Modeling, Prediction, Recommendation from Large-Scale Fitness Data

Project 4

Exercise Freak Consulting, LLC

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

Recap of Progress To Date

Clean Data

- Correct timestamps
- Remove extreme outliers
  - Large-scale data with no quality control
- Visualize data for cleaning and cluster sanity checking
- Create time-series data as equal lengths

Set Up Systems

- KanbanFlow
  - Organization of Tasks
  - Tracking Progress
- Postgres
  - Cleaned Data in Relational Model
  - Exploratory Analysis of Data
  - Feature Generation in Progress
- Spark
  - Framework for large-scale machine learning and clustering

Process

ABV (Always be visualizin')

1

2

Extract Summary Features

Workout (Time-Series) Dimensionality Reduction

Visualization

Cluster Workouts

Route Recommender

Cluster Route Info

Predict Workouts

Iterate

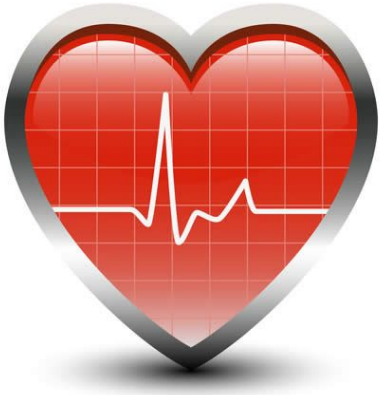
1st Iteration

Recommend routes and predict a user's performance with summarized information from workout time series and route information

- Reduce Dimensionality of Time Series with Summary Statistics
- Create Modular Machine Learning Pipeline for Process
- Start with Subset of Data, then Test Scaling with Full Data

Extract Summary Stats as Features

- Summary Stats from Time Series Data
- Examples
  - Total Distance
  - Average Heart Rate
  - Average and Max Speed
  - Average and Max Acceleration (derivative of speed)
  - Slope (derivative of altitude)



Cluster Workout Summary

- Cluster on Summary Stats of Workouts
  - Group by WorkoutID to create summary statistic dataframe.
  - Normalize values to [0,1]
- Summary Stats DataFrame Creation:
  - Aggregation types: Min, Max, Sum, Average
  - Columns: Duration, Speed, Distance, Heart rate
  - Pyspark Dataframe operations to speed up the process

```
['max_elapsed_time', 'sum_geo_distance', 'avg_heart_rate', 'avg_speed']
+----------------+--------------------+---------+------------------+------------------+
|max_elapsed_time|    sum_geo_distance|workoutid|    avg_heart_rate|         avg_speed|
+----------------+--------------------+---------+------------------+------------------+
|             908|0.014136676100000002|202885174|              null|11.894492307692307|
|            2452|         5.398919E-4|278888647|              null|10.578000000000001|
|               8|0.028470258399999993|280919215|              null|10.904192307692306|
|             320|         8.080148E-4|315716952|              null|           11.5902|
|             987|        0.0863023672|391330335| 161.9485294117647|15.580270588235297|
|             998|  0.3070946213000001|408722698|           170.122|16.148584800000013|
|             396|6.466156999999999E-4|491625790|143.33333333333334|           13.4604|
|             945| 0.04473104770000001|515235094|144.66346153846155|12.796684615384615|
```

Cluster Workout Summary

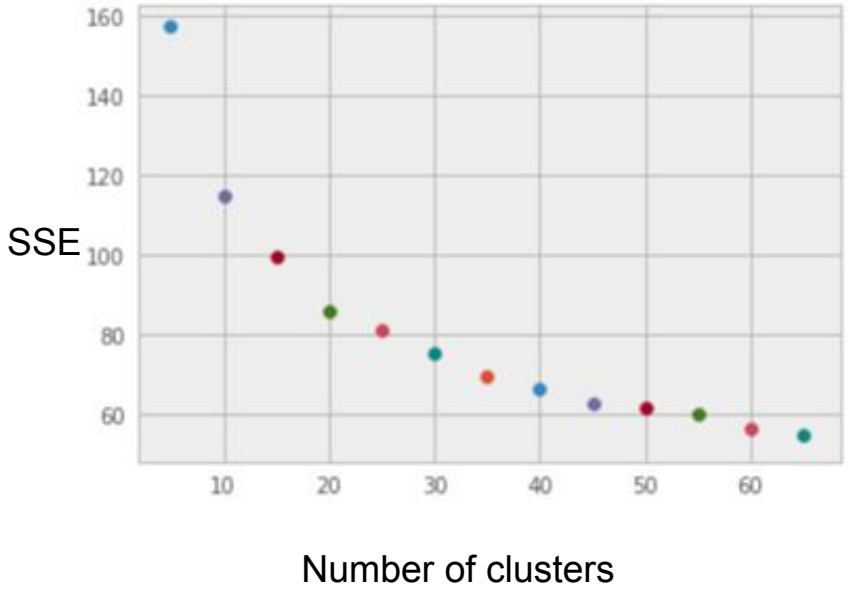
- Preliminary Results
  - Clustering with small K to observe if clustering make sense
  -

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

- Some of the cluster doesn't make sense
  - Avg speed and duration should correlate strongly to distance, but does not show here
  - Include net altitude change may help explain the clusters better

Cluster Workout Summary

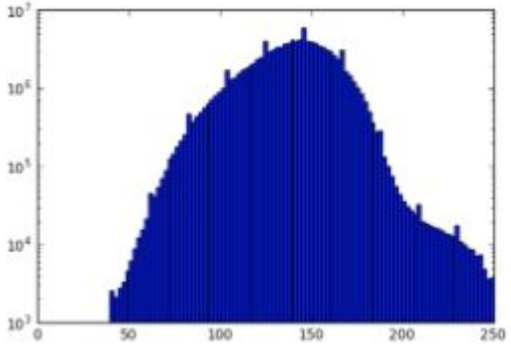
- Next steps
  - Alter normalization min/max for different fields
  - Add more features to the cluster
    - Difference between max/min, etc
  - Find optimal K (elbow curve method) for clustering



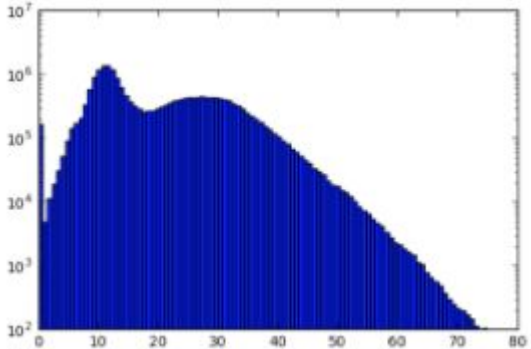
Exploratory Data Visualization

- Exploratory Analysis to Perform Sanity Check on Data
- Assist Classification of Workout Clusters
- Assist Classification of Route Clusters

Heart Rate Histogram



Speed Histogram



Route Recommender (Reco-mondo)

- Goal: Recommend future routes
- Options:

  1) Find overlapping routes in a region to use for a Latent Factor Model

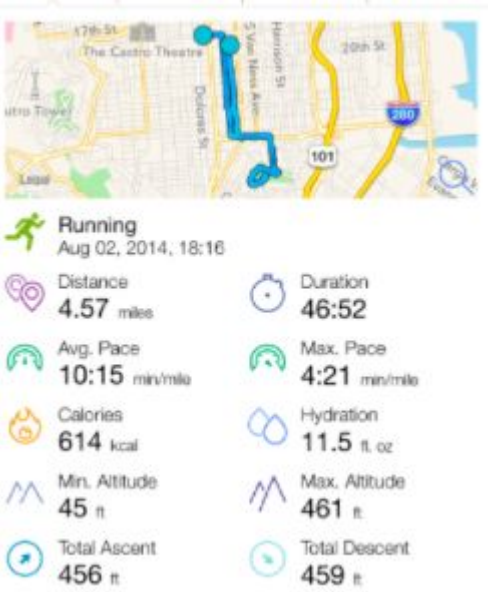
  2) Find any nearby routes using route info or outside route database

  3) Recommend altitude profiles without exact gps location such as standard run types (flat mile, flat 5K, flat half marathon, flat marathon)

- Final recommender could combine some of each of these options

Predict Workouts

- Linear Regression on Workout Summary in clusters
- Score Matrix



Goal: Predict workout duration for recommended workout

Weights: Determined by Naive Bayes (easy) or Logistic Regression (advanced)

Feature: User’s distribution of workouts within each workout cluster

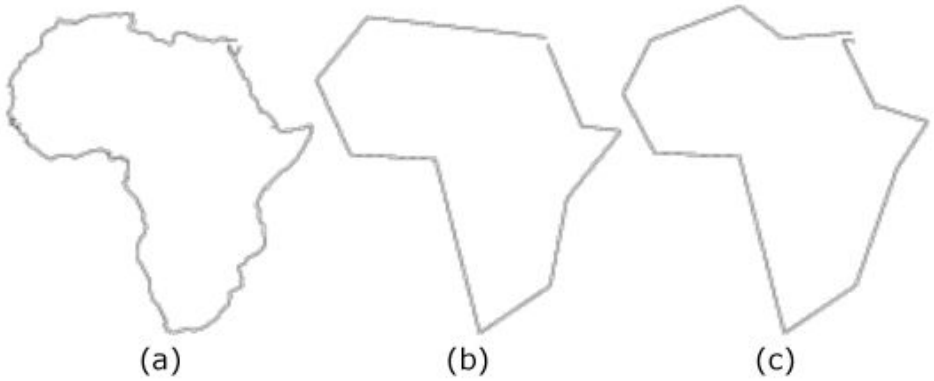
2nd Iteration

Predict workouts from clusters that utilized the full time series data after undergoing dimensionality reduction

- Utilizing full time series data
- Continuously iterating over the features (extracting features and dimensionality reduction on time series data) to improve prediction results
- Replacing existing modules with more complex algorithms

Cluster Time Series Data (1st Goal for 2nd Iter.)

- Dimensionality reduction of time series data
  - Ramer-Douglas-Peucker (RDP) algorithm to find perceptually important points (PIP)
- Cluster on PIPs



Cluster Time Series Data (1st Goal for 2nd Iter.)

- Issues
  - Python implementation of RDP only supports up to 3 dimensions.
  - Performance issue when computing RDP. Figure out how to compute in parallel.
  - Requires implementation of KMeans for time series data as initial clustering method.
    - Consider other clustering method such as dynamic time warping (DTW)