UCLM - Escuela Superior de Informática - Ciudad Real

Machine Learning
Lab Book

Made by:
Benjamín Cádiz de Gracia
Iván García Herrera
Dídimo Javier Negro Castellanos

Date: October, 2018.

Índice

1. Monday, 15 October 2018

1.1. Steps

- First, we have created a *cookiecutter* based directory structure.
- Then, we have revised the dataframe.
- We have deleted non numerical columns, it means, the columns that contain string values.
- We have deleted the columns which had all `NaNs`. If we don't have care enough and we try to delete rows with `NaNs` directly, we will delete the complete dataset, because there are some columns with all null values.
- We have removed duplicate rows, that is, two or more rows with exactly the same values. We have seen that in this dataset there are not duplicate rows.
- We have deleted the rows containing not a number (`NaNs`) values.
- We have tried to identify some outliers in some columns but we haven't have success.

2. Thursday, 18 October 2018

2.1. Steps

- We get the columns that ends with ”MEAN“ because that represents the median values. In that way the dataset is more representative. After that we work on the visualization.
- We do an attemt and the PCA fails because we haven't the version column and there are rows where the version number is not a float, it's a string like ”x.x.x“, so we eliminate the version column. The covariance is so low, that indicate data are not representative.
- Next, we start clustering with k-means. We analice different groups in order to interpret the plots. We part the job: Ivan analice the magnetic field, Benjamin the gyroscope and Dídimo the accelerometer.
- If we do k-means with the complete dataset, the computer are out of memory, that is the reason why we only considerate the values of only 5 days.
- We do k-means in a range of 2 to 20 clusters to know what the ideal number of that. In order to get it we use the silhouette method and we analice which is the best coefficient. When we already know the ideal number of cluster we do the plot showing the different groups.

3. Friday, 19 October 2018

3.1. Steps

- We have cleaned the source code because we have seen that the implemented functionality was correct, so we have decided to organice the code in classes to avoid the repetition of code. In this way, the structure is clearer and it's easier to understand the code.

4. Tuesday, 23 October 2018

4.1. Steps

- We have studied the operation of clustering algorithm to identify outliers, which is DBSCAN
- We have based in the moodle example and we have tried to do `DBSCAN` with a symetric matrix but we didn't understand the results.
- Then we decided to use a conectivity matrix as input for a hierarchical clustering algorithm (`AgglomerativeClustering`). This conectivity matrix was obtained applying the *”k nearest neighbors“* algorithm. This algorithm returns a matrix that indicates what k neighbors is connected an element from the matrix `AgglomerativeClustering` returns a clusters hierarchy.
- We have applied the `DBSCAN` algorithm to know what points from the dataset are core points, what points are border points and which ones are noise. The noise points will be considered outliers.
- We have plotted the `DBSCAN` graphics to see the outliers.

5. Monday, 29 October 2018

5.1. Steps

- We have analysed and interpreted the K-means groups and the outliers.
- To achieve the previous point, we have grouped data by ”*labels*“ column and the columns with `X, Y, Z` contains the average of all data of distinct groups.
  To know what rows from the original dataset are outliers, we have added ”*labels*“ column to the dataset. This column contains a -1 value if the row is an outlier. We have filtered the rows which had a -1 value in the ”*labels*“ column and we have obtained the outliers.
- He have plotted both interpreted results, the k-means groups and the outliers in a 3D graphic, using the original dataset. In this graphic (see Figure **??**), we can see where are the outlier points with regard to the other points of the dataset.

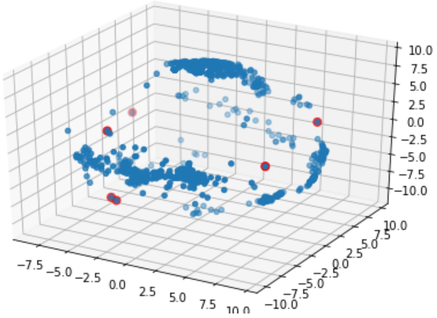

Figura 1: The outlier points are shown in red color. The blue points are all the other dataset points