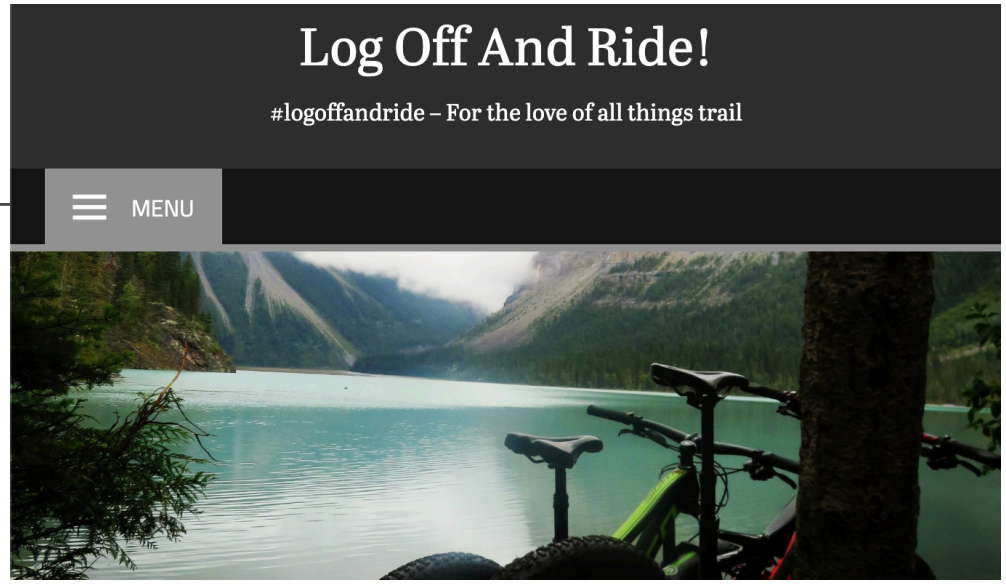
Mountain Bike Trail Recommendation Engine

BRIEN WILSON

Why is this important?

The idea behind bringing modern recommendation engines to trail selection is larger than just the mountain bike world. From local bike shops to REI, more people are pushing to spend more time outside. By providing quality and targeted trail recommendations, we can make it easier than ever to get people outside and moving.



REI Closing its Doors on Black Friday – Invites Nation to OptOutside

$2.2 billion specialty outdoor retailer will pay 12,000 employees to not work on November 27
10.27.2015

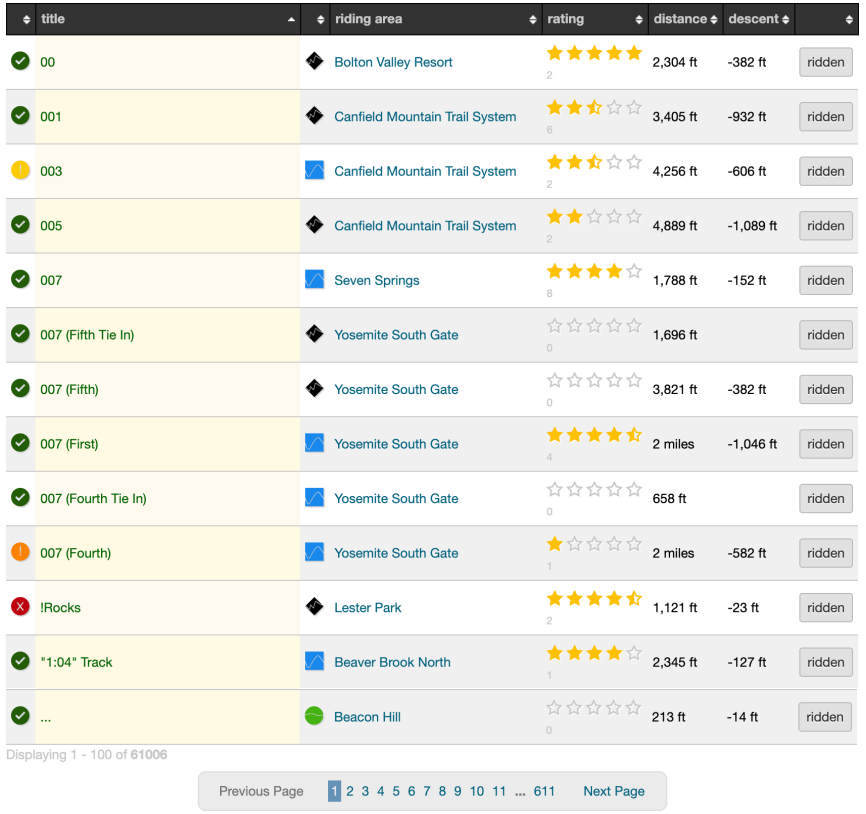
SEATTLE – In a letter addressed to its 5.5 million members, REI, the nation's largest consumer co-op and specialty outdoor retailer, has announced plans to close its doors on Black Friday (November 27) at all 143 of its retail locations, headquarters and two distribution centers. Instead of reporting to work, the co-op is paying its 12,000 employees so they can do what they love most – be outside. Starting today, the co-op is inviting the nation to join in by choosing to #OptOutside to reconnect with family and friends this Thanksgiving holiday.

Outline

- Data Sources
- Data Analysis
  - Ratings
  - Trail Characteristics
- Modeling & Method Selection
  - Scikit-Surprise Collaborative Models
    - Data Prep
    - Algorithm Selection
  - Content based Cosine Similarity
    - Data Prep & Pipeline
- Same User – Different Results
- Full Production Considerations & Future Work

Data Sources: Scraping Info

- All data was sourced from trailforks.com in the United States trail region
  - Scrapy scripts are linked to master list of U.S. mountain bike (mtb) trails so updates to that list can be easily obtained
- Scraping was done in three steps
  1. Trail statistics and descriptions
     - 59,477 trails x 28 stats
  2. User ratings for each trail
     - 66,327 ratings - 10,992 users - 24,730 trails
  3. User comments for each trail
     - 64,471 comments – 5,527 users – 11,738 trails
     - collected for future integration of implicit trail ratings



title	riding area	rating	distance	descent	
00	Bolton Valley Resort	2	2,304 ft	-382 ft	ridden
001	Canfield Mountain Trail System	6	3,405 ft	-932 ft	ridden
003	Canfield Mountain Trail System	2	4,256 ft	-606 ft	ridden
005	Canfield Mountain Trail System	2	4,889 ft	-1,089 ft	ridden
007	Seven Springs	8	1,788 ft	-152 ft	ridden
007 (Fifth Tie In)	Yosemite South Gate	0	1,696 ft		ridden
007 (Fifth)	Yosemite South Gate	0	3,821 ft	-382 ft	ridden
007 (First)	Yosemite South Gate	4	2 miles	-1,046 ft	ridden
007 (Fourth Tie In)	Yosemite South Gate	0	658 ft		ridden
007 (Fourth)	Yosemite South Gate	1	2 miles	-582 ft	ridden
!Rocks	Lester Park	2	1,121 ft	-23 ft	ridden
"1:04" Track	Beaver Brook North	1	2,345 ft	-127 ft	ridden
...	Beacon Hill	0	213 ft	-14 ft	ridden

Displaying 1 - 100 of **61006**

Previous Page 1 2 3 4 5 6 7 8 9 10 11 ... 611 Next Page

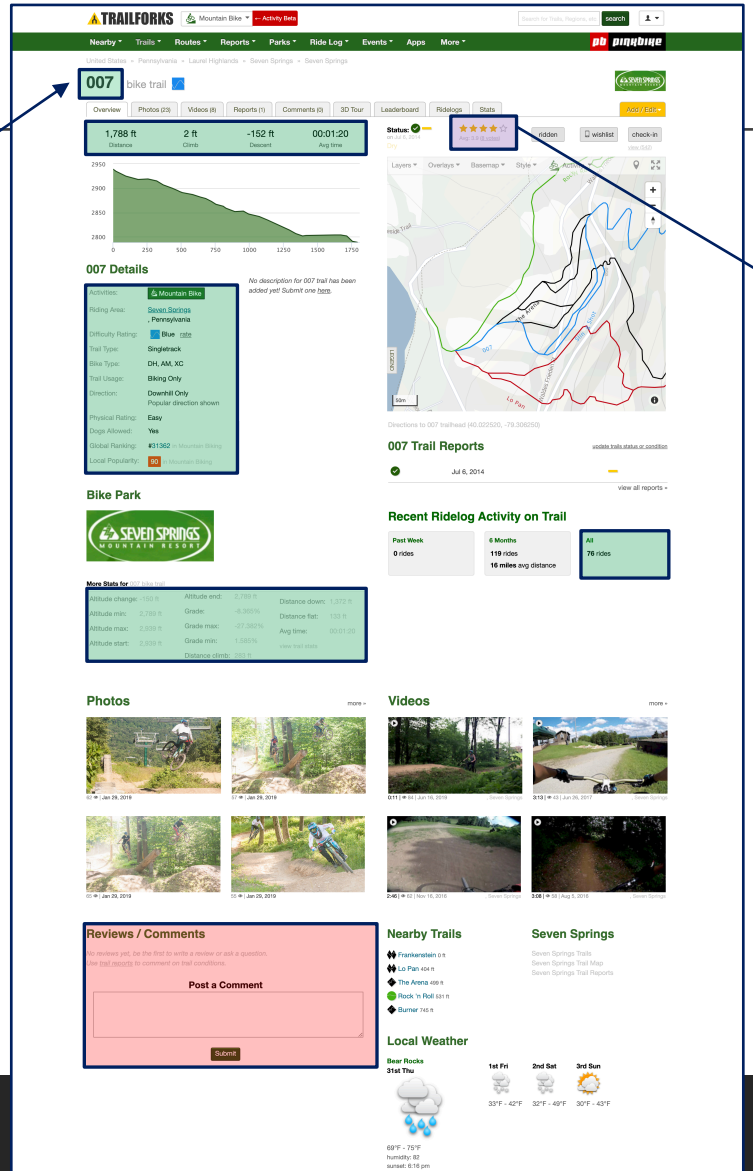
Data Sources: Scraping Path

- Trail Stats
- User Ratings
- User Comments

title	riding area	rating	distance	descent	
00	Bolton Valley Resort	2	2,304 ft	-382 ft	ridden
001	Canfield Mountain Trail System	6	3,405 ft	-932 ft	ridden
003	Canfield Mountain Trail System	2	4,256 ft	-606 ft	ridden
005	Canfield Mountain Trail System	2	4,889 ft	-1,089 ft	ridden
007	Seven Springs	8	1,788 ft	-152 ft	ridden
007 (Fifth Tie In)	Yosemite South Gate	0	1,696 ft		ridden
007 (Fifth)	Yosemite South Gate	0	3,821 ft	-382 ft	ridden
007 (First)	Yosemite South Gate	4	2 miles	-1,046 ft	ridden
007 (Fourth Tie In)	Yosemite South Gate	0	658 ft		ridden
007 (Fourth)	Yosemite South Gate	1	2 miles	-582 ft	ridden
!Rocks	Lester Park	2	1,121 ft	-23 ft	ridden
"1:04" Track	Beaver Brook North	1	2,345 ft	-127 ft	ridden
...	Beacon Hill	0	213 ft	-14 ft	ridden

Displaying 1 - 100 of **61006**

Previous Page 1 2 3 4 5 6 7 8 9 10 11 ... 611 Next Page



TRAILFORKS Mountain Bike search

Nearby Trails Routes Reports Parks Ride Log Events Apps More

Trail Voting History

username	date	rating
acw13	May 20, 2019	
achristopher	Jul 29, 2018	
KeithS-Demo8	Nov 27, 2017	
Dylinsned	Mar 13, 2017	
droppindown	Aug 6, 2014	
martis	Jul 30, 2014	
daylenrides	Jul 28, 2014	
GTImkIV	Jul 6, 2014	

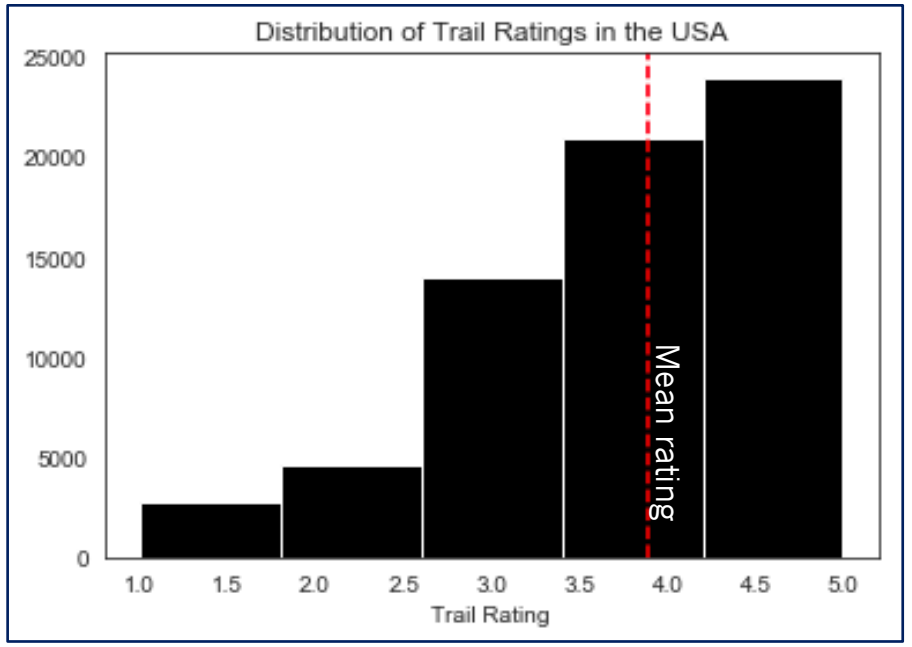
Displaying 1 - 8 of 8

Previous Page 1 Next Page

Data Analysis

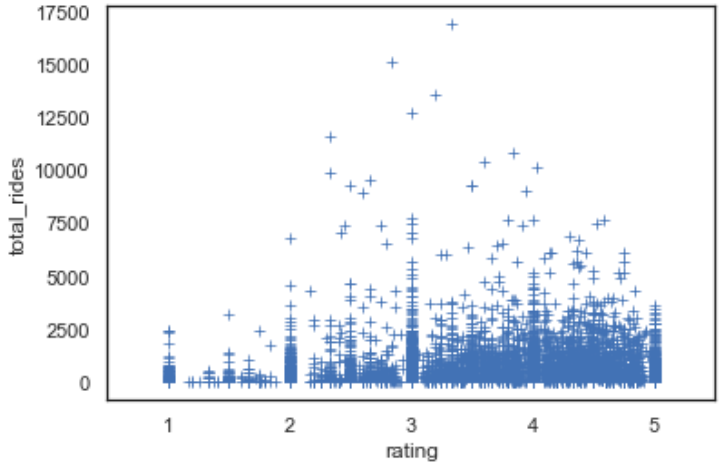
Ratings Distribution

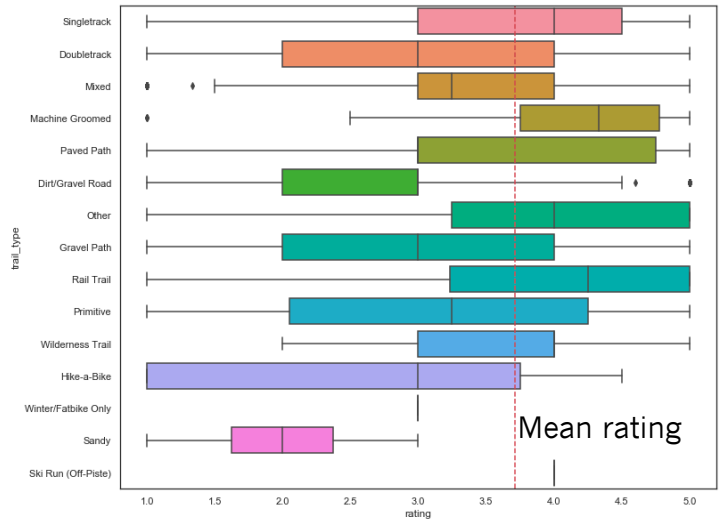
- Global mean ratings: 3.89 stars
- Distribution is left skewed and non-normally distributed
- Users seem to be overall positive towards mtb trails in the U.S.
  - Typical bimodal ratings distribution isn't present in this dataset

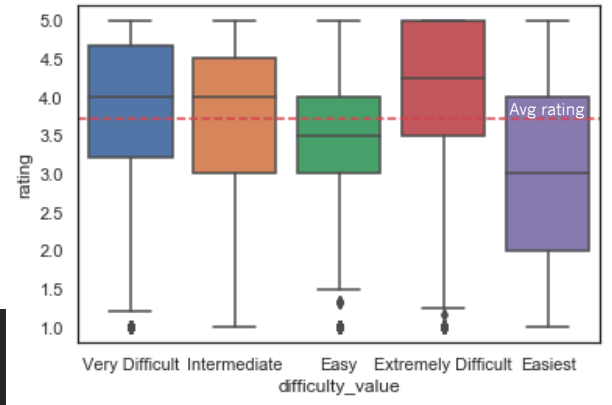


Trail Characteristics: Ratings

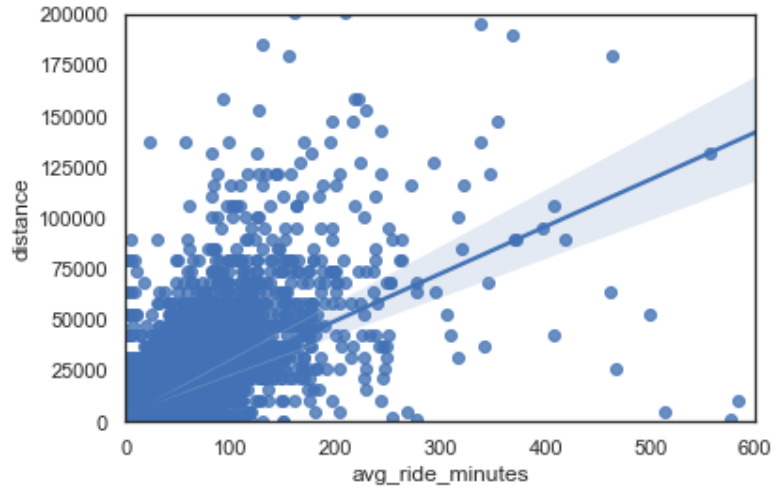
- Total ride count has no affect on the trails average rating
  - Trails with high numbers of rides have mean ratings closer to 3
- Trail type definitely has implications for the average rating
  - Sandy, dirt road, & hike-a-bike trails are not as well liked as the rest of the trail options
- Users are rating more difficult trails higher on average than easier trails



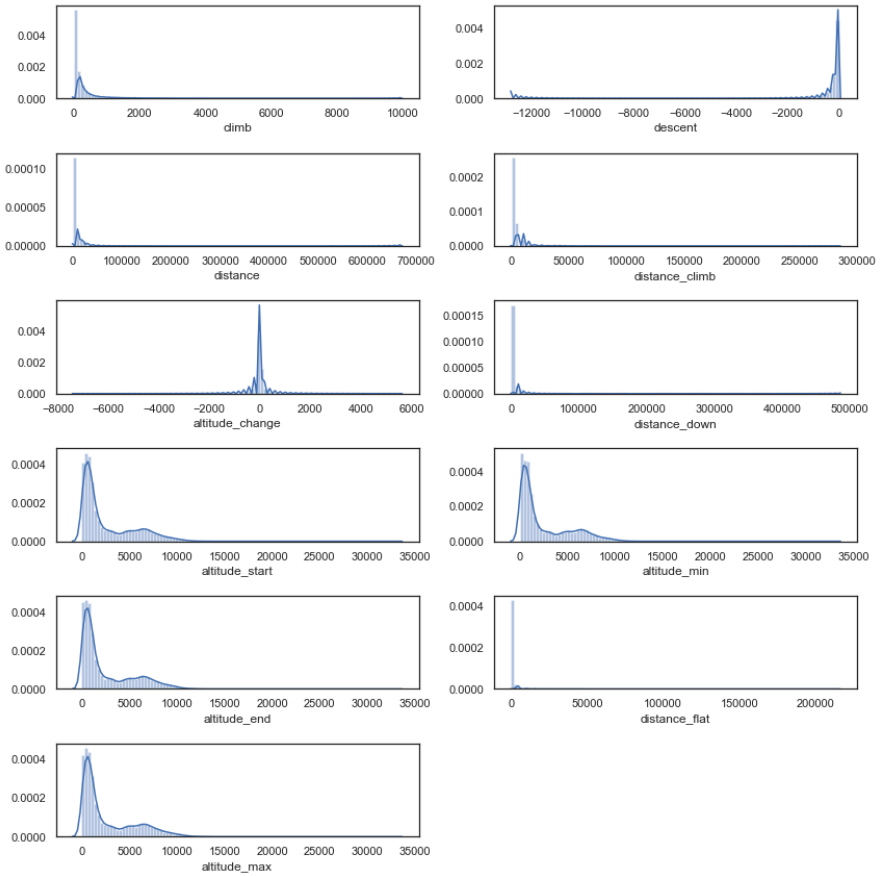




Trail Characteristics: Distance & Alt.



- Trail distance has a weak correlation with average ride minutes
- Altitude change is the only numeric characteristic that is normally distributed



Modeling & Method Selection

Scikit-Surprise: Collaborative Models

- The Surprise module was released September 2019
- Multiple algorithm choices for explicit feedback in user-item matrices
- Surprise adopts model selection and testing functionality (train_test_split, GridSearchCV, etc.) from Scikit-Learn

Algorithm	Description
Normal Predictor	Algorithm predicting a random rating based on the normal distribution of the training set.
KNN Basic	A basic collaborative filtering algorithm.
KNN with Means	A basic collaborative filtering algorithm, taking into account the mean ratings of each user.
KNN with Baseline	A basic collaborative filtering algorithm taking into account a baseline rating.
SVD	The famous SVD algorithm, as popularized by Simon Funk during the Netflix Prize.
Slope One	A simple yet accurate collaborative filtering algorithm.
Co-Clustering	A collaborative filtering algorithm based on co-clustering.

* Table adapted from Surprise Docs

Scikit-Surprise: Data Prep

- Collaborative filtering models need a user, item, and rating
- Surprise's Dataset class will read the data from the file or pandas df and convert it into a user-item-rating sparse matrix
  - Rows = user
  - Columns = item
  - Data = rating

Raw Data

user	trail_id	rating
namdoogttam	trail_99253	5
wanderingMan	trail_99253	4
BackyardTrailsLLC	trail_130598	4
mtnmanpdx	trail_140378	4
dylanmoore	trail_140378	4

Prepped Data

user	trail_99253	trail_130598	trail_140378
namdoogttam	5	0	0
wanderingMan	4	0	0
BackyardTrailsLLC	0	4	0
mtnmanpdx	0	0	4
dylanmoore	0	0	4

Surprise Collaborative Models

- Models evaluated on predicted rating error
- The two best performing models based on RMSE are KNNBaseline and SVD
  - RMSE is used at this stage so that performance can be compared to external recommendation engine performance
- Algorithm choice was made prior to tuning of SVD with GridSearchCV

Algorithm	RMSE
Normal Predictor	1.47
KNN with Means	1.06
KNN with Baseline	0.96
Co-Clustering	1.08
Slope-One	1.07
SVD[1]	0.92

[1]RMSE is from parameter tuning

SVD Performance Compared to Netflix

- Without the inclusion of temporal or implicit rating information, Trailforks SVD model performs slightly better than the original Netflix model.

Model	Model Size	RMSE
Netflix	500,000 users x 17,000 movies	0.9514
Trailforks	10,992 users x 24,730 trails	0.9238

- Figure shows where performance could be with a model inclusive of implicit ratings
- Netflix has since moved to thumbs up/thumbs down due to model simplicity & performance

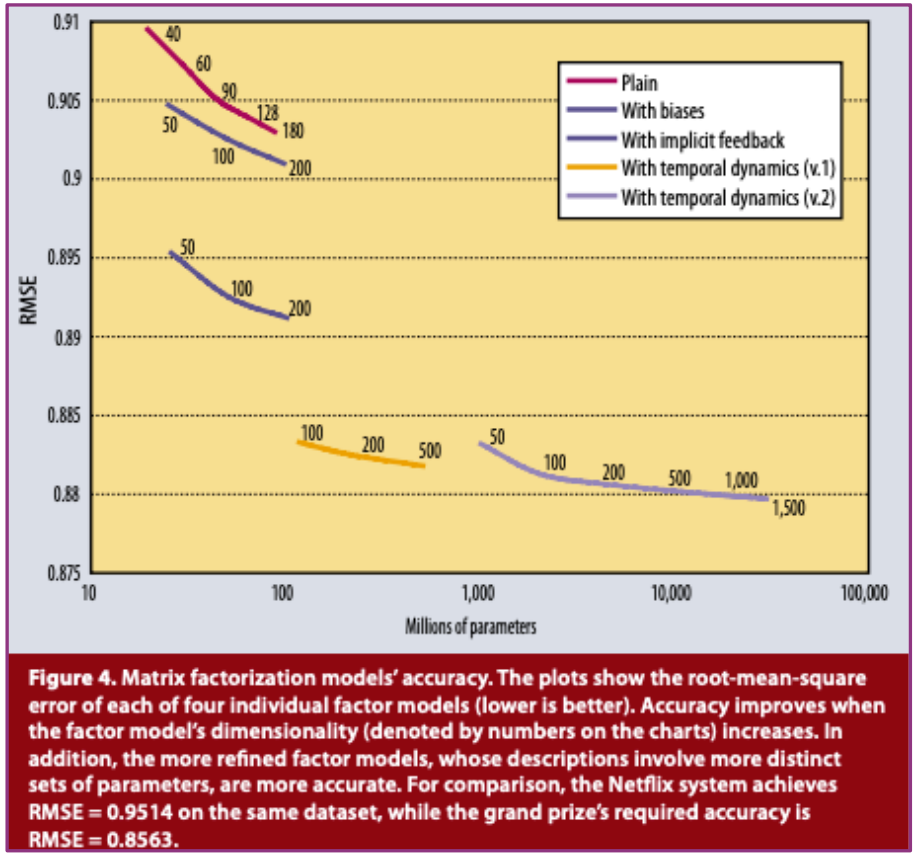


Figure 4. Matrix factorization models' accuracy. The plots show the root-mean-square error of each of four individual factor models (lower is better). Accuracy improves when the factor model's dimensionality (denoted by numbers on the charts) increases. In addition, the more refined factor models, whose descriptions involve more distinct sets of parameters, are more accurate. For comparison, the Netflix system achieves RMSE = 0.9514 on the same dataset, while the grand prize's required accuracy is RMSE = 0.8563.

Cosine Similarity: Data Prep

- From the data prep side the needs are similar to a typical classification or regression problem
- Null imputation and cleaning to retain as many trails as possible
- Since feature data will be vectorized to calculate the cosine similarity, imputation accuracy is more important than volume of rows

Pre-Cleaning

trail_id:	12
trail_name:	1
climb:	5258
descent:	4004
distance:	331
description:	1
grade_min:	703
grade:	3761
distance_climb:	3988
grade_max:	669
altitude_change:	1436
distance_down:	3004
altitude_start:	334
altitude_min:	354
altitude_end:	335
distance_flat:	8906
altitude_max:	16681
average_time:	16527
total_rides:	651
comment_count:	52311
location:	101
riding_area:	101
difficulty_rating:	1
trail_type:	1
bike_type:	25366
dogs_allowed:	2534
ttfs_on_trail:	51667
global_ranking:	16396

Post-Cleaning

trail_id:	12
trail_name:	0
climb:	2087
descent:	1749
distance:	331
description:	0
grade_min:	703
grade:	355
distance_climb:	331
grade_max:	669
altitude_change:	355
distance_down:	331
altitude_start:	357
altitude_min:	377
altitude_end:	358
distance_flat:	331
altitude_max:	361
average_time:	0
total_rides:	0
comment_count:	52311
location:	0
riding_area:	0
difficulty_rating:	0
trail_type:	0
bike_type:	0
dogs_allowed:	0
ttfs_on_trail:	0
global_ranking:	16396

Cosine Similarity: Data Pipeline

- All features need to be vectorized and combined into a trail-feature matrix
- Data Pipeline
  - LabelBinarizer(categorical_features)
  - TfidfVectorizer(trail_descritptions)
  - Concatenation with Numerical
- Total # of Features: 11,730

```
trail_cat_cols = []
for col in df.select_dtypes(include='object').columns:
    if col != 'trail_id' and col != 'trail_name' and col != 'description':
        trail_cat_cols.append(col)

#creating blank array the same size as df, to concatentate labelbinarizer values
cats_transformed = np.empty([len(df), 1])
for col in trail_cat_cols:
    enc = LabelBinarizer().fit_transform(df[col])
    cats_transformed = np.concatenate([cats_transformed, enc], axis=1)

#removing the first column which is generated in the np.empty statement
cats_transformed = cats_transformed[:,1::1]

vectorizer = TfidfVectorizer(
    min_df=5,
    max_df=0.4,
    stop_words='english',
    use_idf=True,
    smooth_idf=True
)
tfidf_features = vectorizer.fit_transform(df['description']).toarray()

numerical = df.select_dtypes(include='number')
numerical = numerical.drop(['rating'], axis=1)
numerical = numerical.to_numpy()

all_features = np.concatenate([tfidf_features, cats_transformed, numerical], axis=1)
```

Cosine Similarity: Methods

- This similarity of two trails is the cos(Θ) where Θ is the angle between the two vectors that are defined by the trail statistics/variables

$$\text{similarity} = \cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\|\|\mathbf{B}\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}}$$

- In this case, there isn't a statistically rigorous way to evaluate the effectiveness of our recommendations since the output is a measurable difference between two trails

- Results in values between -1 and 1
  - 1 = vectors are the same
  - 0 = vectors are orthogonal
  - -1 = vectors are diametrically opposed

Recommendation Logic

- Content Based Recommendations

```
def spec_rider(user, ratings_data, trail_data):
    rider_trail_ids = ratings_data.loc[ratings_data.user == user, ['trail_id']]
    rider_trail_ids = rider_trail_ids.values.flatten()
    rider_trails = trail_data.loc[trail_data.trail_id.isin(rider_trail_ids)]
    return rider_trails

def rider_recs(sim_matrix, user_df, trails_df):
    ids = user_df['trail_id'].values

    #filtering the similarity values to the users trailset
    sim_rider = sim_matrix.loc[ids, ~sim_matrix.index.isin(ids)]

    #using the users trailset to return the top ten trail_id recommendations

    trail_recs = sim_rider.mean(axis=0).sort_values(ascending=False).index[0:10]

    top_ten_trails = trails_df.loc[trails_df.trail_id.isin(trail_recs)]
    return top_ten_trails
```

- Collaborative Filtering Recommendations

```
def all_users_ratings(user_n, dataf):
    return dataf.loc[dataf.user == user_n, 'trail_id'].values

def get_n_trail_recs(rating_df, user_name, trail_df, n):

    all_trails = rating_df.trail_id.unique()
    users_trails = all_users_ratings(user_name, rating_df)

    user_testset = [t_id for t_id in all_trails if t_id not in users_trails]

    rec_list = []
    for trail in user_testset:
        pred = algo.predict(user_name, trail)
        rec_list.append([pred.iid, pred.est])

    rec_list = sorted(rec_list, key=lambda x: x[1], reverse=True)
    rec_list = rec_list[0:n]

    rec_trails = [rec[0] for rec in rec_list]

    recs = trail_df.loc[trail_df.trail_id.isin(rec_trails)]
    for rec in rec_list:
        recs.loc[recs.trail_id == rec[0], 'predicted_rating'] = rec[1]

    return recs.sort_values(by=['predicted_rating'], ascending=False)
```

Same User - Different Results

Why wisemtnbkr? Why Utah?

- Rider with the most ratings in the dataset which would give the most variation in similarity measures for cosine similarity recommendations
- Similarly, the innate biases and rating patterns will be well defined for the SVD predictions
- Utah is a popular riding destination in the U.S. and a top 3 state based on volume

Top 5 Users by Review Count

wisemtnbkr	1286
schillingsworth	1101
lro0001	979
ericfoltz	875
socalstokie	681

wisemtnbkr's Most Reviewed States

Utah	1197
Colorado	63
Nevada	10
Idaho	10
Wyoming	5

Top 5 States by Review Count

California	2813
Utah	2267
Colorado	1765
Arizona	1521
Washington	1491

user: wisemtnbkr – SVD

- All predicted ratings are over 4 stars and have a good mix of trails from across the state
  - Difficulty: intermediate – very_difficult trails
  - Includes trails with larger altitude changes
  - All trails here have been rated by at least one rider

rank	trail_name	city	state	riding_area	trail_type	total_rides	difficulty	dogs?	alt_change	grade	description	pred_rating
1	Jazz Chrome Molly	Vernal	Utah	Red Fleet Reservoir	Singletrack	64.0	Intermediate	Yes	54.0	0.251	Fun and flowing singletrack. Mostly smooth, bu...	4.545969
2	South Fork Little Deer Creek	Sundance	Utah	American Fork Canyon	Singletrack	1235.0	Intermediate	Yes	-619.0	-5.670	Maybe the best stretch of singletrack in AF Ca...	4.469843
3	Hidden Canyon	Hurricane	Utah	Hurricane	Singletrack	669.0	Very Difficult	Yes	-62.0	-0.649	This is extreme because it is difficult to rid...	4.447909
4	Little Creek Slick	Hurricane	Utah	Little Creek Mountain	Singletrack	173.0	Very Difficult	Yes	11.0	0.087	Much like the rest of Little Creek but with a ...	4.398181
5	Tibble Fork	Sundance	Utah	American Fork Canyon	Singletrack	281.0	Very Difficult	Yes	-1403.0	-10.462	The bottom section has seen some work in recen...	4.373867
6	Ridge 157 (North)	Sundance	Utah	American Fork Canyon	Singletrack	48.0	Very Difficult	Yes	-68.0	-0.196	Really fun sections of trail, though often get...	4.344772
7	Stump Hollow (GWT)	Logan	Utah	Logan Canyon	Singletrack	359.0	Intermediate	Yes	1426.0	6.186	Amazing trail with a rough, steep and occasion...	4.324861
8	Syncline North	Logan	Utah	Logan	Singletrack	9.0	Intermediate	Yes	-845.0	-3.552	Incredible trail, but hard to access. Either ...	4.222769
9	Lava Flow	Cedar City	Utah	Iron Hills Trail System (AKA: Southview)	Singletrack	611.0	Intermediate	Yes	-455.0	-4.760	This is a really nice flow trail that has all ...	4.202396
10	South Rim	Hurricane	Utah	Gooseberry Mesa	Singletrack	893.0	Very Difficult	Yes	205.0	0.726	South Rim	4.202363

wisemtnbkr trail stats and counts of trail difficulty

	total_rides	altitude_change	grade	rating
count	1268.000000	1268.000000	1268.000000	1268.000000
mean	868.865931	-42.720032	-1.112689	3.333708
std	1525.189103	461.569324	7.451742	0.775990
min	0.000000	-7401.000000	-48.685000	1.000000
25%	97.750000	-152.500000	-5.682500	3.000000
50%	389.500000	-15.000000	-0.544500	3.250000
75%	942.250000	89.000000	3.832000	4.000000
max	16977.000000	3355.000000	30.017000	5.000000

Intermediate	754
Very Difficult	245
Easy	226
Extremely Difficult	39
Easiest	4

user: wisemtnbkr – Cosine Similarity

- Trails here are more on the intermediate-easy side with less altitude change than the collaborative approach
  - Difficulty: easy-intermediate trails
  - Small altitude changes
  - Includes previously un-rated trails

rank	trail_name	city	state	riding_area	trail_type	total_rides	difficulty	dogs?	alt_change	grade	description	avg_rating
1	Area 51	Orem	Utah	Utah Valley	Singletrack	129.0	Very Difficult	Yes	58.0	1.351000	A good drop out connect with nice spots. A dif...	3.500000
2	BST (22nd to 27th)	Ogden	Utah	Ogden	Singletrack	434.0	Easy	Yes	-4.0	-0.099000	Easy spin from the 27th street access point, m...	3.000000
3	BST (North Ogden)	Ogden	Utah	Ogden	Singletrack	956.0	Easy	Yes	113.0	3.233000	Sweet piece of single track starts here.	3.500000
4	Logan River	Logan	Utah	Logan	Singletrack	418.0	Easy	Yes	-69.0	-2.383000	No Description	0.000000
5	Rim Rock Loop	St. George	Utah	Santa Clara River Reserve	Singletrack	426.0	Intermediate	Yes	13.0	0.614000	Great loop, good technical sections, flowy	4.000000
6	The Gap	St. George	Utah	Snow Canyon State Park	Singletrack	65.0	Intermediate	Yes	-35.0	-1.716000	Nice connector	0.000000
7	Treadstone (Prayer Flags)	Eagle Mountain	Utah	Mountain Ranch Bike Park	Singletrack	458.0	Intermediate	Yes	-7.0	-0.186000	No Description	4.000000
8	Zoltar	Moab	Utah	Klonzo	Singletrack	419.0	Intermediate	Yes	-13.0	-0.382916	No Description	4.166667
9	Cougar Tracks (Treadstone)	Eagle Mountain	Utah	Mountain Ranch Bike Park	Singletrack	586.0	Intermediate	Yes	35.0	1.099000	Nice easy roll with mostly up and down. You ca...	4.000000
10	BST (Jump Off)	Ogden	Utah	Ogden	Singletrack	961.0	Intermediate	Yes	-21.0	-0.539000	Starts out easy, then suddenly turns really te...	2.666667

wisemtnbkr trail stats and counts of trail difficulty

	total_rides	altitude_change	grade	rating
count	1268.000000	1268.000000	1268.000000	1268.000000
mean	868.865931	-42.720032	-1.112689	3.333708
std	1525.189103	461.569324	7.451742	0.775990
min	0.000000	-7401.000000	-48.685000	1.000000
25%	97.750000	-152.500000	-5.682500	3.000000
50%	389.500000	-15.000000	-0.544500	3.250000
75%	942.250000	89.000000	3.832000	4.000000
max	16977.000000	3355.000000	30.017000	5.000000

Intermediate	754
Very Difficult	245
Easy	226
Extremely Difficult	39
Easiest	4

Production Considerations

- Develop logic for blending of both recommendation methods for final user output
  - Collaborative filtering is based on user-trail interaction which immediately limits the potential outcomes
  - Just because something has no interaction doesn't mean it isn't good or a user wouldn't like it
- Integrate cloud storage and computing
  - Run SVD algorithm on full user-item matrix to estimate all un-rated trails for each user
    - ~270 million predictions
    - Once full matrix is calculated a user request for rec's would be a query instead of running the actual prediction
  - Store similarity matrix in cloud and query form there due to local-memory issues
- Streaming ratings information would be helpful, but currently there is no planned integration with the module and pyspark or other distributed computing options
  - Would need to develop version of their algorithms and dataset classes in spark to add this functionality

Future Work

- Implicit ratings with neural-net approaches
  - develop method to collect information on user interactions with trails to include in implicit models
- Integration with other sources for mtb trails (alltrails.com)
  - If this is implemented as a standalone application
- Database management for faster recommendation queries
- Expand to global trail set
- Include more trail information for cosine similarity measures

Questions?

- Thank you to my understanding family as well as my mentor Brett Nebeker

Data Cleaning Functions

```python
def distance_cleaner(dataf, dist_col):
    df.loc[df[col].isna(), col] = 'NaN'

    # capturing the original distance measure in a new column
    dataf.loc[dataf[dist_col].str.contains('ft'), '{}_measure'.format(dist_col)] = 'feet'
    dataf.loc[dataf[dist_col].str.contains('miles'), '{}_measure'.format(dist_col)] = 'miles'

    # cleaning the text to remove non-numeric characters
    dataf[dist_col] = dataf[dist_col].str.replace('miles', '').str.strip()
    dataf[dist_col] = dataf[dist_col].str.replace('ft', '').str.strip()
    dataf[dist_col] = dataf[dist_col].str.replace(',', '')

    # replacing nulls and converting to floats
    dataf.loc[dataf[dist_col] == 'NaN', dist_col] = np.nan
    dataf.loc[dataf[dist_col] == 'BUMMER', dist_col] = np.nan
    dataf[dist_col] = dataf[dist_col].astype('float')

    #converting all non-feet units to feet
    dataf.loc[df['{}_measure'.format(dist_col)] == 'miles', dist_col] =
        dataf.loc[df['{}_measure'.format(dist_col)] == 'miles', dist_col] * 5280
```

```python
def unique_categories(df, col):
    unique_cat = []
    for i in df[col]:
        hold = i.split(',')
        for cat in hold:
            cat = cat.strip()
            if cat not in unique_cat:
                unique_cat.append(cat)
    return unique_cat

def bummer_to_nan(dataf, col):
    #all of the operations happen inplace so reassignment with the function isn't necessary
    dataf[col] = dataf[col].str.replace(' Reviews & Comments', '').str.replace(',', '')
    dataf[col] = dataf[col].str.replace(' Review & Comments', '').str.replace(',', '')
    dataf.loc[dataf[col] == 'BUMMER', col] = np.NaN
    dataf[col] = dataf[col].astype('float')

def string_list_dummies(unique_items_list, dataf, orig_col):
    for item in unique_items_list:
        if item != 'BUMMER' and item != '':
            col_title = item.lower().replace(' ', '_')
            dataf.loc[:, col_title] = 0
            dataf.loc[df[orig_col].str.contains(item), col_title] = 1
```