Software development : Challenge

Alexandre Nicolas

01/04/2021

Prediction part

The first goal of this challenge is to predict the number of bikes passing at Albert 1^{st} on 2021, April 2^{nd} between 0:01 AM and 9:00 AM. To make it, we will work with the number of bikes per day when datas are collected for the last year.

We start by importing our CSV file using the download package with the option "replace = True" which allows us to have the data up to date. Then, we delete the columns that we won't be using (Note, Unnamed, Running Total).

Now, we want to extract only the lines where the time is between 0:01 AM and 9:00 AM. Python should recognize the date and time rather than seeing it as a string. Hence, we use the Pandas' function to_datetime to put the date in international form, and in datetime format. We can now extract the time using the 'hour' attribute of datetime and thus extract the desired rows.

We have the rows in our table at the time we were looking for, but we see that some rows have the same date at two different times. Since our goal is to study the number of bikes per day, we need to add up the number of bikes per date. We do that with the 'groupby' and 'sum' functions.

Thanks to matplotlib and the scatter function, we display the scatter plot of the number of bikes per day according to the date and it shows us an outlier. Indeed, most of the data is between 0 and 300 with some peaks at 600, whereas the number of bikes on 2020, September 18^{th} is equal to 1191. Therefore, we remove this value from our dataframe.

We must also be careful with the health crisis due to Covid. The first lockdown (2020, March 16^{th} - 2020, May 10^{th}) drastically dropped the number of bicycles and is so a special period, which we decide to delete.

We want to do a simple linear regression of our dataset. The problem is that sorting our datas caused us to delete a lot of rows, and the resulting dataframe does not contain all the dates. To fix this issue, we choose to add all the missing dates, and assign them the median of the number of bikes in the corresponding month. For example, if 2020, April 16^{th} doesn't have an associated value, we look at the median of the values for 2020, April and place it in the 2020, April 16^{th} row.

To achieve this, we first import the 'median' function from the 'statistics' package and use it with 'groupby'. Hence, we create a table where each row gives us a month and its associated median.

We will then create a dictionary where the keys will be the numbers of the months (from 1 to 12) and the corresponding median values. Using the 'fillna' function, we replace the 'NaN' by 0 in our table. We finally build the 'replace_median' function which returns us the dataframe with the median of the month for each missing date.

We can finally apply simple linear regression. We obtain the following graphic.

To conclude, we use the function 'polyfit' of Numpy which gives us the equation of the regression line. At the last release of this file (2021, March, 28th), we get the equation (10^{-2} rounded) :

$$y = 0.07x + 111.62$$

x corresponds to the day and y to the number of bikes passed on the x^{th} day between midnight and 9 AM. Since 317 is 2021, March 25^{th}, then 325 is the index for 2021, April 2^{nd}. Hence :

$$y = 0.07 \times 325 + 111.62 \simeq 135$$

Finally, our prediction tells us that there will be 135 bikes on 2021, April 2^{nd} between 0:01 AM and 9:00 AM in Albert 1^{st} in Montpellier.

You can find python files, graphics, GIF and more at the GitHub link : https://github.com/alexxnicolas/Challenge-2021