POLITECNICO
MILANO 1863

Politecnico di Milano

Scuola di Ingegneria Civile,Ambientale e Territoriale Master in Geoinformatics

Feature Selection for Explainable Machine Learning Models

Bresciani Matteo

Referent professor: Brovelli Maria Antonia

Abstract

qq

Contents

Abstract **1**

1 Introduction **3**

2 Background and state of art **5**

3 Overview **7**
3.1 Pre-processing . . . 7
3.1.1 Data Collection . . . 7
3.1.2 Data Cleaning . . . 7
3.1.3 Data Transformation . . . 8
3.1.4 Feature Selection . . . 8
3.2 Model prediction . . . 8

4 Data Collection and Pre-Processing **11**
4.1 Data Collection . . . 11
4.1.1 Source types . . . 12
4.1.2 Spatial resolution . . . 12
Meteo (Table) . . . 12
Pollutants (Table) . . . 14
Soil and Vegetation (Table) . . . 17
GIS (static layers) (Table) . . . 18
Categorical Variable . . . 20
4.2 Data Cleaning . . . 22
4.2.1 Nan Values . . . 22
4.2.2 Remove of variables with low variance . . . 23
4.3 Data Transformation . . . 24
4.3.1 Standardization . . . 24
4.3.2 Normalization . . . 24
4.4 Feature Selection . . . 25
4.4.1 Filter Methods . . . 25
Pearson coefficient . . . 25
Kendall Tau . . . 25
Spearman Rho . . . 26

Fisher Score . . . 26
4.4.2 Wrapper Methods . . . 27
Random Forest Importance . . . 27
4.4.3 Embedded Methods . . . 27
Recursive Feature Elimination . . . 27
4.4.4 Borda Count: averaging FS results . . . 28
4.5 Notebook implementation for Feature Selection . . . 28

5 Case of Study and Data Modelling **30**
5.1 Case of Study . . . 30
5.1.1 Datasets description . . . 30
5.1.2 Results . . . 30
5.1.3 Interpretation of FS results . . . 30
5.2 Data Modelling . . . 30
5.2.1 Results . . . 30

6 Conclusion **31**

Bibliography **32**

Chapter 1

Introduction

Science demonstrated that deterioration of ambient air quality, due to the growing concentration of pollutant in the atmosphere, has caused a significant increment of deaths in the world.

Pollutants such as particulate matter, ozone, carbon monoxide and ammonia cause respiratory diseases, and are important sources of mortality. Almost the entire global population (99%) breathes air that exceeds WHO air quality limits, and threatens their health.
In Europe air is getting cleaner, but persistent pollution, especially in cities, is damaging people's health. One of the last reports which is based on the European Environment Agency's (EEA), shows that exposure to air pollution caused around 500,000 early deaths in the European Union (EU) in 2018 [1].

One of the most harmful pollutants is the **particulate matter (PM25 or PM10)** which can get deep into your lungs or even get into your bloodstream.
Most of the particles come from chemical reactions such as sulphur dioxide and nitrogen oxides, which are pollutants emitted from power plants, industries and automobiles.

However, a significant sources of PM are the chemical reactions generated by intensive farming [3]. In particular this is a relavant issue in the Po Valley, where intensive agricultural activity is very employed.

In this context, human civilization is trying to limit pollution and improve environment with use of technology.
Technology is helping to clean up air pollution, with data analytics-based solutions helping to make our cities healthier places to live.
Monitoring, analysing and predicting the air quality in urban areas is one of the effective solutions for coping with the climate change problem.

The advent of modern Artificial Intelligence (AI) techniques such as Machine Learning (ML) can be considered as new possibilities for researchers to find solutions to various problems affecting air quality and climate change. In this context, the **D-DUST project** (Data-driven moDelling of particUlate with Satellite Technology aid), funded by Fondazione Cariplo's 'Data Science for

Science and Society' call for proposals, counts on Politecnico di Milano, Department of Civil and Environmental Engineering (DICA) as lead partner.
D-DUST aims to provide knowledge about the impact of agricultural and livestock activities on pollutants in the Po Valley (Northern Italy).

For reaching the goal, data from ground sensor are combined with contribution provided by satellite platforms and, with the use of data science techniques like machine-learning and geostatistical models, provide meaningful information related to the contribution of intensive farming on pollution.

The last target of the project is to provide a data-driven best-practices to policymakers, farming operators and citizens in order to minimize the production processes' effects on air quality. In this thesis we propose an ensemble approach for analysing data and provide useful information regarding intense agricultural activity through selection of the most remarkable covariates that impact on PM25 and NH3 pollutants. The final step is to build a model skilled to estimate pollutant estimation locally, better than global scale model.

Chapter 2

Background and state of art

In this chapter I'm going to contextualize the state of the art of my research work. Besides, I'll give explanation about the target to reach and the solution applied.

The goal of my research is to implement a Machine Learning model capable of predicting pollutant locally with better precision than global scale. Data must be pre-processed before training, in order to reduce overfitting and improve accuracy of the final model.
Another aspect to take in consideration is that a ML model trained with so many features it would be a black box, in which a lack of interpretability couldn't be able to explain the decisions taken by the AI. So it's needed to care about interpretability in order to discard eventually confounding variables which can suggest there is a correlation when in fact there is not, even if the model's accuracy is extremely high.
For instance, a new paper by Alex DeGrave et al.[5] shows that Deep Learning model trained with improper data was taking shortcuts in COVID-19 detection on radiographs because of position of certain markers rather than on the actual radiograph. Therefore, the key to increase interpretability of a given model is to wonder if given factor should drive the final decision.

In this context, in which the black-box nature of ML algorithms raises ethical and judicial concerns inducing lack of trust, Explainable Artificial Intelligence aims to create a model fully interpretable. Explainable Artificial Intelligence (or Explainable Machine Learning) helps to understand how ML algorithms make prediction, with the usage of Feature Selection methods for determining how well each feature can predict the target variable. Indeed, before developing a predictive model, feature selection is essential for reducing the number of input variable.
It is desirable to both reduce the computational cost of modelling and, in some cases, to improve the performance of the model.
In addition, inside the D-DUST project this step aims to provided a weighted score of each environmental variable with respect to the pollutants emitted by intense agricultural.
Nowadays, with the large amount of volume and variety in Big Data, FS is becoming increasingly an essential pre-processing step in machine learning algorithms [6].

Due to the fact there isn't a best feature selection technique, I performed and combined different supervised methods.

According to the above, my work comes on this scenario, having the aim to pre-process geospatial data in order to highlight the most weighted input variable that affect the pollutants as target variable.
In the next chapter of my report details I'll show the tools developed and the strategy chosen to reach the target. This is the content of the next chapters:

- chapter 3 (**Overview**): It shows the main steps I take in my work;
- chapter 4 (**Data Collection and Pre-Processing**): It outlines in detail how data are collected and preprocessed, with particular attention to the feature selection;
- chapter 5 (**Case of Study and Data Modelling**): It's focused on the results achieved in the case of study (both feature selection and models built);

Chapter 3

Overview

3.1 Pre-processing

My work is focused on the first phase of a data analysis procedure which is the pre-processing. Data pre-processing (or data preparation) is the process of transforming raw data into a suitable format for modelling. Indeed, raw data is in most cases incomplete and noisy.

Nowadays, dealing with big amount of information, the probability of incorrect data is higher without a proper data pre-processing. Only high-quality data can generate accurate models and predictions.

Hence, it's crucial to process data with the best possible quality before training them with artificial intelligence, and machine learning predictive models.

For doing this I implemented tools collected in Python Notebooks, each one available in the D-DUST repository: (`https://github.com/opengeolab/D-DUST/tree/thesis_MB`).
Its essential steps (shown in figure 3.1) are these.

3.1.1 Data Collection

Relevant data is gathered from their sources and merged in data structures (such as Dataframes). In our work, data come from fixed ground-sensor, satellite-based platform, models and map layers. In this phase are processed (mostly) numerical and categorical data.

3.1.2 Data Cleaning

It involves fixing problems or errors in messy or incomplete data. There are general data cleaning operation, such as identifying:

- duplicate rows of data and remove them;
- rows with NaN values and remove them;
- columns that have low variance and drop them;

3.1.3 Data Transformation

Data need to be scaled. As a matter of fact, each feature in our data has varying degrees of magnitude, range, and units. This is an issue for machine learning algorithms beacuse of highly sensitive to these features. So in input or output data we performed:

- Standardization: Scale a feature to a standard Gaussian distribution;
- Normalization: Scale a feature to the range between 0 and 1;

3.1.4 Feature Selection

Feature Selection is the core part of this study. It's the process of reducing the number of input variables when developing a predictive model by basing on a target (or output) variable. Data collected, even if have been cleaned and transformed, are anyway characterized by big amount of variables which are redundant. Discarding irrelevant data is essential before applying Machine Learning model in order to:

- **Reduce Overfitting**: less opportunity to make decisions based on noise;
- **Improve Accuracy**: less misleading data means that modelling accuracy improves. Predictions can be greatly distorted by redundant attributes;
- **Reduce Training Time**: With less data an algorithm will train faster;

In this step, which will be explained in detail in the next chapters, the reduced input variables are the ones that are meaningless with respect to a target variable as output.
In this study target variables chosen represent the pollution phenomena such as the PM25 and Ammonia emissions. We choose these targets because are the most relevant sources of pollution produced by intensive agricultural.
One of the aim of this step is to detect main pollutant factors which contribute further on the training of PM25 or NH3 emissions. Due to the fact that there isn't a best feature selection technique, many different methods are performed, each one that give different correlation results.

After this step, for every method, a score evaluation is assigned to each variable representing its contribution on the output.

Finally a voting algorithm is performed in order to average the scores obtained in each feature selection method. The highest values are selected for model as input.

3.2 Model prediction

Prediction is a type of analysis that uses techniques and tools to build predictive models and forecast outcomes. In my work predictive analysis is performed for making prediction on pollutants with data processed in the first phase as input.
Model predictions are deployed through regression analysis, used for estimating the relationships between a dependent variable and one or more independent variables.

In particular I use supervised techniques based on Machine Learning where the model built is fit with the training dataset and evaluated its performance with the testset. For doing prediction, I employ 2 supervised AI models:

- **Neural Network regression with Keras**: It's one of the deep learning algorithms that simulate the workings of neurons in the human brain. In a neural network neurons are linked between them forming layers;
- **Machine Learning with Random Forest regressor**: It operates by constructing several decision trees during training time and outputting the mean of the classes as the prediction of all the trees;

After this step an evaluation of the performance of predictions is performed in terms of error and accuracy with a procedure called k-fold cross validation.
Finally, a comparison with CAMS data is performed with the aim to demonstrate that the models produced are better estimated in this local scale.

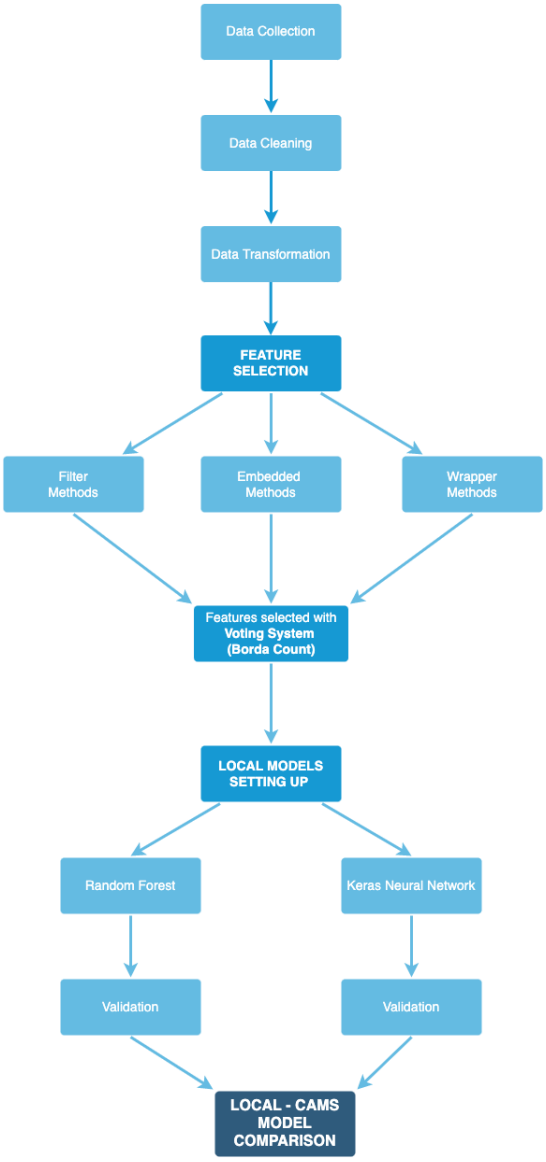

Figure 3.1: Overview of the steps made.

In the next chapters each step will be described in depth about procedures adapted and results obtained.

Chapter 4

Data Collection and Pre-Processing

In this chapter I explain each step taken during the pre-processing phase, by illustrating in details each step taken and tool used.
In order to perform this set of tasks I developed a Python Notebook (available from here).

4.1 Data Collection

Data collection is the process of gathering information in variables of interest for answering relevant questions.
Variables selected are the physical and chemical factors that are most associated with the formation of primary and secondary pollutant.
Therefore, the variables are categorized in 4 different labels:

- Weather: These elements, such as wind speed and direction, precipitation and air temperature, changes in the epochs and can influence air pollution;
- Pollutant: These variables represent primary and secondary pollutant related to the greenhouse effect;
- Soil and Vegetation: Since soil and vegetation degradation are global concerns and can influence the air propagation in the environment, data related to local morphology are collected;
- GIS (static layers): This time-invariant layers are considered to be changeless in the time range considered. Differently from the other types which need a constant monitoring, these variables are update yearly with a lower frequency than the others;

Data chosen are open source and regularly available. In this phase data have been collected (not by me but by other colleagues of D-DUST project at this link) in grids from different sources and provided in Geopackages.

4.1.1 Source types

In order to better distinguish the data sources characteristics, variables selected are labelled with 4 different types of source:

- Ground Sensor: Each ground monitoring stations belongs mainly to ARPA and ESA provides meteorological and air quality data;
- Model: data are estimated through a model built using satellite and meteorological and air quality data as input, such as European data provided by CAMS (Copernicus Atmosphere Monitoring Service);
- Map layer: this data are time-invariant and are related to Lombardy morphology such as density of roads, population or land use;
- Satellite Sensor: They provide data from air quality observation mainly. Satellites provider are Sentinel-5P Tropomi and Terra & Aqua MODIS;

4.1.2 Spatial resolution

Vector grids that are used in the D-DUST project are three and they are generated by the spatial resolution of the source provider.

- Grids with 0.1° resolution with Copernicus CAMS (European);
- Grids with 0.066° resolution based on S5P (this resolution is not included in the case of study);
- Grids with 0.01° Grid defined with maximum one ARPA station for each cell;

Data are scaled and fit in each spatial resolution grid in order to better analyse the final output model by considering each of them.

In the next lines each variable is provided in tables, by showing its type, name and description:

Meteo (Table)

Physical variable	**Source type**	**Variable name**	**Description**	**Unit**	**Source**
Temperature	Model	temp_2m	Mean air temperature at 2 m above the land surface	K	ERA5-Land hourly data.
	Ground Sensor	temp_lcs	Mean air temperature ground measurement - Low Cost Sensor ESA monitoring stations.	°C	ESA Air Quality Platform.
	Ground Sensor	temp_st	Mean temperature - ARPA monitoring stations.	°C	ARPA Lombardia.

Wind	Model	e_wind	Mean eastward wind component 10 m above the land surface	m/s	ERA5-Land hourly data.
	Ground Sensor	wind_dir_st	Wind direction from ground sensor divided in 8 sectors. These are classified into 8 categories as specified in "Notes" column.	cat	ARPA Lombardia.
	Ground Sensor	n_wind	Mean northward wind component 10 m above the land surface.	m/s	ERA5-Land hourly data.
	Ground Sensor	wind_speed_st	Mean wind speed on ground - ARPA monitoring stations.	m/s	ARPA Lombardia.
Precipitation	Model	prec	Mean accumulated liquid and frozen water, including rain and snow, that falls to the Earth's surface. It is the sum of large-scale precipitation.	m	ERA5-Land hourly data.
	Ground Sensor	prec_st	Mean precipitation in each cell in the time range - ARPA monitor stations.	mm	ARPA Lombardia.
Air Humidity	Ground Sensor	air_hum_st	Mean air moisture measurement in the time range - ARPA monitoring stations	%	ARPA Lombardia.
	Ground Sensor	air_hum_lcs	Mean air moisture ground measurement - Low Cost Sensor ESA monitoring stations.	%	ESA Air Quality Platform.
Air Pressure	Model	press	Mean weight of all the air in a column vertically above the area of the Earth's surface represented at a fixed point.	Pa	ERA5-Land hourly data.
Solar Radiation	Ground Sensor	press	Global radiation measurement - ARPA monitoring station.	W/m^2	ARPA Lombardia.

Table 4.1: Table of Meteorological variables.

Pollutants (Table)

Physical variable	Source type	Variable name	Description	Unit	Source
Dust	Model	dust	Mean dust concentration at 0m level provided by CAMS (Ensemble Median - Analysis).	ug/m^3	CAMS Model.
AOD	Satellite Sensor	aod_055	Mean Aerosol Optical Depth at 550nm.	-	MODIS Terra+Aqua.
	Satellite Sensor	aod_047	Mean Aerosol Optical Depth at 470nm.	-	MODIS Terra+Aqua.
	Satellite Sensor	uvai	Mean UV Aerosol Index. A positive index highlights the presence of UV absorbing aerosol (such as smoke/dust).	-	Sentinel-5P
PM10	Model	pm10_cams	Mean PM10 concentration at 0m level provided by CAMS (Ensemble Median - Analysis).	ug/m^3	CAMS Model.
	Ground Sensor	pm10_lcs	Mean PM10 concentration ground measurement - Low Cost Sensor ESA monitoring stations.	?	ESA Air Quality Platform.
	Ground Sensor	pm10_st	Mean PM10 concentration ground measurement - ARPA monitoring stations.	ug/m^3	ARPA Lombardia
PM2.5	Model	pm25_cams	Mean PM2.5 concentration at 0m level provided by CAMS (Ensemble Median - Analysis).	ug/m^3	CAMS Model.
	Ground Sensor	pm25_lcs	Mean PM2.5 concentration ground measurement - Low Cost Sensor ESA monitoring stations.	ug/m^3	ESA Air Quality Platform.
	Ground Sensor	pm25_st	Mean PM2.5 concentration ground measurement - ARPA monitoring stations.	ug/m^3	ARPA Lombardia

SO_2	Model	so2_cams	Mean SO_2 concentration at 0m level provided by CAMS (Ensemble Median - Analysis).	ug/m^3	CAMS Model.
	Satellite Sensor	so2_s5p	Mean SO2 vertical column density at ground level.	mol/m^2	Sentinel-5P.
	Ground Sensor	so2_st	Mean SO_2 concentration ground measurement - ARPA monitoring stations.	ug/m^3	ARPA Lombardia.
NO_2	Model	no2_cams	Mean NO_2 concentration at 0m level provided by CAMS (Ensemble Median - Analysis).	ug/m^3	CAMS Model.
	Satellite Sensor	no2_s5p	Mean NO2 vertical column density at ground level.	mol/m^2	Sentinel-5P.
	Ground Sensor	no2_st	Mean NO_2 concentration ground measurement - ARPA monitoring stations.	ug/m^3	ARPA Lombardia.
	Ground Sensor	no2_lcs	Mean NO_2 concentration ground measurement - Low Cost Sensor ESA monitoring stations.	ug/m^3	ESA Air Quality Platform.
NO	Model	no2_cams	Mean NO concentration at 0m level provided by CAMS (Ensemble Median - Analysis).	ug/m^3	CAMS Model.
NO_x	Ground Sensor	nox_st	Mean NO_x (field: "Ossidi di Azoto") concentration ground measurement - ARPA monitoring stations	ug/m^3	ARPA Lombardia.
CO_2	Ground Sensor	co2_lcs	Mean CO2 concentration ground measurement - Low Cost Sensor ESA monitoring stations	?	ESA Air Quality Platform.

CO	Model	co_cams	Mean CO concentration at 0m level provided by CAMS (Ensemble Median - Analysis).	ug/m^3	CAMS Model.
	Satellite Sensor	co_s5p	Mean CO vertically integrated column density.	mol/m^2	Sentinel-5P.
	Ground Sensor	co_st	Mean CO concentration ground measurement - ARPA monitoring stations.	ug/m^3	ARPA Lombardia.
	Ground Sensor	co_lcs	Mean CO concentration ground measurement - Low Cost Sensor ESA monitoring stations.	ug/m^3	ESA Air Quality Platform.
O_3	Model	o3_cams	Mean O_3 concentration at 0m level provided by CAMS (Ensemble Median - Analysis).	ug/m^3	CAMS Model.
	Satellite Sensor	03_s5p	Mean O_3 total atmospheric column	mol/m^2	Sentinel-5P.
	Ground Sensor	03_st	Mean O_3 concentration ground measurement - ARPA monitoring stations.	ug/m^3	ARPA Lombardia.
CH_2O	Satellite Sensor	ch20_s5p	Mean Formaldehyde tropospheric column number density	mol/m^2	Sentinel-5P.
NMOVOCs	Model	nmvocs_cams	Mean Non-Methane VOCs concentrations at 0m level provided by CAMS.	ug/m^3	CAMS Model.
NH_3	Model	nh3_cams	Mean NH_3 concentration at 0m level provided by CAMS (Ensemble Median - Analysis).	ug/m^3	CAMS Model.
	Satellite Sensor	nh3_lcs	Mean NH_3 concentration ground measurement - Low Cost Sensor ESA monitoring stations	?	ESA Air Quality Platform.
	Ground Sensor	nh3_st	Mean NH_3 concentration ground measurement - ARPA monitoring stations.	ug/m^3	ARPA Lombardia.

Table 4.2: Table of Pollutant variables.

Soil and Vegetation (Table)

Physical variable	Source type	Variable name	Description	Unit	Source
Vegetation	Satellite Sensor	siarlX	Fraction of area in each cell for each agricultural use provided by SIARL Catalog for Lombardy Region.	%	SIARL Lombardia 2019.
	Satellite Sensor	ndvi	Mean NDVI cell value over 16 days period	-	USGS Earth Data.
	Satellite Sensor	siarl	Majority class for agricultural use provided by SIARL Catalog for Lombardy Region.	cat	SIARL Lombardia 2019.
Soil	Model	soil_moist	Mean volume of water in soil layer 1 (0 - 7 cm) of the ECMWF Integrated Forecasting System. The surface is at 0 cm. The volumetric soil water is associated with the soil texture (or classification), soil depth, and the underlying groundwater level.	m^3/m^3	ERA5 Land Hourly Data.
	Map Layer	soilX	Fraction of area for each cell containing the soil type obtained from OpenLandMap soil texture classification.	%	OpenLandMap Soil Texture Class (USDA System).
	Map Layer	soil_textX	Mean NDVI cell value over 16 days period	%	Basi informative dei suoli - Geoportale Lombardia.
	Map Layer	soil	Majority soil type for each pixel from OpenLandMap soil texture classification .	cat	OpenLandMap Soil Texture Class (USDA System) .
	Map Layer	soil_text	Majority soil type for each pixel from Carta pedologica 250K (Lombardy Region).	cat	Basi informative dei suoli - Geoportale Lombardia.

Table 4.3: Table of variables referred to Vegetation and Soil.

GIS (static layers) (Table)

Physical variable	**Source type**	**Variable name**	**Description**	**Unit**	**Source**
Geometry	Map Layer	area	Area of Lombardy Region vector layer in each cell.	km^2	SIARL Lombardia 2019.
	Satellite Sensor	ndvi	Mean NDVI cell value over 16 days period	-	-
Population	Map Layer	pop	Population for each cell.	n° of inhabitants	Gridded Population of the World (GPW).
Land use and cover	Map Layer	dsfX	Land use fraction for each cell containing the classification the classification provided by DUSAF Catalog (Lombardy Region).	% (fraction for each cell)	DUSAF Lombardia 2018.
Map Layer	dusaf	Cover	Land Use majority class for each cell provided by DUSAF Catalog (Lombardy Region).	cat	DUSAF Lombardia 2018.
Terrain	Map Layer	h_mean	DTM average elevation for each pixel.	m	Geoportale Lombardia 2019.
	Map Layer	aspect_major	Aspect derived from DTM. Majority pixel aspect.	Degree North	Geoportale Lombardia 2019.
	Map Layer	slope_mean	Average slope derived from DTM.	Degree North	Geoportale Lombardia 2019.

Road Infrastructures	Map Layer	int_prim	Density of intersection nodes between primary roads for each cell (including highways).	int_s/km^2	Geoportale Lombardia 2019.
	Map Layer	int_prim_sec	Density of intersection nodes between primary and secondary roads for each cell.	int_s/km^2	Geoportale Lombardia 2019.
	Map Layer	int_sec	Density of intersection nodes between secondary roads for each cell.	int_s/km^2	Geoportale Lombardia 2019.
	Map Layer	prim_road	Density of primary importance roads for Lombardy Region inside for each.	km/km^2	Geoportale Lombardia 2019.
	Map Layer	sec_road	Density of secondary importance roads for Lombardy Region foreach cell.	km/km^2	Geoportale Lombardia 2019.
	Map Layer	highway	Density of highways for Lombardy Region inside for cell divided.	km/km^2	Geoportale Lombardia 2019.
Farms	Map Layer	farms	Fration of area covered by farms inside the cell. Obtained from DUSAF dataset.	% (fraction for each cell)	DUSAF Lombardia 2018.
Air quality zones	Map Layer	aq_zone	Majority class of a given air quality zone in each cell.	cat	Geoportale Lombardia.
Climate zones	Map Layer	clim_zone	Majority class of a given air quality zone in each cell.	cat	-

Table 4.4: Table of Static GIS variables.

Categorical Variable

Categorical data are identified with names or labels given to them as value. Even if are represented by numbers, they don't have the same mathematical meaning as a numerical value. This type of data is discarded during the pre-processing phase, since feature selection is done exclusively on numerical input and output values. In the following table is explained the semantic of the values assumed.

Variable name	**Note**
Meteo	
wind_dir_st	1 = North: 0° - 22.5° / 337.5° - 360°, 2 = North-East: 22.5° - 67.5°, 3 = East: 67.5° - 112.5°, 4 = South-East: 112.5° - 157.5°, 5 = South: 157.5° - 202.5°, 6 = South-West: 202.5° - 247.5°, 7 = West: 247.5° - 292.5°, 8 = North-West: 292.5° - 337.5°
Soil and Vegetation	
siarl	2 = Cereal 9 = Mais 12 = Rice
soil	2=Cereal 9=Mais 12=Rice
soil_text	1 = Clay 2 = Silty Clay 3 = Sandy Clay 4 = Clay Loeam 5 = Silty Clay Loam 6 = Sandy Clay Loam 7 = Loam 8 = Silt Loam 9 = Sandy Loam 10 = Silt 11 = Loamy Sand 12 = Sand.

GIS (Static layers)	
dusaf	2 = Agricultural areas 3 = Wooded territories and semi-natural environments 4 = Wetlands 5 = Water bodies 11 = Urbanised areas 12 = Production facilities, large plants and communication networks 13 = Mining areas, landfills, construction sites, waste and abandoned land 14 = Non-agricultural green areas
aq_zone	1 = Highly urbanized plains 2 = Plains 3 = Prealpi, Appennino and mountains 4 = Valley floor Agg. 5 = Urban agglomarated area (Milano, Bergamo, Brescia).
clim_zone	1= Alpi 2 = Prealpi Occidentali 3 = Prealpi Orientali 4 = Pianura Occidentale 5 = Pianura Centrale 6 = Pianura Orientale.

Table 4.5: Table of categorical variables with their values legend.

4.2 Data Cleaning

Data has to be prepared in accordance with the supervised feature selection. Data cleaning aims to fix problems or errors in messy data. There are many reasons data may have incorrect values, such as being corrupted, duplicated or invalid.
This could be done by removing rows or columns. Alternately, it might involve replacing observations with new values.
Firstly data covariates are divided between input (X) and output variable (Y). X represents all of variables collected in the previous part, excepting for the pollutant to be analysed and modelled (such as PM25 or Ammonia) which is assigned to the Y variable.

In this section are underlined the main steps performed in the Data Cleaning phase.

4.2.1 Nan Values

In my work I consider as target variable PM25 and Ammonia coming from ground sensor measurement. Air quality monitoring is usually carried out through ground sensors networks, which represents the primary air quality data source by governance.
In the grids processed there's the problem that a given value provided by measurement tools (such as ground and satellite sensor) could be NaN. It's feasible since:

- A sensor could have no measurement for a given time epoch;
- The set of sensor, because of its limited supply, cannot cover each cell of a grid;

In our case variable provided by ARPA and ESA ground sensor (with the label that ends with '_st' and '_lcs' respectively) has many NaN cells.

However, no country in the world has yet established a monitoring network with a full satisfying coverage[8]. Even in the United States (US), which is characterized by a relatively developed PM2.5 ground monitoring network with 2500 stations has many areas unmonitored[8].

In order to mitigate this, I present this solution in sequence, using methods provided by Pandas library:

- Drop of samples having target variable with NaN value;
- Drop of columns (values assumed by each covariates) having at least a NaN value;

Due to the fact that it results a dataset with a very limited number of sample [15], I perform additionally a k-nearest neighbour classifier[13] for adding a buffer of values close to the location of the ground stations measurement. Values added are computed using a Radial Basis function interpolation[14].
In this way the size of the final sample, as the performance of the feature selection, would increase.

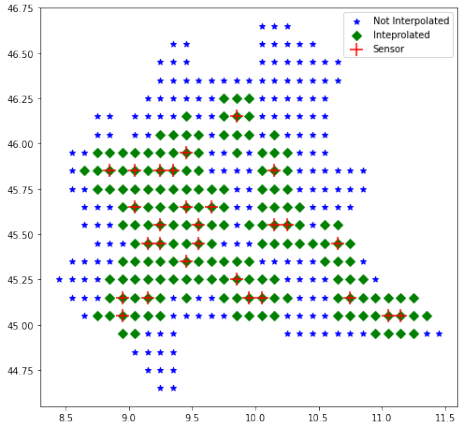

Figure 4.1: Graphical representation of how buffer values interpolated are added through k-nearest neighbour.

4.2.2 Remove of variables with low variance

An approach for removing columns is to consider the variance of each column variable. The variance is a statistic representing the expected value of the squared deviation from the mean of a given variable X μ.

$$Var(X) = E[(X - \mu)^2] \tag{4.1}$$

The variance can be used as a filter for identifying columns to be removed from a given dataset. Using a feature with low-variance only adds complexity and noisy to the feature selection and the predictive.
In order to do that, I performed VarianceThreshold method from the scikit-learn library. In this way, features under a certain variance threshold value should be meaningless and consequently discarded by its dataset.

4.3 Data Transformation

Having input variables with different units (e.g. ug/m^3, °C, hours or mol/m^2) implies data at different scales. This could raise the difficulty of the problem being modelled.
Hence, a common scale through Normalization or Standardization is needed in order to improve the data quality.
Many ML and regression algorithms perform better when numerical input and output variables are scaled to a common standard range.
For instance, it's proved that neural networks trained with scaled data performs better in terms of MSE [11]. In this step, two type of transformation have been done:

4.3.1 Standardization

The most common data transformation is to centre and scale the each variable values. In order to do that, the average value is removed from all the values. As a result of centring, the predictor will have a zero mean.[7] Standardization consists in rescaling data following a gaussian distribution of values with mean equals to 0 and standard deviation equals to 1:

$$Z = \frac{X - \mu}{\sigma} \tag{4.2}$$

$$\mu = \frac{(\sum_{n=1}^{N} X_i)}{N} \tag{4.3}$$

$$\sigma = \sqrt{\frac{(\sum_{n=1}^{N} X_i - \mu)}{N - 1}} \tag{4.4}$$

Where:

- Z is the numeric value standardized of a given covariate;
- X is the numeric value to be standardized of a given covariate;
- μ is the mean value for the set of values assumed by a given covariate;
- σ is the standard deviation for the set of values assumed by a given covariate;

Every terms was computed by using Scipy library (scipy.stats).

4.3.2 Normalization

Data Normalization is a different methods process for adjusting data at different scales. Data a scaled in a range between 0 and 1 and was performed only for the feature selection methods output. Output normalization is an essential step for the comparison of different output, since data ranges vary for each method used.
This was performed in my Notebooks from the scikit-learn library (sklearn.preprocessing) through the MinMaxScaler method.

4.4 Feature Selection

After the previous steps, in which a dataset is cleaned and transformed, FS methods are performed. The output results consist in, for each method a set of score for each variable. The score evaluated corresponds to the importance weighted of each feature with respect to the target variable. In the following subsection each FS method implemented is described in detail, classified in three main categories[12], as we can find in literature:

4.4.1 Filter Methods

Filter-based feature selection methods adopt statistical measures to evaluate the correlation/dependence between input variables.
These select features from the without machine learning algorithm. In terms of computation, they are very fast and are very suitable in order to remove duplicated, correlated, redundant variables[9]. These methods evaluate each feature individually without considering the interaction between them. Therefore, they don't fit well if data has high multicollinearity[4].

Pearson coefficient

Pearson coefficient is one of the most widely used indices for measuring linear correlation in statistics. It ranges between -1 and 1, where:

- 1 indicates a strictly positive correlation;
- -1 indicates a strictly negative correlation;
- 0 indicates no correlation between the features;

Therefore, by taking only its absolute value, 1 implies that a linear equation describes the relationship between X and Y perfectly, for both positive and negative correlation.
The Pearson index between and independent variable X and a target variable Y is defined by the following formula:

$$\rho_{x,y} = \frac{Cov(X,Y)}{\sigma_x \sigma_y} \tag{4.5}$$

Kendall Tau

Kendall Tau index is used to measure monotonic relationship as test statistic to determine whether two variables are statistically dependent.
While in the linear correlation two variables move together at a constant rate, monotonic or rank correlation measure how likely two variables move in the same direction, but not necessarily in a constant manner.
Like Pearson's correlation, Kendall's has a value between -1 and 1, where:

- -1 represent a strictly negative monotonic relationship;
- 1 represent a strictly positive monotonic relationship;
- 0 representing no relationship;

Given a sample X and Y with n as sample size, tau index is computed through the formula:

$$\tau_{x,y} = \frac{n_c - n_d}{\frac{1}{2}n(n-1)} \tag{4.6}$$

where:

- n_c = # of concordant value (concordant value: value are ordered in the same way);
- n_d = # of discordant value (discordant value: value are ordered differently);

Spearman Rho

Spearman's index is very similar to Kendall's. As the previous filter methods, it ranges between -1 and 1, and it's considered less robust than Kendall's. It's computed in this way:

$$\rho_{x,y} = \frac{6\sum_{n=1}^{N} d_i^2}{n(n-1)^2} \tag{4.7}$$

- d_i: difference between each corresponding X_i and Y_i;
- n: size of the sample;

Finally, as I did for Pearson and Kendall coefficient, I take in consideration only its absolute value to weight the correlation for each variable in the Feature Selection.

Fisher Score

This method returns the score of the variables based on the fisher's score in descending order.
Its algorithm is implemented by using SelectKBest method from the scikit-learn library (sklearn.feature_selection).

4.4.2 Wrapper Methods

Wrapper methods, as the name suggests, wrap a machine learning model, with different subsets of input features. In this way the subsets are evaluated following the best model performance. One disadvantage of this approach is the computational costs.
Their execution for many subsets of variables can become unfeasible.

Random Forest Importance

Feature importance is a built-in function of the Random Forest algorithm. It's also called as Gini importance (or mean decrease impurity) and is commonly used as the splitting criterion in decision trees problem. The scores are evaluated as attribute through RandomForestRegressor of the scikit-learn library (sklearn.ensemble).

4.4.3 Embedded Methods

Embedded methods instead are characterised by the benefits of both the wrapper and filter methods, by including interactions of features but also having a reasonable computational cost.

Recursive Feature Elimination

RFE is a wrapper feature selection algorithm that also work with filter-based feature selection internally.
It consists in looking for the best subset of features by starting with all features and removing some of them until the desired number remains.
This is computed using RFE of scikit-learn library (sklearn.feature_selection). In order to obtain a score for each variable I consider the ranking value (with ranking_ attribute) which represent the ranking position for each feature.

4.4.4 Borda Count: averaging FS results

One of the most important challenges in this study is the lack of an universal feature selection method which produces an outcomes in common with all FS technique. Choosing a feature selection method from a vast range of choices can be challenging.
So it needs an ensemble technique aims to makes it more robust across various algorithms. In this work we adopt an ensemble approach described in this study[10], using the Borda Count algorithm. Initially Borda Count was a voting system method, named for Jean-Charles de Borda[2].
In this context Borda Count is used as a rank-based combination technique used for evaluate an average score for each feature. In this method, assuming that each scores evaluated by each FS method are sorted in descending order, points are assigned to candidates (variables) based on their ranking; 1 point for last choice (the most meaningless by its score), 2 points for second-to-last, and so on. Finally the points for all ballots are summed up, and the candidates with the largest point total are the winners (the feature with the largest points are the most significant).

At this point variables that won are used as input in ML models and taken in consideration as the most meaningful factor affecting the target variable.

4.5 Notebook implementation for Feature Selection

A notebook is implemented for processing and analysing data as I explain before.
At the beginning the general parameter needed to configure the pre-processing phase are declared.

```
RESOLUTION= '0_1' //resolution of the data chosen
KNN = True    //If true a k-nearest neighbour is performed
knn_value = 10 //Number of buffer values added close to each target variable value
NO_MOUNTAINS = True
```

In order to manage its configuration and the results obtained, a simple User Interface using ipywidgets package is built. In this interface there are 2 sections:

- Feature Selection scores: they are graphically shown using multiple barplot, one for each dataset previously selected. Barplot are implemented with the use of Plotly library;
- Options: in this s box is possible to configure the feature selection input:
  - target variable;
  - value of the Variance Threshold for discarding meaningless variable before FS (optional);

  and the output:
  - choice of method for visualize its own scores;
  - results normalization (optional);
  - order of the scores by descending order or by labels;

– scale of Y-axis (regular or logarithmic);

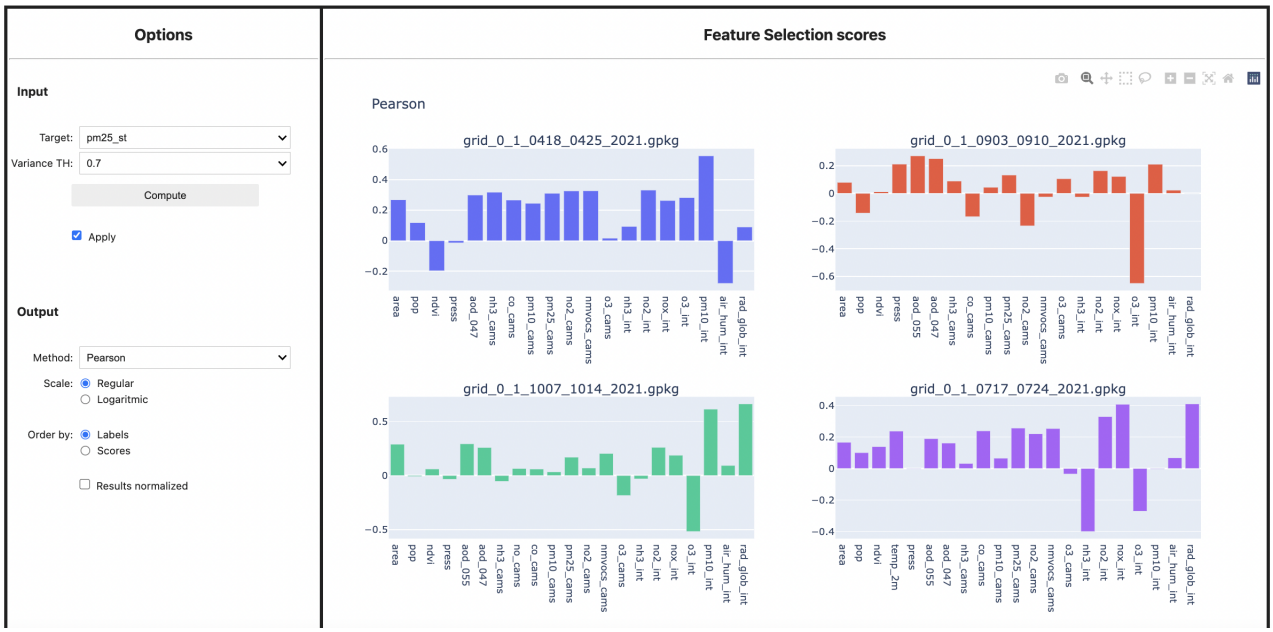

Figure 4.2: Overview of the notebook implemented for FS procedure.

Chapter 5

Case of Study and Data Modelling

5.1 Case of Study

5.1.1 Datasets description

5.1.2 Results

5.1.3 Interpretation of FS results

5.2 Data Modelling

5.2.1 Results

Chapter 6

Conclusion

Bibliography

[1] European Environment Agency. Air quality in europe 2018. *Report No. 12/2018*, 2018.

[2] JC de Borda. Mémoire sur les élections au scrutin. *Histoire de l'Academie Royale des Sciences pour 1781 (Paris, 1784)*, 1784.

[3] MR Burkart. Diffuse pollution from intensive agriculture: sustainability, challenges, and opportunities. *Water science and technology*, 55(3):17–23, 2007.

[4] Jamal I Daoud. Multicollinearity and regression analysis. In *Journal of Physics: Conference Series*, volume 949, page 012009. IOP Publishing, 2017.

[5] Alex J DeGrave, Joseph D Janizek, and Su-In Lee. Ai for radiographic covid-19 detection selects shortcuts over signal. *Nature Machine Intelligence*, 3(7):610–619, 2021.

[6] Saidjon Kamolov. Feature selection: state-of-the-art survey. *Annals of Mathematics and Computer Science*, 4:48–54, 2021.

[7] Max Kuhn, Kjell Johnson, et al. *Applied predictive modeling*, volume 26. Springer, 2013.

[8] Ying Liu, Guofeng Cao, Naizhuo Zhao, Kevin Mulligan, and Xinyue Ye. Improve ground-level pm2. 5 concentration mapping using a random forests-based geostatistical approach. *Environmental Pollution*, 235:272–282, 2018.

[9] Yvan Saeys, Inaki Inza, and Pedro Larranaga. A review of feature selection techniques in bioinformatics. *bioinformatics*, 23(19):2507–2517, 2007.

[10] Chandrima Sarkar, Sarah Cooley, and Jaideep Srivastava. Robust feature selection technique using rank aggregation. *Applied Artificial Intelligence*, 28(3):243–257, 2014.

[11] Murali Shanker, Michael Y Hu, and Ming S Hung. Effect of data standardization on neural network training. *Omega*, 24(4):385–397, 1996.

[12] Urszula Stańczyk. Feature evaluation by filter, wrapper, and embedded approaches. In *Feature Selection for Data and Pattern Recognition*, pages 29–44. Springer, 2015.

[13] Kashvi Taunk, Sanjukta De, Srishti Verma, and Aleena Swetapadma. A brief review of nearest neighbor algorithm for learning and classification. In *2019 International Conference on Intelligent Computing and Control Systems (ICCS)*, pages 1255–1260. IEEE, 2019.

[14] Grady Barrett Wright. *Radial basis function interpolation: numerical and analytical developments*. University of Colorado at Boulder, 2003.

[15] Ying Zhang and Chen Ling. A strategy to apply machine learning to small datasets in materials science. *Npj Computational Materials*, 4(1):1–8, 2018.