PYTHON FOR DATA ANALYSIS

PROJECT – 2021

OCÉANE GÖRKE
MICHELLE HATOUM

CONTEXT OF STUDY

The dataset was the Seoul Bike Data. The purpose was to study the conditions of rental during a whole year to be able to predict future rentals depending on the following parameters:

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

FAMILIARIZING WITH THE DATASET

The shape, head, dtypes, describe and tail functions helped understanding the dataset and opening possibilities of study.

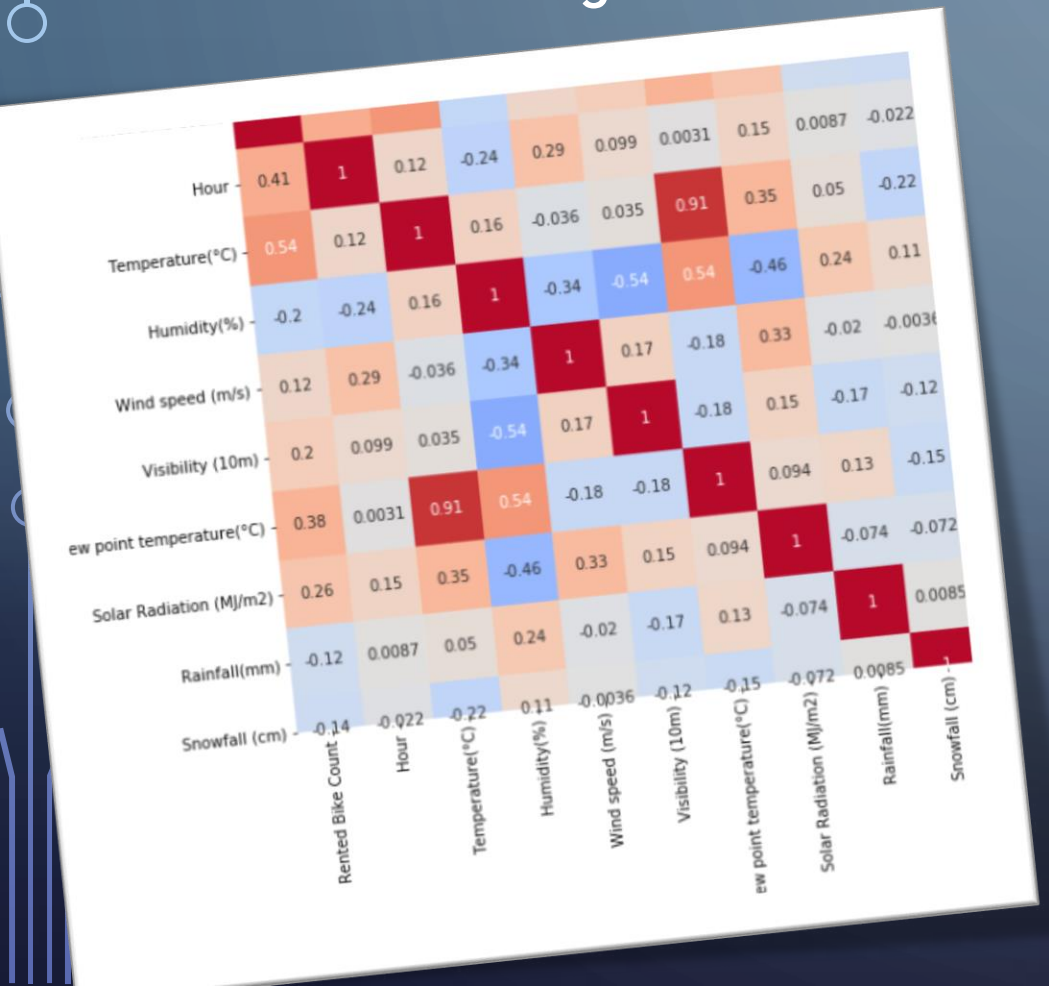
	Date	Rented Bike Count	Hour	Temperature(°C)	Humidity(%)	Wind speed (m/s)	Visibility (10m)	Dew point temperature(°C)	Solar Radiation (MJ/m2)	Rainfall(mm)	Snowfall (cm)	Seasons	Holiday	Functioning Day
0	01/12/2017	254	0	-5.2	37	2.2	2000	-17.6	0.0	0.0	0.0	Winter	No Holiday	Yes
1	01/12/2017	204	1	-5.5	38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

The target value is Rented Bike Count

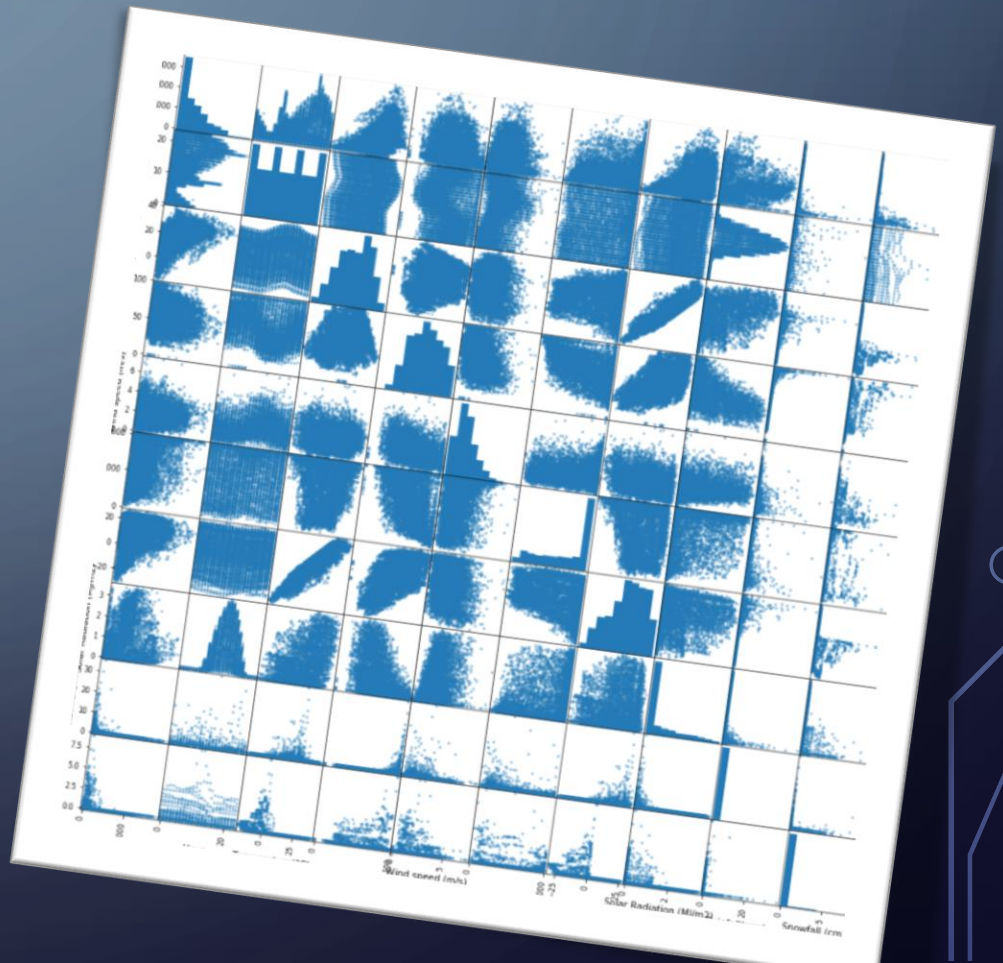
4 features out of 13 aren't ints of floats but object type.

DETECTING THE RELATIONSHIPS

After making sure there was no null values, the scatter plot and correlation matrix gave an idea of the relationships between every feature.



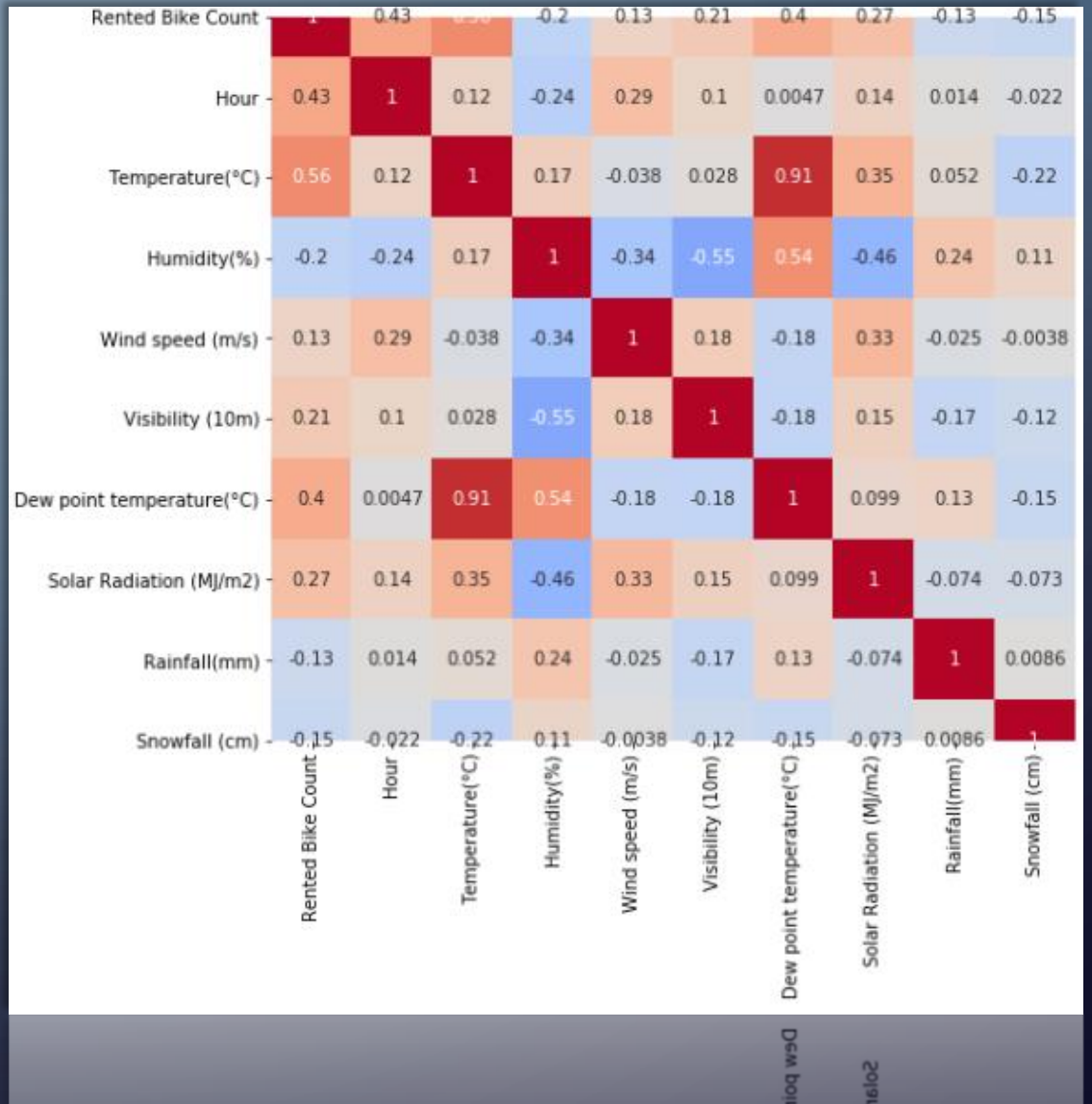
4 values are missing here, Seansons, Holiday, Date and Functioning Day.



MANIPULING SOME COLUMNS

The first column to change is Functioning Day: the rows where this value is « no » are days where there were no rentals because the bikes weren't available. It means that these values are distorting the dataset. The « no » rows were deleted, and the Functioning Day column removed.

The correlation matrix hasn't changed a lot after the removal: Functioning Day column wasn't useful for the study.



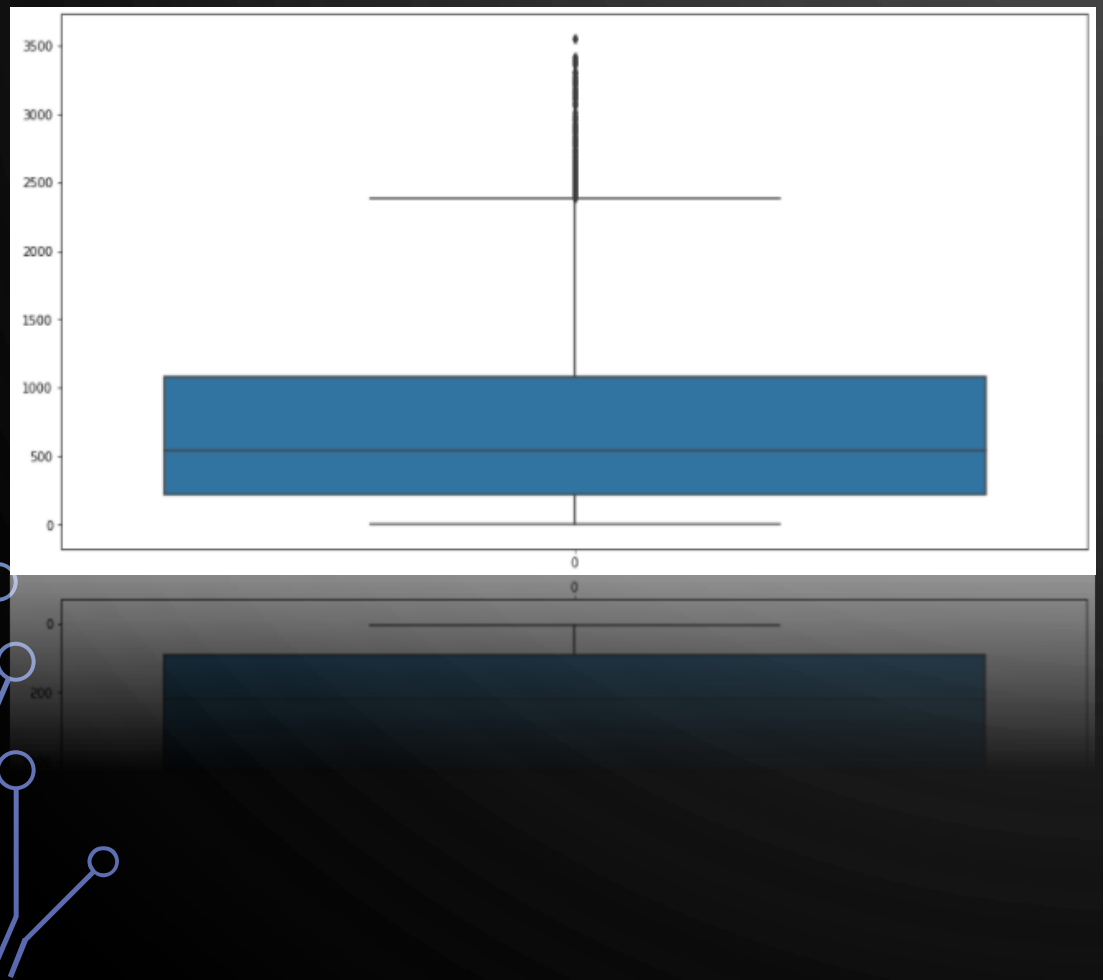
CHANGING THE OBJECT VALUES TYPE

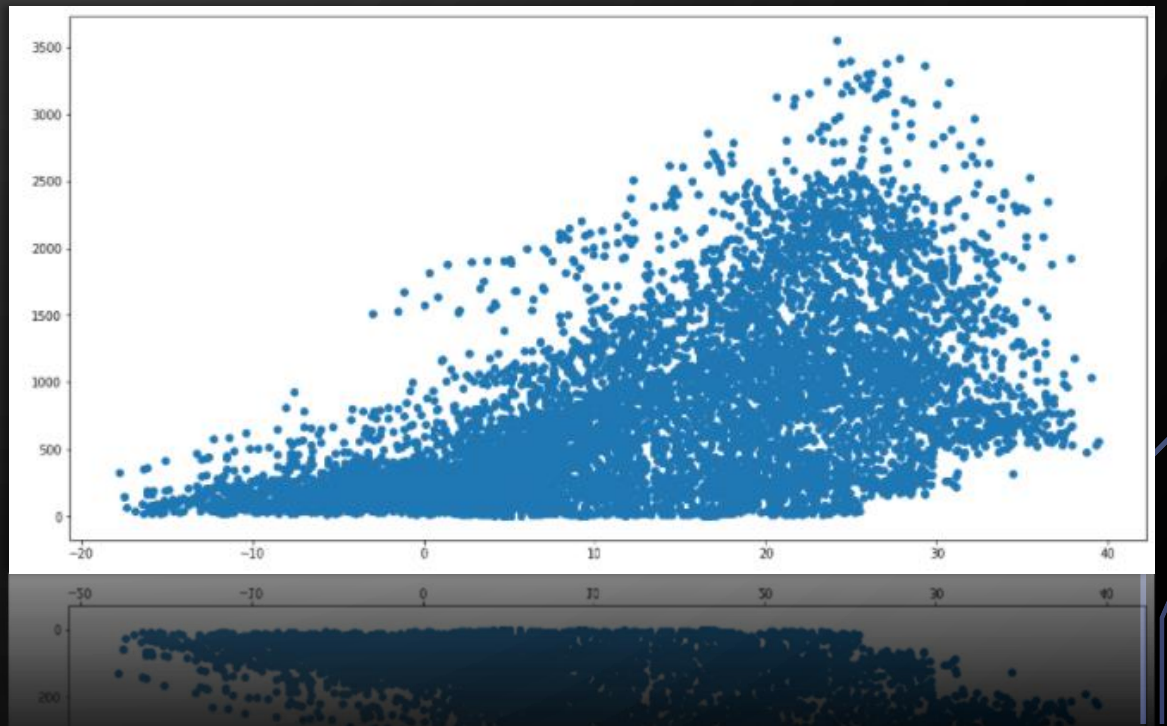
- Seasons column: Winter was changed to 0, Spring to 1, Summer to 2 and Autmn became a 3.
- Holiday became a binary value, 1 if it is a holiday, 0 if not.
- Date was deleted in order to create a new column: there was too many dates so grouping them according to the month seemed to be the right idea. Month column was created and contains ints from 1 to 12

	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	Month
0	254	0	-5.2	37	2.2	2000	-17.6	0.0	0.0	0.0	0	0	12
1	204	1	-5.5	38	0.8	2000	-17.6	0.0	0.0	0.0	0	0	12
2	173	2	-6.0	39	1.0	2000	-17.7	0.0	0.0	0.0	0	0	12
3	107	3	-6.2	40	0.9	2000	-17.6	0.0	0.0	0.0	0	0	12
4	78	4	-6.0	36	2.3	2000	-18.6	0.0	0.0	0.0	0	0	12

LOOKING FOR OUTLIERS

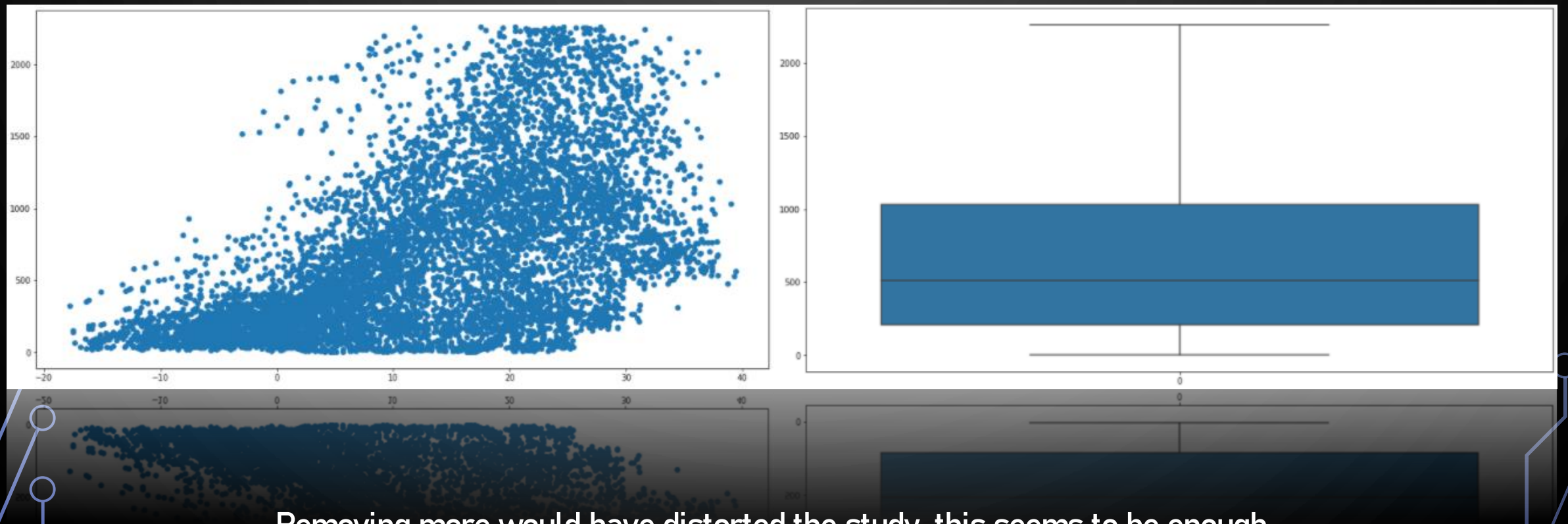
As shown below, there is a good number of outliers when Rented Bike Count is studied





GETTING RID OF OUTLIERS

Using the boxplot, a threshold was found. Deleting every row above it led to this :

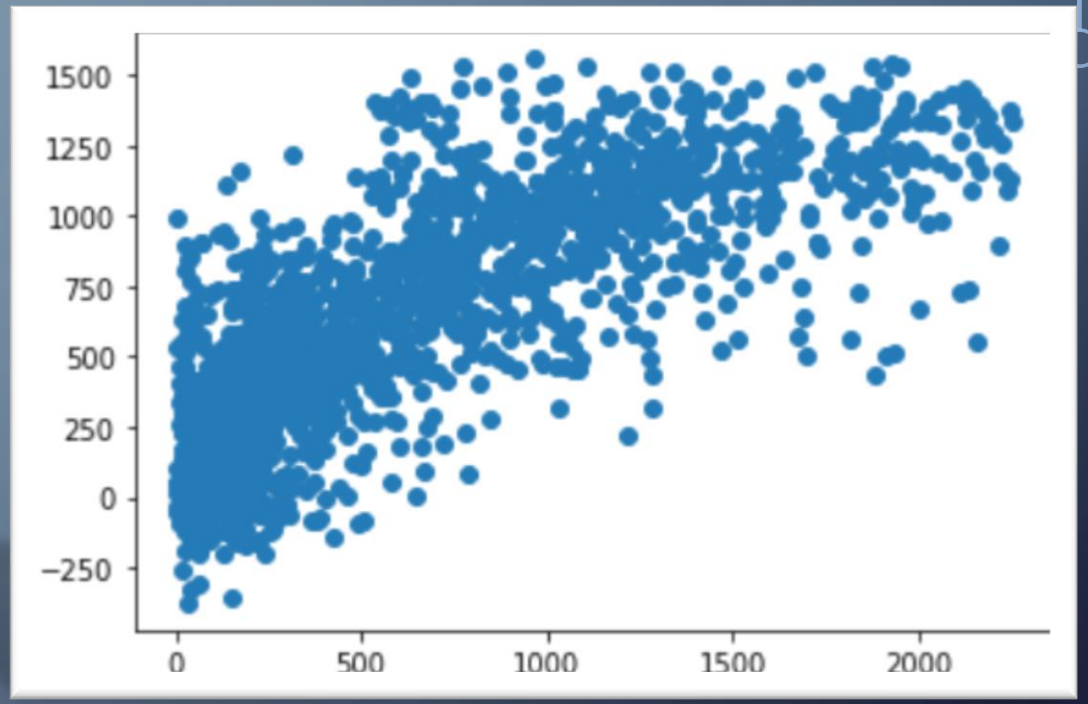


Removing more would have distorted the study, this seems to be enough.

LINEAR REGRESSION

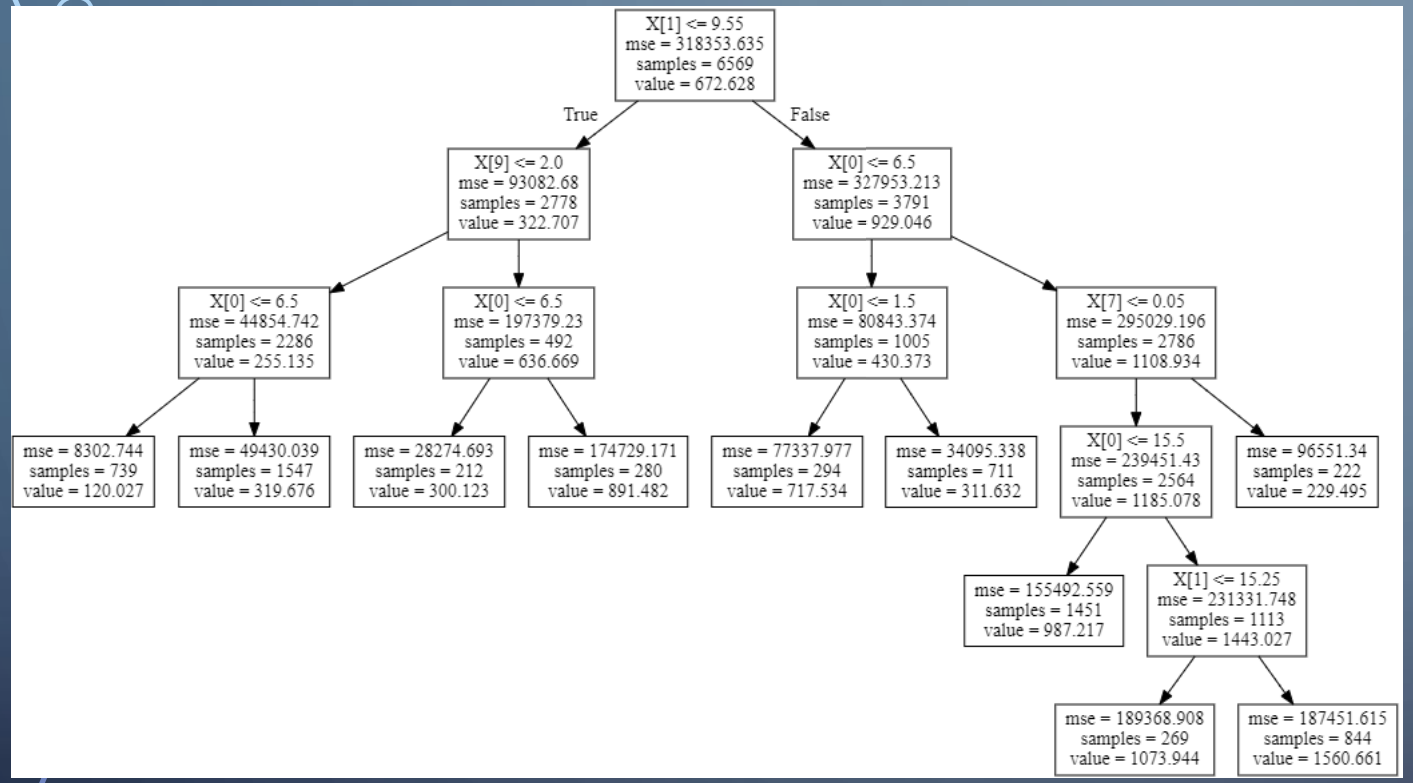
Expectations weren't high for this model, but it was chosen as the first model to start with. Coefficients weren't good, the prediction wasn't very linear, not to mention the low R^2 and high MSE

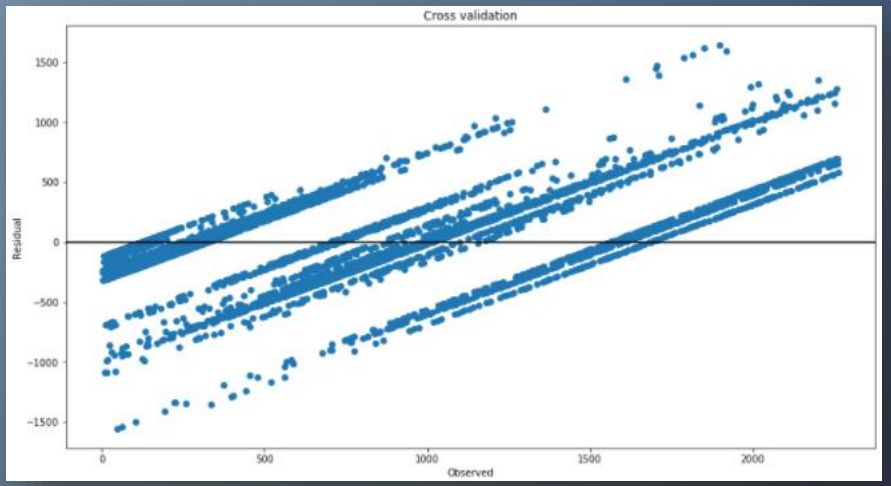
	Coeff
Hour	25.363774
Temperature(°C)	13.113305
Humidity(%)	-9.398217
Wind speed (m/s)	7.343671
Visibility (10m)	-0.011744
Dew point temperature(°C)	9.389498
Solar Radiation (MJ/m2)	-31.385259
Rainfall(mm)	-57.251111
Snowfall (cm)	11.024930
Seasons	114.629639
Holiday	-91.145122
Month	-0.164729




```
R-squared scores : 0.5371366563233642
Root mean square error : 367.04549845511116
Mean absolute error : 279.266398963082
```

DECISION TREE

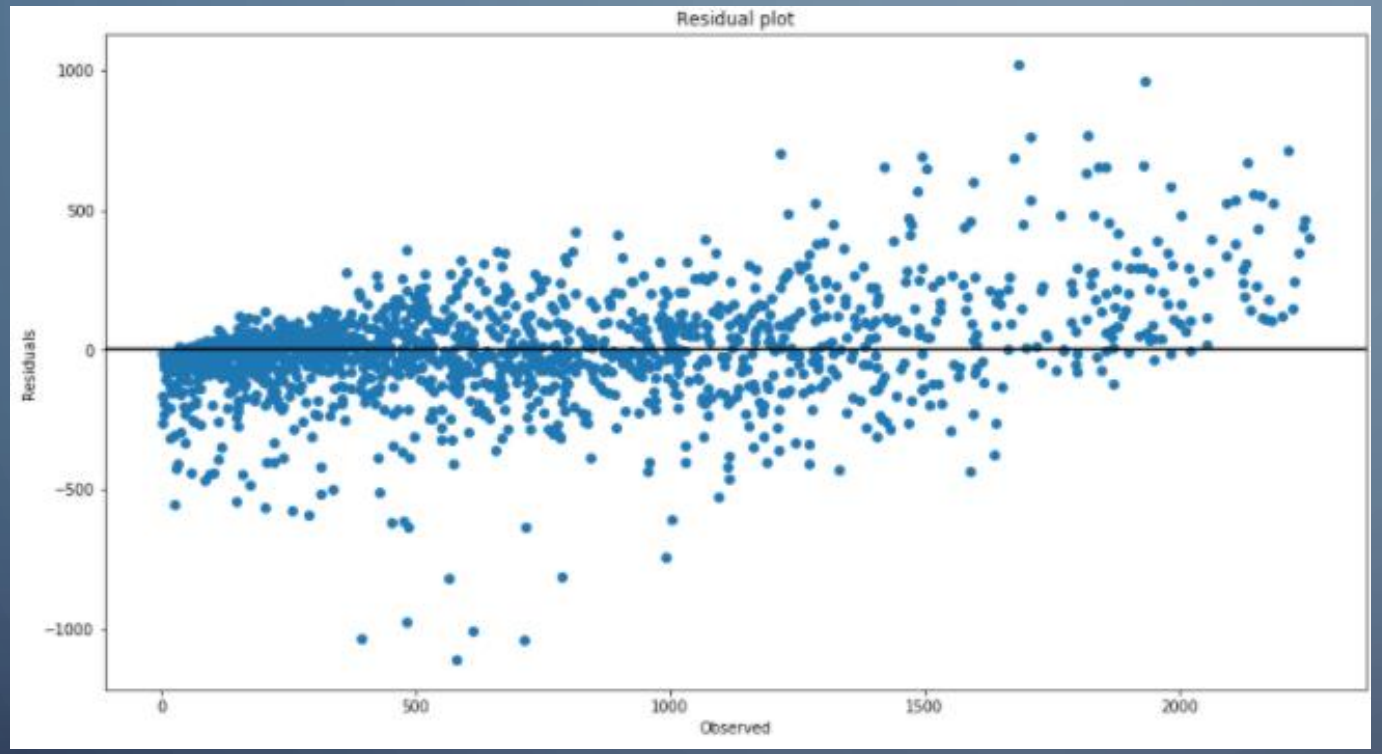




Here, the model was better. The accuracy of model was relatively good. R^2 was better and MSE lower

```
R-squared scores : 0.6812338913285774
Root mean square error : 301.1492601467102
Mean absolute error : 216.19529811699263
```

RANDOM FOREST (USING EVERY FEATURE)



R-squared scores : 0.8542834784919341
Root mean square error : 186.99434177335297
Mean absolute error : 121.05479306147294

	predictions
0	1761.065
1	236.035
2	1004.895
3	199.885
4	332.615
...	...
1638	971.490
1639	893.195
1640	749.480
1641	1854.585
1642	194.145

For this model, the accuracy was surprisingly high: 0.98101869527. This was the best model tried.

PLAYING WITH FEATURES

After trying to remove the columns that weren't correlated too much to Rented Bike Count, the only good model found is the one using the following features: Hour, Temperature(°C), Humidity(%), Dew point temperature(°C), Solar Radiation (MJ/m2), Rainfall(mm), Snowfall (cm), Seasons and Month.

However, it wasn't better than the previous model.

```
rf_score =rf.score(train,y_train)
print('Accuracy of the model :',rf_score)
```

```
Accuracy of the model : 0.980151091922123 1
```

```
r2_scores = cross_val_score(rf, train, y_train, cv=3)
print('R-squared scores :',np.average(r2_scores))
```

```
R-squared scores : 0.8525376982445142
```

COMPARING AND CONCLUDING

	Linear regression	Decision tree	Random forest 1	Random forest 2
Accuracy	0.5470972695736975	0.7016025235112946	0.9822029122451313	0.9819967014791258
R^2	0.5443289406062134	0.6904064769802126	0.8603320890157589	0.8575915591602193
MSE	380.55786677414096	326.7467927318254	213.64721144409134	393.9071326805936

The best results are the random forest 1's. It is the best model tried.