Prediction of CO and NOx Emission produced by Gas Turbine

Erica Delvino, 851269
Vasco D'Ubaldi, 851435
Roberto Carminati, 851017

Abstract

The objective of this analysis is to predict the amount of CO and NOx (in mg/m^3) produced by a gas turbine relative to 2011 data. Initially, we performed preliminary clustering analyses to see if data forms naturally groups through k-means, hiearchical clustering and DBSCAN.
Then we applied regression models using CO as the response variable, later we performed the same analyses for NOx. To model the regression we used K-NN, Random Forest, Support Vector Machine, Neural Network and Gaussian Processes.
The result, according to MAPE, is that Random Forest method reports a more accurate predictive ability.
Finally, by training the model on 2011 data, through Random Forest we predicted CO and NOx emissions for the following years, i.e. from 2012 to 2015.

1. Introduction

The folder of datasets used for our analysis comes from https://archive.ics.uci.edu/dataset/551/gas+turbine+co+and+nox+emission+data+set and includes 5 different datasets, each related to a specific year from 2011 to 2015. We performed the analyses for the dataset related to 2011 and used the others for forecasting purposes. The following consists of 7411 observations and 11 variables.
We used Rstudio and Knime software for the analyses.

1.1. Description of turbine operation

A gas turbine is a type of internal combustion engine that converts the energy of burning fuel into mechanical power. It operates on the principle of the Brayton cycle, which consists of four main processes: compression, combustion, expansion, and exhaust.

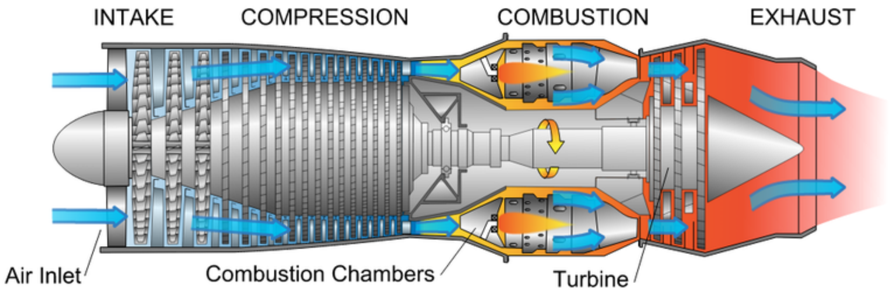


Figure 1: Principle of the Brayton cycle

- **Compression** The process begins with an air compressor, which draws in ambient air and compresses it to a high pressure. The compressor consists of multiple stages, typically using axial or centrifugal compressor blades. As the air passes through each stage, its pressure and temperature increase.

- **Combustion** The compressed air enters in the combustion chamber, where fuel is injected and ignited. The fuel can be natural gas, kerosene, or diesel, depending on the application. The burning fuel releases a high-temperature, high-pressure gas.

- **Expansion** The hot gas produced during combustion flows into the turbine section. The turbine contains a series of blades mounted on a rotor, which is connected to a shaft. As the hot gas flows over the turbine blades, it expands and its pressure drops, causing the blades and rotor to spin.

- **Exhaust** The exhaust gas exit from the turbine and pass through a diffuser, where its velocity decreases, and its pressure increases. The exhaust gas is then released into the atmosphere. The mechanical energy produced by the spinning turbine shaft can be used to drive various applications. In many cases, it is connected to a generator to produce electricity.

1.2. Description of the variables

The 11 variables available are numeric and include gas turbine parameters in addition to the ambient variables.

- **AT** Ambient temperature (in °C)
- **AP** Ambient pressure (in $mbar$)
- **AH** % of ambient humidity
- **AFDP** Air filter difference pressure (in $mbar$)
- **GTEP** Gas turbine exhaust pressure (in $mbar$)
- **TIT** Turbine inlet temperature (in °C)
- **TAT** Turbine after temperature (in °C)
- **CDP** Compressor discharge pressure (in $mbar$)
- **TEY** Turbine energy yield (in MWH)
- **CO** Carbon monoxide (in mg/m^3)
- **NOx** Nitrogen oxides (in mg/m^3)

2. Pre-Processing

In the pre-processing phase, we examined the boxplots and histograms of the variables to get an initial idea of their distribution. Having noticed that the CO distribution was particularly skewed, we decided to use Box-Cox's procedure to identify whether a transformation was desirable; what resulted was that a logarithmic transformation is the most appropriate choice. So we will perform all our analysis considering the logarithm of CO.

Another aspect that is crucial to emphasize is that the ranges of the variables are very different from each other, so we standardized the dataset in order to solve the problem of different scale magnitudes.

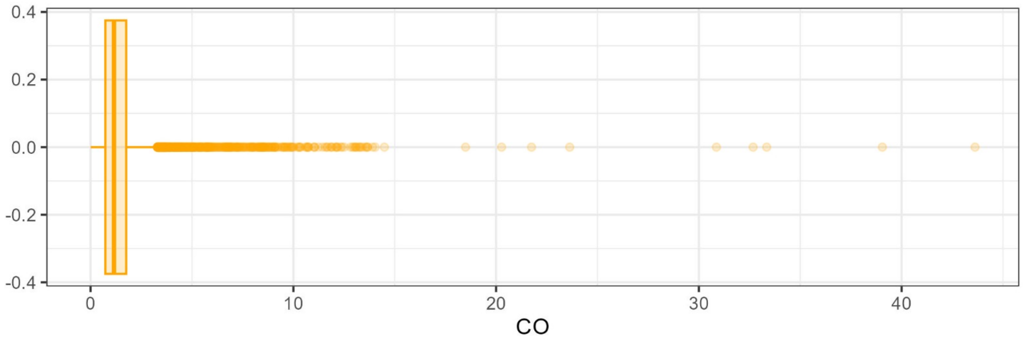


Figure 2: Boxplot of CO

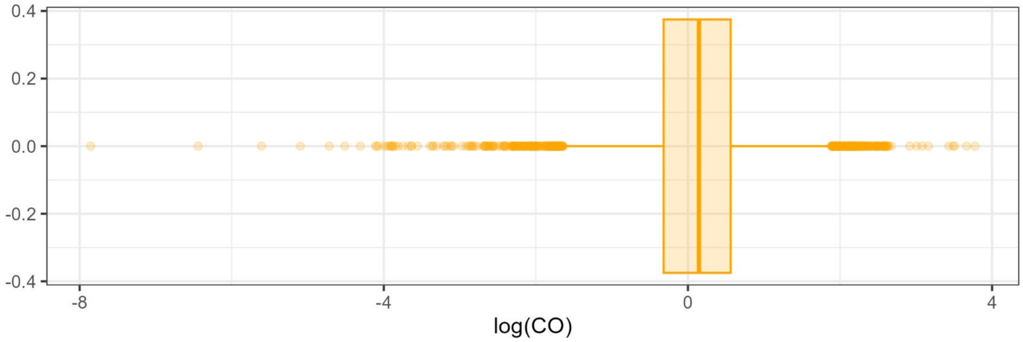


Figure 3: Boxplot of log(CO)

2.1. ***Reduction of variables***

To explore whether there were collinearity problems in this data set, we analyzed the correlation between the variables and we found out that there is a strong correlation between some pairs of them, as can be seen in figure 4.

We used Random Forest to figure out which variables are most important in this analysis, comparing both the case where the response is CO and the case where it is NOx. According to the results we decided to exclude TEY, CDP, AFDP and GTEP. Now we can carry out the analysis without recourse to high correlation problems.

2.2. ***Outliers detection***

Analyzing the boxplots of the variables, we noticed that CO, TIT, NOX, AP, TAT and AH have several outliers. To solve this problem we used Rosner's test to

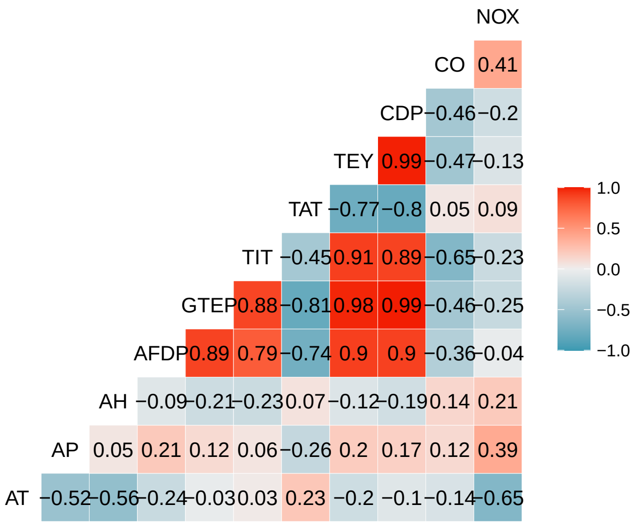


Figure 4: Correlation plot

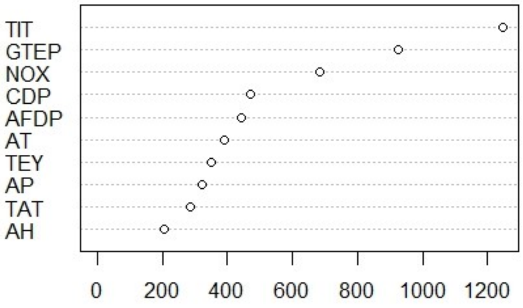


Figure 5: Variable importance plot for CO

exclude only those outliers that are either much smaller or much larger than the rest of the data. In this way we removed only 55 observations from the entire dataset.

3. Clustering

We applied various clustering algorithms to see if the data naturally split into groups, displaying in a small number of dimensions the results.

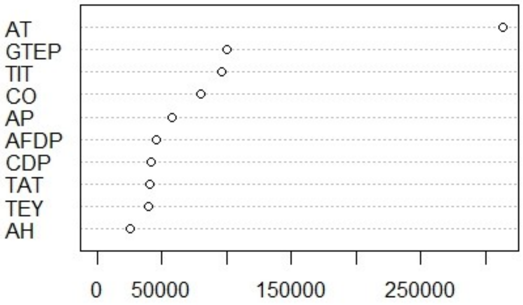


Figure 6: Variable importance plot for NOx

3.1. *K-means*

To apply k-means, we must determine the optimal value of the hyper-parameter k, i.e. the number of groups. Graphically, we choose with the elbow method $k = 4$. The results are displayed in 3-dimension in figure 7 using the variables CO, NOX and TIT.

3.2. *Hierarchical Clustering*

The second clustering method we use is agglomerative hierarchical clustering. We choose Euclidean distance and evaluate every possible linkage that can be used in order to perform the algorithm. Analyses lead us that ”complete-linkage” and also the ”Ward's-linkage” are good choices, as they do not present the chaining problem that characterises the others. Then we optimise the hyperparameter k via silouehette, choosing $k = 5$ for the ”complete-linkage” and $k = 4$ for ”Ward's method”. Comparing the results on figure 8 and figure 9, we noticed that on the ”complete-linkage” one of the 5 clusters is not very large, thus being not very dissimilar to the k obtained with ”Ward's method” and with *K-means algorithm*.

3.3. *DBSCAN*

Afterwards, we used the DBSCAN algorithm. We created a dense grid with possible combinations of values of the two hyperparameters to be optimized:

- ϵ, the length of the radius of the circle associated with each point
- τ, the density threshold

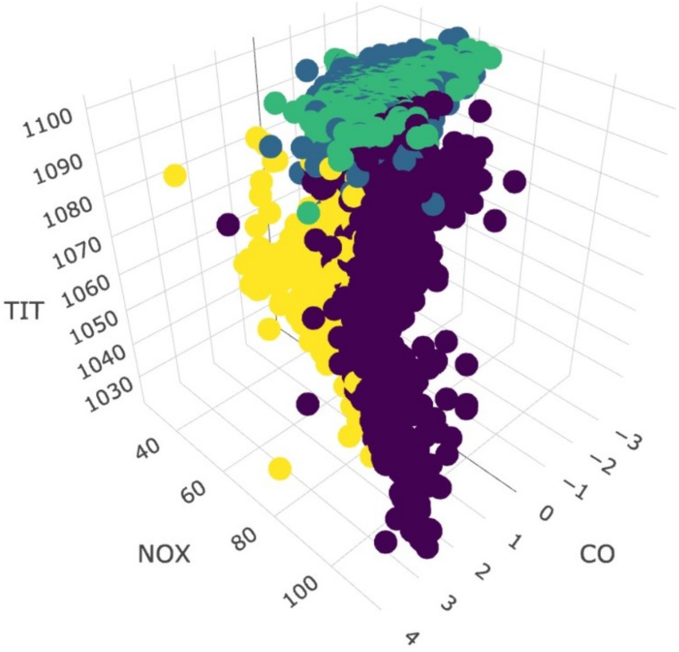


Figure 7: Cluster with best value of *k* in K-NN

The optimization led to the choice of $\epsilon = 1.2$ and $\tau = 50$ based on the highest average silhouette value. The result based on these hyperparameters bring to the creation of a single cluster consisting of 7012 units and an outliers cluster consisting of 344 units, highlighting the ineffectiveness in this context of the algorithm.

4. Regression

We carried out the analyses by applying different regression models considering initially the amount of CO emitted as the response variable and then the amount of NOx. The description of each method shows the results for both models.

In general, for all the models used, we used MSE to evaluate the optimal value of the hyperparameters within the model; while we used MAPE as a metric to compare the different models with each other.

To compare the different kind of models with each other, we arbitrarily divided the dataset into training-set and test-set in proportions of 80% and 20%, respectively.

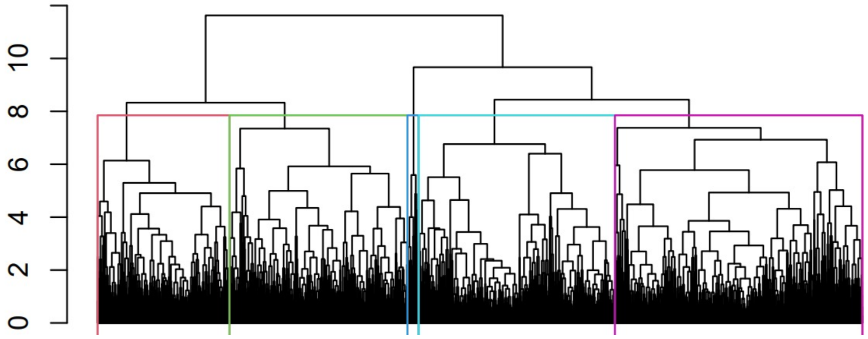


Figure 8: Hierarchical clustering with *k*=5 and complete linkage

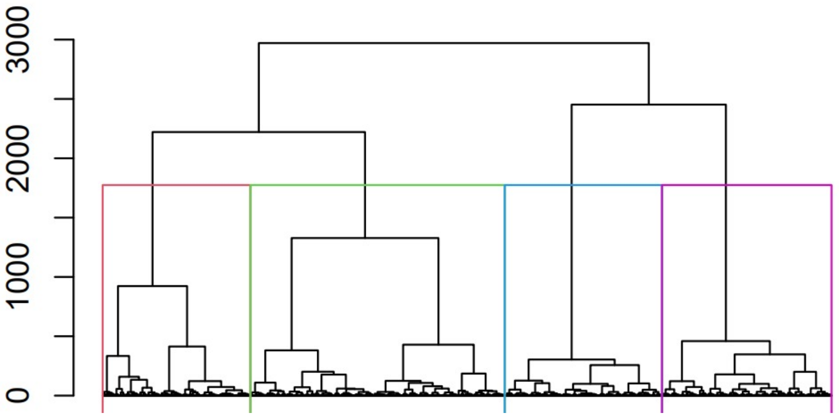


Figure 9: Hierarchical clustering with*k*=4 and Ward's method

4.1. ***K-Nearest-Neighbors***

The first regression method we applied is semi-supervised. Through a 10-fold cross validation we derived the best number of the hyperparameter *k*, which represents the number of nearest points to be considered with respect to the points for which we want to derive a prediction.
We obtained that the *k* that minimizes the MSE is equal to 7 for the model with CO response, while it is equal to 4 for the model with NOx response. The table with the performance of the two regression models with the optimal *k* is shown next.

In figure 10 are shown the results in 2-dimension using the covariate TIT.

MAPE of KNN

	CO	**NOx**
Training Set	3.4362	1.0870
Test Set	3.1556	1.5835

Table 1: Result of KNN on the training and test set

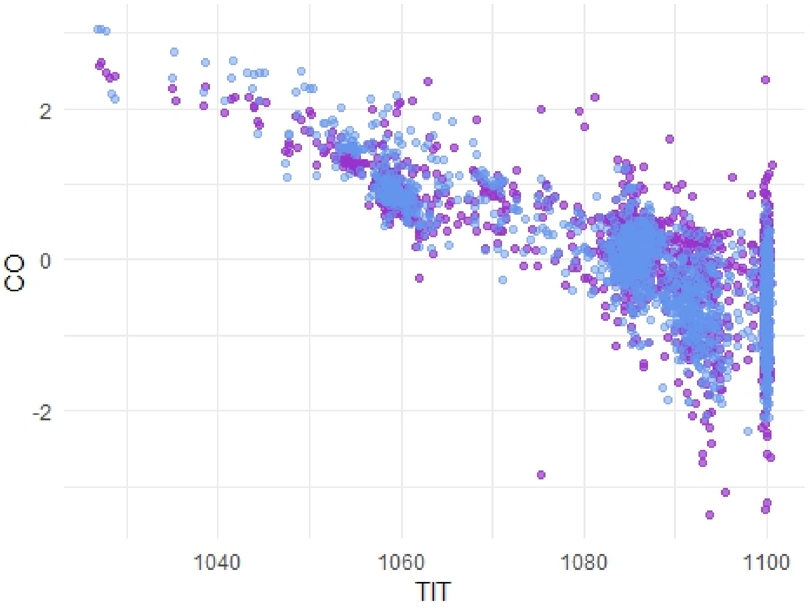


Figure 10: Prediction of CO using K-NN. The real values of CO are the blue point and the violet point are the value of the best prediction of the algorithm.

4.2. ***Random Forest***

We then evaluated the behavior of our two models by applying the ensemble method of Random Forest with 150 trees and 4 variables tried at each split, obtaining excellent results for both models.

In figure 11 and 12 are shown the results in 2-dimension for both the target values and the covariate TIT. The real values of CO are the blue point and the violet point are the value of the best prediction of the algorithm.

4.3. ***Support Vector Machine***

We then focused on modeling the Support Vector Machine algorithm for regression.

MAPE of Random Forest		
	CO	**NOx**
Training Set	0.0554	0.0015
Test Set	0.0386	0.0011

Table 2: Result of Random Forest on the training and test set

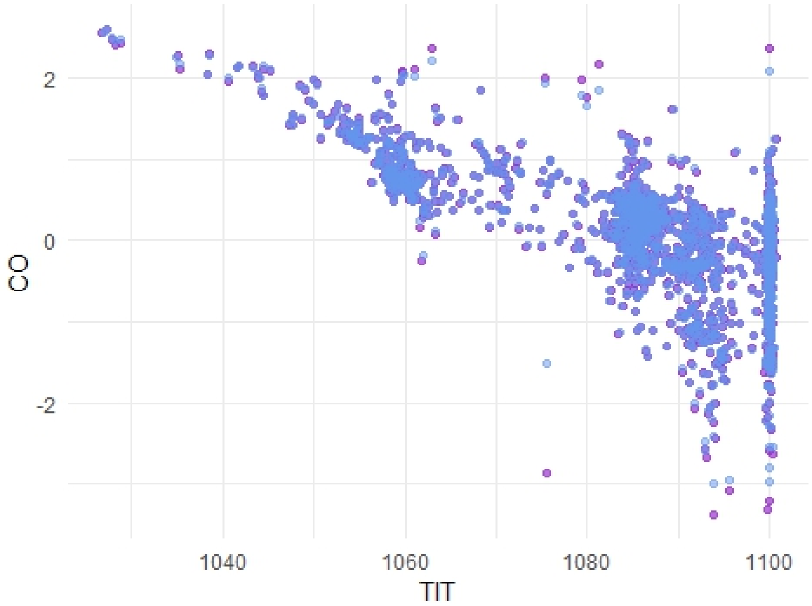


Figure 11: Prediction of CO using Random Forest. The real values of CO are the blue point and the violet point are the value of the best prediction of the algorithm.

Since the data were non-linearly separable, we adopted the "*Kernel Trick*" using the kernel function of type "*radial*". We have then tested the model on a grid of values to test the optimal combination of the two hyperparameters(i.e. those that are able to jointly minimize the MSE):

- C, the cost complexity parameter
- γ, the smoothing parameter of the kernel function

For the model with CO response, the best combination of hyperparameters was found to be with $\gamma = 1.5$ and $C = 2$; instead for the model with NOx response the best choice is with $\gamma = 1.7$ and $C = 10$.

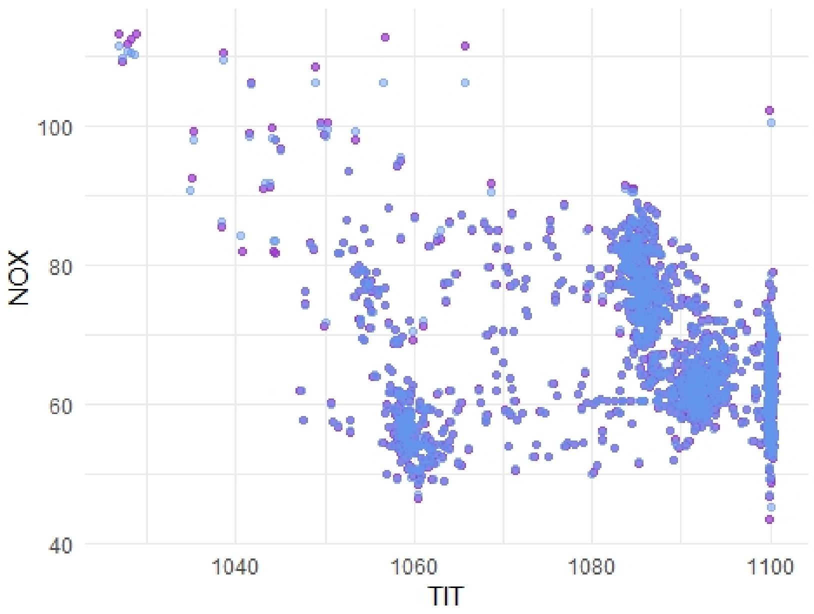


Figure 12: Prediction of NOx using Random Forest. The real values of CO are the blue point and the violet point are the value of the best prediction of the algorithm.

The performance of the SVM on the two models with the best combination of parameters is shown in the table 3.

MAPE of Support Vector Machine

	CO	**NOx**
Training Set	1.7637	0.5727
Test Set	1.7554	1.6995

Table 3: Result of Support Vector Machine on the training and test set

4.4. Gaussian Processes

Another regression method tested was Gaussian Processes, a non-parametric probabilistic method. The choice of kernel is crucial for this method, and we tried the DotProduct $k(x_i, x_j) = x_i * x_j$ and the Radial Basis Function kernel $k(x_i, x_j) = e^{-\frac{|x_i - x_j|^2}{2l^2}}$. In both cases we optimized the contained hyperparameter and the Radial Basis Function kernel gave us the best result for both NOx and CO variable. 4

MAPE of Gaussian Process

	CO	**NOx**
Training Set	2.56	1.041
Test Set	2.854	8.059

Table 4: Result of Gaussian Process regression on the training and test set

MAPE of Neural Network

	CO	**NOx**
Training Set	0.834	0.074
Test Set	0.708	2.462

Table 5: Result of Neural Network on the training and test set

4.5. Neural Network

Finally, we concluded the regression analyses by also testing the neural networks algorithm.
We optimized it by choosing the optimal number of hidden layers parameters and using ”RProp” learning rate. The results we gain for the model with CO response are 1 hidden layer containing 12 hidden neuron each and 2 hidden layers containing 12 hidden neuron for the model with NOx response.
The table 5 reports the results of Neural Network on training and test set.

5. Conclusion and forecast for the following years

Regarding clustering using Dbscan we did not get satisfactory results, as it assigned the observations to a single group. (the second one collects just the noise points).
K-means and hierachical clustering with Ward-linkage agreed in suggesting 4 groups, however this division, when displayed in 3 dimensions, does not seem to have interpretability.

Regarding the regression, the Random Forest shows the best performance with MAPE for the variables CO and NOx on the test set of 0.044 and 0.001 respectively. Another model that performs very well are neural networks, respectively returning for the variables CO and NOx MAPE values of 0.708 and 2.462. In contrast, we suspect an overfitting problem using Gaussian Processes in the NOx

MAPE of RF on next years		
	CO	**NOx**
2012	3.5317	0.0739
2013	2.7520	0.0837
2014	3.5383	0.1725
2015	2.0123	0.2549

Table 6: Result of Neural Network on the following years

variable, as the MAPE in the test set rises to 8.059 from 1.041 in the training set. Support vector machine and K-Nearest Neighbors perform well, but not at the level of Random Forest.

Now we use the best model, the Random Forest, computed now on the entire 2011 dataset as training test set, to make predictions on the data for the successful years, form 2012 to 2015. We collect great predictive ability, especially for the NOx variable. This leads us to conclude that the variables considered are able to explain and predict CO and NOx emissions very nicely even for quite far periods in time.

References

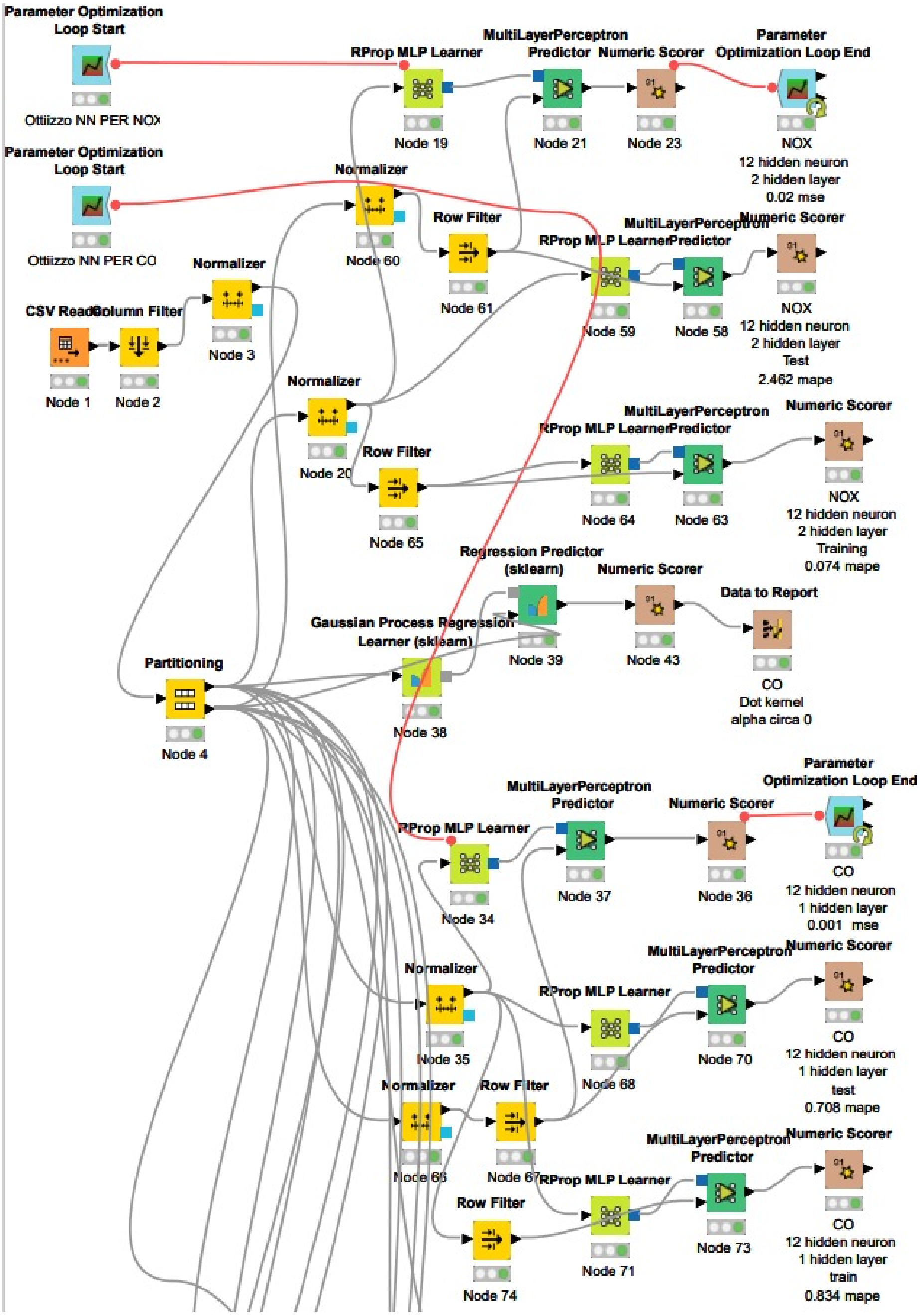
Basics of Gas Turbines, Meherwan P. Boyce (2019)

Data Mining, Charu C.Aggarwal (2015)

Gaussian Processes for Machine Learning, Carl Edward Rasmussen and Christopher K. I., Williams (2006)

A Direct Adaptive Method for Faster Backpropagation Learning: The RPROP Algorithm, Martin Riedmiller and Heinrich Braun(1993)

Appendix A. Knime work flow


Parameter Optimization Loop Start
Ottiizzo NN PER NOX
RProp MLP Learner
Node 19
MultiLayerPerceptron Predictor
Node 21
Numeric Scorer
Node 23
Parameter Optimization Loop End
NOX
12 hidden neuron
2 hidden layer
0.02 mse
Parameter Optimization Loop Start
Ottiizzo NN PER CO
Normalizer
Node 60
Row Filter
Node 61
RProp MLP Learner
Node 59
MultiLayerPerceptron Predictor
Node 58
Numeric Scorer
NOX
12 hidden neuron
2 hidden layer
Test
2.462 mape
CSV Reader
Node 1
Column Filter
Node 2
Normalizer
Node 3
Normalizer
Node 20
Row Filter
Node 65
RProp MLP Learner
Node 64
MultiLayerPerceptron Predictor
Node 63
Numeric Scorer
NOX
12 hidden neuron
2 hidden layer
Training
0.074 mape
Regression Predictor (sklearn)
Node 39
Numeric Scorer
Node 43
Data to Report
CO
Dot kernel
alpha circa 0
Gaussian Process Regression Learner (sklearn)
Node 38
Partitioning
Node 4
RProp MLP Learner
Node 34
MultiLayerPerceptron Predictor
Node 37
Numeric Scorer
Node 36
Parameter Optimization Loop End
CO
12 hidden neuron
1 hidden layer
0.001 mse
Normalizer
Node 35
RProp MLP Learner
Node 68
MultiLayerPerceptron Predictor
Node 70
Numeric Scorer
CO
12 hidden neuron
1 hidden layer
test
0.708 mape
Normalizer
Node 66
Row Filter
Node 67
RProp MLP Learner
Node 71
MultiLayerPerceptron Predictor
Node 73
Numeric Scorer
Row Filter
Node 74
CO
12 hidden neuron
1 hidden layer
train
0.834 mape

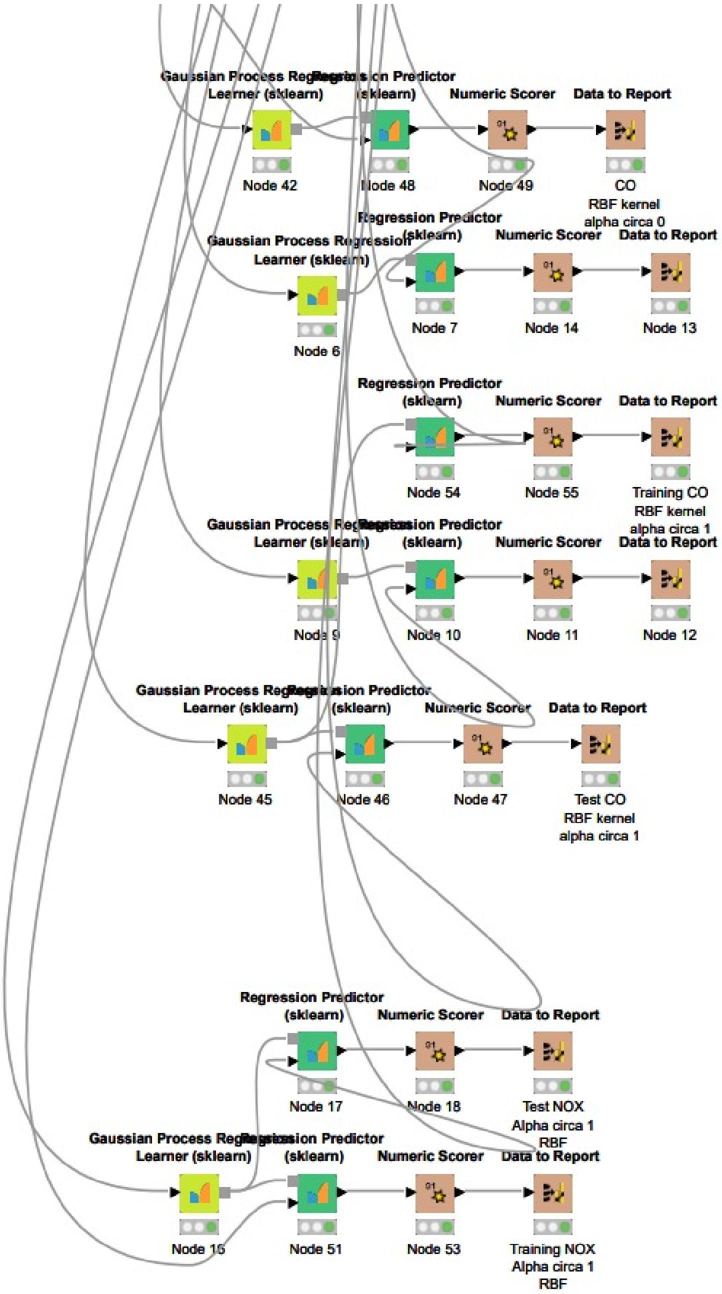


Figure A.13: Knime workflow