Politecnico di Torino

Seoul Bike Sharing Demand

Garbarino Matteo s265386

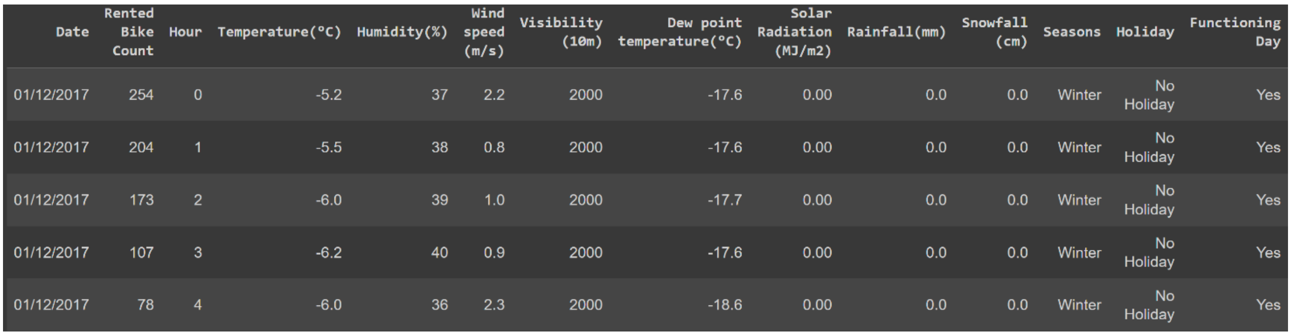
Table of contents

1. Introduction

The aim of this project is to perform an analysis on the Seoul Bike Sharing Demand Data Set [1] and build a model in a supervised setting able to predict newcoming datapoints. Considering the given dataset, the problems setting consists of a modest number of features referring to environmental conditions and date information, associated to the prediction of the number of bikes rented per hour, which is going to be our target variable. Despite being an integer value, the target variable has still to be considered a continuous value, since it would be inefficient, counterproductive and incorrect to treat it as a categorical data. Given the aforementioned considerations, the problem setting is the one of Linear Regression. The analysis, modelling and construction of a solution are conducted through the usage of the Python 3 programming language and a number of libraries which can be found, together with the code of the solution in the GitHub repository of the project [2].

2. Dataset structure

The dataset contains entries about an entire year of functioning of a bike rental service, with information about rented bikes with a granularity of one hour. Therefore, it is composed of a total of 365 days * 24hours a day = 8760 distinct entries.



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
01/12/2017	254	0	-5.2	37	2.2	2000	-17.6	0.00	0.0	0.0	Winter	No Holiday	Yes
01/12/2017	204	1	-5.5	38	0.8	2000	-17.6	0.00	0.0	0.0	Winter	No Holiday	Yes
01/12/2017	173	2	-6.0	39	1.0	2000	-17.7	0.00	0.0	0.0	Winter	No Holiday	Yes
01/12/2017	107	3	-6.2	40	0.9	2000	-17.6	0.00	0.0	0.0	Winter	No Holiday	Yes
01/12/2017	78	4	-6.0	36	2.3	2000	-18.6	0.00	0.0	0.0	Winter	No Holiday	Yes

Fig.1: Sample of the dataset

There is a total of 14 attributes, of which one is the target variable (namely "Rented Bike Count") and the remaining 13 will constitute the set of predictors. A detailed list of the features follows:

- Date (date type dd/mm/yyyy – it will be properly restructured in the Data Preprocessing section in order to provide valuable information for the model)
- Rented Bike Count (numeric, integer – Target Variable)
- Hour (numeric, integer, [0;23])
- Temperature(°C) (numeric, float)
- Humidity(%) (numeric, integer, [0;100])
- Wind Speed(m/s) (numeric, float, [0;+inf])
- Visibility(10m) (numeric, integer, [0;2000])
- Dew point temperature(°C) (numeric, float)
- Solar Radiation(MJ/m2) (numeric, float, [0;+inf])
- Rainfall(mm) (numeric, float, [0;+inf])
- Snowfall(cm) (numeric, float, [0;+inf])
- Seasons (categorical: "Winter", "Spring", "Summer", "Autumn")
- Holiday (binary: "Holiday", "No Holiday")
- Functioning Day (binary: "Yes", "No")

3. Data exploration

This preliminary step is essential to capture information about the dataset and how to treat it in order to obtain a valid and well-performing model in the end. This set of substeps, together with the step of the actual data preprocessing tend to be extremely time consuming, but crucial in order to avoid the “garbage-in garbage-out” situation as well as biased models, outliers and unbalanced classes influencing the learning process, curse of dimensionality, using strongly linearly correlated features and a number of other possible issues. A preliminary consideration is that taking in account the low number of features it is possible that the curse of dimensionality will not affect this work and the attributes should be discarded according to other needs.

3.1 Balance

As a first analysis, the distribution of the target variable is analysed in order to spot possible major unbalanced values.

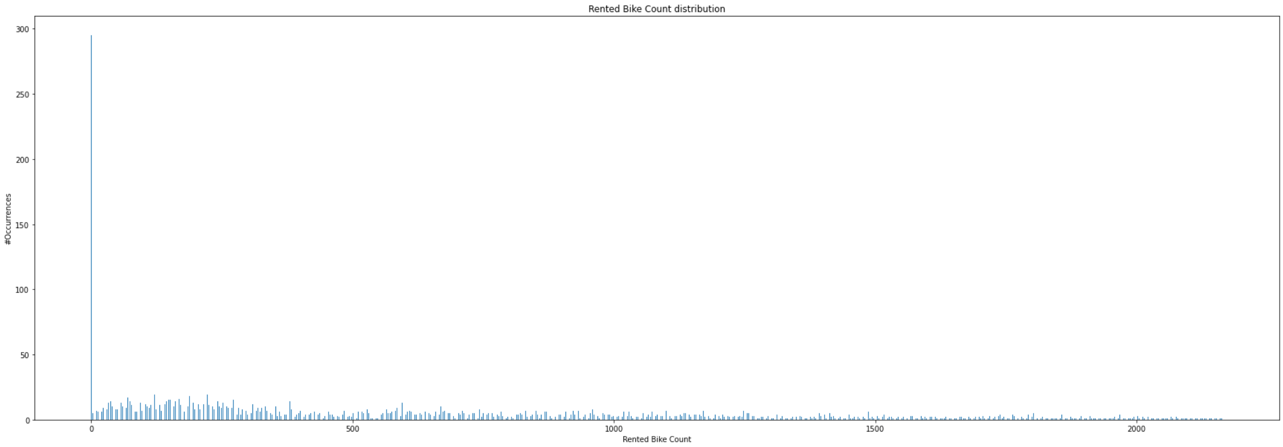


Fig.2: Target variable distribution

It is clearly visible the presence of an imbalanced amount values of that the value 0 has a frequency of 295, while on average the other values are less frequent by one to two orders of magnitude. This situation might require an undersampling of the entries with value equal zero (or an oversampling of all the other entries, but it would be too complex and alter too much the dataset). Another important consideration is that this case might not fall under a major imbalance case which could be instead recognized when there is at least a 1:1000 difference in proportions. Furthermore, this dataset is rather small (8760 entries) and it must be paid a special attention when removing datapoints. The selected approach in this case is going to be to do not alter the dataset composition with oversampling/undersampling since it could bring more harm than benefit (unless the models results will suggest to act differently).

3.2 Completeness

This rather simple check assesses the quality of the data in terms of missing points for each attribute. After a careful exploration the conclusion is that the dataset is complete for all the columns, hence it doesn’t contain any NULL / NaN values. In the unfortunate, but not uncommon, case in which they would have been present there could have been a number of viable solutions such as replacing them with a custom statistic calculated on the remaining entries of the dataset (e.g. mean or median) or simply drop them from the dataset. Luckily this dataset is small, but curated and therefore doesn’t need these transformations.

3.3 Correlation analysis

This step could probably have the most relevance and the higher impact on the analysis and usage of the dataset thanks to the possibility of evaluating the linear correlation among the features (pairwise) and to take action accordingly.

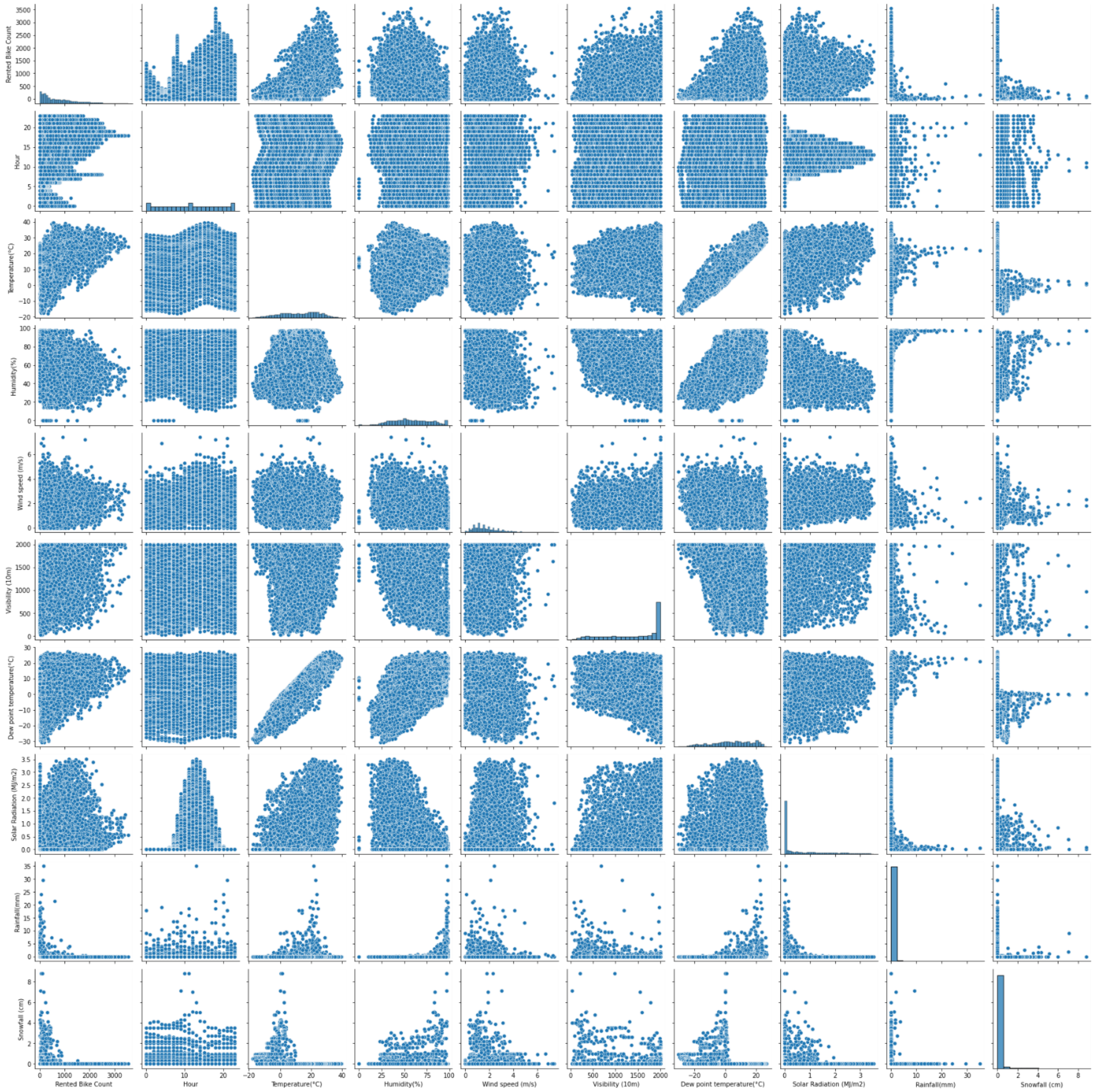


Fig.3 Pairs plot of the numeric features

In *Figure 3* the pairs plot generated through the Seaborn library displays two kinds of information:

- On the diagonal there is the distribution of each individual predictor.
- All the other entries of the matrix can be exploited to study the linear correlation between all the possible couple of numeric predictors.

Not all the distributions of the single attributes present the characteristics of a Normal distribution, which is due to two main factors: they can be treated as by definition one-sided Gaussians (e.g. the Snowfall(cm) is actually a gaussian centred around zero (snowfalls are rare events) with a very small variance and we neglect by definition the negative values which do not make sense for the attribute). Analogously the same happens for the Visibility(10m) whose value is capped at 2000, which also happens to be the most frequent value (fog and other weather conditions affecting the visibility are rare). It doesn’t record values grater than 2000 and the values on the left could resemble a Gaussian. These considerations are useful to understand that there aren’t distributions with unexpected behaviours worth of deeper analyses.

Concerning the linear correlations instead, it is important to clarify what to look for:

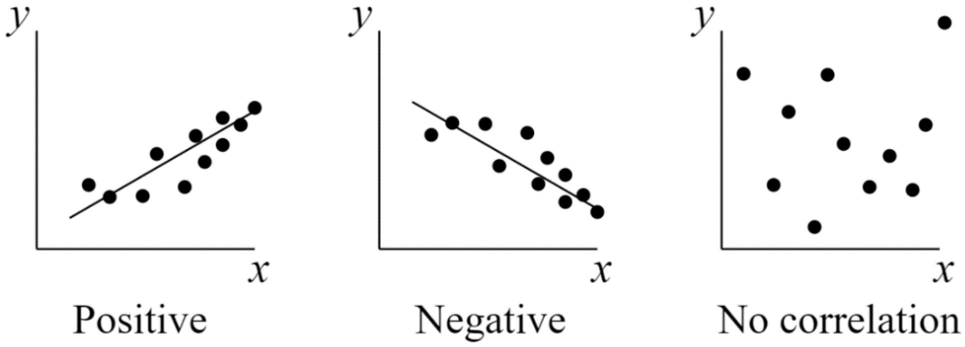


Fig.4: Visual examples of linear Correlation

Having this in mind and observing the pairs plot, it seems obvious that there could be a positive linear correlation between Temperature(°C) and Dew point temperature(°C) which makes perfectly sense since they are both temperatures (and the lower the first the lower the second). This clearly requires a deeper analysis which results in the generation of a heatmap of correlation coefficients.

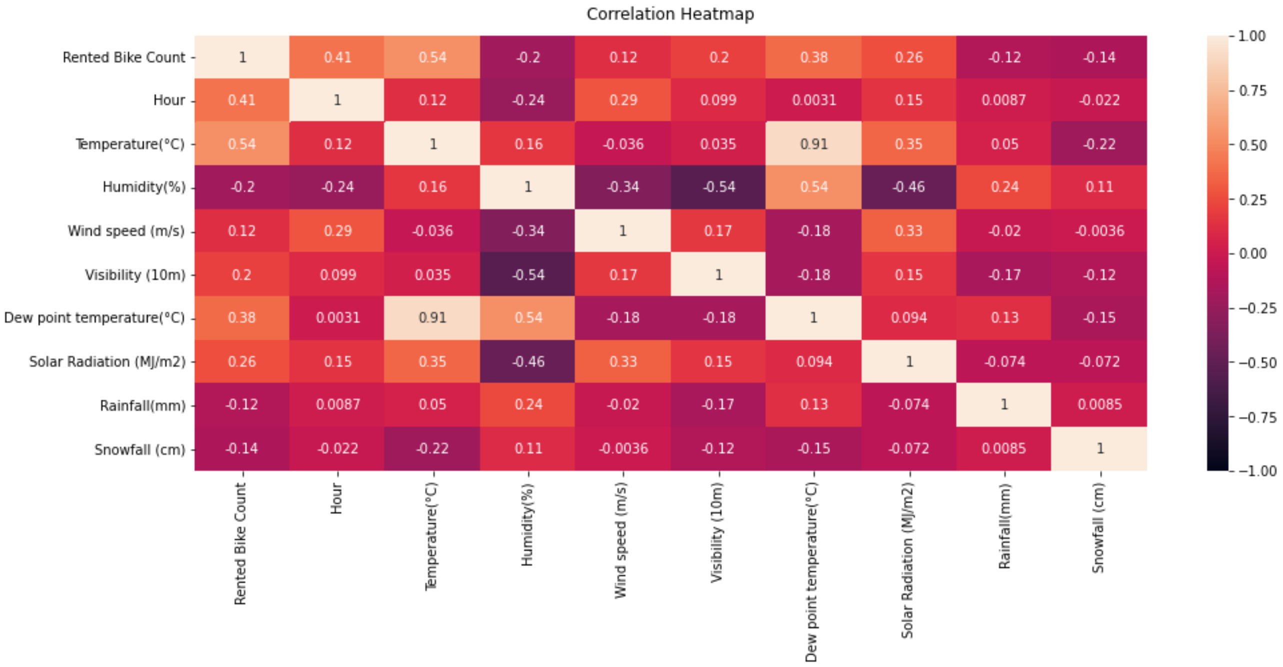


Fig.5: Heatmap of the pairwise correlation coefficients

The heatmap in *Figure 5* confirms the very strong positive correlation between Temperature and Dew point temperature, in which the only difference is very likely the Dew Point Temperature variation throughout the season of the year. As a consequence, having both these predictors doesn’t carry additional information and one of them could be dropped.

Another possible correlation is highlighted, Humidity and Visibility, which is understandable, but the numbers suggest that it’s only a low-moderate negative linear correlation and therefore it’s not obvious, nor needed to drop one of those two attributes.

3.4 Outliers analysis

This last exploratory analysis aims at exploring individually the predictors and discovering the presence of possible outliers. To get a high-level overview the boxplots are presented (i.e. a special attention is posed on the quartiles as an instrument to discover outliers).

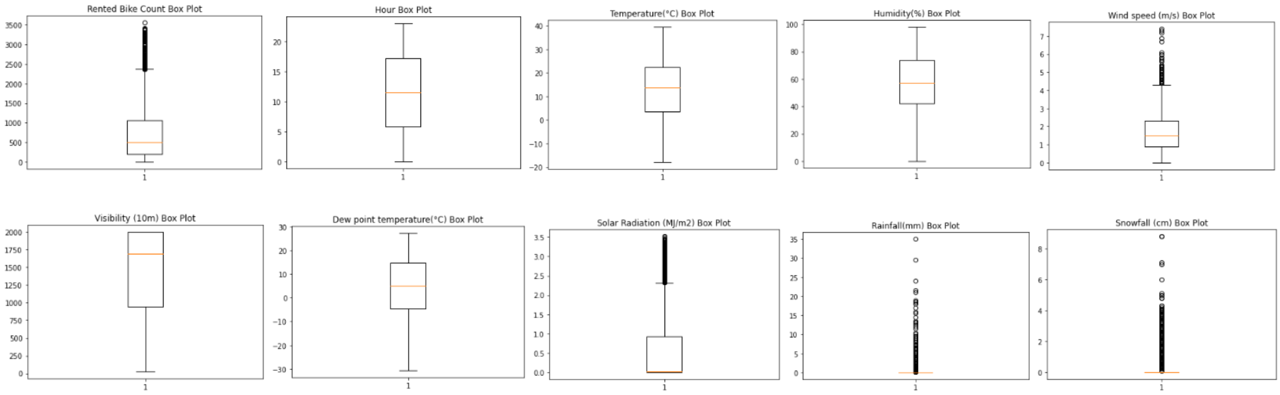


Fig.6: Boxplots of the individual attributes

At first glance there seem to be many outliers to be removed, but some considerations must be done. Rainfall and Snowfall should not be considered for this analysis since their distributions (which have already been previously discussed) tend to this kind of plot since they can be considered one-sided Gaussian centred in zero with very low variance (rare events) and their largest values are reasonable (8cm snow, 35mm rain). Solar radiation and Wind speed instead could be more related to seasonal variations in the year and their values are still reasonable and, especially, helpful determining the behaviour of the target variable under extreme weather conditions.

4. Data preprocessing

This step is the final data preparation phase before moving to model selection, training and evaluation. There are numerous standard techniques which can be applied in addition to some ad-hoc transformations required by the dataset specifically.

4.1 Attributes reworking

This subsection refers to dataset-specific operations all the transformations to be applied to the attributes like the Date which otherwise wouldn’t be useful, nor usable and also to the categorical attributes which need to be encoded in some way into numerical or binary values.

4.1.1 Date

Date is a very peculiar attribute since it seems hard to use at first glance. It contains 24 points per day (associated to hours 0 to 23) for an entire year from 1/12/2017 to 30/11/2018, therefore it’s immediately evident how the piece of information about the year is not relevant at all and can be discarded. It only concerns 1 year and then the behaviour of the data starts a new cycle, whose characteristics can be represented by day and month. The remaining part of the Date therefore is going to be encoded into a categorical attribute regarding the month and another categorical attribute concerning the day of the week, which could turn out to be an information more useful than the day of the month (in a month there aren't usually huge shifts in the overall conditions, while the day of the week capture information about volumes of users potentially using the service.

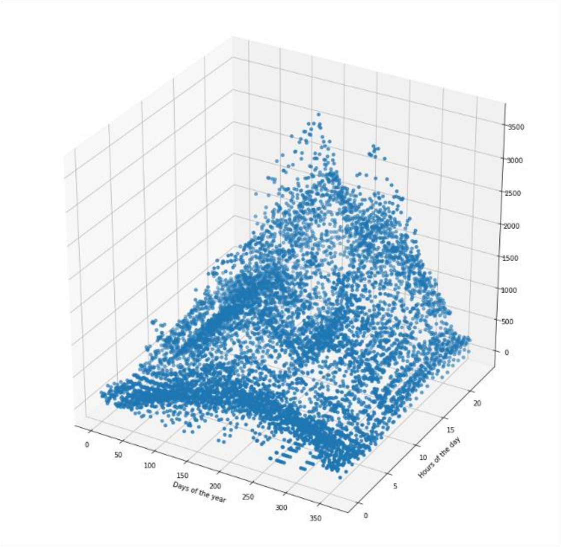


Fig.7: Target variable 3D scatterplot versus the days of the year (and Hours) cyclic behaviour

The newly created categorical attributes with N possible values are going to be encoded with a Dummy Variable encoding into N-1 binary variables (to avoid redundancy).

4.1.2 Seasons

Season is another categorical data with 4 possible values, therefore it can be encoded into 3 binary dummy variables (analogously as what was done for weekdays and months).

4.1.3 Holiday & Functioning Day

Holiday and Functioning Day are binary attributes therefore they are simply encoded into a binary variable with values 1 or 0.

4.2 Dimensionality reduction

A new analysis of the correlation is performed after the manipulation of the categorical data which have been transformed and encoded into new multiple binary variables and therefore need a study of the linear correlation. It seems reasonable to perform again this step since some of the newly created predictors could have an overlapping meaning with some of the existent one (e.g. months and season).

According to the analysis conducted at point 3.3 the Dew point temperature(°C) will be dropped in favour of the Temperature(°C) due to their strong positive linear correlation.

Exploring carefully the new heatmap in *Figure 8* it is understood that also other features can be dropped due to a strong linear correlation:

- Winter
- Sprint
- Summer

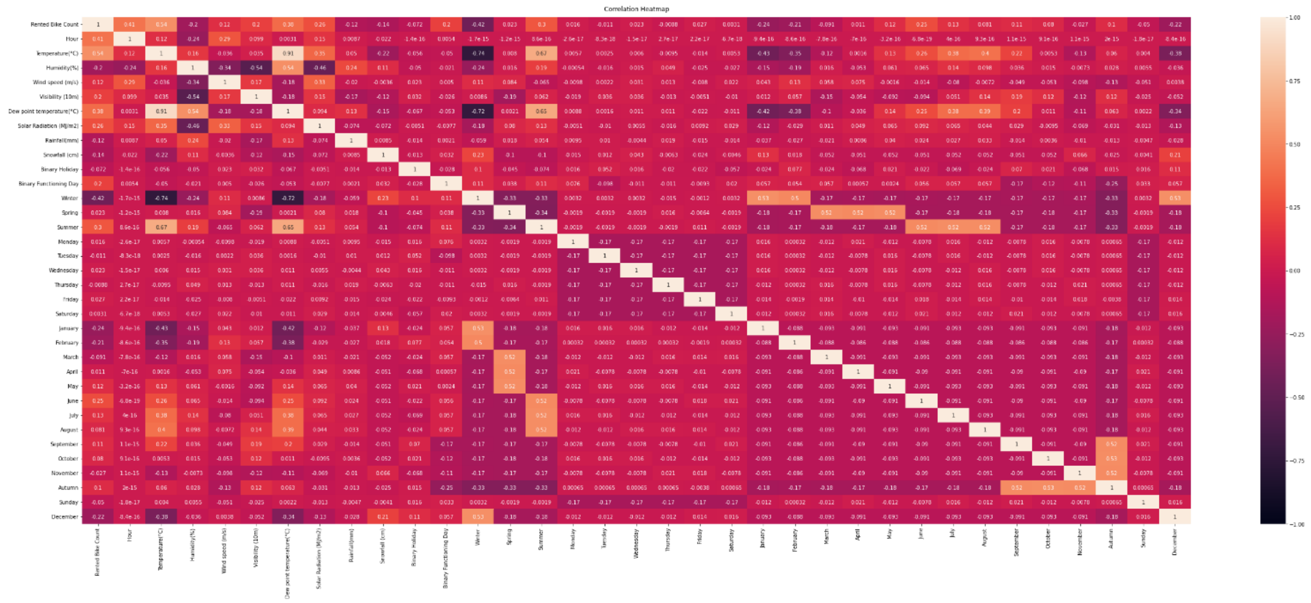


Fig.8: New heatmap of the pairwise correlation coefficients

After these transformations the overall dimensionality has been reduced, the strong correlations have been eliminated and more significant features were created. Considering the dataset reduced number of features it is not necessary to apply other dimensionality reduction techniques as PCA.

4.3 Outliers removal

An attempt to find outliers is performed on the Wind Speed and Solar Radiation attributes which are the only predictors who could contain any according to the analysis conducted at point 3.4.

The standard and optimal way to implement the outliers removal would be to apply the Inter Quantile Range (IQR) to keep only the datapoints closer to the mean within 1.5*IQR. Precisely:

$$IQR = Q3 - Q1$$

$$lower\_boud = Q1 - 1.5*IQR$$

$$upper\_bound = Q3 + 1.5*IQR$$

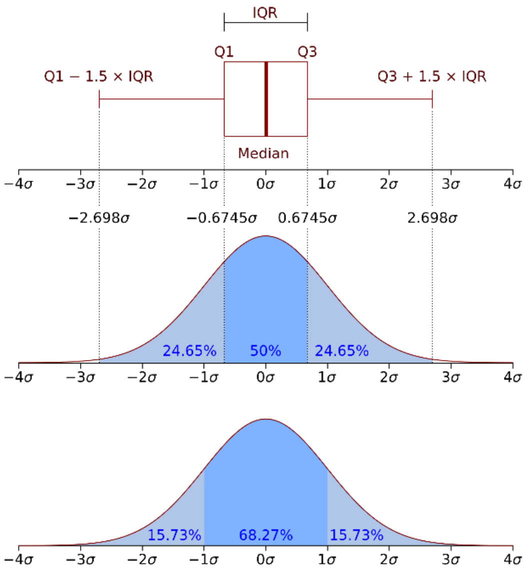


Fig.9: Inter Quantile Range compared to Gaussian Curve [3]

In the case of Solar Radiation and Wind Speed it doesn't make sense to apply this method because it would be too restrictive and, as mentioned at point 3.4 the extreme values could be useful to understand behaviour under extreme but plausible conditions. To this purpose only a filtering on the upper bound has been applied discarding values larger than the 99th percentile. A total of 185 entries has been dropped (out of the initial 8760).

4.4 Standardization

The very last preprocessing step applied to this dataset is a standardization of the values which is needed to prevent attributes that would otherwise have different scales to contribute differently to the final goal of predicting the target value. Standardization is not the only existing transformation of this kind bringing to the desired result, but it's one of the most common and it's been selected for this work. For each attribute the following transformation is applied:

$$z = \frac{x - \mu}{\sigma}$$

$$\mu = \text{Mean}$$
$$\sigma = \text{Standard Deviation}$$

Fig.10: Standardization procedure

After the transformation is possible to compare the new boxplots of all the attributes:

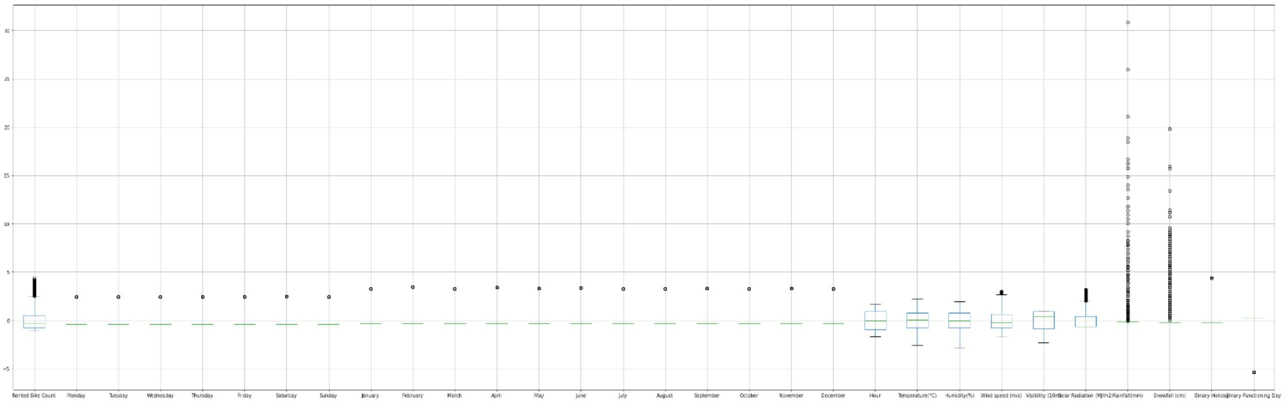


Fig.11: Boxplots of all the standardized attributes

There are still the purposedly accepted values of Wind Speed and Solar Radiation, but now all the scales are comparable and the data are ready to be used.

5.Models, metrics and results

Several regression models can be trained on this dataset, but the focus will be on Linear Regression, Polynomial regression and SVR. The data are going to go through a train-test split or train-validation-test split with grid earch in case of hyperparameters tuning.

The obtained models are going to be tested according two main aspects: the adjusted R^2 statistic and the analysis of the residual plots.

$$\textit{Coefficient of Determination} \rightarrow \quad R^2 = \frac{SSR}{SST} = 1 - \frac{SSE}{SST}$$

$$\textit{Sum of Squares Total} \rightarrow \quad SST = \sum (y - \bar{y})^2$$

$$\textit{Sum of Squares Regression} \rightarrow \quad SSR = \sum (y' - \bar{y'})^2$$

$$\textit{Sum of Squares Error} \rightarrow \quad SSE = \sum (y - y')^2$$

$$R^2_{adj} = 1 - \frac{(1 - R^2)(n - 1)}{n - p - 1}$$

$where:$

$R^2 = R - squared$

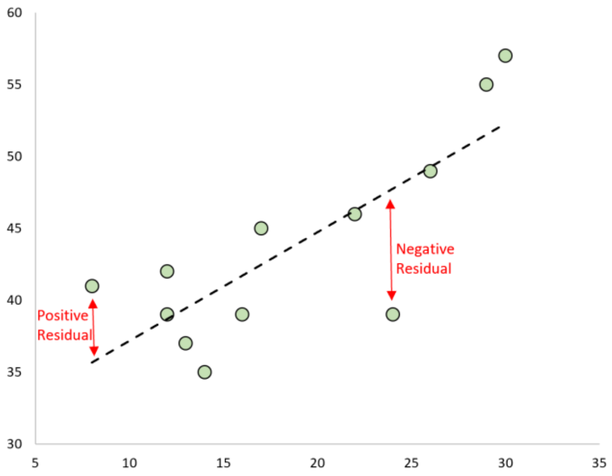
$n \quad = number\ of\ samples/rows\ in\ the\ data\ set$

$p \quad = number\ of\ predictors/features$

Fig.12: Explanation of R^2 (left [5]) and adjusted R^2 (right [6])

The R^2 statistic is the proportion of the variation in the dependent variable that is predictable by the model from the independent variables, is a value generally belonging to the range [0;1] where being closer to 1 indicates a better quality. The adjusted R^2 statistic is needed in the context of multiple linear regression (i.e. multiple predictors), otherwise the simple R^2 metric would just increase with the addition of new predictor variables.

The study of the residuals plot instead wants to observe the residuals and their distribution with respect to the dependent variable. The residuals are obtained by a pointwise subtraction of the predictions from the true values, then they are plotted against the predicted (or fitted) values.



$$\text{Residual} = \text{actual } y \text{ value} - \text{predicted } y \text{ value}$$
$$r_i = y_i - \hat{y}_i.$$

Fig.13: Residuals graphically (left [7]) and formula (right [8])

The desired result is to observe a residual plot with randomly and evenly scattered residuals without any particular shape or trend and normally distributed. In the case of evenly scattered and normally distributed residuals, it would be safe to assume that the model preserved the homoscedasticity assumption of the residuals (the residuals preserve the same variance at every level of X, the predictor(s)), otherwise the model residuals suffer from heteroscedasticity which translates into the conclusion that the model is not suitable for the data or the data require additional transformations.

5.1 Linear regression

The first attempt is to generate a model on these data is done using a Linear Regression model from Scikit Learn [4]. It doesn’t require any kind of hyperparameter tuning, therefore the data are split in two sets, train and test, with the test size being 20% of the whole dataset.

After training the model on the training set and fitting the test predictors, the values fitted to the regression line are obtained and the evaluation starts.

The obtained adjusted R^2 is 0.5943 (belonging to the range [0;1]) which isn't an excellent result.

The obtained residual plot is shown in *Figure 14* and provides two pieces of information:

- The residuals are close to being normally distributed
- The residuals are NOT evenly scattered, but they follow a trend

The conclusion is that even though the residuals are almost normally distributed, they suffer from heteroscedasticity and, therefore, this model isn't well performing on the dataset.

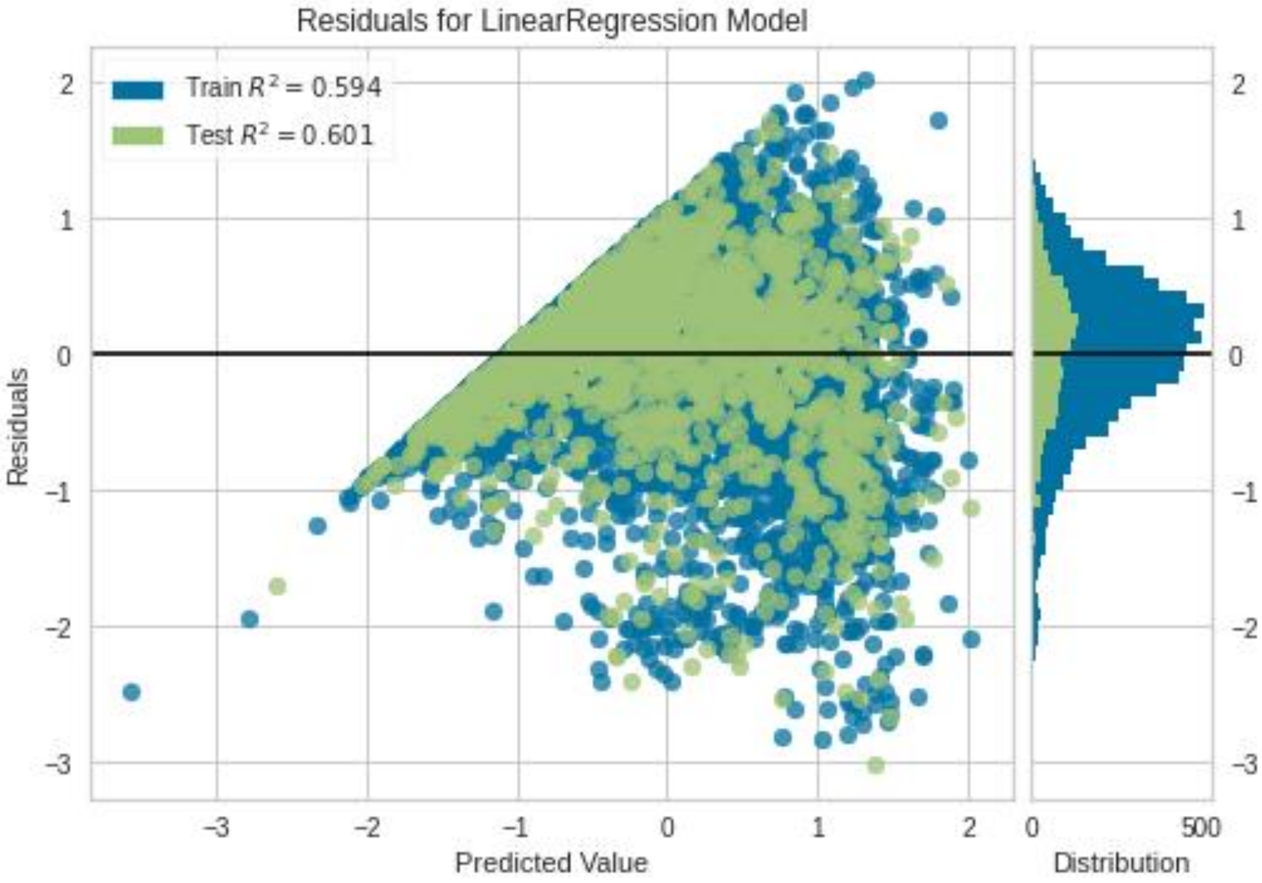


Fig.14: Residuals plot of the Linear Regression model

5.2 Polynomial regression

The second model tested is the Polynomial Regression using again the Linear Regression model from Scikit Learn, but with an additional transformation to the predictors X in order to move from the Multiple Linear Regression to the Multiple Polynomial Regression setting.

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ y_n \end{bmatrix} = \begin{bmatrix} 1 & x_1 & x_1^2 & \dots & x_1^m \\ 1 & x_2 & x_2^2 & \dots & x_2^m \\ 1 & x_3 & x_3^2 & \dots & x_3^m \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & x_n & x_n^2 & \dots & x_n^m \end{bmatrix} \begin{bmatrix} \beta_0 \\ \beta_1 \\ \beta_2 \\ \vdots \\ \beta_m \end{bmatrix} + \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \varepsilon_3 \\ \vdots \\ \varepsilon_n \end{bmatrix},$$

Fig.15 Polynomial Regression setting [9]

To transition to the Polynomial Regression setting, the predictors X must be transformed by introducing the higher degree predictors, but it is important to remark that it still falls under the Linear Regression setting in which, by definition, the linearity constraint refers to the unknown underlying parameters (Betas) and not to the predictors.

Using the Scikit Learn method PolynomialFeatures [10] it is possible to apply the correct transformation to the predictors X with the desired degree of the polynomial. In this work two values of the "degree" hyperparameter have been tested, degree=2 and degree=3, and it wouldn't have made any sense to rise it any further.

For each of the tested degrees the following steps have been carried out:

- Apply the predictors transformation adding
- Split the dataset in train and test sets, with the test size being 20% of the total data (as in 5.1)
- Train and test the model

For degree = 2, the obtained adjusted R^2 = 0.6692

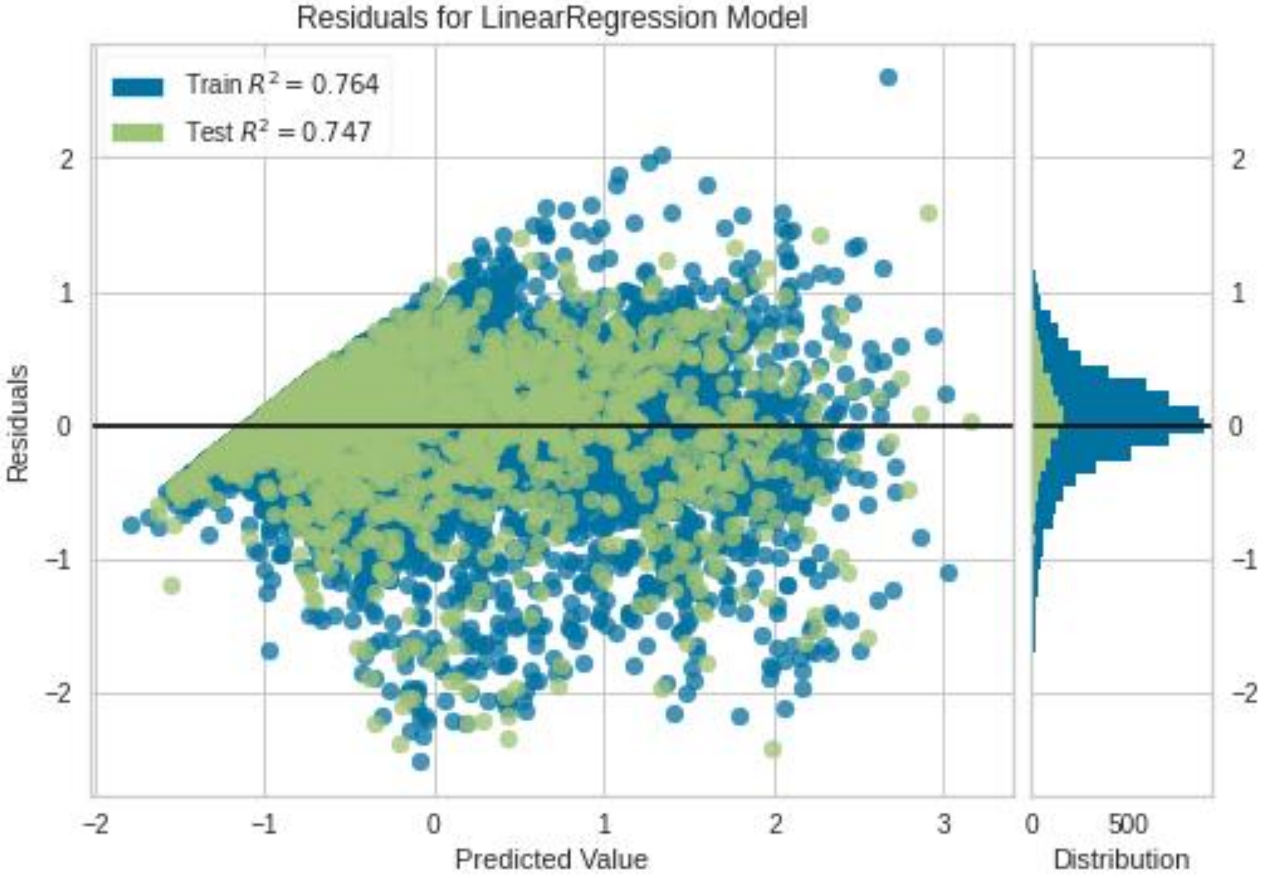


Fig.16: Residuals plot of the Polynomial Regression model with degree = 2

The obtained residuals plot is shown in *Figure 15* and provides two pieces of information:

- The residuals are normally distributed
- The residuals are sufficiently evenly scattered

The conclusion is that even though the residuals are not perfectly evenly scattered, the randomicity improved and seems to be sufficient to be acceptable. Also, they are normally distributed and the R^2 is sufficiently good.

For degree = 3, the obtained R^2 is negative (which means the prediction is worse than the average straight line) and the residuals plot in *Figure 17* is degenerate. Considering that the training phase R^2 is quite high, this case can be classified as a strong overfitting and it doesn't make sense to test higher values of the degree hyperparameter.

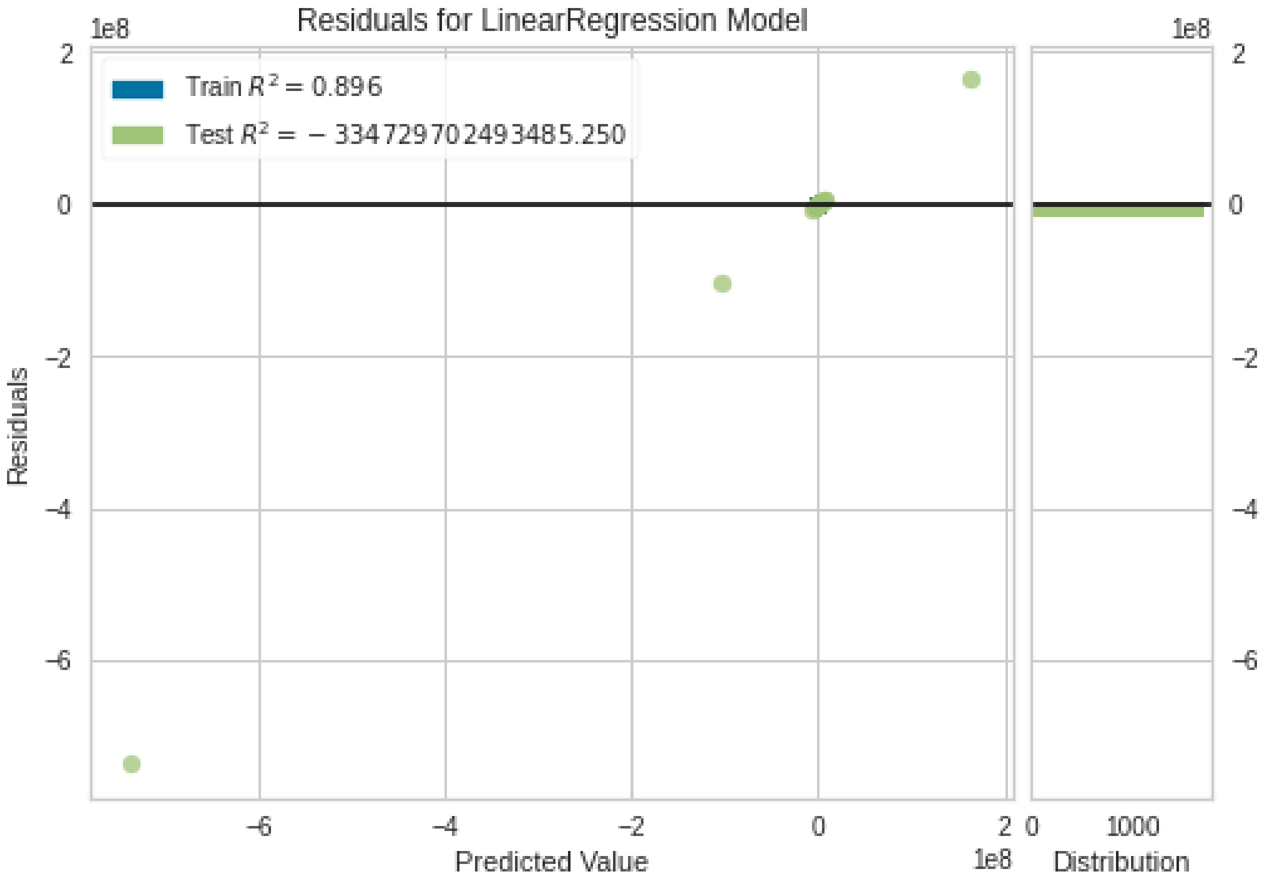


Fig.17: Residuals plot of the Polynomial Regression model with degree = 3

5.3 SVR

The last approach considered in this work exploits the theory of Support Vector Machine applied to a regression problem, namely Support Vector Regression (SVR). While Linear Regression minimizes the error between the true values and the intercepts, on the other hand the SVR tries to fit a line (or higher dimensional analogous) within a threshold of values which depends on the allowed (selected) margin of error.

The approach has been to perform a split of the dataset in train, validation and test set to correctly perform an hyperparameter tuning phase.

The dataset has been split according to the following percentages:

- Train set 84%
- Validation set 16%
- Test set 20%

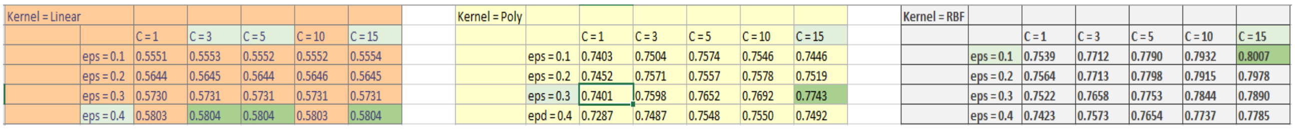
A simplified grid search has been implemented to tune the following parameters:

- Kernel: Linear, Poly, RBF
- C: 1, 3, 5, 10, 15

- Epsilon: 0.1, 0.2, 0.3, 0.4

As a consequence, a total of 60 model instances have been trained and validated computing the R^2 and adjusted R^2 statistics.

The best performing model on the validation set (adjusted $R^2 = 0.8007$), had parameters: RBF kernel, C=15, epsilon=0.1. For a rigorous analysis a second round of Grid Search has been conducted only on RBF kernel, with higher C values (20, 30, 50) and same epsilon range. It appears to be risky to exceed with the value of C which could cause overfitting, therefore the final model which has been selected and tested on the test set had parameters: RBF kernel, C=30, epsilon=0.1 with a final adjusted $R^2 = 0.8187$. For higher values of C (from 50 upwards), the results improvements on the validation set were not reflected on the test set which could be interpreted a first sign of overfitting.



Kernel = Linear						
		C = 1	C = 3	C = 5	C = 10	C = 15
	eps = 0.1	0.5551	0.5553	0.5552	0.5552	0.5554
	eps = 0.2	0.5644	0.5645	0.5644	0.5646	0.5645
	eps = 0.3	0.5730	0.5731	0.5731	0.5731	0.5731
	eps = 0.4	0.5803	0.5804	0.5804	0.5803	0.5804

Kernel = Poly						
		C = 1	C = 3	C = 5	C = 10	C = 15
	eps = 0.1	0.7403	0.7504	0.7574	0.7546	0.7446
	eps = 0.2	0.7452	0.7571	0.7557	0.7578	0.7519
	eps = 0.3	0.7401	0.7598	0.7652	0.7692	0.7743
	epd = 0.4	0.7287	0.7487	0.7548	0.7550	0.7492

Kernel = RBF						
		C = 1	C = 3	C = 5	C = 10	C = 15
	eps = 0.1	0.7539	0.7712	0.7790	0.7932	0.8007
	eps = 0.2	0.7564	0.7713	0.7798	0.7915	0.7978
	eps = 0.3	0.7522	0.7658	0.7753	0.7844	0.7890
	eps = 0.4	0.7423	0.7573	0.7654	0.7737	0.7785

Fig.18: Grid search results with adjusted R^2 statistic as evaluation metric

6. Conclusions

The goal of this analysis was to correctly study, process and use the Seoul Bike Sharing Demand Dataset to train models to be carefully evaluated, with a special focus on the methodologies of statistical analysis and machine learning presented during the course. This work results put much emphasis on the special effort and skills needed in the preliminary phases of the pipeline, well before the model training and testing which is of course a fundamental phase to be carried out with the maximum care, but nonetheless any effort would be worthless without a proper analysis and preprocessing of the dataset of interest. Considering the main subject of this analysis being the regression problem, it appears to be clear that after a considerable effort under the model evaluation phase and the techniques available to do such an operation, it is still not straightforward to asses the quality and validity of a Linear Regression Model. In terms of models' performances, the SVR had the best results, but overall were comparable with the Polynomial Regression of second degree. SVR would probably still be the final selected solution thanks to its versatility in terms of hyperparameters tuning which could theoretically leave room for even better results rather than depending only on the dataset itself and the applied transformations.

7. References

[1]: http://archive.ics.uci.edu/ml/datasets/Seoul+Bike+Sharing+Demand#

[2]: https://github.com/MatteGarba/Mathematics-in-Machine-Learning/tree/main

[3]: https://en.wikipedia.org/wiki/Interquartile_range

[4]: https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LinearRegression.html

[5]: https://www.saedsayad.com/mlr.htm

[6]: https://towardsdatascience.com/demystifying-r-squared-and-adjusted-r-squared-52903c006a60

[7]: https://www.statology.org/residuals/

[8]: https://www.ncl.ac.uk/webtemplate/ask-assets/external/maths-resources/statistics/regression-and-correlation/residuals.html#:~:text=%E2%88%92%5Eyi.-,Residual%20%3D%20actual%20y%20value%20%E2%88%92%20predicted%20y%20value%20%2C%20r%20i,minimise%20the%20sum%20of%20residuals.

[9]: https://en.wikipedia.org/wiki/Polynomial_regression

[10]: https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.PolynomialFeatures.html