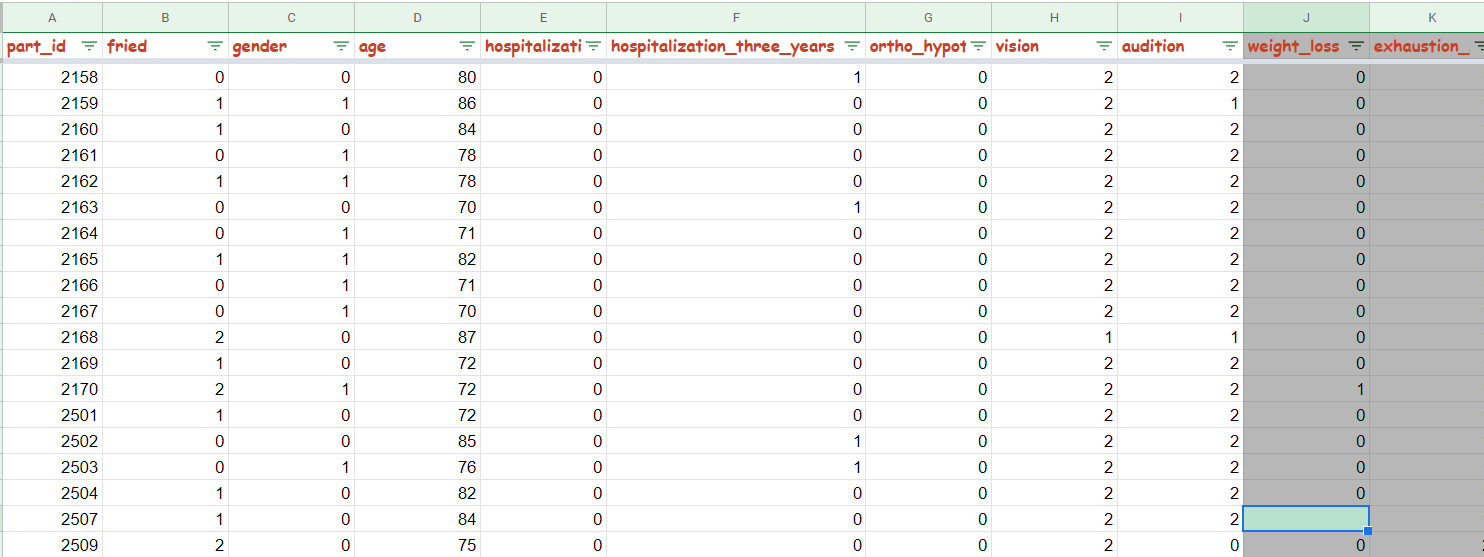
Project in Βig Data Analytics

By Siope Alexis and Impro Vanessa (ΣΜΗΝ)

**PREPROCESSING**

* First of all we use the *split text to columns , separate with “;”*.
* CTRL + H :Find and replace nominal values with numerical values.
* CTRL + H :Find and replace “*999*” and “*test non applicable/adequate*”.
* Using *ColumnStats* we detect erroneous values (Lean body mass = -321, bmi score =921) and remove them.
* Remove entries if we have a null on these columns: weight\_loss, exhaustion\_score, gait\_speed\_slower, grip\_strength\_abnormal and low\_physical\_activity.



* Create a column with =countblank(2:2) to count nulls in each line and then we create a filter to view values which have more than 5 null values.
* We delete the entries shown.
* This process causes approximately a 5% loss on our data.
* Select the data type (integer , float).
* Find the average in every column (e.g.=AVERAGE(B:B))
* Place the data respectively.

After preprocessing the csv file, we import the data. We name our dataset as df .

Εικόνα που περιέχει κείμενο

Περιγραφή που δημιουργήθηκε αυτόματα

Τhe format of our database:

Εικόνα που περιέχει κείμενο, στιγμιότυπο οθόνης, εσωτερικό

Περιγραφή που δημιουργήθηκε αυτόματα

Εικόνα που περιέχει κείμενο

Περιγραφή που δημιουργήθηκε αυτόματα

* Only numeric values
* No null values

**Classification**

* drop columns that don't contribute in classification.

*df.drop(['leisure\_club','social\_visits','social\_calls','social\_phone','social\_skype','social\_text','living\_alone','part\_id'], axis=1,inplace=True)*

* divide the set into dependent and independent variables. The dependent is fried (y) and the rest are independent (x).

*x=df.values*

*y=df['fried'].values*

*#delete the fried column from x*

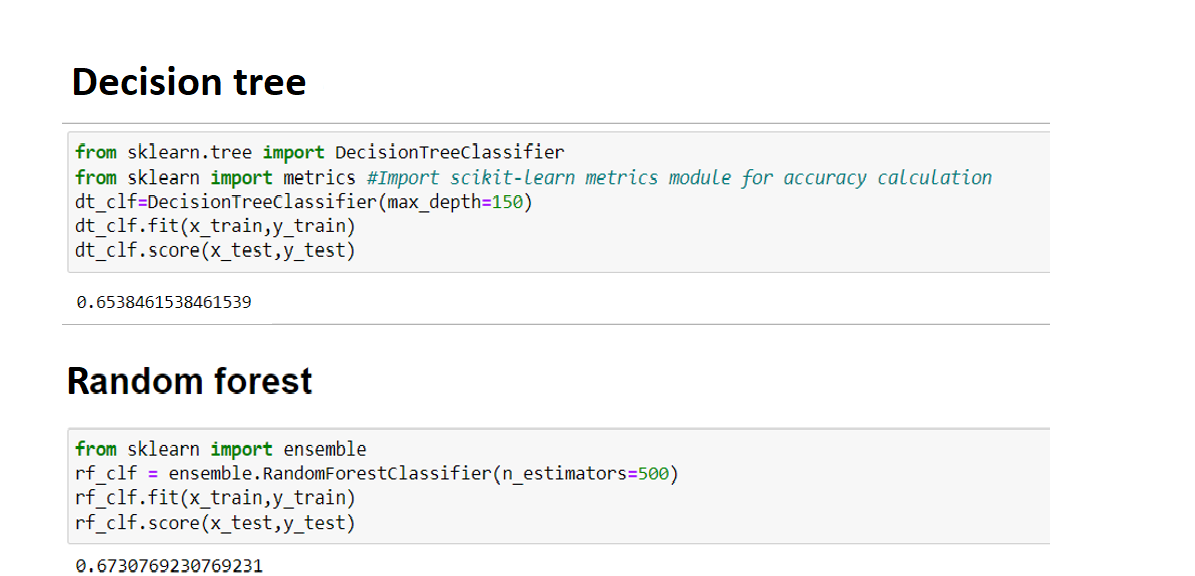
*x=np.delete(x,0,axis=1)*

* split the dataset in train and test.

*from sklearn.model\_selection import train\_test\_split*

*x\_train, x\_test,y\_train, y\_test= train\_test\_split(x,y,test\_size=0.1,random\_state=0*

**Classification Results**



Εικόνα που περιέχει κείμενο

Περιγραφή που δημιουργήθηκε αυτόματα





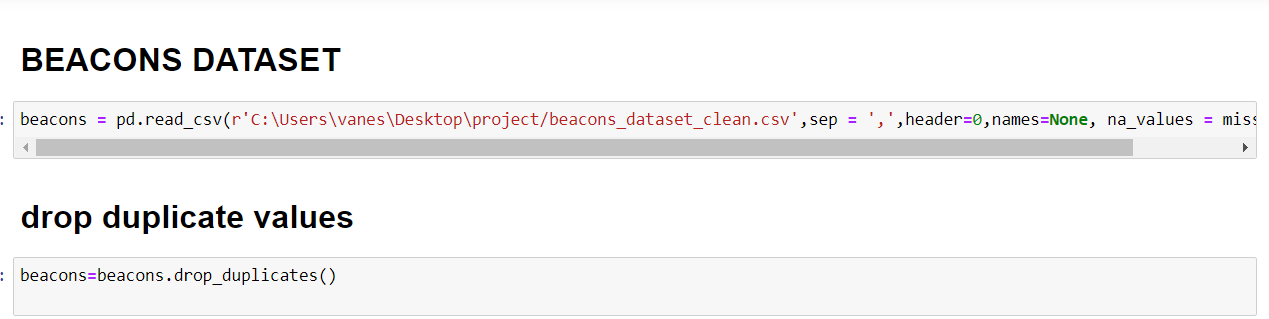
Εικόνα που περιέχει κείμενο

Περιγραφή που δημιουργήθηκε αυτόματα

Support Vector Machine gives the best results. Effective in high dimensional spaces. Still effective in cases where number of dimensions is greater than the number of samples.

**Beacons Dataset Preprocessing**

* With ColumnStats we view the data “*room*” and use Find and replace to correct the room labels.
* Delete irrelevant “*room*” entries (e.g., “One”).
* Using a filter for part\_id with non 4-digit part\_id we remove the entries.



Εικόνα που περιέχει κείμενο

Περιγραφή που δημιουργήθηκε αυτόματα

**Time every person has spent in «Bedroom», «Bathroom», «Livingroom» and «Kitchen»:**

* In order to determine the time each person was in each room on all the days measurements were taken, we need to sort our data by part\_id.
* For each person we successively subtract the times measured in each room on all dates and place them in the corresponding column for that room.
* We note each time how much is the total time measured for each person so that I finally get percentages for each room.

*Code for the process described before:*

*Εικόνα που περιέχει κείμενο

Περιγραφή που δημιουργήθηκε αυτόματα*

*Εικόνα που περιέχει κείμενο

Περιγραφή που δημιουργήθηκε αυτόματα*

For instance, here we have the times for part\_id 1001 measured on 30/04/2018

*Εικόνα που περιέχει κείμενο

Περιγραφή που δημιουργήθηκε αυτόματα*

Εικόνα που περιέχει κείμενο

Περιγραφή που δημιουργήθηκε αυτόματα

Our data is stored in a dictionary.

**A new dataframe containing the percentage of time for each id in each room.**

Εικόνα που περιέχει πίνακας

Περιγραφή που δημιουργήθηκε αυτόματα

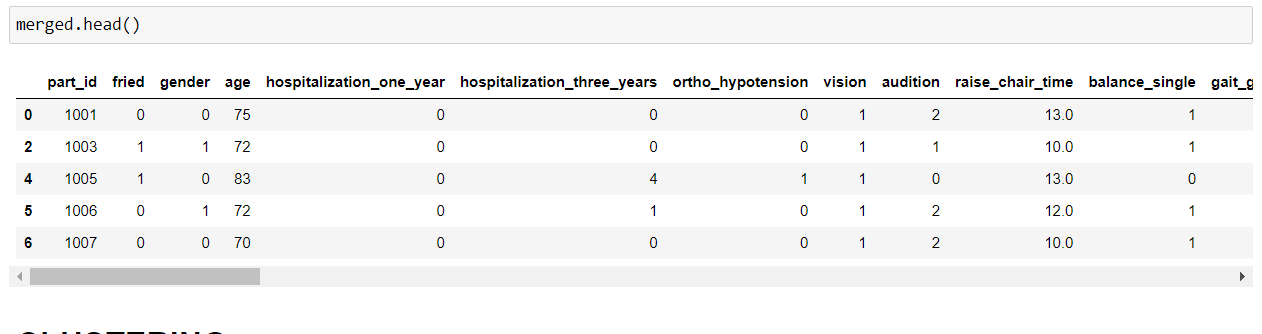
**Merging the two datasets by the same part\_id**

Εικόνα που περιέχει κείμενο

Περιγραφή που δημιουργήθηκε αυτόματα

* Insert the clinical dataset
* Add columns for each room where the part\_id of clinical dataset is the same as the one on beacons dataset.
* The new merged dataset without null values

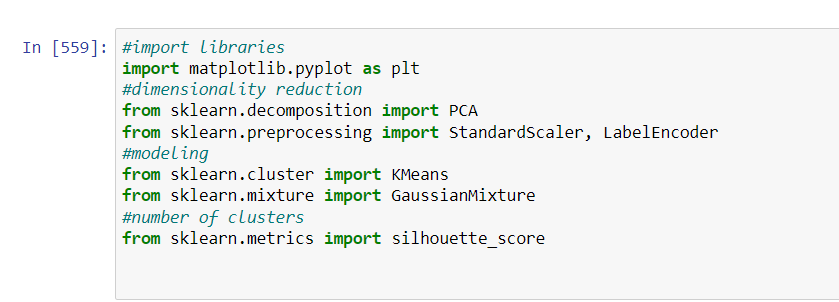
**Merging Result**

Εικόνα που περιέχει πίνακας

Περιγραφή που δημιουργήθηκε αυτόματα

**CLUSTERING**

* Import libraries.

****

* Create a new dataset named x from merged without columns 'part\_id','fried','leisure\_club','social\_visits','social\_calls','social\_phone','social\_skype','social\_text','living\_alone’ .

Εικόνα που περιέχει πίνακας

Περιγραφή που δημιουργήθηκε αυτόματα

**Dimensionality Reduction**

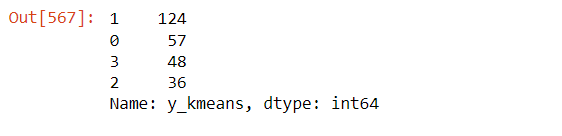
* Principal Component Analysis (Linear dimensionality reduction using Singular Value Decomposition of the data to project it to a lower dimensional space**)**
* StandardScaler (Standardizes a feature by subtracting the mean and then scaling to unit variance)

Εικόνα που περιέχει κείμενο

Περιγραφή που δημιουργήθηκε αυτόματα

**CLUSTERING MODELS**

Εικόνα που περιέχει κείμενο

Περιγραφή που δημιουργήθηκε αυτόματα

Εικόνα που περιέχει κείμενο

Περιγραφή που δημιουργήθηκε αυτόματα

Εικόνα που περιέχει κείμενο

Περιγραφή που δημιουργήθηκε αυτόματα

Εικόνα που περιέχει κείμενο

Περιγραφή που δημιουργήθηκε αυτόματαΕικόνα που περιέχει κείμενο

Περιγραφή που δημιουργήθηκε αυτόματα

Columns added:

We saved the predictions as columns

Εικόνα που περιέχει πίνακας

Περιγραφή που δημιουργήθηκε αυτόματα

**Evaluate Clustering**

Εικόνα που περιέχει κείμενο

Περιγραφή που δημιουργήθηκε αυτόματα

**How did we decide the number of clusters?**

Decide number of clusters with silhouette score.

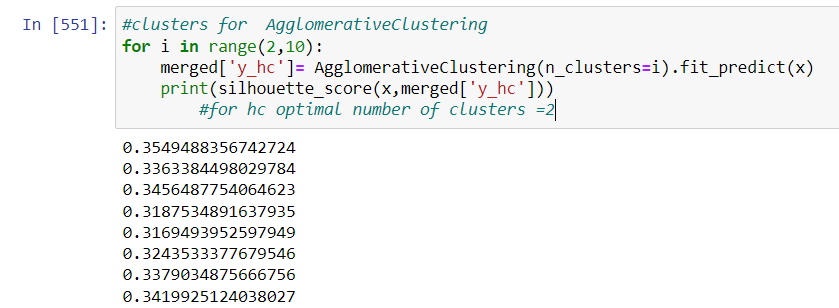
Εικόνα που περιέχει κείμενο

Περιγραφή που δημιουργήθηκε αυτόματα

Εικόνα που περιέχει κείμενο

Περιγραφή που δημιουργήθηκε αυτόματα

Optimal number of clusters for kmeans🡪4



Optimal number of clusters for AgglomerativeClustering🡪2

Εικόνα που περιέχει κείμενο

Περιγραφή που δημιουργήθηκε αυτόματα

Optimal number of clusters for Gaussian Mixture🡪3

\*\*\*(whole code included in a ipynb file to open with jupyter notebook, also the preprocessed csv files.)