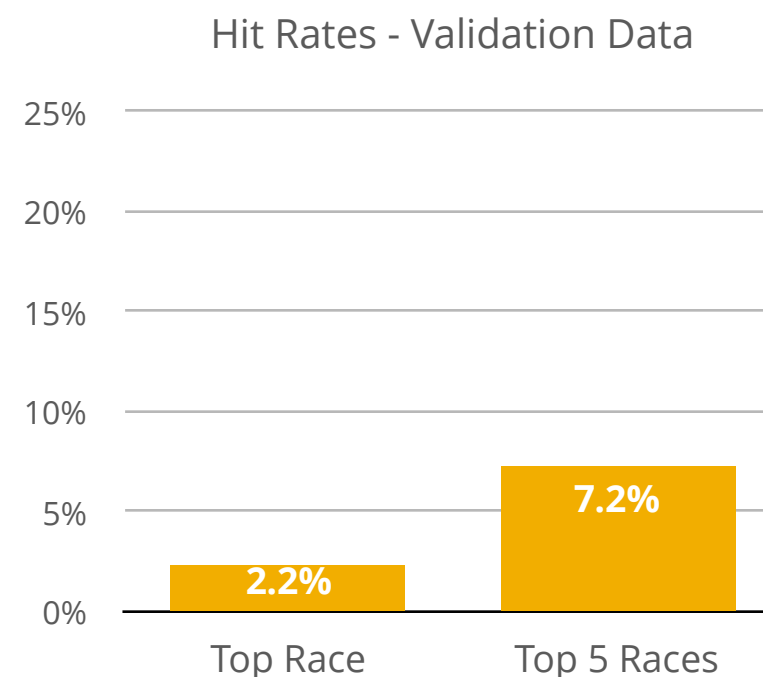
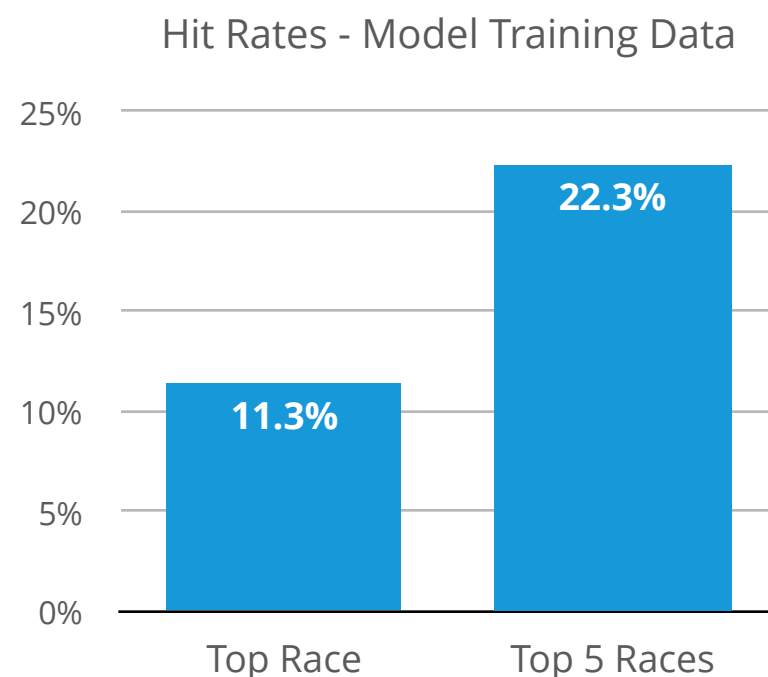
Matching Racers & Races

- Euclidean Distance was used to measure the difference in topic probabilities between each racer and the 1,768 races available to them in the training dataset
 - Euclidean Distance is the straight-line distance calculated using the Pythagorean Theorem across all 30 topic probabilities
- Races were ranked for each racer based on the shortest “distance” between racer and race

Building a Race Recommender - Evaluation

Hit Rates

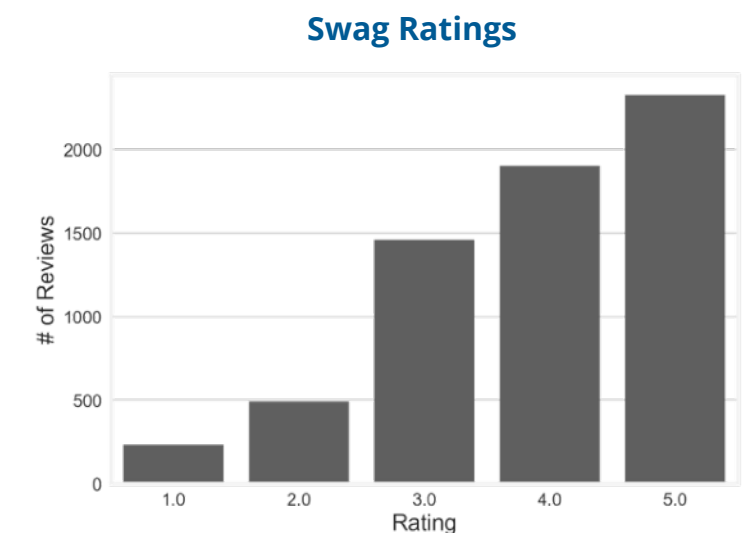
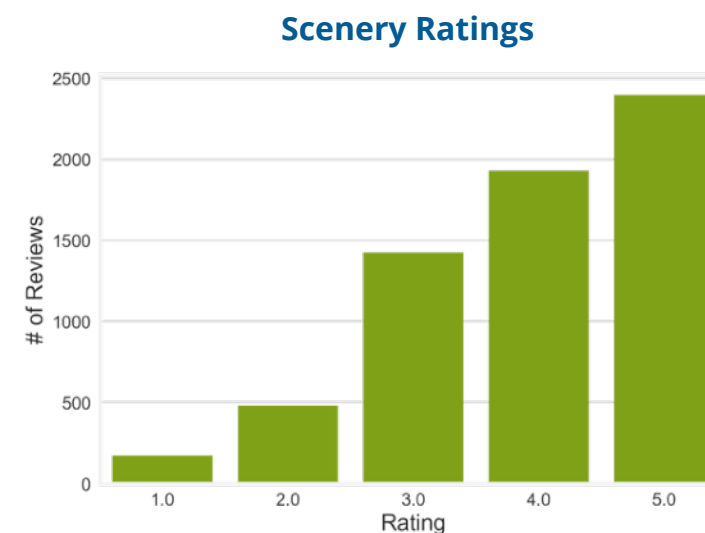
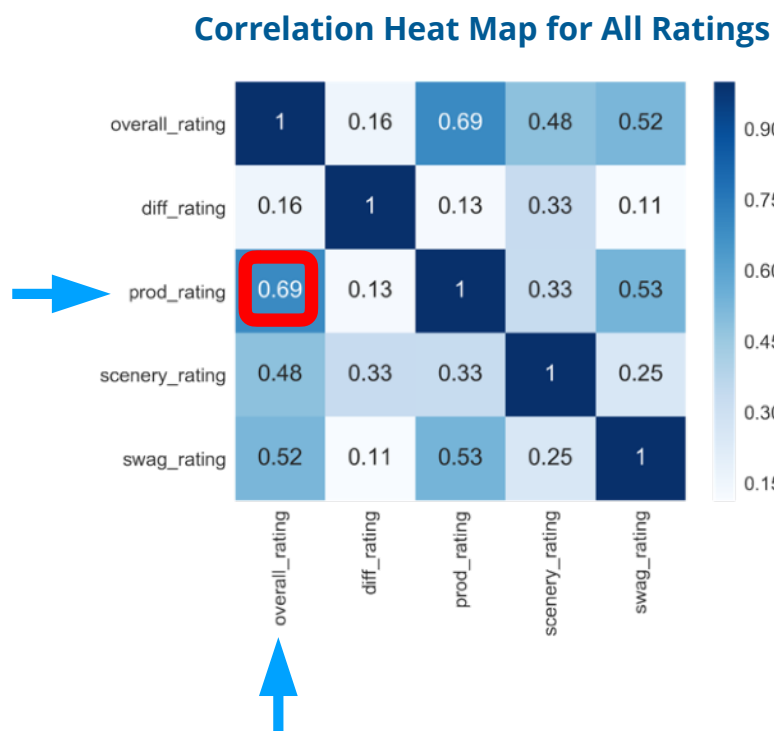
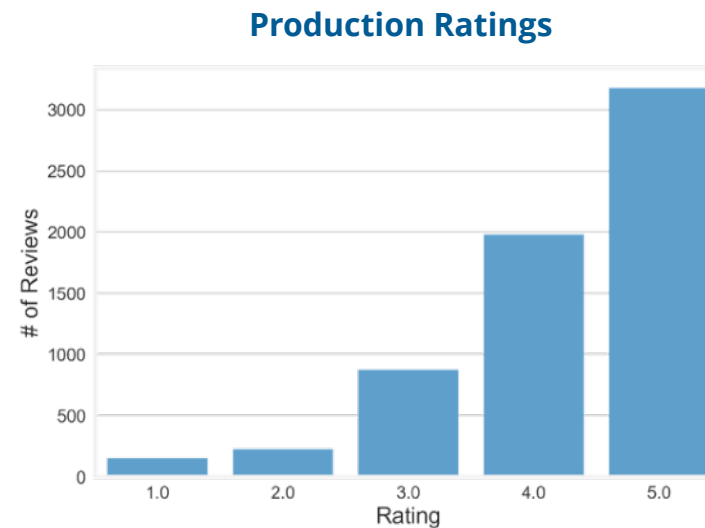
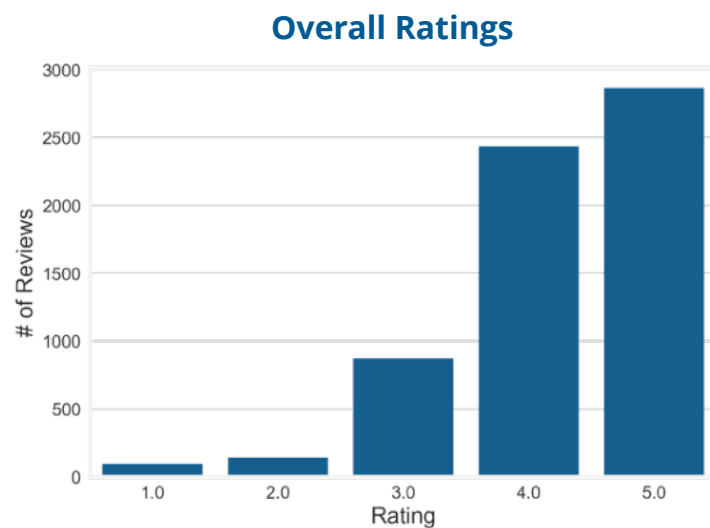
- When building the recommender, reviewed races were left in to calculate a “hit rate” for the model
 - If the recommender tends to pick races that the racer has already reviewed, then we know it’s choosing races that the racer wanted to race at some point
 - In the final implementation, the recommender can be set up to exclude past races if desired
- For validation, the model was run on the 20% of racers who weren’t used to build the model
 - 401 racers in the holdout sample were matched to the 1,768 races used to build the recommender model (without their reviews)
 - Nearly half of the races run by the 20% holdout did not appear in the set used to match, driving the hit rate downward
 - Even with this handicap, the recommender was able to match races for 7% of racers



Beyond the Recommender - Race Ratings

Racers generally give races high ratings - most are rated 4 or 5, and the average overall rating is 4.04

- Production Ratings are the most highly correlated with Overall Ratings, indicating that race production is the leading factor in determining a racer's overall happiness
 - Simple linear regression was used to determine that nearly 50% of the overall rating comes from production, and an additional 10% is explained by scenery and swag



Beyond the Recommender - User Profiling

Clustering

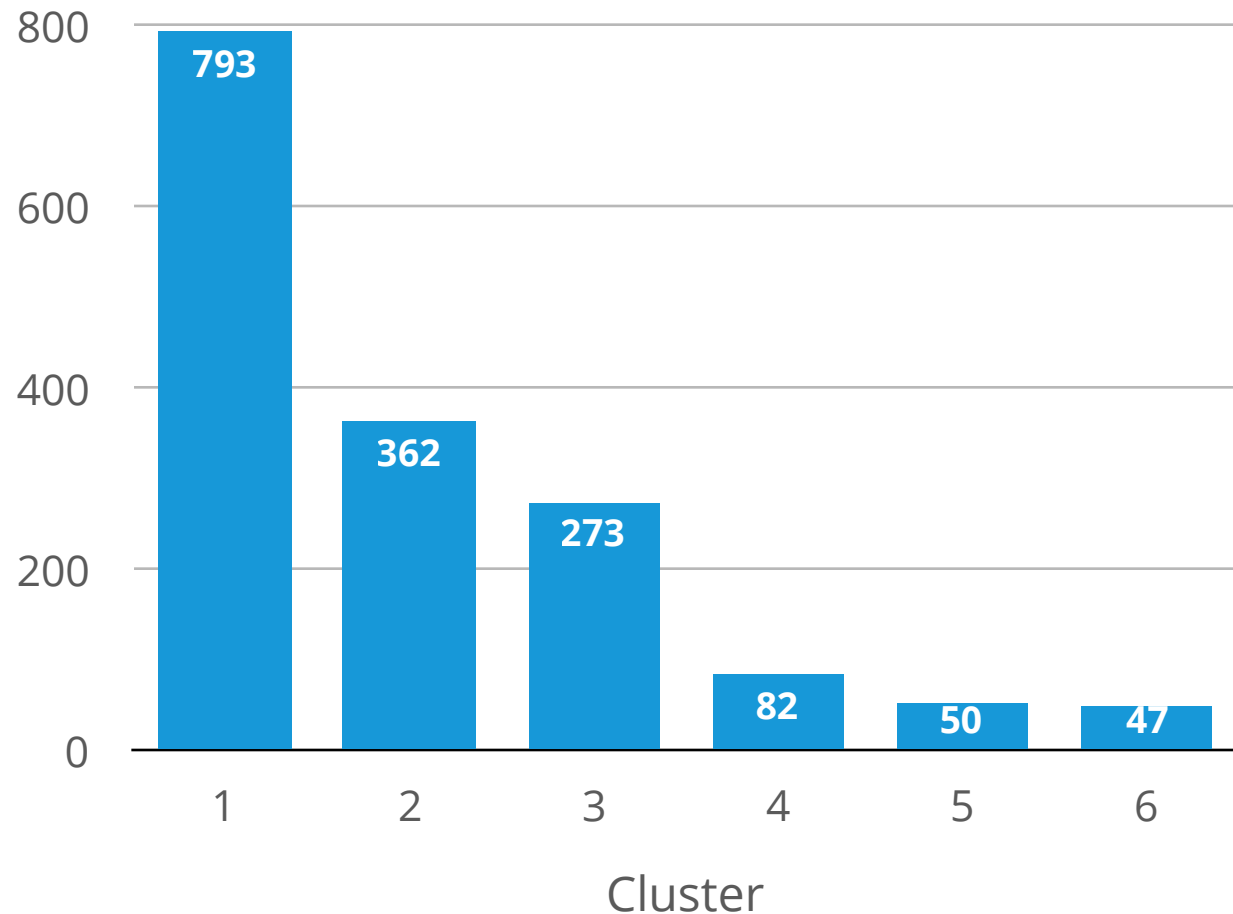
- In addition to building the race recommender, I took some time to analyze the users of RaceRaves.com to get a sense of what they look like
 - Once the training population was selected and the LDA model was run, clusters were built using all of the profile data available from the site, along with the LDA topic model probabilities
- The methodology used for clustering was K-Means
 - The basic idea is that individuals are clustered by calculating their distance from the average value within the cluster for all variables being used
 - The “K” in K-Means represents the number of clusters, which has to be specified at the start of the process
 - I also tried other clustering methods, including hierarchical and spectral clustering, but K-Means gave me the best separation into logical clusters with interpretable profiles

Profiles

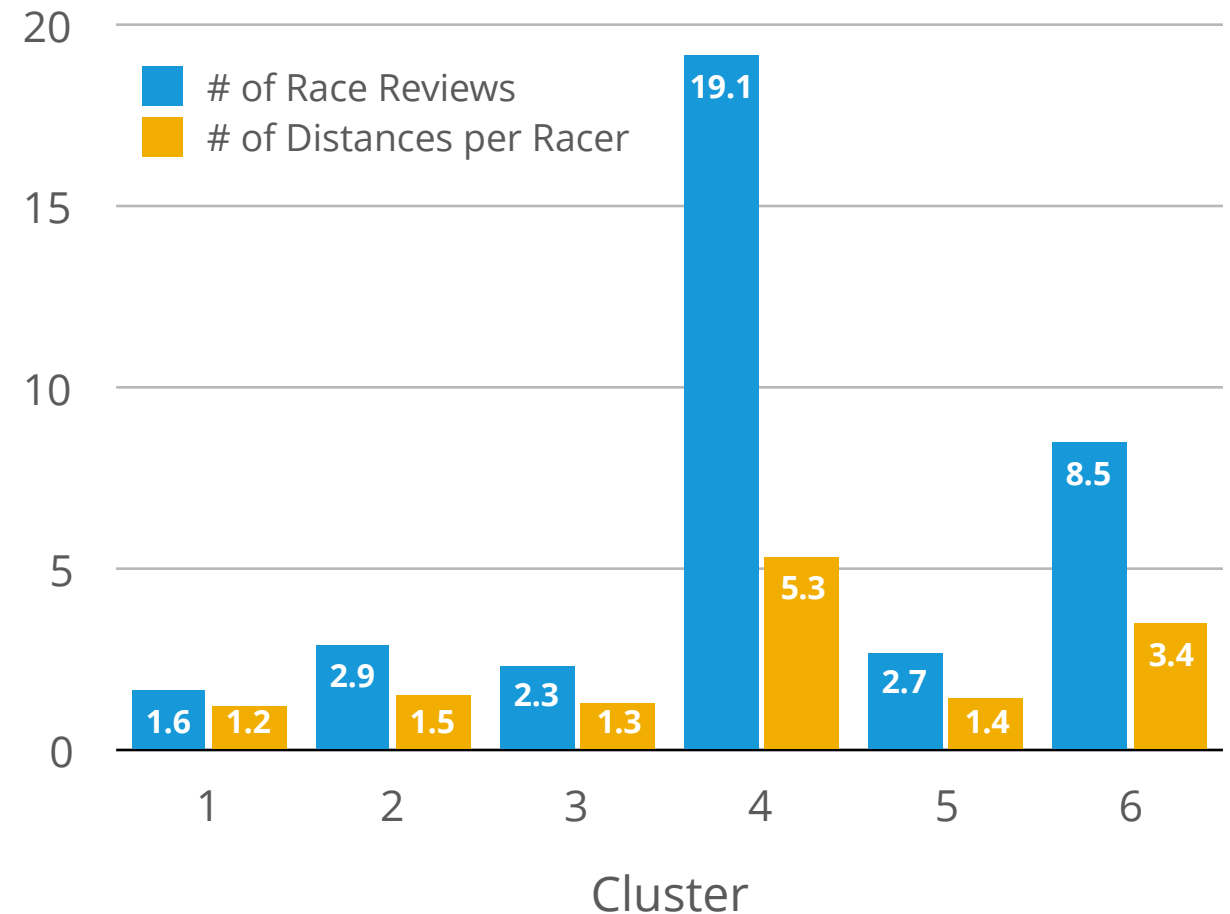
- Six clusters were chosen for the profiles
 - The final clusters had fairly clear differentiation across key measures, as seen in the following slides
 - The last two clusters are fairly small, but they show clear differences in their profiles that prevented them from being combined into larger clusters

The largest cluster has the fewest reviews per racer

of Racers in Each Cluster



of Reviews and Distances Run by Cluster



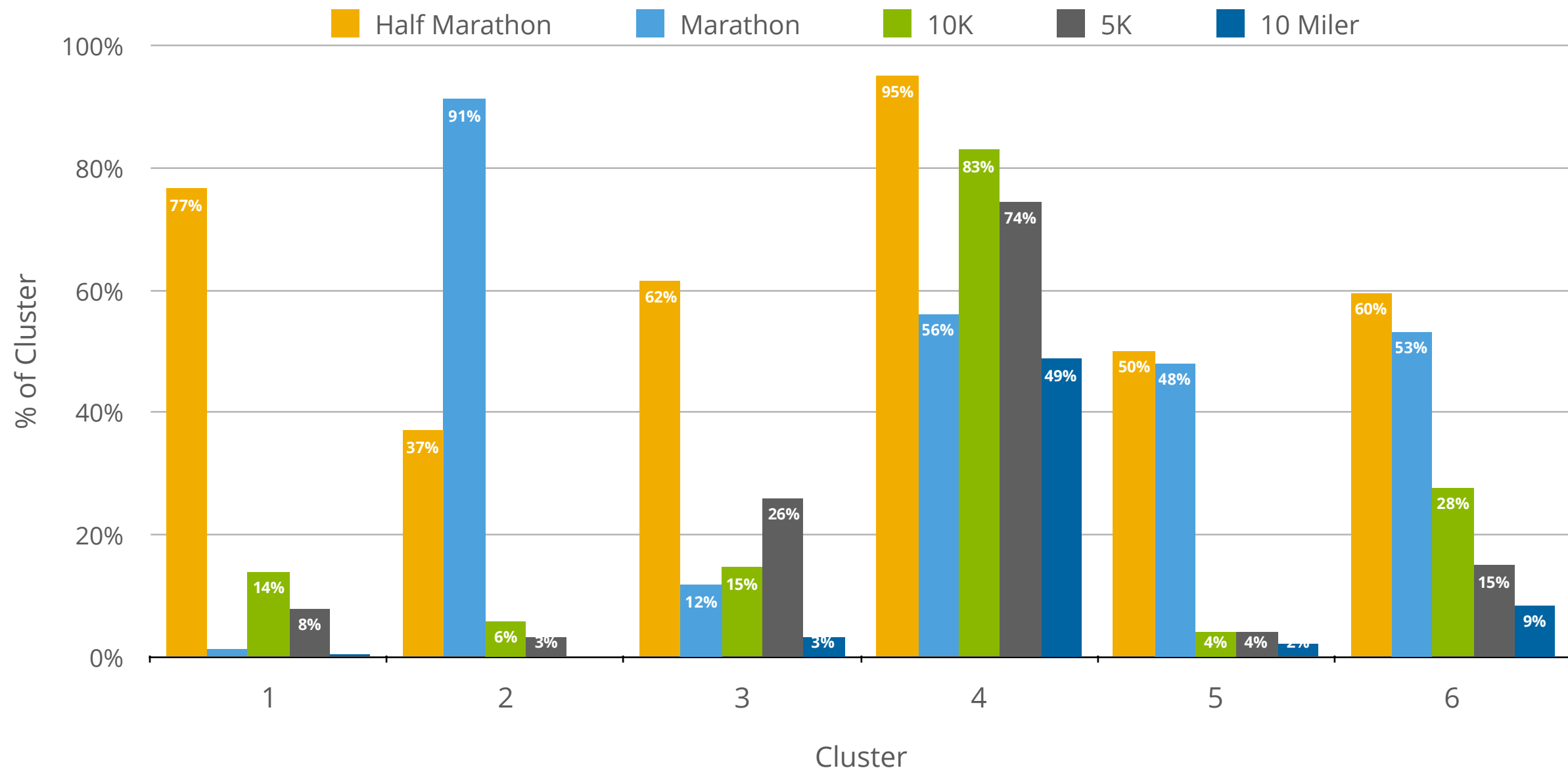
- The largest cluster makes up nearly 40% of all reviewers, and they average less than 2 races reviewed
- Clusters 2 & 3 look similar in size, but their differences will become clear on later slides
- Cluster 4 has the most reviews and distances run - these are the most engaged users of RaceRaves
- Cluster 6 is also highly engaged, with an average of 8.5 reviews

Cluster 1 tends to give the highest rating



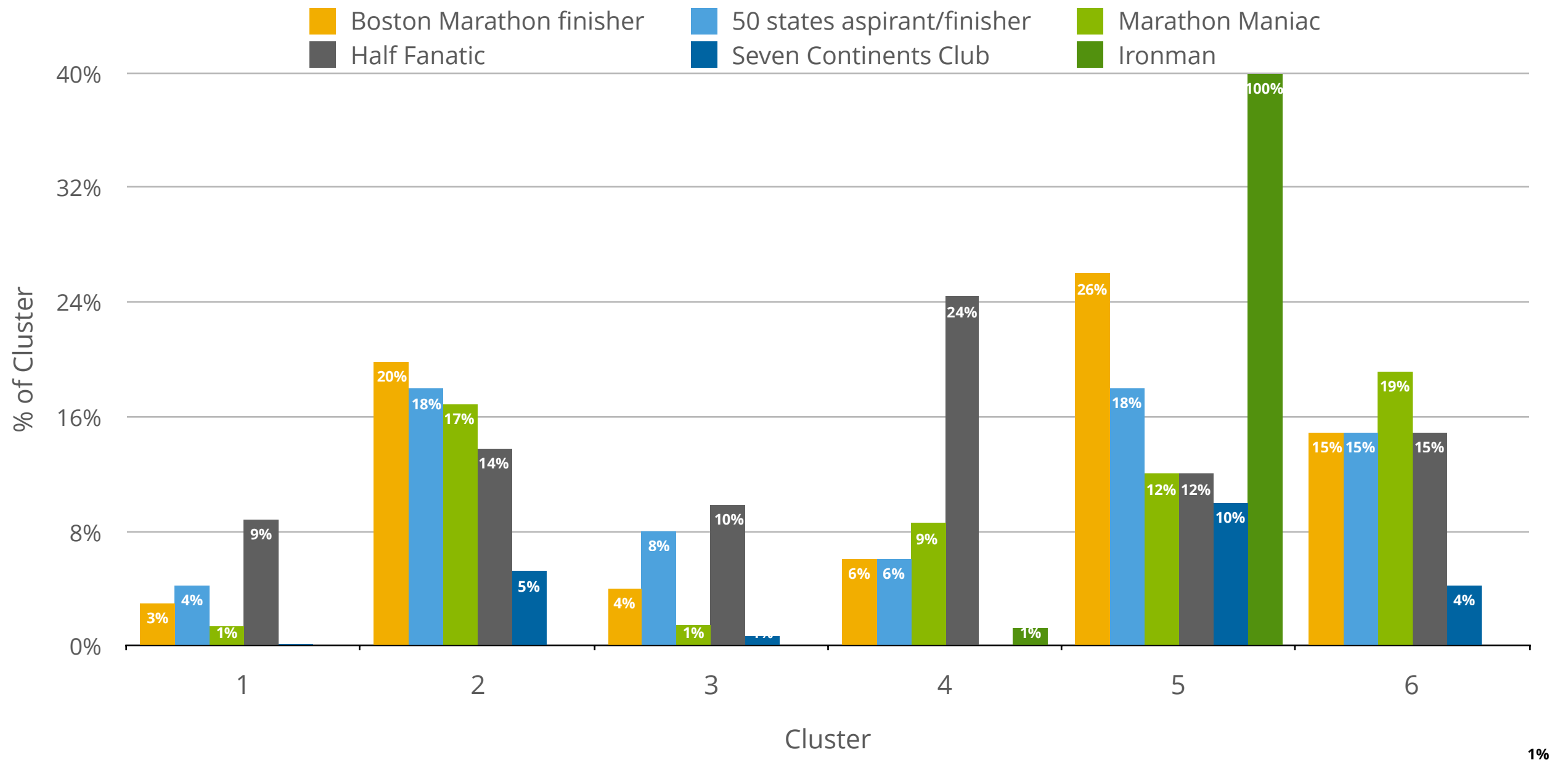
- Cluster 1 gave the highest ratings across all categories, while Cluster 3 consistently gave the lowest
- The other clusters generally give similar ratings, but Cluster 4 is more likely to rate difficulty < 3
- As seen with the correlation data, Production Ratings tend to drive the Overall Ratings
 - Average production ratings are within 0.2 of overall ratings for all clusters except Cluster 3
 - Statistically speaking, production ratings drive about 50% of the variation in overall ratings
 - Difficulty ratings are not correlated with overall ratings, but they do differ by cluster

Half marathons are the most reviewed races



- Half marathons made up 45% of all reviews included in the analysis, and they were the most likely race for 5 of 6 clusters
 - Cluster 2 is dominated by marathon runners - 91% reviewed a marathon, and only 37% reviewed a half
- Cluster 4 reviewed the widest range of races - more than half reviewed the 4 most popular distances
- Cluster 5 is equally as likely to have reviewed a full or half marathon

Cluster 4 dominates the Affiliations



- Affiliations are concentrated in clusters 2, 5 and 6
 - Clusters 2 & 5 have the most Boston Marathon runners and 50 state aspirants
 - Cluster 5 is all Ironman competitors (note that the bar is cut short to make the chart readable)
- The highly engaged users in Cluster 4 tend to be Half Fanatics, which is consistent with the review data

RaceRaves User Clusters - Snapshots

Cluster 1

Largest cluster with the fewest average reviews but highest ratings; made up mostly half of marathon runners

Cluster 2

Primarily marathon runners, and least likely to review half marathons; includes the most Boston Marathon finishers

Cluster 3

Lowest ratings, mostly half marathon runners

Cluster 4

Most engaged users, with an average of 19 reviews and the most distances reviewed

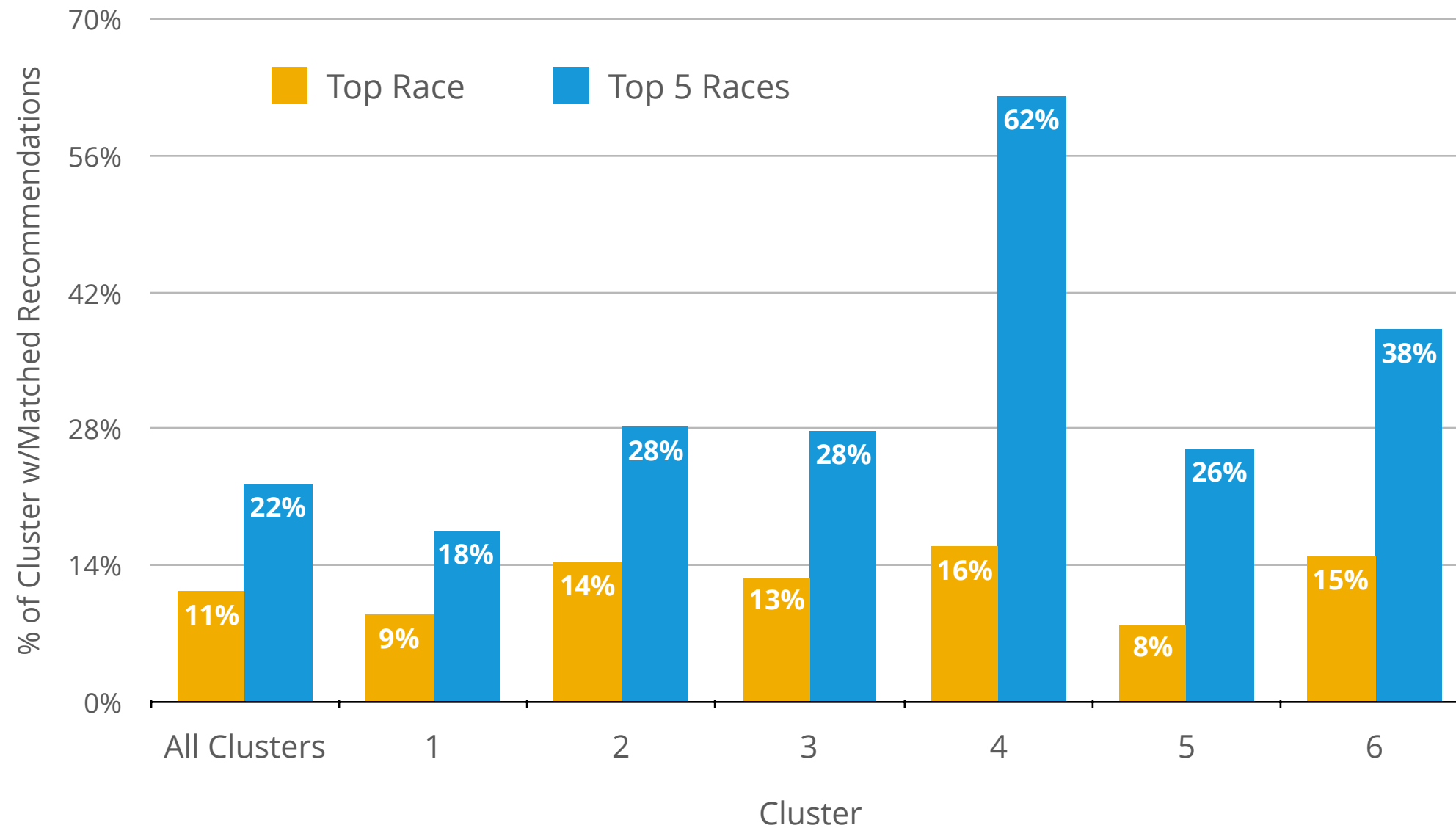
Cluster 5

100% Ironman affiliation, equally as likely to have reviewed marathons and half marathons

Cluster 6

Second highest average # of reviews, but lower ratings and higher affiliations than Cluster 4

Bringing it all together - Cluster Hit Rates



- Given what we've learned about the clusters, it's not surprising that their hit rates differ
 - The engaged users in Cluster 4 have the highest hit rates, likely because they've provided the most data
 - Cluster 2, 3 and 6 all have higher than average hit rates as well
- As more users become engaged with the site and submit reviews, the recommender will become stronger

Next Steps

Recommender

- Overlay filters to ensure that recommender and filters can work together
- Add ratings filter to ensure recommended races are well-reviewed
 - Should include a method for prioritizing on the rating that matters most to racer - production/scenery/swag

Clustering and Profiles

- Incorporate more data from RaceRaves.com
 - Owners have asked to include an indication of whether the racer came to the site via promotion or more organically