Fitbit Calories Burned Prediction
Springboard DSCT Capstone Project II
By: Mindy Ng
August, 2017

The Problem†

In recent years, there has been a movement called the "quantified self", which according to wikipedia.org is also known as lifelogging, is a movement to incorporate technology into data acquisition on aspects of a person's daily life in terms of inputs (food consumed, quality of surrounding air), states (mood, arousal, blood oxygen levels), and performance, whether mental or physical. In short, quantified self is self-knowledge through self-tracking with technology."

The problem is what to do with all these body sensor readings? One answer would be for users to take initiative in improving their health. A popular measure of a person's health is the number of calories the person burns per day. This would be helpful for a person who wants to lose weight. And monitoring the number of calories burned per day would be a way of measuring one's progress about that goal. Even better, one can determine which combination of activities a person could perform to burn the most calories. This would be useful in order to use one's time and effort more effectively throughout the day to reach personal exercise goals. In this way, a Fitbit device user can take initiative in their own healthcare. My capstone project's aim was to determine crucial activities that contribute to high calorie counts. For example, based on a person's activity log for a certain month, a model can be built and used to predict the activity levels needed in order to burn a certain amount of calories.

Description of Potential Client and Their Motivation†

The potential client is anyone who is concerned about their personal activity levels. A classic case would be someone who just came from a doctor's visit and has been diagnosed with diabetes. If it is Type II diabetes, then one of the causes could be a sedentary lifestyle that can be improved by being more active.

The motivation for the client could be that they now have personal activity level measurements. When they see their physical activity through tangible numbers, they could gradually increase the amount of activity per day. This could translate into going to the gym more frequently or simply into increasing the number of steps a user takes per day by going on more walks and therefore spending less time sitting. And in doing so, a person can lose weight and get into a healthier weight bracket, which would potentially contribute to getting them out of diabetic/obese category.

Formulation of this Project as Data Science Problem†

The dataset was fetched by determining all the available activity measurements from Fitbit through their web API, and then picking which measurements seemed to contribute most to the

metricknown as “value_activityCalories”. A total of 6 features were chosen: “elevation”, “activityCalories”, “fairlyActiveMinutes”, “floors”, “steps and “veryActiveMinutes”. Each feature was grabbed from Fitbit via its web API and stored as an individual JSON file containing data that spanned from the beginning of the month of April to the end of July.

This was made into a Pandas dataframe that was split into a training set and a test set. The training set was fed into various regression models, which were eventually used to predict on the test set data. The output were predictions which were measured for their accuracy against known values.

Whichever model had the best RMSE (Root-Mean-Squared-Error) on the test set, not too distant from the RMSE on the training set, was considered to have the best prediction. And this would be used to help the client predict calories burned, given the values of the predicting features.

The Dataset†

Each variable was stored as a JSON file spanning from April 1st, 2017 to July 31st, 2017. Each file was read into a Jupyter notebook and transformed into a Pandas dataframe. Once each feature became a dataframe, common table manipulations such as “join” and “merge” were performed to create a big dataframe of a feature set. Eventually, the data became a large table with the “dateTime” column dropped. The predicting features were: “elevation”, “activityCalories”, “fairlyActiveMinutes”, “floors”, “steps” and “veryActiveMinutes”. The target variable was “activityCalories”. And the rest were 5 different variables that contributed to the target variable. With this set up, various Regression models were built using the training set (data from April to June). The test data was then used to predict “activityCalories” for the month of July.

Some Caveats About this Dataset

Measured values are not specific to a certain activity such as “Elliptical” or “Running” unless an activity is programmed to categorize data recording when exercise/activity is performed. Therefore, if the user wanted to repeat the activity that resulted in high calorie count, the user would need to determine through trial and error which activity caused high calorie counts.

Justifications for Using this Dataset

Though this project does not specify a certain gym workout, it does determine the combination of features/activity needed in order to produce a certain calorie count. Therefore, although the

data set is not specific to the point of recommending an exercise machine, the sort of activity needed to perform to achieve high calorie counts is still answered.

Data Wrangling†

I accessed current Fitbit data through their API. I made separate JSON files for each feature that seemed to contribute most to calorie count. Then each JSON file was imported into a Jupyter notebook and transformed into its own Pandas dataframe. Ended up using "join's" and other combination functions to create main table with all relevant features for the problem.
In order to create the Training set and Test set, the 'dateTime' column had to be dropped. What was left was a dataframe with 6 features. The Training set and Test set were fed into all three Regression models. Once the model was built, a prediction was made which essentially answered the question, "How much activity does it take to burn a certain amount of calories?"

Performing Regression using Linear Regression Model

Linear Regression models were chosen because there is a trend with increased activity level, calories increase as well. Therefore, calorie count was modeled as a linear function. From Hands-On Machine Learning with Scikit-Learn & Tensorflow, "a linear model makes a prediction by simply computing a weighted sum of the input features, plus a constant called the bias term (also called the intercept term)." This is exactly what was done with to the data set in order to arrive at the predictions.

After the model was built upon the training set, the test set was fed into the model. And the result were predictions that were compared to the actual results. Most of the time, the differences between the actual results and the predictions were not too big. In the same book from above, it is stated that "the most common performance measure of a regression model is the Root Mean Square Error (RMSE)." RMSE values and residuals are to be looked at to determine how accurate the model is. After running the Linear Regression model, the RMSE values for the Training set was: 62.732425178623323 and the Test set was: 51.104461388814308. L.R.'s RMSE values are high, but not the highest. Also, it has a relatively small gap between its Training RMSE and Test RMSE values. This means that the model was trained to be general enough, by minimizing its cost function and overfitting was not present. However, one downside to this model is that its Training RMSE value is higher than its Test RMSE value. Usually, it is the other way around. Though there are ways to improve this, which are discussed at the end.
Other regression models were run in an attempt to improve RMSE results by lowering them.

Improvement to this model would be: 1. Cross-validate/reshuffle the data 2. Have a larger Training set/Bigger Test set 3. Fewer parameters to make Training set less noisy 3. Constraining

model. At the same time, it could have random chance that the Test data just happened to fit better to the model than the Training set was able to do.

Performing Regression using Decision Tree Regression Model

Decision Tree Regressor was chosen because according to Aurelien Geron in Hands-On Machine Learning with Scikit-Learn and TensorFlow they are capable of fitting complex datasets. This model is also the fundamental component of Random Forests, which are among the most powerful Machine Learning algorithms available today." It works by starting at a root node (depth 0) which asks a question based on a feature from the model's feature set. Based on the answer from the question, the branch moves down to the left or the right child node (depth 1). And this goes on until the tree's max depth is reached, which is specified as a hyperparameter when setting up how to run the Decision Tree model. When the max depth is reached, a classification decision is made.

After running the Decision Tree model, the RMSE value for the Training set was: 33.58454696540948. This was a significant change after doing GridSearchCV to optimize max_depth from 7 to 5. It is common for Decision Trees to overfit due to the model being set to default parameters, which means less restrictions on the model. This model has the lowest RMSE value, which comes from its Training data set. Though D.T.'s are known to fit its data very well. Given this close fit, the Test RMSE value suffered with huge RMSE value. Thus, the gap between its Training and Test set RMSE values is huge. Though a good side to this is that its Training RMSE is lower than its Test RMSE value, which is expected.

Thus, one improvement to this model was to change the max_depth parameter in the Decision Tree model set up so that it would not go so low in the tree (set max_depth to lower number). This enabled the model to be more general. And thus, the RMSE value for the Training set is relatively better in the sense that there is less of a gap between Training RMSE and Test RMSE values. The greater the max_depth setting tends to constrict the model further than needed, making a fit on unseen data very difficult. Thus, another model, Random Forest Regressor, which is an ensemble of Decision Trees, was considered.

Perform Regression using Random Forest Regression Model

As mentioned before Random Forest Regression Model is an ensemble method. An ensemble method is a group of predictors, which is advantageous. And as Aurelien Geron in Hands-On Machine Learning with Scikit-Learn and TensorFlow said, "If you aggregate the predictions of a group of predictors, you will often get better predictions than with the best individual predictor. Despite its simplicity, this is one of the most powerful Machine Learning algorithms today."

Straight from the same reference book: “They work by being trained via the bagging method typically with max_samples set to the size of the training set...The Random Forest algorithm introduces extra randomness when growing trees; instead of searching for the very best feature when splitting a node, it searches for the best feature among a random subset of features. This results in a greater tree diversity, which trades a higher bias for a lower variance, generally yielding an overall better model.”

After running the Random Forest Regression Model again by performing GridSearchCV (cross-validation) and retrieving oob_score after the model was built. The Training set R^2 value was 0.54101403158296468. Random Forest model's assessment is different in the sense that for Training error rate, the R^2 value is reported. Given that an R^2 value of 1means that the regression line perfectly fits the real data points, approximately half of the regression points fit the actual data points. Though, its Test RMSE value is comparable to L.R.'s Training and Test RMSE values.

In the end, Linear Regression model wins as the best model because it has relatively the lowest RMSE values, has minimal gap between its Training RMSE and Test RMSE values. Thus, it does not overfit its data.

Recommended Use of Regression on Fitbit health data†

This model can be used to predict the sort of activities needed in order to burn a certain number of calories. It is more common these days to count how many calories you are burning, but with a predictive model, a Fitbit user can forecast their exercise regimen. They can choose the activity variables that are tracked on their Fitbit to determine which combinations result in high calorie burn. This would allow them to be more effective in exercising. Essentially, this would allow users to see which sort of activities output the greatest calorie burn. In general, any business that sells a device to track a person’s performance or even eating habits would benefit from building a regression model applied on a user’s personal health data.

Concrete Recommendations On How to Use My Findings†

a. Use regression model to predict a person’s overall health, which includes the sort of food intake and exercise would result in a certain weight loss/gain.
b. If there was a device to measure one’s mood, this model would also be used to predict when someone would be the happiest and therefore, most approachable during the day or the opposite in an attempt to modify their own behavior to be more genial.

Potential Next Steps†

a. Answer other inverse questions after building Linear Regression model on some features to predict target value. Other questions to answer are:
    i. If oxygen levels are known throughout day, which time during the day is best to exercise?
    ii. If mood levels are known throughout the day, what are the times when I am not the most upbeat in order to go on a walk, get a cup of coffee.
b. Data-related:
    i. Get live data via Fitbit API and predict with most current user data
    ii. To remedy Trending error rate being higher than Training error:
        1. Cross-validate, split/reshuffle the data differently so that the Test data is not so small
        2. Have fewer parameters to reduce possible noise in Training data set
        3. Constrain the model so there are less errors
        4. Gather more Training data
        5. Lastly, it could have just been random chance that the Test data was able to fit the model better than the Training data

Conclusions/Summary†

Due to the "quantified self" movement, a need has arisen in figuring out what all the personal sensory data means. One way to extract useful conclusions from "lifelogging" is to build a regression model like the one I have built in this project in order to predict calories burned, from the values of other predicting variables. This is huge business for companies to help their clients/users understand and utilize their data better to take their own initiative in improving their health/remain fit. And these regression models I have built can be used to develop comprehensive data products to address these needs.