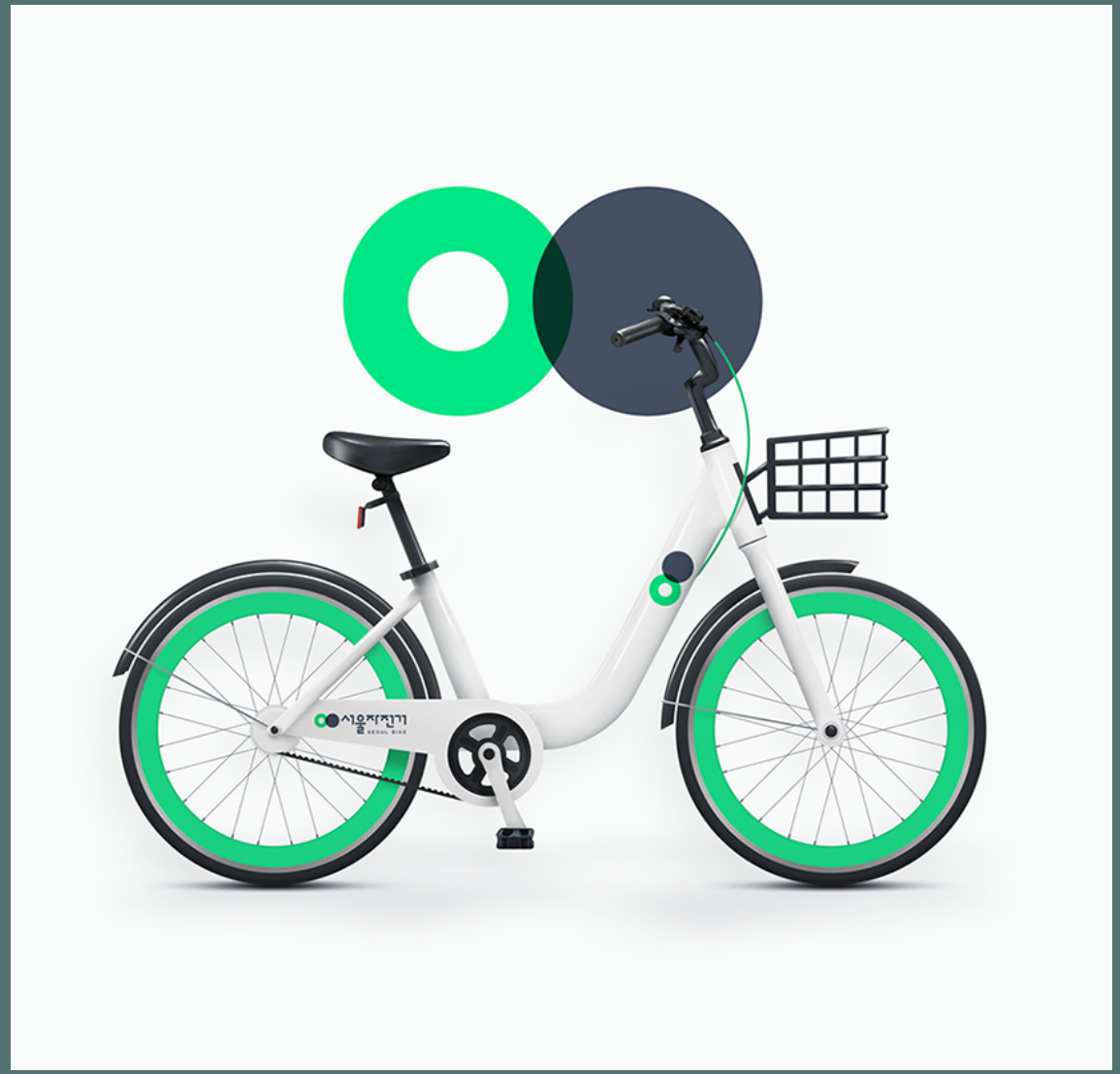
ESILV

SEOUL BIKE SHARING DEMAND

Python for Data Analysis

Final project



Mariam BARHOUMI

Nacima BEN SOUNA

Mentor : Benjamin BEJNBAUM

SUMMARY

1.INTRODUCTION

Seoul is a city spread over more than 600 km2, six times the size of Paris. It is necessary to take precautions when it comes to traveling, especially at peak hours, which generate end less traffic jams in the city's streets.

Paris has its "vélib'" and, for a few years now, Seoul has its equivalent the "Ddareungi". Bicycles, which were not very visible until a few years ago, are now widely used in the South Korean capital and even in other provincial cities.

In this report we will study the bicycle rental data in Seoul for the year 2018. The objective of our analysis is to discover the factor(s) that determine the demand for self-service bicycle rentals, build statistical models and then try to make rental prediction based on the information and models available to us. Our data mining and analysis will be done in Python.

2. ABOUT THE DATA

The data we will analyze was extracted from the UCI Machine Learning Repository*. These bicycle rental data contain sample values over the period from December 01, 2017 to December 31, 2018.

The data set we will study contains 8760 observations and 14 variables.

*https://archive.ics.uci.edu/ml/datasets/Seoul+Bike+Sharing+Demand

Data Information :

- Date: day-month-year format
- Rented Bike count: Count of bikes rented at each hour (target)
- Hour: Hour of the day
- Temperature - 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 - No(Non Functional Hours)/Yes(Functional hours)

OVERVIEW OF THE DATASET

	Date	Rented Bike Count	Hour	Temperature(°C)	Humidity(%)	Wind speed (m/s)	Visibility (10m)	Dew point temperature(°C)	Solar Radiation (MJ/m2)	Rainfall(mm)	Snowfall (cm)	Seasons	Holiday	Functioning Day
0	01/12/2017	254	0	-5.2	37	2.2	2000	-17.6	0.0	0.0	0.0	Winter	No Holiday	Yes
1	01/12/2017	204	1	-5.5	38	0.8	2000	-17.6	0.0	0.0	0.0	Winter	No Holiday	Yes
2	01/12/2017	173	2	-6.0	39	1.0	2000	-17.7	0.0	0.0	0.0	Winter	No Holiday	Yes
3	01/12/2017	107	3	-6.2	40	0.9	2000	-17.6	0.0	0.0	0.0	Winter	No Holiday	Yes
4	01/12/2017	78	4	-6.0	36	2.3	2000	-18.6	0.0	0.0	0.0	Winter	No Holiday	Yes
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
8755	30/11/2018	1003	19	4.2	34	2.6	1894	-10.3	0.0	0.0	0.0	Autumn	No Holiday	Yes
8756	30/11/2018	764	20	3.4	37	2.3	2000	-9.9	0.0	0.0	0.0	Autumn	No Holiday	Yes
8757	30/11/2018	694	21	2.6	39	0.3	1968	-9.9	0.0	0.0	0.0	Autumn	No Holiday	Yes
8758	30/11/2018	712	22	2.1	41	1.0	1859	-9.8	0.0	0.0	0.0	Autumn	No Holiday	Yes
8759	30/11/2018	584	23	1.9	43	1.3	1909	-9.3	0.0	0.0	0.0	Autumn	No Holiday	Yes

2.1 DATA CLEANING

A preliminary cleaning of the data is performed, converting the date variable to datetime. Seasons, Holidays and Functioning Day features are also converted into factors to better represent their categorical nature.

The conversions are therefore :

- Date -> DateTime.
- Seasons -> Categorical: 1 (Winter), 2 (Spring), 3 (Summer), 4 (Autumn)
- Holiday -> Categorical: 0 (No Holiday), 1 (Holiday)
- Functioning Day -> Categorical: 0 (No), 1 (Yes)

We have also added columns linked to the date:

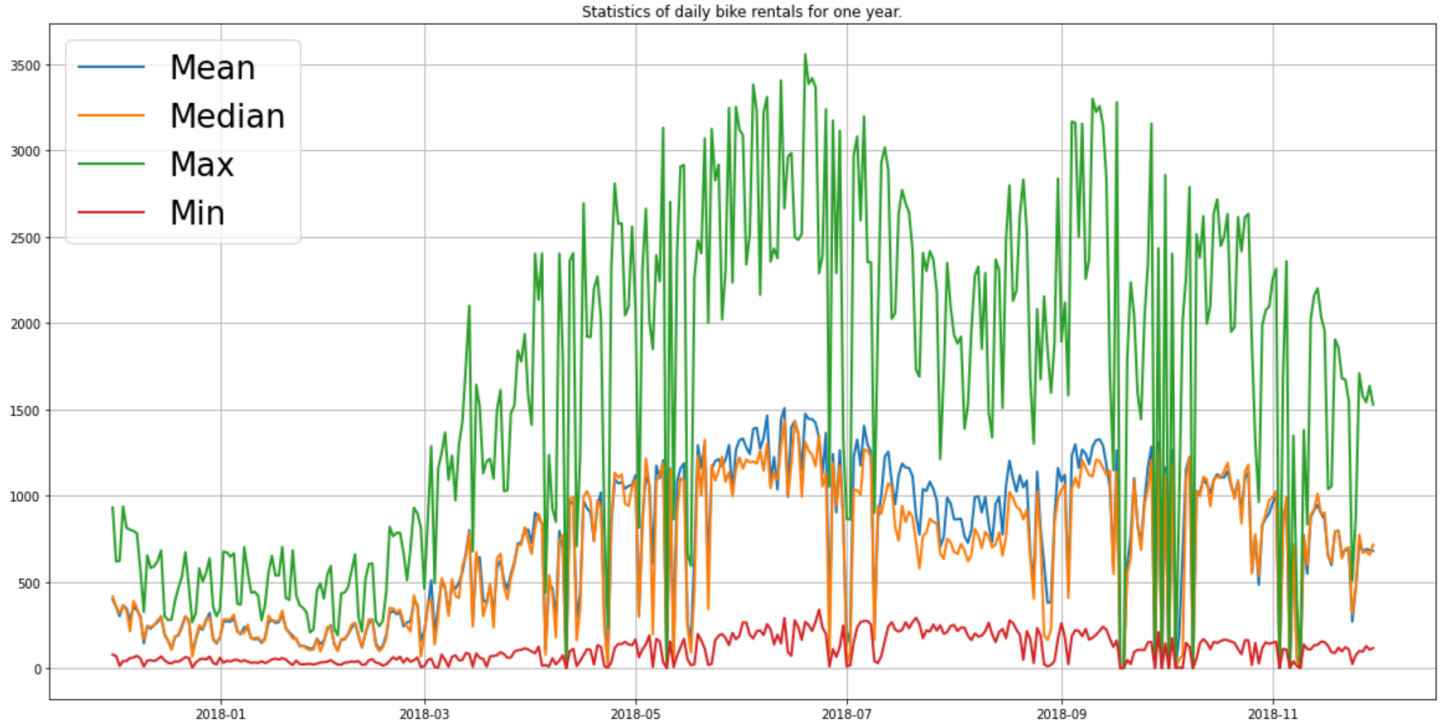
- Day : Values between 1 and 31
- Month: Values between 1 and 12
- Year: 2017 or 2018
- Weekday: 1 for Monday, 2 for Tuesday, 3 for Wednesday, 4 for Thursday, 5 for Friday, 6 for Saturday and 7 for Sunday.

There are no missing data, null values or anomalies in the dataset. The result of the data cleaning is a data set that still includes the 8760 observations and 18 variables.

3.EXPLORATORY DATA ANALYSIS

3.1 DATA VISUALISATION

- At first, we were interested in the daily evolution of bicycle rental in Seoul.



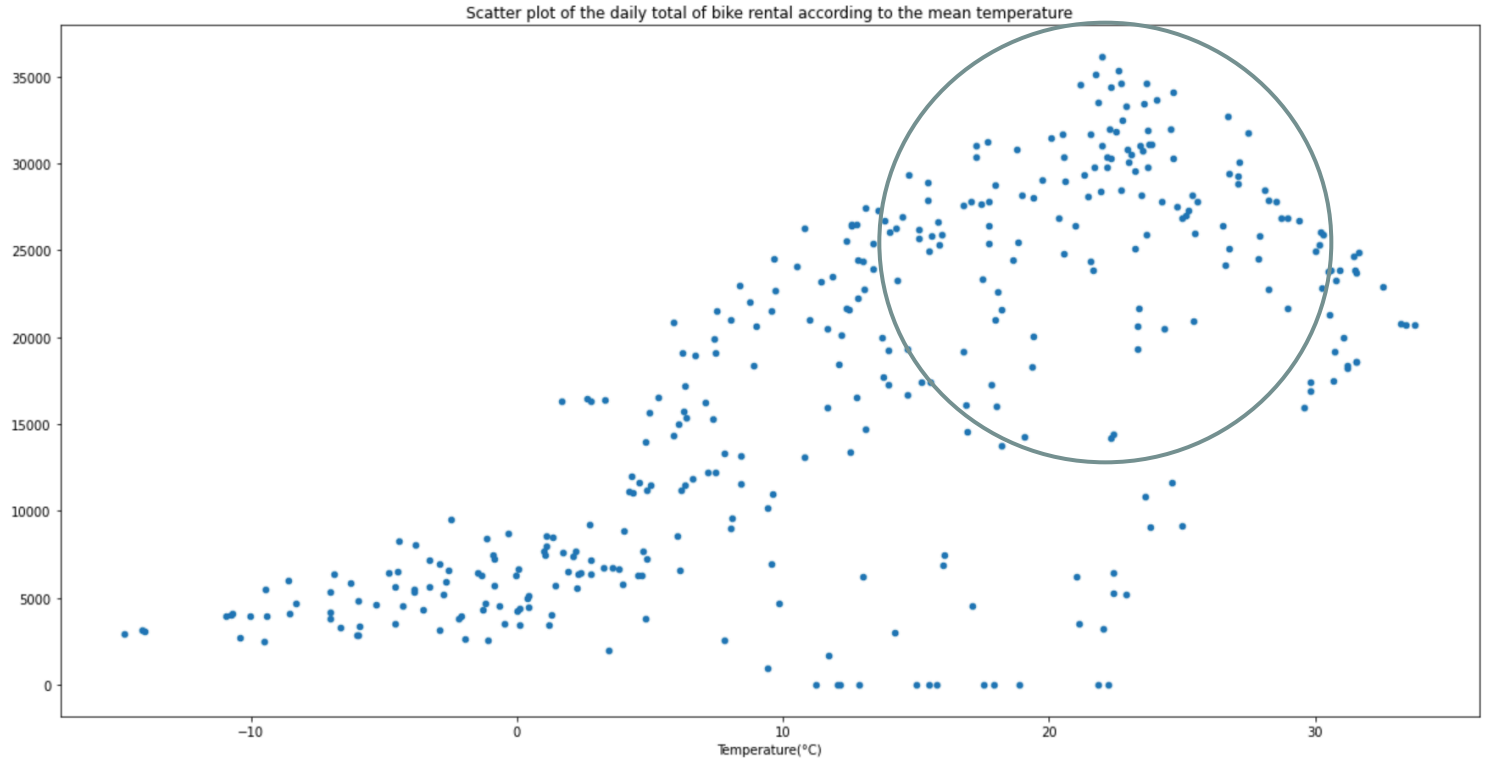
If we look at the curves of the maximum, minimum and average bike rentals over the year, we can see that there are big differences according to the periods.

Therefore, we will examine the number of bike rentals per month and per season and look at other factors that might influence bike rentals in Seoul.

Total of rented bike per day according to the daily mean temperature

The temperature graph shows that in general, the higher the temperature, the greater the demand for bicycle rentals.

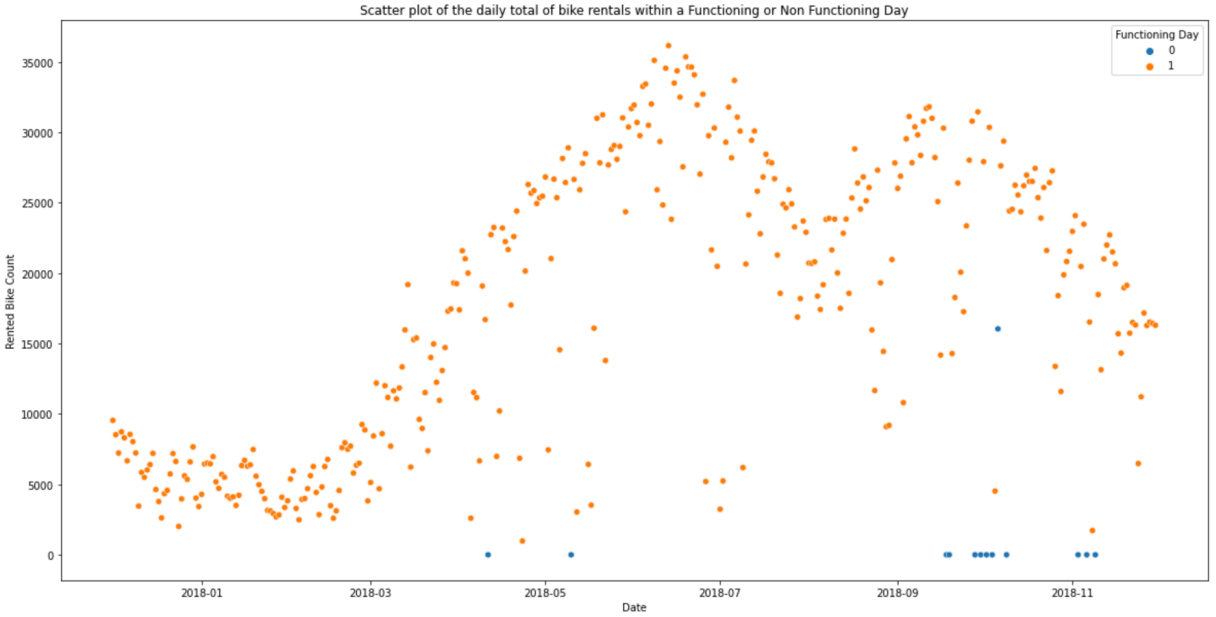
In fact, we can see that when the temperature is close to 20°C we have a high demand for bike rental : around 30000 rentals per day.

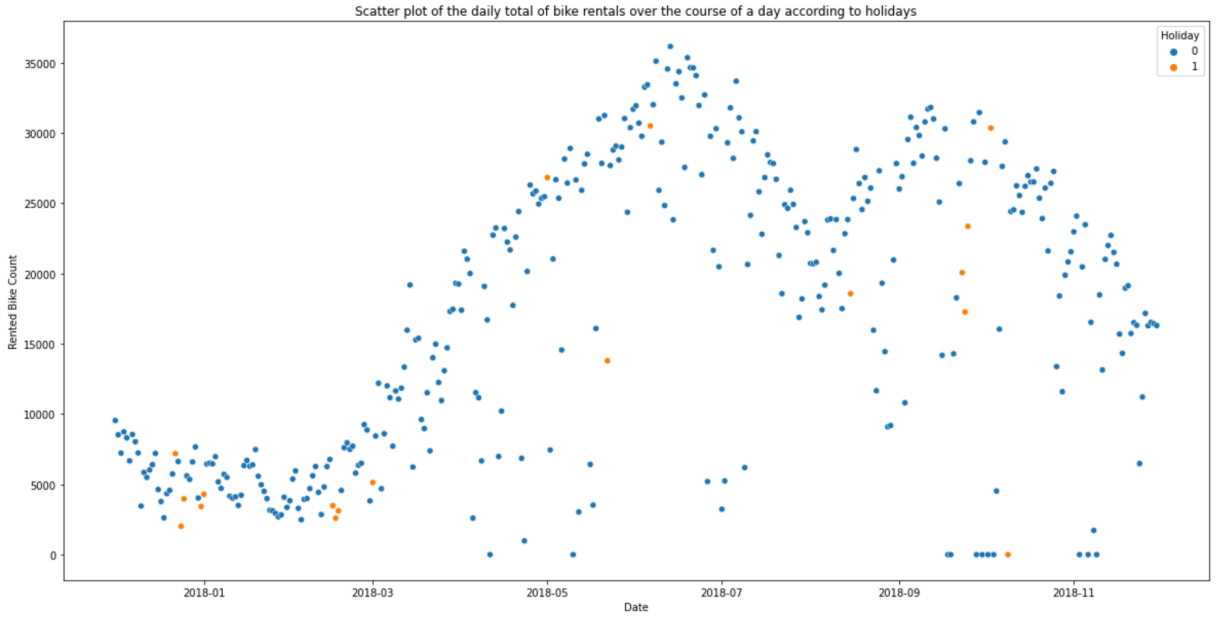


WHAT ABOUT HOLIDAY AND FUNCTIONING DAY ?

We can see that :

- In Non-Functional Days, people don't rent bikes.
- Holiday feature seems to not affect the daily mean of rentals. : It seems that the requests for bike rentals per day are about the same whether it is a vacation or not. However, it is important to note that due to the small sample size for vacations, the range of rentals is generally smaller than for non-vacation rentals.

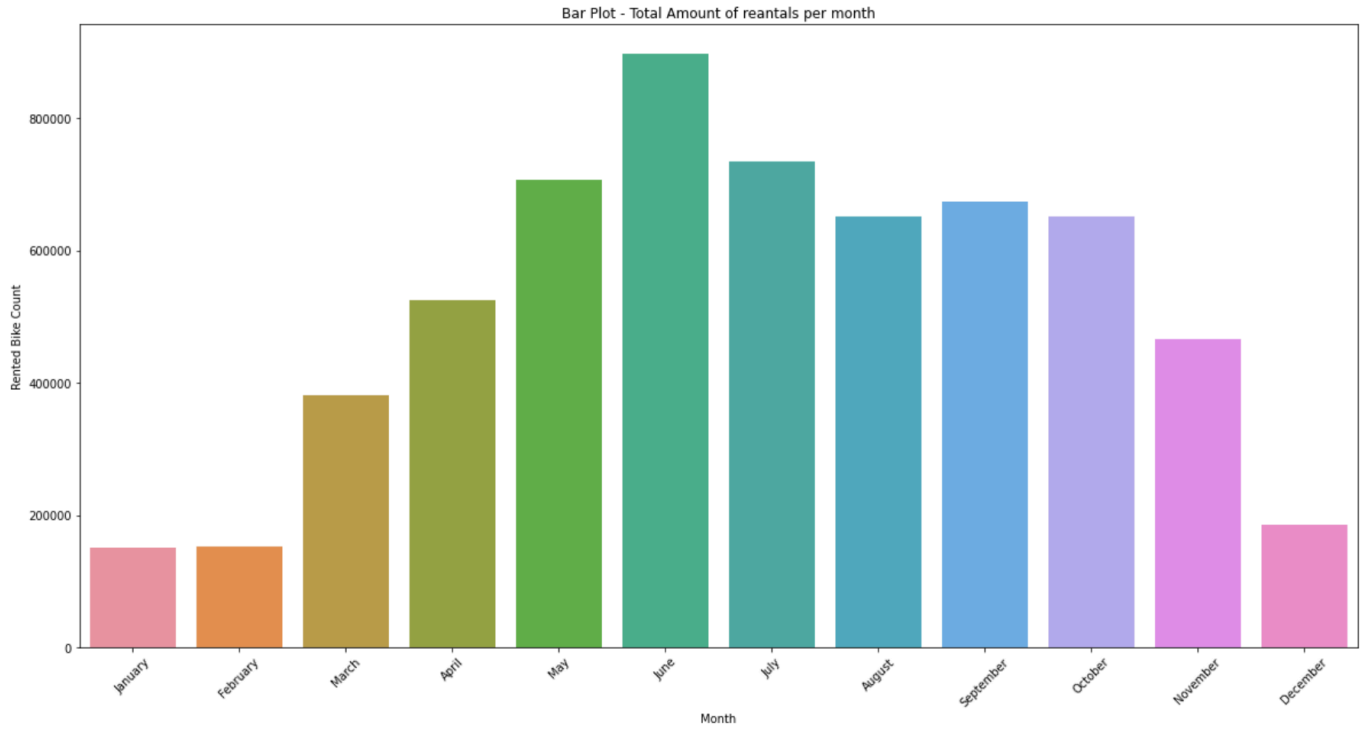




MONTHLY EVOLUTION OF BIKE RENTAL IN SEOUL

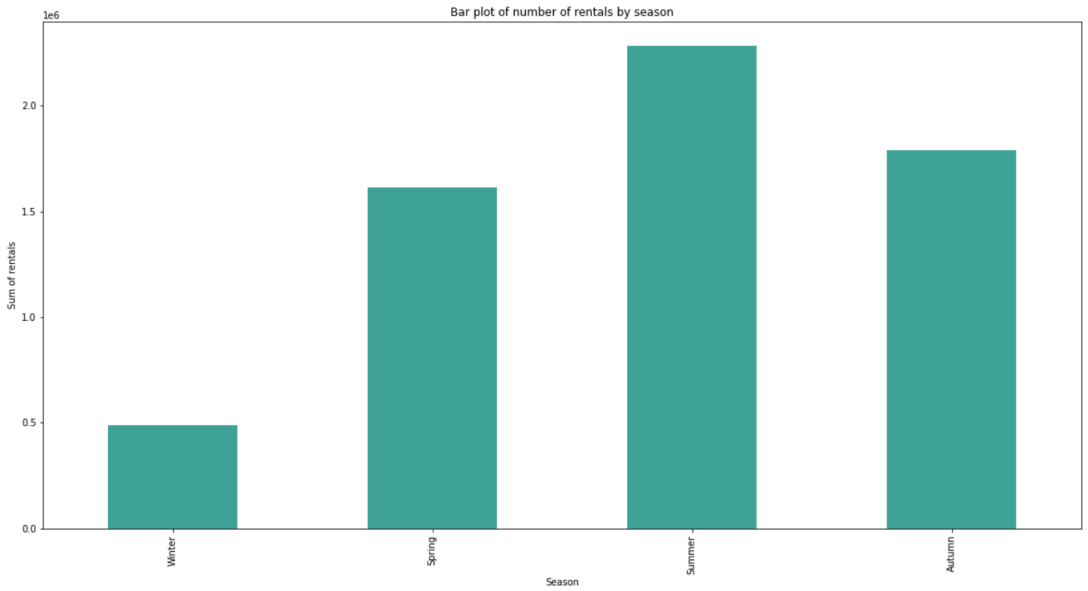
When we looked at the influence of the months on the demand for bike rentals, we realized that it does indeed exist. We can see on this diagram that during the months of December, January and February the demand is very low compared to that of June-July.

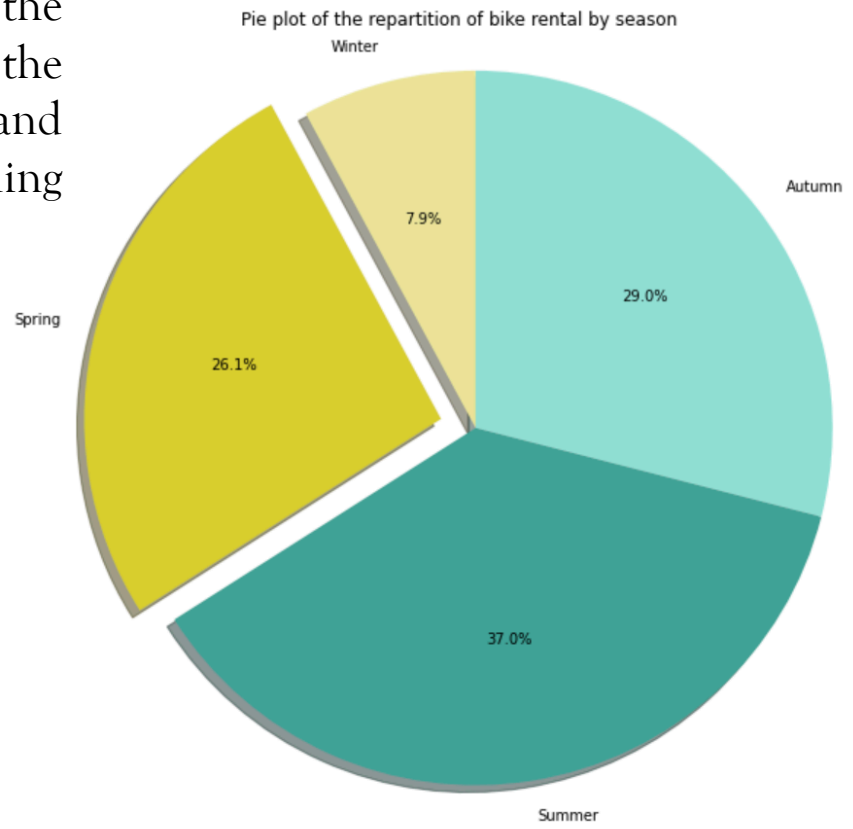
We can therefore assume that the seasons have an influence on the demand for bike rentals : They are more numerous in the summer period rather than in the winter period.



SEASONLY REPARTITION OF BIKE RENTAL IN SEOUL

The bar chart and pie chart of the different seasons in relation to the number of bike rentals reveals that there is a seasonal trend in the number of rentals. The number of rentals is generally low in winter and reaches its peak in summer. The season can be one of the determining factors affecting the number of bicycles rented.





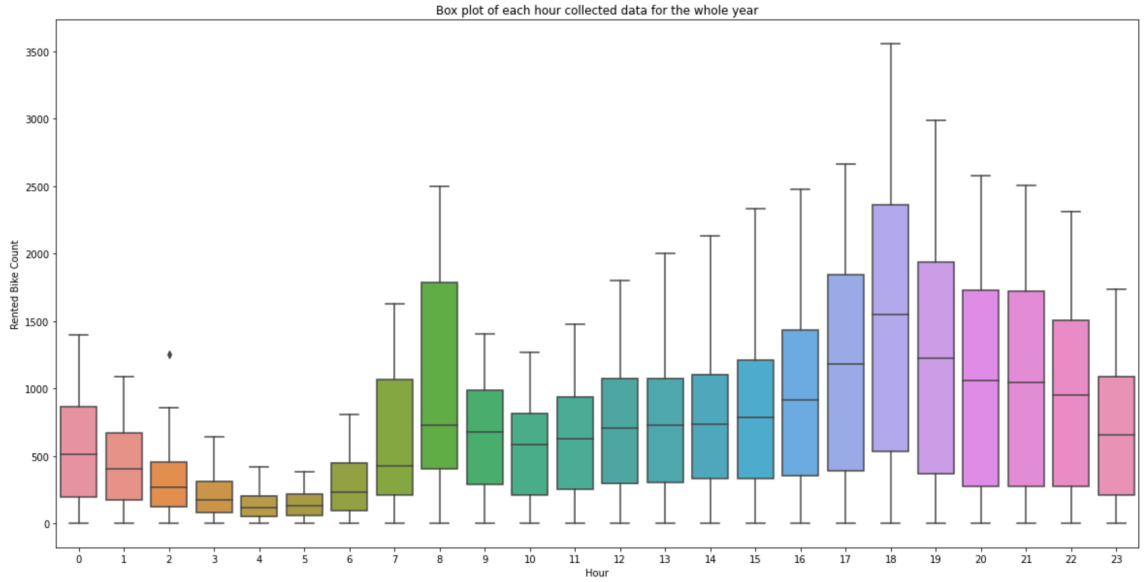
HOURLY EVOLUTION OF BIKE RENTAL IN SEOUL

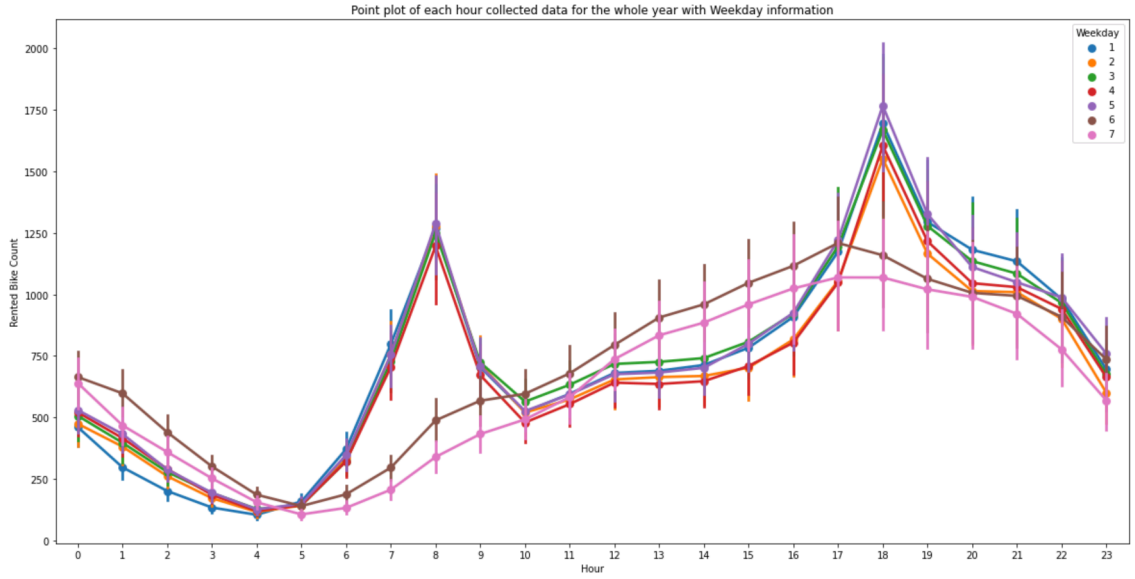
The line plot of hour of the day against bike rental count categorize by day of the week shows the difference of rental demand for weekday and weekend in different hours.

The peak demand during the week is between 6 and 9 a.m., the peak time for going to work and school, and similarly between 4 and 5 p.m. in the afternoon, perhaps because people have finished work and need transportation home.

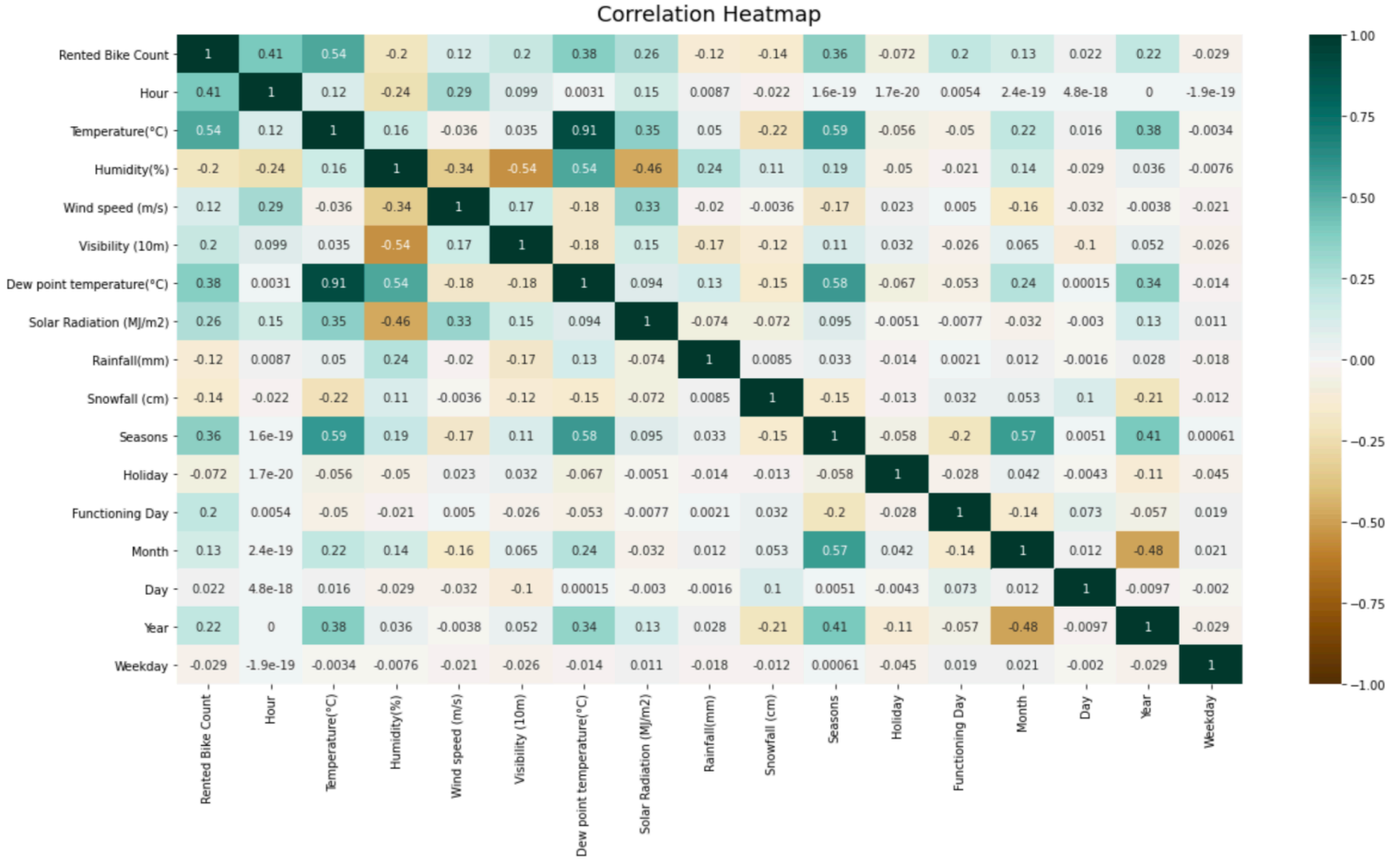
It can also be seen that the number of rented bicycles is declining during off-peak hours and at night.

From all these graphs, we can observe that there can be strong correlations between different variables. Let's take a closer look at this with a correlation matrix.





3.2 CORRELATION



In this matrix we can see that :

Temperature (°C) and *Hour* have the strongest correlation with the number of bicycles rented, all variables combined.

Temperature(°C) and *Dew point temperature*(°C) are very high correlated. To avoid redundant information, we may drop the second feature.

We can also see that there is a strong correlation between *Season* and *Month (and Temperature*(°C) , which is quite normal.

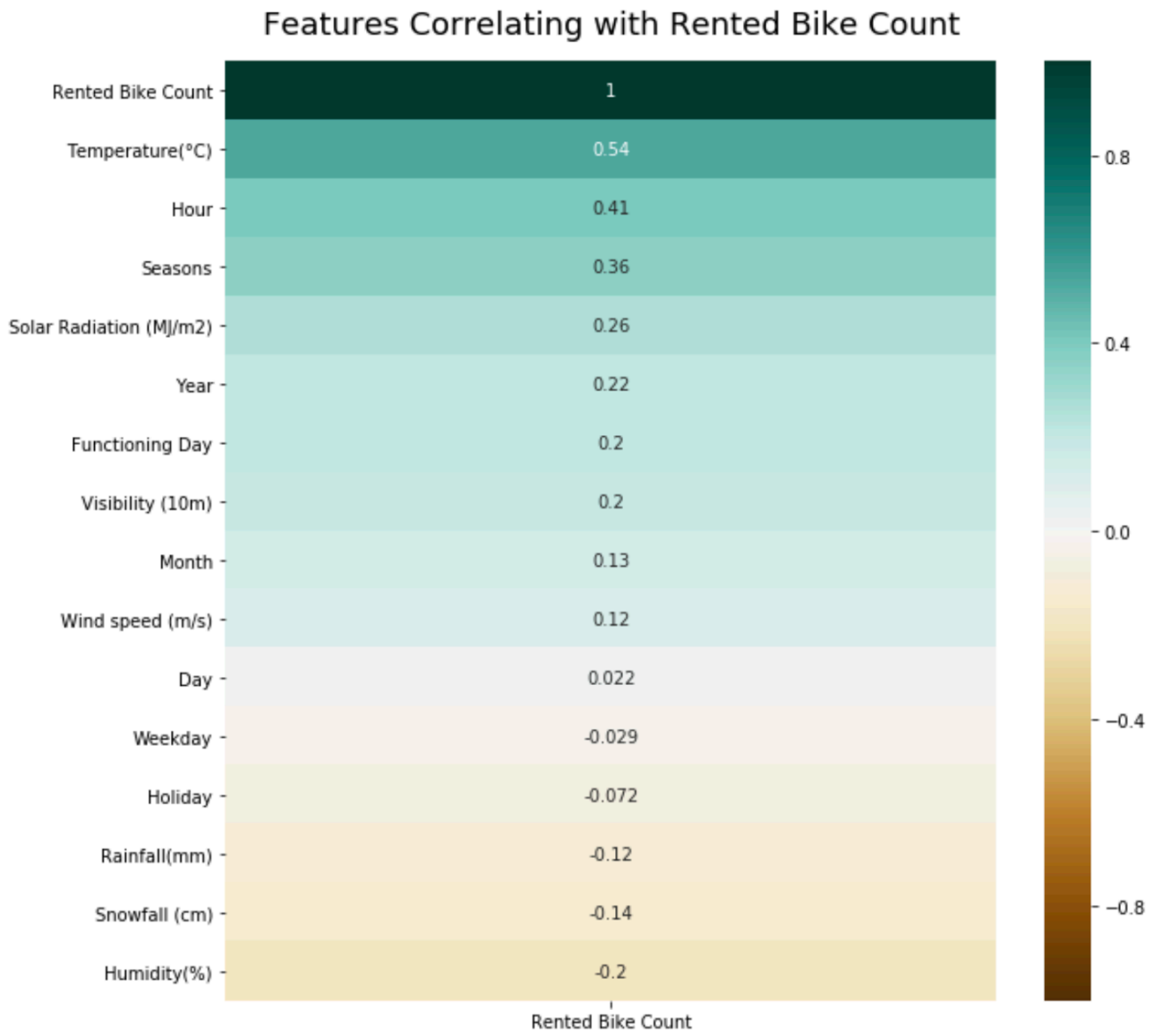
By preparing a correlation matrix, we can have a straighter forward view of what variables are strongly correlated and what is weakly correlated.

We were more precisely interested in the correlation between features and *Rented Bike Count* that we would like to predict.

Looking at this heatmap we notice that in fact the temperature is the feature most correlated with Rented count. These values indicate a moderate positive relationship between the variables. Here, the higher the temperature increases, the more bike rentals there are.

3.3 CONCLUSIONS ON EXPLORATORY DATA ANALYSIS

From the initial data exploration, we can clearly see that hour of the day and temperature are the strongest factors that determines the bike share rental demand.



4.MODEL BUILDING

4.1 DATA PRE-PROCESSING

Solar Radiation (MJ/m2), *Rainfall(mm)*, *Snowfall (cm)*, and *Visibility (10m)*, all give information about the weather. So maybe it's better if we transform it into categorical features in order to create a new feature named *Weather Condition*. The different conditions are :

- 1: Sunny, Few Clouds - no Rain neither Snow
- 2: Lot of Clouds - no Rain neither Snow
- 3: Rain or Snow (exclusive)
- 4: Both Rain & Snow

Before creating this column, we first converted these features into factor. For *Solar Radiation (MJ/m2)*, *Rainfall(mm)* and *Snowfall (cm)*, if the values are positive, we replace them by 1 otherwise by 0.

For Visibility(10m) if the value is less than or equal to 500, we replace it by 1, if it is between 501 and 1000, we replace the value by 2, if it is between 1001 and 1500, we replace it by 3, otherwise 4.

CREATION OF THE COLUMN "WEATHER CONDITION"

We have grouped the 4 features with the different possible combinations to be able to set up rules and create 4 categories :

1: Sunny, Few Clouds - no Rain neither Snow

2: Lot of Clouds - no Rain neither Snow

3: Rain or Snow (exclusive)

4: Both Rain & Snow

- if visibility = 4 or 3 and Rainfall and Snowfall = 0 the assigned value will be 1
- if visibility = 1 or 2 and Rainfall and Snowfall = 0 the assigned value will be 2
- if Rainfall =1 and Snowfall = 0 or Rainfall =0 and Snowfall = 1 the assigned value will be 3
- if Rainfall =1 and Snowfall = 1 the value will be 4

	Visibility (10m)	Solar Radiation (MJ/m2)	Rainfall(mm)	Snowfall (cm)
0	1	0	0	0
1	1	0	0	1
2	1	0	1	0
3	1	0	1	1
4	1	1	0	0
5	1	1	0	1
6	1	1	1	0

e.g. Here the value assigned for water condition in the first case (line 0) where :

- Visibility = 1
- Solar Radiation = 0
- Rainfall = 0
- Snowfall = 0,

the assigned value will be 1. For the fourth case (line 3), it will be a 4 because Rainfall =1 and Snowfall = 1

- Before starting building statistical models, we have to partition our dataset into two sets, training and testing. Training set will be used to train statistical models and estimate coefficients, while testing set will be used to validate the model we build with the training set.

- The resulting training set contains 5869 observations and testing set contains 2891 observations.

4.2 MODELLING

We want to predict a quantitative variable : ***Rented Bike Count***.

It is therefore a regression problem.

We tested several regression models using the Scikit-learn python library which provides many supervised and unsupervised learning algorithms. The features that Scikit-learn provides include Regression, including linear and logistic regression.

Metric used : R-square

Linear Regression
Lasso
Random Forest
Extra Trees
XGBoost

LINEAR REGRESSION

- To test the learning of our dataset we started with a linear regression model.
- We have chosen the R-square indicator which allows us to determine the extent of the linear relation of a value with respect to time.

Here is the result :

```
Score for Linear Regression Model :  0.5726484213215801

Duration time : 0.01684260368347168
```

LASSO

The hyperparameters we tuned are max_iter, alpha and selection.

Best score : 0.56945

With parameters :
Max_iter = 1000
Alpha = 0.02
Selection = cyclic

Duration Time 9.51 s

param_alpha	param_max_iter	param_selection: cyclic	param_selection: random
0.001	500	0.569450	0.569450
	1000	0.569450	0.569450
0.01	500	0.569451	0.569451
	1000	0.569451	0.569451
0.02	500	0.569452	0.569451
	1000	0.569452	0.569453
0.025	500	0.569452	0.569452
	1000	0.569452	0.569452
0.05	500	0.569452	0.569452
	1000	0.569452	0.569452
0.1	500	0.569448	0.569448
	1000	0.569448	0.569448
0.25	500	0.569395	0.569395
	1000	0.569395	0.569394
0.5	500	0.569164	0.569161
	1000	0.569164	0.569160
0.8	500	0.568665	0.568655
	1000	0.568665	0.568700
1.0	500	0.568206	0.568197
	1000	0.568206	0.568230

RANDOM FOREST

The hyperparameters we tuned are n_estimators, min_samples_leaf, max_depth,min_samples_split and bootstrap

Best score : 0.920789

With parameters :
n_estimators : 200
min_samples_leaf : 1
max_depth : 80
min_samples_split : 2
bootstrap : True

Duration Time : ~ 1 h 06

	param_min_samples_leaf	1	3	7
param_n_estimators	param_max_depth			
10	60	0.883599	0.882695	0.876968
	70	0.884498	0.882417	0.876608
	80	0.884044	0.882179	0.876464
	90	0.883730	0.881604	0.875700
	100	0.884503	0.882944	0.876751
100	60	0.889140	0.886421	0.879910
	70	0.889155	0.886381	0.879744
	80	0.889322	0.886587	0.879829
	90	0.888922	0.886295	0.879809
	100	0.889310	0.886140	0.879883
200	60	0.889298	0.886419	0.880149
	70	0.889471	0.886420	0.880262
	80	0.889577	0.886444	0.880063
	90	0.889531	0.886302	0.880045
	100	0.889437	0.886578	0.880166

* We made the pivot table only with 3 parameters, that's why the display does not appear on it.

EXTRA TREE

The hyperparameters we tuned are : n_estimators, min_samples_leaf, max_depth,min_samples_split and bootstrap

Best score : 0.931053

With parameters :
n_estimators : 200
min_samples_leaf : 1
max_depth : 100
min_samples_split : 4
bootstrap : False

Duration Time : ~ 25 min

param_n_estimators	param_max_depth	param_min_samples_leaf 1	3	7
10	60	0.914942	0.907380	0.889291
	70	0.916949	0.907799	0.890026
	80	0.914988	0.909559	0.888903
	90	0.914943	0.908726	0.890137
	100	0.913265	0.908486	0.888962
100	60	0.925222	0.916950	0.897349
	70	0.925208	0.916953	0.896875
	80	0.925145	0.917212	0.896957
	90	0.925195	0.916945	0.896748
	100	0.925101	0.917115	0.897007
200	60	0.925639	0.917793	0.897198
	70	0.925810	0.917669	0.897623
	80	0.925917	0.917613	0.897236
	90	0.925687	0.917819	0.897584
	100	0.925914	0.917676	0.897310

* We made the pivot table only with 3 parameters, that's why the display does not appear on it.

XGBOOST

The hyperparameters we tuned are : max_depth and gamma.

Best score : 0.94078

With parameters :
max_depth : 10
gamma : 1.5

Duration Time : ~ 6 min 40

param_gamma	0.5	1.0	1.1	1.2	1.5
param_max_depth					
1	0.638603	0.638603	0.638603	0.638603	0.638603
5	0.921823	0.921823	0.921823	0.921823	0.921823
10	0.940473	0.940206	0.940087	0.940208	0.940780
60	0.932155	0.932199	0.932159	0.932173	0.932102
70	0.932155	0.932199	0.932159	0.932173	0.932102
80	0.932155	0.932199	0.932159	0.932173	0.932102
90	0.932155	0.932199	0.932159	0.932173	0.932102
100	0.932155	0.932199	0.932159	0.932173	0.932102

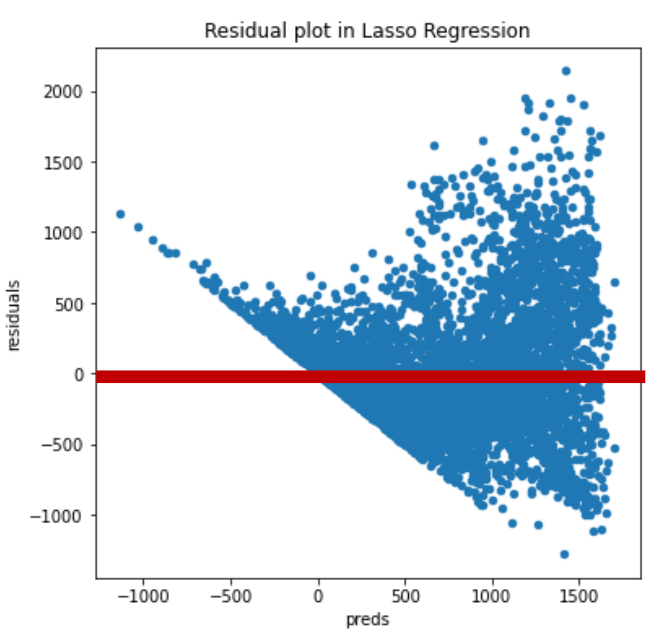
EXAMINING PREDICTED VS. RESIDUAL ("THE RESIDUAL PLOT")

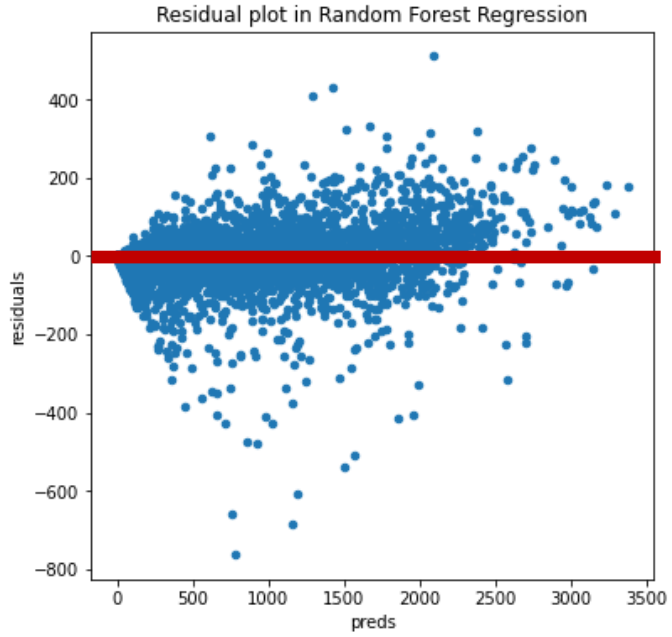
In these plot, each point is one hour, where the prediction made by the model is on the x-axis and the residual is on the y-axis. The distance from the line at 0 is how bad the prediction was for that value.

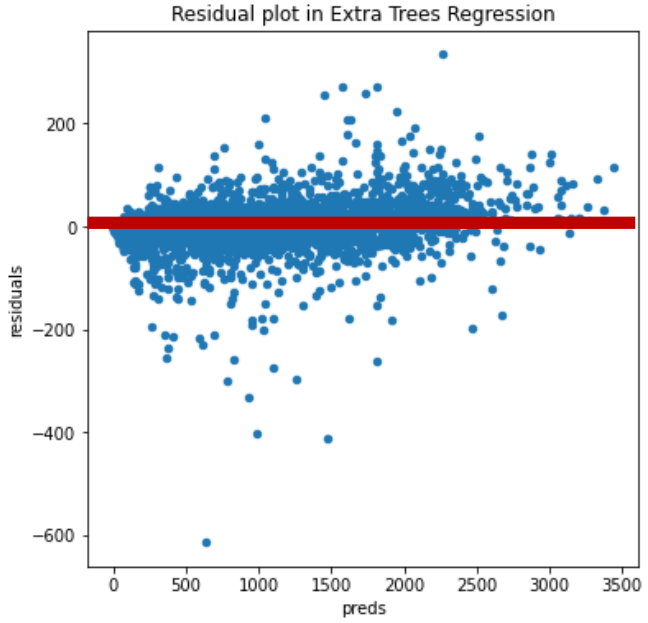
Residual = Observed – Predicted

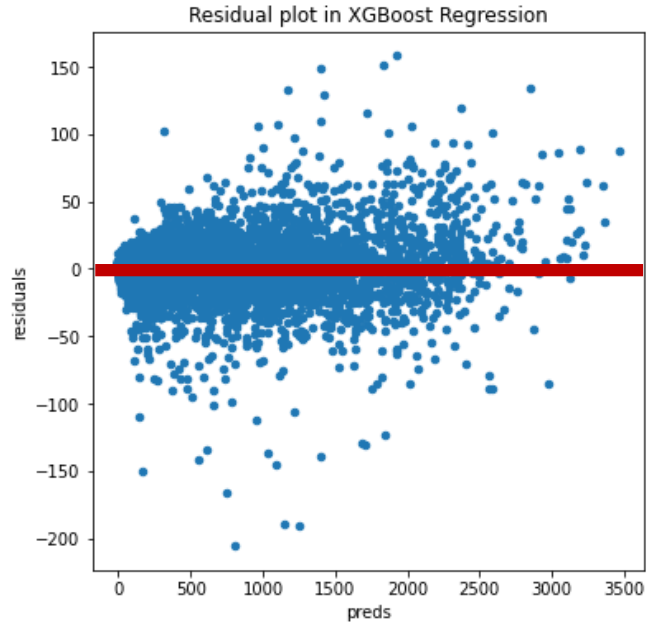
Positive values for the residual (on the y-axis) mean the prediction was too low, and negative values mean the prediction was too high; 0 means the guess was exactly correct.

So we can see that the best model correspond to XGBoost Regression.









FINAL RESULT

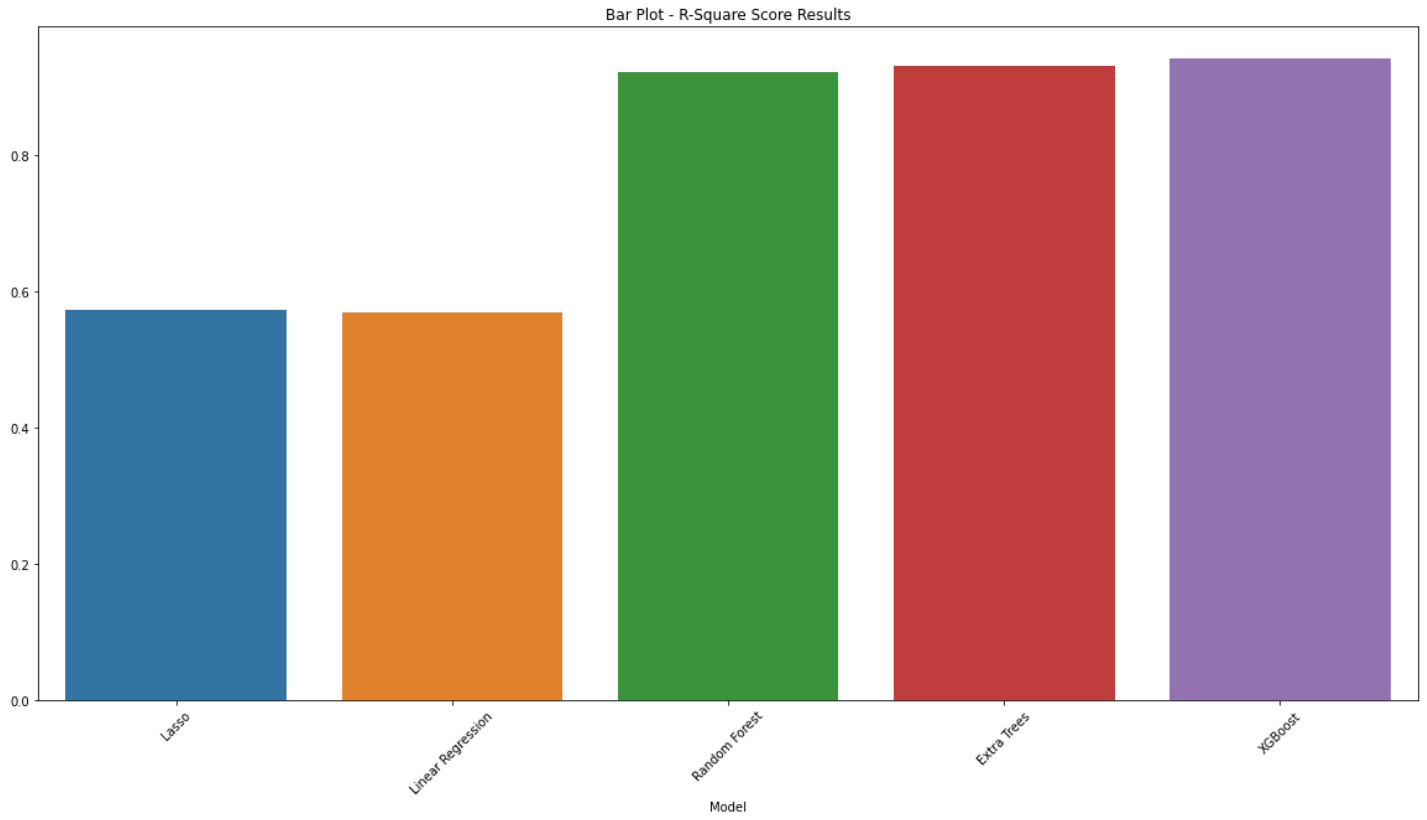
By carrying out a standardisation we obtained less good results. So we keep our models without standardization.

Result with standarization

	Model	Score
0	Linear Regression	0.572648
1	Lasso	0.569706
2	Random Forest	0.916112
3	Extra Trees	0.927239
4	XGBoost	0.928145

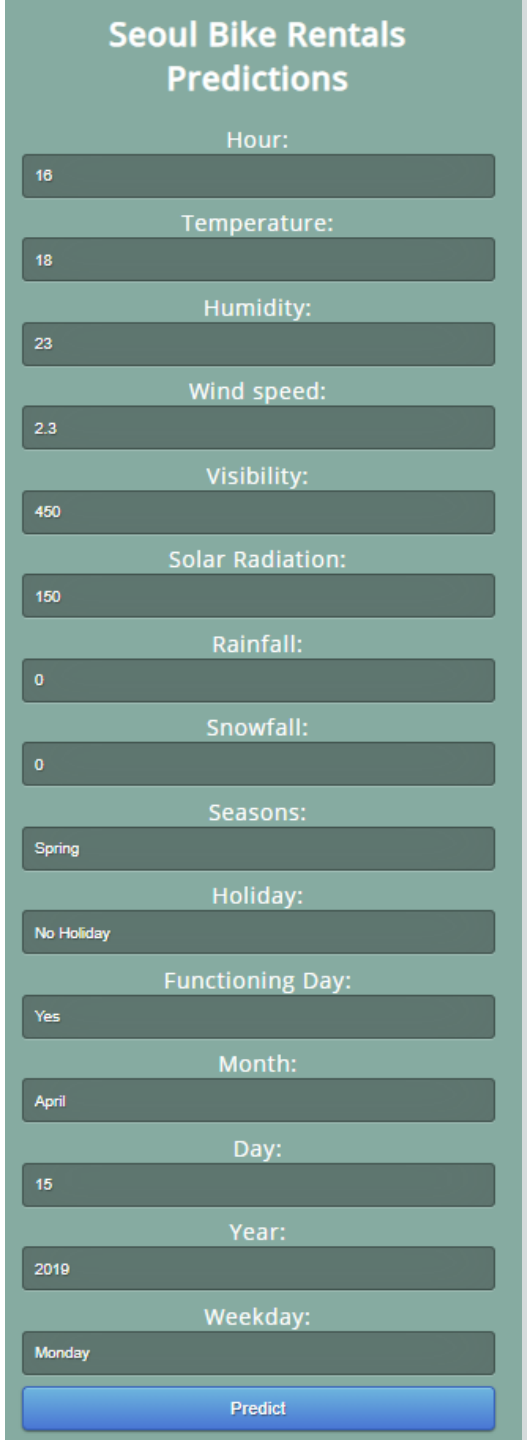
Result without standarization

	Model	Score
1	Lasso	0.569453
0	Linear Regression	0.572648
2	Random Forest	0.920790
3	Extra Trees	0.931053
4	XGBoost	0.940780



5.API

- An API (Application Programming Interface) is a set of functions that allows applications to access data and interact with external software components, operating systems, or microservices.
- We will create an API to allow the customer to predict the number of bike rentals per hour in Seoul through an interface.
- To turn the notebook model into an API, we decided to use the Pickle module, which allows us to transform it into a file that can then be easily reused on the interface to predict new values.
- The API will automatically fit the inputs of the user with the pickle model (Categorization of climate data and creation of *Weather Condition* feature).

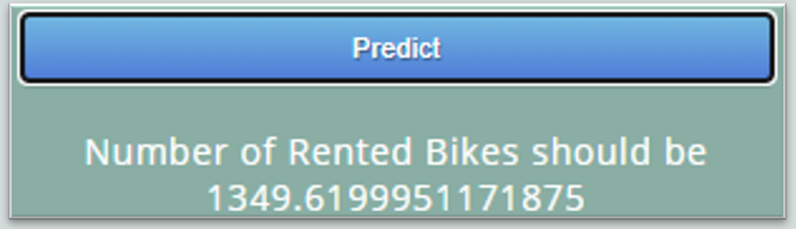


The interface is easy to use. The customer has to fill in each of the fields and click on the "Predict" button, so at the bottom of the page the number of bike rentals in Seoul, for the time and date filled in, will be given.

Notebook prediction :

```
[122] best.predict(np.array([[16,18,23,2.3,1,1,0,0,2,0,1,4,15,2019,1,1]]))[0]

      1366.8906
```



6.CONCLUSION

서울자전거
SEOUL BIKE 따릉이

Through the exploration of this dataset and the analysis performed on it we discovered that time of day and temperature are the two most important factors that determine the demand for bicycle rental in Seoul.

Using Xgboost , a well advanced scientific tool, we were able to predict the number of bikes with a relatively high accuracy. We could try several other models to perhaps build better statistical models that more accurately explain the variations caused by different variables.