Third homework of TOML

# Introduction

The third project consists in the calibration of an air pollution sersor in a network. Precisely, this sensor measure these data:

* The amount of ozone in Kohm;
* The temperature in Celsius;
* The percentage of relative humidity;
* The amount of nitrogen dioxide;
* The amount of nitrogen monoxide;
* The amount of sulfur dioxide;
* The amount of PM10.

Moreover we have the measurements relative to the O3 concentration in ugr/m^3, this is the Reference station, the value we have to predict for the calibration.

To perform our purpose, we used different machine learning models in order to see which model fits better.

# What we used

To perform this project, we used the followings Python libraries:

* Pandas: a library for data manipulation;
* Scikit learn: a library containing many machine learning models in order to perform our project.

# Data Analysis

Before calibrating the sensor, we analyse the data to verify if there is correlation between the reference station value and the measured ones.

## Reading the data

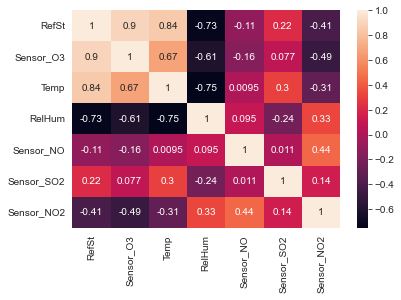
To read the data, we implemented the functions sensorData() and prepareData(), they allow us to read each csv files, make a unique dataframe and add some columns in order to simplify the plotting.

Immagine che contiene testo

Descrizione generata automaticamente Immagine che contiene testo

Descrizione generata automaticamente

## The correlation matrix

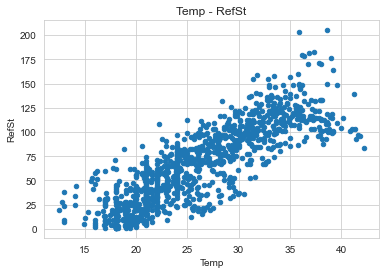
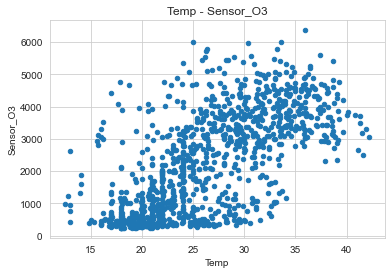


What we can conclude from the correlation matrix above? We can observe that the best features with the highest correlation are Sensor\_O3, Temp, ReHum and Sensor\_NO2, so these features are the one which resume better the value of the reference station. This information will be useful after in this report when we’ll do the feature selection.

But to avoid fast conclusions, let’s go to analyze the plots.

## Plots

### The temperature plots

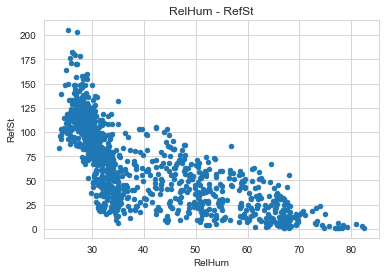
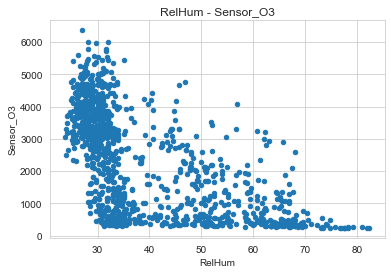
 

The two images on the tops are the plots about the temperature and the value of the O3/Reference station.

As we can see, the shapes in the plots is a little similar, but in the first one is more defined. Always in the first plot, is clearly visible that there is a linear evolution of the data, a thing also visible in the O3 plot but with some noise.

So, we can conclude that the temperature is one of the features that allow to compute the RefSt value in the linear regression.

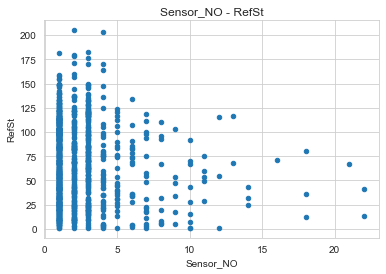
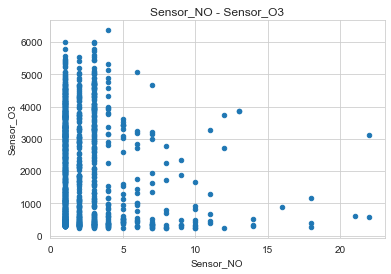
### Humidity plots

Also in this case, we can see that the plots above have more or less the same shape, with more sparsity in the second one. We remember that the humidity is the best negative correlation for RefSt in the negative matrix, this is visible in the plot because the data doesn’t follow any evolution and are concentrated in a region of the plot (in this case when the umidity is around 30). On the other hand, these is another data concentration in the second plot: it is in the low part, where the value of O3 is less than 1000. So, the humidity can be resumed in a hyperbole in the last plot.

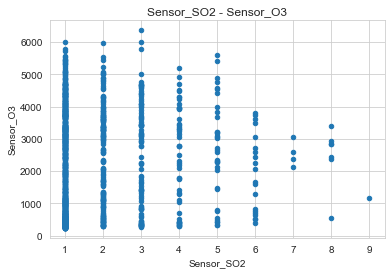
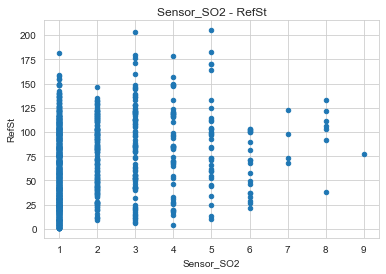
To make some conclusions, Surely the humidity doesn’t impact significantly in the linear regression, but it could inpact in the anothers. We’ll see it later.

### Nitrogen monoxide plots

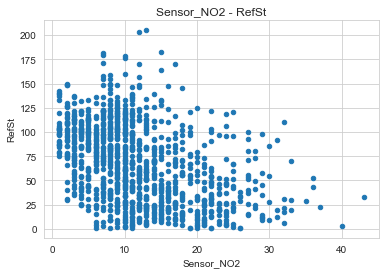
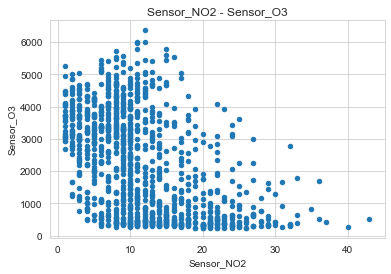
In these plots the data are concentrated in the first values. So, we can make more or less the same conclusions: the NO will impact less in a linear regression and probably it impact more or less at the same way with other algorithms because the correlation is very near to zero respect to the others.

### Sulfur dioxide plots



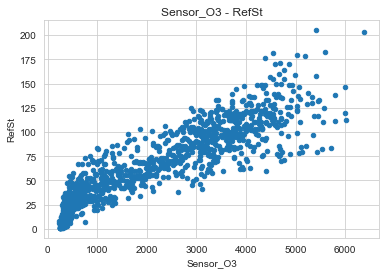
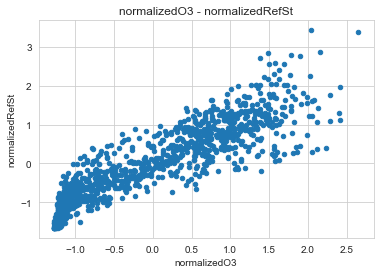
In this situation, we can make the same make the same conclusions of the previous case: because of this feature as a low correlation with RefSt and the plots has not a defined shape, The SO2 is not a good estimator of RefSt.

### Nitrogen Dioxide plots

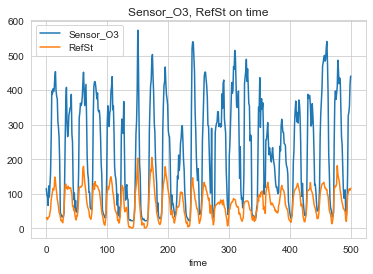
 

This is one of the features we chose from the correlation matrix, we can confirm it because the two plots above present similar shapes. The evolution is not linear, so this feature could be relevant in other algorithms.

### O3 plots

Plotting the O3 values respect to the reference station allows us to see the linear evolution of the last feature. This is a characteristic of an high correlation and we can confirm it with a value of 0.9 on the correlation matrix. As we can see in the second plot, the data normalization



Another confirmation is the plot above, in fact it is clearly visible how the O3 and the reference station follow the same evolution.

NOTE: In order to see this similarity more clearly, the O3 data has been scaled of a factor 50, but it does not affect the evolution! Also we considered only half of the data, but the conclusions we did count for the whole set.

In order to make conclusions, Sensor\_O3 is the best feature to summarize the value of the reference station!

So, for summarizing everything:

* Sensor\_O3 and Temp are the best features!
* Sensor\_NO2 and RelHum are good features!
* Sensor\_NO and Sensor SO2 are bad features.

In the phase of calibration, we considered for each model a 70% of data for the training set and the 30% for the test set.

# Models

For this project we considered the following machine learning algorithms:

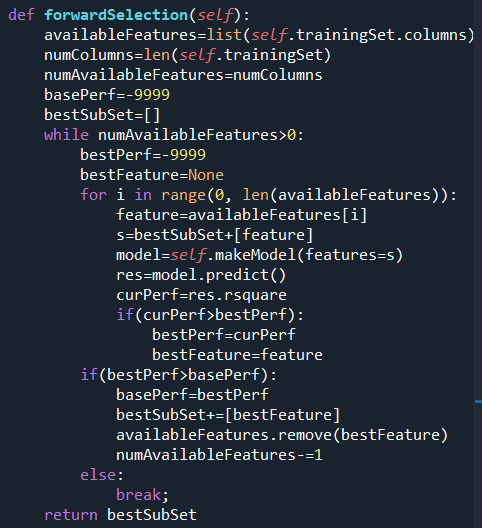
* Normal Linear regression with subsset selection:
* Lasso Linear Regression;
* Ridge Linear Regression;
* Kernel Regression;
* KNN Regression;
* Random Forest Regression;
* Support Vector Regression.

The implementation in Python consists in two general classes:

* The class Algorithm is an abstract class where we implemented many general methods that are useful for the training and the predictions;
* The class Model is an abstract class that train the model and make the predictions.

The class Algorithm presents the following methods:

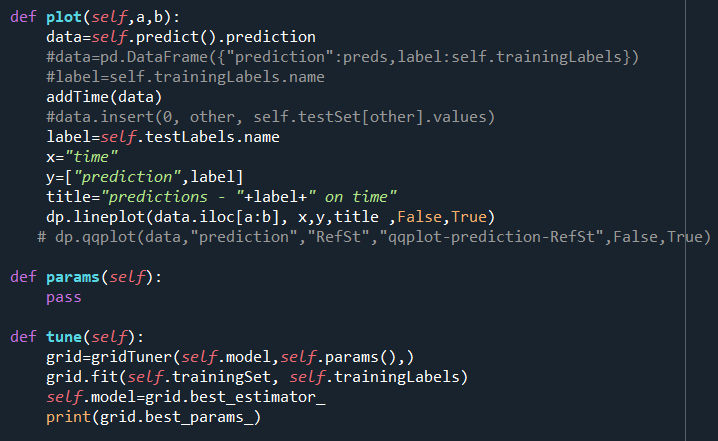
* makeModel(): it creates the model in base of the parameter of the algorithm. Moreover we can specify the features of our interest as parameters. This is an abstract method, so each class extending Algorithm will implement it;
* forwardSelection() analyzes each subset of features and returns the one maximizing the R2 and minimizing the RMSE. This method is useful because it allows us to make the most accurate predictions without falling in the overfitting.



The class Model implements the following methods:

* getCoefficients(): it returns the coefficients of the models;
* getIntercept(): it returns the intercept, the coefficient accrossing the axe y;
* redefineSet(): it adapts the training set and the test set on the features of our interest;
* plot(): make a lineplot in order to compare the predictions with the reference station values;
* tune(): it do the tuning of the hyperparameters in order to increase the accuracy of the model;
* predict(): it makes the predictions basing on the test set and compute the R2, the RMSE and the MAE.

Immagine che contiene testo

Descrizione generata automaticamente

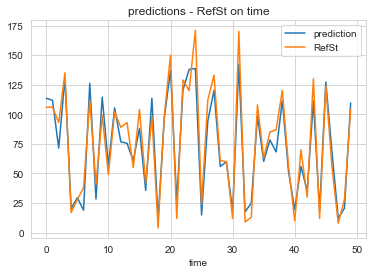
## Linear Regression

The first model we applied is the linear regression but, before doing it, we did a forward subset selection i order to avoid the overfitting. In the data analysis we said that O3 and temperature are the best features of this model, but the forward selection algorithm we implemented returns five features: O3, the temperature, SO2, NO and NO2.

This subset of features allow us to get the best R2 and a reduction of the dataframe by two columns.

## Results

|  |  |  |
| --- | --- | --- |
| R2 | RMSE | MAE |
| 0.917 | 11.636 | 9.026 |



Considering the table and the plot above, we can conclude this is a very nice model! The subset selection allowed us to get more or less the 92% or R2, thing that is clearly visible in the plot because the prediction fits very well with the reference station values.

### Coefficients

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| B0 (Intercept) | B1 (O3) | B2 (Temp) | B3 (SO2) | B4 (NO) | B5 (NO2) |
| -42.306 | 0.175 | 2.738 | 1.06 | -0.45 | 0.178 |

## Lasso Linear Regression

Now we apply always the linear regression but we do a Lasso regularization instead of using the subset selection.

For the definition of Lasso and Ridge regression (we’ll see this one after), we have a parameter alpha working a a weight for the norm of the vector of coefficients.

This is an hyperparameter of the model, so to improve the performance we have to do the tuning.

### Tuning

The tuning is a way to improve the performance of a model changing its hyperparameters. In order to search the best value of alpha, we have tried all the values we can see in the image below.

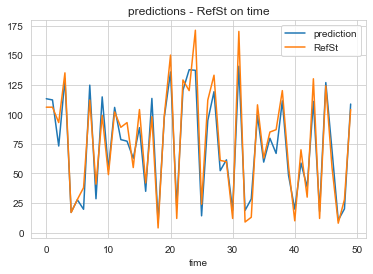
Immagine che contiene testo

Descrizione generata automaticamente

From this set of values, we keep the one giving us the best performances, in case of the Lasso, the best alpha is 0.000001.

### Results

|  |  |  |
| --- | --- | --- |
| R2 | RMSE | MAE |
| 0.917 | 11.66 | 9.044 |



Also in this case we got a a model fitting very well. Comparing these results with the normal linear regression, we can see that this one is less precise but it fits a little better in some parts, in any case these differences are not significative.

### Coefficients

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| B0 | B1 | B2 | B3 | B4 | B5 | B6 |
| -24.27 | 0.17 | 2.426 | -0.232 | -0.384 | 0.953 | 0.192 |

## Ridge Linear Regression

The last linear regression we’ll use is the one which regularize using the Ridge. Also in this case we have an hyperparameter alpha that works exactly at the same way of the Lasso.

### Tuning

We used the same set of values to find the best value of alpha improving the performances of the model. Because of the ridge regression use a two norm, the best value of alpha will be different from the Lasso, in fact it is 100 instead of 0.000001.

### Results

|  |  |  |
| --- | --- | --- |
| R2 | RMSE | MAE |
| 0.917 | 11.67 | 9.045 |

### 

To make some conclusion about the ridge regression, we can say more or less the same things about the Lasso because the results are equivalent! In fact the value of R2, RMSE and MAE differ very little and the plots does not present differences.

So, to conclude this part about the linear regression, we can say that all the models we implemented could be used without problems for the calibration of the sensor, in fact the result are very similar and each one fit almost perfectly the data. Momentaneosly, the linear regression is the best way to calibrate the sensor but let’s go to analyze the behaviour of other models.

### Coefficients

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| B0 | B1 | B2 | B3 | B4 | B5 | B6 |
| -23.58 | 0.172 | 2.41 | -0.237 | -0.372 | 0.89 | 0.19 |

## KNN Regression

Now we go to apply the KNN regression, a model predicting the values finding the best K nearest point to the input. Also in this case we use the forward selection algorithm to choose a subset of feature to use in the prediction. For this algorithm, the best subset is formed by O3, the temperature, SO2 and NO, curiously they are the feature with the highest correlation!

KNN models are very different from the linear ones, in fact we have some hyperparameters:

* the number of neighbors we have to compute;
* the weight function indicate how are weighted the points. KNN can weight the points in a uniform way or using a distance function (so, the nearest points will be more inflent), but we can also use a user-defined function;
* the size of a leaf;
* the power parameter of the Minkowski metric p, in base of the values the model will use a different distance function;

### Tuning

We did the tuning of the hyperparameter using the function below, from this we found the following best values:

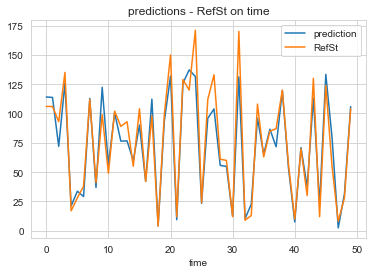
* a leaf size of 10;
* a number of neighbors to compute equals to 10;
* a p equals to 1, so the model will use the Manhattan distance;
* the use of a distance function, so the weights will not be uniform.

Immagine che contiene testo

Descrizione generata automaticamente

### Results

|  |  |  |
| --- | --- | --- |
| R2 | RMSE | MAE |
| 0.9004 | 12.761 | 9.301 |



Thank to the feature selection and the tuning, we got a model fitting very well the data. The accuracy is clearly visible in the plot above, in fact the predictions and the reference station values follows the same evolution with differences in some parts that are not a problem in any case. Comparing these results with the ones we got with the linear models, we can see that this model is less precise, in any case this difference is very little, so it doesn’t matter.

To conclude this part, we can say the linear models remains the best for the sensor calibration, but the KNN one give results very similar, so it is right too to accomplish this purpose.

## Kernel Regression

### Tuning

### Plots

## Random forest regression

### Tuning

### Plots

## Support vector regression

### Tuning

### Plots

Fourth homework of TOML

# Introduction

# Case of overfitting

## Plot

# Case of underfitting

## Plot

# Right values for the model

## Plot

# Problems

## Plot