**Data Science Project: Predicting “Sedentary” Lifestyle**

**Question:** Can we identify people with a "Sedentary" lifestyle based on the other information given in the “sign up” process?

Business Objective: To create a Machine Learning model able to predict if a user has a “sedentary” lifestyle based on the information given on the “sign up” process with at least 0.95 accuracy.

**Hypothesis**

H0: The ML model isn’t able to predict if a person has a “Sedentary” lifestyle based on the information given in the “sign up” process.

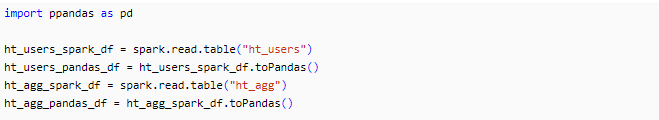
HA: The ML model is able to predict if a person has a “Sedentary” lifestyle based on the information given in the “sign up” process.

**Experiment/Analysis**

Information: The data provided to the model comes from the “ht\_users” and “ht\_agg” tables, which will be converted to Spark DFs and then to Pandas DFs, for their further use. Each one of this tables have 3000 rows and consist in the next columns:

TABLE 1. Table’s Schema

|  |  |
| --- | --- |
| ht\_users | ht\_agg |
| device\_id: string  first\_name: string  last\_name: string  lifestyle: string  country: string | device\_id: string  mean\_bmi: double  mean\_active\_heartrate: double  mean\_resting\_heartrate: double  mean\_vo2: double  mean\_steps: double  lifestyle: string |

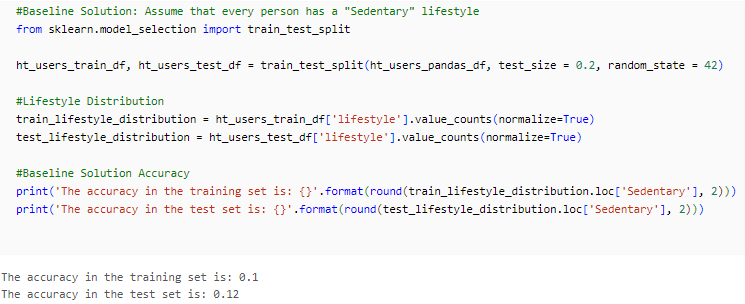


*Fig. 1 Pandas DataFrames Creation*

Baseline Solution

Now that we have framed the business objective and we know the data available for our analysis, it is time to develop our solution. In this case, we will start with a baseline solution an assume that every person has a “Sedentary” lifestyle, since it is the value and feature in which we are interested.

In order to apply the base solution, the methodology followed consisted in splitting the data into Training and Test sets. Then, we calculate the distribution, to get the percentage that each lifestyle represents in the sets. And finally, we determine the accuracy of the model, by getting the percentage of the people who has a “Sedentary” Lifestyle.



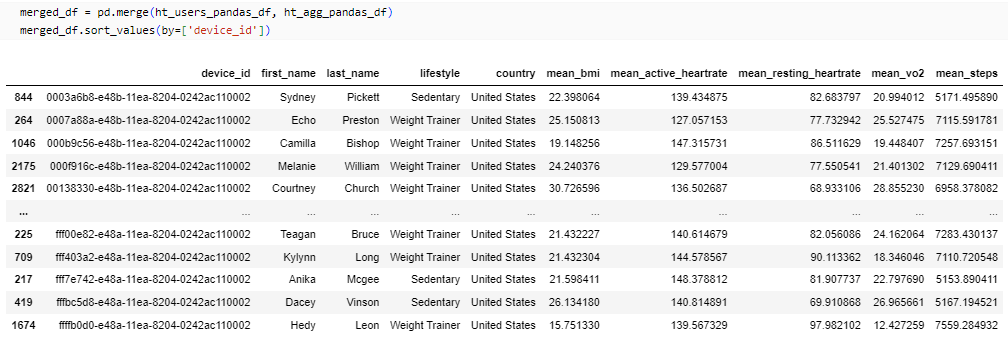
*Fig. 2 Baseline Solution*

The results obtained were that the training and test set had an accuracy of 0.1 and 0.12 respectively, which failed our business objective of predicting a “Sedentary” lifestyle with 0.95 accuracy. For this reason, the baseline solution is discarded, and from now on, it will only be used as a benchmark.

Machine Learning Solution

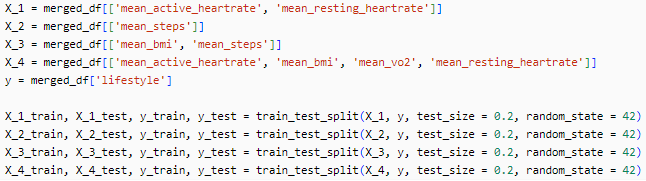
The first step to implement our ML solution is to define the features, labels and algorithm. For the first two, we can go back to the information section and analyze the columns that we have available; in this way, we can identify that the label will be the “lifestyle” column, and the columns that are useful for the features are: “mean\_bmi”, “mean\_active\_heartrate”, “mean\_resting\_heartrate”, “mean\_vo2” and “mean\_steps”, which are part of the “ht\_agg” table. Now that we have defined this, we can continue with our solution design by selecting the algorithm; in this scenario, our labels are classes, so we know that the model has to be a classification algorithm. Another thing to notice is that the features are numerical but they are not in the same scale, so we can either scale them or use an algorithm which is not distance based. Taking into account all the previous information, the path selected is to create a Decision Tree Classifier, which will allow us to use the features in their respective scales with great accuracy.

To start with the algorithm implementation, we are going to join our two tables (“ht\_users” and “ht\_agg”), so it becomes easier to visualize all the columns that we have available for the analysis. To achieve this, we use the pandas function “merge”, with both of the table’s data frames as parameters, which will create a join by the “device\_id” column since it is found in both DFs.



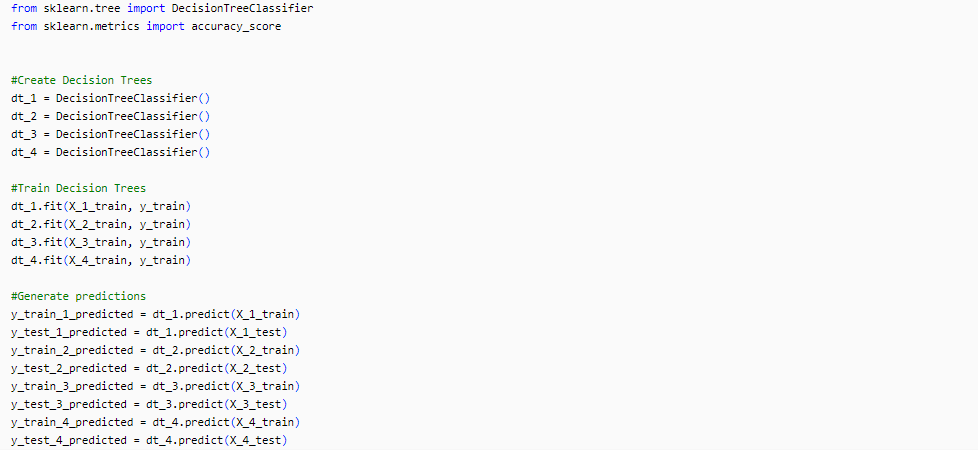
*Fig. 3 Merged DataFrame*

The following action consists in creating the features matrix and the labels vector, that will be used to fit the Decision Tree Classifier. For the purpose of the project, we are going to create four sets of features (X\_1, X\_2, X\_3 and X\_4), each one with different set of columns to fit distinct Decision Trees and determine which one aligns better to our objectives. On the other hand, the labels vector (y), will be created by only taking into the “lifestyle” column, and it will be the same for all of our classifiers. After defining the matrixes and vector, we are going to split our data into training and test sets, so we can use the first one to fit the model and the second ones to evaluate them.



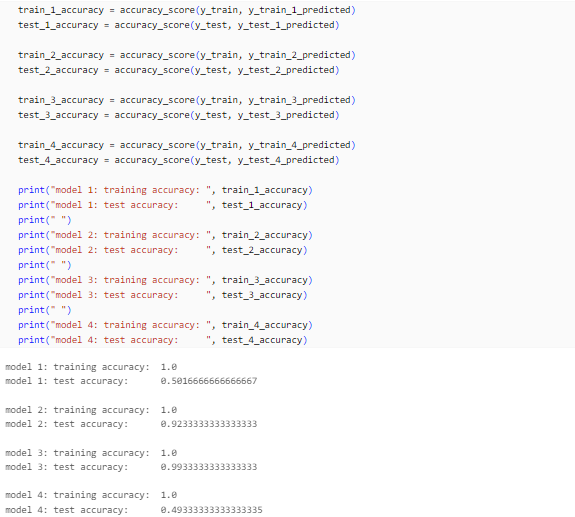
*Fig. 4 Training and Tests Splits*

Once our training and test splits have been computed, the next step lies in the creation of the Decision Tree Classifiers, fitting them with the training data and creating the predictions, with both, the training and test data to evaluate accuracy and concepts such as underfitting or overfitting.



*Fig. 5 Decision Tree Classifiers Creation*

In the final step, of the Machine Learning algorithm implementation, we evaluate all of the models, by calculating their accuracy in training and test data, by using the scikit-learn function, “accuracy\_score”.

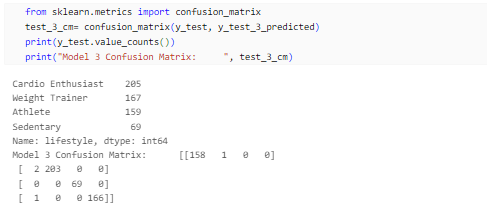


*Fig. 6 Accuracy of the Models*

**Analyze and Interpret Results**

As we can appreciate in the results of the accuracy computation (Fig. 6), models 1 and 3 present an extreme overfitting, since their accuracy metric in training model is excellent, but in test model drops drastically. Regarding model 2, the results are moreover the same as the previous ones, despite its accuracy doesn’t drop as extreme as with the other models, it is clear that is also presents a degree of overfitting. Finally, although model 4 also has a 1.0 accuracy and could be consider to be overfitted, this one stands out from the others by having a stronger relationship between features and labels, so the performance is also great in the test set (0.993).

In order to determine if model 3 meets with the business objective of “creating a Machine Learning model able to predict if a user has a “sedentary” lifestyle based on the information given on the “sign up” process with at least 0.95 accuracy” we need to check specifically the accuracy on our predictions for the “sedentary” lifestyle, since the 0.993 value, represents the accuracy on all of the lifestyles. Therefore, we are going to display the confusion matrix of the model with the test set predictions.



*Fig. 7 Confusion Matrix*

By displaying the count of every lifestyle and looking at the confusion matrix, we can realize that the model predicted 69 sedentary values out of 69 correctly, which gives us a 1.0 accuracy and rejects the Null Hypothesis.

**Communicate and Deliver Results**

Findings:

* ‘mean\_active\_heartrate” and “mean\_resting\_heartrate” are two features which doesn’t seem to have a deep relationship with the perception that people has of their “lifestyle”, since these two columns were used in model 1 and 4, which were the ones with the worst predictions.
* ‘mean\_steps” by itself is a feature with huge impact in the “lifestyle” column, since model 2, was only trained with it and it got good results.
* ‘mean\_bmi” is another useful feature to predict the “lifestyle” column, since it was combined with “mean\_steps” in model 3, and obtained the best results, meeting our business objective.

Potential Issues:

* Decision Trees models tend to overfit or underfit training data.
* The overall quantity of data available data is small (only 3000 rows), so we can’t take for granted the obtained results.
* By seeing at the value counts of the model 3 labels in test set, we could think that there exists a class imbalance, since the proportion of sedentary people is low compared to the others.
* In the current project, we didn’t consider the case of missing values in rows.

Potential Improvements:

* All of the 4 models, got a perfect score in the training accuracy, which far from being good, tells us that out Decision Trees tend to overfit. In order to reduce this, we could use another ML classification algorithm or tune some of the hyperparameters and use ensemble modeling.
* Confirm if there is data imbalance, and if it is the case, use synthetic data to decrease it and increase the overall data quantity, as much as it can be.
* Obtain a bigger dataset (more observations).
* Use some kind of feature engineering method to deal with missing values.