Themes on machine learning (TOML)

Third project

Contenido

[Introduction 2](#_Toc106556364)

[Data observation and first graphs 2](#_Toc106556365)

[Modeling and calibration 5](#_Toc106556366)

[Forward Subset Selection (MLR- FSS) 5](#_Toc106556367)

[Multiple Linear Regression (MLR) with regularization 5](#_Toc106556368)

[Ridge Regression (RR) 5](#_Toc106556369)

[Lasso Regression (LR) 5](#_Toc106556370)

[K-nearest neighbor (KNN) 5](#_Toc106556371)

[Kernel Ridge Regression with RBF kernel (KRR-RBF) 6](#_Toc106556372)

[Random Forest (RF) 6](#_Toc106556373)

[Support Vector Regression (SVR) 6](#_Toc106556374)

[Comparison and conclusions 6](#_Toc106556375)

# Introduction

The main goal of this project is to calibrate an air pollution sensor in an air pollution monitoring sensor network. To do this the data obtained by this network is modeled using different methods. The results will be compared using graphs and metrics and one of them will be selected as the best method in the conclusions part of the report.

The distribution of the report is as follows: first the sensor data files and data is explained and analyzed using graphs, then several modeling techniques are applied to this data and, finally, the conclusion of which is the best modeling technique is made using graphs and metrics to compare in the last section.

# Data observation and first graphs

At the beginning various data files must be merged into a single data file to enable the modeling in the next steps. The final merged file has the following columns containing data from different sensors:

* **date:** This column has the timestamp of the data.
* **RefSt:** This column has the O3 value of the reference station. The value is in µg/m³.
* **Sensor\_O3:** This column has the data of the O3 sensor in KOhms.
* **Temp:** This column has the temperature data in ºC.
* **RelHum:** This column has the relative humidity data in %.
* **Sensor\_NO:** NO sensor data.
* **Sensor\_NO2:** NO2 sensor data.
* **Sensor\_SO2:** SO2 sensor data.

First, let us compare the plots of the reference station data and the O3 sensor data. The magnitude should be different, but the shape should be the same since the data is the same and is taken in the same periods/intervals of time.

Gráfico, Gráfico de líneas

Descripción generada automáticamenteGráfico, Gráfico de líneas

Descripción generada automáticamente

The shape of both graphs is the same. It has slight differences but overall, it has the same shape. The next step is to plot the relationship between both parameters. As mentioned before, both parameters are the same kind of data but taken by different sensors at the same time and with different units, thus, the relationship should give us a scatter plot with a 45 degrees shape approximately.

Gráfico, Gráfico de dispersión

Descripción generada automáticamente

The graph confirms that the data of both sensors is related and has the same kind of variations. Next scatter plots show the relationship between the Sensor\_O3 data end all the other features to be in the model.

Gráfico, Gráfico de dispersión

Descripción generada automáticamenteGráfico, Gráfico de dispersión

Descripción generada automáticamenteGráfico, Gráfico de dispersión

Descripción generada automáticamenteGráfico, Gráfico de dispersión

Descripción generada automáticamente

Gráfico

Descripción generada automáticamente

The same is done for the RefSt data. Next plots show the relationship of the reference station data with all the other parameters in the data file.

Gráfico, Gráfico de dispersión

Descripción generada automáticamenteGráfico, Gráfico de dispersión

Descripción generada automáticamente

Gráfico, Gráfico de dispersión

Descripción generada automáticamenteGráfico, Gráfico de dispersión

Descripción generada automáticamente

Gráfico, Gráfico de dispersión

Descripción generada automáticamente

In both groups of plots (for the O3 sensor and for the reference station sensor), the temperature data is the one that has the most visible pattern. The next feature that shows a relationship pattern is the RelHum data. All the other variables do not have a distinctive pattern with the dependent variables of the model (Sensor\_O3 and RefSt). This indicates that these parameters should be the most important in terms of the construction of a model. As we will see in the next sections, this is true.

For the plotting of the former graphs, the data is not normalized since the analysis done in this section is more focused in the shape rather than in the magnitude or the value of the variables. For the modeling part, all the data is normalized due that is a mandatory previous step to get correct models.

# Modeling and calibration

This section has the explanation of the different methods used to calibrate the data of the sensors and the different graphs obtained from the implementations in python.

## Forward Subset Selection (MLR- FSS)

The first part of the project is picking which variables of the data in the datasets model the reference station data with the best metrics (the most significative amongst all the features). This selection is done by implementing Forward Subsect Selection in python.

Forward subset selection has two phases:

1. First, it fits all the possible models that come from the combinations of k features (independent variables of the model) out of a total of P. For example, if the data has P = 3 features it will create a model with k = 1 (3 models with a single feature), k = 2 (3 combinations of two features) and k = 3 (one model involving all features). From each k group (k = 1, k = 2 and k = 3) the best model is selected using metrics as RSS, CV error or Adjusted R-squared.
2. The last step consists of picking the best model of all the models gotten in the first step. The selection in this phase is done using the same metrics as in the first step.

## Multiple Linear Regression (MLR) with regularization

Multiple linear regression is a method to create models using two or more independent variables to predict the outcome of a dependent variable. The model is fitted using training data which adjusts the coefficients of the model to get to a point where the cost function between the model line and the training data is at its minimum. The cost function usually applied is least squares error (LS).

One of the problems with the models is that they can get complex whenever they have a lot of features (as in our case). When a model gets too complex it can suffer from overfitting, i.e., it fits the training data perfectly, but it has a huge bias with test data (hence, it has huge variance between the train and the test data). A way of solving this problem is to use a regularizing method. These methods shrink the coefficient estimates of the model to zero, so the model gets less complex (some of its less significative features disappear as they have coefficients close to 0 or equal to 0) and it avoids the risk of overfitting. Another way of saying the same is that whenever the model is penalized it becomes less sensitive to the changes in the features it has, so the ones that already had low impact on the model become inexistent (or almost inexistent) for the model. This also reduces the variance of the model without substantially increasing its bias. Ridge regression and Lasso regression are two examples of MLR with regularization methods.

### Ridge Regression (RR)

Ridge regression, also called L2 regularization, has the following formulation for N data points and P features:



The goal is to find the lambda that gets the minimum of the former formulation where RSS stands for Residual Sum of Squares. A lambda equal to zero corresponds to a simple linear regression (the penalizing term goes to zero). With greater lambda is, the less flexible the model gets, and it reduces model complexity and multicollinearity. Since great values of coefficients mean more flexibility (complexity), increasing the lambda of our model should give coefficients that get asymptotically near zero (but never equal to zero). This can be seen in the next graph where the coefficients of the model are plotted against the value of lambda (alpha in python).

Gráfico, Gráfico de líneas

Descripción generada automáticamente

Also, we should have better fits for the data with alpha values in the lower range of the 0 to 1000 interval of the plot, because higher alphas give a hipper simplified model (insensible to almost all changes in data). Next plots show the reference station data in blue and the prediction in orange for different values of alpha.

Gráfico

Descripción generada automáticamente Gráfico

Descripción generada automáticamenteGráfico

Descripción generada automáticamente

The first graph corresponds to a simple linear regression prediction. The next two graphs to a low penalization (left) and a high penalization (right). The low penalization model gets closer to a good prediction as the simple linear model. The high penalization model gets worse predictions, the amplitude of the data is smaller in comparison with the other two plots.

Next plot shows the impact of the value of lambda in the metrics of the model. What we should see is a loose of relevance of the model (R squared should go low) and an increase in the error metrics since the less flexible a model is (less features the model has), less accuracy it has when predicting data.

Gráfico, Gráfico de líneas

Descripción generada automáticamente

### Lasso Regression (LR)

Ridge regression, also called L2 regularization, has the following formulation for N data points and P features:



Diagrama

Descripción generada automáticamente

Gráfico, Gráfico de líneas

Descripción generada automáticamente

## K-nearest neighbor (KNN)

## Kernel Ridge Regression with RBF kernel (KRR-RBF)

## Random Forest (RF)

## Support Vector Regression (SVR)

# Comparison and conclusions

Lasso better than ridge because can put to 0 some coefficients (ridge cannot do that).