Thursday, April 1st, 2021

EL KAÏM Laura, master MIND

HMMA238
Challenge prediction

Git link: https://github.com/LauraElKaim/Challenge_prediction

As part of the software development course, I will present you the process of my prediction.

The totem company, located in Montpellier, has set up eco-counters in different places of the city to measure the passage of bicycles.

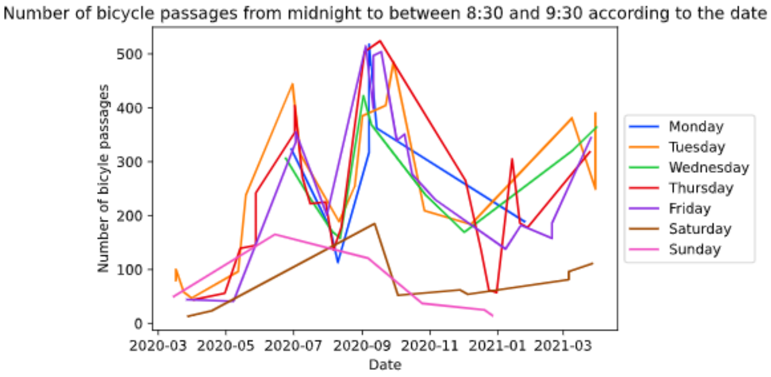
The purpose of this project will be to predict the number of bicycles that will pass by the *Albert 1^{er}* station on Friday April 2^{nd}, 2021, between 00:01 and 09:00 am.

The 1^{st} step consists in importing and reading the data table. The data contains several observations from March 12, 2020 to this day. We can observe 6 columns, and by creating a function we can delete the empty column and the one named “remark”. We also delete the missing data. Now we have the date and the hour of the statement, the total number of bikes since 01/01 (it resets to 0 every year) and the number of bikes on the given date between midnight and the given time (these columns are renamed for better visibility).

Then a function will allow us to format the date. We can create a new column corresponding to the day of the week, 0 for Monday to 6 for Sunday.

In order to predict the number of bicycles that will pass through the *Albert 1^{er}* counter on April 2^{nd} between 00:01 and 9:00 am, we can first perform a simple linear regression.

For this regression, we are only interested in the date, so we can keep only the data between 8:30 and 9:30 am. It would be interesting to see the number of bicycle passages according to the day of the week. Therefore, we notice fewer passages on weekends. Perhaps a higher number of bikes on weekdays would correspond to people cycling to work.



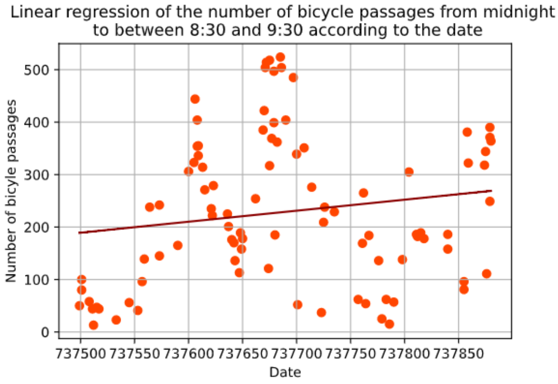
We want to find a line that is close as possible to the set of points, this one represented by the observed data.

We will predict the number of bicycles (variable Y) according to the date (predictor variable X). For this we want to estimate the slope and the intercept.

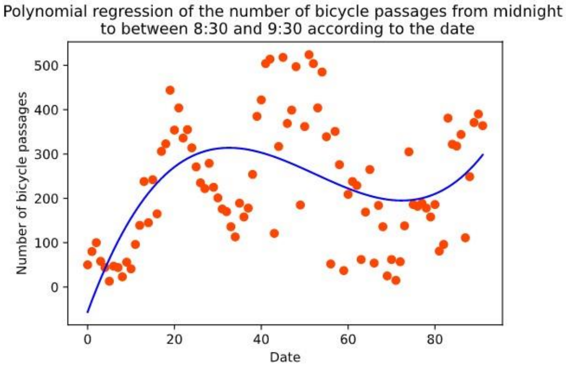
A first function will allow us to return the result *slope * x + intercept.*

Then we can use the **spicy.stats** module which has the **linregress** function. But first we need to convert the date into number of days. Thus, April 2, 2021 would correspond to the number

737882. Using our predict function which retrieves the slope and intercept values we get a prediction of approximately 269 bikes.

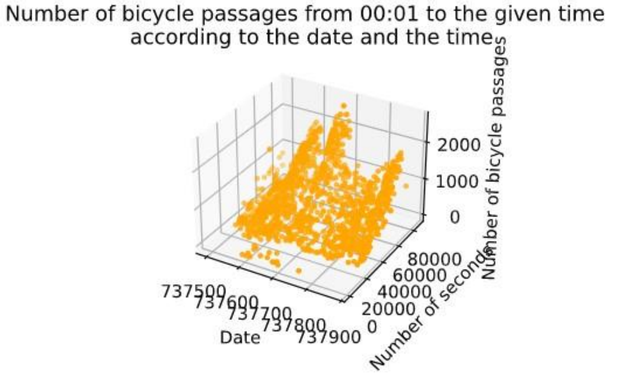


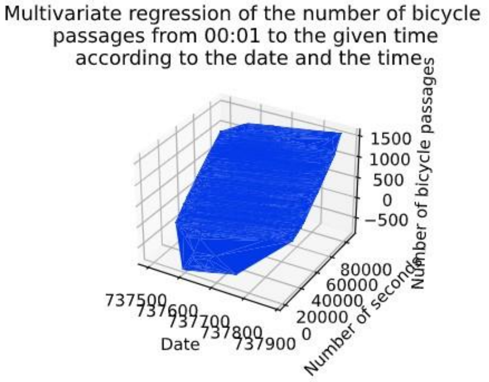
The data seems far from linear, so to refine this prediction we can perform a polynomial regression. For this we use the package **sklearn.** The **LinearRegression** function of the package **sklearn.linear_model** will allow us to make a linear regression (which also gives a prediction of about 269). So, with the **PolynomialFeatures** function of the package **sklearn.preprocessing** we will be able to adjust this regression with a polynomial of degree 3. We have to transform our date column into an array and the value to predict will be the date associated to the number 93. Finally, we get a new prediction of about 325 bicycles and an R^2 of about 30%.



Although having reduced the data to hours from midnight to between 8:30 and 9:30 am, this prediction only considers the dates.

Finally, we can verify this prediction with a multivariate linear regression. This regression will consider the date as well as the time. We will use the **StandardScaler** function from the **sklearn.preprocessing** package and the **statsmodels.api** package. Before that we have to convert the time into number of seconds. This function will return a prediction of type β_0 + β_1 * *Date* + β_2 * *Number seconds*. We want to predict the number of days 737882 and the number of seconds 9*3600. Thus, with the estimated coefficients we obtain a prediction of about 325 and an R^2 of about 58%.





Finally, my prediction will be 325 bicycles.

Note that bicycles do not always pass through the eco-counter but next to it and some factors are not considered here such as the weather and could distort our prediction.

(Source for the linear regression: Régression linéaire en Python par la pratique | Mr. Mint : Apprendre le Machine Learning de A à Z
Source of the polynomial regression: Machine Learning: Polynomial Regression with Python | by Nhan Tran | Towards Data Science
Source for the multivariate regression: Multivariate Regression : Faire des prédictions avec plusieurs variables (mrmint.fr))