Foresee Your Activity

Andrew Cliffe, John Bruner, Mike Occhicone, Zhiyi Li

Introduction: Strava and API

Strava is an internet service for tracking human exercise with social network features.

Strava works with partners such as GarminConnect to procure source data.

Strava has heavy limitation on the API usage. Only with given permission could we access the dataset.

Questions

Can we use fitness tracking to predict an athlete's performance and activity type?

Regression Analysis

1. Can running pace be predicted based on max pace, elevation, heart rate, temp, and athlete count?

Classification Analysis

1. Do specific running features relate more to a specific time of day?
2. Can we use machine learning to accurately predict the **type of workout** based on factors such as **heart rate, start time**, and **length of workout**?

Workflow

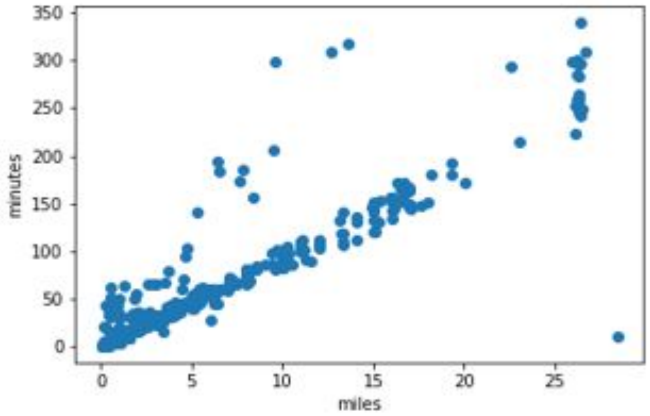
- Describe Fitness Tracking Data via Tableau
- Machine Learning Analysis
- Database Architecture using SQLite
- Frontend Design with HTML/CSS/Bootstrap

Fitness Tracking Visualization

tableau

Machine Learning: Regression

Distance & Time



sklearn.metrics mean_squared_error, r2_score

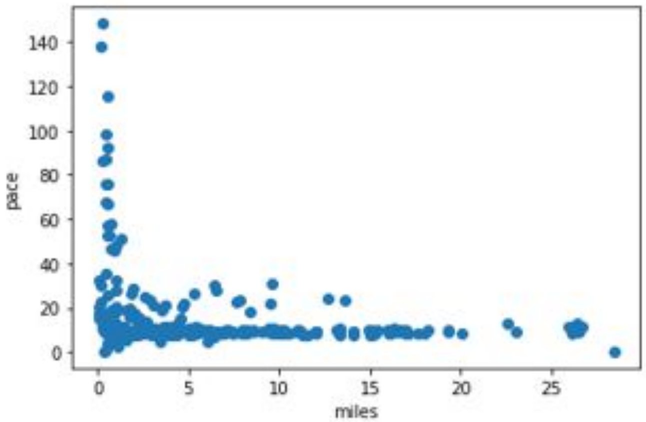
Mean Squared Error (MSE): 1001.2709321339569

R-squared (R2): **0.7361863782512363**

model.score(X test, y test)

0.736186378251236

Distance & Pace (Minutes/Mile)



sklearn.metrics mean_squared_error, r2_score

Mean Squared Error (MSE): 288.356214769644l

R-squared (R2): **0.04804724664149407**

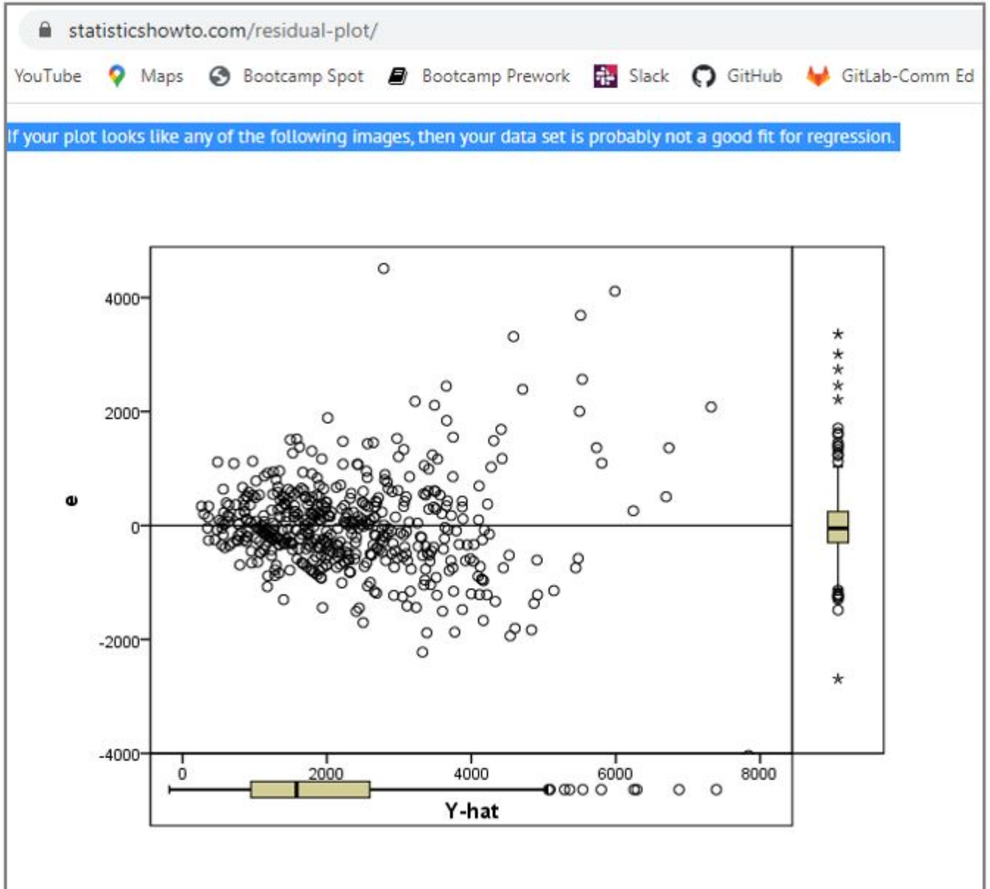
model.score(X test, y test)

0.048047246641494

Machine Learning: Regression

X Variables Modeled on Y Variable Pace (Minutes/Mile)

```
x-Axis start_h, x_mi, max_mph, total_elevation_gain,
       average_heartrate, max_heartrate, average_cadence
       elev_low, elev_high, athlete_count, average_temp

y-Axis pace
Shape:  (49, 11) (49, 1)

model.fit(X_train, y_train)
       Training Score: 0.9804983057972124
       Testing Score: 0.7967471989354613

sklearn.metrics - mean_squared_error, r2_score
       Mean Squared Error (MSE): 8.24187967547557
       R-squared (R2 ): 0.7967471989354613

model.score(X_test, y_test) = 0.7967471989354613
```



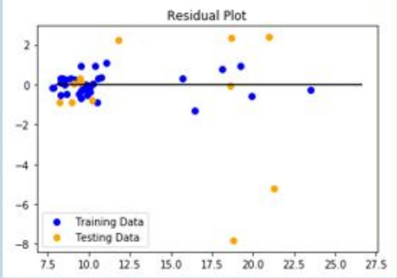
This plot of absolute residuals vs Y-hat clearly shows a heteroscedastic (cone-shaped) pattern. Image: UCLA

Machine Learning: classification

X Variables Modeled on Y Variable Pace (Minutes/Mile)

x-Axis start_h, x_mi, max_mph, total_elevation_gain,
average_heartrate, max_heartrate, average_cadence
elev_low, elev_high, athlete_count, average_temp
y-Axis pace
Shape: (49, 11) (49, 1)
sklearn.metrics - mean_squared_error, r2_score
Mean Squared Error (MSE): 8.24187967547557
R-squared (R2): 0.7967471989354613

Text(0.5, 1.0, 'Residual Plot')



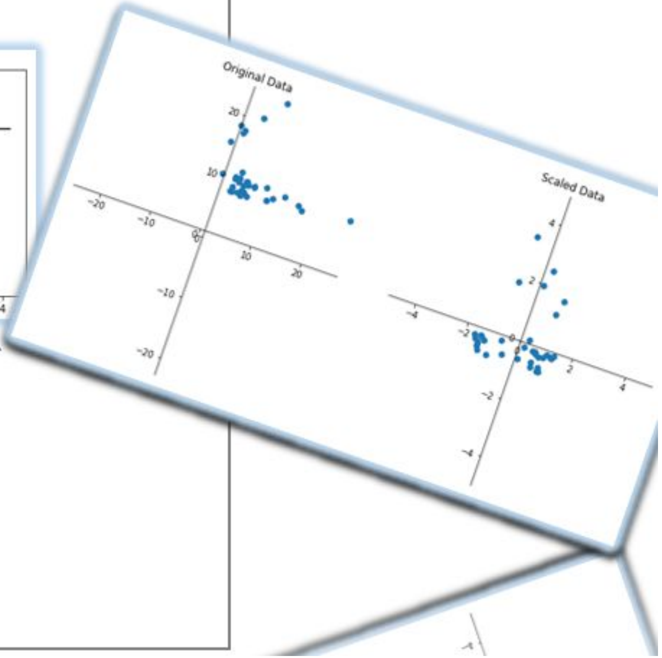
model.fit(X_train, y_train)
Training Score: 0.9804983057972124
Testing Score: 0.7967471989354613

model.score(X_test, y_test) = 0.7967471989354613

Scaled X Var Modeled on Y Var Pace (Minutes/Mile)

x-Axis start_h, x_mi, max_mph, total_elevation_gain,
average_heartrate, max_heartrate, average_cadence
elev_low, elev_high, athlete_count, average_temp
y-Axis pace
Shape: (49, 11) (49, 1)
sklearn.metrics - mean_squared_error, r2_score
MSE: 0.5754050467855296
R2: 0.7967471989354646

sklearn.metrics - mean_squared_error
MSE: 0.5754050467855296
R2: 0.7967471989354646
LASSO model
MSE: 0.5409298615265087
R2: 0.8089250169964288
Ridge model
MSE: 0.5744471330178381
R2: 0.7970855669382293
ElasticNet model
MSE: 0.5538689392136826
R2: 0.8043544908986798



Machine Learning: Time of Day Classification

Classification

Features Affecting Time of Day (Morning, Afternoon, Evening)

Let's Include the Start Hour

```
x-Axis start_h, x_min, x_mi, total_elevation_gain,
      average_cadence, elev_low, elev_high, athlete_count,
      athlete_count, pace
y-Axis name_short      Morning, Afternoon, Evening

tree-DecisionTreeClassifier
      clf.score(X_test, y_test) = 0.9908256880733946

sklearn.ensemble RandomForestClassifier
      rf.score(X_test, y_test) = 0.981651376146789

Feature Importances
      [(0.6374950313456574, 'x_start_h'),
       (0.07878299134365213, 'x_min'),
       (0.07049855266500368, 'x_mi'),
       (0.05635676293222627, 'average_cadence'),
       (0.049016802843989224, 'elev_high'),
       (0.04459163327779931, 'elev_low'),
       (0.03453829710197454, 'total_elevation_gain'),
       (0.024306727441764216, 'pace'),
       (0.004413201047933222, 'athlete_count')]
```

Features Affecting Time of Day (Morning, Afternoon, Evening)

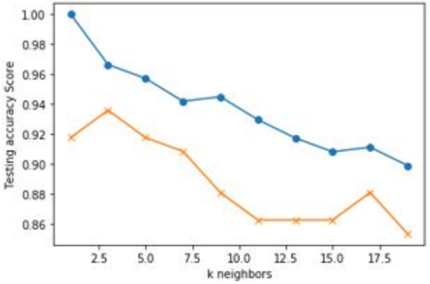
Let's Not Include the Start Hour

```
x-Axis x_min, x_mi, total_elevation_gain, average_cadence,
      elev_low, elev_high, athlete_count, athlete_count,
      pace
y-Axis name_short      Morning, Afternoon, Evening

tree-DecisionTreeClassifier
      clf.score(X_test, y_test) = 0.6697247706422018

sklearn.ensemble RandomForestClassifier
      rf.score(X_test, y_test) = 0.6146788990825688

Feature Importances
      [(0.176293282781754, 'x_mi'),
       (0.16119468680784835, 'x_min'),
       (0.14757054356361138, 'elev_high'),
       (0.13706006377785376, 'average_cadence'),
       (0.12839558070435464, 'elev_low'),
       (0.12601173693380313, 'total_elevation_gain'),
       (0.10330339153043291, 'pace'),
       (0.020170713900341884, 'athlete_count')]
```

Machine Learning: Classification

Neighborly Relations k in range(1, 20, 2)

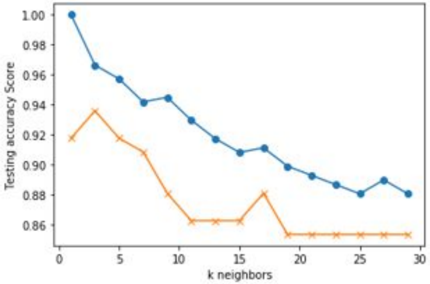
```
x-Axis start_h, x_min,  x_mi, total_elevation_gain,
       average_cadence, elev_low, elev_high,
       athlete_count, pace
y-Axis name_short      Morning, Afternoon, Evening

for k in range(1, 20, 2):
       k: 1, Train/Test Score: 1.000/0.917
       k: 3, Train/Test Score: 0.966/0.936
       k: 5, Train/Test Score: 0.957/0.917
       k: 7, Train/Test Score: 0.942/0.908
       k: 9, Train/Test Score: 0.945/0.881
       k: 11, Train/Test Score: 0.929/0.862
       k: 13, Train/Test Score: 0.917/0.862
       k: 15, Train/Test Score: 0.908/0.862
       k: 17, Train/Test Score: 0.911/0.881
       k: 19, Train/Test Score: 0.899/0.853
```




```
k=15 Test Acc: 0.862
```

Neighborly Relations k in range(1, 30, 2)

```
x-Axis start_h, x_min,  x_mi, total_elevation_gain,
       average_cadence, elev_low, elev_high,
       athlete_count, pace
y-Axis name_short      Morning, Afternoon, Evening

for k in range(1, 30, 2):
       k: 1, Train/Test Score: 1.000/0.917
       k: 3, Train/Test Score: 0.966/0.936
       k: 5, Train/Test Score: 0.957/0.917
       k: 7, Train/Test Score: 0.942/0.908
       k: 9, Train/Test Score: 0.945/0.881
       k: 11, Train/Test Score: 0.929/0.862
       k: 13, Train/Test Score: 0.917/0.862
       k: 15, Train/Test Score: 0.908/0.862
       k: 17, Train/Test Score: 0.911/0.881
       k: 19, Train/Test Score: 0.899/0.853
       k: 21, Train/Test Score: 0.893/0.853
       k: 23, Train/Test Score: 0.887/0.853
       k: 25, Train/Test Score: 0.880/0.853
       k: 27, Train/Test Score: 0.890/0.853
       k: 29, Train/Test Score: 0.880/0.853
```




```
k=17 Test Acc: 0.881
```

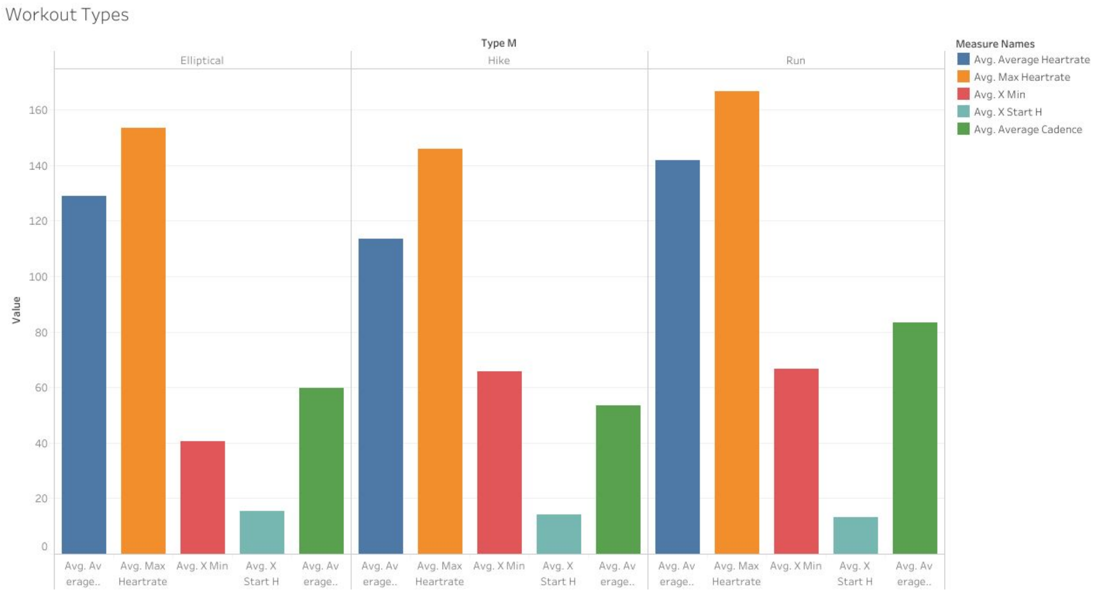
https://towardsdatascience.com/how-to-find-the-optimal-value-of-k-in-knn-35d936e554eb
https://www.youtube.com/watch?v=j-uHsi-KjOs

Machine learning: Classification Workout Type

Dataset: 528 records, (records after 12/2/2019 had to be manually reclassified)

Features: Start Time, Length of Workout, Max Heart Rate, Average Heart Rate

Output: Workout Type: (Run, Hike, Elliptical)

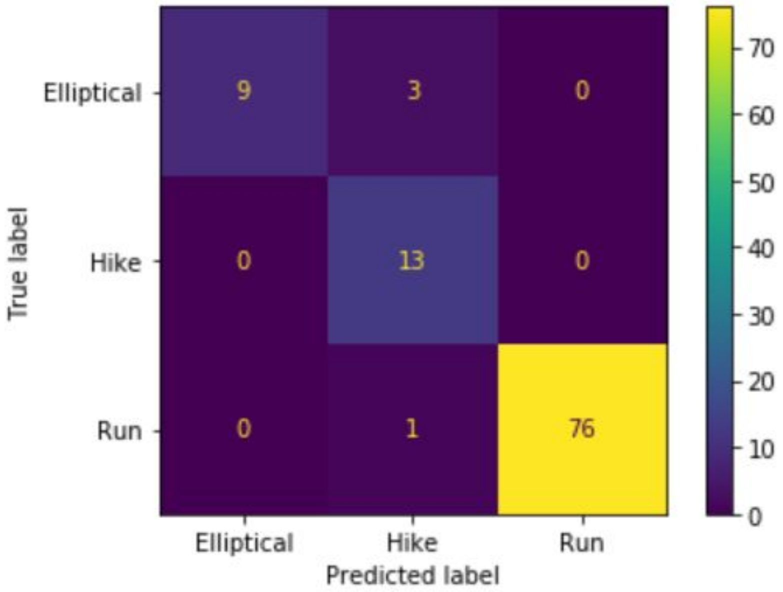


Machine learning: Classification - Workout Type

Results:

Model	Train Score	Test Score	Precision	Recall	F1 Score
Logistic Regression	.95	.93	.93	.93	.93
Random Forest	.94	.96	.94	.94	.94
Deep Learning	.97	.95	.95	.95	.95
KNN	.98	.96	.97	.96	.96
Support Vector	.95	.97	.97	.97	.97

Support Vector Confusion Matrix



Machine learning: Classification Take 2 - Workout Type

Datasets:

- Train Data (< 12/03/2019)

Runs	281
Hikes	24
Elliptical	3

- Test Data unclassified (>= 12/03/2019)

Runs	61
Elliptical	39

- Test Data corrected, manually classified (>= 12/03/2019)

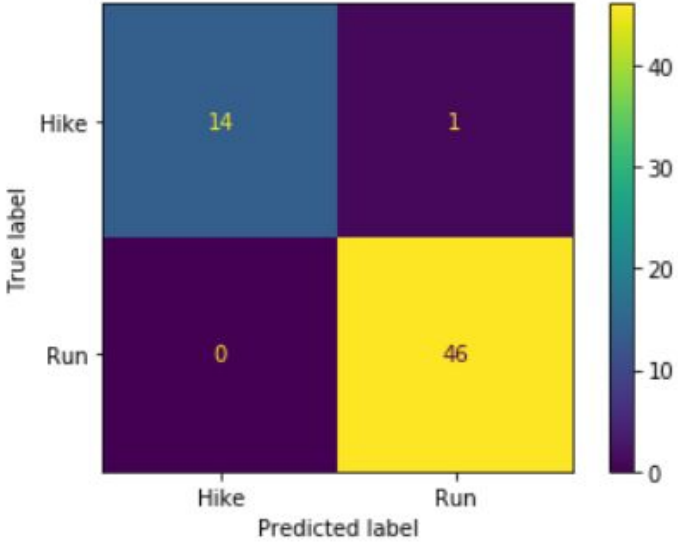
Runs	46
Hikes	15
Elliptical	39

Machine learning: Classification Take 2 - Workout Type

Results:

Model	Train Score	Test Score	Precision	Recall	F1 Score
Support Vector (with Elliptical)	.97	.62	.84	.62	.50
Support Vector (without Elliptical)	.94	.96	.94	.94	.94

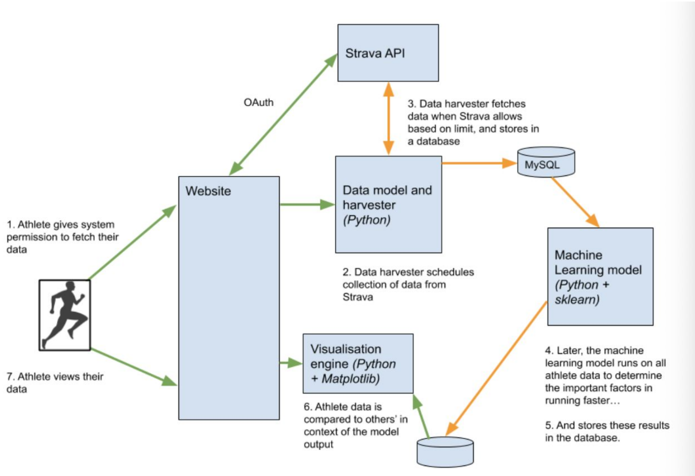
Support Vector Confusion Matrix (Without Elliptical)



Foresee Your Activity (2.0)

Once we get more funding, our machine learning app will be able to predict more activities based on fitness tracking data that is provided by users

- Database Architecture
- Website Design



SQL Database

SQLite is library that implements a small, fast, self-contained, high reliability and full-featured SQL database engine.

The database is not much used in this project.

However, it could be useful for the future project to stay organized and keep information easily accessible.

SQL Database Creation

1. Extract the data

Extract the Data from Strava API and then convert the Excel the file into CSV file

2. Transform the data

There are over 30 columns in the datasets. We find some datasets are reduplicate, such as distance data in both kilometers and miles, or some columns have a lot of values missing. Only selected columns are kept for future analysis

SQL Database Creation

3. Load

```
#create connection to sqlite database
connection_string = "sqlite:///fitness_database.sqlite"
engine = create_engine(connection_string)
```

```
#create tables in database
engine.execute("CREATE table garmin_ac (activity varchar,
```

```
<sqlalchemy.engine.result.ResultProxy at 0x7faf179a0750>
```

```
garmin_df.to_sql(name='garmin_ac',con=engine, if_exists ='append',
```

```
garmin_data = engine.execute("select * from garmin_ac")
```

Limitations

- Source data included 2 individuals for analysis
- Strava has rigorous permission restrictions on obtaining data
    - Not easy to get one's own Strava data emailed
- 3rd-party website Torben's Strava Äpp was used to obtain Strava data
    - https://entorb.net/strava/

Current Website Under Development

The general structure of the website is complete, but we need investment to complete the app!