FIIT DATA SCIENCE REMOTE EXERCISE REPORT

Author: Laureano Nisenbaum (laureano.nisenbaum@gmail.com)

1. Summary

This report includes my findings on FIIT Data Science challenge. My work included an Exploratory Data Analysis (EDA) followed by a user segmentation using Unsupervised Learning KMeans Clustering model. This report is to be complemented with the attached *'FIIT_challenge.ipynb'* notebook which contains all the code used for my analysis.

2. Data Cleaning

My project started by inspecting FIIT club leaderboard data from *'workouts__fiit_club.csv'* file. I identified 33 users with null `*app_tracker*` field who did not receive a leaderboard position so I decided to clean those instances because I was interested in the subset of users scoring points during the competition. This resulted in a DataFrame with 248 data rows.

Additionally, there were some users at the bottom of the users table with null scores. Most of them `CANCELLED` their session before starting the workout so I assigned a score 0 of FIIT points to all of them. Some of them did `COMPLETE` the session but maybe they were not wearing their registered device during the workout and did not score any point.

There were also two users with non-null scores in last position, I will assume that the leaderboard is to be trusted and assigned them null scores for my analysis (as every other competitor at the bottom of the leaderboard).

Lastly, I noticed that two users who started the competition and received valid leaderboard positions were not assigned any points. I decided to trust the leaderboard order once again and used *`pandas.fillna()`* function to input their scores with a *'bfill'* method and added a value of 0.0001 points to their scores. These users both had `STARTED` on their session *'state'* feature. Maybe this tagging error was somewhat related to the fact that they were not assigned a valid score after finishing the competition.

I will make a discussion on users `CANCELLING` (17) or `QUITING` (5) their sessions below while plotting the users' score distribution.

Note: even though it was not a strict requirement for the exercise I performed an Exploratory Data Analysis of *'workouts_history__fiit_club.csv'* to get a better understanding of the variables I was going to be working with. A summary of this analysis can be found at the bottom of this report under the Appendix section. Please refer to the notebook for related code.

3. Exploratory Data Analysis

This EDA is focused on *'workouts__fiit_club.csv'* file. Firstly, I plotted a histogram of scores and identified a skewed distribution with a long tail to the left for smaller values of scored FIIT points.

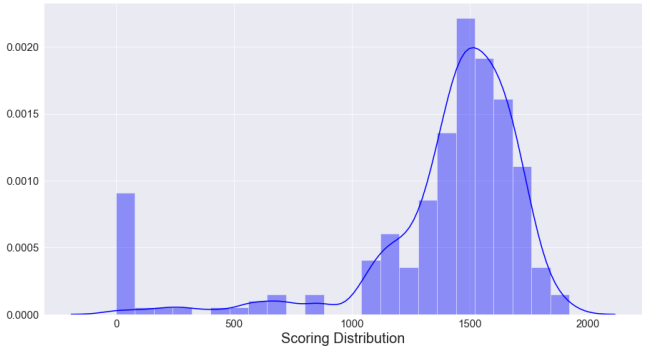

Figure 1. Skewed histogram of FIIT score distribution

By analyzing a subset of users scoring less than 500 points, I realized that out of 22 instances:

- 15 people cancelled their club session (out of a total of 17 CANCELLED sessions)
- 1 person quitted (out of 6 QUIT sessions)
- 6 COMPLETED it with low performance

I then decided to cut scores under 500 as to approximate meaningful `COMPLETED` sessions and to also lower the Skewness in my FIIT points scoring distribution. I also noticed that some small amount of users who `CANCELLED` or `QUITTED` their session did score meaningful amount of points.

Note that I also tried cut-points scores of 750 and 1000 but that only increased the amount of `COMPLETED` sessions discarded (up to 13). Hence, I assumed that those people were just low performance users to avoid losing more data rows.

Finally, I plotted the results on a 25 bins Histogram in the range 0 to 2000 (based on max-min scores observed during the session of interest).

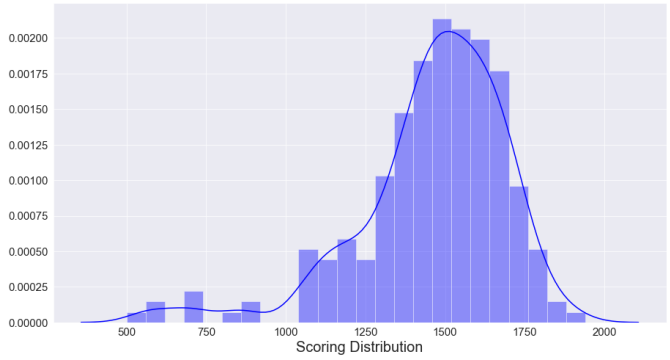

Figure 2. Modified histogram of FIIT score distribution including only scores greater than 500.

4. Modeling: KMeans Clustering

Considering the problem at hand and the available data I decided to proceed with an Unsupervised Learning algorithm. In particular, I worked with KMeans Clustering which proves to be a good algorithm to segment users and find insights on unlabeled datasets.

Initially, I expanded my original `fiit_club` DataFrame by adding numeric features that were suitable for cluster analysis. These numeric features were obtained from *'workouts_history__fiit_club.csv'* and from *'fiit_club_bpm_series.json'* datasets.

Some options for building my model I considered were:

- `*bpm*` data (from `.json` file)
- `*user_age*`
- `*min_recorded_hr*`
- `*max_recorded_hr*`
- `*recency*`: a measure of how recently each user had an interaction with FIIT app
- `*frequency*`: a measure of how many interactions the customer has done in the last `T` period of time

While considering my options I had to keep in mind that the algorithm of my choice works with numeric variables where a certain order between variables can be stablished (i.e. ordinal variables). I discuss some additional requirements over my variables during my implementation below. In the end, I chose to work with: `*mean_bpm*` , `*user_age*` , `*score*` to cluster my data. In order to obtain `*mean_bpm*` data I used heart rate data stored in the *'.json'* file provided. I could have used `*median*`, `*min*` or `*max*` bpm as well. The histograms for `*mean_bpm*` and `*user_age*` are shown below.

I also tried to keep my model simple at first and then proceeded to discuss some possible enhancements. Adding more variables would just have made my model more difficult to interpret and visualize. Once we get a simple model running we can always iterate and add or subtract variables.

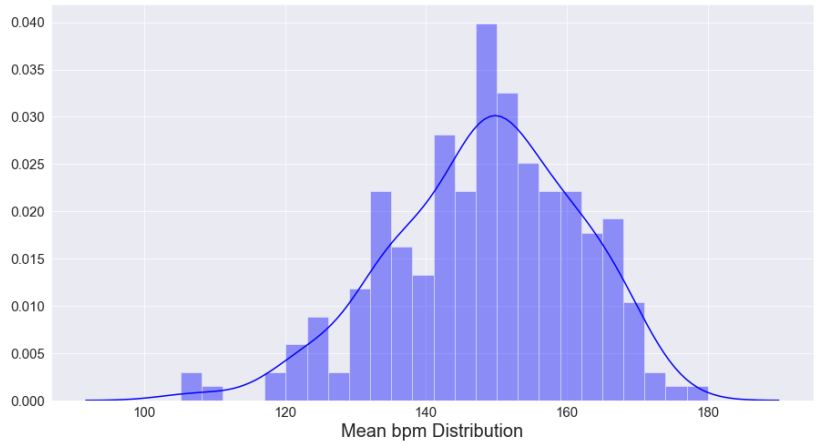

Figure 3. Histogram of FIIT Club instance representing the users' `mean_bpm` distribution

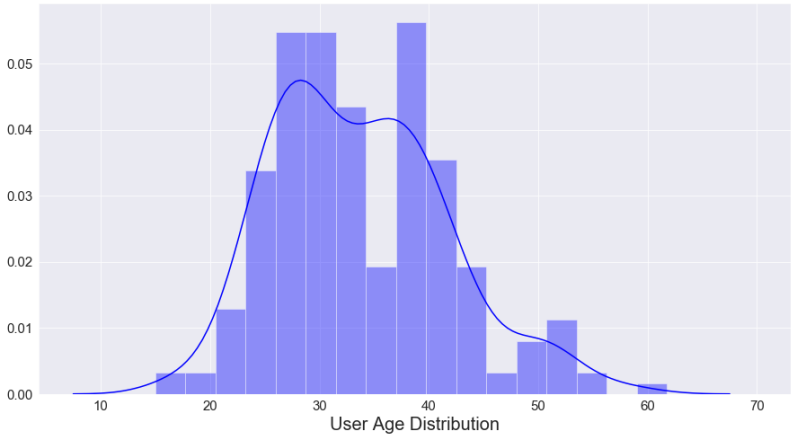

Figure 4. Histogram of FIIT Club instance representing the users' `mean_age` distribution

KMeans Clustering Requirements

In order to build my clustering model I followed several steps:

i) The first step in the segmentation process was to pre-process the data. I applied logarithmic transformations to reduce any skewness present in the distributions of the variables I chose to work with

ii) Secondly, I normalized the data to prepare it for clustering with *scikit-learn's* `*StandardScaler()*` module

iii) Next, I checked that my variables showed distributions that were roughly normal. The

standardized distributions can be seen in Figure 5 below.

Note: I considered my transformed and scaled features "roughly normal" for the purpose of my model. Having said this, I noticed two things:

- `total_points` distribution still showed some skewness, this could be further reduced by removing more low performance users under the assumption that this people were not actually completing their training sessions.

- `user_age` seems like a bimodal distribution. For instance, there might be two groups of users with different mean ages using FIIT app more frequently. For example looking at the unscaled `user_age` histogram: 1) A group of people in their late 20s / early 30s and; 2) Other group on their early 40s.

To continue my analysis I assumed my distributions were acceptable but segmenting users by Age might be an interesting avenue to keep exploring the data in the future.

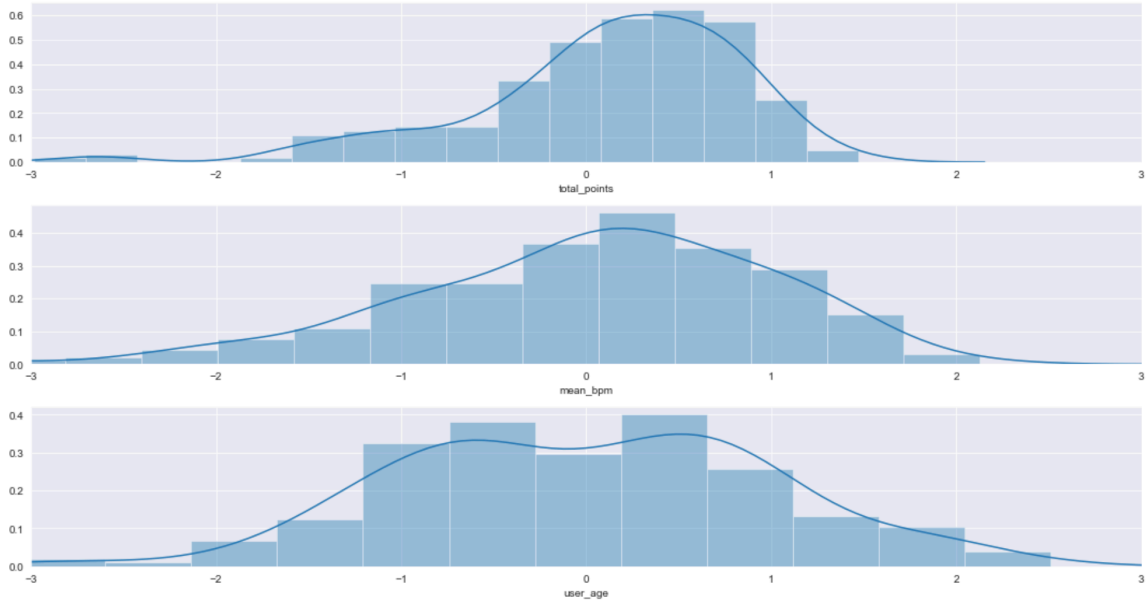

Figure 5. Histograms of preprocessed and standardized variables for the KMeans Clustering model.

iv) Following that, I plotted the sum of squared errors for each value of `K` (number of clusters) I tried while building my KMeans clustering models.

v) In order to identify an optimal number of Clusters `K`, I used the "Elbow Criterion". This criterion guided me towards the recommended number of clusters to use. From the plot below one can identify an elbow somewhere around 4 clusters. Hence, I decided to build my KMeans model with 4 Clusters.

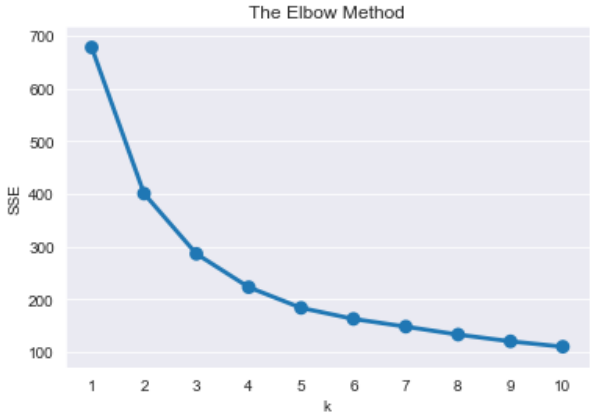

Figure 6. Plot of the sum of squared errors for several values of `K`. The optimal value was determined to be 4.

5. Results

The results from my KMeans Clustering model are shown below in Figures 7 and 8:

	total_points	mean_bpm	user_age	
	mean	mean	mean	count
label				
0	714.4	121.8	36.7	9
1	1571.5	156.5	26.8	94
2	1566.9	152.3	39.5	66
3	1273.3	134.0	38.0	57

Figure 7. KMeans Clustering summary metrics

In the clusters plotted below there seems to be two groups of users scoring higher amounts of points. These users correspond to the Red (1) and Green (2) clusters and seem to complete workouts at higher rates of `*mean_bpm*`. In particular, Red (1) cluster has the most extreme `*mean_bpm*` values and the higher mean value at 156.5 bpms.

Additionally, when considering age groups it seems that the Red (1) cluster correspond to younger users (< 33 years) than the ones belonging to the Green (2) cluster (> 33 years). The Red (1) cluster exhibit the lowest mean `*user_age*` value at 26.8 years. The Orange (3) cluster seems to cover a broad range of ages. Lastly, the Blue (0) cluster comprise a low number of users (only 9), so it is more difficult to make conclusions about its age range. However, it seems to cover a broad range of ages as well.

Moreover, The Blue (0) Cluster is the one associated with users with the lowest performance and it also includes users with the lowest `*mean_bpm*` value at 121.8 bpms.

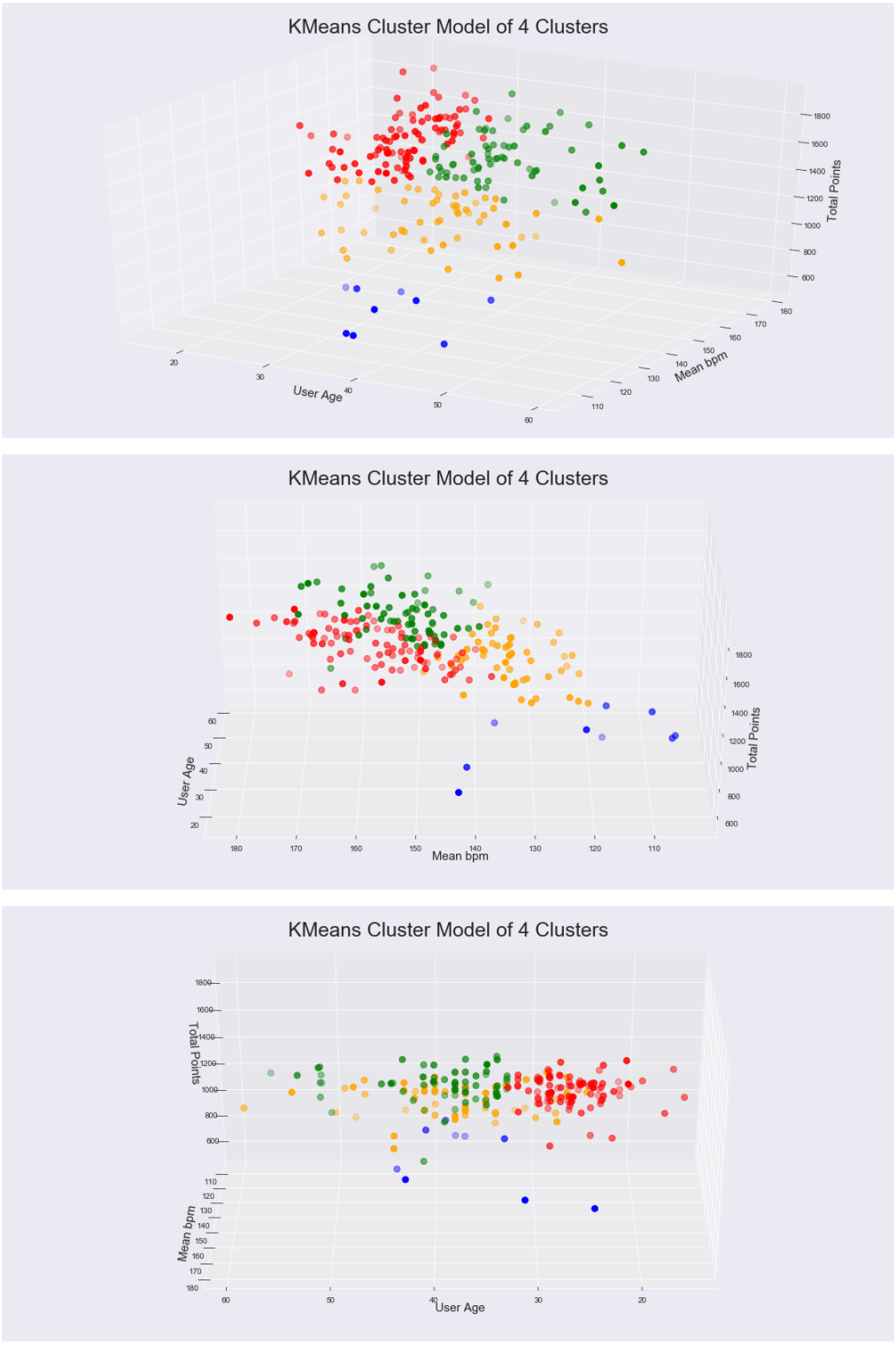

Figure 8. Three different views of FIIT Club instance KMeans model with 4 Clusters

Lastly, the Orange (3) cluster seems to be performing in between the Blue (0) cluster and the Red (1) and Green (2) clusters in terms of `*mean_bpm*` performance.

Ideas for future work

My initial KMeans model could be expanded by using more variables such as:

- `*recency*`: a measure of how recently each user had an interaction with FIIT app. This variable can be constructed by analyzing the history of every user workout, segmenting them in a small number of groups (ie. Very High, High, Medium, Low, Very Low recency) and adding the resulting variable to our Clustering model. This variable could help us answer the following question: *Are recurrent users more likely to rank higher or is the system somehow penalizing them?*

- `*frecuency*`: a measure of how many interactions the customer has done in the last *T* period of time. This variable can be engineered by counting the number of workouts for each individual user from FIIT historical data. This variable might help us analyze whether users with more activity tend to score higher or not,

We could also try segmenting users by categorical variables such as `*gender*`:

- MALE and FEMALES tend to have different mean heart rates and also might differ in their BPM working zones. I would like to see how the model performs by splitting data on `gender`, but at this point I decided to pool all data together specially because I have considerably more data coming from women (90% vs 10%). See EDA in Appendix.

Similarly we could segment data by `*age*`:

- Again this would reduce de number of data points, but given more data we could try building more targeted models for specific age groups.

Complementary Statistical Analysis

As an additional analysis, I decided to split FIIT Club data in five Quintiles (according to the variable `*leaderboard_position*`). The results are included in the table below:

	mean_age	mean_bpm	mean_min_hr	mean_max_hr	females	males	fem_to_male_prop	cluster_0	cluster_1	cluster_2	cluster_3
5	32.2	161.8	89.8	187.4	41.0	4.0	10.2	0.0	27.0	18.0	0.0
4	31.2	155.0	83.3	183.1	41.0	4.0	10.2	0.0	28.0	17.0	0.0
3	33.3	149.6	83.7	178.8	40.0	5.0	8.0	0.0	22.0	22.0	1.0
2	33.8	142.4	78.1	174.2	38.0	7.0	5.4	0.0	15.0	7.0	23.0
1	37.8	132.8	77.4	164.8	38.0	8.0	4.8	9.0	2.0	2.0	33.0

Figure 9. FIIT Club scoring analysis based on a Quintiles split approach. Several summary statistics are shown for the five different quintiles.

As a general tendency the Quintile analysis showed that older people working out at lower `*mean_bpm*` rates scored less amount of points. Additionally, there seems to be more males than females in the lower quintiles (bottom of the leaderboard). However, there are considerably more data points for workouts completed by females (around 9:1 relationship) which makes it a bit difficult to know if this tendency is meaningful or if it is only due to lack of data points coming from men workouts. Having said this, we should keep in mind the two statements that follow:

"The average adult male heart rate is between 70 and 72 beats per minute, while the average for adult women is between 78 and 82 beats. This difference is largely accounted for by the size of the heart, which is typically smaller in females than males"

(Source: Hyperlink 1)

"Older hearts simply can't beat as fast as younger hearts. So the older person who's doing 120 beats per minute is probably working harder -- at a higher percentage of maximum heart rate -- than the younger person who is at 150 beats per minute."

(Source: Hyperlink 2)

Hence, if greater scores are correlated with higher `*mean_bpm*` rates, males might have a harder time scoring FIIT points specially if they are older. Moreover, if one person starts improving his/her physical state thus lowering their `*mean_bpms*` during workout, then this person might start finding more difficult to score FIIT points.

Additionally, when analyzing the labels assigned by the KMeans clustering model, I saw that:

i) Users belonging to Clusters 1 (Red) and 2 (Green) mostly belong to the top 3 quintiles (Q3, Q4 and Q5).
ii) Users belonging to Cluster 0 (Blue) belong to the bottom quintile (Q1)
iii) Lastly, users belonging to Cluster 3 (Orange) are split over the bottom two quintiles (Q1 and Q2).

6. Tentative scoring improvements

If we observe people with lower `*mean_bpm*` rates are scoring less FIIT points, we might try for example:

A) Segmenting users in different age groups
B) Segmenting users by their biological sex

After segmenting users using those two categorical features we could compute their mean (or median) FIIT points scores by pooling data from as many workouts as possible. With those mean scores we could obtain relationships between different groups and thus be able compute correction coefficients, for instance:

Group 1 Score (G1) : mean scores obtained by Female Users <= 33 years old

Group 2 Score (G2): mean scores obtained by Female Users > 33 years old

$$CorrectionFactor = \frac{G1}{G2}$$

Lastly, after each FIIT Club session, we could decide to assign a bonus amount of points to the older group age, for example:

Score (S_2) : score obtained by a single female user from > 33 years old group (G2)

$$CorrectedScore = CorrectionFactor * S_2$$

This correction would increase workout scores of those groups that might be experiencing disadvantages while earning FIIT points during a session. Similarly, if enough data is available, a second models could be built to account for biological differences based on `*gender*` feature for male users.

7. Conclusions

I was able to successfully deploy a KMeans clustering Unsupervised Learning model to get some useful insights on a FIIT Club instance of interest. Additionally, I complemented my work with a statistical analysis by splitting data in five quintiles. With my findings is mind, I was able to suggest a tentative scoring approach to improve FIIT scoring system.

8. Appendix

Summary of insights from *'workouts_history__fiit_club.csv'* EDA (check Jupyter notebook for complete code):

i) The percentage of `COMPLETED` workouts in the analyzed sample was 88%

ii) Five most popular lessons observed were: *`Full Body Stretch #2`, `Mobility Flow #12`,`Full Body Stretch #3`,`Upper Body Stretch #2`,`Full Body Stretch #4`*

iii) From all observed workouts: 46.6% belonged to `CARDIO` studio, 29.6% to `REBALANCE` studio and 23.8% to `STRENGTH` studio

iv) Most FIIT workouts observed were performed on `iOS` (89%) and a relatively small amount on `Android` (10%). Other platforms constitute the remaining 1%.

v) The observed type of classes distribution was: 83% `INDIVIDUAL` Classes, 15% `FIIT_CLUB` classes and 2% of `PRIVATE_GROUP `

vi) From all FIIT (unique) users considered: 52% reported `LOSE_WEIGHT` goals, 23% `GENERAL` goals , 22% `BUILD MUSCLE` goals , 2% `FLEXIBILITY` goals and 1% `POSTNATAL` goals.

vii) From all FIIT (unique) users considered : 64% reported `MEDIUM` activity level, 25% `HIGH` activity level , 22% `LOW` activity level and 1% `VERY_LOW` activity level.

viii) The distribution of activity duration as grouped by *`lesson_id`* observed was: 72.0 % for `ABOUT_25_MINUTES` activities, 22.0 % for `ABOUT_40_MINUTES` activities and 6% for `ABOUT_10_MINUTES` activities.