ESILV - Python for data analysis - devoir 2021

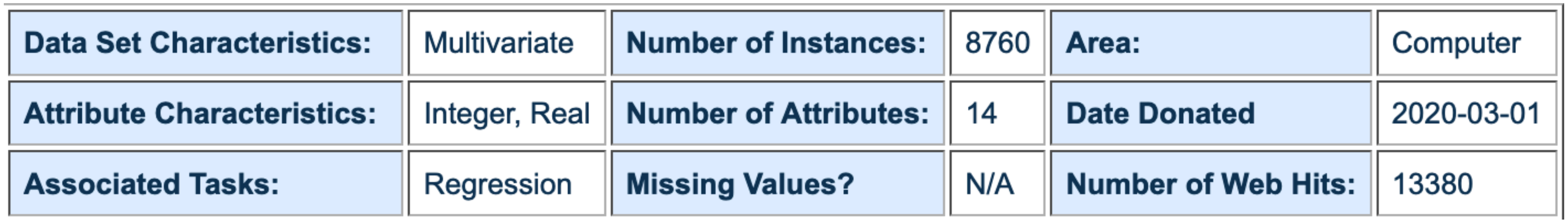
Submitted by
Kanika Mahajan & Yuning LI

Outline

1. **Dataset basic introduction**
2. **Data visualization**: visualize the basic information in the original dataset to have a better understanding, including the correlationship, changement, etc.
3. **Data preparation for modeling**: In order to do modelization, we should firstly prepare the data, including the missing value exclusion, changing data type, dropping the incorrect data, etc.
4. **Modelization and validation**: try different regression models to predict the result, compare them and select the best algorithm, then optimize its parameter to get the best model.

1-Dataset basic introduction

- **Name**: Seoul Bike Sharing Demand Data Set
- **Org**:https://archive.ics.uci.edu/ml/datasets/Seoul+Bike+Sharing+Demand?fbclid=IwAR0kA9lVpTHUikp5xuQKmz9VVeHXeTDkNyON3PUMLqKE6UWB4iReOBS4fP0
- **Abstract**: The dataset contains count of public bikes rented at each hour in Seoul Bike haring System with the corresponding Weather data and Holidays information.



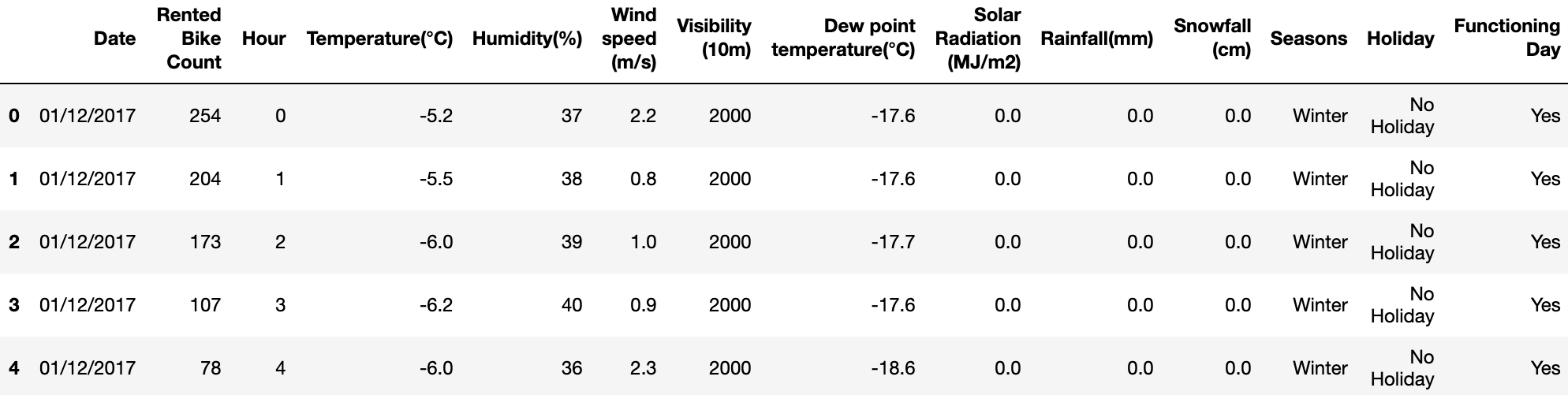
Data Set Characteristics:	Multivariate	**Number of Instances:**	8760	**Area:**	Computer
Attribute Characteristics:	Integer, Real	**Number of Attributes:**	14	**Date Donated**	2020-03-01
Associated Tasks:	Regression	**Missing Values?**	N/A	**Number of Web Hits:**	13380

1-Dataset basic introduction

- **Dataset Information**: Currently Rental bikes are introduced in many urban cities for the enhancement of mobility comfort. It is important to make the rental bike available and accessible to the public at the right time as it lessens the waiting time. Eventually, providing the city with a stable supply of rental bikes becomes a major concern. The crucial part is the prediction of bike count required at each hour for the stable supply of rental bikes.
  The dataset contains weather information (Temperature, Humidity, Windspeed, Visibility, Dewpoint, Solar radiation, Snowfall, Rainfall), the number of bikes rented per hour and date information.
- **Attribute Information**:
  Date : year-month-day
  Rented Bike count - Count of bikes rented at each hour
  Hour - Hour of the 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)

2-Data visualization

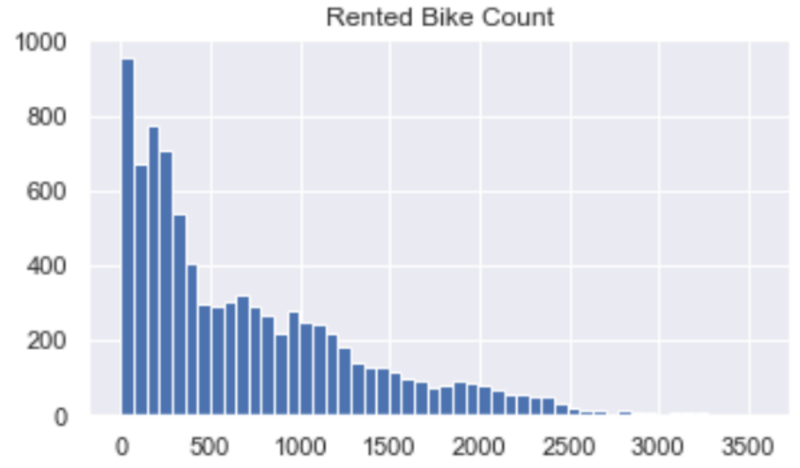
- **Basic information**: we have the dataset of 8760 items * 14 attributions. Our aim is to make prediction of bike count required at each hour for the stable supply of rental bikes.

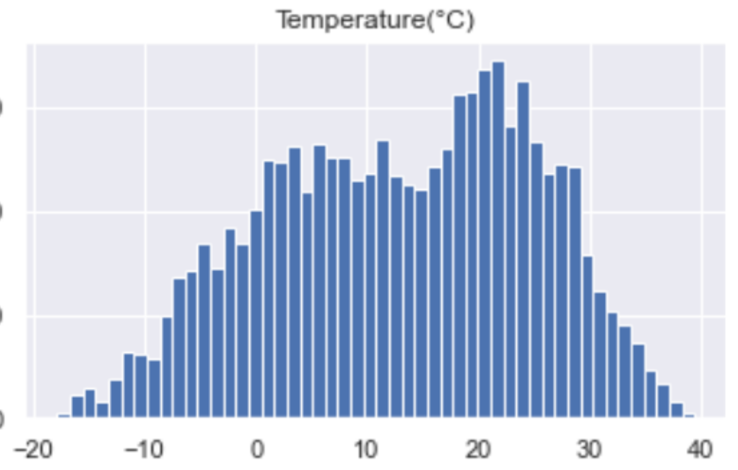


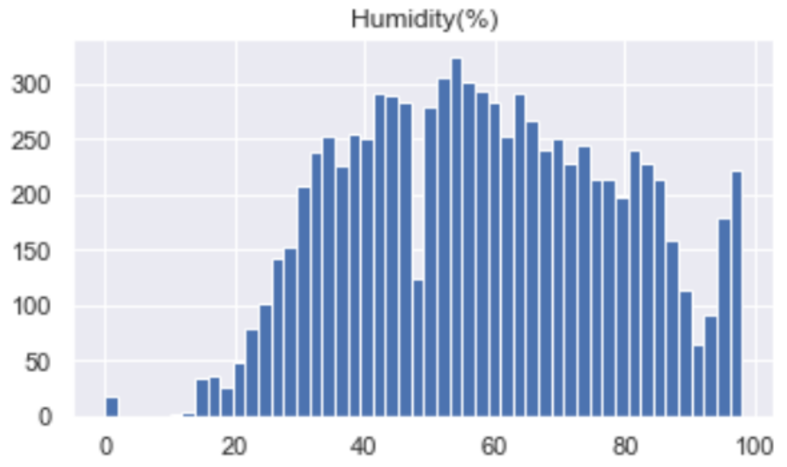
	Date	Rented Bike Count	Hour	Temperature(°C)	Humidity(%)	Wind speed (m/s)	Visibility (10m)	Dew point temperature(°C)	Solar Radiation (MJ/m2)	Rainfall(mm)	Snowfall (cm)	Seasons	Holiday	Functioning Day
0	01/12/2017	254	0	-5.2	37	2.2	2000	-17.6	0.0	0.0	0.0	Winter	No Holiday	Yes
1	01/12/2017	204	1	-5.5	38	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

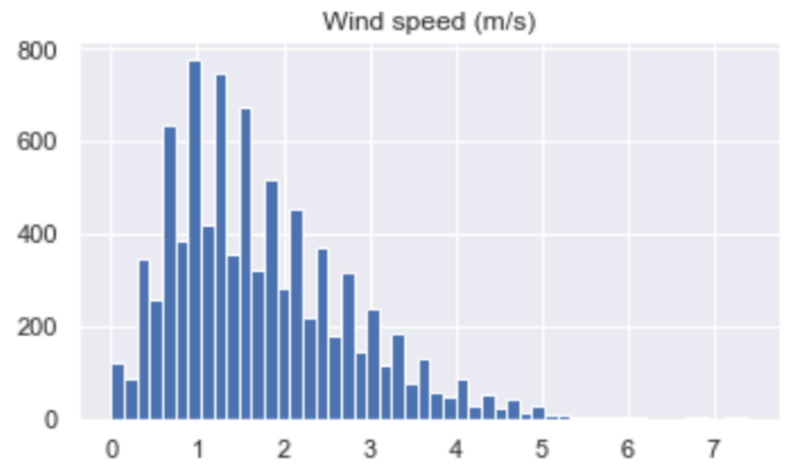
2-Data visualization

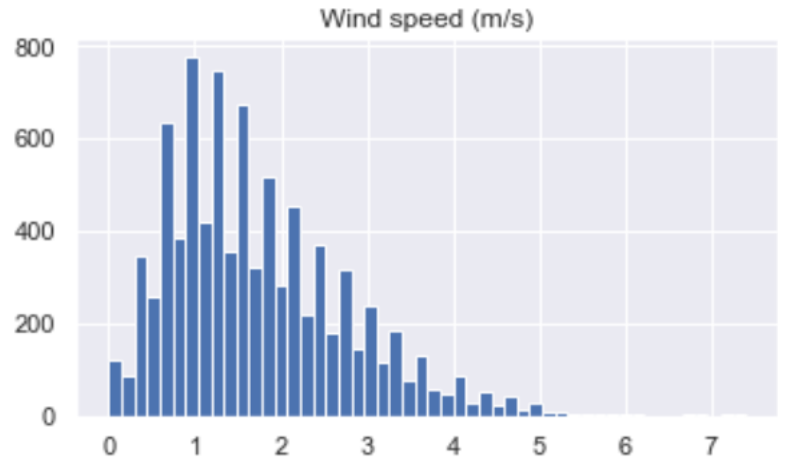
- **Basic attribution visualization- Histogram** :

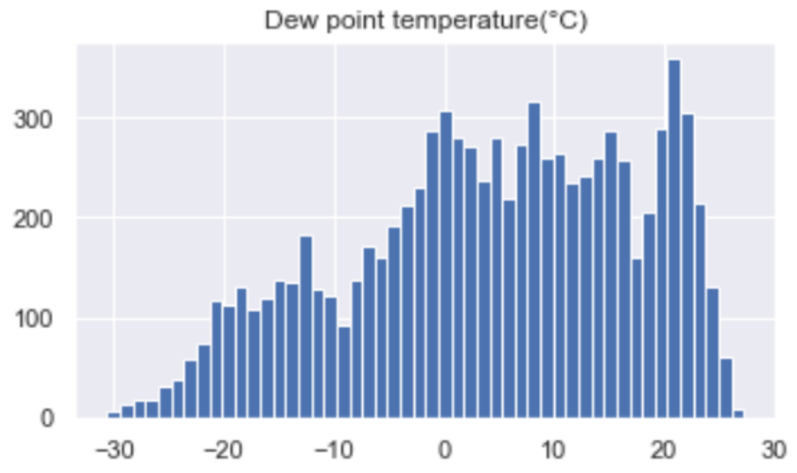








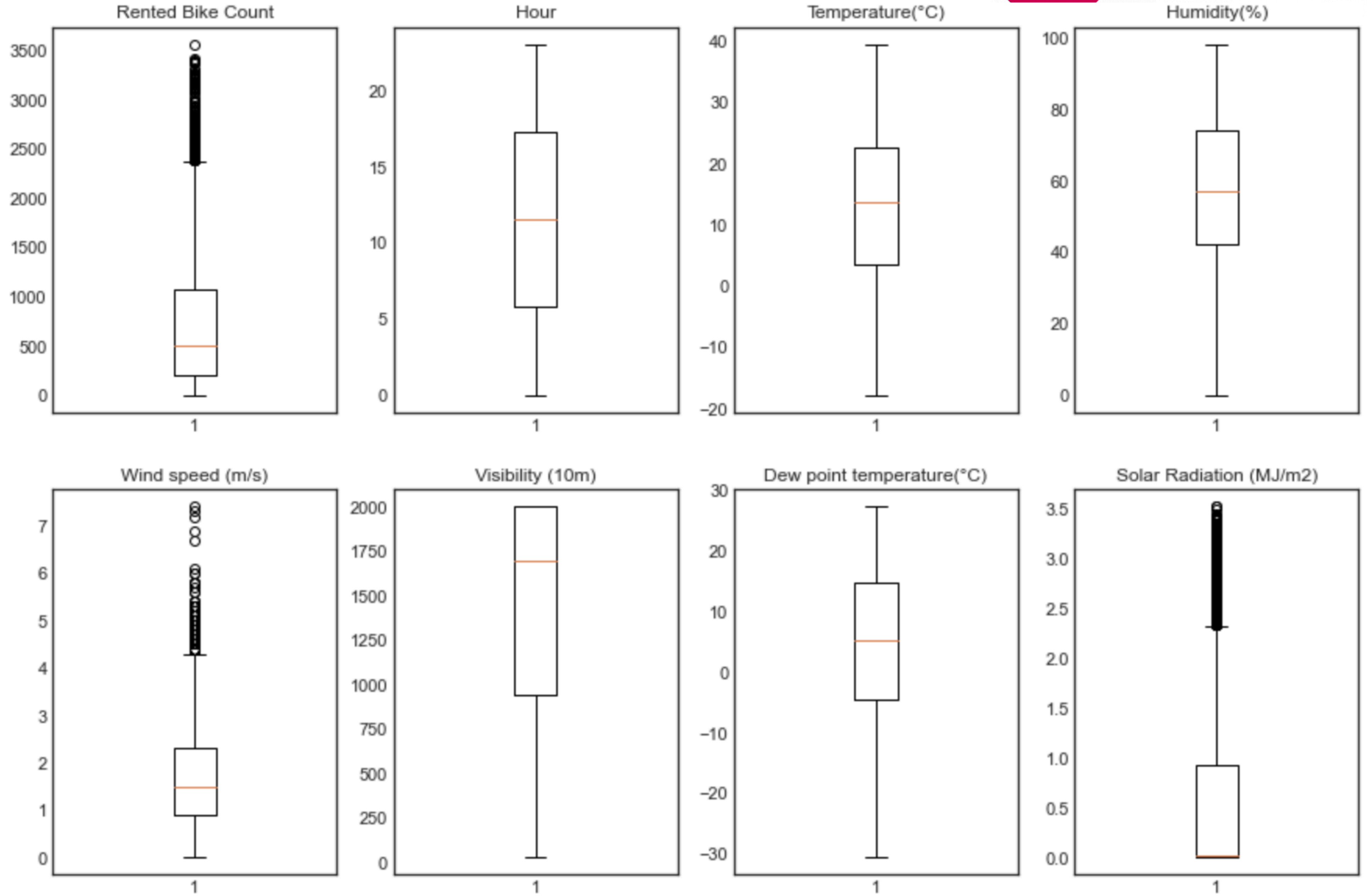




2-Data visualization

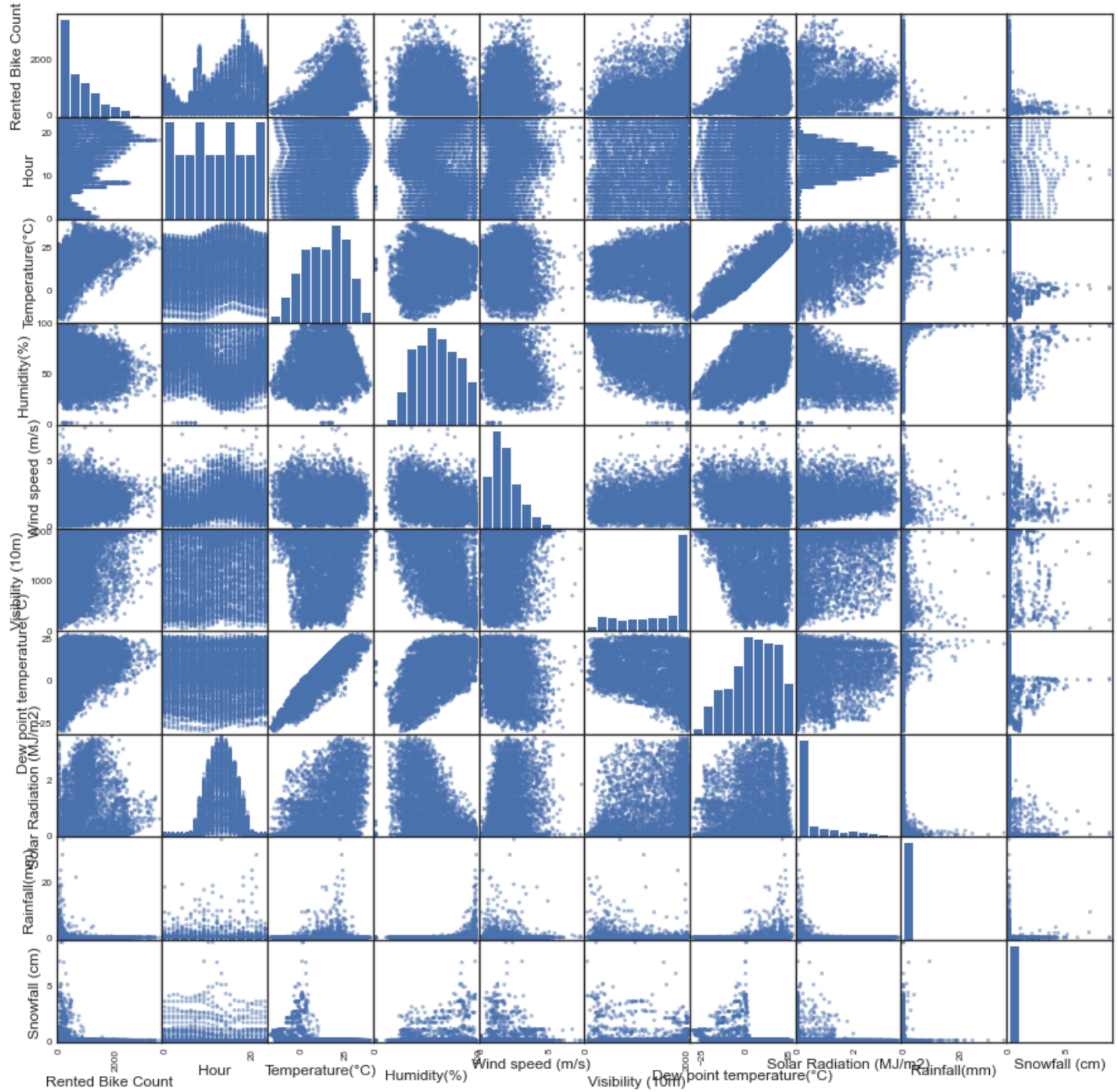
- **Visualization**

-Boxgram :



2-Data visualization

- **Pairwise comparison between features**

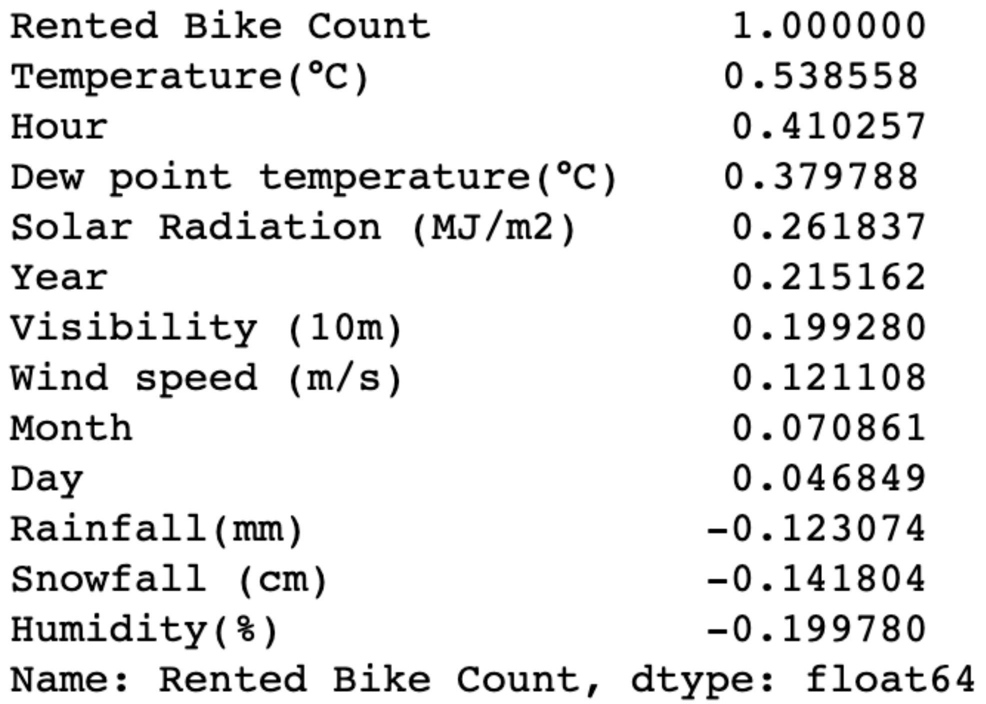


2-Data visualization

- **Show the correlations between aim attribution and others**:

-The 'Temperature' 'Hour' 'Dew point temperature' are the three most positive attribution correlated to 'Rented Bike Count', scored '0.53, 0.41, 0.37'.

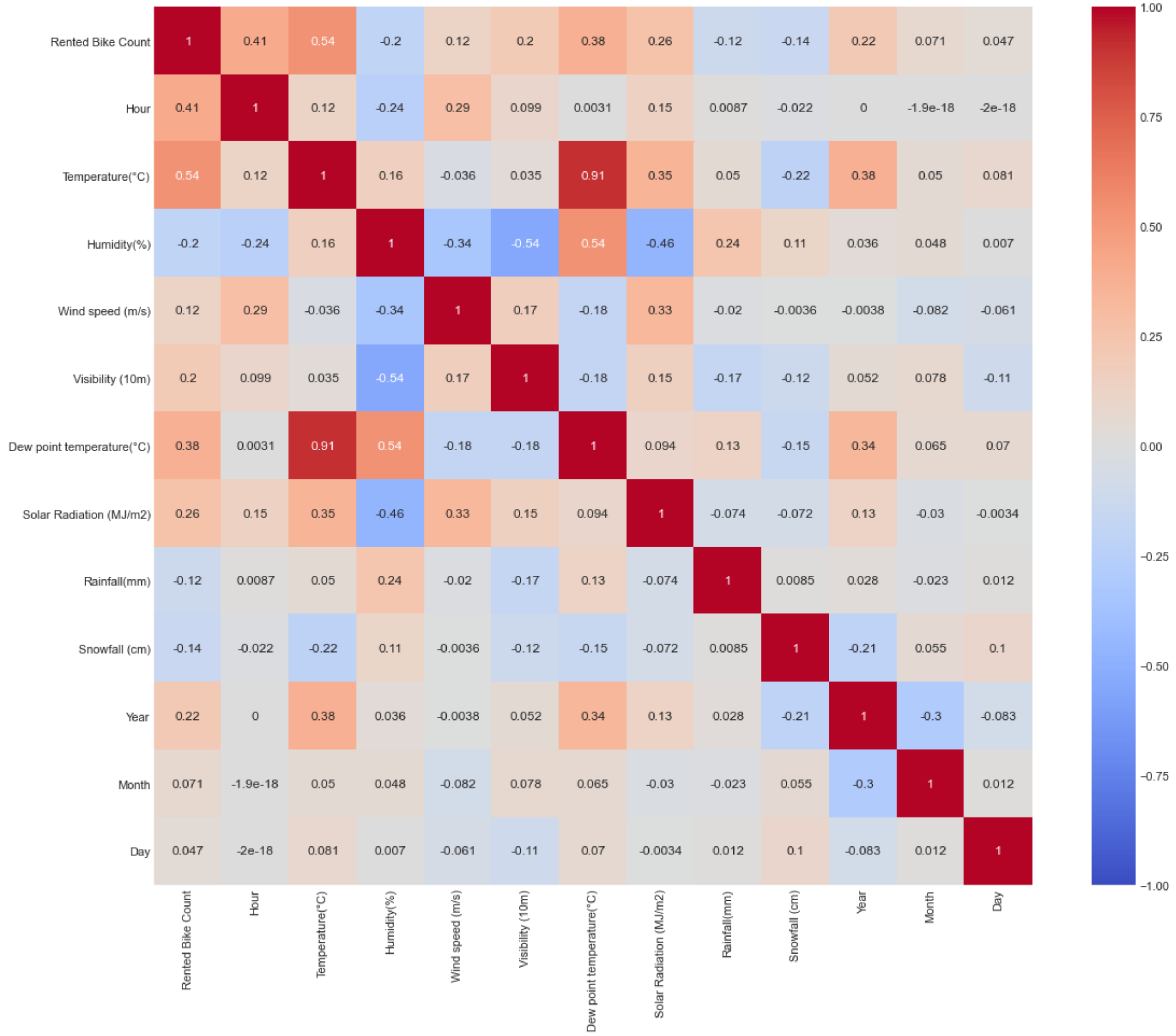
-The 'Rainfall' ' Snowfall' ' Humidity' are the three negative attribution correlated to ' Rented Bike Count', scored' -0.12, -0.14, -0.19'



```
Rented Bike Count             1.000000
Temperature(°C)               0.538558
Hour                          0.410257
Dew point temperature(°C)     0.379788
Solar Radiation (MJ/m2)       0.261837
Year                          0.215162
Visibility (10m)              0.199280
Wind speed (m/s)              0.121108
Month                         0.070861
Day                           0.046849
Rainfall(mm)                 -0.123074
Snowfall (cm)                -0.141804
Humidity(%)                  -0.199780
Name: Rented Bike Count, dtype: float64
```

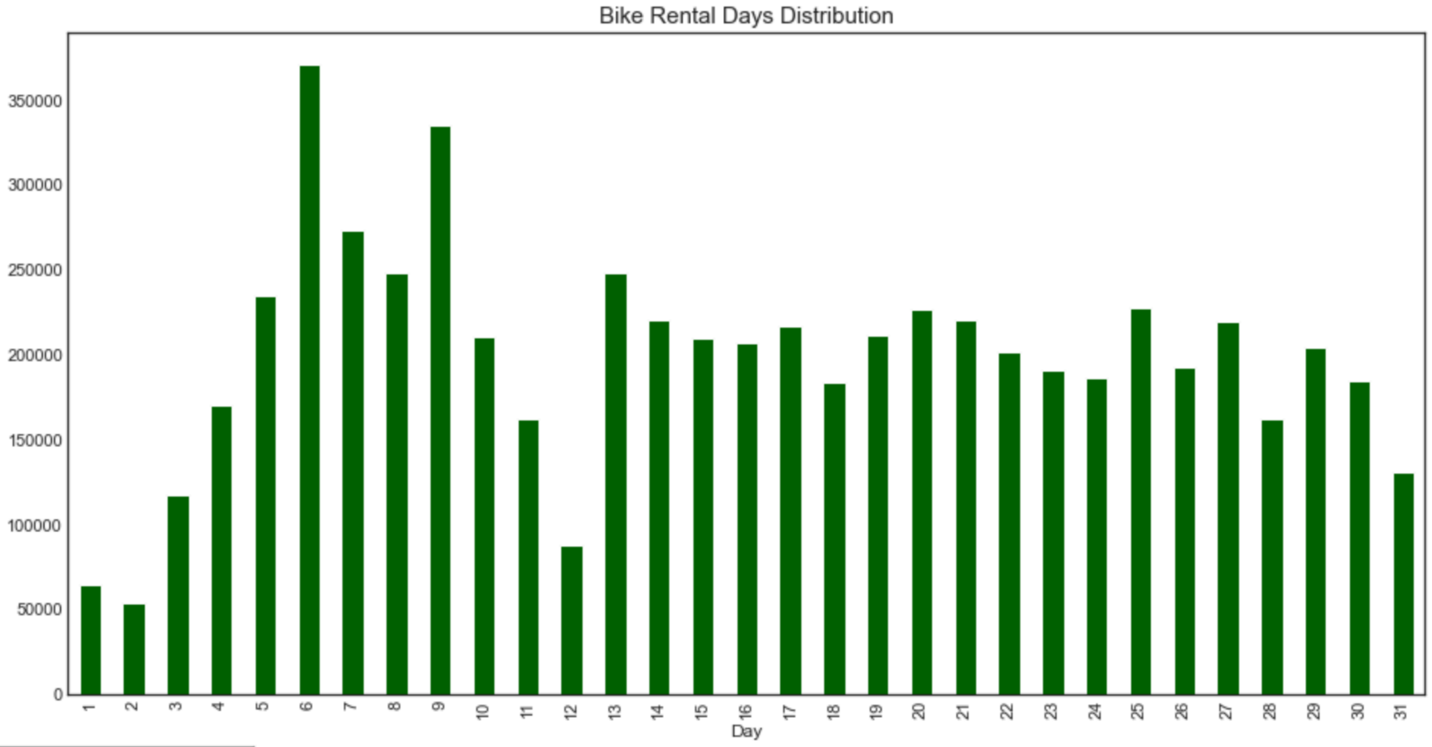
2-Data visualization

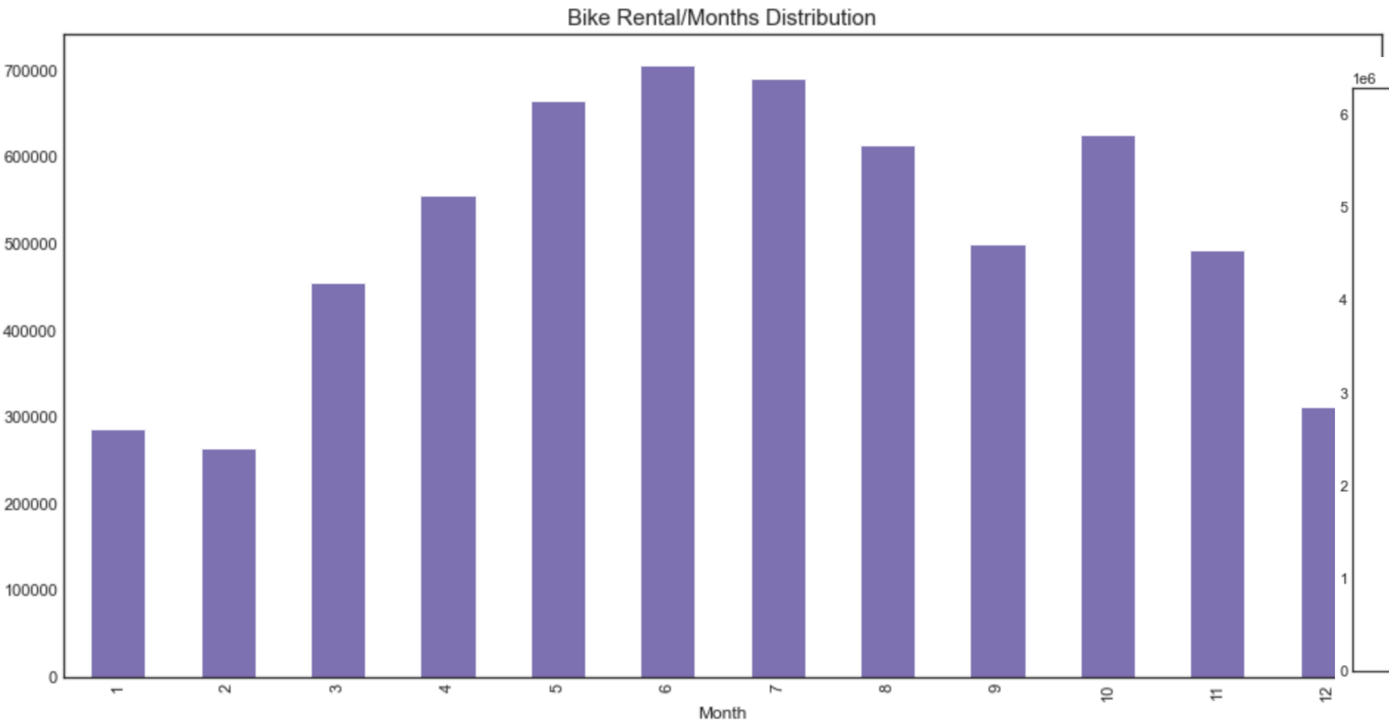
- **Show the correlation between each other attribution with heatmap**

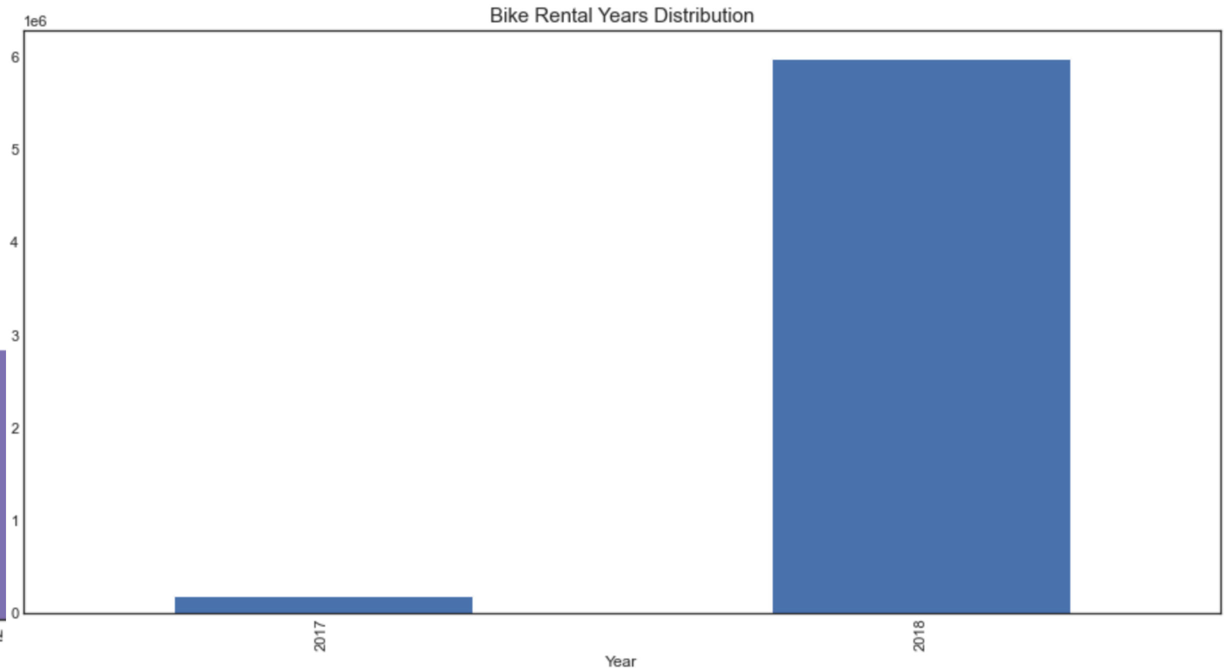


2-Data visualization

- **Show the rental data change by day/month/year.**

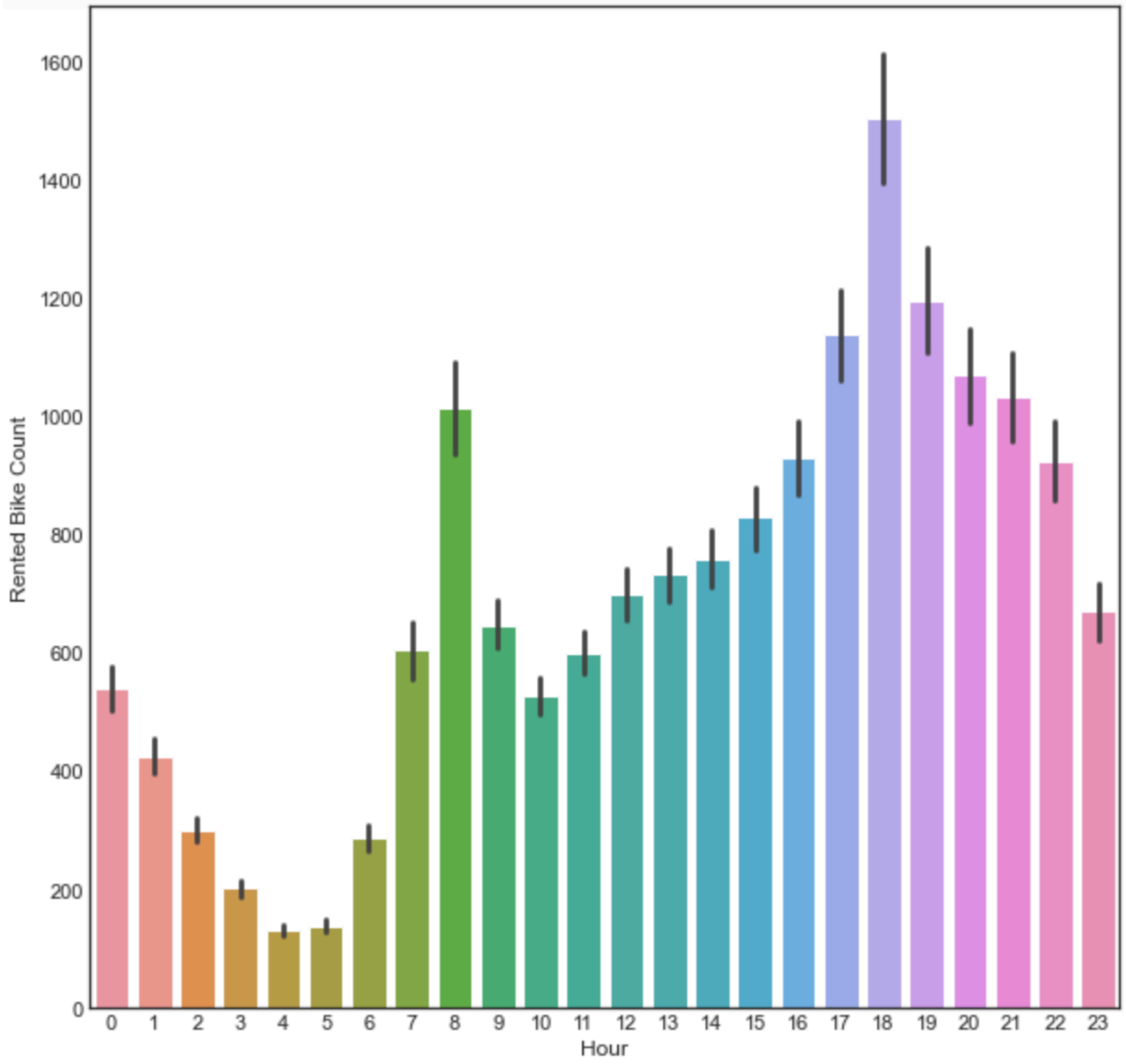






2-Data visualization

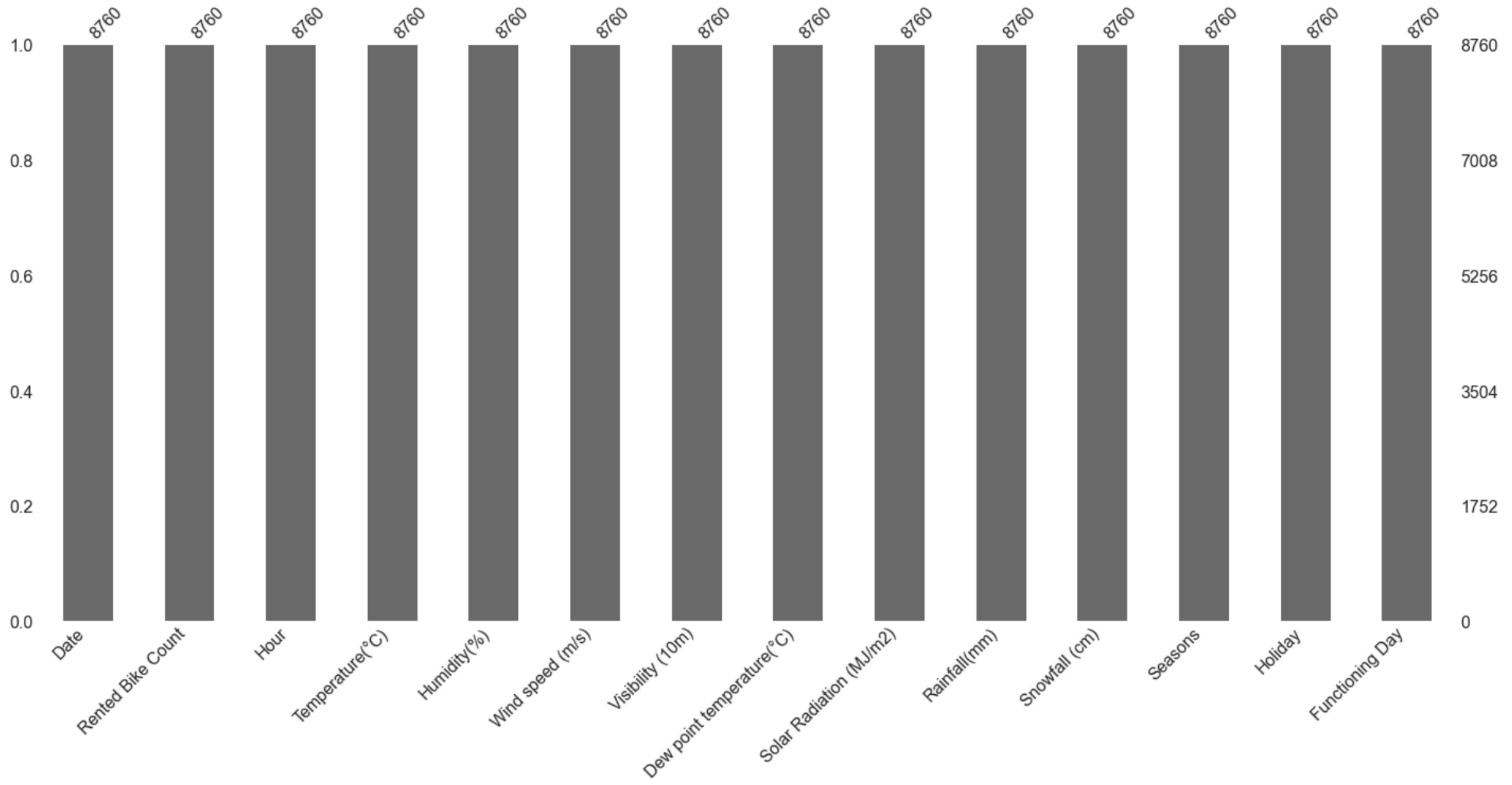
- **Show the rental data change by hour**: bike renting happen mostly in the mornings and evenings.



3-Data preparation for modeling

- **Check the missing values**: we can see the number of the Null value is 0, and all the attributions has 8760 items(same with the dataset), so our dataset is good with no missing values.

```
Date                          0
Rented Bike Count             0
Hour                          0
Temperature(°C)              0
Humidity(%)                   0
Wind speed (m/s)              0
Visibility (10m)              0
Dew point temperature(°C)    0
Solar Radiation (MJ/m2)       0
Rainfall(mm)                  0
Snowfall (cm)                 0
Seasons                       0
Holiday                       0
Functioning Day               0
dtype: int64
```

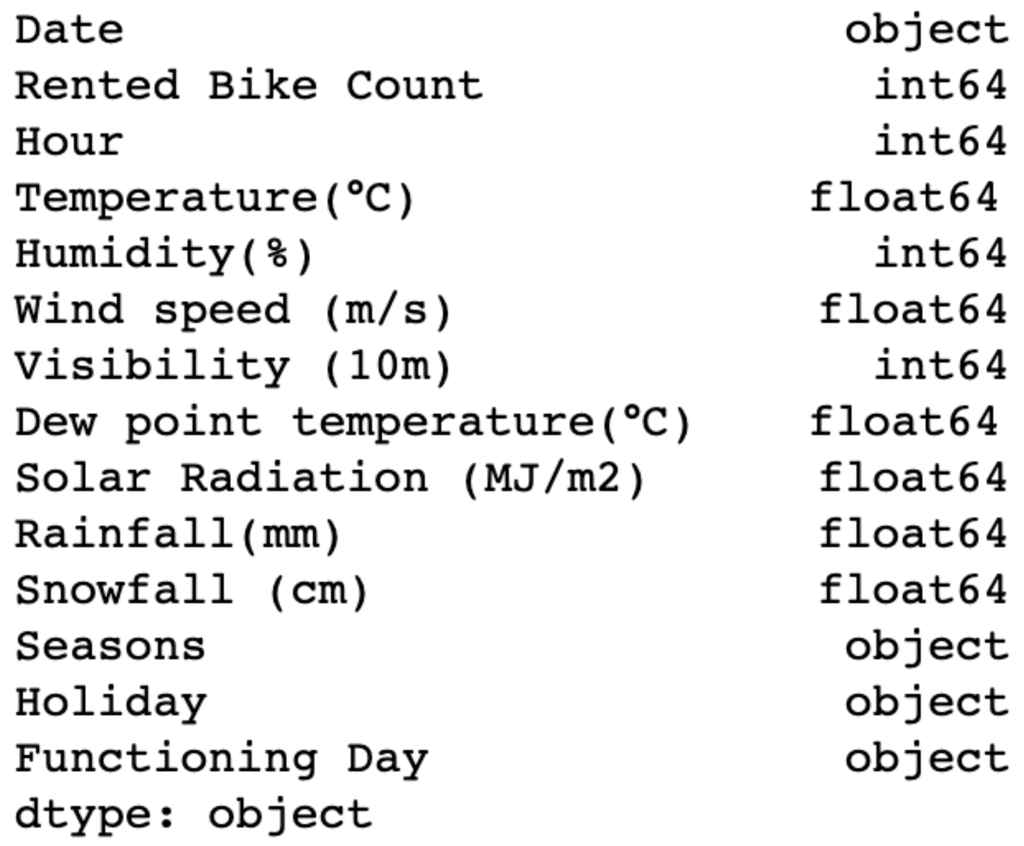


3-Data preparation for modeling

- **Check the data type:**

-’ Date’ is in type of Object, we need to split that

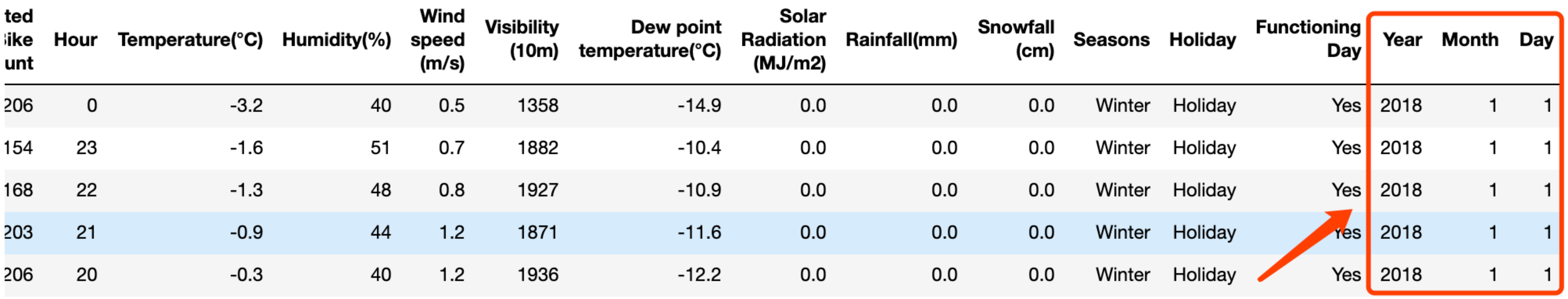
-’ Seasons’ ‘Holiday’ ‘Functioning Day’ is in type of Object, we need to change them into Int type.



```
Date                              object
Rented Bike Count                  int64
Hour                               int64
Temperature(°C)                  float64
Humidity(%)                        int64
Wind speed (m/s)                 float64
Visibility (10m)                   int64
Dew point temperature(°C)        float64
Solar Radiation (MJ/m2)          float64
Rainfall(mm)                     float64
Snowfall (cm)                    float64
Seasons                           object
Holiday                           object
Functioning Day                   object
dtype: object
```

3-Data preparation for modeling

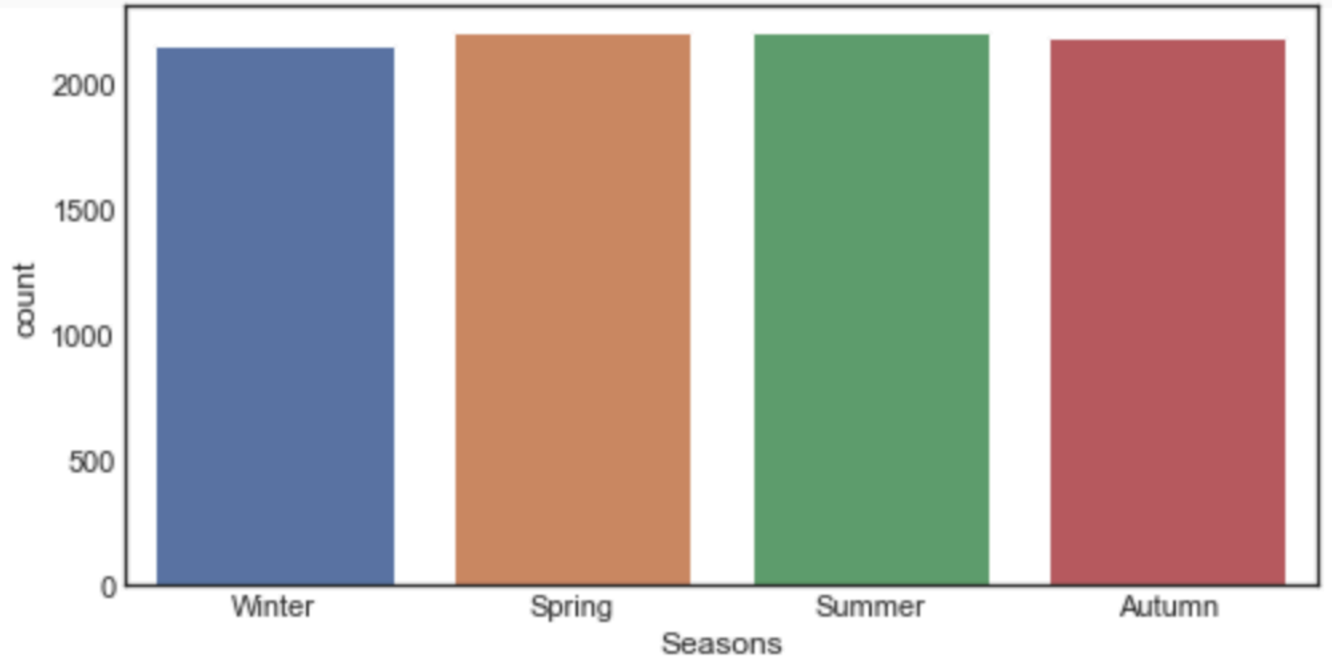
- **Spliting the “Date” feature into 3 independent features, then drop the original attribute, and we get the new dataset as follow.**

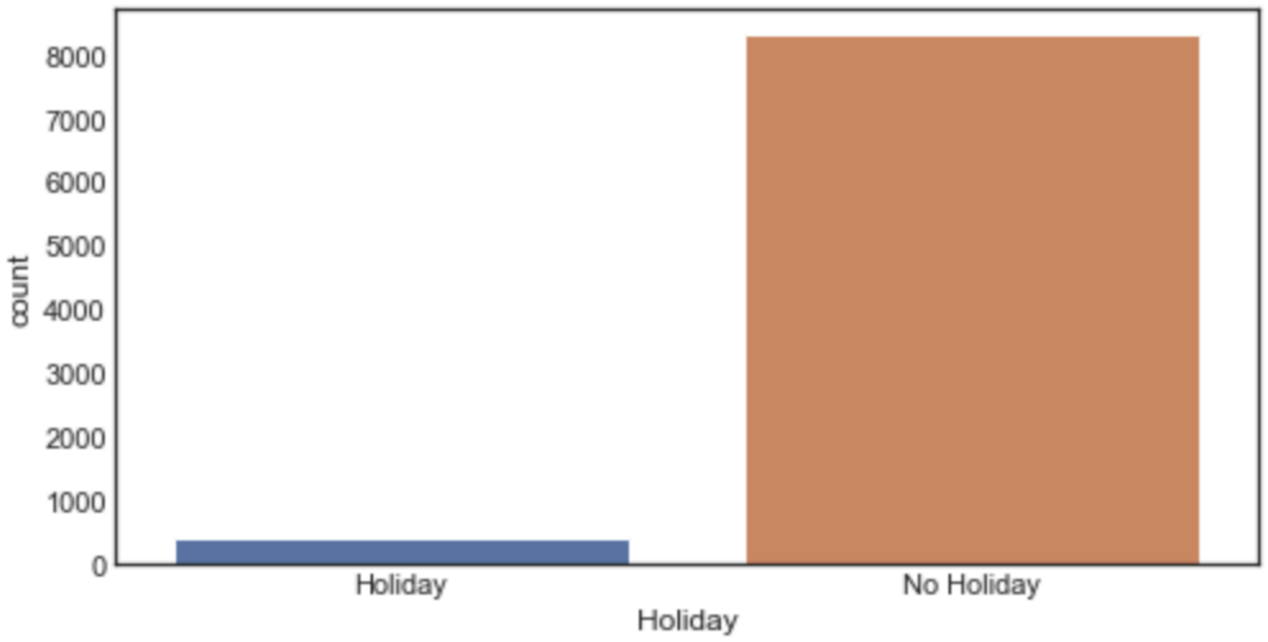


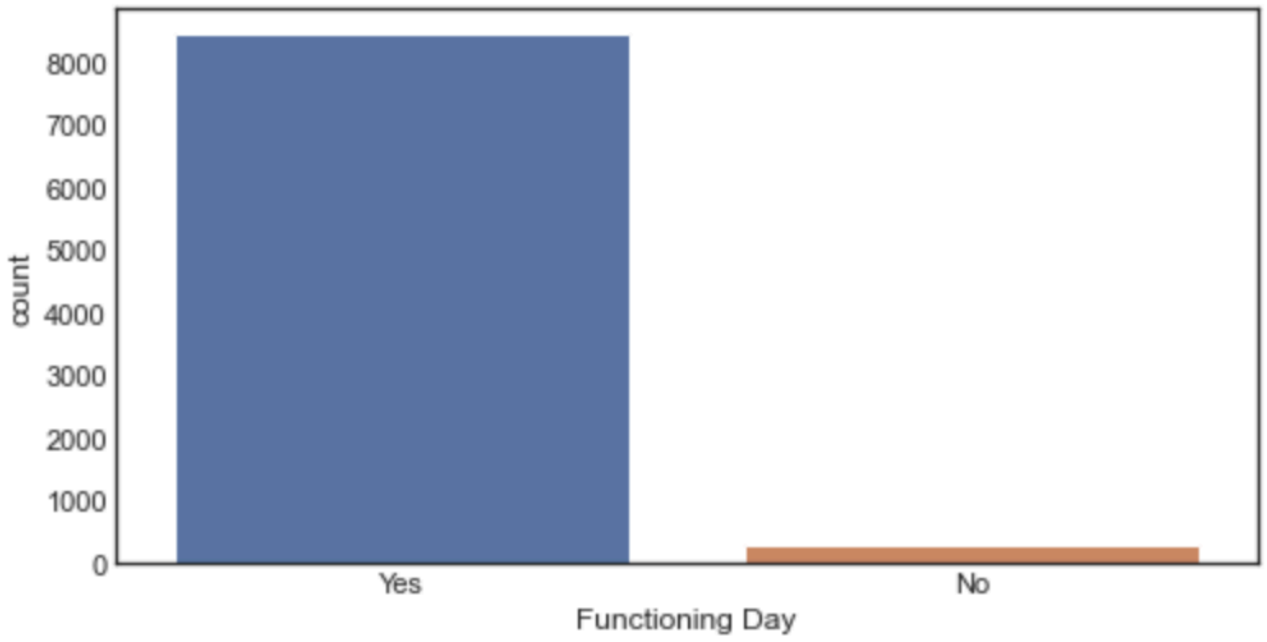
ted ike unt	Hour	Temperature(°C)	Humidity(%)	Wind speed (m/s)	Visibility (10m)	Dew point temperature(°C)	Solar Radiation (MJ/m2)	Rainfall(mm)	Snowfall (cm)	Seasons	Holiday	Functioning Day	Year	Month	Day
206	0	-3.2	40	0.5	1358	-14.9	0.0	0.0	0.0	Winter	Holiday	Yes	2018	1	1
154	23	-1.6	51	0.7	1882	-10.4	0.0	0.0	0.0	Winter	Holiday	Yes	2018	1	1
168	22	-1.3	48	0.8	1927	-10.9	0.0	0.0	0.0	Winter	Holiday	Yes	2018	1	1
203	21	-0.9	44	1.2	1871	-11.6	0.0	0.0	0.0	Winter	Holiday	Yes	2018	1	1
206	20	-0.3	40	1.2	1936	-12.2	0.0	0.0	0.0	Winter	Holiday	Yes	2018	1	1

3-Data preparation for mode

- **The “Seasons” “Holiday” “Functioning Day” are categorical data with Object data type.**

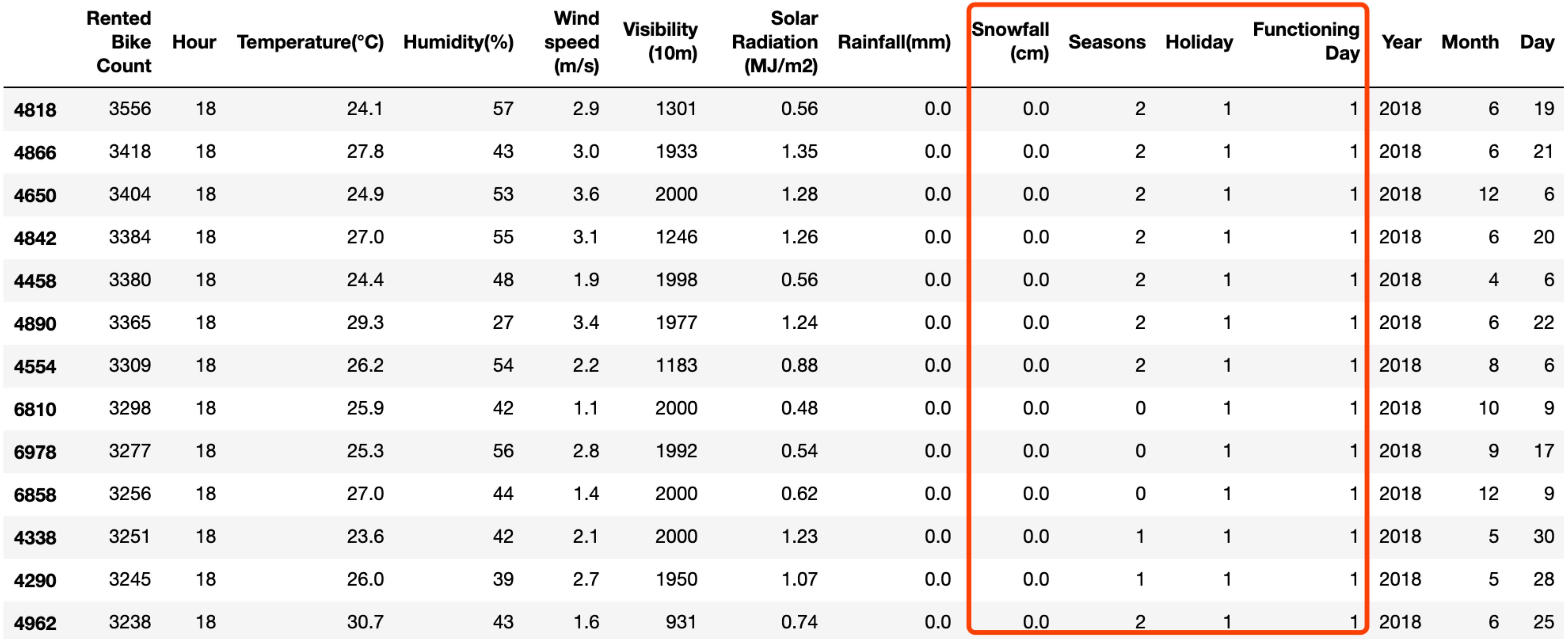






3-Data preparation for modeling

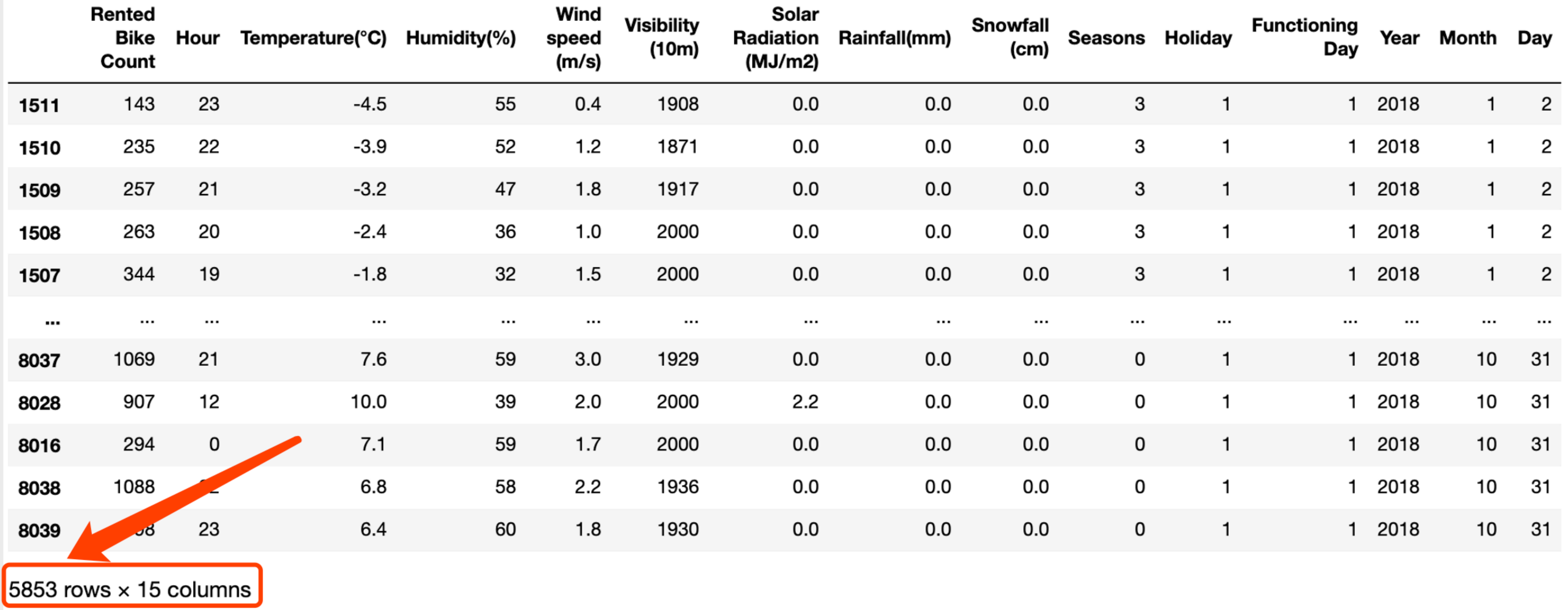
- **We change the "Seasons" "Holiday" "Functioning Day" into Int type, then drop the original attributions and get the new dataset.**



	Rented Bike Count	Hour	Temperature(°C)	Humidity(%)	Wind speed (m/s)	Visibility (10m)	Solar Radiation (MJ/m2)	Rainfall(mm)	Snowfall (cm)	Seasons	Holiday	Functioning Day	Year	Month	Day
4818	3556	18	24.1	57	2.9	1301	0.56	0.0	0.0	2	1	1	2018	6	19
4866	3418	18	27.8	43	3.0	1933	1.35	0.0	0.0	2	1	1	2018	6	21
4650	3404	18	24.9	53	3.6	2000	1.28	0.0	0.0	2	1	1	2018	12	6
4842	3384	18	27.0	55	3.1	1246	1.26	0.0	0.0	2	1	1	2018	6	20
4458	3380	18	24.4	48	1.9	1998	0.56	0.0	0.0	2	1	1	2018	4	6
4890	3365	18	29.3	27	3.4	1977	1.24	0.0	0.0	2	1	1	2018	6	22
4554	3309	18	26.2	54	2.2	1183	0.88	0.0	0.0	2	1	1	2018	8	6
6810	3298	18	25.9	42	1.1	2000	0.48	0.0	0.0	0	1	1	2018	10	9
6978	3277	18	25.3	56	2.8	1992	0.54	0.0	0.0	0	1	1	2018	9	17
6858	3256	18	27.0	44	1.4	2000	0.62	0.0	0.0	0	1	1	2018	12	9
4338	3251	18	23.6	42	2.1	2000	1.23	0.0	0.0	1	1	1	2018	5	30
4290	3245	18	26.0	39	2.7	1950	1.07	0.0	0.0	1	1	1	2018	5	28
4962	3238	18	30.7	43	1.6	931	0.74	0.0	0.0	2	1	1	2018	6	25

3-Data preparation for modeling

- **Outliers Removal**: We use IQR to define a multiplier which is 1.5 ideally that will decide how far below Q1 and above Q3 will be considered as an Outlier. Delete the outliers and get the new set with 5853 items left.

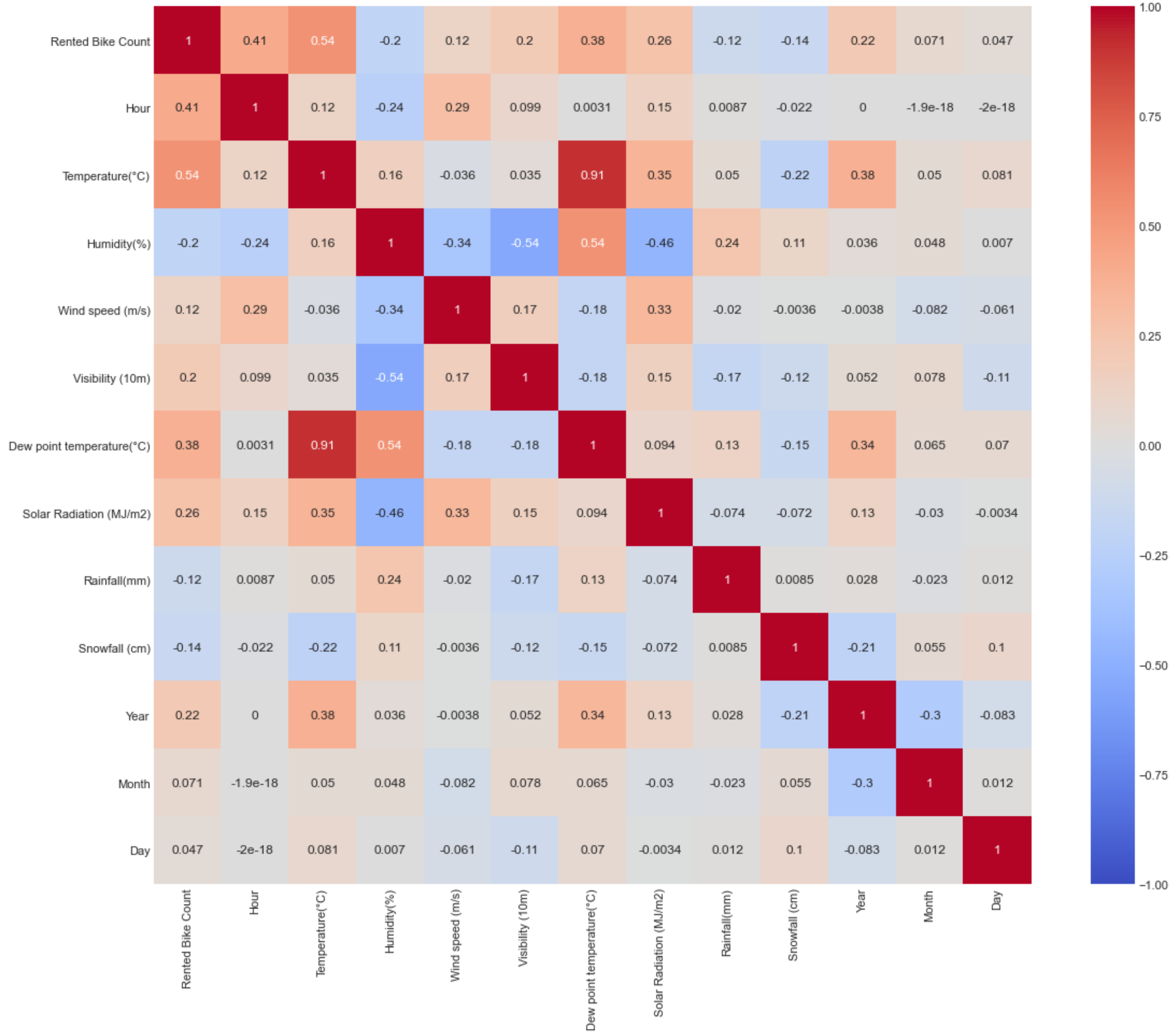


	Rented Bike Count	Hour	Temperature(°C)	Humidity(%)	Wind speed (m/s)	Visibility (10m)	Solar Radiation (MJ/m2)	Rainfall(mm)	Snowfall (cm)	Seasons	Holiday	Functioning Day	Year	Month	Day
1511	143	23	-4.5	55	0.4	1908	0.0	0.0	0.0	3	1	1	2018	1	2
1510	235	22	-3.9	52	1.2	1871	0.0	0.0	0.0	3	1	1	2018	1	2
1509	257	21	-3.2	47	1.8	1917	0.0	0.0	0.0	3	1	1	2018	1	2
1508	263	20	-2.4	36	1.0	2000	0.0	0.0	0.0	3	1	1	2018	1	2
1507	344	19	-1.8	32	1.5	2000	0.0	0.0	0.0	3	1	1	2018	1	2
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
8037	1069	21	7.6	59	3.0	1929	0.0	0.0	0.0	0	1	1	2018	10	31
8028	907	12	10.0	39	2.0	2000	2.2	0.0	0.0	0	1	1	2018	10	31
8016	294	0	7.1	59	1.7	2000	0.0	0.0	0.0	0	1	1	2018	10	31
8038	1088	22	6.8	58	2.2	1936	0.0	0.0	0.0	0	1	1	2018	10	31
8039	[illegible]	23	6.4	60	1.8	1930	0.0	0.0	0.0	0	1	1	2018	10	31

5853 rows × 15 columns

3-Data preparation

- **Delete the correlated attributions**: from the map we can see temperature and dew point temperature are highly correlated, se we drop one to avoid the multicollinearity, and left one which has higher correlation with our aim ‘Rented Bike Count’ .



3-Data preparation for modeling

- **Make the training dataset and test dataset**:

-separate the ‘Rented Bike Count’ which is our aim from the original dataset, and get the Bike_Data_y, the rest data is Bike_Data_x.

-then use the train_test_split model in sklearn to split the data.

-after splitting we will get for dataset X_train, X_test, y_train, y_test.

-do normalization before modelization step.

4-Modelization and Validation

- **We choose linear regression model as it is regression problem.**

-choose LinearRregression model from sklearn

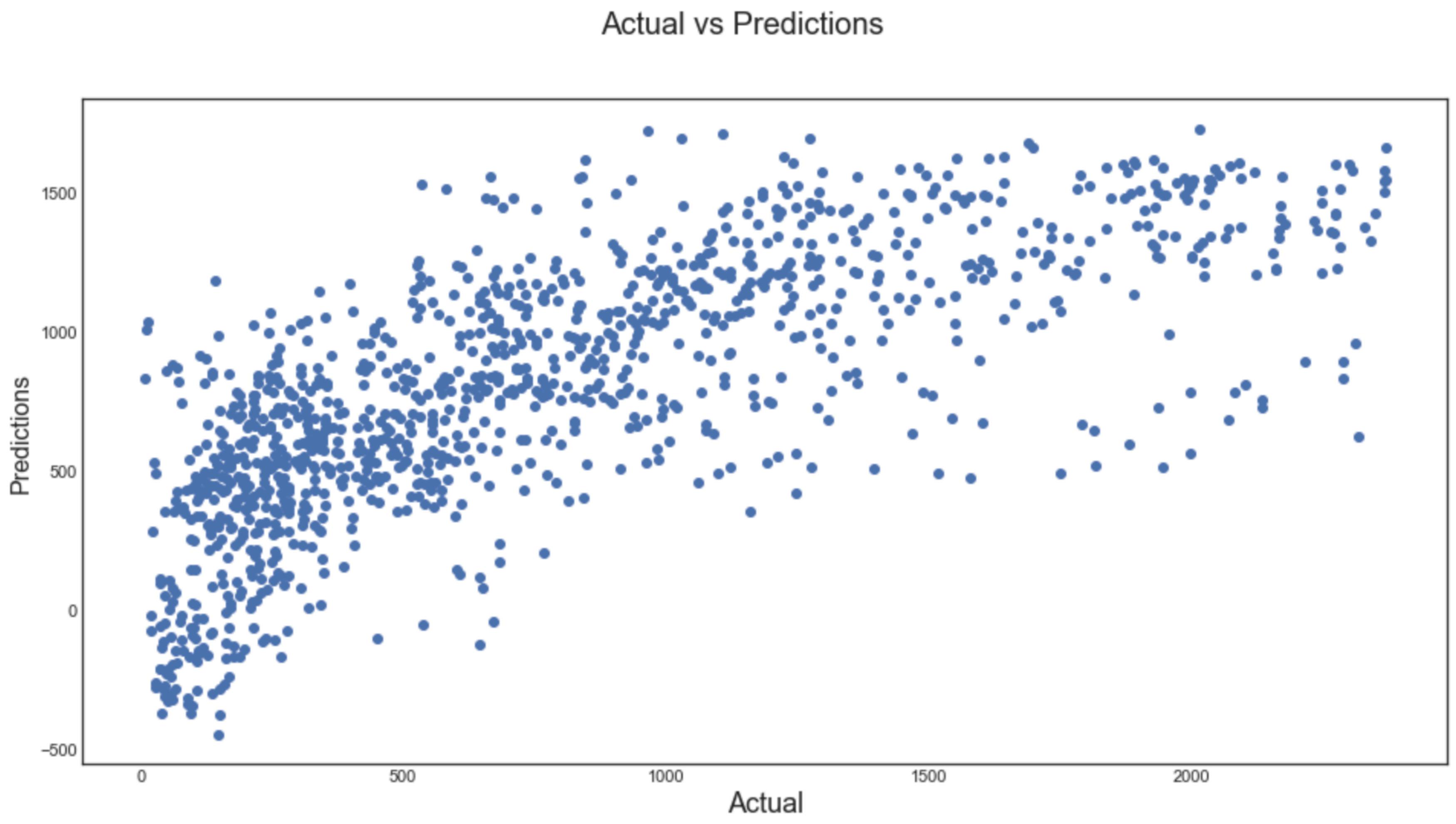
-fit the model with training set

-make prediction with training data

-get the final score of this regression model which is 0.54

4-Modelization and Validation

- **Make prediction with the test data to evaluate the model, plot the actual data and predicted data on same graph.**



4-Modelization and Validation

- **Model comparation**: import from sklearn the other models, then do modelization in same way.

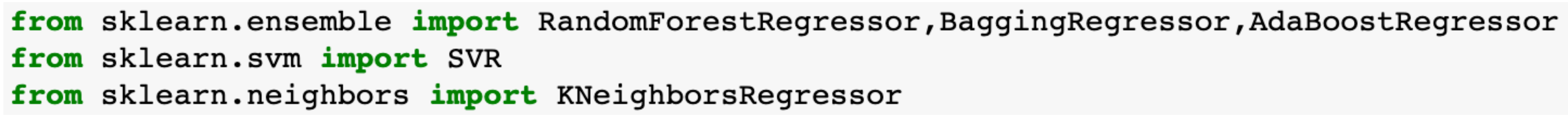
  RandomForestRegressor

  AdaBoostRegressor

  BaggingRegressor

  SVR

  KNeighborsRegressor

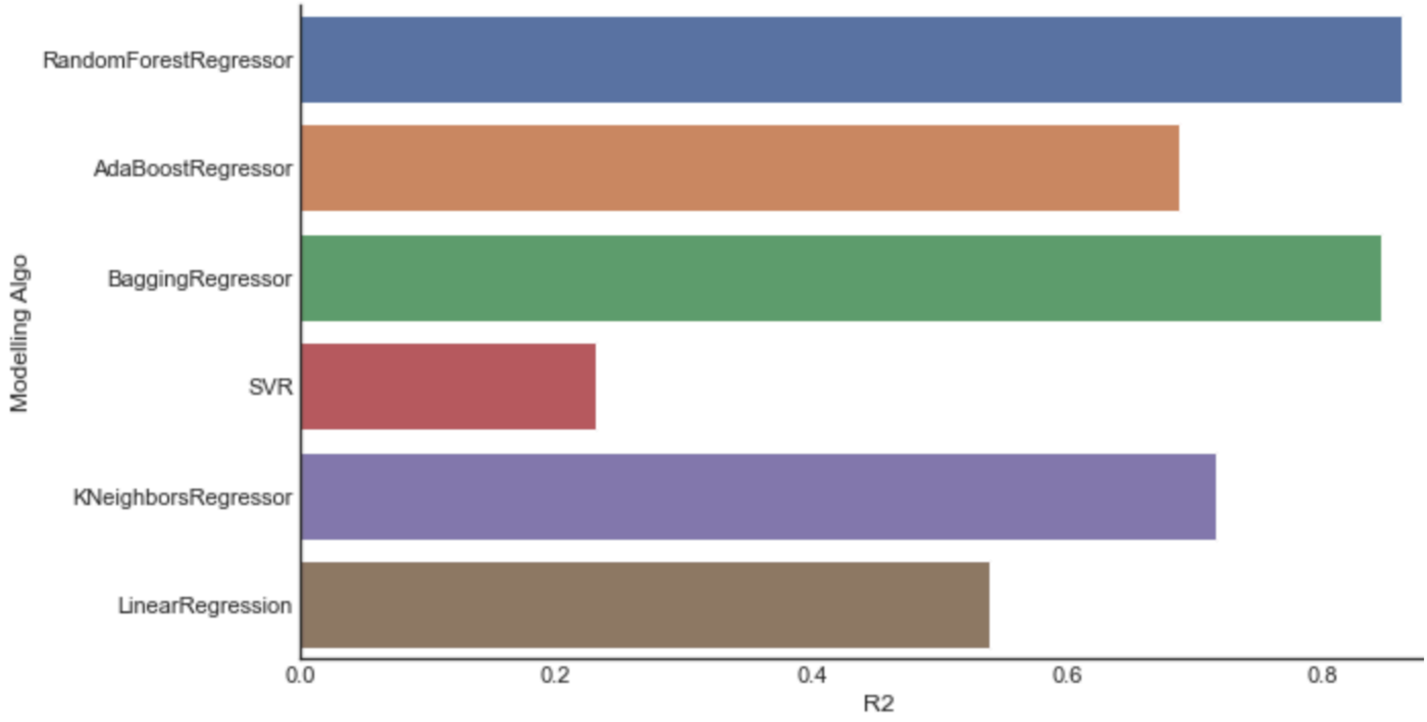


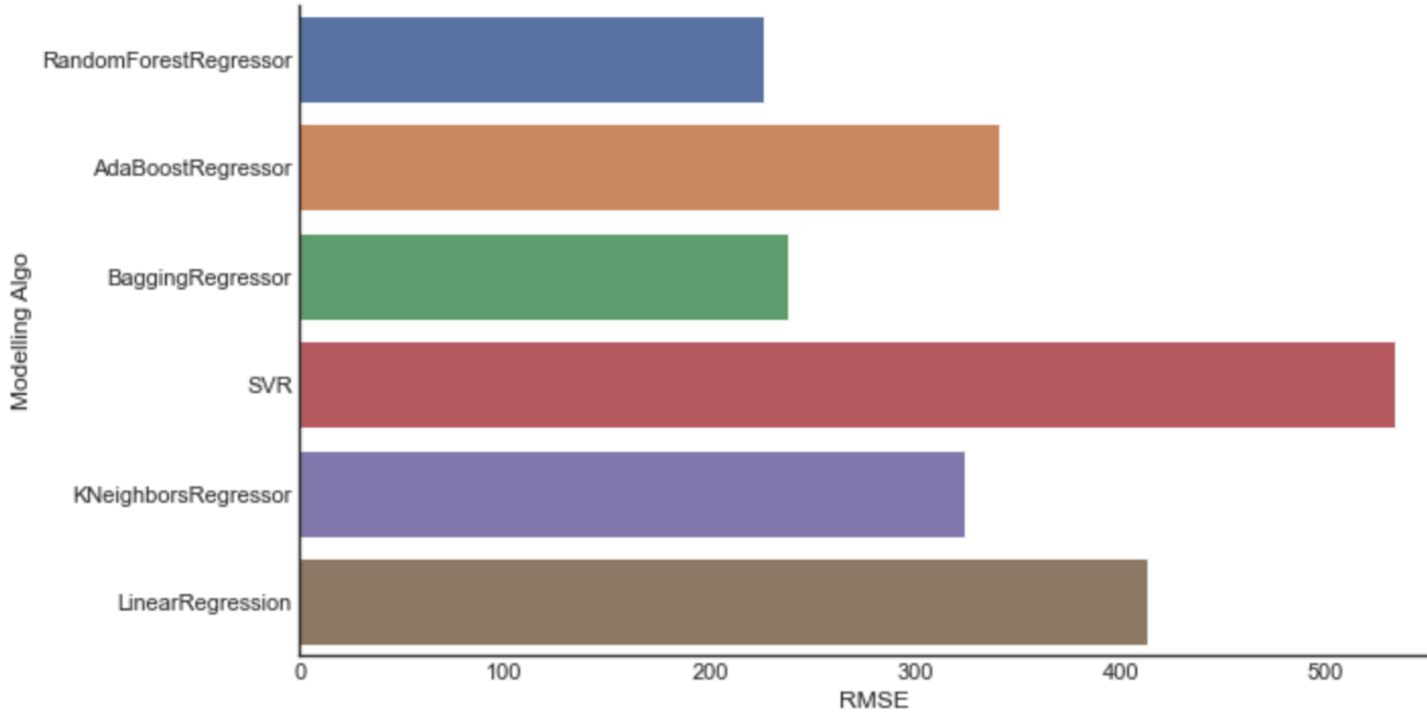
```python
from sklearn.ensemble import RandomForestRegressor,BaggingRegressor,AdaBoostRegressor
from sklearn.svm import SVR
from sklearn.neighbors import KNeighborsRegressor
```

4-Modelization and Validat

- **Model comparation**:

According to the comparation, RandomForestRregression has the less RMSE value and high R2 value, so we choose random forest model and further optimize its hyper parameters.





4-Modelization and Validation

- **Hyper parameter optimization**:

-Using Gridsearch(GridSearchCV model from sklearn) to find the best set of hyper parameter, and use the best parameter to get the final model.

-save the model we get into new file for it can be used easily afterwards.

-use Flask to get the API.

Thank you!