

# PYTHON PROJECT



## SEOUL BIKE RENTAL PREDICTION



2021 - 2022

BORDAS ALEXANDRE & CARVAL NICOLAS

# PROJECT OVERVIEW

---

- **Introduction**
- **The Dataset**
- **Data Exploration**
- **Data Preprocessing**
- **Data Visualization**
- **Machine Learning**
- **Project API**
- **Conclusion**

# INTRODUCTION

This project consists in the final project of the course 'Python for Data Analysis' being part of our degree at ESILV.

Our team, composed of Nicolas and Alexandre, was given a dataset to work with. The goal of this project was to explore and analyze this dataset by applying the knowledge, skills and tools acquired along the course.

At first, we started by discovering the data via some reading, exploration and visualization in our Jupyter Notebook. We then arranged our dataset so it would fit our Machine Learning models. We improved our results by trying different models and by using different parameters. We used graphs through the whole project so we could have a visualization of our results.

Finally, we implemented an API in which one's can test our final model and get a prediction by entering specific parameters.

# THE DATASET

Our dataset represents some information about bike rentals in Seoul within a year. Our goal is to study the link between the data and the number of rented bikes.

It contains 8760 instances, each of them representing an hour of a specific day.

The data consists of 11 numerical and 3 categorical attributes. Our 'target' feature for our analysis is the 'Rented Bike Count' one. It represents the number of bikes rented in 1 hour.

This problem is a regression problem : we will try to predict how many bikes will be rented at a specific time of the year.

# THE DATASET

These are the different attributes composing our dataset:

**Date** *year-month-date*

**Rented Bike Count** *number of rented bikes at each hour*

**Hour** *Hour of the day*

**Temperature** *Celsius*

**Humidity** %

**Windspeed** *m/s*

**Visibility** *10m*

**Dew Point Temperature** *Celsius*

**Solar Radiation** *MJ/m<sup>2</sup>*

**Rainfall** *mm*

**Snowfall** *cm*

**Seasons** *Winter, Spring, Summer, Autumn*

**Holiday** *Holiday/No holiday*

**Functional Day** *if the bike rentals are operational or not*

# DATA EXPLORATION



Every columns are in the correct format except the Date one, thus we will convert it later.

As previously said, each instance is an hour of the day : when grouping them all together day by day, we can confirm that our dataset is composed of 365 days.

When the 'Rented Bike Count' equals 0, it corresponds to a non-functioning day. In fact, there are 13 days in the year that are non-functioning days. These days represent all the days without any bike rented.

Our dataset did not contain any missing values.

# DATA PREPROCESSING



As our target feature is 'Bike Rented Count', there is no point in keeping the non-functioning days for our analysis, as we know that the number of rented bikes will always be 0. Thus, we drop the column 'Functioning Day'.

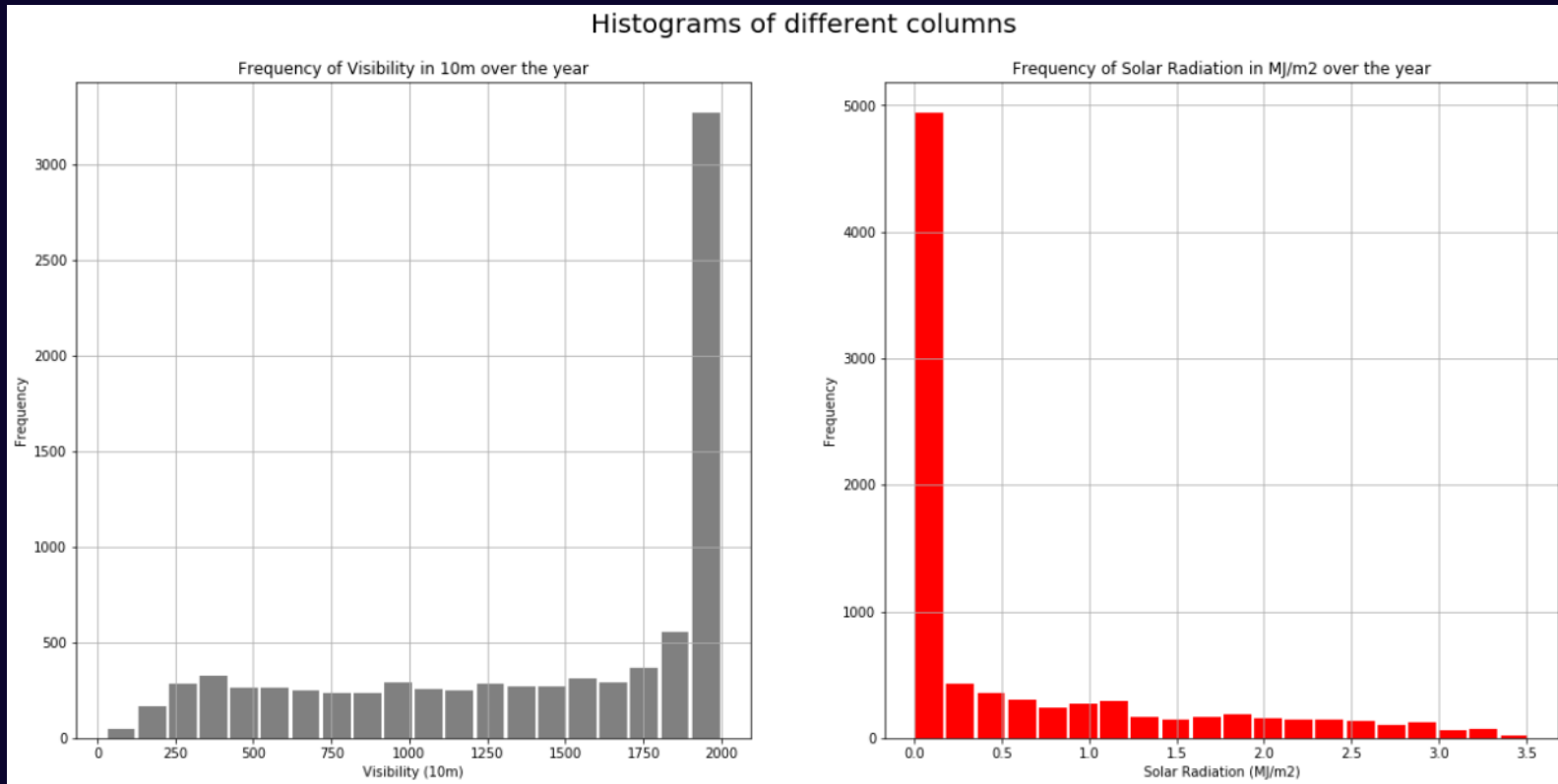
We have 353 days left, which corresponds to  $365 - 12$ . There is no mistake. In fact, if you look at all the non-functioning days, you'll notice that for 06/10/2018 there are only 7 hour in that day that aren't functioning.

We then convert the 'Date' column into the correct format in order to be able to use it later in the project.

We decide to add a 'Month' column so we could analyze more closely our dataset.

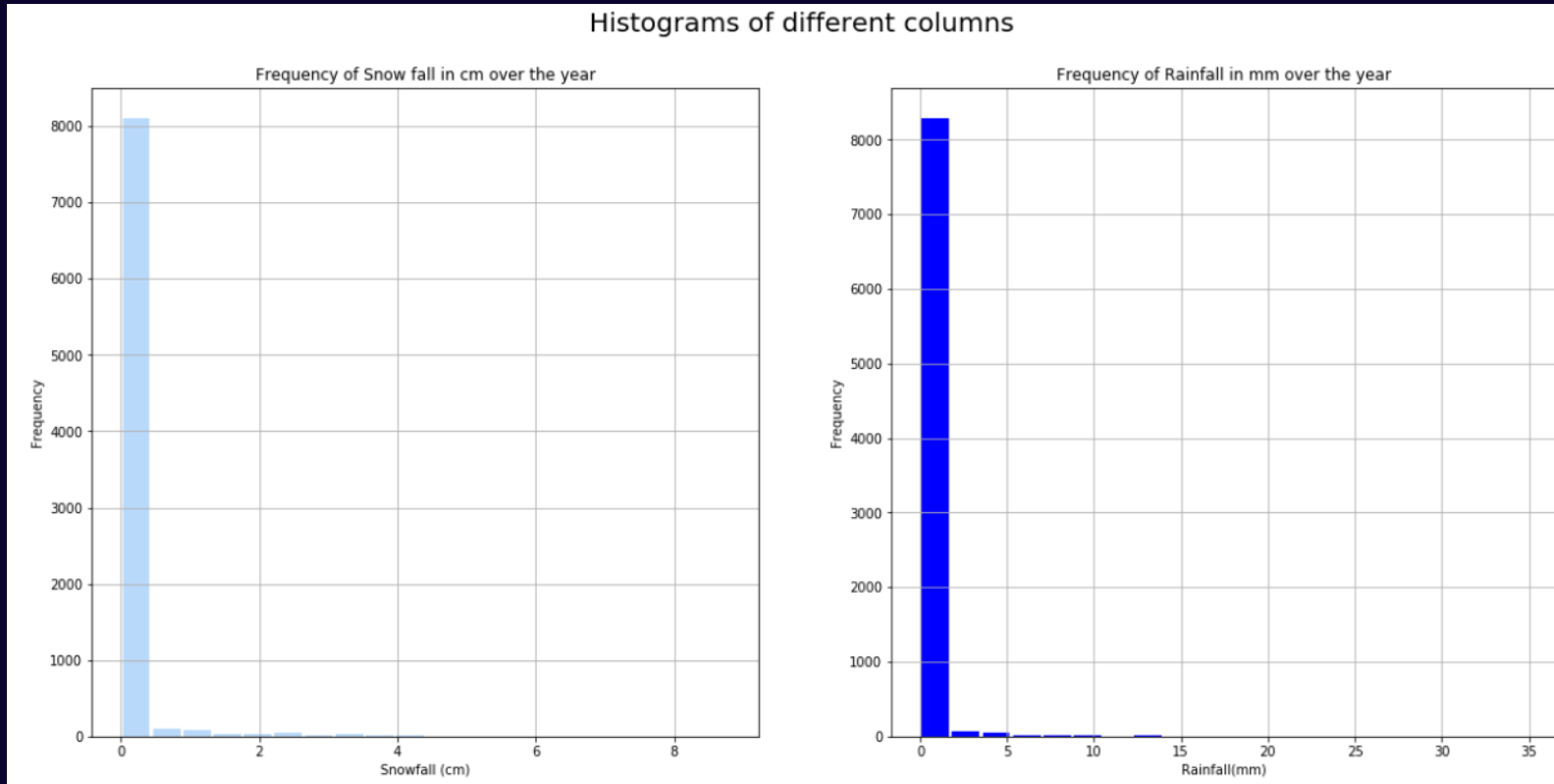
# DATA PREPROCESSING

We already know what to expect of standard columns such as date, hour, seasons and others. However, some may be interesting to analyze aside from others as they may not be that much variated. So, we're looking at these columns more closely :





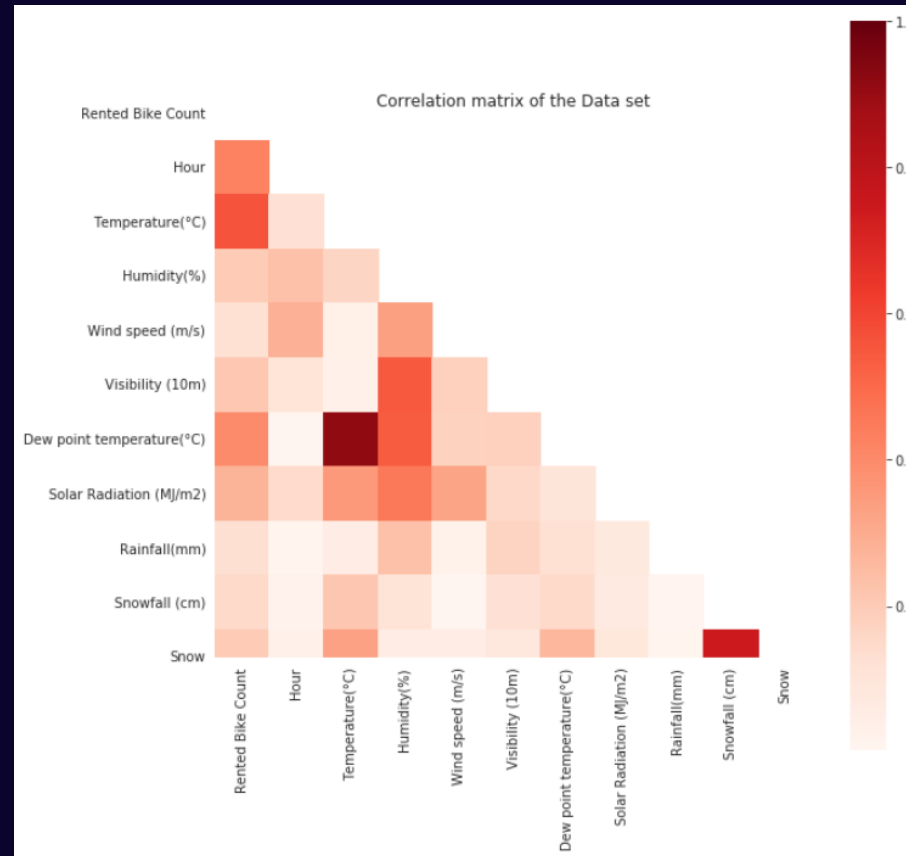
# DATA PREPROCESSING



**Snow and Rain values aren't much varied, thus let's create 2 new columns for rain and snow to convert them into categorical variables.**

# DATA PREPROCESSING

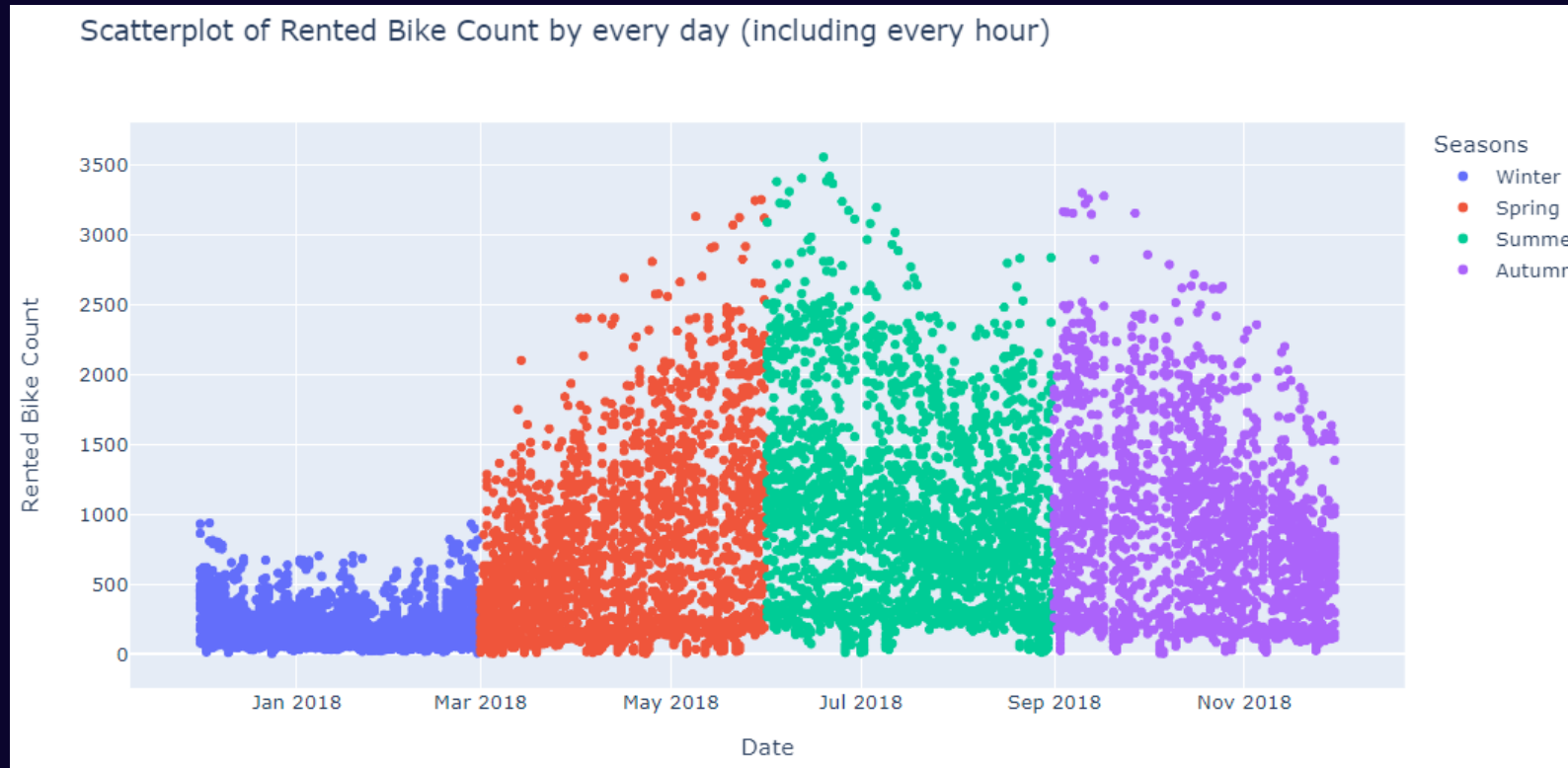
Now let's check the correlation between every variables :



For our regression model we will decide to drop dew point as it is too correlated to temperature and temperature is better correlated with rented bike count. we notice an improvement of the correlation between Snow info and Rented bike count, as it is now a Boolean (Snowfall (cm) vs Snow).

# DATA VISUALIZATION

Here are some of our most interesting graphs

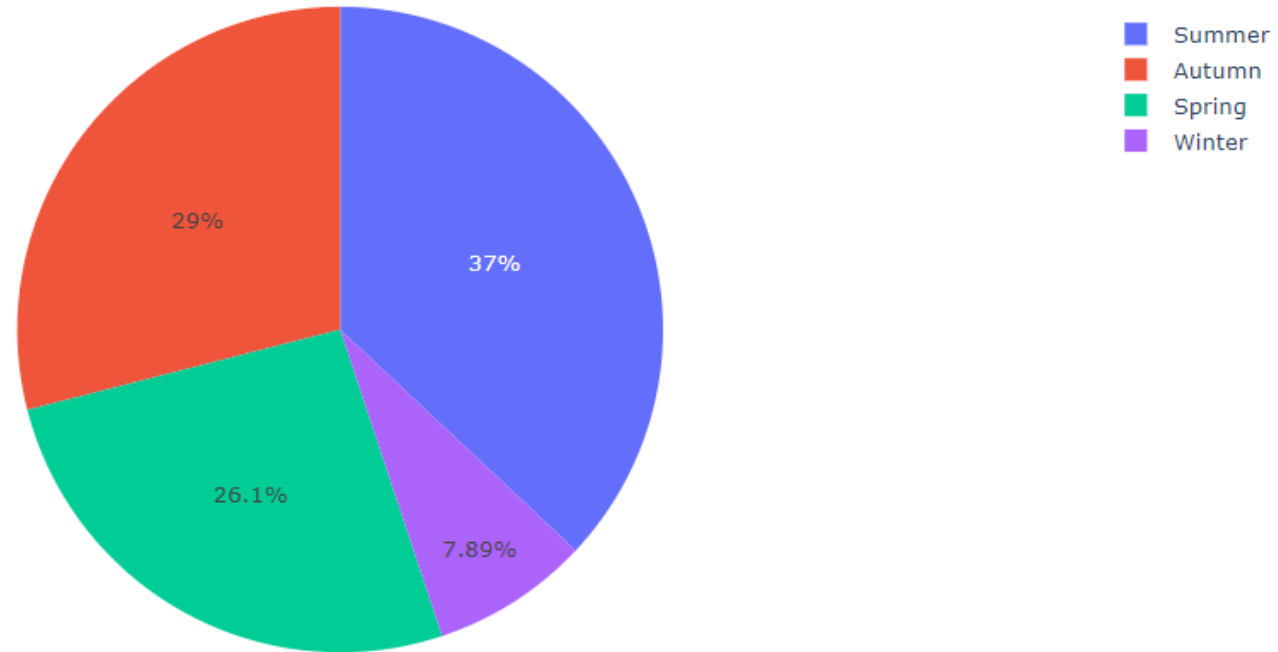


We can clearly see that winter is the season that stands out of the others. The values are massed together from 0 to 500 bikes per hour. Which is pretty low compared to other seasons.

We can also see that there is much more volatility when we get closer to summer, values are really dispersed and can reach 3500 bikes per hour.

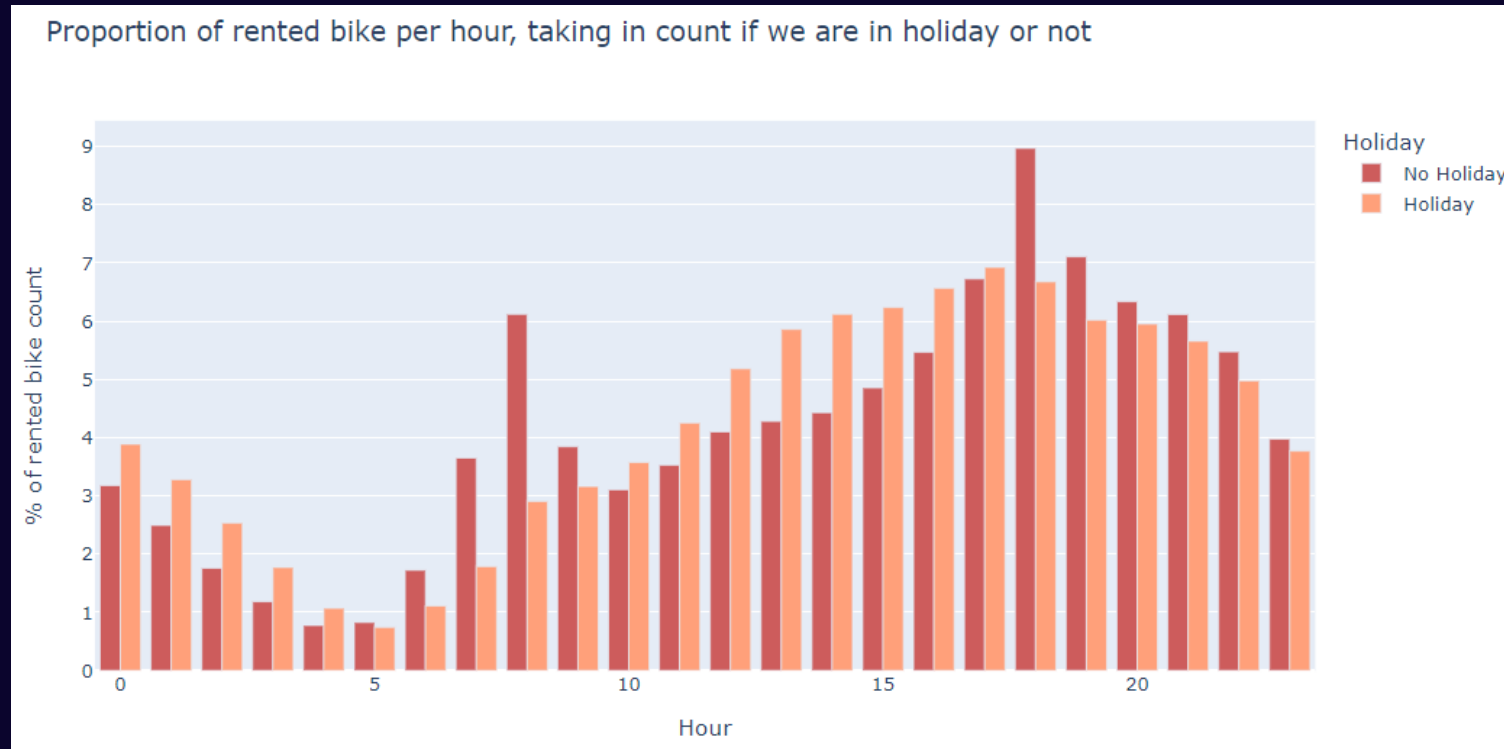
# DATA VISUALIZATION

Pie Chart representing the proportion of total rented bike by seasons



**As we expected from the previous graphs, winter doesn't even represent 1/10 of the total bikes rented and summer is the dominant season with more than 1/3 of the proportion.**

# DATA VISUALIZATION

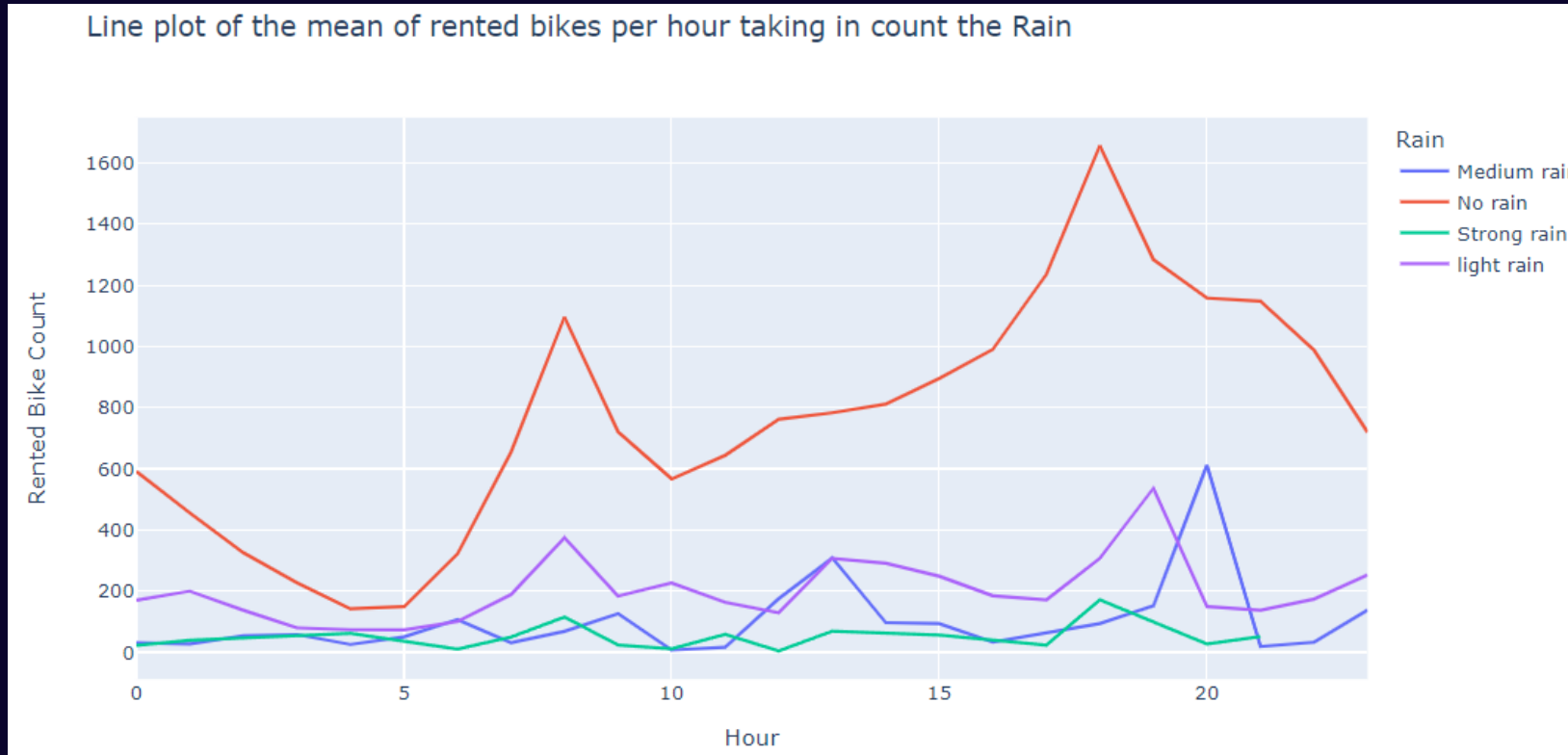


Every columns for holiday adds up to 1, same for non holiday columns.

This graph enables us to understand the trend underlying holidays. As there are 10x more non holiday row, we don't see the impact if we don't weight the values and then can see the trend.

It follows the same stable pattern for both categories: a drop during the night and a peak during afternoon. However, we can see that for non-holiday days, 8h and 18h are much more dominant. this is explained by the beginning and the end of days for most workers. They are travelling home or going to work. Which of course isn't the same during holidays.

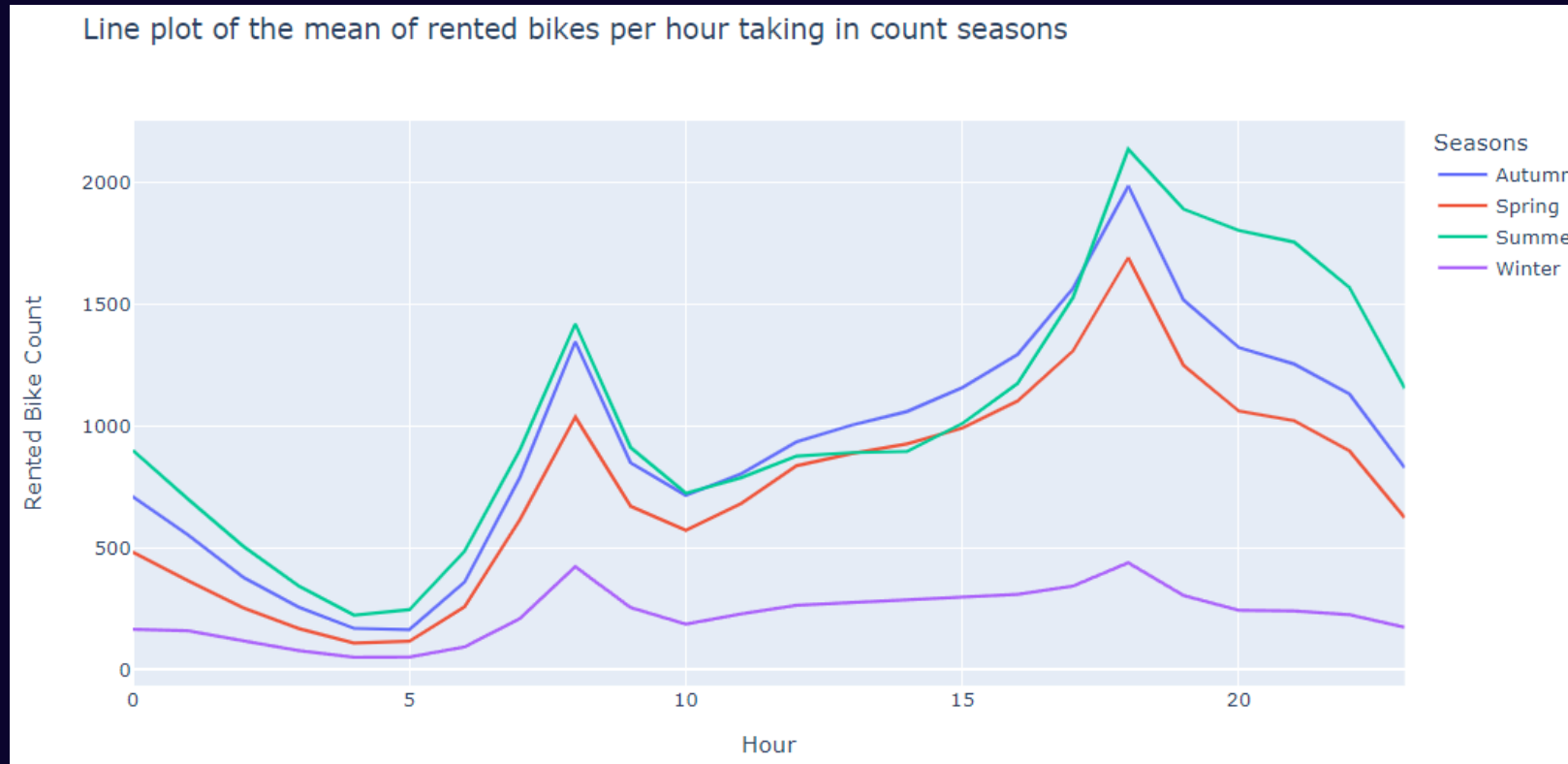
# DATA VISUALIZATION



**It doesn't rain much in the year. Thus, we use the mean of rented bikes to interpret the trend of this column.**

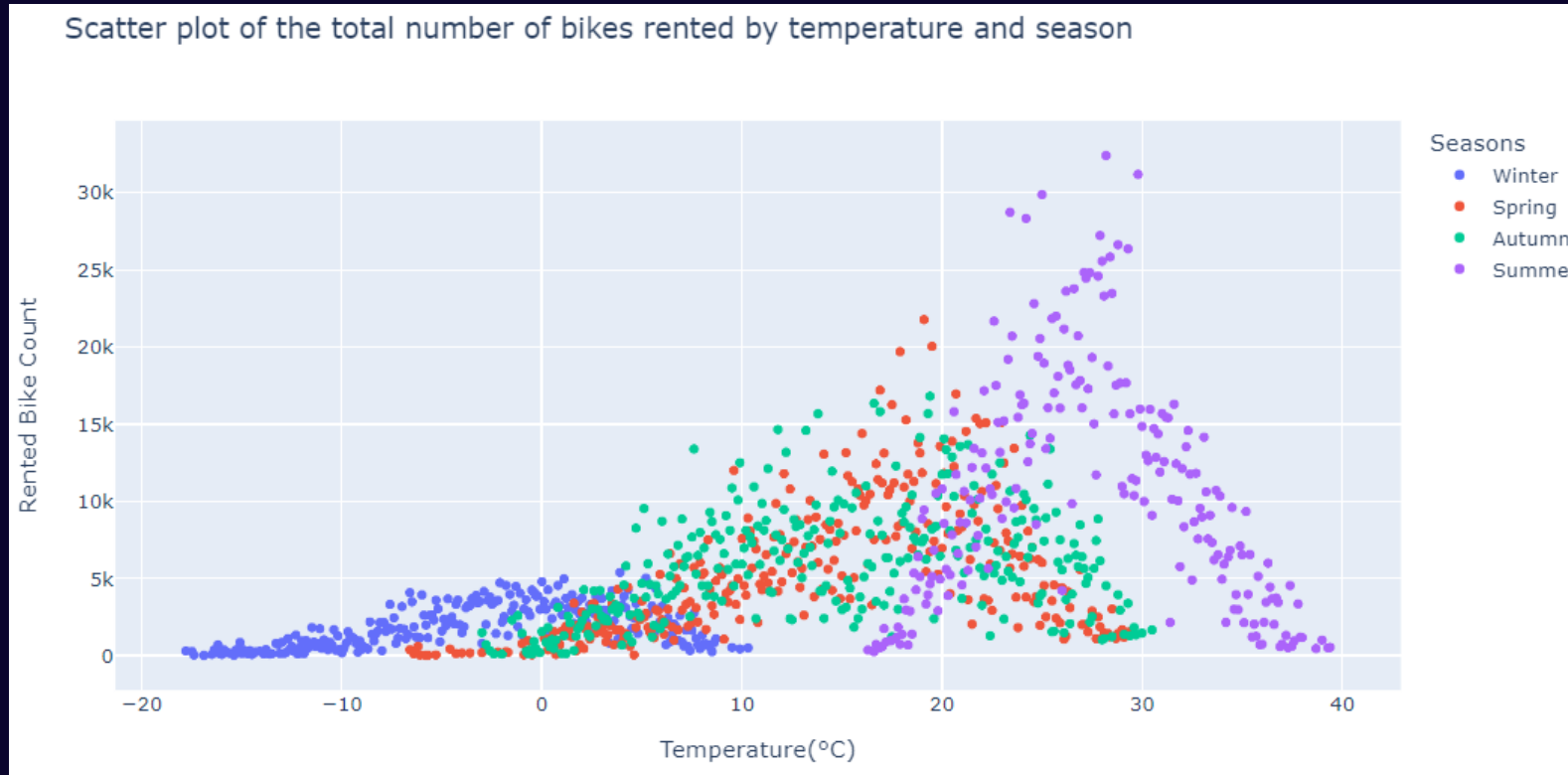
**We can clearly see that the rain affects the number of rented bikes. When it rains, people don't rent bikes and probably use public transports. However, we can still distinguish the 8h and 18h peaks when it rains.**

# DATA VISUALIZATION



Same as before, we clearly see the trend that happens at 8h and 18h every day. We also see that every seasons seems to follow the same pattern except winter. However, at night we see that there are more bike rented in summer, it may be due to the higher temperature at night and the sun going down later.

# DATA VISUALIZATION



We can see that the temperature is very correlated to the rented bikes, as it gets higher and closer to 20 °C the number of bikes rented are higher in general. If it surpass 30 °C, then it decreases drastically as the temperature is maybe too hot for people to go biking.

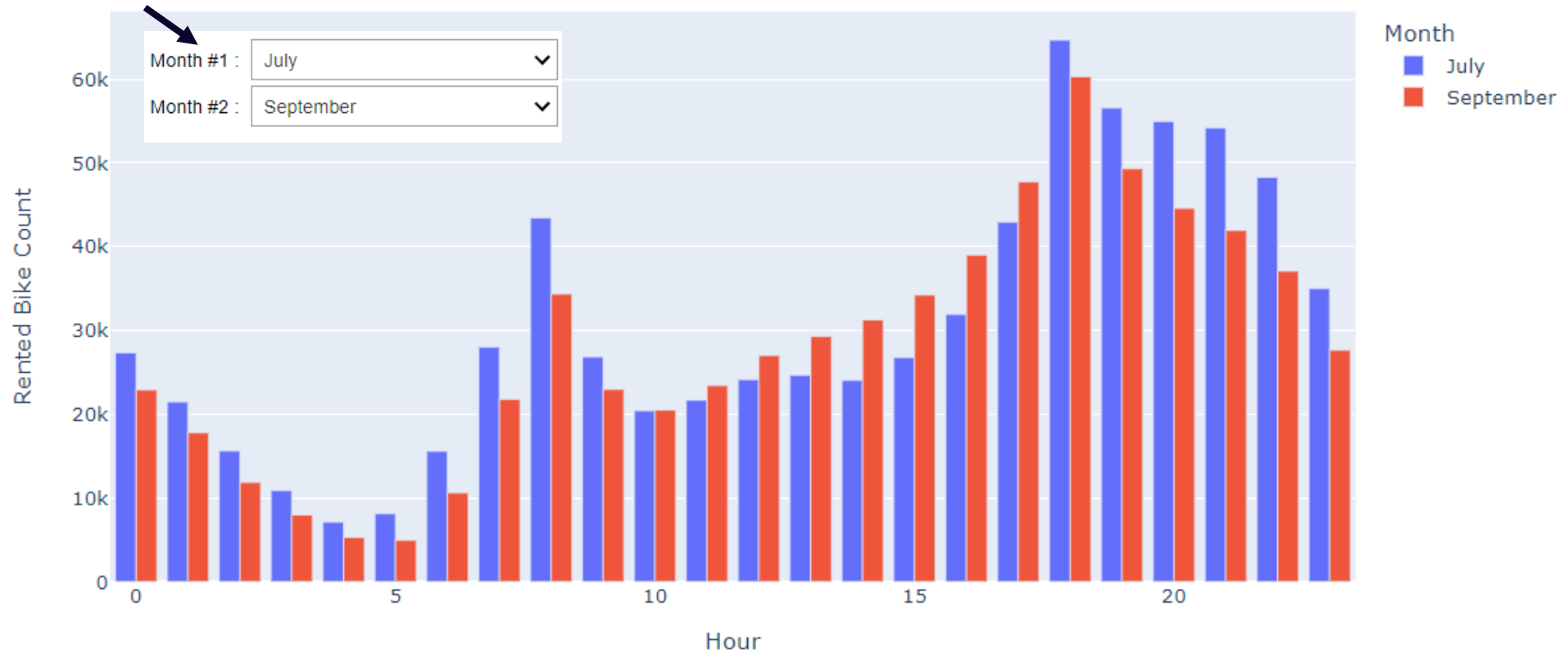
In winter, temperatures are low thus rented bike number is low, and we get the same conclusion as before for the other seasons.



# DATA VISUALIZATION

Bar plot of the total number of bike rented by hour between July and September

*Interactive graph where you can select 2 different months to compare*



**This graph is helpful when we want to compare two months together side by side.**

# MACHINE LEARNING

The goal of this project is to predict an output when we give a model some parameters. In our case, we want to predict how many bikes will likely be rented on given parameters (hour, season...).

Therefore, we use machine learning : we will try different models that will predict the output with a certain accuracy. In the end, we aim to seek the best machine learning model that fits our dataset and that gives us the most accurate prediction. We will save it to a Pickle file in order to use it in our flask API.

## Here are the different steps:

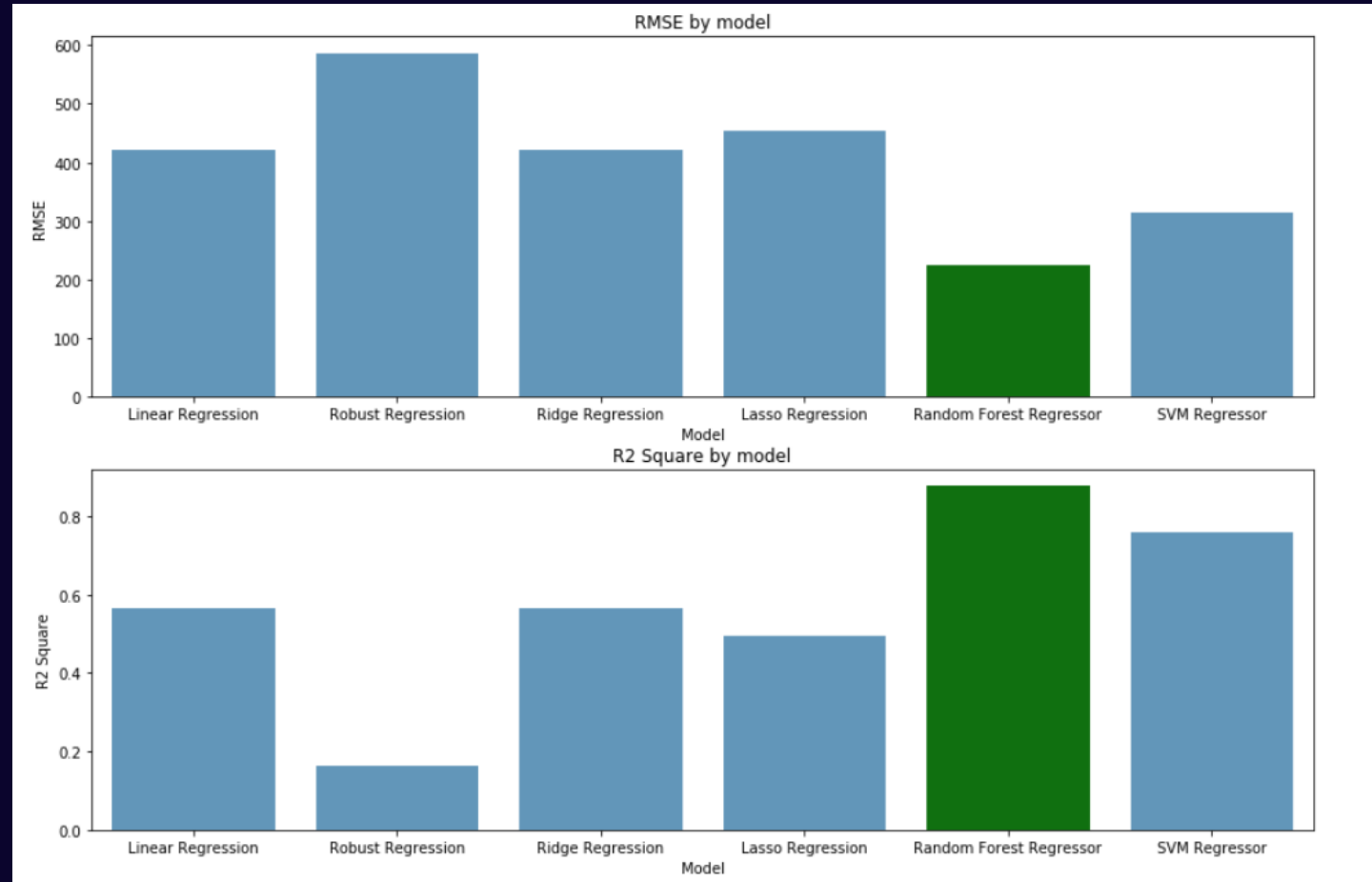
- Preprocessing
- Definition of results storing methods
- Regressions model :
  - > Linear regression
  - > Robust regression
  - > Ridge regression
  - > Random forest
  - > SVM regression
- Model selection
- Model tuning
- Model saving

# MACHINE LEARNING

	Model	RMSE	R2 Square
1	Robust Regression	482.351741	0.431221
3	Lasso Regression	453.821408	0.496516
2	Ridge Regression	421.252070	0.566190
0	Linear Regression	420.896029	0.566923
5	SVM Regressor	314.114824	0.758792
4	Random Forest Regressor	225.929381	0.875215

Here are all the results we got from our different models on the test set.

# MACHINE LEARNING

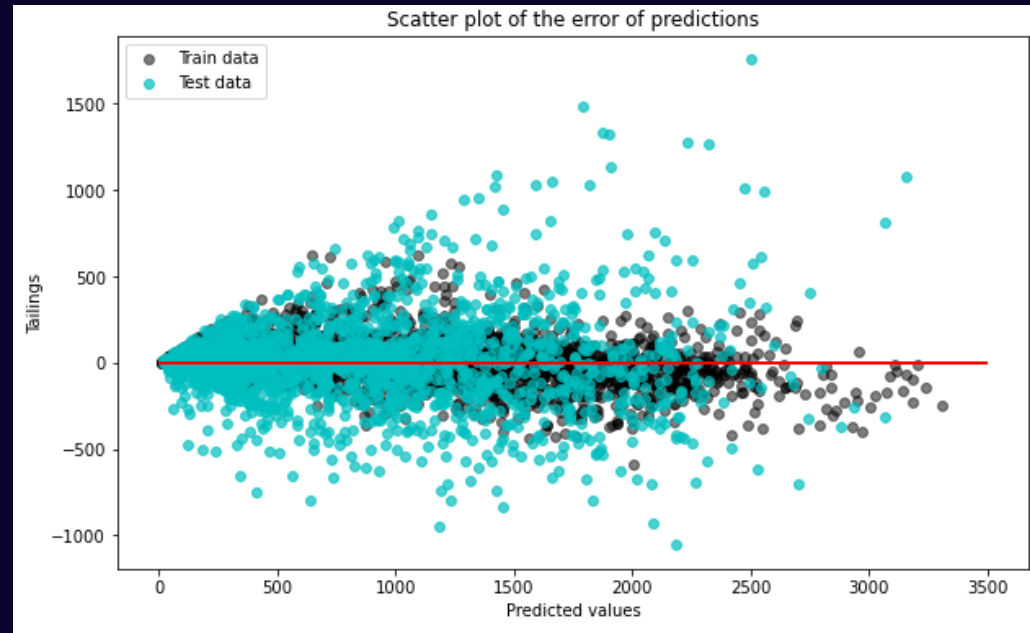


**We put the previous results into a graph so we could see better which one stands out.**

**We decided to go with RandomForest, as it got the best results.**

# MACHINE LEARNING

## Looking at RandomForest closer :

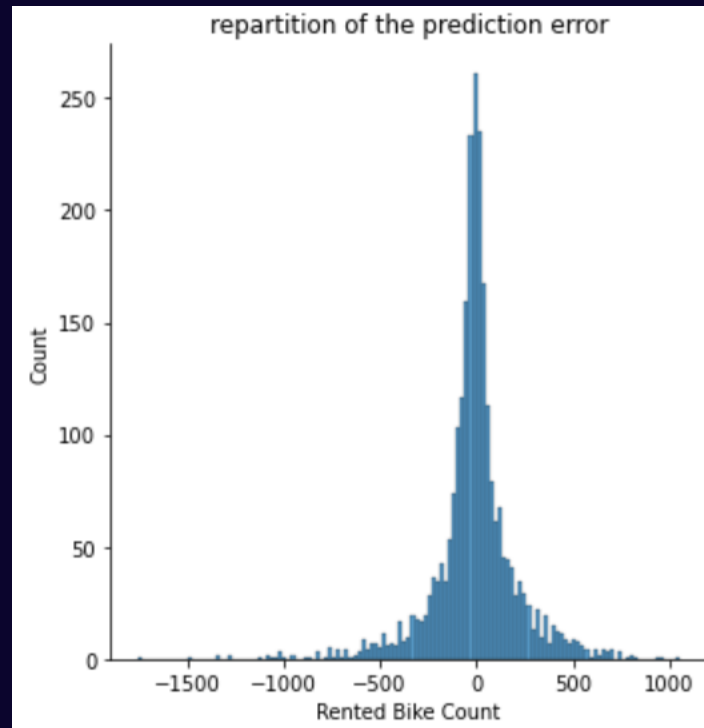


**This plot enables us to understand for which kind of values we have the most significant errors. On the train set we see that the model is pretty stable, following the red line, errors are massed around 0 with max error of 500 bikes.**

**However, for the test set we see that the error spreads a lot more when predicting big values, the maximum error is now higher to 1500 bikes. Again, it is pretty correct for values predicted between 0 and 1500.**

# MACHINE LEARNING

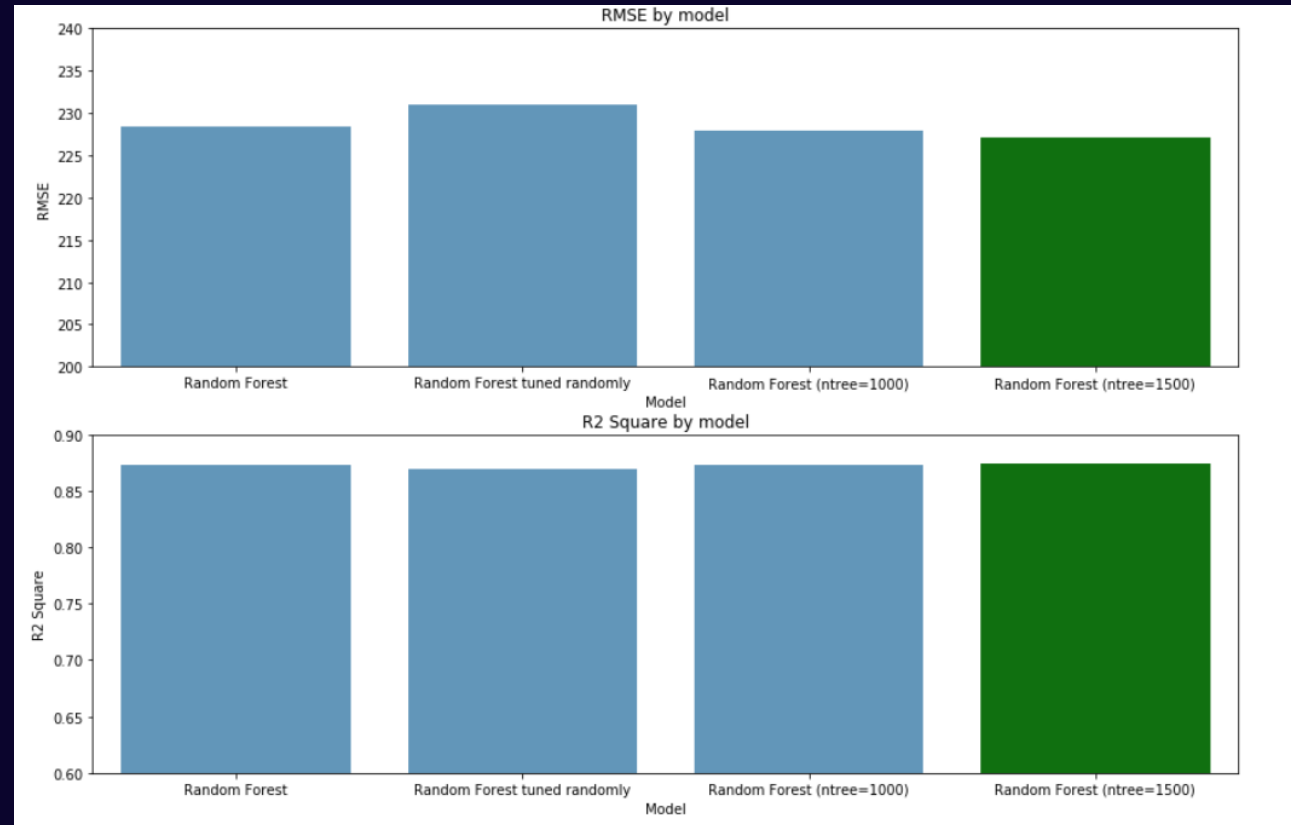
Let's check if the errors observed are normal, if our model is correct, and we just need to tune it or not.



It looks like a normal distribution, no outliers are present, it centered in 0. Thus, our model is not incorrect we can continue with its improvement.

# MACHINE LEARNING

We then tuned our RandomForest model with different parameters in order to maximize it. We used a random gridsearch for tuning one of our models. For the others, we just tried different n-trees.




The number of estimators was the only parameter of improvement for the model, it is great to increase it until 1500, as the improvement is not significant anymore after this level (increases the R2 square by 0.000001).

# PROJECT API

We used Pickle in order to be able to do a prediction and use our model in our Python file with Flask.

## SEOUL BIKE LOCATION PREDICTION

BORDAS ALEXANDRE & CARVAL NICOLAS



Result : ...

Hour  
ex : 8

Temperature (°C)  
ex : 24

Humidity (%)  
ex : 70

Wind speed (m/s)  
ex : 0.3

Visibility **YES** NO

Sun **YES** NO

Holiday **YES** NO

Snow **YES** NO

Rain **YES** NO

WINTER

SPRING

SUMMER

AUTUMN

Month

PREDICT

Here is what our API looks like.



# CONCLUSION

After analyzing the data and removing the irrelevant columns, we tried different kind of machine learning models to see which one suits our dataset the best. In the end, we can clearly see that the Random Forest model provides us the best results ( $r^2:0.87$  / RMSE:222) on the test set, and that is why we take the decision to use this model in our flask API.

After trying to improve this model, we finally concluded that the number of estimators was the only parameter of improvement for the model, it is great to increase it until 1500, as the improvement is not significant anymore after this level (increases the R2 square by 0.000001).

Our model is efficient when predicting values under 1500, when it is larger it doesn't match reality. This can be the result of several rows of high value (higher than 1500) not big enough to train the model correctly for huge values.

If we take the MAE metric, we can tell that our predictions are correct with +-139 bikes of error.

In fact, if we look at the dataset, we see that 75% of the rows corresponds to values (number of bikes rented) under 1085 which is too narrowed.

We trained our model on a dataset excluding values above 1500 and the MAE was divided by 2, without losing too much of R2 square. It shows that if we had a dataset with more diversified values for the target, then we would surely have better results.

We tried to complement our dataset and find some more data so we could have a more diversified dataset, but the South-Korean government provided only this dataset, and we would need to be fluent in Korean to translate and navigate on the website.