A Predictive Machine Learning Pipeline for Large-Scale Fitness Data

Project 4

Predictomondo, Inc.

The Predictomondo Team

David Doerner

Chief Analytics Officer

Jason Gilberg

Chief Business Development Officer

Patrick Mulrooney

Chief Data Architect

Masashi Omori

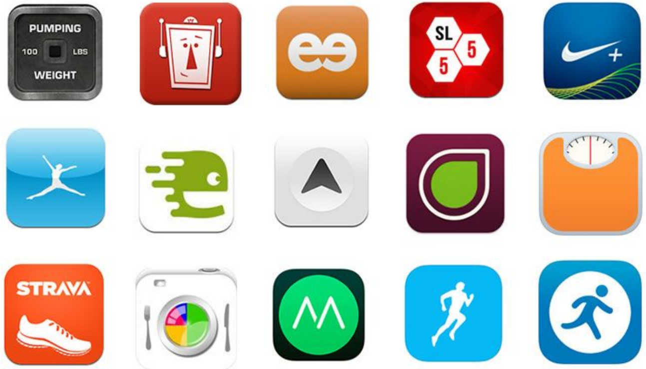
Chief Financial Officer

Advisor

Prof. Julian McAuley

Motivation:

[2]

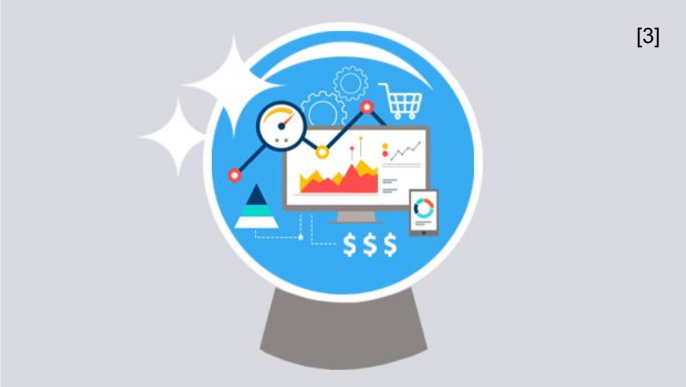


- In 2016 there was an estimated 10 million unique users per month of fitness tracking apps in the US [1]
- Most fitness apps provide similar functionality
- Performance prediction provides users with information necessary to find the workouts aligned with their goals
- PaaS - Performance (Prediction) as a Service

[3]

[1] - https://medium.com/@sm_app_intel/these-fitness-app-statistics-show-whats-going-right-and-wrong-for-fitbit-da2c4c3be142
[2] - http://www.revolutionaryfitness.org/wp-content/uploads/2014/10/health-fitness-apps.jpg
[3] - http://cdn.app.compendium.com/uploads/user/e7c690e8-6ff9-102a-ac6d-e4aebca50425/f4a5b21d-66fa-4885-92bf-c4e81c06d916/Image/41b7fb3e99a27866a3a18db73cae447d/paas.jpg

Objectives:



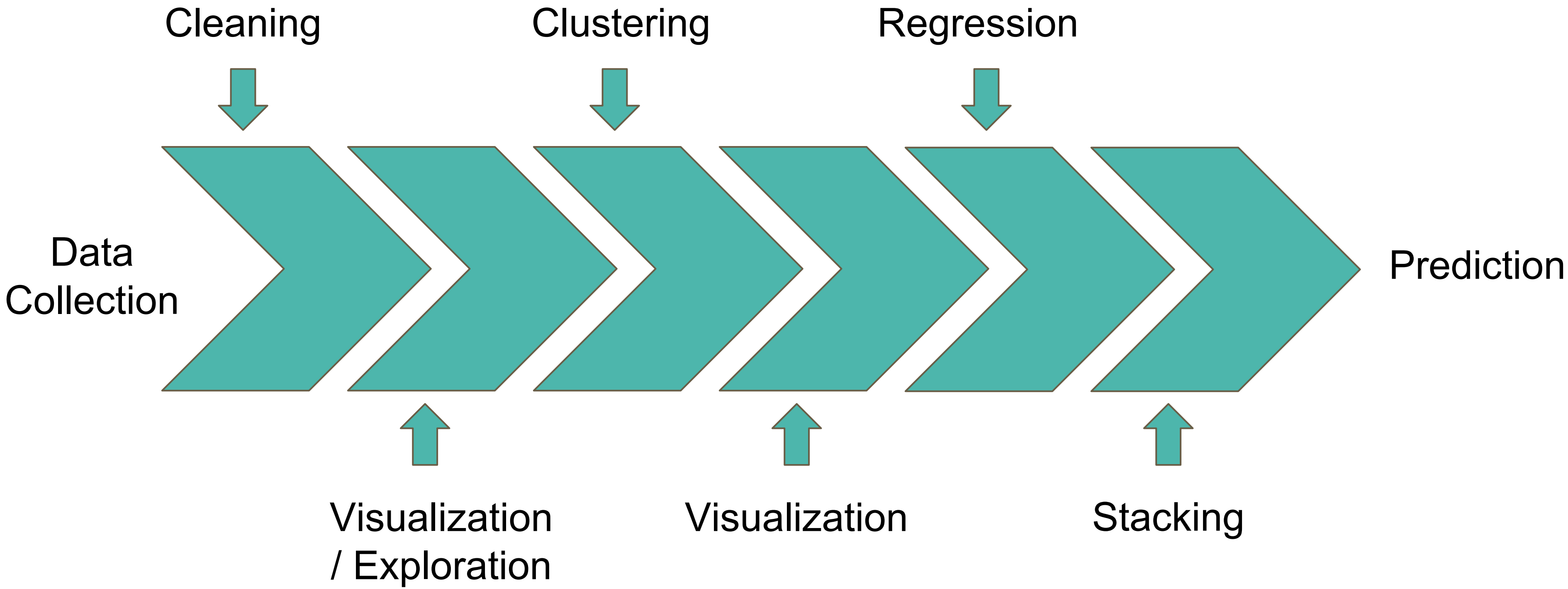
- Determine if historic workout performance can be combined with route characteristics to create a model capable of predicting user performance on that route

- Find a model that can be regenerated regularly in a timely cost effective manner

- Identify changes to current platform that might improve quality of prediction

[1] - http://images.fitnessmagazine.mdpcdn.com/sites/fitnessmagazine.com/files/1200-woman-running-on-road.jpg
[2] - http://www.wallpapervortex.com/ipad-air-wallpaper-19349_1_other_wallpapers_motivational_running_path.html#.WTpI5BPyvq0
[3] - http://blog.contentsquare.com/wp-content/uploads/2016/08/predictive-analysis-contentSquare-blog-5-620x350.png

Approach:



Cleaning
Clustering
Regression
Data Collection
Prediction
Visualization / Exploration
Visualization
Stacking

Data Source

- Popular workout tracking app
- Large diverse user base all over the world
- User data can be public or private, much of it is public by default
- Sequential IDs make it easy to collect

[1] - http://techpastors.com/wp-content/uploads/2013/06/endomondo-sports-tracker-iphone-android-symbian-blackberry-logo_0.png

Data - Sports:

Sport	Workout Count	Percent of workouts
run	347,324	36.08%
bike	252,397	26.22%
walk	100,362	10.43%
bike_transport	98,596	10.24%
mountain_bike	41,778	4.34%
indoor_cycling	21,876	2.27%
weight_training	17,673	1.84%
swimming	14,272	1.48%
core_stability_training	6,940	0.72%
hiking	6,722	0.70%
circuit_training	6,671	0.69%
treadmill_running	5,316	0.55%
elliptical	5,119	0.53%
...		
Total	962,673	

gender	workoutid	userid	start_time	id	altitude_max	altitude_min	calories	distance	duration	hydration	speed_avg	speed_max
male	224750889	7710890	1375415470	85011			1.59	0.00	6.90	0.00	0.00	0.00
male	255364647	709097	1381154895	29094			544.99	5.51	2,042.00	0.06	0.00	0.00
male	274909769	3055418	1386265933	23607			356.84	5.01	1,318.00	0.00	13.68	13.68
unknown	252209131	12742275	1380372058	164011			829.98	4.83	2,100.00	0.00	8.28	8.28
male	219075119	7191585	1374370754	139051			0.03	0.00	1.00	0.00	0.00	0.00
female	464177040	9571309	1422300439	26001			5.18	0.00	23.93	0.00	0.00	0.00
male	281922808	3085942	1387698476	174231			0.00	0.00	1.05		0.00	0.00

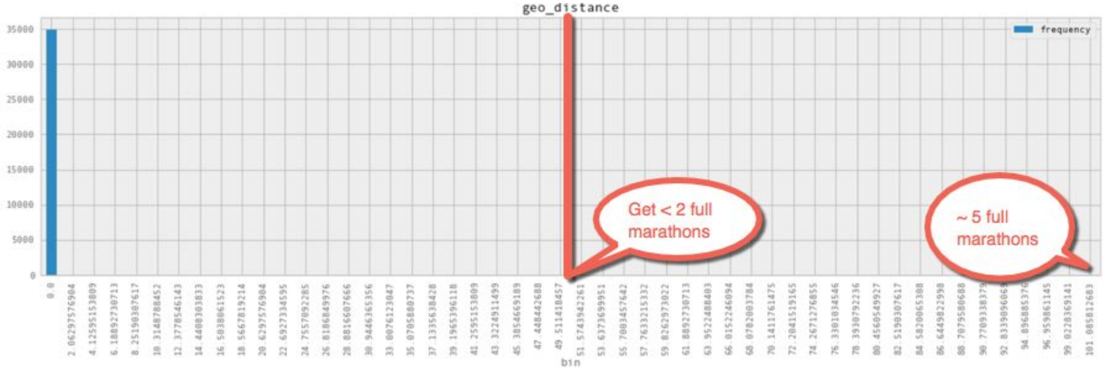
[1] - https://greatist.com/sites/default/files/running.jpg
[2] - https://backroads-web.s3.amazonaws.com/images/search/thumbnail/crater-lake-biking.jpg

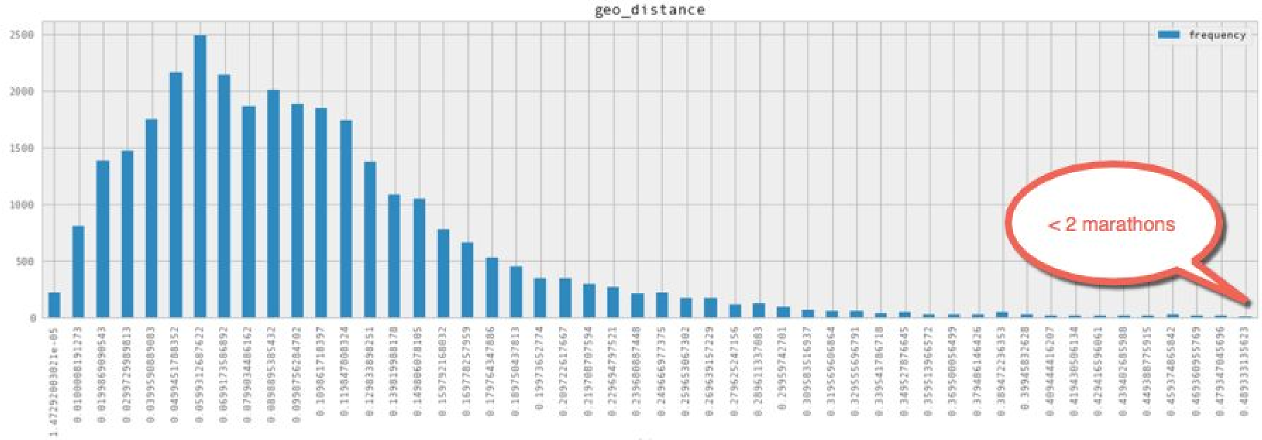
Data - Time Series:

time	altitude	heart_rate	latitude	longitude	speed	workoutid
1353816914			55.79536	37.67927		109669845
1353723007	54.40	157.00	32.67034	35.15908	11.06	109670675
1353820228			52.33786	14.62622	10.36	109673256
1353820614			52.34421	14.63743	13.09	109673256
1353821210			52.34766	14.62678	11.00	109673256
1353823543			52.34033	14.61554	10.39	109673256
1353818802	197.63		51.19067	-2.54721	0.00	109674079
1353787720			1.35163	103.94148	9.67	109674481
1353789977			1.34626	103.95204	7.56	109674481
1353793774			1.33643	103.94112	9.69	109674481
1353699856			1.38154	103.93960		109674491
1353615714			1.38176	103.96448	9.64	109674497
1353441342			1.38615	103.94287	9.06	109674501
1353353175			1.38154	103.93959		109674512
1353358595			1.38033	103.94089		109674512
1353816281			47.95707	16.30224		109687718
1353422850			50.35642	18.25035	13.46	109772944
1353424147			50.33326	18.23801	12.61	109772944
1353424936			50.33900	18.23293	11.70	109772944
1353830871			46.16459	-1.16469	0.00	109778652
1353834753			46.15492	-1.14959	8.10	109778652
1353842749		140.00	55.08966	10.69283		109784738
1353844547			52.20793	10.28677	10.13	109788179
1353821184	1.80	162.72	55.64169	12.64634		109789281
1353842438			46.66591	21.12381	3.30	109795400
1353838722			55.90648	12.14655		109797340

time	altitude	heart_rate	latitude	longitude	speed	workoutid
1353718836	74	180.32	32.69607	35.20433	4.00	109670675
1353718845	75	180.32	32.69593	35.20424	7.22	109670675
1353718861	79.2	180.32	32.69570	35.20405	6.92	109670675
1353718886	85.6	180.32	32.69532	35.20375	8.30	109670675
1353718909	88.2	180.32	32.69492	35.20341	8.09	109670675
1353718920	89	180.32	32.69472	35.20322	9.22	109670675
1353718939	89.2	180.32	32.69439	35.20294	8.50	109670675
1353718962	89.4	180.32	32.69400	35.20258	8.71	109670675
1353718981	90.8	180.32	32.69367	35.20231	8.32	109670675
1353718999	91.2	180.32	32.69334	35.20204	8.90	109670675
1353719025	92.8	130.00	32.69289	35.20160	9.88	109670675
1353719047	93.2	129.00	32.69260	35.20113	9.14	109670675
1353719066	91.8	133.00	32.69237	35.20071	8.77	109670675
1353719081	90.6	132.00	32.69220	35.20038	8.77	109670675
1353719101	88	133.00	32.69196	35.19994	9.00	109670675
1353719124	87.2	134.00	32.69166	35.19942	9.14	109670675

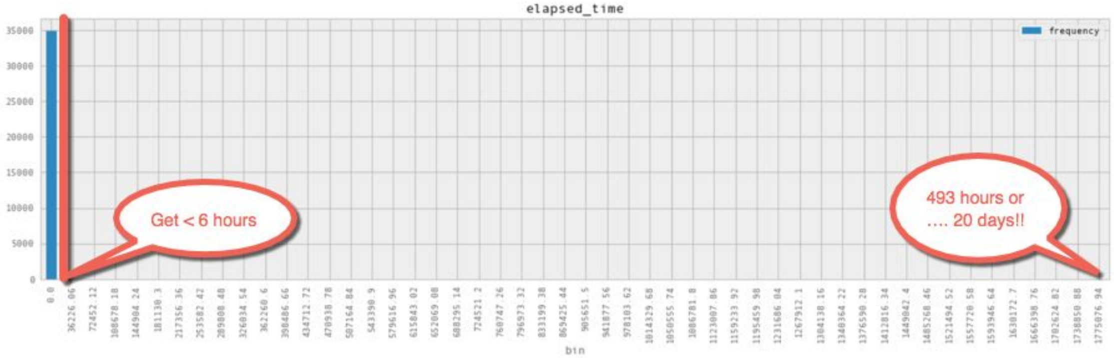
Data Cleaning - Distance

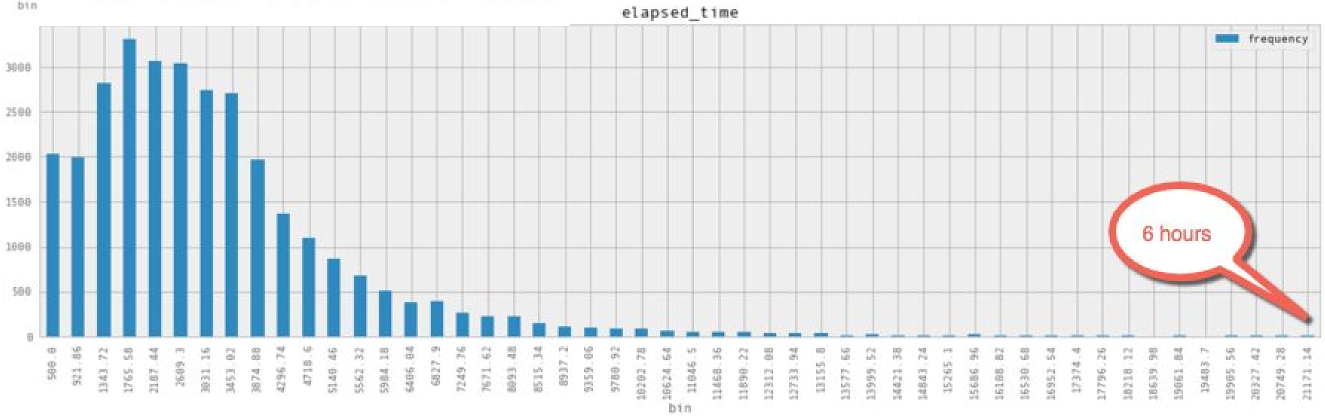




- 5 marathons is 131 miles
- Set reasonable limits
- Remove outliers

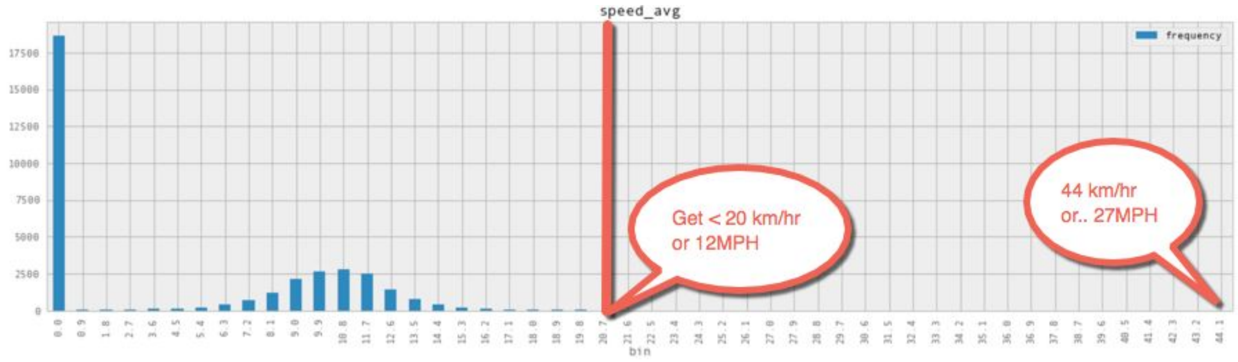
Data Cleaning - Duration



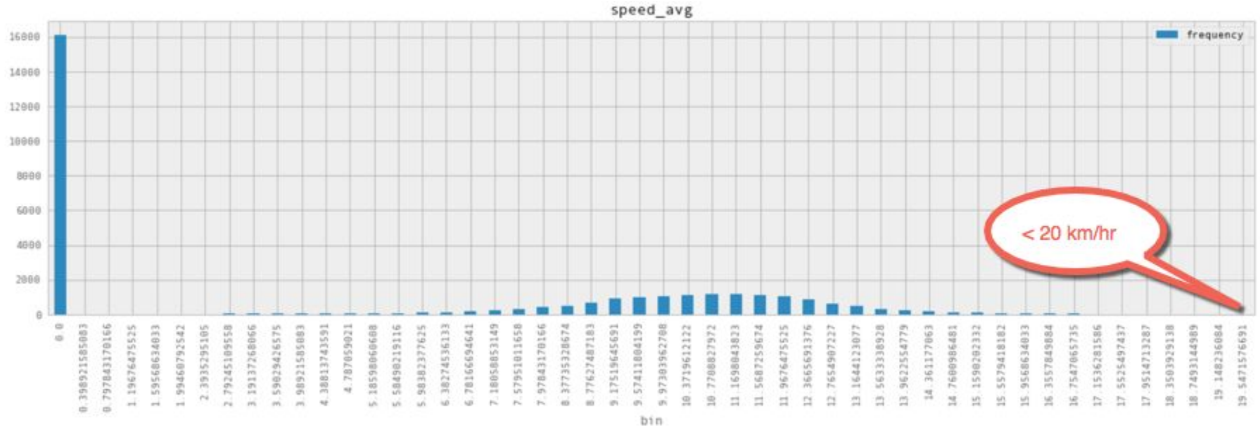


- Deltas to find series anomalies
- Remove only data at start or end of series

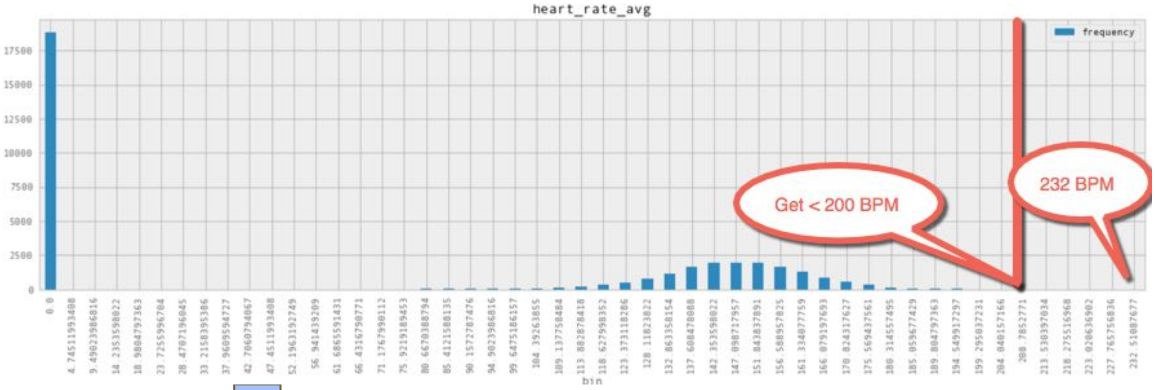
Data Cleaning - Average Speed

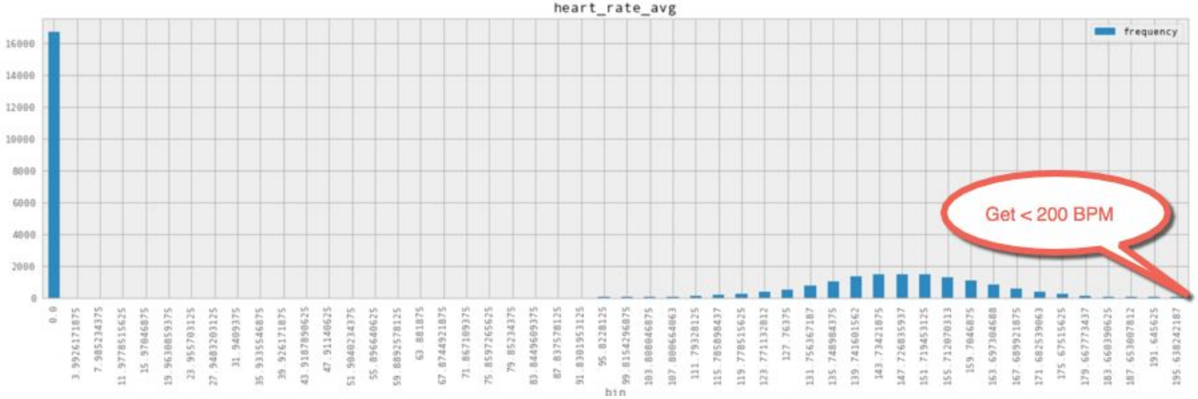


- Usain Bolt’s top speed = 27.8 MPH
- Derive new fields to find errors



Data Cleaning - Average Heart Rate

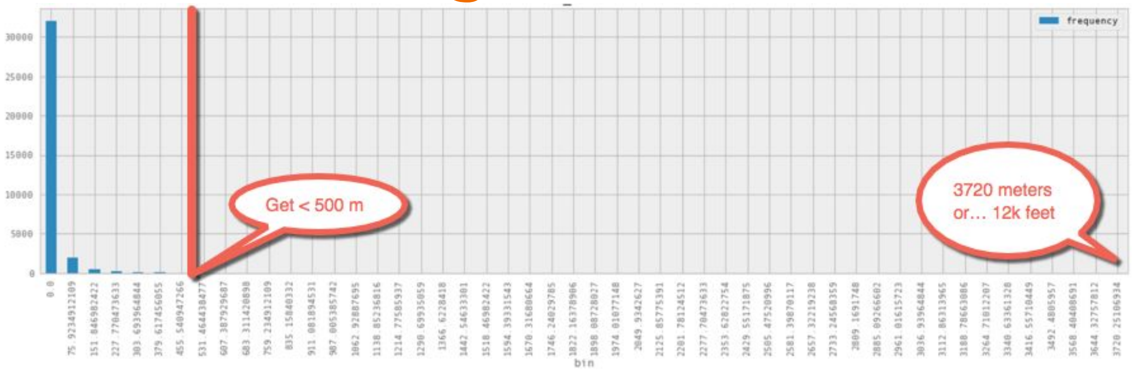




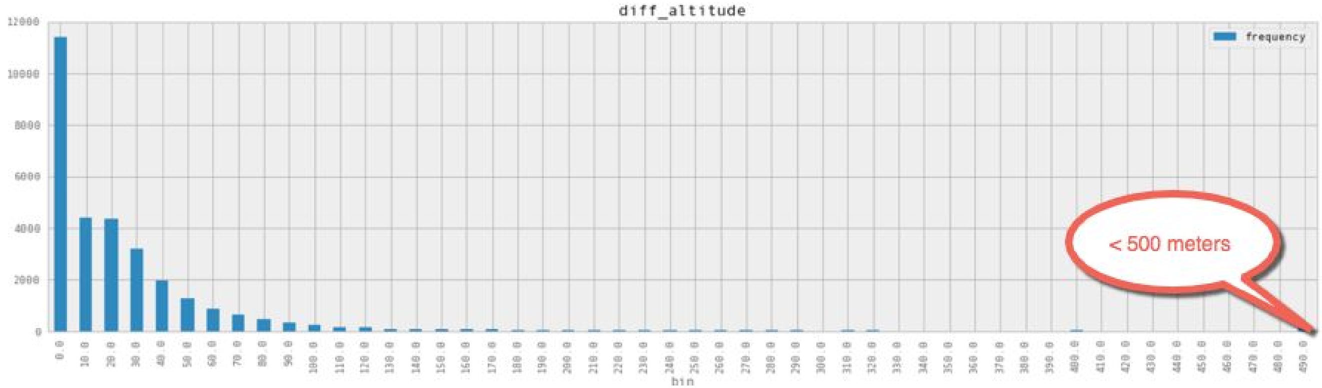
- Data that is hard to quantify why it is incorrect

- What to do with it?

Data Cleaning - Altitude Difference



- External APIs as for data verification.
- Altitude at Lat X & Long Y should be ~ Z



Clustering - Summary Stats

Time Series Data:

workoutid integer	heart_rate numeric(10,5)	speed numeric(20,10)	elapsed_time integer	geo_distance numeric(20,10)	altitude2 numeric(10,5)
197900748	215.49931	2.6064000000	0		249.24420
197900748	215.49931	2.7864000000	1	0.0000131361	249.24420
197900748	215.49931	3.3372000000	2	0.0000220603	249.24420
197900748	215.49931	4.8960000000	5	0.0000845361	249.24420

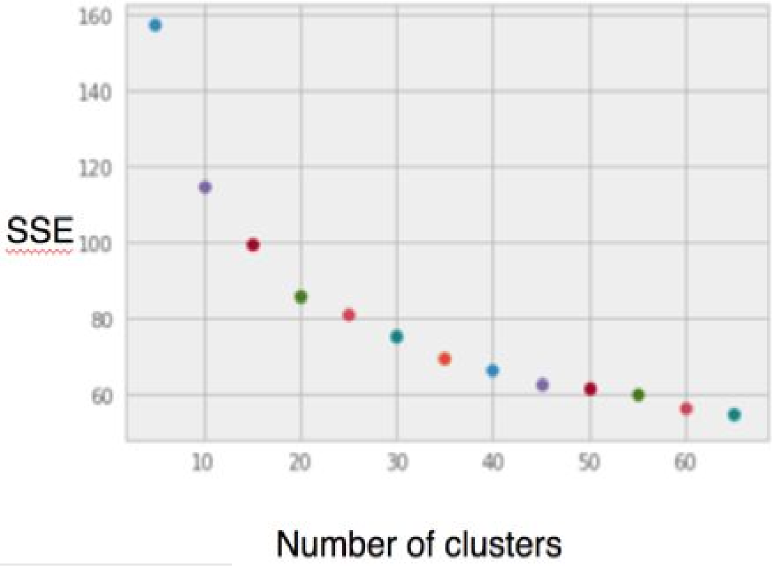
Summary statistics:

workoutid	userid	elapsed_time	diff_altitude	geo_distance	heart_rate_avg	speed_avg
197900748	1912029	1618.0	15.294235	0.05643702	195.48166	11.247107
220954943	9345869	1351.0	22.245594	0.046508357	117.810356	9.29343
246808630	1912029	2154.0	23.505724	0.06895277	169.36867	8.857457
260961987	4362441	4489.0	63.033417	0.16640514	142.12593	11.588177
273495603	324779	3395.0	169.8	0.08341786	150.04448	8.896097

Initial Cluster Analysis

- Cluster (K-Means) on all selected attributes
  - Elbow around 60 clusters
- Clustering with small K to see if results make sense

- Problems
  - Clusters doesn't make sense
    - Distance should be correlated with speed and duration.
  - Need to separate route analysis from performance analysis
  - Choose clustering algorithm/data normalization method



data in cluster	duration	distance	avg heart rat	avg speed	description
3%	6006	0.02	145	11	short distance, but long duration.
60%	514	0.03	152	11	high heart rate, but low duration/distance. Bad at pacing
30%	256	0.007	116	10	short duration and distance, slower pace. Beginners
6%	3171	0.005	145	10	medium duration, low distance with a high heart rate.
1%	9211	0.15	148	11	longest distance and longest duration. Experienced

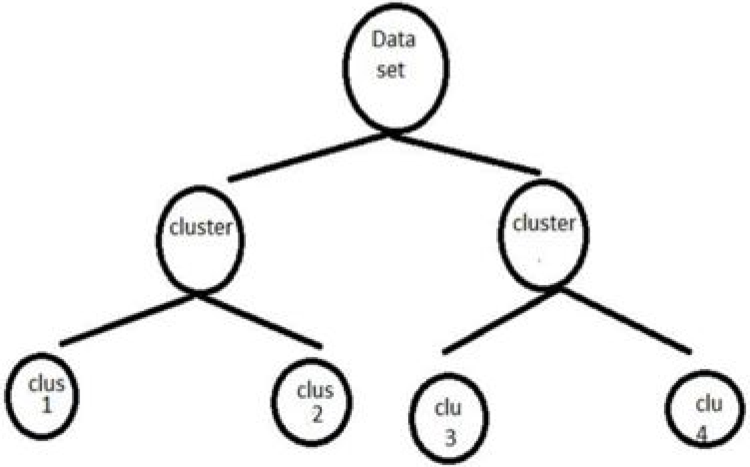
Clustering Approach

- Choosing the algorithm to use
  - K-Means (regular clustering) vs Bisecting K-Means (Hierarchical clustering)

	K-Means	Bisecting K-Means
Reproducibility	Random Initialization	Reproducible
SSE	Can fall in local minima	Tends to global minima (While testing, found 20~25% better SSE)
Performance on 10% of data with 50 clusters	~ 15 seconds	~ 25 seconds
Performance on all data with 50 clusters	~ 70 seconds	~ 70 seconds

Clustering: Bisecting K-Means

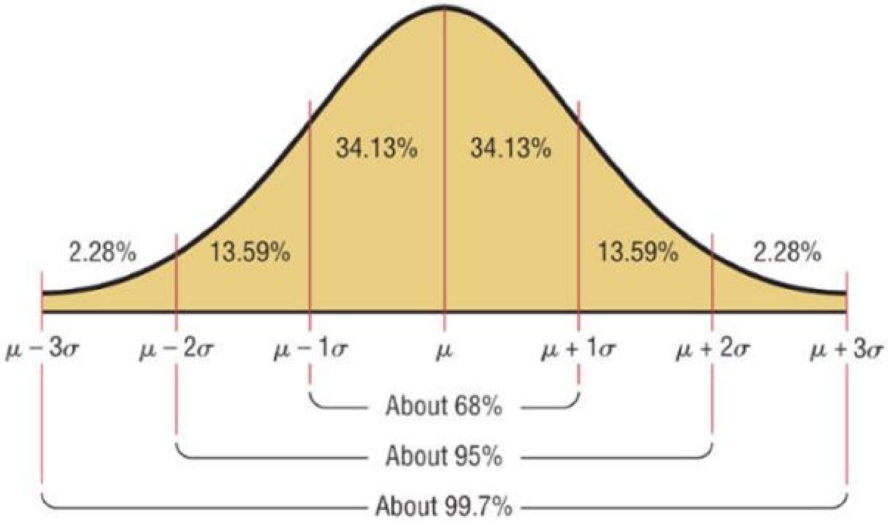
- Starts with 1 cluster with all data points
- Divides into two clusters using k-means
    - Finds two clusters with lowest total SSE
- More parallelized
    - Pyspark docs: “The bisecting steps of clusters on the same level are grouped together to increase parallelism”



Clustering tree for k=4

https://i.ytimg.com/vi/SsdaNQIEgXU/maxresdefault.j

Data Normalization

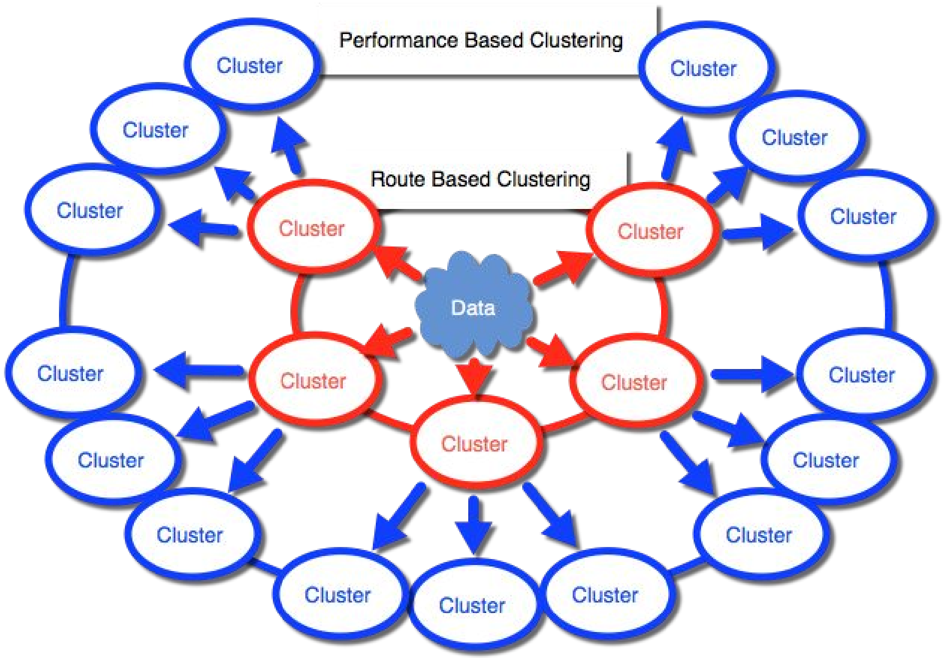


- K-Means uses euclidean distance to measure SSE
  - Which normalization to use?
- MaxAbsScaler
  - Divide values by abs(max)
- MinMaxScaler
  - Linear scaling between [min,max]
- StandardScaler
  - Removes mean and scales to unit variance

http://sites.csn.edu/istewart/Math120/Statistics/normal.ht

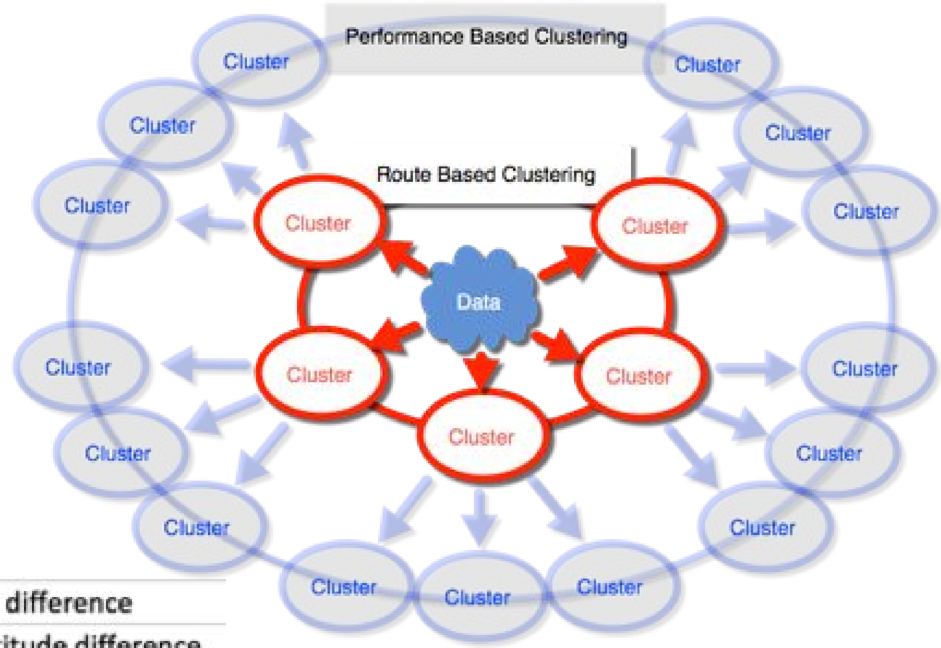
Consolidated Approach

- Add altitude as a route descriptor
- 2 - Step Clustering
  - Use bisecting k-means
  - Cluster on route info
  - Cluster further using performance info
- Normalization of data
  - StandardScaler



Route Clusters

- Create first step clusters
  - Altitude
  - distance
  - Cluster using bisecting k-means

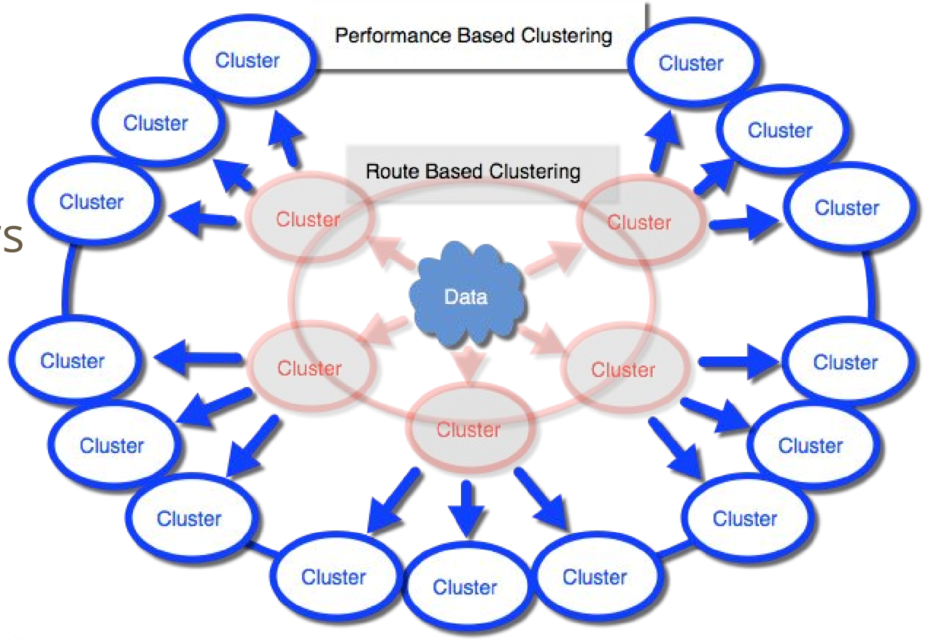


- Route Cluster Centroids

altitude	distance	description
16.20094756	2.413029375	short run with small altitude difference
26.48036589	7.96240671	medium distance with medium altitude difference
15.91379454	5.254519596	semi-short run with small altitude difference
29.11552823	12.05712951	long distance with medium-high altitude difference
99.6013325	5.276607161	high altitude difference runs

Performance Clusters

- Create performance based clusters in each route clusters
  - Duration
  - Avg Speed
  - Avg Heart Rate

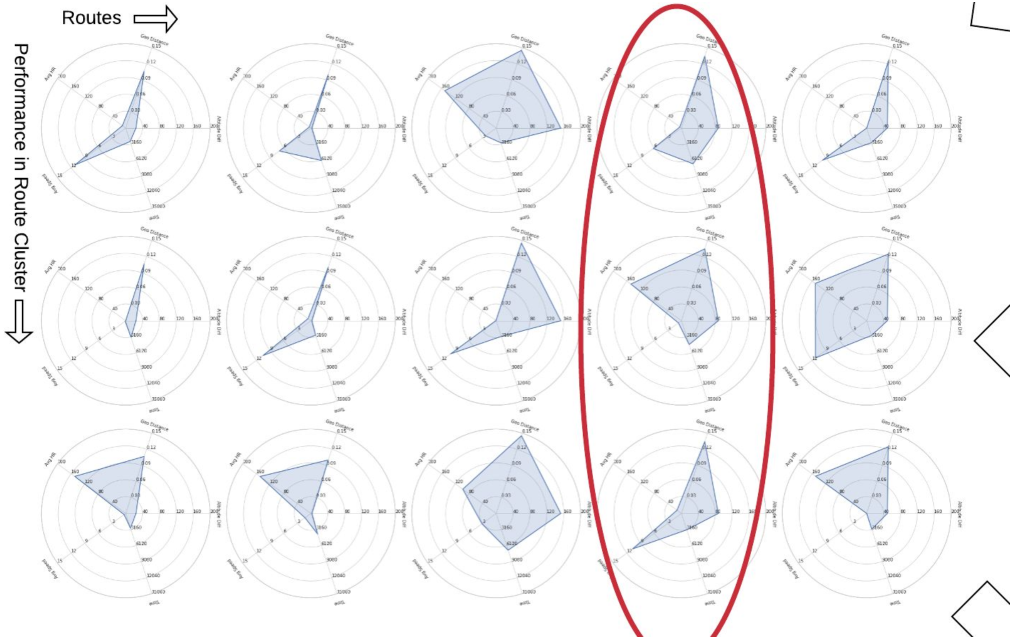


- Sample Performance Cluster

avg heart rate	avg speed	duration	description
151.8596091	2.378453066	102	high heart rate workout
0.284129298	9.376300654	105	high speed, missing lot of heart rate data
97.00516289	1.762997868	156	long duration with slower heartrate

Final Clustering Output

- Two more features created
  - User’s avg speed
  - User’s avg distance



Regression Overview

Features: $\boxed{\text{diff\_altitude, geo\_dist,}}$ $\boxed{\text{user\_avg\_speed, user\_avg\_dist}}$

Route Features — Historical User Features

Target: elasped_time

Requirements of Regression Models:

- Scalable (distributed)
- Continuous Target
- Cross-Validated

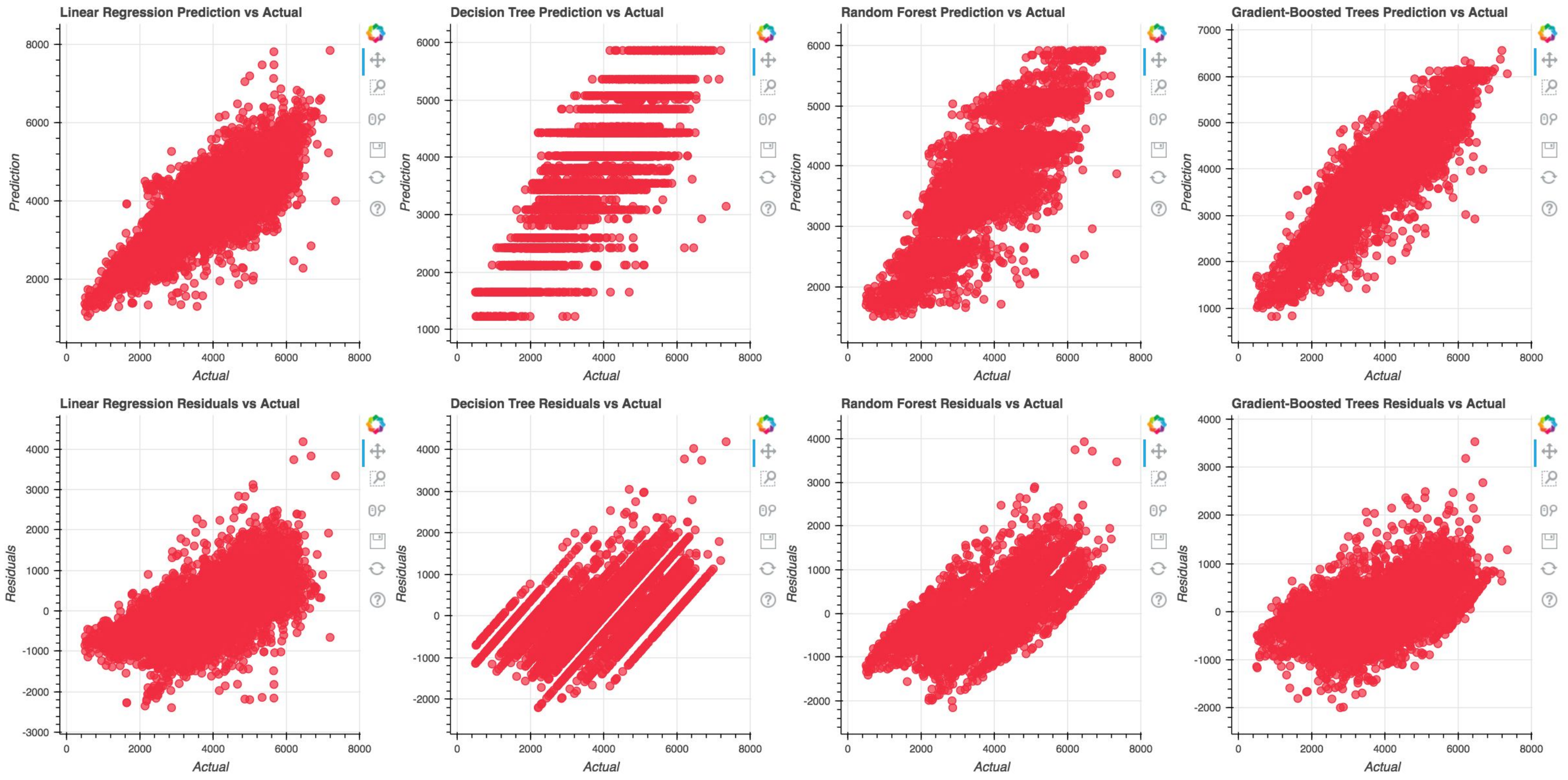
Regression Metrics:

- R^2
- Mean Absolute Error (MAE)
- Root Mean Squared Error (RMSE)

Regression Models and Parameter Maps

Model Type	Parameter Map Variable	Parameter Map Values
Linear Regression	Maximum Number of Iterations Regularization Parameter Elastic Net Parameter	[5, 10] [0, 0.1, 1, 10] [0, 0.5, 1]
Decision Tree Regression	Max Depth Minimum Information Gain	[3, 5] [0, 0.1, 1]
Random Forest Regression	Max Depth Max Iterations	[3, 5] [10,20,40]
Gradient-Boosted Trees Regression	Max Depth Number of Trees	[3, 5] [10,20,40]

Predicted vs. Actual Values and Residual Analysis



Comparison of Regression Models across Clusters

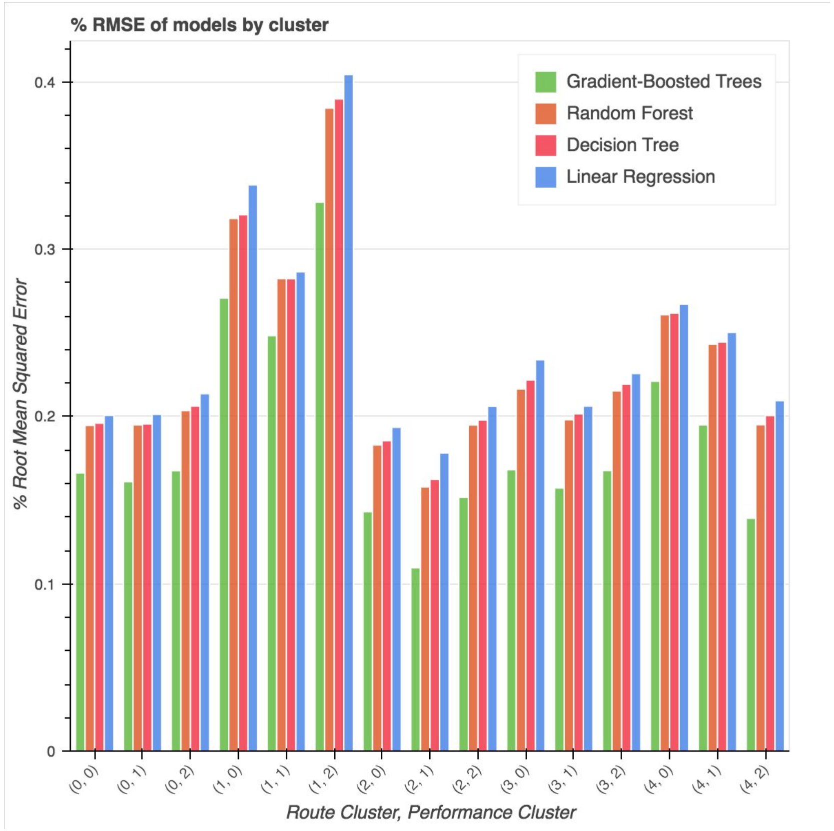
Ranking of Regression Model Types:

1. Gradient-Boosted Trees
2. Random Forest Regression
3. Decision Tree Regression
4. Linear Regression

Ranking of Prediction across Route Clusters:

1. Cluster 2 (Long Distance, Highest Altitude Change)
2. Cluster 0 (Short Distance, Low Altitude Change)
3. Cluster 3 (Long Distance, High Altitude Change)
4. Cluster 4 (Long Distance, Low Altitude Change)
5. Cluster 1 (Short Distance, Lowest Altitude Change*)

*Either altitude missing or city running



Assembling an Ensemble - Stacking

Features: Predictions of elapsed_time from each of the four best models

Target: elasped_time

Requirements of Regression Models:

- Scalable (distributed)
- Continuous Target

Regression Metrics:

- R^2
- Mean Absolute Error (MAE)
- Root Mean Squared Error (RMSE)

Stacking Method

Model: Linear Regression (from pyspark.ml)

Parameter Map: Maximum Number of Iterations: [5, 10]

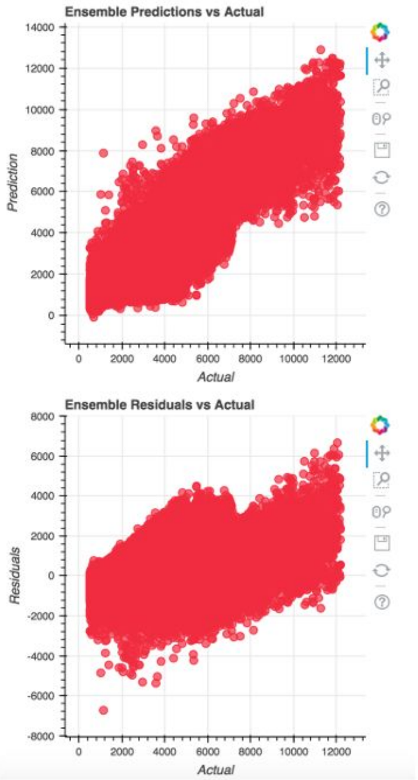
Regularization Parameter: [0, 0.1, 1, 10]

Elastic Net Parameter: [0, 0.5, 1]

Cross-Validation: 10 folds

Size of Data: 350,000 rows

Evaluating the Ensemble



Distribution of Workout Duration:

Mean: 3073 seconds

Standard Deviation: 1731 seconds

Ensemble Regression Metrics:

MAE: 406 seconds

RMSE: 595 seconds

R^2: 0.882

% MAE: 13.2% of the Mean Actual Value

% RMSE: 19.4% of the Mean Actual Value

Findings:

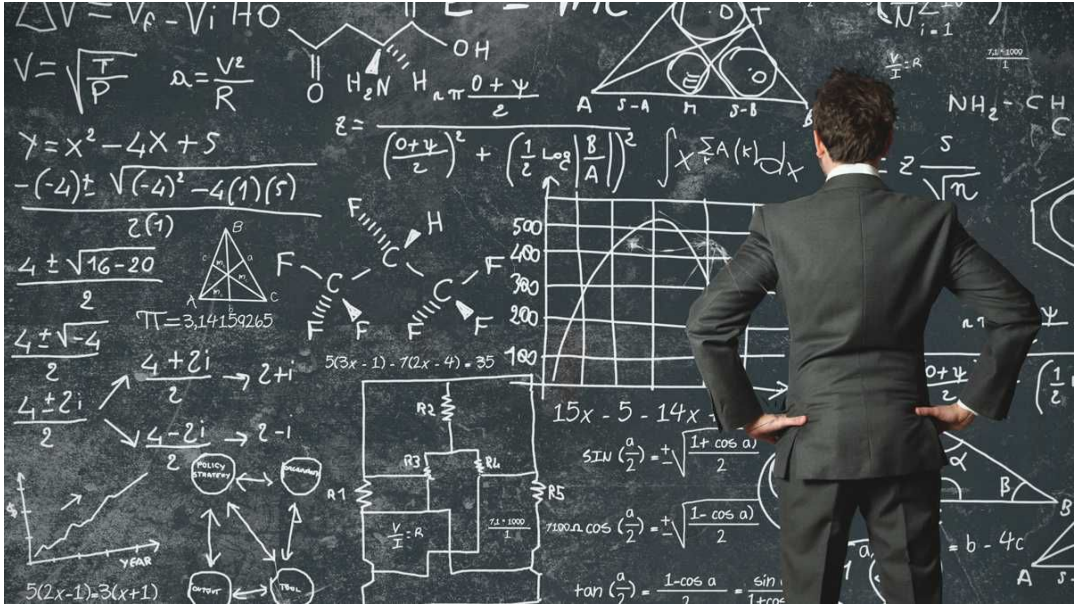
- Raw consumer wearable data is messy with many incorrect values

- Data can be cleaned with reasonable domain knowledge and used for predictive modeling with reasonably accurate results

- Clustering and predictions could be improved with more accurate and consistent data

- Lots of value hidden in the raw data

[1]



[1] - http://www.newsworks.org/images/stories/flexicontent/l_shutterstock_105432542.jpg

Recommended Changes:

- Collect additional information, such as device type
- Implement ‘gatekeeper’ sanity checking
- Deploy on-prem solution for analysis
- Consult with risk-analysis about sensitive, potentially embarrassing findings in data security

[1] - http://www.entrepreneur-resources.net/wp-content/uploads/2014/01/5-Reasons-Your-Business-Needs-To-Make-Essential-Changes.jpg

Target Consumer(s):

Target Demographic:

- Age 25 to 45 males and females
- Mobile App Users/Tech Savvy
- Active Fitness Lifestyle
- Focused on training or improvement
- With future features, expand target to include more casual runners looking for new routes

- Target Consumer
    - Runner
        - Regular Runner
            - 1 to 30+ sessions per month
        - Performance Targets
    - Race organizers
        - Route difficulty
        - Targeted advertising
        - Verifying competition times
- Consumer Features
    - Recommended Routes
    - Predicted Performance
    - Targeted Workout Plan

Future development:

Model Improvements

- Mapping weather, air quality, and more accurate altitude data
- Include private user data

Feature Development

- Expand model and predictive services to biking data
- Recommend new routes with targeted difficulty

Business Growth

- Implement custom challenges to help users accomplish their performance goals
- Create social networking opportunities through the fitness community
- Host competitions among users

Acknowledgements:

We would like to thank our friends, family, and employers for their support over the last two years.

We would like to thank all those that develop and contribute to the open software packages that made this project possible.

We would like to thank the outstanding faculty and staff that make the UCSD Data Science and Engineering program possible.

And finally we would like to thank our advisor who provided indispensable advice and guidance throughout the project.