POLITECNICO
DI TORINO

Master degree course in Computer Engineering

Master Degree Thesis

From Mineralogy to Petrography: Study on the Applicability of Machine Learning

Supervisor
Prof. Elena Baralis
Dott. Andrea Pasini

Candidate
Francesca Cibrario

December 2018

Abstract

This thesis aims to understand if machine learning techniques may be applied to forecast the petrographical composition of a sample of soil given its mineralogical structure. The problem formalization requires predicting, from a set of continuous attributes (mineralogical constitution), a set of continuous features (petrographical constitution). The soil samples we have at disposal, for which both compositions are known, present two challenges: their limited quantity and the non-uniform distribution of values of their features. These distributions are approximatively gaussian or exponential.
We have addressed the prediction task with both regression and classification techniques. In the second case, to obtain class labels, we have discretized petrographical attributes using a custom methodology. This allows classifying a sample by indicating the possible range of values of the attribute. We applied Linear, Lasso, Ridge and Support Vector Machine (SVR) as regression techniques. The classification method used is the Decision Tree. Furthermore, we tested a third regression approach exploiting Non-Negative Matrix Factorization (NMF) technique. We have designed a way to evaluate the ability of regression models to predict less recurrent values in the dataset: it consists of two visual metrics called regression-precision and regression-recall inspired by precision and recall classification metrics. Results show that models perform better on exponentially distributed petrographical attributes. Linear, Lasso and Ridge regressions are more promising than Decision Tree. NMF and SVR behave the worst. In the future we will focus on the possibility to exploit some relations that partially link petrographical composition with mineralogical one, with the help of a domain expert, trying in such a way to compensate the lack of data with domain knowledge. However, although we still have not achieved final results, our preliminary analysis reveal that there is the possibility to successfully apply regression techniques for this kind of soil analysis.

Contents

I Background **4**

1 Introduction 5
- 1.1 Context . . . 5
- 1.2 Objective . . . 5
- 1.3 Proposed solution and contribution . . . 6
- 1.4 Overview . . . 8

2 Regression 9
- 2.1 Technique . . . 9
- 2.2 Linear . . . 10
- 2.3 Lasso . . . 10
- 2.4 Ridge . . . 11
- 2.5 Polynomial . . . 11
- 2.6 SVR . . . 12

3 NMF 14
- 3.1 Technique . . . 14
- 3.2 Applications . . . 15

4 Classification 17
- 4.1 Technique . . . 17
- 4.2 Decision tree . . . 17
- 4.3 Neural network . . . 18
- 4.4 Rule based . . . 20
- 4.5 K-Nearest Neighbor . . . 20
- 4.6 SVM . . . 21

5 Discretization 22
- 5.1 Technique . . . 22
- 5.2 Equal Width and Frequency binning . . . 23

II Implementation **24**

6 Method 25
6.1 Dataset . . . 25
6.1.1 Consideration on error of measurement . . . 28
6.1.2 Distribution . . . 30
6.2 Prediction techniques selection . . . 33
6.3 Process overview . . . 35
6.4 Discretization . . . 36
6.5 NMF approach . . . 39
6.6 Metrics . . . 40
6.6.1 Regression Metrics . . . 40
6.6.2 Classification Metrics . . . 41
6.6.3 Custom Metrics . . . 42
6.6.4 Regression and Classification comparison methodology . . . 44

7 Experiments 46
7.1 Aggregated vs non aggregated dataset . . . 46
7.2 Classification . . . 49
7.2.1 Discretization . . . 49
7.2.2 Decision Tree . . . 53
7.3 Regression techniques comparison . . . 56
7.4 Classification vs Regression . . . 64

8 Conclusions and future works 66

Part I

Background

Chapter 1

Introduction

1.1 Context

Soil analysis is important for oil companies in order to understand which areas are more promising for oil and gas extraction. Geological samples are extracted to determinate their internal composition, which is useful to evaluate the suitability of the area. A sample constitution can be expressed in terms of minerals or rocks: respectively mineralogical and petrographical composition. Mineralogy is the discipline that studies minerals. Petrography instead focuses on rocks. Rocks are composed by minerals, this leads to some dependencies between the two compositions. However, there isn't a chemical formula that allows to obtain petrographical constitution given mineralogical one. To extract these two types of composition is necessary to subject the sample to two separated procedures, one for mineralogy and one for petrography. Cost of performing the latter is higher than the one required for the former. Unfortunately information provided by petrographical composition are more interesting for this kind of soil analysis.

1.2 Objective

The aim of this thesis is to predict a sample petrographical composition given its mineralogical one, to avoid performing petrographical analysis, which is an expensive procedure. We apply our analysis to some samples for which both mineralogical and petrographical constitutions are known. These compositions are expressed through a set

of attributes. Each attribute codes a specific rock or mineral. Every sample posses values for these features, representing the percentage of the rock/mineral present in the sample, with respect to its total composition. The formalization of the problem is predicting continuous values (petrographical composition) having at disposal a set of continuous features (mineralogical composition). The difficulties of this specific application field are the small quantity of data at disposal and the non uniform distribution of attributes. In particular some attributes present an approximatively gaussian distribution, while others an exponential one. The necessity is to predict correctly also those values that are less recurrent in the dataset, especially for exponentially distributed attributes, where all predictions tend to be part of a same range of values. Petrographical attributes that are available for our study are organized in two versions, one is more general and presents more attributes, while the other is an aggregated version of the first one. In practice the last one presents a less detailed description of petrographical composition, because groups of rocks are considered together.

1.3 Proposed solution and contribution

Three data mining techniques have been taken into account: regression, classification and a Non Negative Matrix Factorization (NMF) based approach. Regression and classification work separately for each petrographical attribute: one different model is built for each of them. All attributes of mineralogy are used to build the model together with the petrographical attribute in consideration, no other attribute of petrography is exploited.

Regression is the more obvious choice. As already stated the formalization of the problem is to predict continuous values having at disposal continuous features and this is actually the aim of regression. Different regression models have been tested on the dataset: Linear, Lasso, Ridge and SVR.

Classification, instead of predicting a continuous value, indicates a range in which the real could fall. These ranges have been obtained discretizing attributes values. This approach allows choosing properly ranges, to simplify the prediction of non recurrent values. In fact, enlarging ranges leads to more samples per range, augmenting the recurrence of the range in the dataset. This leads to a less precise prediction too, because the value effectively assumable by the attribute belongs to a largest range. A limitation

of this approach is that the prediction is strongly linked to the provided dataset: ranges whose values are not present in the dataset will never be predicted. Only one classification model is considered: the Decision Tree.

NMF approach exploits a different procedure: all petrographical attributes are predicted simultaneously, only one model is built. NMF has been chosen because it bonds petrographical attributes with each other. Moreover it is possible to add further links and constraints between attributes by slightly modifying the original algorithm. This approach is treated and evaluated as a regression technique.

The evaluation of models performances has been done considering the need of correctly predicting less recurrent values. Classification is evaluated through F-measure, precision and recall. These two last metrics are well suited to evaluate rare classes. Regression is evaluated through Mean Absolute Error (MAE) and Explained Variance Score (EVS). None of these two metrics allows to analyse in detail performances for less recurrent values. Because of this need we have defined some custom visual metrics called regression-precision and regression-recall to permit this kind of evaluation.
We also propose a way to compare performances of regression and classification models: regression continuous predictions have been discretized. In this way a direct comparison can be done by exploiting same metrics used for classification.
As we will see in Chapter 7 range enlargement strategy provides an improvement of performances for Decision Tree, but Linear, Lasso and Ridge regressions in general perform better than classification, SVR and NMF. It is worth mentioning that performances of models strongly depend on petrographical attribute taken into account and its distribution. Models perform in general better with exponential distribution with respect to gaussian one.

1.4 Overview

This thesis is divided into two parts. The former is a theoretical overview, the latter concerns the presentation of the solution. First part is organized as follows.
An analysis of some regression techniques is reported in Chapter 2, while some classification ones are presented in Chapter 4. NMF mathematical procedure is shown in Chapter 3, together with common data mining application fields. Discretization aim is described in Chapter 5, with some common implementations.
The second part of the thesis, the proposed solution, is structured as reported below. Chapter 6 starts with the description of the considered datasets (Section 6.1). Techniques that we have decided to test on the dataset and the reason why we chose them are reported in Section 6.2. Then, in Section 6.3, is reported in detail how each dataset is exploited for models generation and validation. In particular custom implementations of discretization and NMF approach are detailed respectively in Section 6.4 and in Section 6.5. An overview of quality metrics is reported in Section 6.6. These are used in Chapter 7 to perform comparisons between different techniques and datasets. Finally conclusions deduced by this work are reported in Chapter 8.

Chapter 2

Regression

In this chapter is performed an overview of regression technique. In Section 2.1 is presented a general description and aim of the method, with an introduction to the terminology used in this chapter. In other sections is provided a picture of some common regression algorithms applicable to our dataset: more than one continuous feature used to predict a continuous value. These techniques are Linear (Section 2.2), Lasso (Section 2.3), Ridge (Section 2.4), Polynomial (Section 2.5) and Support Vector (Section 2.6) regression.

2.1 Technique

Regression is a data mining technique that provides prediction of a continuous value (predicted variable) given a set of features (predictors). A regression model is an entity able to perform this kind of prediction. It aims to find the mathematical relation existing between N predictors ($\{x_i\}_{i=0}^{N-1}$) and the predicted variable (y). The model stores a function that approximates this relation:

$$\hat{y} = f(x_0, x_1, \ldots, x_{N-1}) \tag{2.1}$$

The relation f, that links input features and the prediction ($\hat{y}$), has a format dependent on the specific regression method. This function presents some parameters that must be learned by the model in the training phase. In the following sections some regression methods are reported and the focus is given on f function and the way it is learned.

2.2 Linear

Linear regression technique is based on the assumption that the function linking predictors and predicted variable is linear. Relation f, introduced in Equation 2.1, is then a linear combination of predictors and has the format:

$$\hat{y} = a_0 x_0 + a_1 x_1 + \ldots + a_{N-1} x_{N-1} + a_N \tag{2.2}$$

Parameters to be learned by the model are $\{a_i\}_{i=0}^{N}$ coefficients. Given a certain training set, these components can be found by minimizing a quantity, the loss function, that expresses the error committed by the model in predicting training samples. There are different types of loss functions and procedures to minimize it, as stated in [1]. A common loss function is the sum of squared errors:

$$loss = \sum_{i=0}^{M-1} (y_i - \hat{y}_i)^2 \tag{2.3}$$

M is the number of samples in the training set, y_i is the true value for sample i in the dataset, while $\hat{y}_i$ is the prediction for the i-th sample given by regression model. Procedures that optimize this kind of loss function are called ordinary least squares [2].

2.3 Lasso

Lasso regression is a variant of Linear regression (described in previous section): it supposes a linear relation between predictors and predicted variable. However, the function to be minimized in learning phase is slightly different from Linear regression one, with the aim to avoid overfitting. An operation that prevents from this phenomena is called regularization. Overfitting is a condition showing up when a model learns too much training set characteristics. This leads to wrong predictions when the model has to forecast new unseen data. This is due to model inability to properly generalize what it has learned. The function to be minimized is:

$$loss + \lambda \sum_{i=0}^{N-1} |a_i| \tag{2.4}$$

loss is to be intended as the one defined in Equation 2.3. The added term is needed to activate or deactivate features: it performs feature reduction and parameters shrinkage.

Consider to minimize only the second term: it decreases the more values of $\{a_i\}_{i=0}^{N-1}$ are closer to zero.
If a_i is equal to zero feature selection is performed: in Equation 2.2 the dependency of $\hat{y}$ from x_i is deleted. This avoids the model to learn from features that are not so relevant to determine y value.
The fact that parameters are limited in magnitude is call parameters shrinkage. Limiting parameters range performs regularization too: it avoids that they fit too much the training set. In fact, allowing them only small variations, leads to the construction of a function that has a limited complexity and consequently cannot follow exactly the trend defined by the training samples.

2.4 Ridge

Ridge regression is a variant of Linear regression (Section 2.2) and presents the same aim as Lasso (Section 2.3): to avoid overfitting. It has, as Lasso, same format of f defined for Linear regression in Equation 2.2 and it adds a term to the function (Equation 2.3) to be minimized. Differently from Lasso, no feature selection is performed by Ridge. The function to be minimized is:

$$loss + \lambda \sum_{i=0}^{N-1} a_i^2 \tag{2.5}$$

the second term substantially pushes values of $\{a_i\}_{i=0}^{N-1}$ toward a zero, but forces none of them to assume a zero value, as Lasso does. Ridge in conclusion avoids overfitting by only shrinking coefficients. For a statistical and geometrical interpretation of Lasso and Ridge see [3].

2.5 Polynomial

Polynomial regression is used when the following assumption on data is done: $\{x\}_{i=0}^{N-1}$ and y are linked by a relation that is not linear. It proposes a relation f (with reference to Equation 2.1) that is a polynomial function with grade greater than one. This means that y cannot be well modeled by a first grade polynomial: a linear combination of $\{x\}_{i=0}^{N-1}$, as for Linear regression (see Equation 2.2). When using this kind of regression it is

necessary to choose the proper grade of polynomial to use.
Polynomial regression, in practice, can be treated as a Linear regression. Polynomial in fact is a sum of monomials. For translating the problem from a Polynomial regression to a Linear one all operations among literals defined by monomials are applied to predictors. The result of this operation is a new set of predictors, whose cardinality is equal to the number of monomials, on which to apply Linear regression. An example is given with a second grade polynomial in two variables:

$$\hat{y} = a_0 x_0 + a_1 x_1 + a_2 x_0^2 + a_3 x_1^2 + a_4 x_0 x_1 + a_5 \tag{2.6}$$

This is the relation f (with reference to Equation 2.1) for a Polynomial regression having as predictors x_0 and x_1.
For treating it as a Linear regression, operations defined by monomials are applied to obtain new predictors:

$$\begin{aligned} X_0 &= x_0 \\ X_1 &= x_1 \\ X_2 &= x_0^2 \\ X_3 &= x_1^2 \\ X_4 &= x_0 x_1 \end{aligned} \tag{2.7}$$

This is done for all elements in the dataset and leads to a new dataset whose records are made of five predictors $\{X_i\}_{i=0}^{4}$. These new features are exploited using Linear regression, solving a problem that is equivalent to the one defined by Equation 2.6:

$$\hat{y} = a_0 X_0 + a_1 X_1 + a_2 X_2 + a_3 X_3 + a_4 X_4 + a_5 \tag{2.8}$$

2.6 SVR

Support Vector Machines are supervised learning methodologies used for both classification and regression. In the case they are used for regression, the technique that exploits them takes the name of Support Vector Regression (SVR). Its aim is to find a function f that has the same format as for Linear regression (see Equation 2.2), that links predictors

and predicted value. First of all the same function is presented in vectorial form:

$$\begin{aligned} \vec{x} &= (x_0, x_1, \ldots, x_{N-1}) \\ \vec{a} &= (a_1, a_2, \ldots, a_{N-1}) \\ \hat{y} &= \vec{x} \cdot \vec{a} + a_N \end{aligned} \tag{2.9}$$

Learning of parameters $\{a_i\}_{i=0}^{N}$ is done by minimizing a quantity under two constraints:

$$\begin{aligned} &\text{minimize} && \frac{\|\vec{a}\|^2}{2} = \frac{\vec{a} \cdot \vec{a}}{2} \\ &\text{under conditions} && \begin{cases} y - \vec{a} \cdot \vec{x} - a_N \leq \epsilon \\ \vec{a} \cdot \vec{x} + a_N - y \leq \epsilon \end{cases} \end{aligned} \tag{2.10}$$

$\vec{a} \cdot \vec{x} - a_N$ is the prediction of SVR for $\vec{x}$, y is the true value and ϵ is a tolerance factor that indicates a range inside which the prediction can be considered correct. The part to be minimized presents again the same concept of regularization explained for Lasso (Section 2.3) and Ridge (Section 2.4) regression.

Moreover SVR provides a way to model relations between y and $\vec{x}$ that are non linear. Note that all vectorial operations in Equation 2.10 exploits the dot product. Kernel functions provides a definition of the dot product in a space in which the function to be modeled is linear.

Chapter 3

NMF

In this chapter a mathematical presentation of Non Negative Matrix Factorization is done in Section 3.1. In Section 3.2 state of art data mining application fields of NMF are described: recommendation systems, feature extraction and clustering. In particular, focus is given on the interpretation that each application ascribes to NMF procedure.

3.1 Technique

Non Negative Matrix Factorization is a procedure that allows to express a matrix T, in function of two other matrices X and Y, by factorizing it:

$$T \approx XY \tag{3.1}$$

X and Y are non negative, algorithms for calculating them are proposed in [4]. The purpose of these algorithms is to solve the following problem:

$$\min_{X,Y} \mathrm{d}(T, XY) \tag{3.2}$$

d is a measure of dissimilarity between two matrices, in such a case T and XY. It must be minimized to obtain a matrix $XY = \hat{T}$ similar to T.

Speaking about dimensions, if T matrix has size $M \times N$, X has size $M \times K$ and Y $K \times N$, K is chosen minor of $\min(M, N)$.

3.2 Applications

State of art applications exploiting NMF are recommendation systems [5], feature extraction [6] and clustering [7]. In the following a brief description of how NMF is adapted to each application field is given:

- Recommendation system is used in the scope of users assigning ratings to items. Not all items are rated by all users. It aims to predict the rating that a user would give to an item. This problem can be formalized by creating a matrix of dimensions $|users| \times |items|$ holding in each entry the rating that the user has given to the item. This matrix presents unknown values. Considering it the T matrix, matrices X and Y are built exploiting known entries only. In such a way it is possible to build, by multiplying X and Y, the new matrix $\hat{T}$ filled with values that were unknown in T.

- Feature extraction is a technique used to reduce dimensionality of data. A data sample is described by a set of M features. The aim is to transform them in such a way that the same sample can still be completely characterized by a reduced number of features. In practice the problem is to project data from a M-dimensional space to a K-dimensional one, with $K < M$. Considering every sample a vector of features, T matrix is built by organizing samples as columns. In this way the resultant matrix is a $M \times N$ matrix (N is the number of samples). Columns of matrix Y can be considered as same records in T columns represented in a different feature space, where number of features is equal to K.

- Clustering aims to find common features conformation in data records. It agglomerates similar data in groups (clusters) and provides for each of them an element (centroid) that is promoted to representative of the cluster. Arranging again samples in columns, matrix T is composed by $M \times N$ entries, N is the number of samples and M the number of features. The idea behind the use of NMF for clustering lies in the fact that matrix X can be considered as a list of K centroids, one for each column. While Y matrix can be seen as an indicator of membership. So a sample T_i is member of a cluster with centroid X_j only if the element Y_{ji} indicates membership. Membership is represented by 1 value, while non membership by 0. This means that Y has only one non zero value per column. In such a way formula

in Equation 3.2 acquires the meaning of finding clusters centroids similar as much as possible to records that belong to them.

Chapter 4

Classification

In this chapter a high level definition of classification is given in Section 4.1. It follows a description of some classification methods adaptable to our study. Our dataset requires in fact the use of a multi-label classification, for each attribute there are typically more than two class labels, in such a case we would speak about binary classification . Features characterizing data are continuous. For this purpose suitable algorithms are: Decision Tree (Section 4.2), Neural Network (Section 4.3), Rule Based (Section 4.4), K-Nearest Neighbor (Section 4.5) and Support Vector Machine (Section 4.6).

4.1 Technique

Classification is defined as the task of associating a category/class label to a given input sample described with a set of features. A classifier is a model able to perform classification after learning from a set of training data.

4.2 Decision tree

Decision Tree is a classification technique that provides a tree shaped classifier. In leaf nodes are assigned the labels. Every other node expresses one (binary splitting) or more conditions (multiway splitting) on a feature. When an input sample is supplied to the classifier it follows a path of the tree from the root node to a leaf, according to conditions satisfied by the value of its attributes. The leaf in which it arrives provides its label. An

example of a simple Decision Tree trained model is reported in Figure 4.1.

There are different ways to build a Decision Tree, the most common is a top down inductive approach [8]. This approach is described in the following. The learning phase organizes all samples in the root node of the tree, that has initially only one node: the root. A feature and relative conditions to associate to the root node are chosen. For binary splitting two splits (partitions of the training set) are created: the first is composed by samples for which the condition is verified and the second one by samples for which it is not. For multiway one the number of splits is equal to the number of conditions, samples are assigned to splits depending on which of them is verified.

Purity index is the metric that is used to choose what is the best attribute and relative conditions to assign to the node. It is calculated on splits and expresses the uniformity of labels distribution inside them: the more samples present the same label the higher the index is. Some purity indexes have been summarized in [9]. A branch for each split is created and splits are routed in the appropriate branch. Branches terminate with a node, here the process starts again till one node is pure. Pureness means that in the node, all training samples are of the same label. This node is promoted as a leaf and the label of samples is assigned to it. The constraint of pureness can be relaxed by specifying a minimum value of purity index as acceptable for promoting the node as a leaf. In this case the majority class among samples is assigned to the node.

4.3 Neural network

Neural Network (NN) can be either a supervised and unsupervised machine learning technique. In this section the focus is given on supervised NN that are used for classification problems. NN is inspired by the organization of human brain. A neural network is composed by neurons and connections. Neurons are unit of calculations while connections link neurons each other. Neurons are organized in layers, each one in a layer is independent from all others in the same layer, this means that NN are suited for parallelizing calculations. Output layer, the last one, has a number of neurons equal to the number of classes, and has the aim to indicate the class of belonging of the input sample.

Learning of a Neural Network consists of properly assigning weights to connections on the basis of a loss function, that represents the error of prediction committed by the

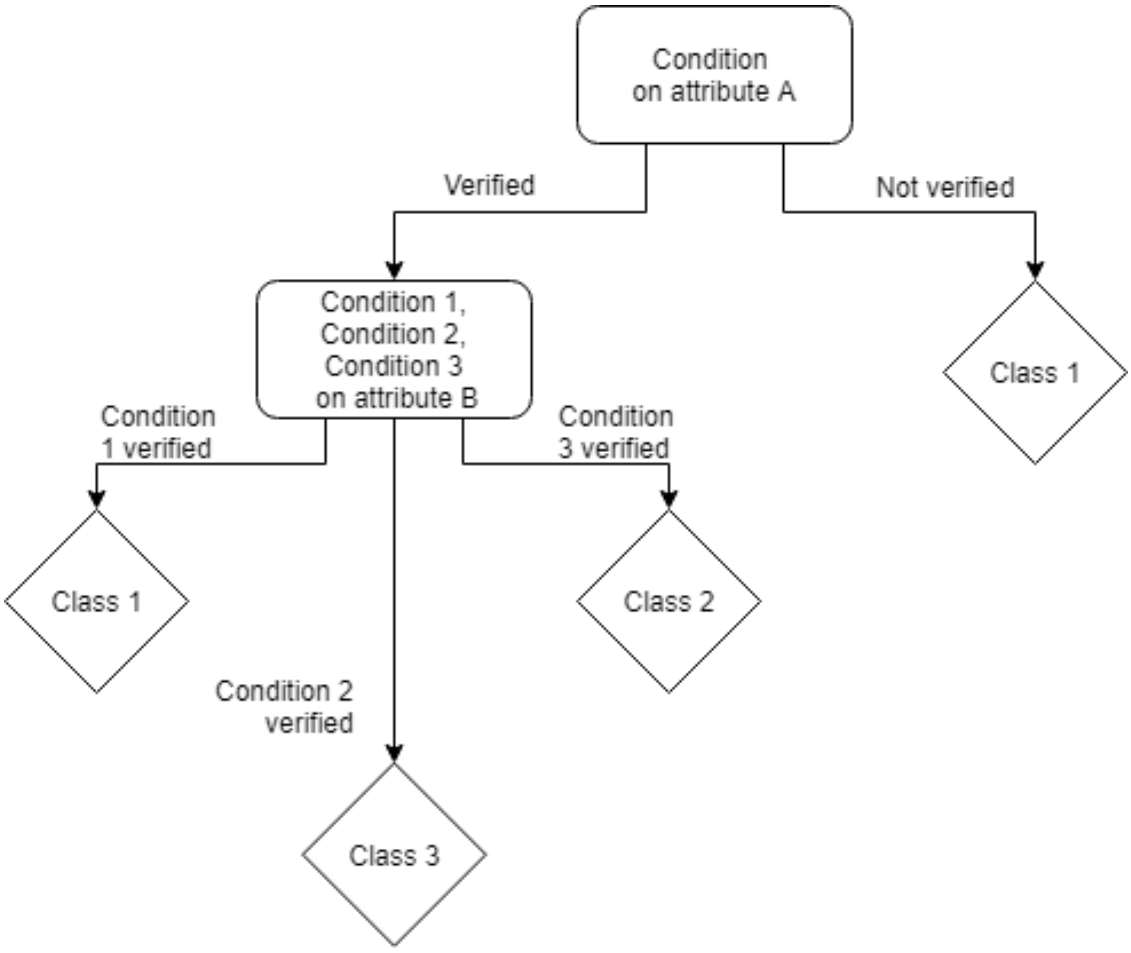


Figure 4.1: A Decision Tree trained model. It is the result of two splitting operations. The first node is an example of a binary splitting performed on attribute A. While the other condition node shows a multiway splitting on attribute B.

network for a given training sample. This process is iterative: the whole training set is seen more than one time by the network, and weights are updated according to the loss function till convergence or till a finite number of iterations.

A single layer Neural Network is able to properly classify data that are linearly separable, a two layers one can design convex decision boundaries, while a three layers one can potentially design any decision region [10].

4.4 Rule based

A Rule Based classifier is a classification model based on rules. A rule has the form

if <condition> than <class>

Condition is expressed on values of features. The model is composed by a database of rules. When a new record has to be labeled, a rule that covers it (a rule whose condition is satisfied by the new record) is searched in the database.
Rules can be:

- exhaustive: every sample is at least covered by one rule.
- mutually exclusive: every sample is at most covered by one rule.

If rules are not exhaustive a default label must be provided, because some records can potentially not be covered by any rule. If they are not mutually exclusive a system must be defined to decide which label of matched rules has to be assigned to the record.

Learning this kind of rules is a process called rules induction, in [11] are reported some algorithms able to perform it. One method consists of using a Decision Tree classifier (presented in Section 4.2): every path of the tree, from the root to the leaf is converted to a rule. Every rule is then composed by the logic AND of conditions in nodes of the path, according to the followed branches that lead to the leaf. Rules built in such a way are exhaustive and mutually exclusive.

4.5 K-Nearest Neighbor

K-Nearest Neighbor (KNN) is classification technique that exploits the concept of data similarity. A survey of similarity measures is done in [12]. Training set is used as a database. The classification of a new sample is provided by selecting the closest (in terms of similarity) K samples in the database. These records are called neighborhood. The class label assigned to the new sample is the label of majority in the neighborhood.

4.6 SVM

SVM, as already said in Chapter 2, can be used both for regression and classification. Learning phase of this type of classifier, seeing the problem in a $\mathbb{R}^N$ space, where N is the number of features, aims to find more hyperplanes that are able to separate all samples belonging to different classes. The way in which hyperplanes are calculated aims to maximize the distance between hyperplanes and samples coordinates. In particular during learning, SVM creates a margin around the hyperplane inside which no point is likely to lie. This margin must be maximized during the learning phase. Obviously, some margin of misclassification is tolerated by the algorithm. The operation required for learning is the dot product between vectors, considering samples vectors of features. Hyperplanes based separation requires every class in the dataset to be linearly separable from each other. If it is not the case SVM define some kernel functions that provides the definition of dot product in a space where these are linearly separable.

A class label is associated to a new sample according to its position in the space, i.e. in which area delimited by hyperplanes it falls.

Chapter 5

Discretization

In this chapter is presented discretization as a procedure finalized to machine learning. In Section 5.1 is reported its purpose and the distinction between supervised and unsupervised discretization is clarified. The dataset we exploit is not suitable for applying supervised one, because there are no class labels and the attribute we want to discretize is the one to be predicted. For this reason only unsupervised techiniques are reported in Section 5.2.

5.1 Technique

Discretization is a data transformation technique that is applied to reduce data domain and to make data conform to the use of algorithms that require categorical attributes. Discretization operates on continuous features. The aim is to provide some bins/categories/integers that became the new domain of a feature. In practice domain of the feature migrates from a continuous to a discrete one. These discrete counterparts can then be interpreted as intervals or categories.

Discretization can be supervised or unsupervised [13]. The former is done by only considering feature values distribution among records, while the latter takes into account also the class label of records. In the last case it is possible to create groups of samples, that will be associated to the same discrete value, by optimizing a function dependent from labels distribution inside the groups. This provides a discretization that is finalized to improve performances of a later performed classification.

5.2 Equal Width and Frequency binning

Binning is an unsupervised discretization technique that is used to create bins out of a continuous values distribution. Bins are intervals of feature values. After discretization every sample is assigned to a bin, that represents the new value of the new discretized feature. A sample is assigned to a bin if its non discretized feature value falls inside of that interval. A parameter of binning discretization is the number of bins to create: N, that is also the dimension of the new discretized feature space. Feature continuous domain is divided into N intervals. There are two ways of performing this division:

- **Equal width binning** provides bins having the same width. Suppose that values assumed by the feature (x) are contained in a domain limited by $l, r \in \mathbb{R}$, $x \in [l, r]$. Equal width binning generates bins of width $\mathrm{ceil}\left(\frac{r-l}{N}\right)$. The number of samples per bin is dependent from the distribution of feature values.

- **Equal frequency binning** generates bins in such a way that the number of records per bin is equal. Bins have a number of samples equal to $\mathrm{floor}(M/N)$ or $\mathrm{ceil}(M/N)$, with M the total number of samples. The width of these bins is dependent from where is necessary to cut the initial interval for having the defined number of samples per bins.

Equal width maintains initial data distribution of samples while frequency binning makes uniform the final data distribution.

Part II

Implementation

Chapter 6

Method

In this chapter, the practical implementation of our solution is described. In Section 6.1, a detailed description of the two datasets we have at disposal is given, underlying difficulties related to them and relations existing between petrographical and mineralogical attributes. In Section 6.2, with reference to the analysis on regression and classification done in Chapters 2-4, are reported which techniques we have chosen to be tested on the datasets and why. In Section 6.3 is depicted the process that, starting from the row data, arrives to obtain prediction results evaluable with metrics that will be described in Section 6.6. The discretization methodology we designed for obtaining class labels is explained in Section 6.4. The use we have done of NMF procedure for solving the regression problem is clarified in Section 6.5

6.1 Dataset

The dataset under analysis is composed of NS samples ($NS = 62$) presenting a set of mineralogical attributes (M) and petrographical features (P). Petrography and Mineralogy are two different ways of expressing the chemical composition of a geological sample. Petrography expresses composition in terms of rocks, while mineralogy in terms of minerals. Each petrographical and mineralogical attribute represents, respectively, a rock or a mineral. Features domain are floats between 0 and 100; they express the percentage of the feature present in the sample, with respect to its total composition. This means that, for a given sample, the sum of all values of the petrographical attributes is

100 and the same property applies to the mineralogical ones:

$$\sum_{i \in M} x_{ki} = 100 \quad \text{and} \quad \sum_{j \in P} y_{kj} = 100 \tag{6.1}$$

where x_{ki} and y_{kj} are the values assumed by i mineralogical and j petrographical attribute for sample k, respectively. A graphical representation of the dataset structure is reported in Figure 6.1. The presented notation is used in all the rest of the thesis when possible.

Each attribute value is affected by an absolute measurement error around 8%, mainly due to the precision of the machinery used to extract composition from the geological sample. In the rest of the thesis the percentage symbol is always omitted when values of attributes and errors are reported.

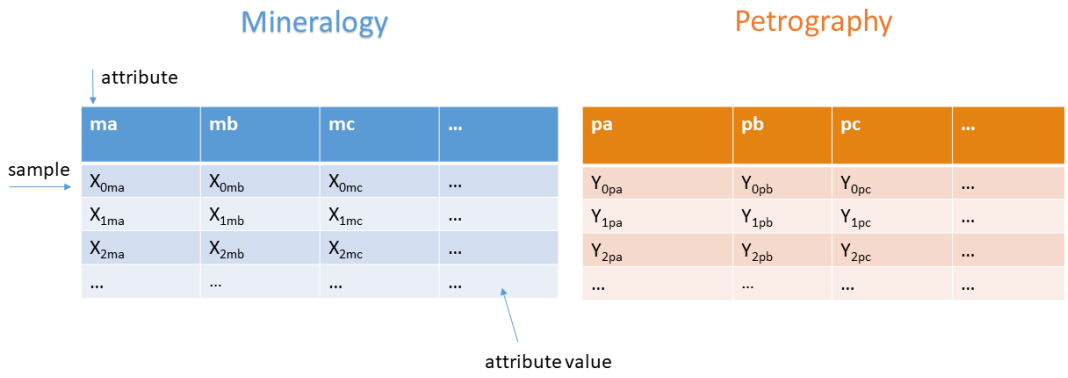


Figure 6.1: Portions of mineralogy and petrography matrices reporting notation and terminology.

The dataset is provided in two versions. The number of attributes of mineralogy, for both versions, is $|M| = 10$. For petrography, instead, some rocks are considered grouped together in one version, with respect to the other, this means that it is a less informative version. The grouped dataset version will be called "aggregated dataset" while the second will be named "non aggregated dataset". The latter version has a number of petrographical attributes equal to $|P| = 26$, while the former has only $|P| = 7$. Aggregation means that

some attributes are collapsed together by summing up their values:

$$Y_{ij} = \sum_{k \in AR_j} y_{ik} \tag{6.2}$$

Y_{ij} is an element (attribute j value for i-th sample) of the aggregated dataset. AR stands for Aggregation Rule. There is one aggregation rule for each attribute of the aggregated dataset. Each aggregation rule is expressed by a set of non aggregated attributes, that are going to compose the aggregated one. Aggregation rules are reported in the following:

$$\begin{aligned}
AR_{PA} &= \{pa1, pa2, \ldots, pa8\} \\
AR_{PB} &= \{pb1\} \\
AR_{PC} &= \{pc1\} \\
AR_{PD} &= \{pd1, pd2, pd3, pd4\} \\
AR_{PE} &= \{pe1, pe2, pe3\} \\
AR_{PF} &= \{pf1, pf2\} \\
AR_{PG} &= \{pg1, pg2, \ldots, pg7\}
\end{aligned}$$

As it is possible to see some attributes are present in both datasets. These are called *PB*, *PC* in the aggregated dataset and *pb1*, *pc1* in the non aggregated one.

During this thesis if no specification is provided the dataset in analysis has to be considered the non aggregated one.

Notice that a rock composition can be expressed in terms of minerals, because rocks are agglomerates of minerals. Unfortunately there is no chemical relation that links petrographical composition to mineralogical one. However, a partial relation exists because in each rock type can only be present a limited set of minerals. Given this observation the problem could migrate to a different formulation: how a mineralogical attribute spreads into petrographical attributes it is allowed to. In the following a formalization of these relations is given.

Attributes are organized in 5 groups, identified through colors. Minerals in a group can only spread into rocks of the same group. This leads to the following property: summing up all the values of mineralogical attributes belonging to a group the result should be equal to the sum of all the petrographical attributes of the same group. In formula:

$$\sum_{i \in M_g} x_{ki} = \sum_{j \in P_g} y_{kj}, \quad \forall g \in \{purple, green, blue, orange, red\} \tag{6.3}$$

where k is a specific sample, M_g and P_g are respectively the sets of attributes of mineralogy and petrography belonging to the g-th group of attributes. To clarify the correspondence between colors and attributes, the M_g and P_g sets are listed below:

$$M_{purple} = \{ma\}$$
$$M_{blue} = \{mb, mc, md\}$$
$$M_{green} = \{me, mf\}$$
$$M_{red} = \{mj\}$$
$$M_{orange} = \{mg, mh, mi\}$$

$$P_{purple} = \{pa1, pa2, \ldots, pa8, pb1, pc1\}$$
$$P_{blue} = \{pd1, pd2, pd3, pd4, pe1, pe2, pe3\}$$
$$P_{green} = \{pf1, pf2\}$$
$$P_{red} = \{pg7\}$$
$$P_{orange} = \{pg1, pg2, \ldots pg6\}$$

As we will see in the next section, these kind of relations are unfortunately non exploitable for our study.

6.1.1 Consideration on error of measurement

It has been decided not to exploit properties reported in Equation 6.1 and 6.3 because of an evidence that came up during a dataset exploration. In fact the relation in Equation 6.3 is not satisfied for all samples, i.e.:

$$\sum_{i \in M_g} x_{ki} - \sum_{j \in P_g} y_{kj} = \Delta_{kg} \quad \text{with} \quad \Delta_{kg} \neq 0 \tag{6.4}$$

where Δ_{kg} specifies how much, for a given group g and a given sample k, the sum of petrographical attributes differs from the sum of mineralogical attributes.
Boxplots of Δ_{kg} distributions can be observed in Figure 6.2. It is possible to see that for the *green* group, 75% (III quartile) of samples present a Δ_{kg} under 10, while the largest Δ_{kg} is above 20. An explanation of this evidence can be that propagating the measurement error of individual attributes values to their sum leads to a more significant uncertainty. It is not trivial using these sum relationships between mineralogy and petrography for the predictions, because they are extremely affected by measurement errors. In fact they

can totally mislead predictions. For now, these constraints based on sums have not been taken into account, but they will be exploited for future analysis, with the support of a domain expert.

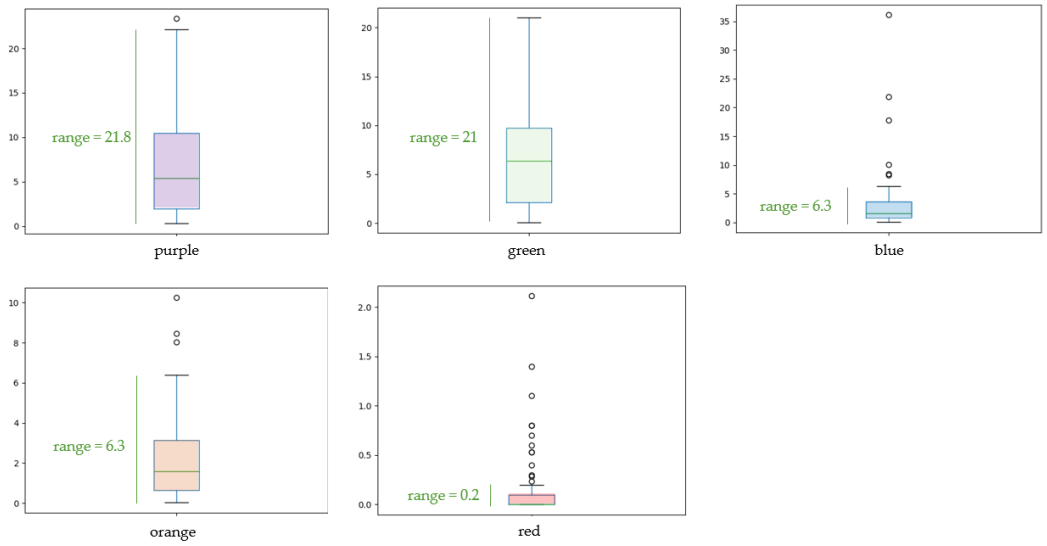


Figure 6.2: Δ_{kg} boxplots, see Section 6.1.1 for explanation. Each boxplot shows how values are distributed (I, II, II and IV quartiles). Black dots correspond to isolated values and are called outliers.

Table 6.1 reports petrographical attributes that in the whole dataset never assume a value greater than the error of measurement (8). Values for all these attributes are smaller than the measurement error and this characteristic makes difficult to evaluate the prediction quality. Hence the predictions for these attributes will not be considered in the following analyses. For these attributes it is needed further domain knowledge such as specific (possibly lower) measurement errors or whether their prediction is fundamental from a domain expert perspective.

Attribute	Maximum Value
pa3	7.98
pa4	2.53
pa5	2.02
pa6	0.53
pa7	1.08
pa8	5.89
pd2	4.65
pd3	2.71
pe2	4.01
pf2	1.59
pg1	1.55
pg2	1.54
pg3	1.08
pg4	2.68
pg7	2.11

Table 6.1: Maximum values for attributes whose distribution stays under 8.

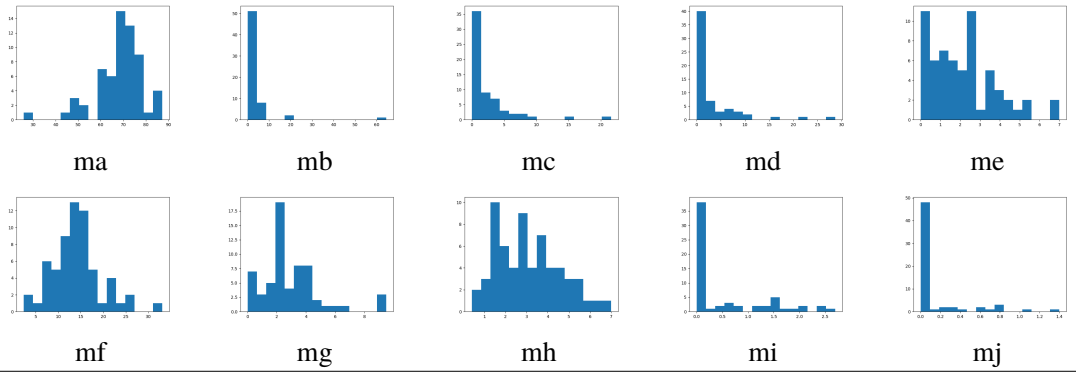
6.1.2 Distribution

Distribution presented by values of mineralogical and petrographical attributes is substantially of two types: exponential or gaussian. These distributions however span over different values. This means that not all attributes have the same range of variation, they tend to assume values always in an interval that it is proper of the attribute. As already seen in Table 6.1, there are attributes whose values always ranges under 8 (the measurement error). For completeness maximum value of remaining petrographical attributes is reported in Table 6.2. In Table 6.3 are contained histograms presenting datasets attributes distribution. Examples of gaussian distributions are: *mf*, *PF* and *pf1*; while exponential ones are: *md*, *PB*, *pd4*.

Attribute	Maximum Value
pa1	66.80
pa2	10.58
pb1	43.47
pc1	19.89
pd1	8.76
pd4	40.33
pe1	11.54
pe3	16.06
pf1	40.18
pg5	11.46
pg6	9.62

Table 6.2: Maximum values for attributes whose distribution spans over 8.

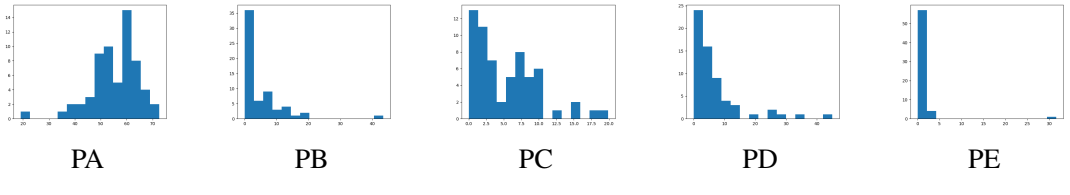
Mineralogical attributes distribution



ma mb mc md me

mf mg mh mi mj

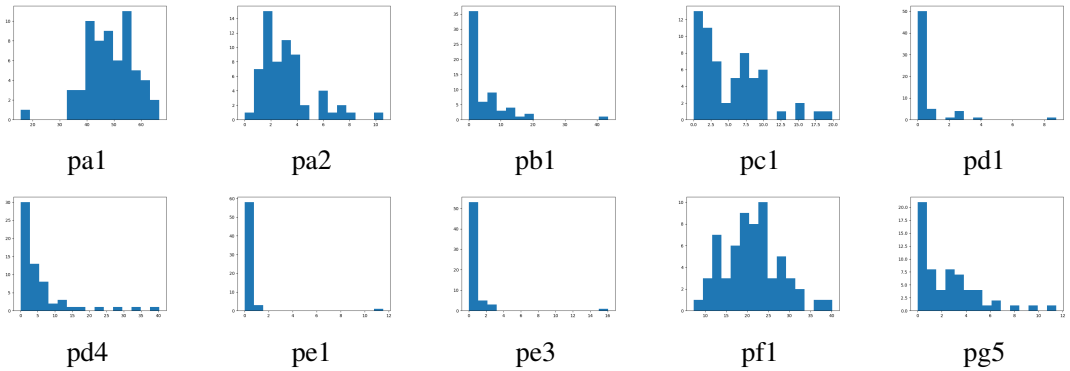
Aggregated petrographical attributes distribution



PA PB PC PD PE

PF PG

Petrographical attributes distribution



pa1 pa2 pb1 pc1 pd1

pd4 pe1 pe3 pf1 pg5

pg6

Table 6.3: Distributions of attributes.

6.2 Prediction techniques selection

The aim of our study is to predict, from a set of continuous features, a set of continuous values i.e. from mineralogical attributes, predict petrographical ones. We decide to differentiate our approach by exploiting both regression and classification techniques and to apply a NMF based approach. This latter approach is described in Section 6.5.
Since values to predict are continuous, for classification approach is necessary to opportunely discretize the attributes to predict, to use discretized values, coded as intervals, as class labels. The process of discretization is described in Section 6.4.
Discretization and regression models are applied separately for each petrographical attribute, while NMF aims to capture also internal relations between petrographical attributes, only one model for all attributes is built.

In Chapters 2-4 common regression and classification algorithms applicable to our dataset have been described. In this section is reported which of them have been selected for the problem in analysis explaining the main reason why they have been chosen.
Substantially our datasets present two challenges:

- **Small dataset**, only 62 samples with 10 features. Small with respect to the suggested number of samples to be exploited for a regression problem [14] and for a classification one [15].

  The main problem in applying machine learning to a small dataset is overfitting [16]. There is not a way to surely avoid overfitting, but in general the model has to be as simple as possible, to avoid that a complicate model adapts too much to training data. For such a reason we decided to exploit Linear regression, supposing a linear dependence of predicted variable from predictors. To avoid overfitting, regularization techniques Lasso and Ridge have been selected. For exploring a non linear relation we opted for SVR with a Radial Basis Function (RBF) kernel, which is a kernel composed by an infinite sum of polynomial kernels. Polynomial kernel is used when a polynomial relation of defined grade between y and $\vec{x}$ is supposed. Hence, RBF could be a complete way to exhaustively evaluate a polynomial dependency. We try to use as classifier the Decision Tree, because of its interpretability and the possibility to extract association rules from it, that can be analyzed and eventually modified by a domain expert.

- **Non uniform distribution** of predicted variable and, as a consequence, non balanced distribution of class labels. This can be a problem for classification tasks as stated in [17]. Theoretically there is no problem for linear regression, as no particular assumption is done on the distribution of y [18]. For classification this problem is typically approached in two ways when no other data can be easily retreived [17]:
  - undersampling, that means reduce the number of labels of the majority classes to reach balance.
  - oversampling, that means replicate samples belonging to the minority classes, with the same purpose of reaching class balance.

  We have truly too few data to apply an undersampling strategy because, especially for exponential distributed attributes, some class labels are presented by a very limited number of samples (such as 1 or 2). In this situation undersampling for obtaining such a number also for majority classes would lead to a very poor resulting dataset. Oversampling strategy presents a problem too. With such a few samples per class label, replicating them will probably prevent the model to properly generalize rules for classifying them. In conclusion none of these two techniques have been applied; but this problem has been addressed in the process of discretization (Section 6.4), by enlarging ranges of intervals to create more samples belonging to the poorly populated labels.

6.3 Process overview

Summarizing what has been stated in Section 6.2, machine learning techniques that have been applied to both aggregated and non aggregated dataset are:

- Regression
  - Linear
  - Lasso
  - Ridge
  - Support Vector Machine
- Classification
  - Decision Tree
- NMF

An important phase of our work is the comparisons between these techniques. We have used some metrics, as we will see in Section 6.6, to evaluate results (predictions performed for the test set) of different techniques and to quantitatively and graphically compare them. As we have a small number of samples, we use a leave one out approach, in this way the number of samples used for training the model is maximized to avoid overfitting. In leave one out process NS models are created, each one trained with $NS-1$ samples and tested with the remaining one. For every model the sample used for testing is different. At the end of leave one out process a new database with all predictions for every sample is created. Figure 6.3 depicts the outline of the process. This pipeline, for regression and classification techniques, has been repeated for each petrographical attribute in the dataset. While for NMF this is applied only one time, because NMF model predicts all petrographical attributes jointly. Notice that for the Decision Tree classifier a preprocessing step is required, in fact attributes of petrography must be discretized before applying the whole pipeline.

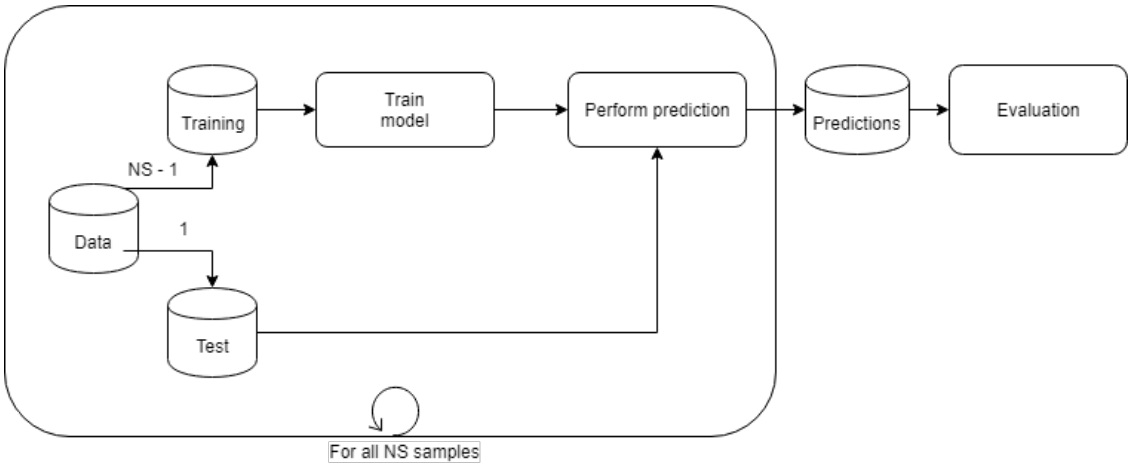


Figure 6.3: Leave one out strategy.

6.4 Discretization

Consider that petrographical values to predict are continuous. As we said in Section 6.2 we would like to exploit a classification technique for our study. A classifier is able to learn and predict samples labels, not continuous values. For such a reason petrographical attributes have been discretized to find ranges to be used as labels for the Decision Tree. Every attribute has its own intervals. Instead of using standard discretization techniques reported in Section 6.4 it has been designed a method useful to consider both the non uniform data distribution and the error of measurement on values of the attributes. Our custom discretization procedure is composed of two phases:

1. creation of uniform bins
2. collapse of bins

The first phase consists in creating for each attribute a number of bin equal to:

$$NB = \text{ceil}\left(\frac{y_{\text{max}} - y_{\text{min}}}{e}\right) \tag{6.5}$$

where e is the measurement error, y_{max} and y_{min} are respectively the maximum and minimum petrographical value in the dataset for that attribute. This means that the number of bin is dependent on the error of measurement.
An equal width binning discretization is applied (refer to Section 5.2), by using NB as

number of generated bins. This phase may lead to empty or not much populated bins because of non-uniform values distributions. Notice that a Decision Tree classifier can predict a label only if it is present in the training set. For this reason bins made of only one sample and empty bins cannot be used in the classification process. The motive is evident for empty bins. For one-element bins the reason is linked to leave one out approach (Section 6.3). When the sample belonging to one-element bin is used in test set its label is not comparing in the training set; the Decision Tree built in such a way does not present the interested label in leaves and so it has not the possibility to properly classify the test sample. Hence, not less than two samples per bin must be present in the dataset. In this way at least one sample of the two is present in the training set for each classifier generated in leave one out approach. Notice that the more samples are present in a bin the more likely the model will correctly classify them, because it has more examples for learning.

The solution to the sparse distribution of data, which entails the presence of bins with less than two samples, is to collapse them, creating larger and more populated intervals. This is performed by the second phase of the discretization process. Bins are collapsed only when necessary, trying to maintain data values as more differentiated as possible. In our process poorly populated bins are collapsed together with the less populated adjacent bin. Collapsing two adjacent bins means expanding the first one in such a way that it includes the second one:

$$\begin{gathered} B1 = [l_1, r_1) \\ B2 = [l_2, r_2) \\ B1 \text{ and } B2 \text{ are two adjacent bins} \\ \text{adjacency imposes} \quad r_1 = l_2 \\ \text{the resultant collapsed bin } B12 \text{ is:} \\ B12 = [l_1, r_2) \end{gathered} \tag{6.6}$$

A pseudocode of the collapsing procedure is reported in Algorithm 1. The input parameter $mins$ represents the minimum number of samples that are required for not collapsing a bin. It is tunable and controls the trade off between number of sample in bins and the width of the intervals.

After applying the discretization method some attributes present only one bin. The problem with this type of attributes is that it makes no sense to build a classifier, since

there is only one class. The creation of only one bin can happen in both discretization phases:

1. If the width of attribute range of values is, for all samples, minor or equal to the measurement error. The first phase, in this case leads to the creation of only one bin because starting from Equation 6.5:

$$0 \leq \frac{y_{\max} - y_{\min}}{e} \leq \frac{e}{e} \tag{6.7}$$

   On this quantity the ceil function is applied, leading to $NB = 1$

2. The first phase terminates with the creation of more than one bin, but the bins collapse cause only one resulting interval. This happens because condition in line 5 in the pseudocode (Algorithm 1) is never satisfied for all iterations: less populated bin, at each iteration does not reach the minimum number of samples for not being collapsed.

In Figure 6.4, is shown the result of the discretization process on attribute *pd4*.

Algorithm 1 Collapsing procedure

Require: $mins \geq 2$
1: $nb \leftarrow$ number of bins
2: **while** $nb > 1$ **do**
3: $b \leftarrow$ bin with less samples
4: $ns \leftarrow$ number of samples of b
5: **if** $ns \geq mins$ **then**
6: **return** ▷ all bins have a sufficient number of samples
7: **else**
8: $bc, bd \leftarrow$ bins adjacent to b
9: $ba \leftarrow$ who has minimum number of samples between bc and bd
10: collapse b and ba
11: $nb = nb - 1$

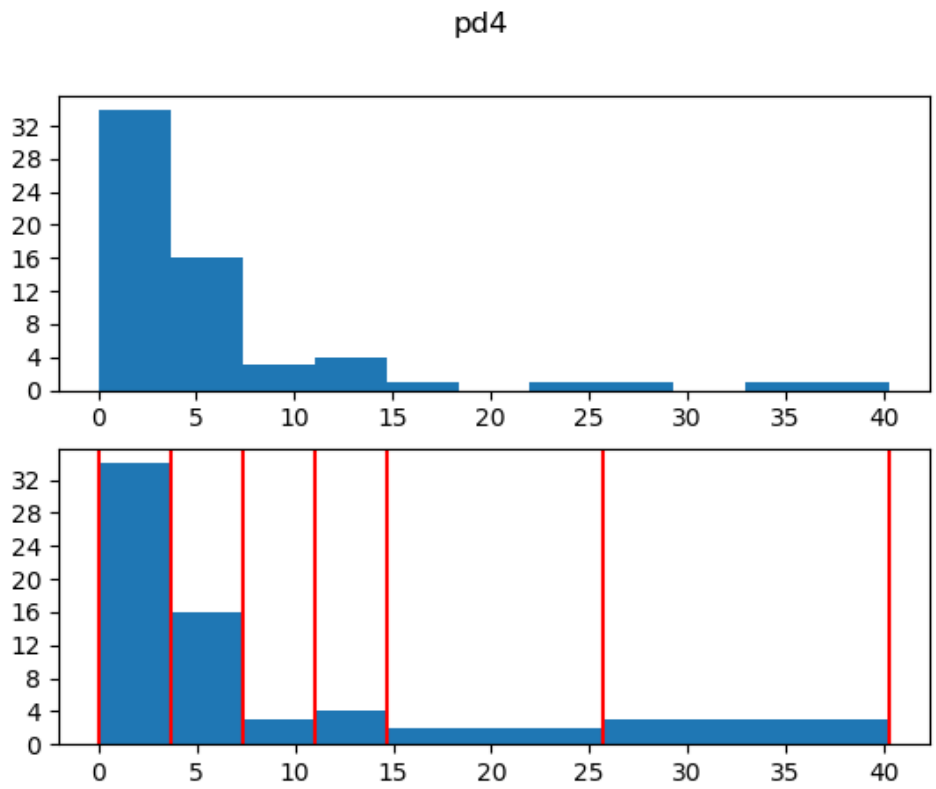


Figure 6.4: In the first histogram is reported the distribution of *pd4* with a bin size equal to 4. In the second one red lines indicate borders of bins deriving from discretization process for *pd4*, on y-axis is reported the number of samples belonging to the bin. Considered error is 4, the minimum number of samples per bin is 2.

6.5 NMF approach

Some state of art applications of NMF have been summed up in Section 3.2. We exploit NMF procedure for performing regression. Refer to Equation 3.1 for notation. We have built matrix T by inserting training samples as rows. Samples are to be intended as vectors of values of attributes, first mineralogical then petrogrephical ones.

$$\vec{t_k} = (x_{i,ma}, x_{i,mb}, \ldots, x_{i,mj}, y_{i,pa1}, y_{i,pa2}, \ldots, y_{i,pg7}) \tag{6.8}$$

i is sample identifier. $\vec{t_k}$ is the k-th row vector in matrix T. The insertion of all training samples leads to a T matrix that is made by $NS - 1$ rows and $|P| + |M|$ columns. Given

the test sample with id j, for which to predict petrographical values, its mineralogical values are added to the matrix, while petrographical are left unknown:

$$\vec{t}_{NS-1} = (x_{j,ma}, x_{j,mb}, \ldots, x_{j,mj}, ?, ?, \ldots, ?) \tag{6.9}$$

Now, the same procedure exploited for recommendation system application field is applied (Section 3.2). Matrix $\hat{T}$ is calculated such that $T \approx XY = \hat{T}$ and it holds predictions for test sample in elements $\{T_{NS-1,i}\}_{i=|M|}^{|M|+|P|-1}$. Notice that this kind of procedure forecasts all petrographical attributes jointly. This could allow to express other relations among attributes, such as ones listed in Section 6.1, adding some constrains to the minimization problem of NMF.

This approach is not suitable for a big T matrix, because every time a prediction is needed, computational time for solving NMF problem (Equation 3.2) is required. Our small dataset is totally suitable for this kind of approach.

6.6 Metrics

In following subsections are briefly presented and described quality metrics used to evaluate and compare results obtained by algorithms we applied on the datasets. In particular in Subsection 6.6.1, state of art used regression metrics are listed. In Subsection 6.6.2 the same analysis is applied for classification metrics. Instead, in Subsection 6.6.3 are reported some custom metrics we have defined for evaluating the behaviour of selected regression techniques when facing rare values. In the end (Subsection 6.6.4) some considerations have been done on the strategy we adopted to compare regression and classification results.

6.6.1 Regression Metrics

The quality metrics that have been taken into account for evaluating and comparing selected regression techniques (remarked in Section 6.2) are: **MAE** (Mean Absolute Error), **EVS** (Explained Variance Score) and some custom metrics that will be detailed in Subsection 6.6.3.

MAE

MAE is calculated both per attribute and globally. It is the mean of absolute errors made in predicting values. More formally:

$$MAE_j = \frac{\sum_{i=0}^{NS-1} |\hat{y}_{ij} - y_{ij}|}{NS} \tag{6.10}$$

$$global\ MAE = \frac{\sum_{j \in P} MAE_j}{|P|} \tag{6.11}$$

where MAE_j is the MAE for the j attribute, $\hat{y}_{ij}$ is the predicted value for sample i and attribute j, y_{ij} is the actual value (target), NS is the number of samples in the dataset. The best value reachable by MAE is 0 and it is obtained when all predicted values are equal to the actual ones.

EVS

EVS is a metric defined per attribute. It expresses the ability of the model in predicting variations of attribute values. The best value that this metric can assume is 1. Lower values (can also be negative) mean worse quality of the predictions. If the error $|\hat{y}_{ij} - y_{ij}|$ of all the predictions ($\forall i$) for a j attribute is constant, the EVS will assume the value 1. This because adding an offset to the predictions of the model would give the correct expected values. The EVS mathematical definition for a given attribute is:

$$\text{EVS} = 1 - \frac{\text{Var}(y - \hat{y})}{\text{Var}(y)} \tag{6.12}$$

6.6.2 Classification Metrics

As we have an high class imbalance, we decide to exploit metrics as **precision**, **recall** and **F-measure**, that offer a per class quality evaluation under different perspectives.
Considering one class label, a record correctly predicted (its assigned class is equal to the actual class) is a **true positive**. When the predicted class is wrong, the sample is a **false positive** for the assigned class, and a **false negative** for the class it actually belongs to. TP/FP/FN are the sum of all true positive/false positive/false negative for a given class.

- *precision* is used to evaluate if the considered class label is assigned to records which effectively belong to it. It ranges from 0 to 1. 1 is the best case: all records

that are assigned to the considered label are correct. More formally:

$$\frac{TP}{TP + FP} \tag{6.13}$$

- *recall* is used to evaluate if records, which actually belong to the considered class, have been correctly predicted. It ranges from 0 to 1. 1 is the best case: all records actually belonging to the label are correctly predicted.

$$\frac{TP}{TP + FN} \tag{6.14}$$

- *F-measure* is used to evaluate together precision and recall, having only one metric, that goes from 0 to 1 (the best case):

$$\text{F-measure} = 2 \cdot \frac{\text{Recall} \cdot \text{Precision}}{\text{Recall} + \text{Precision}} \tag{6.15}$$

Trying to generate models which maximize recall would penalize precision. Indeed, if many items are recalled by a specific class, the likelihood of having false positives in the same class is higher (lower precision). Hence, good quality classifiers should maximize the trade off between these two metrics.

Confusion matrix is used to report visually the results. There is one confusion matrix per attribute. It is a matrix that has for rows actual classes, while columns are predicted classes. The values in each cell of the matrix are the number of records with the corresponding actual and predicted classes. Hence, a good confusion matrix has all the non zero values on the diagonal or closer to it.

6.6.3 Custom Metrics

Accuracy is a metric typical of classification, that we haven't exploited because its value is strongly polarized towards performances obtained by the most populated class. We propose a slightly modified version of the standard accuracy metric that can be applied to evaluate also regression algorithms. First of all here is reported standard accuracy definition according to notation defined in Subsection 6.6.2:

$$\frac{\sum_{l \in L} TP_l}{\sum_{l \in L} (TP_l + FP_l)} \tag{6.16}$$

This formula calculates the fraction of correct predictions with respect to the total number of predictions. L is the set of labels. TP_l and FP_l are respectively true positive and false positives for l label. Accuracy can assume values between 0 (worst case) and 1(best case). Our aim is to build a metric for regression that has the same meaning, which models the measurement error on attribute values. For example, with a measurement absolute error of $\pm 4\%$, if the prediction is 17.5% and the actual value is 18%, this value has to be considered coherent, because the distance between the predicted and the actual value is lower than the measurement error. This reasoning can also be seen as a way of considering correct a prediction with a certain tolerance. We consider correct values those included in a certain interval of error. A new definition of correctness is given:

$$\text{if} \quad y_{ij} - e \leq \hat{y}_{ij} \leq y_{ij} + e \quad \longrightarrow \quad \hat{y}_{ij} \text{ is a correct prediction} \tag{6.17}$$

e is the tolerance factor, if prediction error is smaller than it the prediction is considered correct.
For every attribute j the **regression-accuracy** is:

$$\text{regression-accuracy} = \frac{\text{number of correct predictions}}{\text{total number of predictions}} \tag{6.18}$$

where correctness has to be intended as defined in Equation 6.17. Notice that this metric, such as classification accuracy, doesn't permit to evaluate how the model behaves for rare values.

For this reason, we defined other two metrics related to correctness defined in Equation 6.17: **regression-precision** and **regression-recall**. Regression-precision is obtained by plotting the distribution of *predicted values*, in two different colors. The first color (blue) represents correct predictions. The other (wrong) predictions are plotted in orange. Regression-recall is obtained by plotting the distribution of *actual values* and presenting again two different colors for correct and wrong predicted values. An example of these charts are shown in Figure 7.19 and Figure 7.20 in Chapter 7.

Regression-precision allows inspecting the number of correct predictions along the distribution of the predicted values. Instead, regression-recall allows inspecting the number of correct predictions along the distribution of the actual values. Regression-precision distribution of values must be as similar as possible to the distribution of actual values. This shows that the model is able to properly capture attribute variation, avoiding

to predict all values in the same range or assigning samples to totally non-coherent ranges, such as ranges even not present in the initial distribution. Regression-recall instead is more intuitive because presents the same distribution of actual values and immediately identifies ranges which model has difficulties to predict.

6.6.4 Regression and Classification comparison methodology

There are no common metrics for directly comparing regression and classification methods; in fact regression predictions are continuous values, while classification predictions are labels. Therefore results of regression must be transformed to permit a comparison with classification results, or vice versa. It has been decided to discretize regression predictions exploiting same bins used for classification. Hence, the metrics used for the comparison are the ones typical of classification: precision, recall, F-measure and confusion matrix.

It is important to observe that this approach introduces a particular issue. Suppose to have a discretization with two class labels (l_1 and l_2) for a specific petrography attribute. The actual value y to be predicted is assigned to one of the two labels following this logic:

$$y_1 < y \leq y_2 \Rightarrow y \in l_1$$

$$y_2 < y < y_3 \Rightarrow y \in l_2$$

Consider the case where y assumes the following value:

$$y = y_2 + \frac{\epsilon}{2} \Rightarrow y \in l_2$$

where ϵ is a real value greater than 0 and smaller than e (error of measurement). In this case the value y is assigned by discretization process to l_2 label.
Then if the associated prediction is:

$$\hat{y} = y_2 - \frac{\epsilon}{2} \Rightarrow \hat{y} \in l_1$$

The prediction is assigned to l_1 label.
Calculating the difference between predicted and actual value the relation is:

$$y - \hat{y} = \epsilon$$

$$y - \hat{y} < e$$

This means that even if $\hat{y}$ belongs to l_1, the difference with the actual value is minor than the error of measurement. It is a correct prediction considering correctness as defined in Equation 6.17, but for a classification point of view the prediction is wrong. From this reasoning Decision Trees giving better predictions using this method of comparison do not necessary entail better quality. Conversely, if better results are given by regressors, regression gives necessary better results than classification.

Chapter 7

Experiments

In this chapter performances of all methods applied on the datasets are compared, with the aim of identifying the best one. A first analysis concerning how to compare results obtained on aggregated dataset with outcomes reached on non aggregated one is done in Section 7.1. In all other sections instead, the analysis is totally focused on outcomes obtained on non aggregated dataset, as we observe that results for the two datasets have the same quality. In particular for each aggregation group has been identified a leading attribute (belonging to non aggregated dataset) whose predictions can be easily compared with the ones obtained for the exponent of aggregated dataset in the same aggregation group. In Section 7.2, Decision Tree performances on leading attributes are discussed. In Section 7.3, best performing regression methods are pointed out. In Section 7.4, Decision Tree results are compared with the ones of best performing regression techniques (identified in Section 7.3).

7.1 Aggregated vs non aggregated dataset

In this section is performed a comparison between results achieved with aggregated and non aggregated datasets by regression techniques.

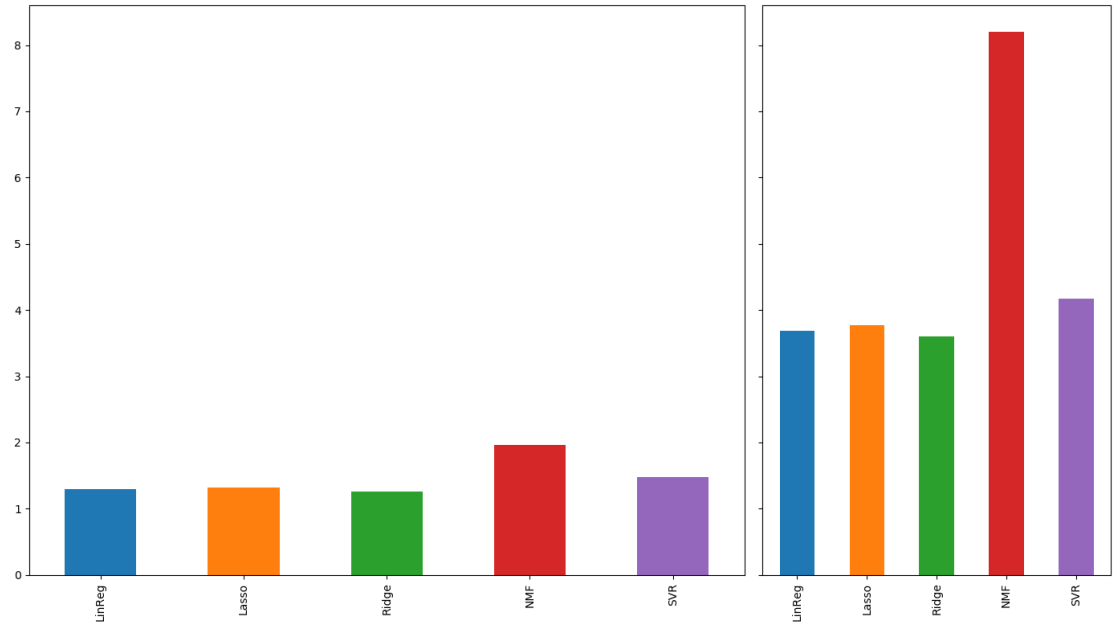


Figure 7.1: Global MAE for different regression techniques. On the left side the MAE is referred to non aggregated dataset, while on the right MAE is referred to aggregated one.

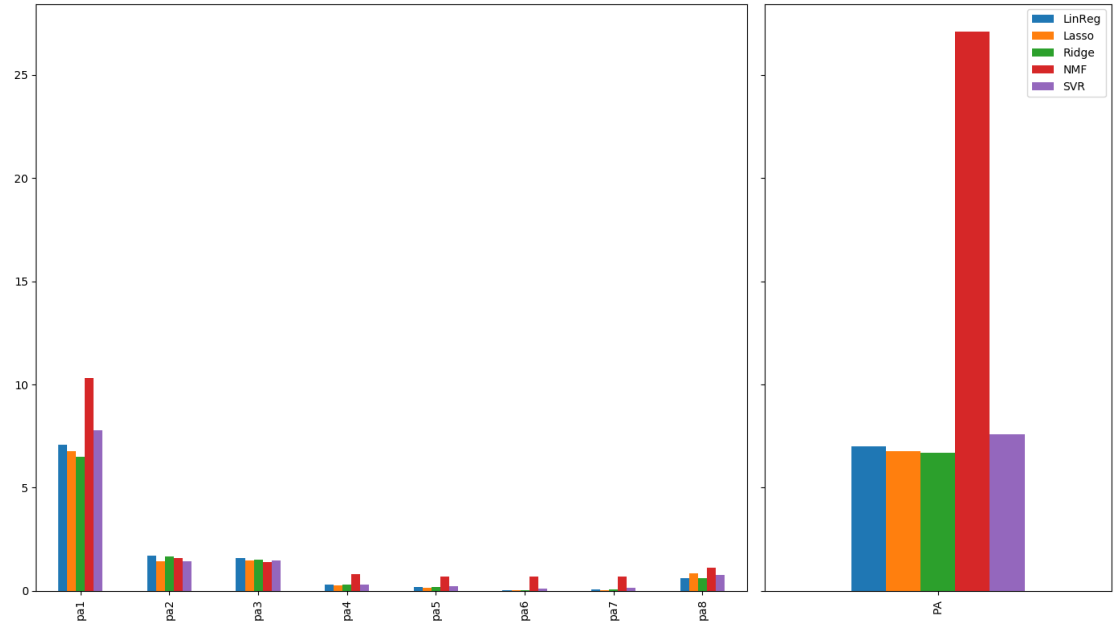


Figure 7.2: On the right is reported the MAE for the attribute *PA*, part of the aggregated dataset. On the left is reported MAE for all attributes composing *PA*.

At a first glance, by looking at the global MAE (Fig. 7.1) the aggregated dataset seems to give worst results with respect to the non aggregated one. Indeed, MAE values for non aggregated dataset are closer to zero.

This observation can be overturned when looking at Figure 7.2. It shows the detail of the MAE score for the attributes which are aggregated in the *PA* group. The MAE of *PA* in the aggregated dataset (right part of Figure 7.2) is comparable to the one of *pa1* in the non aggregated dataset. Instead, the MAE of all the other non aggregated attributes inside the group *PA* are far lower. When computing the global MAE on the non aggregated dataset, all these attributes with low MAE tend to lower the final value (that is a mean), obtaining better results. Conversely, while computing the global MAE on the aggregated dataset, only the higher value of *PA* is present and the final value will be higher. The same reasoning can be applied to all the other aggregated attributes and explains the reason why the global MAE is higher for the aggregated dataset.

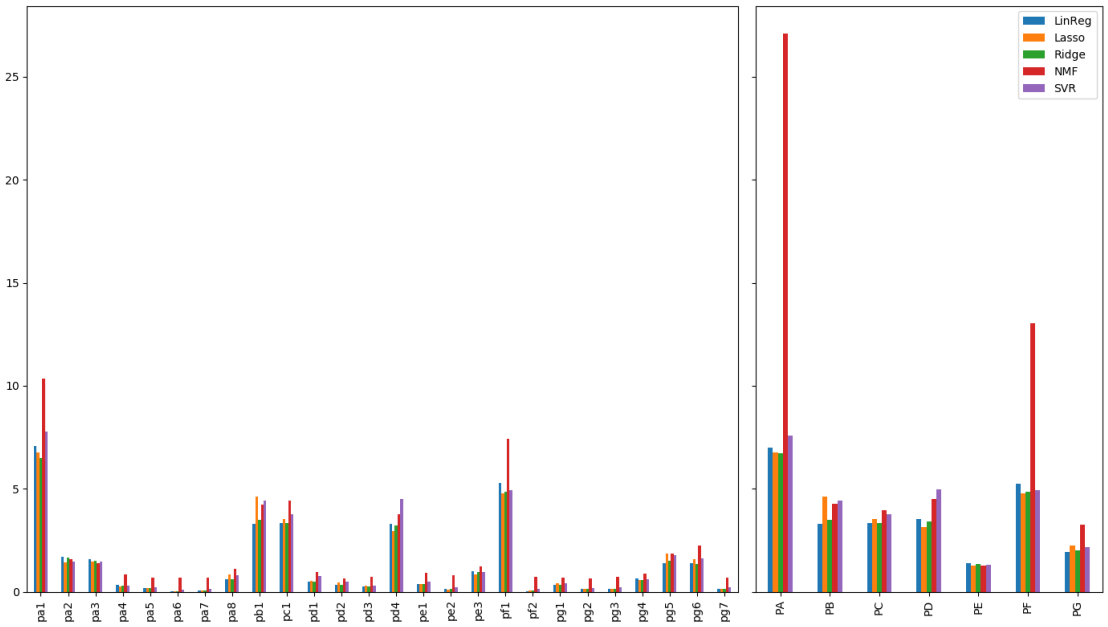


Figure 7.3: Detail of MAE for each attribute. On the right for aggregated dataset; on the left for non aggregated one.

In conclusion, it is possible to compare aggregated dataset performances with respect to the non-aggregated one by looking at the (*leading*) attribute with largest MAE values among each aggregation group. Hence, the MAE of the aggregated attributes is similar

to the one of leading attributes in the non aggregated dataset. By looking at Figure 7.3 the leading attributes of the aggregation groups are *pa1*, *pb1*, *pc1*, *pd4*, *pe3*, *pf1*. For the *PG* group there is not a prevalent leading attribute.

The prediction quality for aggregated attributes and the one for the corresponding leading attributes are similar. From now on, we will focus only on leading attributes to show the most relevant results.

7.2 Classification

In this section an analysis of performances of Decision Tree is done, by evaluating results through metrics defined in Subsection 6.6.2. In Subsection 7.2.1, some considerations concerning discretization minimum number of samples per bin (*nums*) is done. Then, best conditions identified by these considerations are fixed and the evaluation of results given by Decision Tree for leading attributes is reported in Subsection 7.2.2.

7.2.1 Discretization

In Figure 7.4 are shown metrics values for *pd4* with a discretization done by using a minimum number of samples per bin equal to 2. Figure 7.5 shows instead the outcomes with the minimum number of samples set as 4. Observing metrics for the Decision Tree (in brown) it is possible to see that using a coarser discretization (Figure 7.5) gives better results. Discretization algorithm in the case of $nums = 4$ with respect to the other case ($nums = 2$), collapsed bin $(7.3, 11]$ with $(11, 14.7]$ and $(14.7, 25.7]$ with $(25.7, 40.30]$. Notice that the last resulting range $(14.7, 40.30]$ is quite big with respect to the starting dimension of bins chosen, that is minor than 4. However it has to be pointed out that in this range only 5 samples are present. Augmenting the number of samples per bin, to avoid overfitting and to augment the number of examples from which to learn for the label worths sacrificing the width of the interval and consecutively the strictness of the prediction. This collapse in fact improves precision, recall and F-measure for all bins.

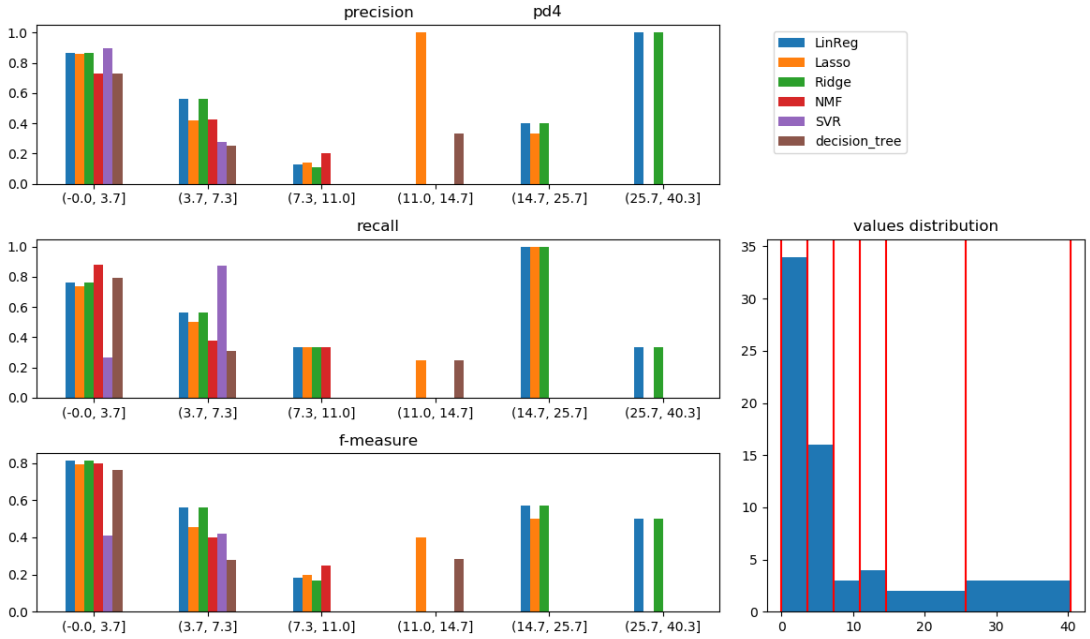


Figure 7.4: Metrics for *pd4*, error: 4, minimum samples per bin: 2.

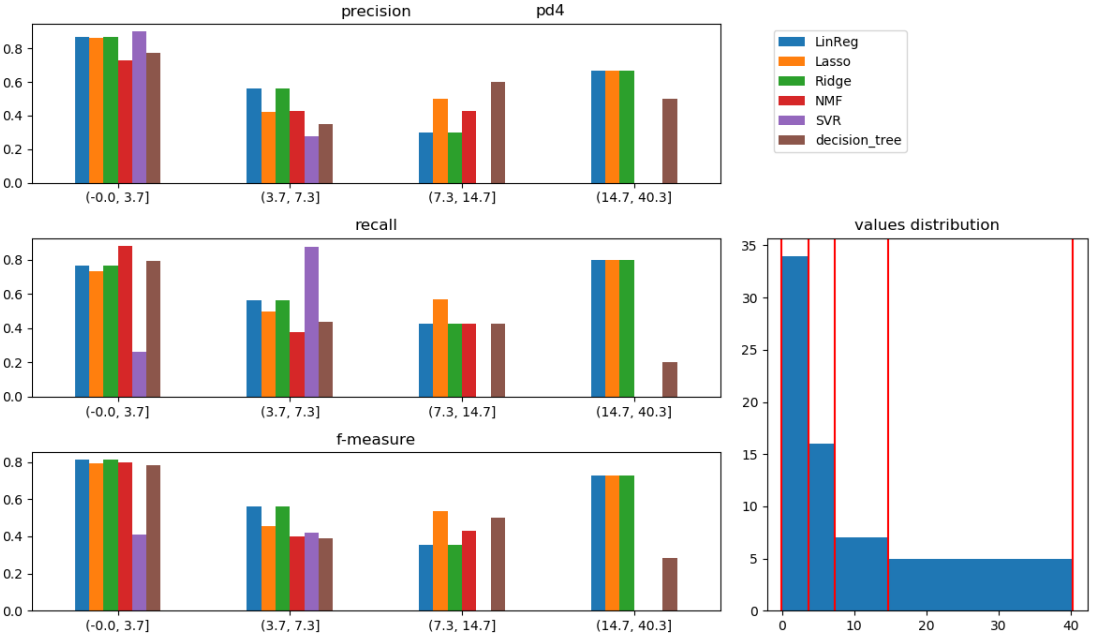
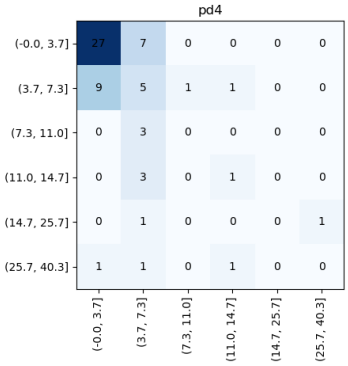


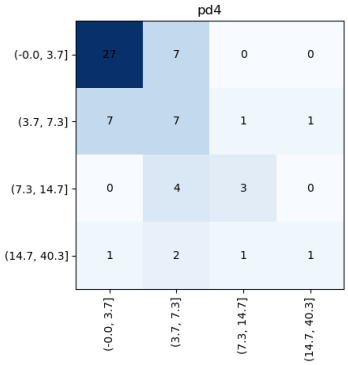
Figure 7.5: Metrics for *pd4*, error: 4, minimum samples per bin: 4.

Figure 7.6 also shows the improvement. It is presented a confusion matrix calculated on results obtained by collapsing bins of predictions, *after* classification process (Figure 7.6c). This means that the classifier works on the data discretized as reported on the right in Figure 7.4. Predictions of the classifier are then ri-assigned to new bins, because two new collapses are done: $(7.3, 11]$ is collapsed with $(11, 14.7]$ and $(14.7, 25.7]$ with $(25.7, 40.30]$. On the outcomes of this operation the confusion matrix is calculated. Figure 7.6b instead shows outcomes for bin collapse performed *before* classification process, on actual values. Indeed the classifier is trained with data discretized as reported in Figure 7.5 and confusion matrix is calculated directly on test predictions. This latter case introduces benefits with respect to the former: it augments the number of samples present in the diagonal of the confusion matrix. This is due to a higher number of samples, in each collapsed bin, that makes easier training the classifier. This considerations are generalizable for all attributes in the dataset. Indeed, in this section and in Section 7.4 are reported results for classification performed on attributes discretized using $nums = 4$.

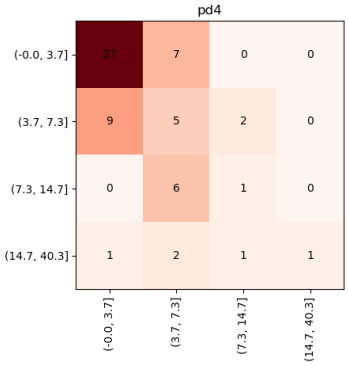
Figure 7.6: Confusion matrices for *pd4*. Matrix (a) is calculated on predictions of Decision Tree after having performed a discretization setting the minimum number of samples per bin to 2. While for (b) this parameter is set to 4. Matrix (c) is a cumulative version of (a) aggregating labels as in (b).



(a)



(b)



(c)

7.2.2 Decision Tree

Decision Tree obtains satisfying results for *pd4* and *pb1*, both exponentially distributed. They present a class, corresponding to the peak of the exponential distribution (see range $(0, 3.7]$ in Figure 7.5 for *pd4* and range $(0, 4]$ in Figure 7.7 for *pb1*), for which F-measure is at least 0.7 and all other classes are correctly predicted at least one time. For all other leading attributes there is one or more label that is never correctly predicted. Among leading attributes must not be considered *pe3*, because discretization process leads to only one class for this attribute.

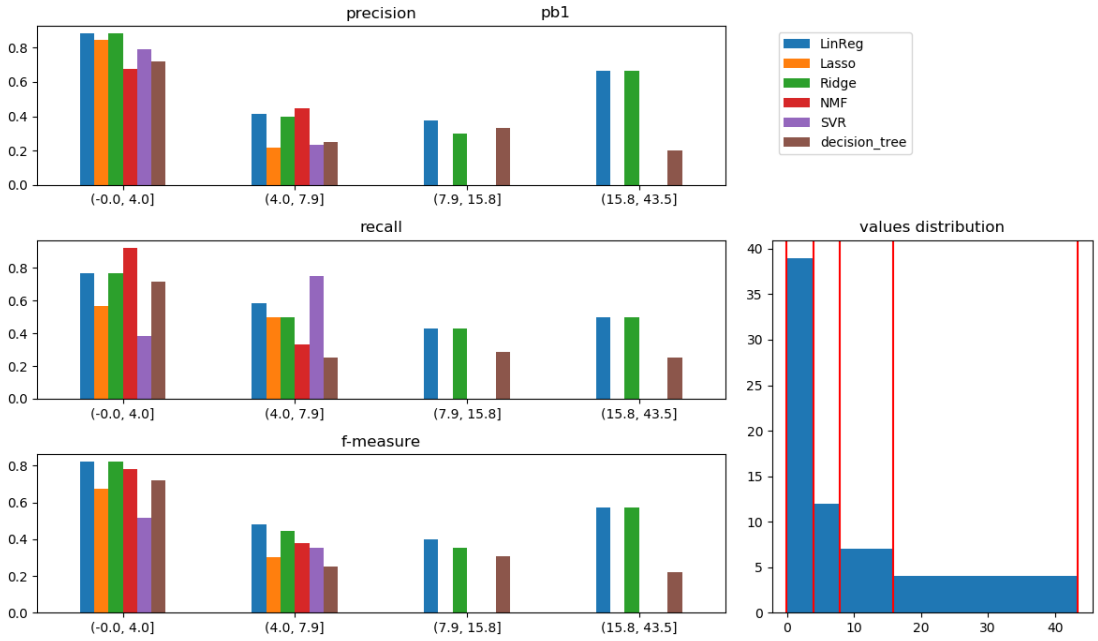


Figure 7.7: Metrics for *pb1*, error: 4, minimum samples per bin: 4.

All gaussian distributed attributes (*pa1* and *pf1*) never present an F-measure over 0.4 (Figures 7.9 - 7.10). Looking at confusion matrices (Figure 7.8) values spread all over them. This bad performance is probably due to the fact that discretization mode brings to a distribution of labels that is quite uniform, with a small number of samples per class in the order of 10. The model in this case is strongly overfitted.

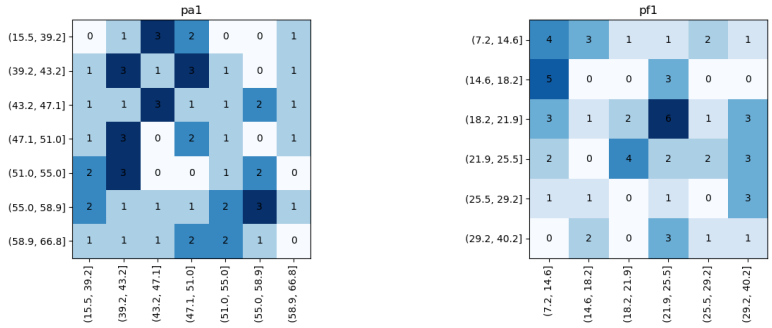


Figure 7.8: *pa1* and *pf1* confusion matrices, error: 4, minimum samples per bin: 4.

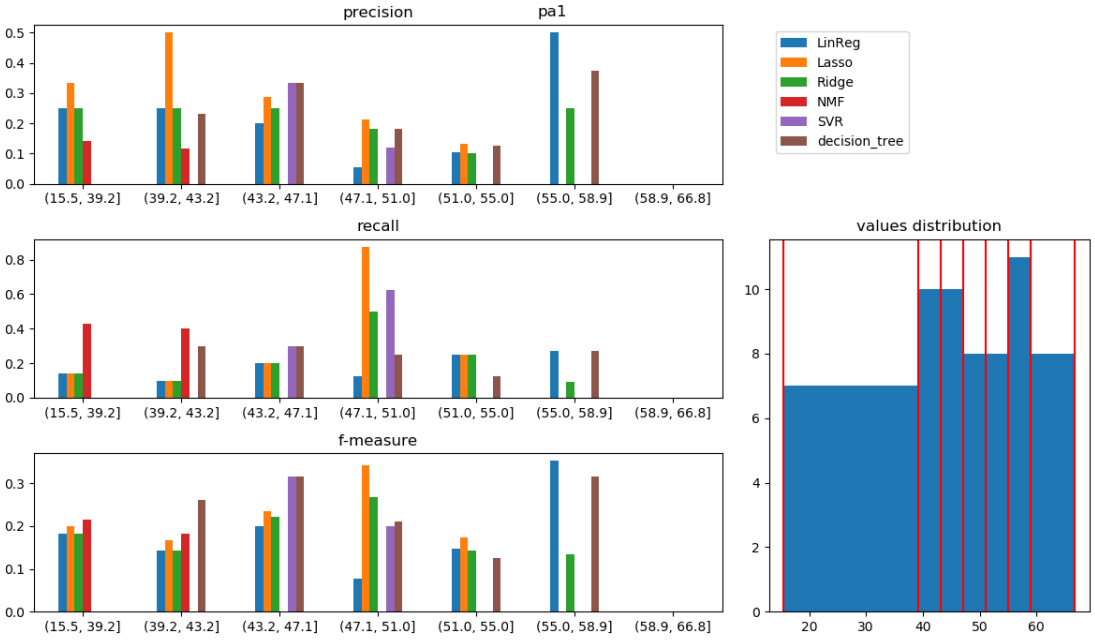


Figure 7.9: Metrics for *pa1*, error: 4, minimum samples per bin: 4.

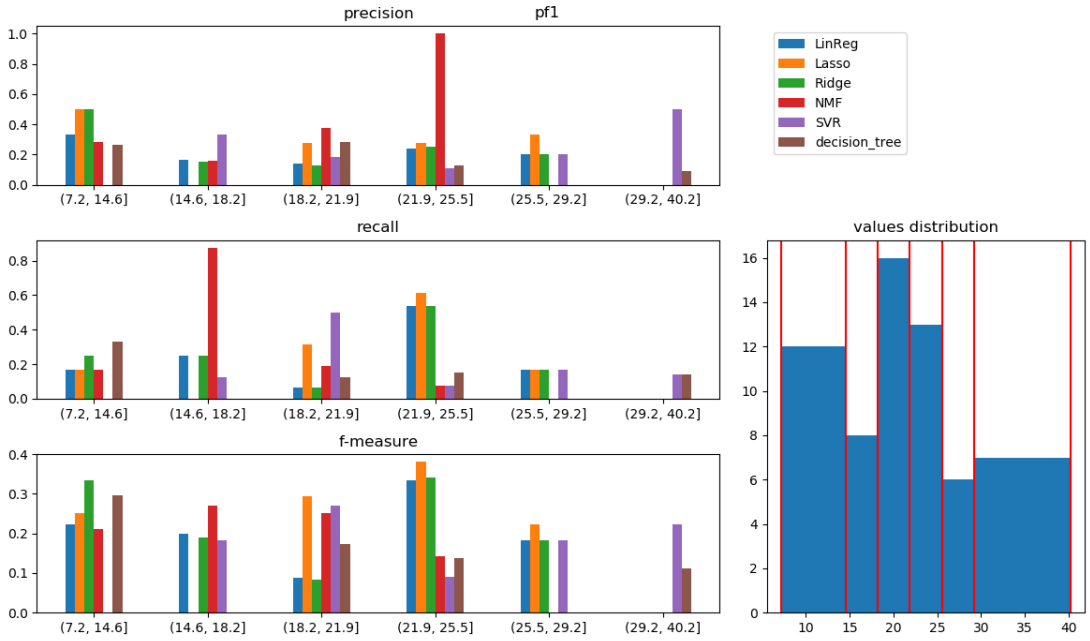


Figure 7.10: Metrics for *pf1*, error: 4, minimum samples per bin: 4.

7.3 Regression techniques comparison

In the following a comparison between results obtained applying different regression techniques is performed, using metrics defined in Subsections 6.6.1 - 6.6.3.

Looking at accuracy (Figure 7.11) it is possible to notice very good performances (accuracy reaches the maximum value for most non aggregated attributes). Actually, since for calculating accuracy a tolerance factor of 4 is considered, predictions for attributes with a variation range lower than 4 are quite always considered correct. Table 6.1 reports the list of attributes whose maximum value is lower than 8 for which the accuracy is high, but not significant since the low variation range. These attributes are not going to be taken in consideration in the following analysis.

Looking instead at leading attributes, good accuracies can be considered the ones of *pb1*, *pc1*, *pd4* and *pe3*, for which at least one regression technique provides a value over 0.70.

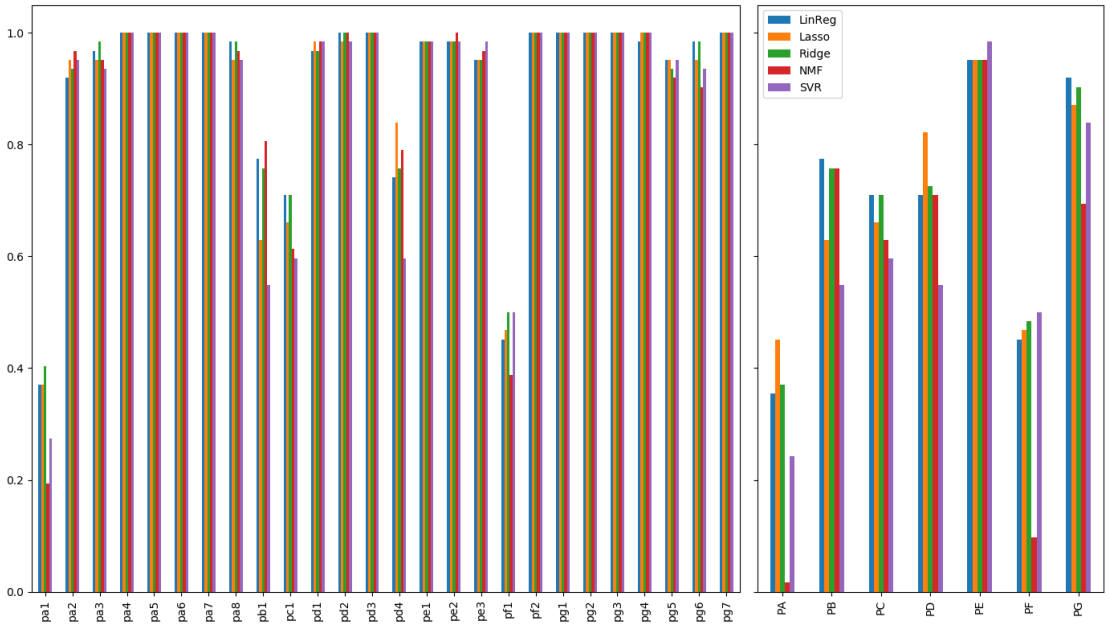


Figure 7.11: Details of accuracy for each method and attribute, for the non aggregated dataset on the left and for aggregated one on the right.

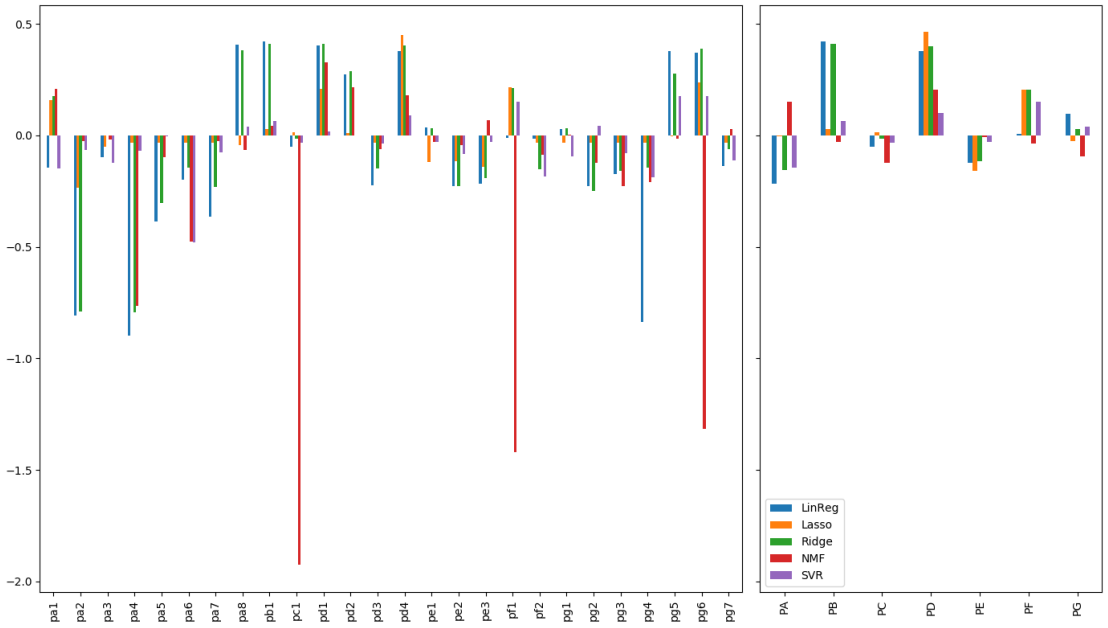


Figure 7.12: Details of EVS for each attribute and method, on the right the aggregated dataset, on the left the non aggregated one.

Looking at the EVS (Figure 7.12) it is possible to observe that only for some attributes has a value greater than zero, for both aggregated and non aggregated datasets. It is also interesting to compare EVS and accuracy. For attributes that have a good accuracy (*pb1*, *pc1*, *pd4* and *pe3*), it is possible to see that only two of them (*pd4* and *pb1*) also have an acceptable value of EVS for best performing regression method (over 0.4). All other leading attributes have a positive EVS for at least one method, but it is very close to zero.

In the following, are compared regression techniques results for attribute *pb1*, that present an acceptable EVS and accuracy. EVS indicates as more performing technique Linear regression, followed by Ridge regression. Accuracy, instead, points out NMF as more promising. We can analyze performances of those techniques in deeper, with the support of regression precision and recall. Figure 7.13 reveals that predictions performed by Ridge and Linear regression for *pb1* are of good quality. Orange values are less than blue ones and the distribution of predicted values is very similar to the initial distribution. Also some rare values (≥ 10) are well captured. NMF instead, clusters all prediction values in the same range. Accuracy is high because the range is in correspondence of the peak of exponential distribution (values in range $[0, 4]$), for this reason the majority of

values is considered correctly predicted.

Look at Figure 7.14, here are reported actual vs predicted values. NMF always predict the same range of values (the trend is an horizontal line). Ridge and Linear regressions, instead, provide more data on the diagonal.

Spending some words also on other methods, SVR has the lowest accuracy and Lasso the lowest EVS. Their predicted distributions are very different from the initial (actual) one. SVR has the lowest accuracy because it totally misleads predictions for peak values: its forecasts distribution presents a peak in correspondence of the wrong range: $[4,8]$. Summing up, Ridge and Linear regressions give best results and allow to reach good prediction quality for this attribute.

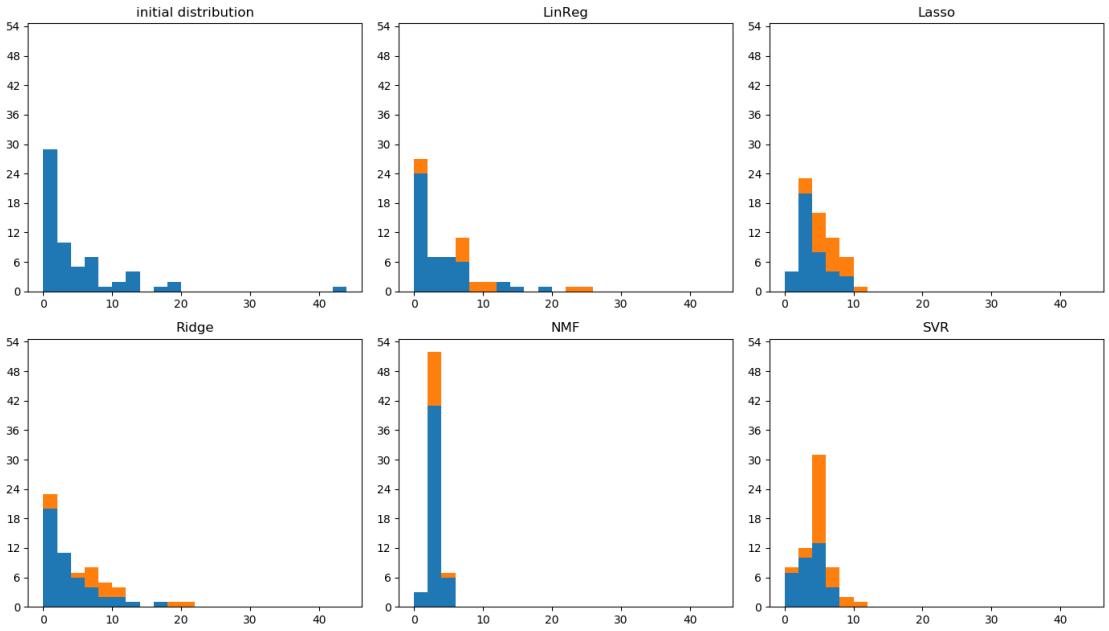


Figure 7.13: *pb1* regression-precision, tolerance: 4.

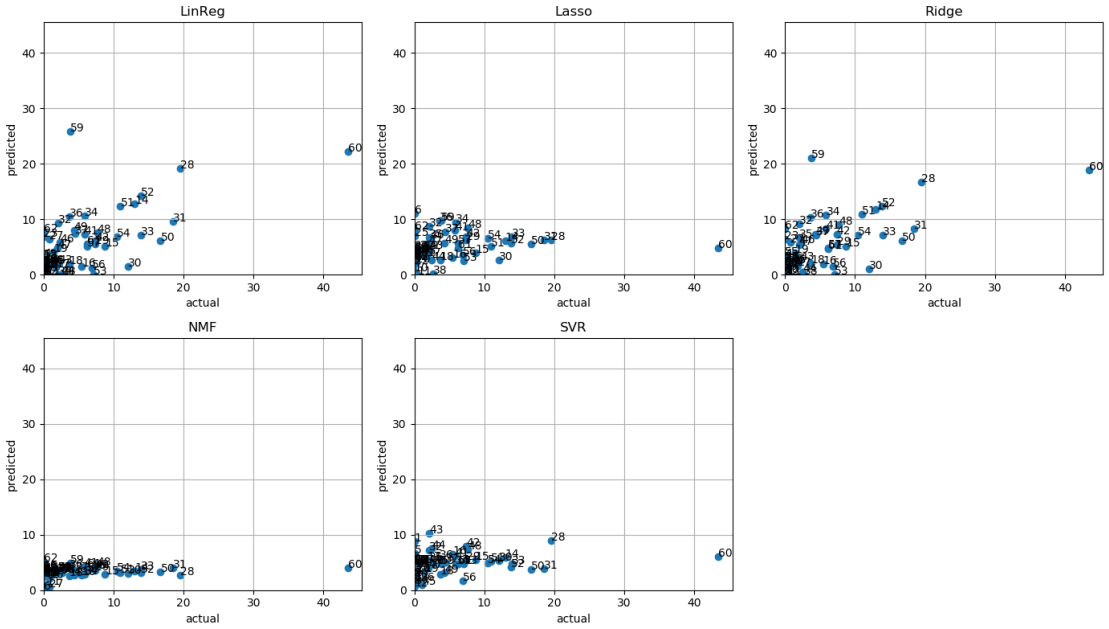


Figure 7.14: *pb1* scatterplot, actual values on x-axis, predicted ones on y-axis.

Attribute *pd4*, is now taken into account. It has a promising accuracy and EVS. In terms of these two metrics Lasso overcomes all other techniques, followed by Ridge, that has a good trade-off between accuracy and EVS. However, NMF has higher accuracy with respect to Ridge, but it has a low EVS. Looking at regression-recall in Figure 7.16, is evident that NMF predicts very well the peak of exponential distribution (few orange values in range $[0, 2]$) and looking at regression precision (Figure 7.15) we can see that this interval does not recall too much values from other ranges. The problem of NMF is for highest rare values, above 15 no value is correctly predicted.

For Lasso, the best technique in terms of EVS and accuracy, Figure 7.16 shows a good recall especially for medium values (range $[6, 30]$). Looking insted to precision (Figure 7.15), range reaching the maximum value of recall ($[6,10]$) presents a low precision, this means that the range recalls also values not belonging to it. Moreover the peak of exponential distribution is shifted towards higher values.

Ridge has a better precision in terms of forecasts distribution than Lasso but a worse recall. Lasso and Ridge regression can be both considered a good achievement for the prediction of this attribute.

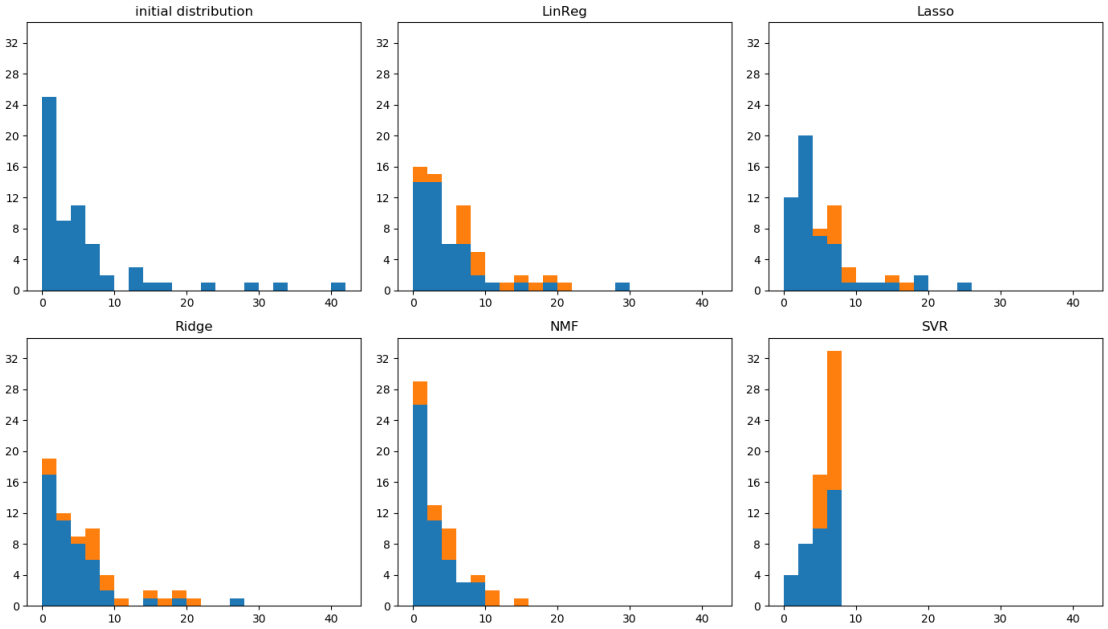


Figure 7.15: *pd4* regression-precision, tolerance: 4.

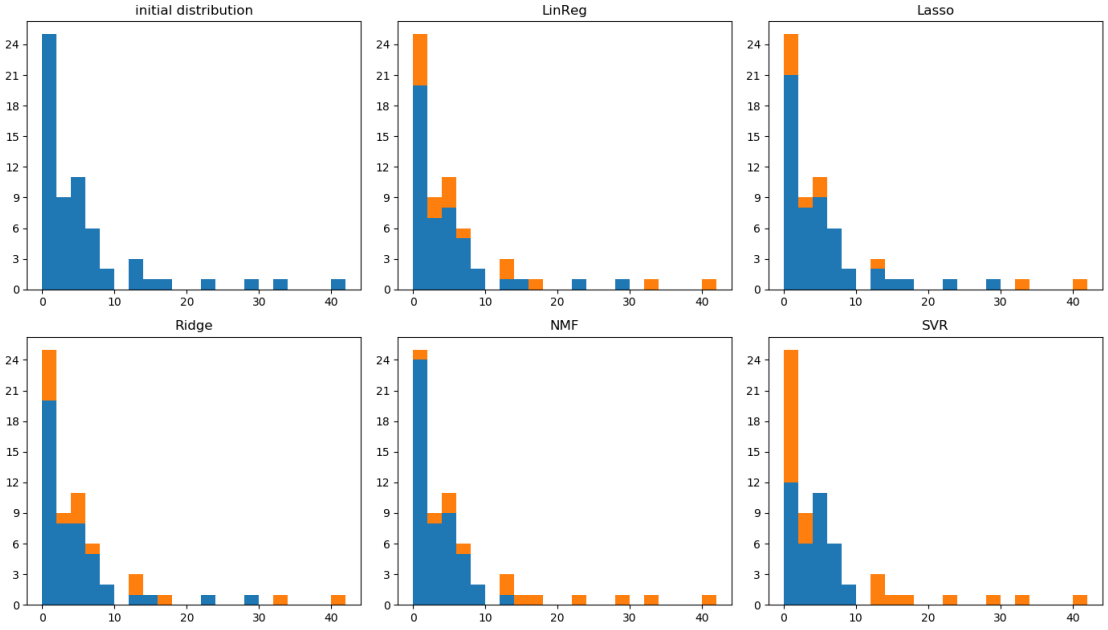


Figure 7.16: *pd4* regression-recall, tolerance: 4.

The leading attribute with the highest accuracy is *pe3* (refer to Figure 7.11). It has been already stated that, however, it has a low EVS. In Figures 7.17 is reported

regression-precisions for *pe3*. Taking the distribution of the predictions and looking at correct instances (in blue) is evident that every method predicts correctly low values of *pe3*. Instead, when the methods try to guess higher values they always make wrong predictions (orange is prevalent). This is actually not a good result because the model is able to predict only values in the first part of the distribution, it is not able to capture attribute variation, this is why it has a low EVS. The fact that it has an high accuracy is again due to the fact that the distribution is exponential, and peak values, the majority, are well detected. The technique with the highest accuracy is SVR, that instead behaves the worst. In fact all predicted values can be found in the same range ($[0, 2]$). This is visible also looking at the scatterplots actual vs predicted in Figure 7.18: data does not follow properly the diagonal of the chart at all, it composes instead a straight horizontal line. By the way this attribute is very complicated to be predicted, only one value is higher than 5. Notice that, leading a leave one out approach, when this sample is in the test set, all training values present a range of variation under 5. This makes its prediction very difficult to perform.

NMF is the only method that, for *pe3*, has an EVS greater than zero. In fact, by looking at the corresponding scatterplot in Figure 7.18, data in the range $[2.5, 4]$ tend to follow a bit more the diagonal.

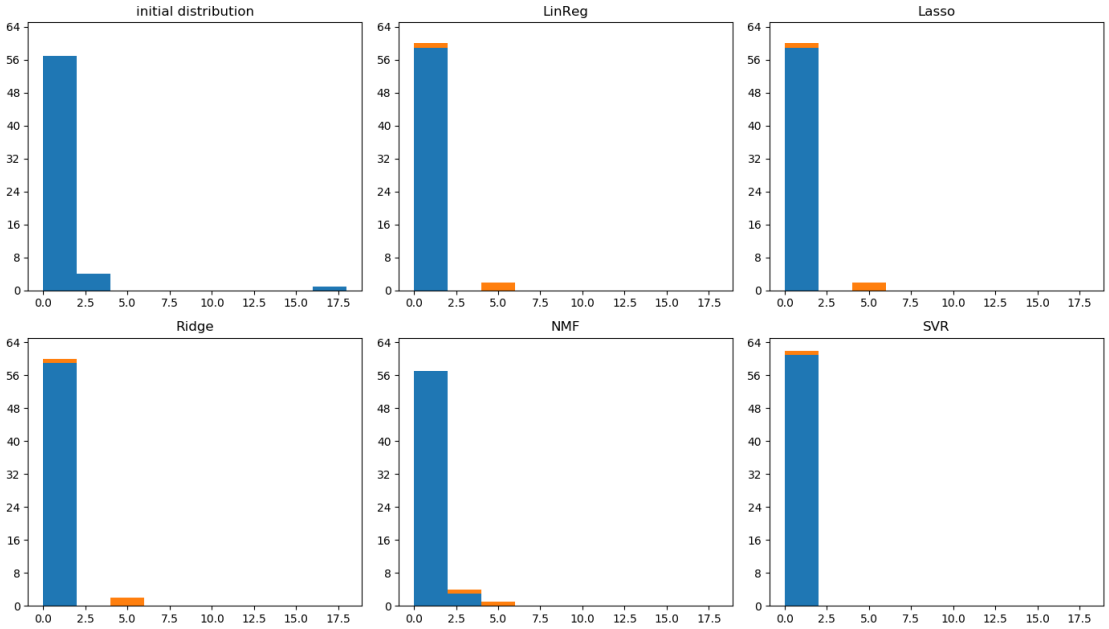


Figure 7.17: *pe3*, regression-precision, tolerance: 4.

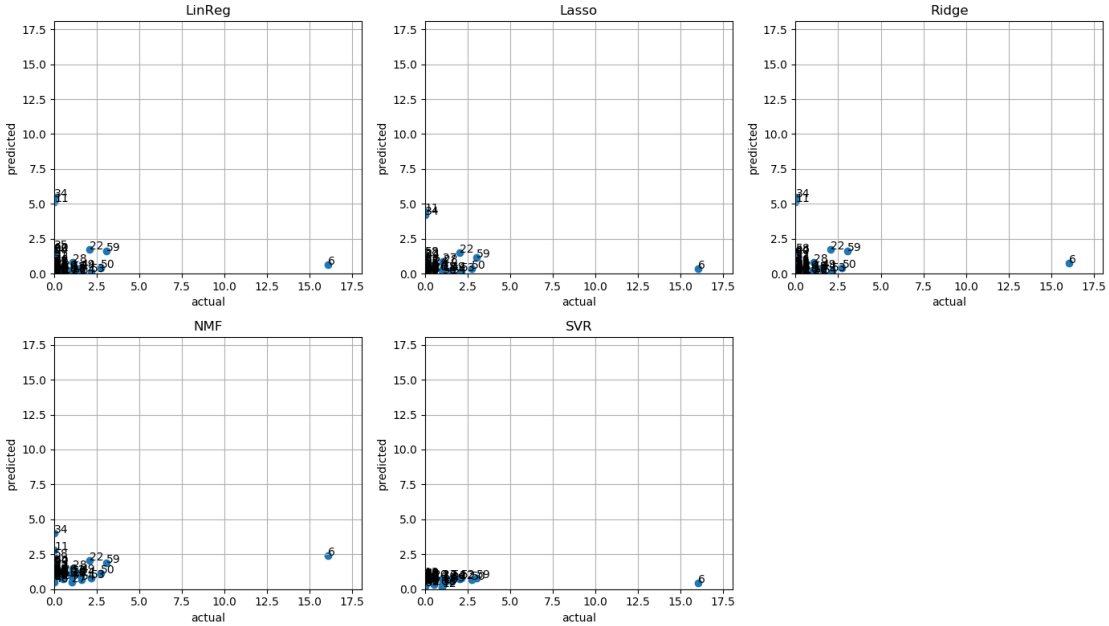


Figure 7.18: *pe3* scatterplot, actual vs predicted values.

Figure 7.19 reports regression-recall for the leading attribute *pa1* . Figure 7.20 shows instead the regression-precision for the same attribute. For *pa1* both accuracy and EVS are

low. Both precision and recall charts present many orange samples (wrong predictions). The distribution of the predictions is different from the actual one. In particular NMF and SVR perform very badly.

In conclusion observing and comparing all results it is possible to see that there is not a method that is better than the others for all attributes. On average, looking at current analyses, Ridge, Linear and Lasso regressions have best performances, while SVR and NMF have the worst.

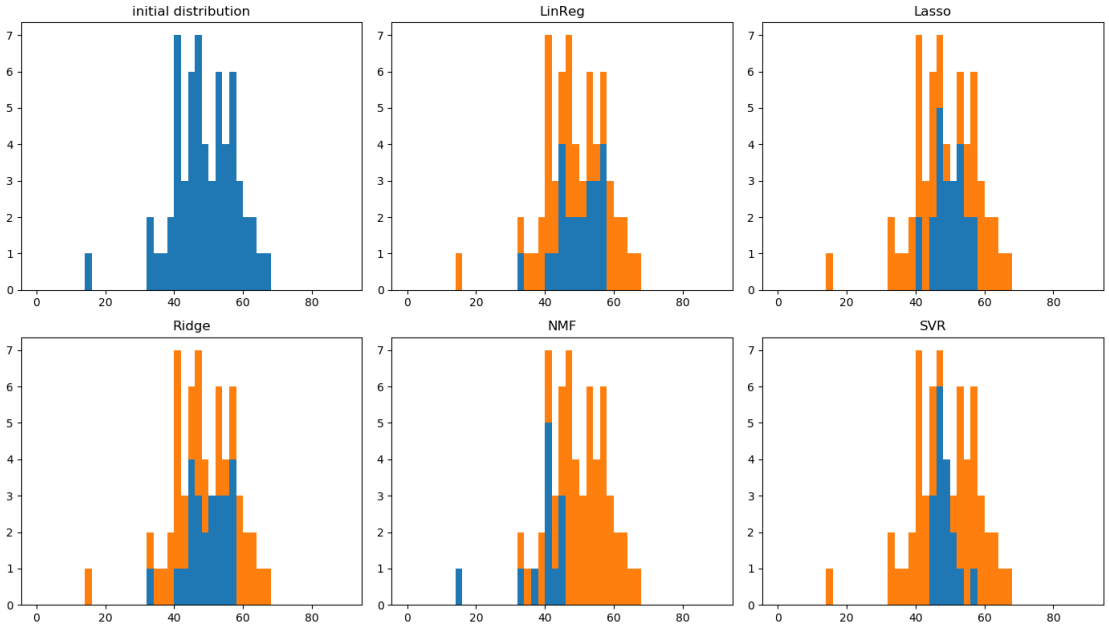


Figure 7.19: Regression-recall for *pa1*, tolerance: 4.

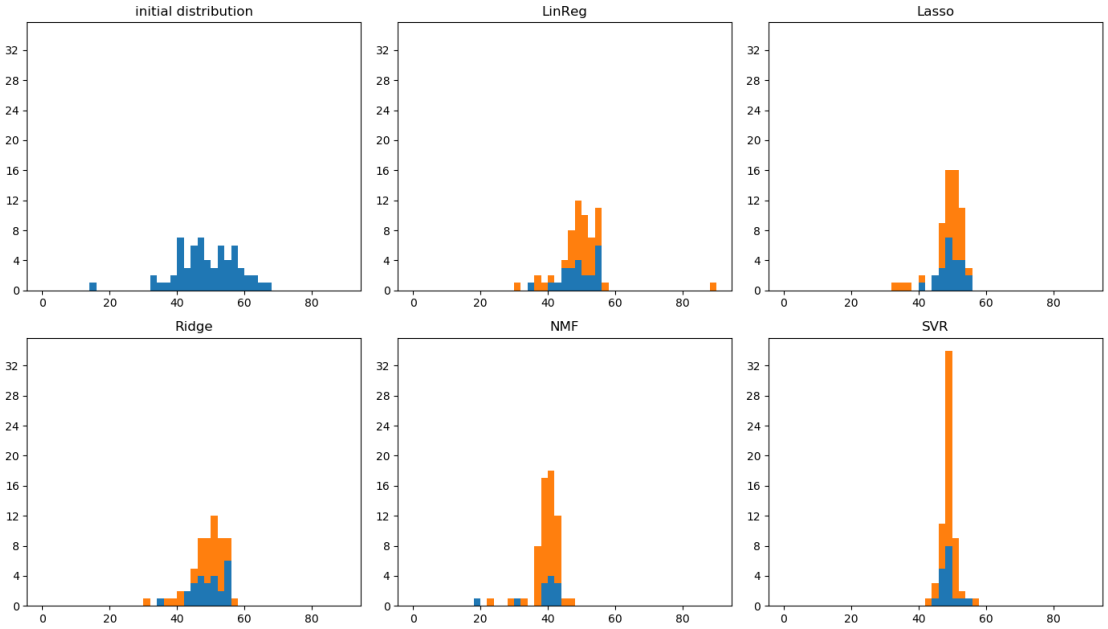


Figure 7.20: Regression-precision for *pa1*, tolerance: 4.

7.4 Classification vs Regression

Briefly summing up what stated in Subsection 6.6.4, regression and classification results are compared through metrics proper of classification (Subsection 6.6.2). In particular regressions results have been discretized to obtain the same format of classification outcomes. Conclusions of Subsection 6.6.4 state that a comparison performed in such a way can only determine if regression techniques perform better than classification and not vice versa.

We will focus on *pb1* and *pd4*, attributes for which both regression and classification give promising results.

In Figure 7.5 (page 50) and in Figure 7.21 is reported the case of *pd4*. We are going to lead a comparison between Decision Tree and Lasso regression followed by a comparison between Decision Tree and Ridge regression. In fact, these regression techniques, have been identified as more promising for this attribute in Section 7.3. Looking at Figure 7.5 it is possible to see that Lasso regression provides a F-measure greater than the one of the Decision Tree, for all ranges. In this case we can say that Lasso performs better than Decision Tree. Ridge behaves worse, in terms of F-measure, than Decision Tree only for

the bin: $(7.3, 14.7]$; the reason is that precision of Ridge regression is lower than the one of Decision Tree. Consulting confusion matrix for *pd4* in Figure 7.21, it is visible that some values actually belonging to range $[0, 7.3)$ are assigned to the discussed label, this is the reason why precision is lowered. Indeed range $(0, 3.7]$ has a lower recall for Ridge with respect to Decision Tree, because some values have been assigned to the wrong class. However, in the final value of F-measure for range $(0, 3.7]$ the low recall is compensated by an higher precision. High precision of the peak range is a good achievement, so we can totally consider both Ridge and Lasso regression better performing than Decision Tree for attribute *pd4*.

Figure 7.7 (page 53), reports instead metrics for *pb1*. Methods that give best results for this attribute among regressions, as stated in Section 7.3, are Ridge and Linear regression. Looking at F-measure is evident, for both Ridge and Linear regression, that the metric is higher than the one of Decision tree, for every range. So it is possible to conclude that for attribute *pb1* regression techniques provide better forecasts than the ones carried out by Decision Tree.

Observing metrics for all attributes, in general, regression overcomes classification in this specific application field.

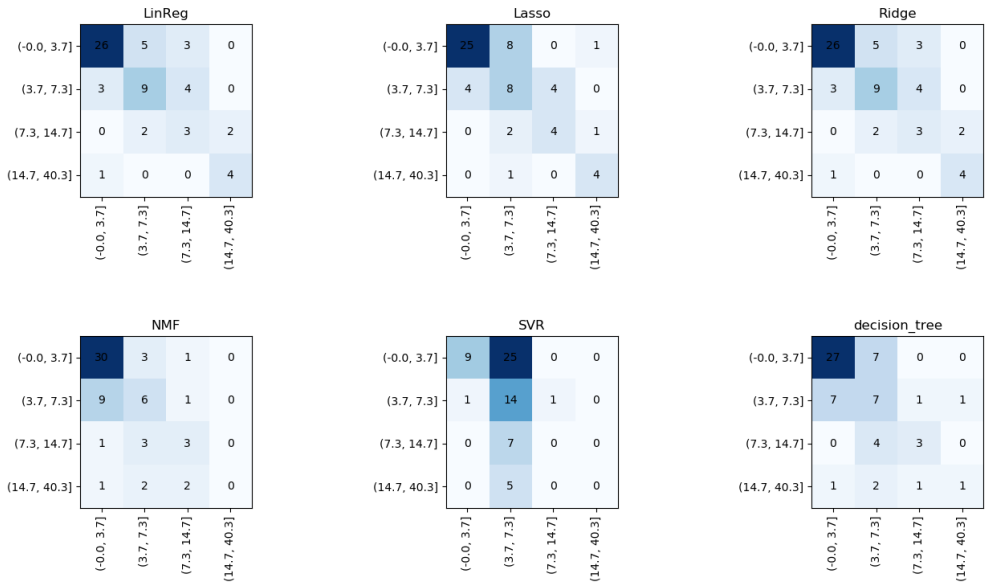


Figure 7.21: *pd4* confusion matrices, error: 4, minimum samples per bin: 4.

Chapter 8

Conclusions and future works

The aim of this study is to predict, for a geological sample, petrographical composition given mineralogical one. Constitutions are expressed through a set of continuous attributes, whose values represent the percentage in which the feature composes the sample. The dataset we worked on, has been provided in two versions; one (aggregated) version has a less detailed petrographical description: some attributes are collapsed together with respect to the non aggregated version, by summing up their values. There are mathematical links between mineralogical composition and petrographical one but we did not exploit them because of measurement uncertainty that affected data we have at disposal. In our study we decided to focus on petrographical attributes that have a range of variation larger than the error of measurement communicated by domain experts.

Challenges of the application field are the small dataset and the non uniform distribution of the attributes values. Distributions, approximatively gaussian and exponential, present some very few rare values that we need to correctly predict. To perform predictions we tried three approaches: regression, classification and a custom NMF (regression) method. We used, as classifier, the Decision Tree and as regression techniques Linear, Lasso, Ridge and Support Vector regressions. There is more than one feature to predict, indeed we built one different regression and classification model for predicting each of them. Each model has as input all mineralogical features and as output only one petrographical attribute. NMF instead has as output all petrographical features and as input all mineralogical ones. Features to be predicted are continuous and we needed a way to obtain discrete values for the output of the classifier. For this reason we implemented a

custom methodology of discretization for obtaining class labels out of continuous features. Each attribute has been discretized in such a way that the classifier indicates as output the possible range in which the value could fall. The method aims to maximize the number of samples belonging to rare values ranges, this has been done by enlarging progressively these ranges finding a trade off between the width of the interval and the number of samples belonging to it. Precision and recall classification metrics allow to properly evaluate how well the classifier predicts each range separately, this permits to properly evaluate the performances of Decision tree for rare values. We designed, for the same purpose, two visual metrics for evaluating regression techniques inspired by these classification ones: regression-precision and regression-recall.

By a preliminary analysis we identified a set of six (leading) attributes in the non aggregated dataset, for which predictions quality can be directly compared with the quality of forecasts for their aggregated version; we selected them as key attributes on which we leaded analysis of results. We focused on the non aggregated dataset because performances of methods are quite the same for both dataset, this means that aggregating some attributes does not bring to any kind of benefit. Discretization method we designed augment predictions quality of the classifier. In fact increasing the number of samples per bin and consecutively enlarging intervals, cause an improvement of models performances. Regression methods that fit better to our dataset are Linear, Lasso and Ridge regressions. SVR and NMF do not reach good results at all, according to metrics that we have defined. Between Linear, Lasso and Ridge it is not possible to identify the best one for the whole dataset, because regression method that performs better depends on the considered attribute. Regression results have been discretized to permit a direct comparison with the outcomes given by the Decision Tree. From this direct comparison came out that the regression approach is more promising than the classification one. Good performances have been obtained, both by regression and classification models, on two exponentially distributed attributes (*pd4* and *pb1*). All the remaining four leading attributes are very badly predicted. For *pd4* and *pb1* is reached a regression-accuracy over 0.7 and an EVS over 0.4. For what concerns classification, instead, they reach an F-measure for the range corresponding to the peak of exponential distribution greater than 0.7 while all the other classes are correctly predicted at least one time. Good predictions obtained for these two features, also on rare values, suggest to continue the study. In particular we will try

different regression techniques on attributes that presently are not well predicted, with the aim to eventually use for each attribute the best performing method. Moreover, with the help of a domain expert, we will exploit also mathematical links between mineralogical and petrographical attributes to overcome dataset difficulties. A possible exploitation of these properties may consist in using well predicted attributes values to constrict the predictions on the other features.

Bibliography

[1] George AF Seber and Alan J Lee. *Linear regression analysis*, volume 329. John Wiley & Sons, 2012.

[2] Clara Dismuke and Richard Lindrooth. Ordinary least squares. *Methods and Designs for Outcomes Research*, 93:93–104, 2006.

[3] Trevor Hastie, Robert Tibshirani, and Martin Wainwright. *Statistical learning with sparsity: the lasso and generalizations*. CRC press, 2015.

[4] Daniel D Lee and H Sebastian Seung. Algorithms for non-negative matrix factorization. In *Advances in neural information processing systems*, pages 556–562, 2001.

[5] Xin Luo, Mengchu Zhou, Yunni Xia, and Qingsheng Zhu. An efficient non-negative matrix-factorization-based approach to collaborative filtering for recommender systems. *IEEE Transactions on Industrial Informatics*, 10(2):1273–1284, 2014.

[6] Satoru Tsuge, Masami Shishibori, Shingo Kuroiwa, and Kenji Kita. Dimensionality reduction using non-negative matrix factorization for information retrieval. In *Systems, Man, and Cybernetics, 2001 IEEE International Conference on*, volume 2, pages 960–965. IEEE, 2001.

[7] Ali Caner Türkmen. A review of nonnegative matrix factorization methods for clustering. *arXiv preprint arXiv:1507.03194*, 2015.

[8] J. Ross Quinlan. Induction of decision trees. *Machine learning*, 1(1):81–106, 1986.

[9] Leo Breiman. Some properties of splitting criteria. *Machine Learning*, 24(1):41–47, 1996.

[10] Richard Lippmann. An introduction to computing with neural nets. *IEEE Assp magazine*, 4(2):4–22, 1987.

[11] William W Cohen. Fast effective rule induction. In *Machine Learning Proceedings 1995*, pages 115–123. Elsevier, 1995.

[12] Sung-Hyuk Cha. Comprehensive survey on distance/similarity measures between probability density functions. *City*, 1(2):1, 2007.

[13] James Dougherty, Ron Kohavi, and Mehran Sahami. Supervised and unsupervised discretization of continuous features. In *Machine Learning Proceedings 1995*, pages 194–202. Elsevier, 1995.

[14] Barbara G Tabachnick and Linda S Fidell. *Using multivariate statistics*. Allyn & Bacon/Pearson Education, 2007.

[15] Jianping Hua, Zixiang Xiong, James Lowey, Edward Suh, and Edward R Dougherty. Optimal number of features as a function of sample size for various classification rules. *Bioinformatics*, 21(8):1509–1515, 2004.

[16] Petra Perner. Machine learning and data mining in pattern recognition 13th international conference, mldm 2017, new york, ny, usa, july 15-20, 2017, proceedings. In *Conference proceedings MLDM*, page 1. Springer, 2017.

[17] Nathalie Japkowicz. The class imbalance problem: Significance and strategies. In *In Proceedings of the 2000 International Conference on Artificial Intelligence (ICAI*, pages 111–117, 2000.

[18] Xiang Li, Wanling Wong, Ecosse L Lamoureux, and Tien Y Wong. Are linear regression techniques appropriate for analysis when the dependent (outcome) variable is not normally distributed? *Investigative ophthalmology & visual science*, 53(6):3082–3083, 2012.