ESILV LÉONARD DE VINCI | ÉCOLE D'INGÉNIEURS PARIS-LA DÉFENSE

Python for Data Analysis

ON SEOUL BIKE SHARING DEMAND DATA SET

ROUSSELET Alexandre
RIPAUD Jasmine

Summary

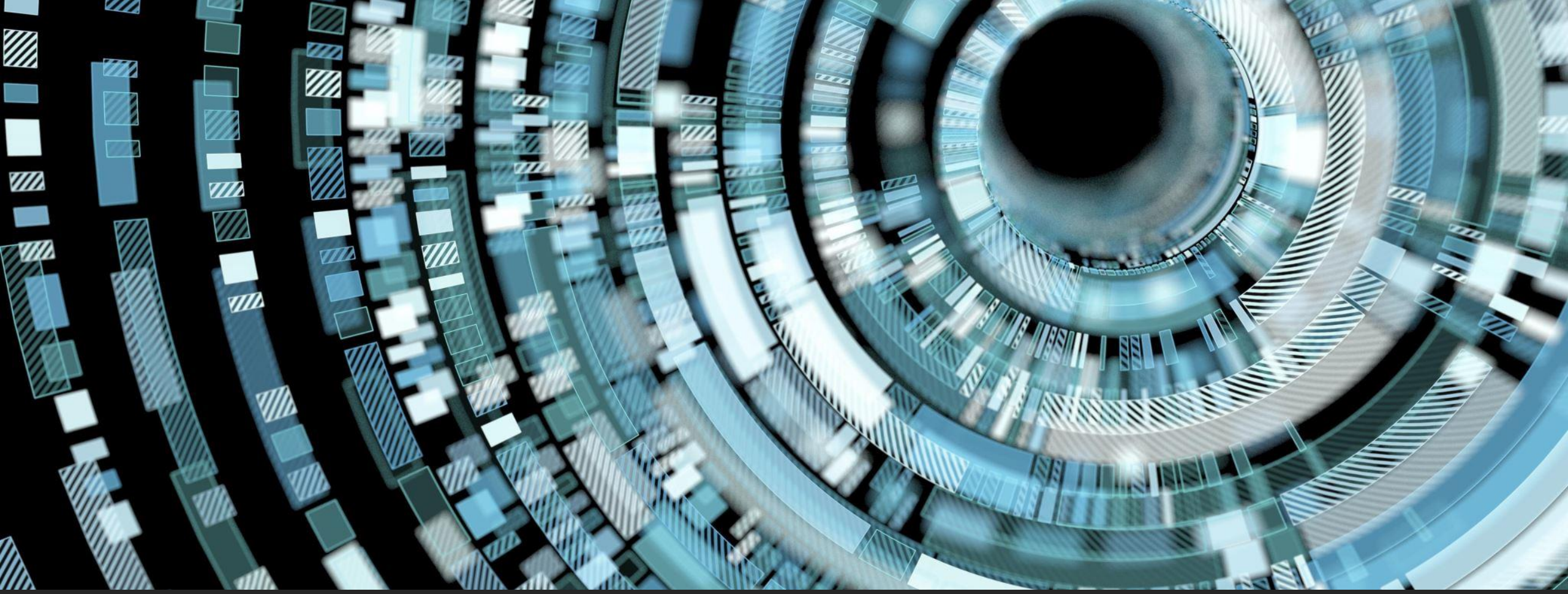
The Dataset

From December 2017 to end of November 2018, the dataset informs of how many bikes have been rented for every hour.

We know the **date** of the day, how was the **weather** in Seoul, what **season** it was and if that was a day of **Holiday** or not.

Attribute Information:

Date : year-month-day
Rented Bike count - Count of bikes rented at each hour
Hour - Hour of he day
Temperature-Temperature in Celsius
Humidity - %
Windspeed - m/s
Visibility - 10m
Dew point temperature - Celsius
Solar radiation - MJ/m2
Rainfall - mm
Snowfall - cm
Seasons - Winter, Spring, Summer, Autumn
Holiday - Holiday/No holiday
Functional Day - NoFunc(Non Functional Hours), Fun(Functional hours)



DATA-VISUALISATION

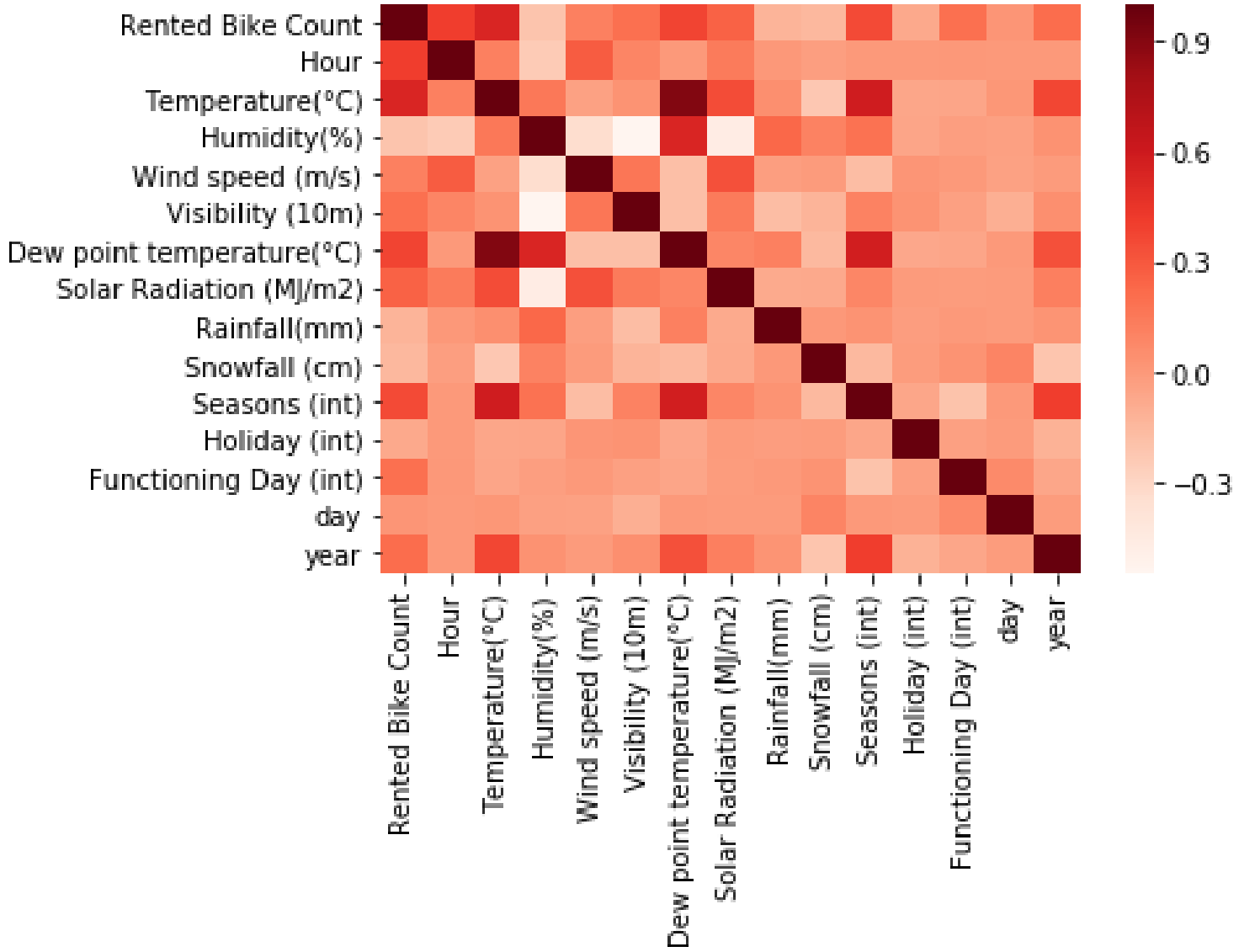
The problem

We may wonder how do the weather, holidays and season affect the rent of bikes in order to provide enough bikes to Seoul and limit the waiting time for everyone when they need a bike.

What are the interesting variables ?

To begin, we use a correlation matrix and we observe that the most interesting variables to be analyzed are the **Temperature**, the **season**, the **hour** and the **Dew Point temperature.**

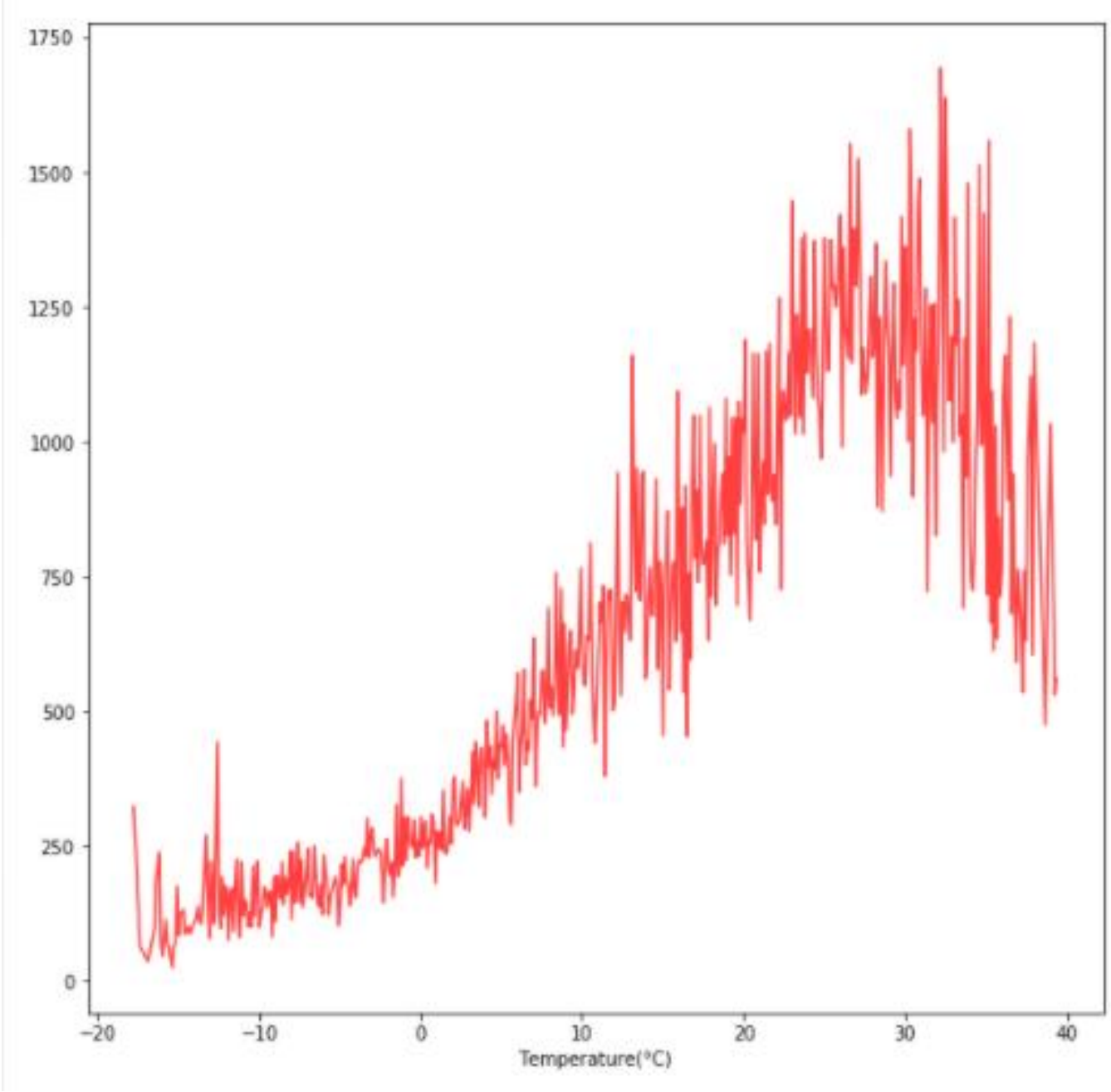
Let’s analyse these correlations !



Correlation Rent/ Temperature

We see there are lots of rents when the temperature is situated between 22 and 34°C

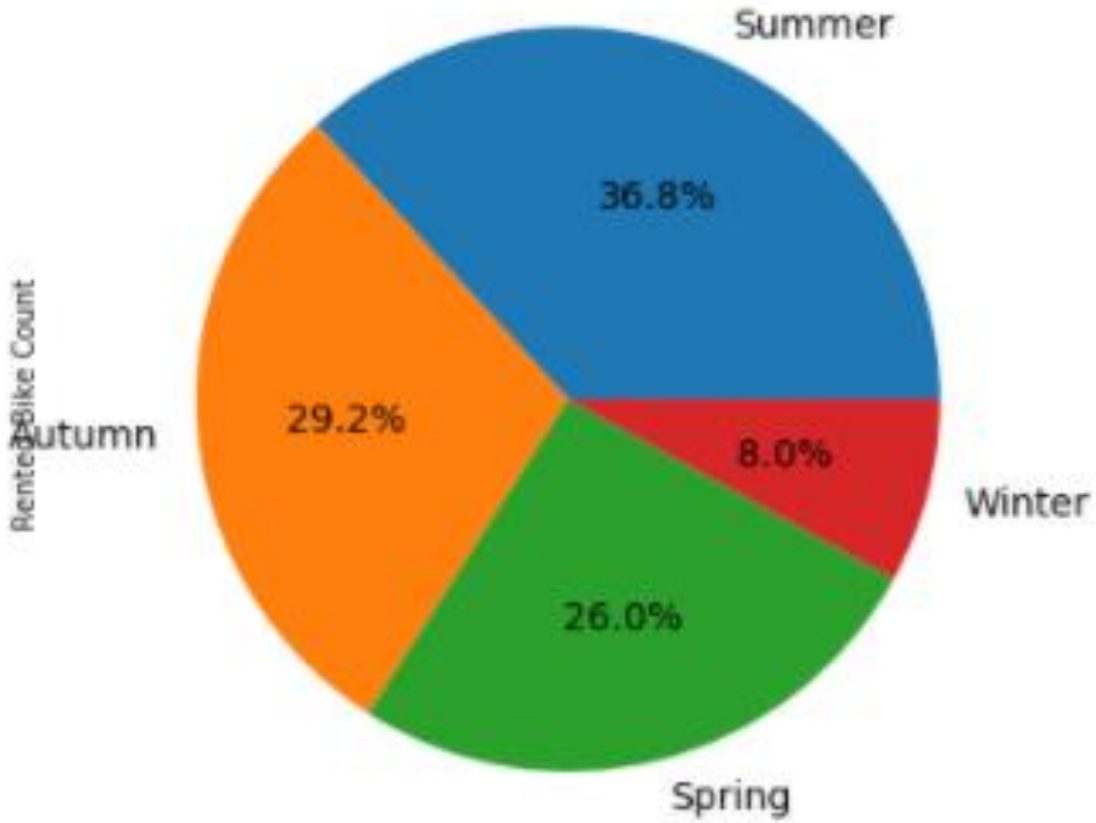
When the temperature are too low or too warm, people use to not rent bikes.



Correlation Rent/Season

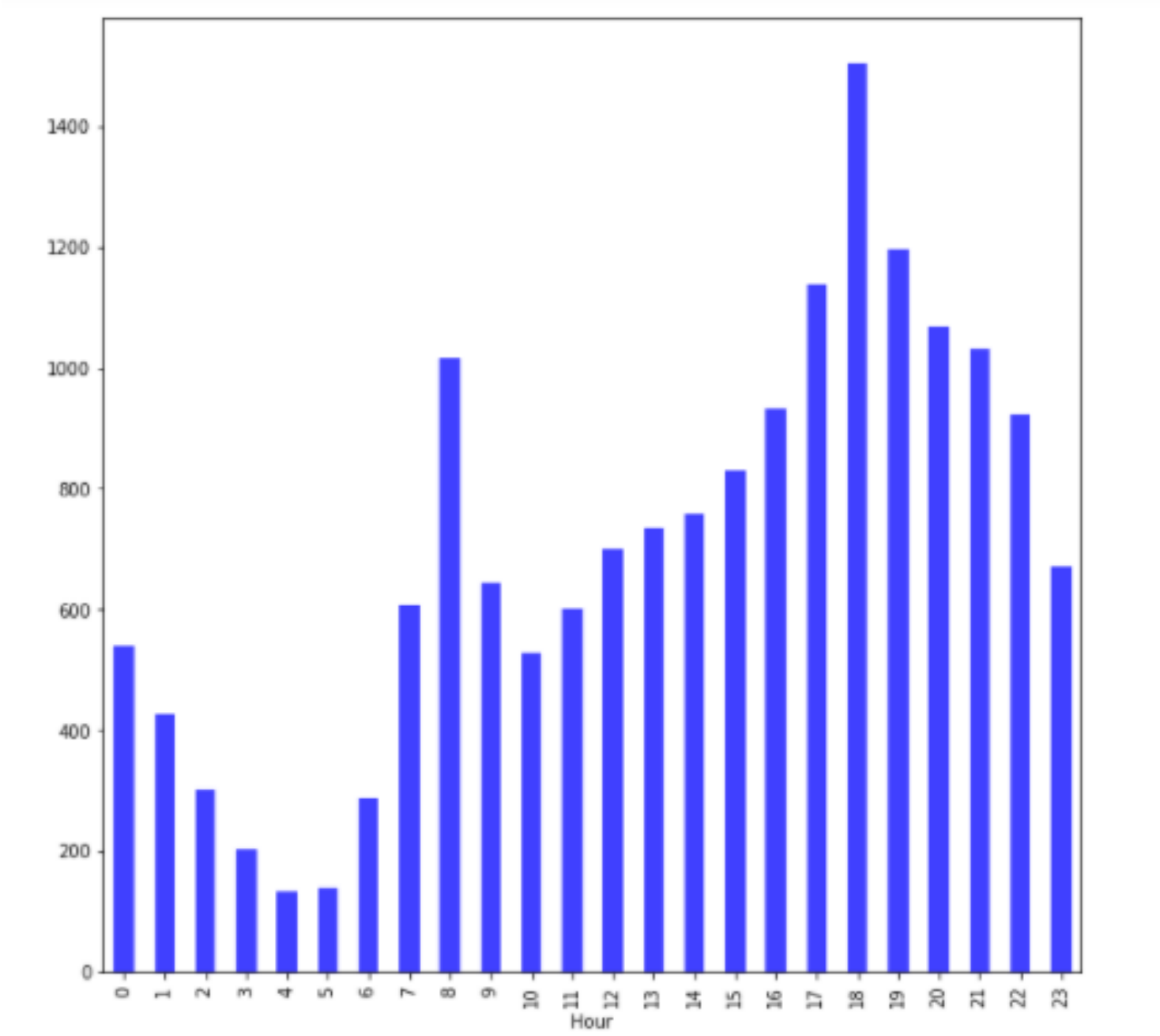
We can observe there are fewer rents in winter, that matches with our last observation.

There are low temperatures in winter and as we just observed, rare are the people to rent a bike when it is cold.



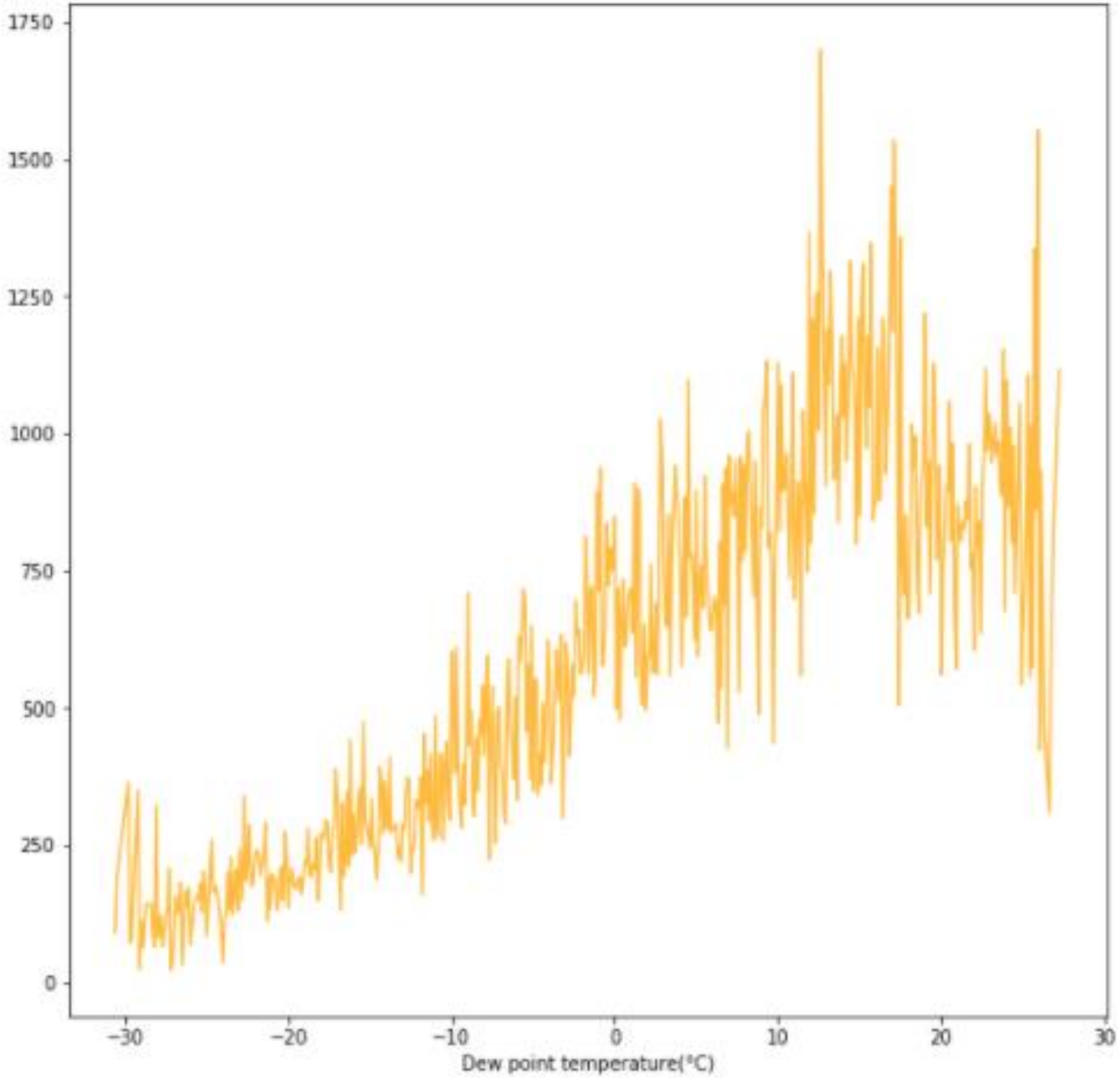
Correlation Rent/hour

We observe some peaks at 8 am and between 5pm and 9pm.



Correlation Rent/Dew Point Temperature

The rents are very high when the dew point temperature rises.



MODELISATION

Applying algorithms

After creating a training and a testing set, we can apply some algorithms to our dataset and see which one has the best accuracy.

The algorithms we are going to use are:

1) Linear Regression

2) Logistic Regression

3) Support Vector Regression

4) Random Forest Classifier

Linear Regression

While applying this algorithm to the model, we observe an accuracy of **52,16% on the training set** and **48,79% of the testing set.**

That is not satisfying enough. Let's see with the Logistic Regression.

```
#Affichage
print(score_total)
print(score_partiel)
```

```
[0.5216649738375605]
[0.4879504160340503]
```

Logistic Regression

We observe an accuracy of 0,2% on the training test and 0,07% on the testing set.

It is obvious that we made a mistake in the implementation of the algorithm, and the results should be better than the Linear Regression.

Random Forest Classifier

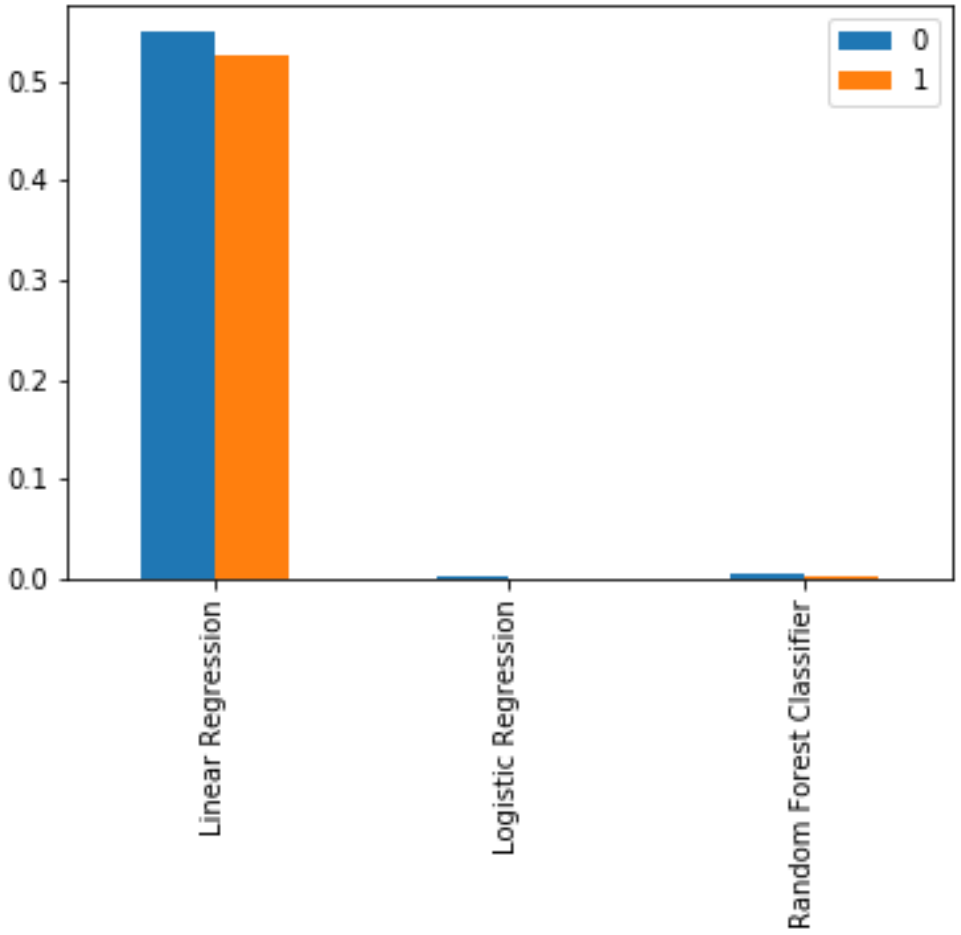
We observe an accuracy of 0,4% on the training test and 0,2% on the testing set.

It is obvious that we made a mistake (again) in the implementation of the algorithm, and the results should be better than the Linear Regression and the Logistic Regression.

Conclusion

The graphic comparison of these algorithms shows that the most accurate one is the Linear Regression, because our results are distorted.

We can expect the best algorithm to be the Random Forest Classifier.



TRANSFORMATION IN API DJANGO