Instituto de Ciências Matemáticas e de Computação

ISSN - 0103-2569

TRANSFORMING A REGRESSION PROBLEM INTO A CLASSIFICATION PROBLEM OBSERVING THE DISCRETIZATION PROCESS: A CASE STUDY

LUIS CARLOS MOLINA FÉLIX
SOLANGE OLIVEIRA REZENDE
MARIA CAROLINA MONARD
CHANDLER WELLINGTON CAULKINS

Nº 76

RELATÓRIOS TÉCNICOS DO ICMC

São Carlos
Ago./1998

Table of Contents

Transforming a regression problem into a classification problem observing the discretization process: A Case Study[1]

Luis Carlos Molina Félix
Solange Oliveira Rezende
Maria Carolina Monard
Chandler Wellington Caulkins

University of São Paulo
Institute of Mathematical Sciences and Computer Science
Av. Dr. Carlos Botelho 1465, Cx.P. 668 CEP 13560-970
São Carlos, SP, Brasil
{lmolina,solange,mcmonard,chandler}@icmc.sc.usp.br

Abstract

This work shows the effects of the discretization of a continuous attribute-class when transforming a regression problem into a classification problem. To do this, a case study was performed based on the data and criteria given by (Rogers et al., 1995). The main objective is to try to predict the permeability of an oilwell using induction rules, taken from the depth, porosity and permeability of neighboring oilwells. Three different scenarios based on geological considerations are presented. By making use of statistical methods and visualization techniques, an analysis was done beforehand to get a better understanding of the domain. A comparison is also presented of the various error rates obtained using the symbolic Machine Learning algorithms, CN2 and C4.5-rules. The discretizations of the permeability values are considered both in a *stand-alone* manner and according to the considerations of an expert, thus emphasizing the influence of the discretization on the precision of the rules rather than the actual knowledge that was obtained.

Keywords: Discretization, continuous attributes, discrete attributes, oilwells.

[1] This work was partially supported by the Brasilian Research Agencies CNPq and FINEP (RECOPE-IA Project)

1. Introduction

Machine Learning addresses the question of how to build computer programs that learn, thus improving their performance at some task through experience.

One of the dimensions that influences learning is the degree of supervision. In supervised learning, the learner is given direct feedback about the appropriateness of its performance. This contrasts sharply with unsupervised learning in which feedback is absent. In this work we focus on supervised learning.

In supervised learning, a learning program is given training examples (cases) of the form $\{(X_1,Y_1), (X_2,Y_2),...,(X_m,Y_m)\}$ for some unknown function $y = f(X)$. The X_i values are typically vectors of the form $(\chi_{i,1}, \chi_{i,2},..., \chi_{i,m})$ whose components are discrete or real values. These are also called the *features* of X_i. Table 1 illustrates their general organization where a row refers to the *i-th* case, and column entries refer to the *j-th* feature of X_i.

Table 1: Training examples (cases) format.

Case	f_1	f_2	...	F_m	y
1	$\chi_{1,1}$	$\chi_{1,2}$	...	$\chi_{1,m}$	Y_1
2	$\chi_{2,1}$	$\chi_{2,2}$	...	$\chi_{2,m}$	Y_2
...	...	...	...	...	...
...	...	...	...	...	...
n	$\chi_{n,1}$	$\chi_{n,2}$	...	$\chi_{n,m}$	Y_n

The y values are typically drawn from a discrete set of classes {i,...,k} in the case of *classification* or from the real line in the case of *regression*. When working with regression problems the attribute-class must be discretized to transform it into a classification problem.

In essence, a discretization algorithm takes as input the values of a continuous attribute (integer or real) and generates output as a small list of sorted intervals. Each interval is represented as $[V_{lower}, V_{upper}]$, where V_{lower} and V_{upper} are the lower and upper limits of the interval, respectively. In general, discretization algorithms have two steps: first the input attribute values are sorted, and then a discretization procedure is applied to the sorted values to produce a set of sorted intervals.

In this work a case study of discretization of the continuous class attributes is presented based on the data and criteria of the article "*Predicting Permeability from Porosity Using Artificial Networks*" (Rogers et al., 1995). The idea is to predict the permeability of an oilwell using induction rules and data about the depth, porosity, and permeability of neighboring oilwells. Three different scenarios based on geological considerations are presented. Using statistical methods and visualization techniques, an analysis was done beforehand to get a better understanding of the domain. Subsequently, a comparison is also made of the various error rates obtained by the symbolic Machine Learning algorithms, CN2 (Clark & Niblett, 1989) and C4.5-rules (Quinlan, 1987), considering the discretizations done on the permeability values both in a *stand-alone* manner and based on considerations from an expert.

Finally, in this work the importance of interaction between experts and *stand-alone* discretization methods is emphasized, especially when determining the criteria for discretizing. ML algorithms that use discretizations supplied by the experts have proven useful for analyzing both nominal and discretized data. Also, the *stand-alone* discretization methods can considerably influence the criteria of the expert, offering information about the data.

This work is organized as follows: Section 1, gives an introduction to the problem. In Sections 2 and 3, the state of the art and the context of the problem are presented. A methodology for solving the problem is shown in Section 4. The results that were obtained are presented in Section 5 and finally, in Section 6, some conclusions are made.

2. State of the Art

The process of extracting knowledge from databases is defined as a non-trivial, recent and valid identification process, which is potentially useful for finding comprehensible patterns embedded in the data (Fayyad et al. 1996). Thus, knowledge extraction from data can be seen as a process made up of 5 phases:

1. understanding and describing the domain;
2. preparing the data;
3. pattern discovery (data mining);
4. interpretation/evaluation of the results;
5. using the obtained results.

The discretization of the data can be found in the step before pattern discovery. Some researchers consider it as a step apart from ML, which has great relevance in the process of extracting knowledge from databases (Ventura & Martinez, 1995).

The discretization of continuous values has some advantages and disadvantages. The first advantage has to do with storage space in memory. Continuous values require, in general, a much larger area in memory than that used for discretized values. Also, the processing of continuous data is often more time-consuming. Some researchers have shown reductions in the time taken to learn, which can be considerably smaller when using discretized data (Catlett, 1991). Additionally, classifiers of a much larger size are produced when processing continuous data. A disadvantage of discretizing a continuous value is the loss, in some cases, of information available in the continuous values. For example, two different values in the same interval of discretization are considered equal, even though they may be at two different extremes of the interval. Such an effect can reduce the precision of the algorithms (Ventura & Martinez, 1995). One representation of the interaction between a discretization algorithm, the data types and an induction algorithm is shown in Figure 1.

According to some of the criteria that are currently available, the discretizations (and simultaneously the discretization methods) are classified into many categories which can be: supervised or non-supervised, local or global, parameterized or non parameterized.

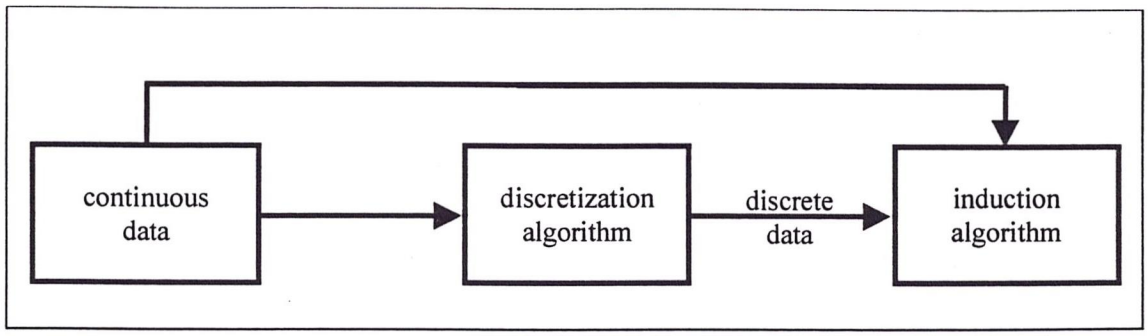


Figure 1: Interaction between a discretization algorithm, an induction algorithm and the different kinds of data.

Supervised discretization methods take into consideration the class values when producing the discretization, while the non-supervised methods are *decision-blind* or hidden-class, thus not making use of instantiation levels. Local methods discretize one attribute per run or from certain regions of the instantiated space, while the global ones try to do a simultaneous discretization of all the attributes. Parameterized methods are those for which the maximum number of intervals generated for an attribute are specified ahead of time, whereas the non-parameterized methods determine this value automatically.

Various elements should be considered when doing a discretization, for example, statistical techniques, discretization algorithms, experts, etc. Many discretization algorithms with different search methods for determining the cut points are found in the literature (Chmielewski & Grzymala-Busse, 1995; Dougherty et al., 1995; Fayyad, 1993; Kerber, 1992; Lenarcik & Piasta, 1992, 1993, 1995; Nguyen & Skowron, 1995; Pfahringer, 1995; Ventura & Martinez, 1994). The context in which the problem treated in this work is situated will be presented in the following section.

3. Context of the Problem

The present case study is based on the work "*Predicting Permeability from Porosity Using Artificial Networks*" developed by S.J. Rogers, H.C. Chen, D.C. Kopaska-Merkel and J.H. Fang (Rogers et al., 1995) which shows the results of a classification done by a neural net used for predicting the permeability of an oilwell based on its porosity and depth. The results obtained from this are compared with the results from using linear regression and with the observed data. The dataset is determined by measurements taken from six oilwells located in Big Escambia Creek, Alabama, in the United States. Three scenarios were created, based on geological considerations made by experts. In the first scenario, oilwells 1877 and 1928 were used as the training set, oilwell 1802 as the validation set and oilwell 1930 as the oilwell to be predicted. The second scenario used oilwells 1802, 1877 and 1930 as the training set, oilwell 1928 as the validation set and oilwells 1704 and 1705 as the oilwells to be predicted. For the third scenario, oilwells 1802, 1877, 1928 and 1930 were used as the training set, oilwell 1705 as the validation set and oilwell 1704 as the "predicted" oilwell. There is a large absence of data, both for the porosity and permeability as well as a lack of sequence for the depth.

Using the symbolic Machine Learning algorithms CN2 and C4.5-rules and based on the three scenarios mentioned previously, we considered oilwells 1802, 1877 and 1928 as the training set and oilwell 1930 as the test set, in the first scenario. In the second set,

oilwells 1802, 1877, 1928 and 1930 were considered the training set, and oilwells 1704 and 1705 were considered as two separate test sets. Finally, for the third scenario, oilwells 1705, 1802, 1877, 1928 and 1930 were considered as the training set and oilwell 1704 was considered the test set. In Figure 2, the relative depth of the six oilwells (in feet) and the quantity of cases for each oilwell, as well as the three scenarios considered, is presented in visual form.

Based on the context presented above, this work tries to show the influence of class discretization on the learning precision, comparing various permeability discretizations done with different criteria used by the CN2 and C4.5-rules rule induction algorithms, and also comparing the various error rates that were obtained.

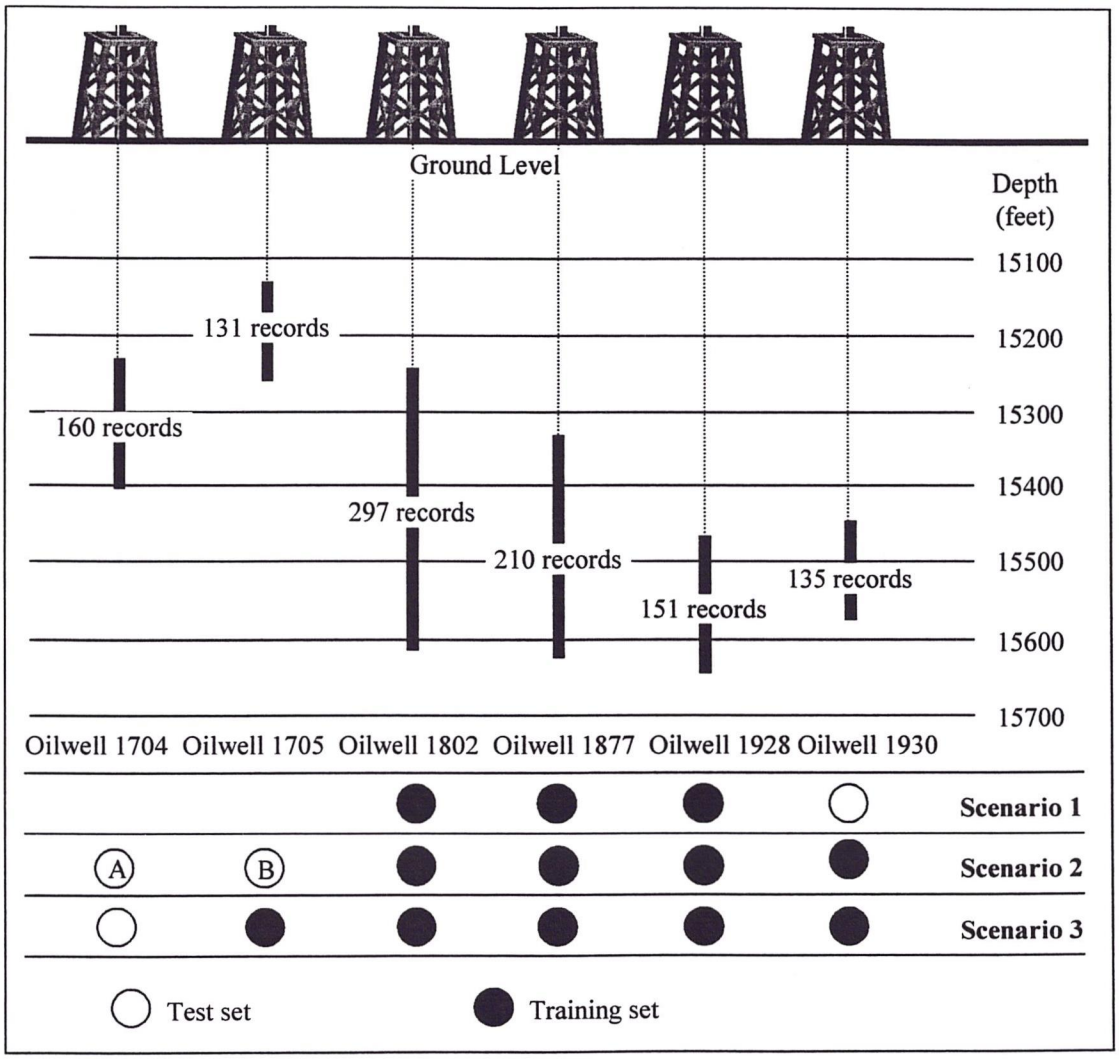


Figure 2: Information from the six oilwells (total 1084 records).

4. Methodology

Below we take a look at how this case study was done, starting with the tools and algorithms used. Next, the preliminary analysis of the data is discussed, and to the results that were obtained.

4.1 Tools and Algorithms

Two statistics tools were used in this work, STATISTICA™, version 5.0, from StatSoft Inc., Statistics and Visualizer™ from Silicon Graphics. The Scatter Visualizer™ from Silicon Graphics was used for simultaneously analyzing the behavior of data from the six oilwells in many dimensions.

The discretizations were done with the software MineSet™, version 2.01, a tool from Silicon Graphics (Silicon Graphics, 1998), which incorporates Data Mining techniques with multidimensional data visualization incorporating some discretization tools for data analysis to find, among other things, classifications and associations between elements in the databases.

In these experiments the CN2 (Clark & Niblett, 1989) and C4.5-rules (Quinlan, 1987) algorithms were used to determine and compare the error rates using the different discretizations for the attribute class *permeability*. CN2 is a non-incremental learning algorithm which takes a set of examples as input generating a set of "*if...then*" production rules to classify the examples. C4.5 is a decision tree generator and C4.5-rules produces *if-then* rules from the decision trees generated (Quinlan, 1993).

Both algorithms were executed using MLC++ (*Machine Learning Library in C++*) (Kohavi et al., 1994; Molina et al., 1998), a software package developed at Stanford University which contains various machine learning algorithms, offering the advantage of not having to change the format of the input data when using another algorithm. It also provides standardized methods for running experiments using these algorithms.

4.2 Experiments

4.2.1 Data Analysis and Data Preparation

The domain is denoted by the class attribute *permeability* related to *porosity* and *depth*. Permeability is defined as the ability with which a gaseous or liquid fluid penetrates a material through its pores, when subject to a pressure, measured in *darcies*. Porosity is the property that a material has to contain pores or interstices (intervals that separate the molecules of a body). It is defined as the relation (in percentage) between the volume of the interstices and the volume of the mass of the material, depending on the number, shape and distribution of the empty spaces (Water Resource Research Center, 1998). The original data about depth, porosity and permeability used in this work, and described in Table 2, are in the form of continuous attributes.

The first step of this experiment was the analysis of the data from the oilwells both separately and together, so as to get a better understanding of the domain in which we were working. The use of statistic methods has an important role in this first step, such as finding the average, the standard deviation, the mode, the maximum and minimum values of each attribute, etc.

STATISTICA™ was used during this step to analyze the depth, porosity and permeability of the six oilwells (1084 cases), and determine the maximum and minimum values, their distribution, the average, the median, the standard deviation,

total number of known values, total number of unknown values, number of distinct values, the relationships between permeability and porosity, permeability and depth, etc. Some of the statistical results from all six oilwells are shown in Table 2.

Table 2: Statistical results from the six oilwells.

	Depth	Porosity	Permeability
Total Records	1084	1084	1084
Missing Values	0	96	96
LOST CORE Values	0	6	6
Known Values	1084	982	982

First, 96 cases which had missing values for both permeability and porosity were eliminated, as well as the 6 cases that had the legend "*LOST CORE*". This legend indicates that the detector instrument lost the porosity and permeability values for a certain depth. Having eliminated these cases, a total of 982 cases were left for analysis.

Next, it was observed that the ordinal data contains several permeability values <0.01 (smaller than 0.01) as well as many other values equal to 0 (see Table 3).

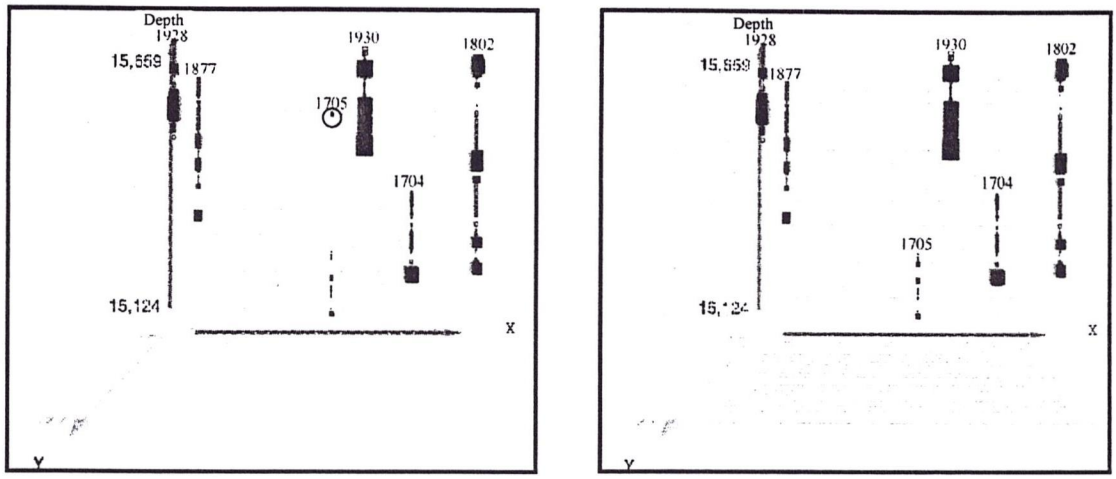
Table 3: Permeability distribution of 0's and <0.01 values.

Oilwell	Cases	0 values	<0.01 values	%
1704	159	66	0	41.5
1705	129	0	78	60.4
1802	249	121	0	48.6
1877	182	0	58	31.8
1928	130	0	38	29.2
1930	133	0	15	11.3
TOTAL	982	187	189	38.3

All 0's as well as <0.01 permeability values were substituted for the value 0.009. This means that any values generated by either the CN2 or the C4.5-rules algorithms with *permeability=0.009* should be interpreted as *permeability<0.01*.

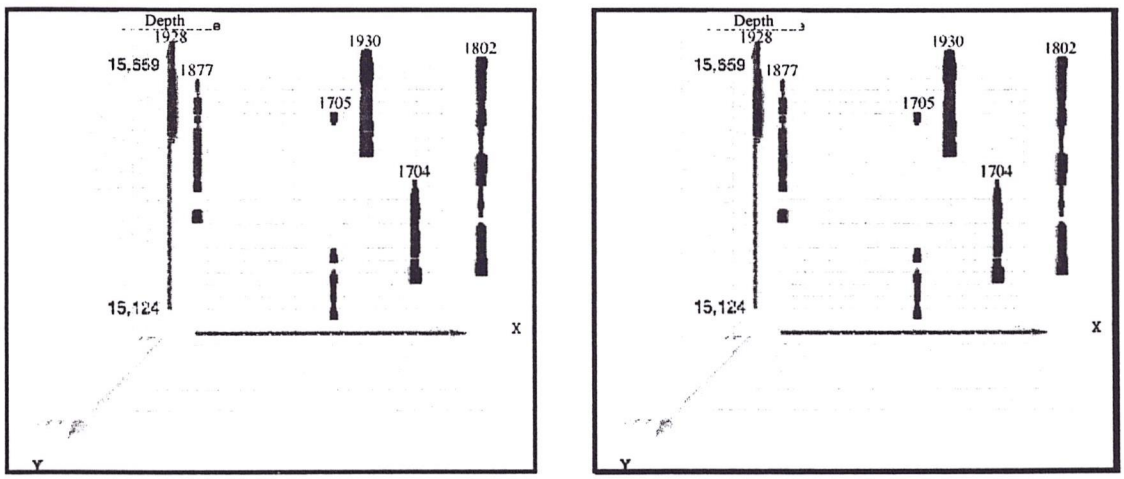
Being able to visualize the data beforehand often allows for a better understanding of the problem domain, but it is even more important to have knowledge of the domain prior to the visualization so as to determine the type of graph and the information that should be considered. We used the Scatter Visualizer® tool for the first visualization of the data, with squares of the graph representing the values of the attributes. A small square indicates a small value and a large square indicates a large value. In Figure 3(a) and Figure 4(a), the graphs of the analysis of permeability and porosity in relation to the depth are shown.

In the graphs shown in Figure 3(a) and Figure 4(a), a high absence of data was detected in oilwell 1705. By analyzing the original data, we were able to conclude that there was probably an error in the original data, because of its sequence and the values of the other attributes. Notice that after the depth of 15220 feet the next value is 15521 feet, after which comes 15541 and then 15242. The data was corrected and the result seen in all the figures in this text uses this correction. Table 4 shows the original data and the corrected data after this consideration.



(a) (b)

Figure 3: (a) Visualization of the permeability with an error detected in oilwell 1705
(b) Permeability after having been corrected.



(a) (b)

Figure 4: (a) Visualization of the porosity with an error detected in oilwell 1705
(b) Porosity after having been corrected.

Analyzing the graphs in Figure 3(b) and Figure 4(b), high entropy can be observed among the values of permeability, compared with the values of porosity, which have a more homogeneous behavior. It can also be observed that oilwell 1930 has larger porosity and permeability values in comparison to those of the other oilwells. It is also interesting to note that oilwells 1704 and 1802 have smaller permeability values, but much more porosity in relationship to the other oilwells.

In Figure 5, a statistical analysis of the data after this substitution, including mean, median, standard deviation, histograms and quartiles, is shown in graphical form. Similar results are shown in Table 5.

Table 4: Error detected with the visualization and later corrected for oilwell 1705.

Original Data			Corrected Data		
Depth	**Porosity**	**Permeability**	**Depth**	**Porosity**	**Permeability**
15217	2	<0.01	15217	2	0.009
15218	0.7	<0.01	15218	0.7	0.009
15219	0.7	<0.01	15219	0.7	0.009
15220	0.6	<0.01	15220	0.6	0.009
→15521	0.7	<0.01	15221	0.7	0.009
15522	0.7	<0.01	15222	0.7	0.009
15523	1.3	<0.01	15223	1.3	0.009
15524	1.1	<0.01	15224	1.1	0.009
....	....	....	....	....	....
....	....	....	....	....	....
15538	9.4	0.9	15238	9.4	0.9
15539	9.8	0.8	15239	9.8	0.8
→15540	7.8	0.05	15240	7.8	0.05
15541	0.8	<0.01	15241	0.8	0.009
15242	7.1	0.57	15242	7.1	0.57
15243	8.3	0.05	15243	8.3	0.05
15244	15.1	0.34	15244	15.1	0.34
15245	11.7	0.29	15245	11.7	0.29

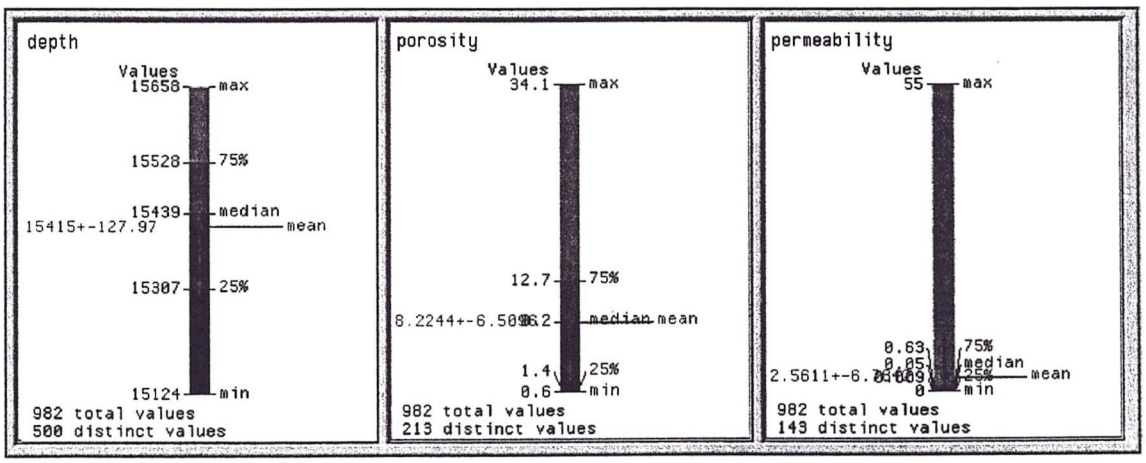


Figure 5: Statistical results from the six oilwells using Statistics Visualizer™ (982 cases).

Table 5: Statistical description of the six oilwells (982 cases).

	Depth	**Porosity**	**Permeability**
Distinct Values	500	213	142
[Max,Min] Values	[15568,15124]	[34.1,0.6]	[55,0.009]
Mean	15415±127.9	8.2±6.5	2.5±6.7
Median	15439	8.2	0.05

Figure 6 shows information similar to that shown in Figure 2. Only the 982 cases from which the results of this work were obtained are presented.

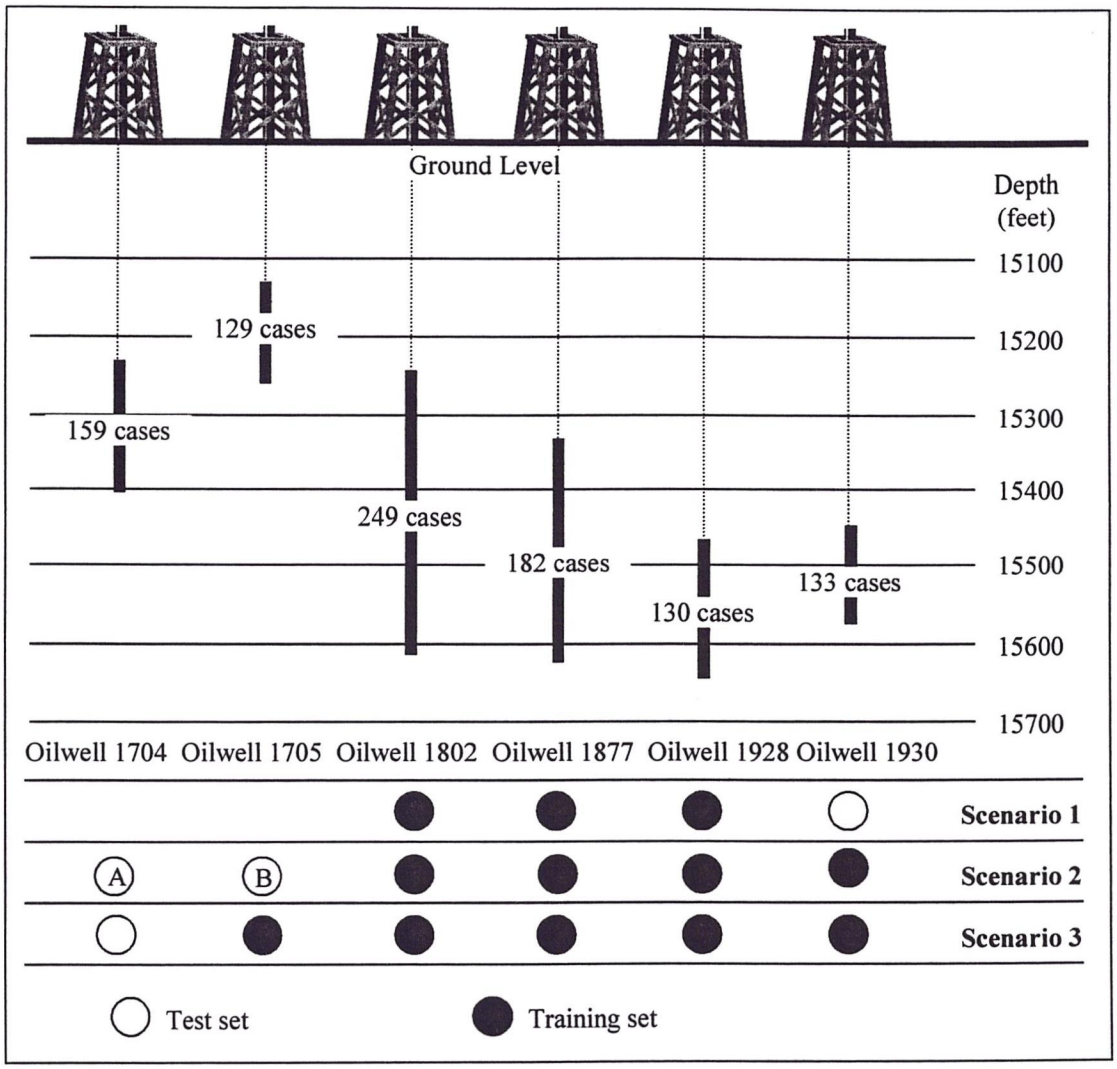


Figure 6: Information from the six oilwells after cleaning the data (total 982 cases).

4.2.2 Discretization Process

After this initial analysis and data correction, we concentrated on the process of discretizing the *attribute class permeability* so that we could use the classification supervised learning algorithms, CN2 and C4.5-rules.

In this work, three discretizing methods were applied using MineSet™:

1) a *stand-alone* discretization of the attribute class *permeability* based on frequency distribution;
2) an empirical method suggested by an expert, which discretizes the attribute class *permeability* considering the porosity values;
3) a hybrid method which tries to improve on the discretization suggested by the domain expert with respect to the precision of the CN2 and C4.5-rules algorithms, by varying the discretization intervals.

There are many different criteria for making a discretization, as was seen in Section 2. After an analysis and a number of tests, the supervised, parameterized and local criteria were chosen for making a *stand-alone* discretization. The number of intervals generated

for the attribute class *permeability* was specified beforehand, without considering the other attributes (*porosity* and *depth*), attempting to distribute the data among the discretization intervals. Discretizations were made by gathering the data from all six oilwells (982 cases), so as to make the correct comparisons of the three scenarios. In this work, three discretizations were made, forming 2, 3 and 5 groups.

According to the expert, each terrain where the drillings were made is unique, which makes it difficult to determine the discretization intervals for these scenarios. The expert who offers the best discretization, in general is the one who knows a particular kind of terrain, since the values of permeability are different for each case. It is also important to know, according to the expert, that in most cases a large porosity in the rocks indicates the possibility of petroleum in that area. Consequently, to determine the criteria for discretization it is important to consider the porosity of the oilwell. Although the expert did not determine the discretization values, he suggested discretizing the permeability using the corresponding porosity value as a weight. The task became to use MineSet™, which also has a capability for discretizing an attribute considering the weight of another attribute, for determining the discretizations. The results of the different discretizations using the *stand-alone* uniform weight method and the method suggested by the expert are shown in Table 6.

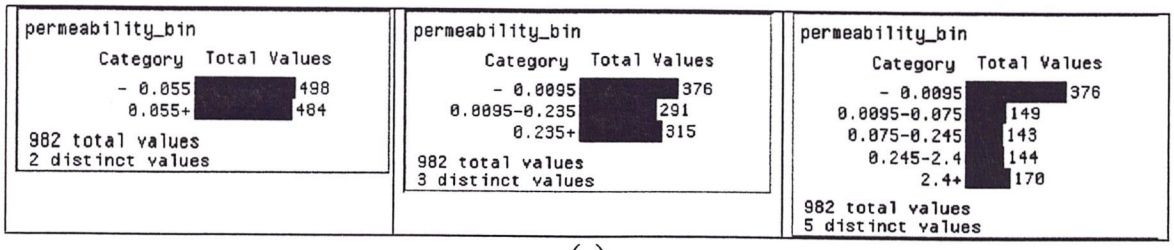
Table 6: Intervals obtained by the two discretization methods.

	***Stand-alone* discretization**	**Discretization suggested by the expert**
2 intervals:	<0.055, >0.055	<0.445, >0.445
3 intervals:	<0.0095, [0.0095,0.235], >0.235	<0.155, [0.155,2.4], >2.4
5 intervals:	<0.0095, [0.0095,0.075[, [0.075,0.245[, [0.245,2.4], >2.4	<0.055, [0.055,0.215[, [0.215,1.05[, [1.05,7.85], >7.85

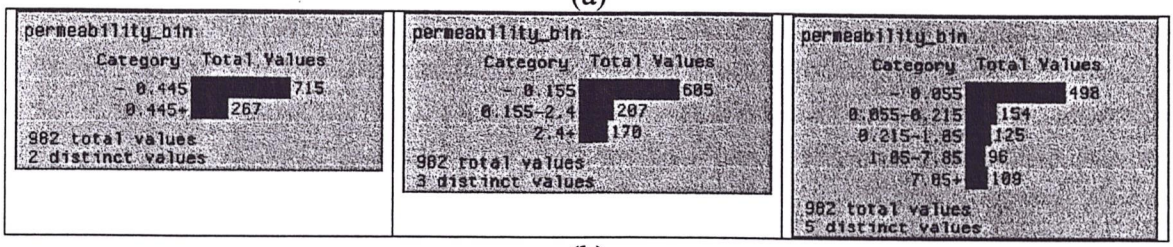
In Figure 7(a) and Figure 7(b) the different distributions of the discretization values are represented graphically, using the *stand-alone* method and the method suggested by the expert. The discretization algorithm used in these experiments was entropy, because it presented the best results in experiments done in other works related to the subject (Dougherty et al., 1995).

The results of the two discretization methods used in this work are presented in the next section.

It should be noted that some other criteria for discretization were also applied, such as the non-parameterized criteria case, where the discretization algorithm itself tries to find the number of discretization intervals automatically. In this case, 12 discrete intervals were found, with a large amount of error being generated when applying the CN2 and C4.5-rules rule induction algorithms.



(a)



(b)

Figure 7: (a) Distribution of 2, 3 and 5 discretizations using the *stand-alone* method. (b) Distribution of 2, 3 and 5 discretizations using considerations from the expert.

4.2.3 Classification

In these experiments the CN2 (Clark & Niblett, 1989) and C4.5-rules (Quinlan, 1987) algorithms were used to determine and compare the error rates using the different discretizations for the attribute class *permeability*. Both algorithms were executed using MLC++ (*Machine Learning Library in C++*) (Kohavi et al., 1994; Molina et al., 1998). In both cases, the algorithms were run with the *default* parameters from the MLC++ library. The resulting error rates obtained when running both algorithms for the three scenarios as well as the majority class error rate of the training set in each scenario, are presented in Tables 7,8 and 9.

The results show that only in a few cases the error rate is relatively small for both the CN2 and C4.5-rules algorithms. Furthermore, it can be observed that there is a high variation in the error rates. Both in Scenario 1 using 3 and 5 discretizations and in Scenario 2 using 3 discretizations suggested by the expert, the error rate is higher than the majority class error rate. This is unacceptable.

We now turn to study a particular scenario with a fixed discretization, aiming to improve discretization and obtain better classification rules. Based on previous considerations, scenario 2 (with 3 discretizations given by the expert using test set A) was chosen for this study. The confusion matrix for this case is presented in Table 10.

Table 7: Error Rate for Scenario 1.

	***Stand-alone* discretization**			**Discretization suggested by the expert**		
No. of discretizations	**CN2 Error Rate**	**C4.5-rules Error Rate**	**Majority Class Error Rate**	**CN2 Error Rate**	**C4.5-rules Error Rate**	**Majority Class Error Rate**
2	12.8 %	22.6 %	47.2%	23.3%	13.5%	26.7%
3	36.1 %	27.1 %	61.3%	42.1%	28.6%	36.5%
5	46.6 %	41.4 %	61.3%	67.7%	68.4%	47.2%

Table 8: Error Rate for Scenario 2.

	Stand-alone discretization			Discretization suggested by the expert		
No. of discretizations	**CN2 Error Rate**	**C4.5-rules Error Rate**	**Majority Class Error Rate**	**CN2 Error Rate**	**C4.5-rules Error Rate**	**Majority Class Error Rate**
2 test set A	13.8 %	6.9 %	47.3%	8.8%	10.1%	30.8%
2 test set B	13.2 %	9.3 %	47.3%	6.2%	5.4%	30.8%
3 test set A	17.0 %	16.4 %	64.7%	30.2%	17.0%	42.2%
3 test set B	7.8%	7.8 %	64.7%	5.4%	7.8%	42.2%
5 test set A	28.9 %	27.7 %	66.6%	34.0%	32.7%	52.7%
5 test set B	22.5 %	22.5 %	66.6%	25.6%	19.4%	52.7%

Table 9: Error Rate for Scenario 3.

	Stand-alone discretization			Discretization suggested by the expert		
No. of discretizations	**CN2 Error Rate**	**C4.5-rules Error Rate**	**Majority Class Error Rate**	**CN2 Error Rate**	**C4.5-rules Error Rate**	**Majority Class Error Rate**
2	14.5 %	6.9 %	48.8%	8.8%	10.1%	27.2%
3	21.4 %	15.7 %	62.3%	29.6%	18.2%	37.8%
5	27.7 %	25.2 %	62.3%	28.9%	32.1%	48.8%

Table 10: Confusion matrix for scenario 2 with 3 discretizations done with the aid of an expert using test set A

Discretization	**CN2**					**C4.5-rules**				
		(a)	(b)	(c)	Error		(a)	(b)	(c)	Error
<.155	(a)	90	3	0	3.2%	(a)	92	1	0	1.0%
[.155,2.4],	(b)	41	9	14	83.3%	(b)	25	28	1	48.1%
>2.4	(c)	0	0	12	0%	(c)	0	0	12	0%

The frequency distribution for the permeability values of oilwell 1704, which was used as the test set in this scenario, are presented in Figure 8. Value "0.009" contains 66 elements which are not shown in this graph.

Trying to improve the precision of the results from the CN2 and C4.5-rules algorithms, we set one limit of the interval at a fixed value and varied the other limit with higher and lower values. For each variation, the error rates for both algorithms were determined. We started by setting the upper limit at 2.4 and varying the lower limit (0.155). The results of the experiments are presented in Figure 9. The lowest error rates were found for the CN2 algorithm with the lower limit set at 0.015, 0.025 and 0.035. For the C4.5-rules algorithm, the lowest error rates were found with the lower limit set at 0.015, 0.025 and 0.035.

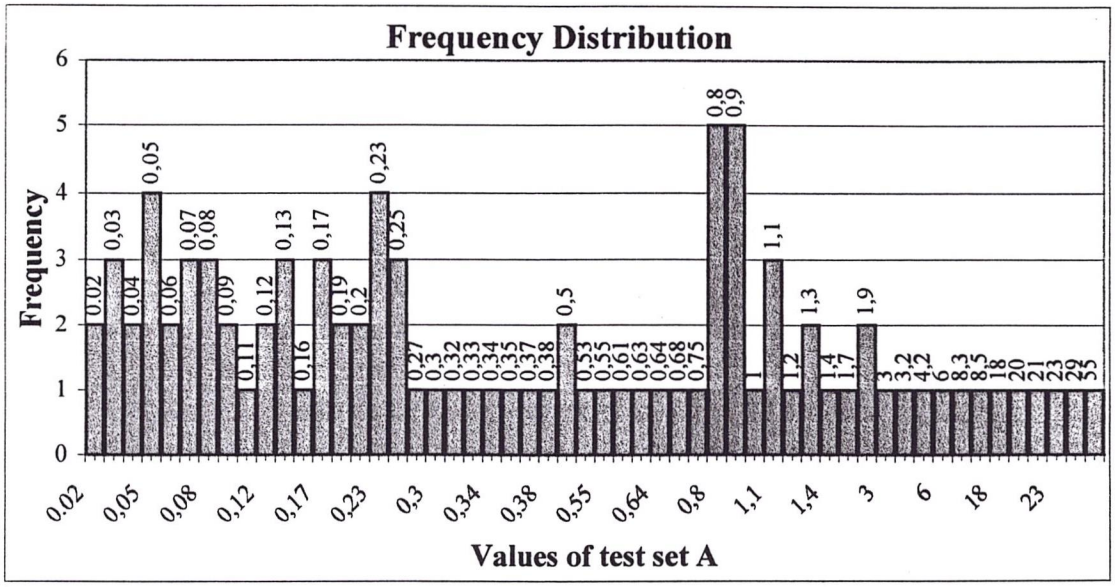


Figure 8: Frequency distribution for the permeability values of oilwell 1702.

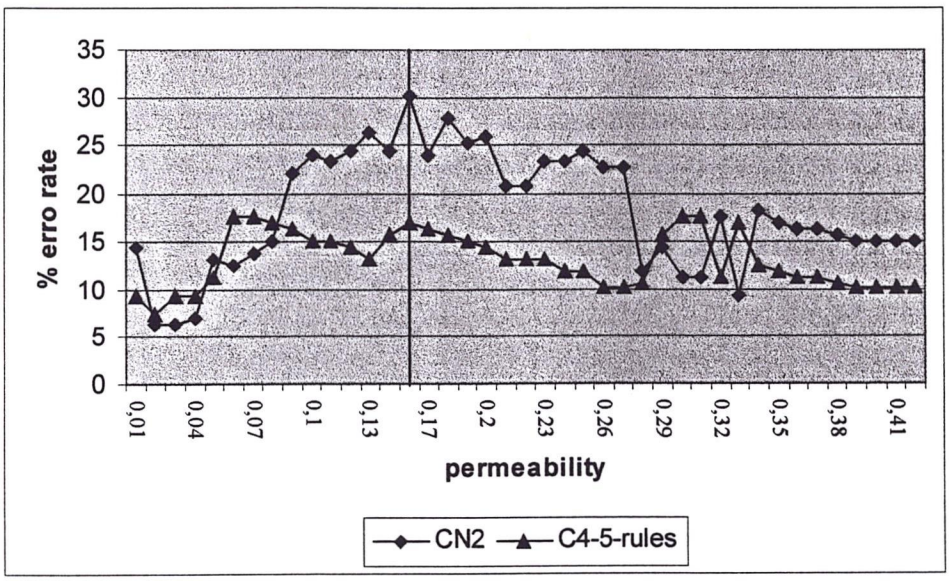


Figure 9: Error rates presented by the CN2 and C4.5-rules algorithms, setting the upper limit at 2.4 and varying the lower limit.

Next, the lower limit, 0.155, was set and the upper limit, 2.4, was varied using higher and lower values. In Figure 10, the results of the experiments are shown. Values 0.65 and 0.75 had the lowest error rates for the CN2 algorithm and values 1.75 and 1.85 had the lowest error rates for the C4.5-rules algorithm. It is important to note at this point that only the value 1.75, and not 1.85, was used in the other tests, since the 2 values have similar results.

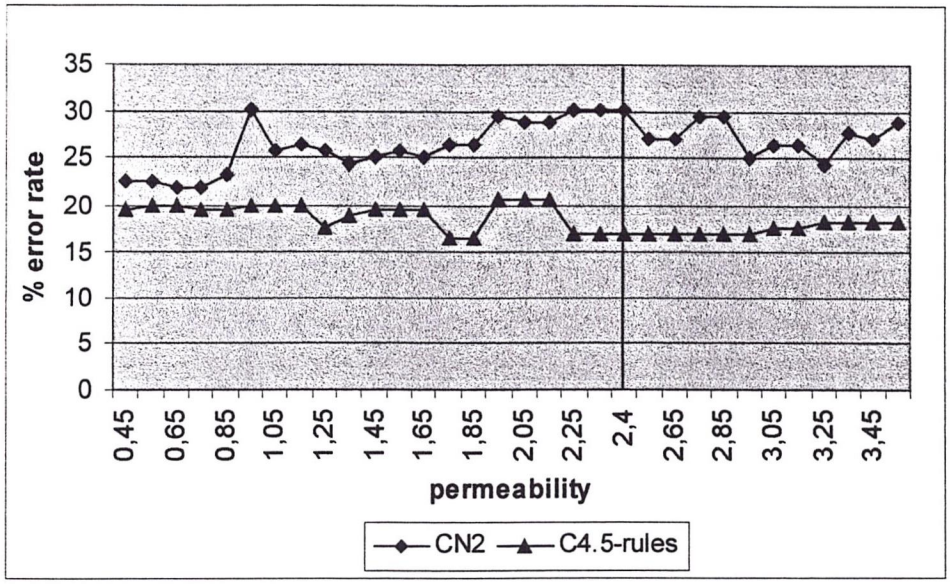


Figure 10: Error rates presented by the CN2 and C4.5-rules algorithms, setting the lower limit at 0.155 and varying the upper limit.

Based on the lowest error rates presented by the algorithms, the best lower limits closest to 0.015 and the best upper limits near 2.4 were chosen. Having done this, the confusion matrices were generated for each case, trying to determine the interval with the smallest error rate and considering the distribution of the datasets. In this scenario 694 examples were used for the training set and 159 examples for the test set. The confusion matrices for CN2, considering the lowest error rates for this algorithm, are presented in Appendix I. Similar results for the C4.5-rules algorithm considering the lowest error rates are presented in Appendix II.

Based on these results, the discretizations with the lowest error rate and a relative distribution in its data were "0.015 and 0.75" and "0.015 and 1.75". Considering that these points worked well for this scenario, they were placed in the other scenarios to determine its behavior. The confusion matrices using these two new discretization intervals are presented in Appendix III.

4.3. Results

Table 11 shows a comparison between the results of the discretization with the aid of an expert and the discretization using the cut points "0.015 and 1.75", which is where the best results in all of the scenarios were obtained.

It can be observed that, except for scenario 1, the new discretization improves accuracy. One of the goals of transforming a regression problem into a classification problem is to improve human understanding, by expressing, with a set of rules, concepts learned by the system in a simpler way.

Table 11: Comparison between the discretization with the aid of an expert and the discretization using the cut points 0.015 and 1.75.

	Discretization suggested by the expert with [0.155,2.4]			Discretization using [0.015,1.75]		
Scenario	**CN2 Error Rate**	**C4.5-rules Error Rate**	**Majority Class Error Rate**	**CN2 Error Rate**	**C4.5-rules Error Rate**	**Majority Class Error Rate**
1	42.1%	28.6 %	36.5%	38.3%	39.8%	64.4%
2 test set A	30.2%	17.0 %	42.2%	13.8%	6.9 %	59.9%
2 test set B	5.4%	7.8 %	42.2%	3.9%	3.1%	59.9%
3	29.6%	18.2 %	37.8%	8.2%	5.7 %	56.7%

The C4.5 algorithm induced 10 rules by learning with the training set for scenario 2 (694 cases). The results of the learning are shown in Appendix IV. The rules that were generated do not cover 99 cases (14.3%) of these 694 cases. Below the results of applying three rules to the test set are presented. The conclusion of each rule shows the class as well as information in the following format: [AA%] [BB CC] [DD%] [EE FF], where [AA%] is the classification error when applying the rule to the training set, [BB CC] is the number of examples correctly and erroneously classified for the same application, [DD%] is the classification error when applying the rule to the test set and [EE FF] is the number of examples correctly and erroneously classified for this second application.

Rule 20:

porosity > 15.7
-> class > 1,75 [11,1%] [112 14] [14,3%] [12 2]

Rule 1:

porosity <= 2.9
-> class < 0,015 [0,0%] [203 0] [1,6%][60 1]

Rule 19:

porosity > 3.9
porosity <= 15.7
-> class [0,015:1,75] [17,5%] [188 40] [7,3%][76 6]

For the same scenario and training set (694 cases), CN2 generated 120 rules that do not cover 18 cases (2.6 %). However, more than 50% of these 120 rules are specialized to cover 1, 2 or 3 examples.

The experiment results show that the selection of the discretization directly affects the precision of the CN2 and C4.5-rules algorithms. When making a *stand-alone* discretization, the C4.5-rules algorithm presents a better precision in comparison with that of CN2 in 83.3% of the cases and 8.3% present the same results. When the discretization is done with the aid of an expert, C4.5-rules presents a better result than CN2 in 41.6% of the cases, and in 25% of them it presents the same results.

It is important to emphasize that, besides determining the precision for each set in the confusion matrix, the element distribution must be analyzed. For example, in Table 10, a precision of 100% was determined with CN2 for 12 elements of set *c*, but this isn't as

relevant, when compared with set *a* for CN2, which presents a precision of 96.8% for 93 elements.

5. Conclusion

In this work we described a method for transforming a regression problem into a classification problem. Although our work was oriented toward the CN2 and C4.5-rules rule learning algorithms, it can be used as a strategy for using other classification learning algorithms.

One of the advantages of transforming a regression problem into a classification problem is the rules that are obtained in the process. By expressing them as symbolic knowledge, the rules learned by the system can be represented in a way that human beings can understand.

The hybrid discretization discussed in this work considerably improved the precision of the machine learning algorithms in all the scenarios, except for Scenario 1. One alternative could be to apply the hybrid method locally to this scenario, so as to improve the precision even more. The problem with discretizing each scenario locally is that each scenario will have different discretization intervals, making it difficult to evaluate the three scenarios together.

Another problem is the data has shown itself to be inappropriate for a classification problem, especially because of the number of undetected values (38.3% of the whole set). There is also a lack of uniformity in the data. Thus, the problem should be treated as a regression problem, for example using Neural Networks, rather than as a classification problem.

It was also observed that, although the hybrid method was considerably better for obtaining more accurate results from the Machine Learning algorithms, the domain expert has a fundamental role in the definition of the discretization. By obtaining discretizations with the help of *stand-alone* methods, the domain expert can also get important help when determining the best intervals. In addition to all this, the number of intervals can influence the precision of the ML algorithms and the hybrid method tends to increase the complexity of the problem by adding more intervals.

An empirical comparison with many discretizations was presented using a *stand-alone* method with statistical methods and a method with the orientation of an expert. The main objective of the experiment was to obtain precise and "good" symbolic rules for determining the behavior of the permeability of an oilwell by analyzing the depth, porosity and permeability of the neighboring oilwells.

This work presented a case study which shows the fundamental role that discretization plays in obtaining more precise rules. There are many methods for discretization, and a good choice depends on a detailed analysis of the domain. The role of the expert in this choice is also fundamental. At the same time, obtaining discretizations using *stand-alone* methods offers an important help to the expert when determining cut points. It is also good to emphasize that the number of discretization intervals considerably influences the final result.

6. Acknowledgments

The authors would like to thank Prof. Dr. Edson Rodrigues of the Molecular Physics Group at the Chemistry Institute of São Carlos, for the orientation as to how to do the discretizations in this work, as well as Jaqueline Brigladori Pugliesi for the valuable comments and help given while writing this text. This work was done with the support of the Brazilian Cooperative Research Network (RECOPE-FINEP), Research Council (CNPq) and the Mexican Petroleum Institute.

Bibliography

Breiman, L., Friedman, J., Olshen R. and Stone C., *Classification and Regression Trees*, Chapman and Hall, New York, 1993.

Catlett, J., *Megainduction: Machine Learning on Very Large Databases*, Ph.D. Thesis, University of Sydney, 1991.

Chmielewski, M. and Grzymala-Busse, J., *Global Discretization of Continuous Attributes as Preprocessing for Machine Learning*, In: Lin, T.Y. and Wildberger, A.M., (eds.), Soft Computing, Society for Computer Simulation, San Diego, 1995, pp. 294-301.

Clark, P. and Niblett, T., *The CN2 Induction Algorithm*, Machine Learning, Vol. 3, 1989, pp. 261-283.

Dougherty, J., Kohavi R. and Sahami M., *Supervised and Unsupervised Discretizations of Continuous Features*, In: Proceedings of the 12th International Conference on Machine Learning, Morgan Kaufmann Publishers, 1995, pp. 194-202.

Fayyad, U.M., Piatesky-Shapiro, G., Smyth, P. and Uthurusamy, R., *Advanced in Knowledge and Data Mining*, AAAI/MIT Press, Menlo Park, CA., 1996.

Fayyad, U.M. and Irani K.B., *Multi-interval discretization of continuous-valued attributes for classification learning*, In: Proceedings of the 13th International Conference on Machine Learning, Morgan Kaufmann Publishers, 1993, pp. 1022-1027.

Grzymala-Busse, W., *LERS - A system for learning from examples based on rough sets*, In: Slowinski R., (ed.), Intelligence Decision Support. Handbook of Applications and Advances of the Rough Set Theory, Kluwer Academic Publishers, Dordrecht, 1992, pp. 3-18.

Kerber, R., *ChiMerge: Discretization of Numeric Attributes*, In: Proceedings of the 10th National Conference on Artificial Intelligence, 1992, pp. 123-127.

Kohavi, R., John, G., Long, R. Manley, D. and Pfleger, K., *MLC++: A Machine Learning Library in C++*, In: Tool with Artificial Intelligence, IEEE Computer Society Press, 1994, pp. 740-743.

Lenarcik, A., and Piasta, Z., *Discretization of Condition Attribute Space*, In: Slowinski

R., (ed.), Intelligence Decision Support. Handbook of Applications and Advances of the Rough Set Theory, Kluwer Academic Publishers, Dordrecht, 1992, pp. 373-389.

Lenarcik, A. and Piasta, Z., *Probabilistic Approach to Decision Algorithm Generation in the Case of Continuous Condition Attributes*, Foundations of Computing and Decision Sciences, Vol. 18, No. 3-4, Poznan, Poland, 1993, pp. 213-224.

Lenarcik, A. and Piasta, Z., *Minimizing the Number of Rules in Deterministic Rough Classifiers*, In: Lin, T.Y. and Wildberger, A.M., (eds.), Soft Computing, Society for Computer Simulation, San Diego, 1995, pp. 32-35.

Molina, F.L.C., Oliveira, R.S., Doi, C.Y., De Paula, M.F. e Romanato, M.J., *MLC++: Biblioteca de Aprendizado de Máquina em C++*, Technical Report 72, ISSN-0103-2569, ICMSC, USP, São Paulo, Brazil, 1998.

Nguyen, S.H. and Skowron, A., *Quantization of Real Value Attributes: Rough Set and Boolean Reasoning Approach*, In: Proceedings of the Second Joint Annual Conference on Information Sciences, Wrightsville Beach, North Carolina, 1995, pp. 34-37.

Pfahringer, B., *Compression-based discretization of continuous attributes*, In: Prieditis A. and Russel S., (eds.), Proceedings of the Twelfth International Conference on Machine Learning, Morgan Kaufmann Publishers, 1995.

Quinlan, J.R., *Generating production rules from decision trees*, In: Proceedings of Fourth International Machine Learning Workshop, Morgan Kaufmann, San Mateo, CA., 1987, pp. 304-307.

Quinlan, J.R., *C4.5 Programs for Machine Learning*, Morgan Kaufmann Publishers, San Mateo, California, 1993.

Quinlan, J.R., *Induction of Decision Trees*, In: Shavlik, J.W. and Dietterich, T.G., (eds), Readings in Machine Learning, Morgan Kaufmann Publishers, Inc., San Mateo, California, 1990, pp. 57-69.

Rogers, S.J., Chen, H.C., Kopaska-Merkel, D.C. and Fang, J.H., *Predicting Permeability from Porosity Using Artificial Neural Networks*, American Association of Petroleum Geologists Bulletin, v. 79, December 1995, pp. 1786-1797.

Silicon Graphics, *MineSet*, (1997), Available at http://www.sgi.com/Products/software/MineSet/

Ventura D. and Martinez T.R., *An Empirical Comparison of Discretization Methods*, In: Proceedings of the Tenth International Symposium on Computer and Information Sciences, 1995, pp. 443-450.

Ventura D. and Martinez T.R., *BRACE: A Paradigm For the Discretization of Continuously Valued Data*, In: Proceedings of the Seventh Florida Artificial

Intelligence Research Symposium, 1994, pp. 117-121.

Water Resource Research Center, *Glossary of Organizations and Acronyms*, College of Agriculture, The University of Arizona, (1998), [Online] Available. URL: http://Ag.Arizona.Edu/Azwater/Gloss.Html

Appendix I. Lowest error rates for the CN2 algorithm.

Interval	CN2					C4.5-rules					
		(a)	(b)	(c)	Error %		(a)	(b)	(c)	Error %	Majority Class Error rate %
<.015 [.015,0.65] >0.65	(a)	61	5	0	7.5	(a)	60	6	0	9.1	
	(b)	11	41	5	28	(b)	1	50	6	12.2	
	(c)	4	4	28	22.2	(c)	0	7	29	19.4	
TOTAL					18.2					12.6	59.9

Interval	CN2					C4.5-rules					
		(a)	(b)	(c)	Error %		(a)	(b)	(c)	Error %	Majority Class Error rate %
<.015 [.015,0.75] >0.75	(a)	63	3	0	4.5	(a)	60	6	0	9.1	
	(b)	2	58	0	3.3	(b)	1	54	5	10	
	(c)	0	18	15	54.5	(c)	0	7	26	21.2	
TOTAL					14.5					11.9	59.9

Interval	CN2					C4.5-rules					
		(a)	(b)	(c)	Error %		(a)	(b)	(c)	Error %	Majority Class Error rate %
<.025 [.025,0.65] >0.65	(a)	64	4	0	5.8	(a)	61	7	0	10.2	
	(b)	15	35	5	36.3	(b)	0	49	6	10.9	
	(c)	2	6	28	22.2	(c)	0	7	29	19.4	
TOTAL					20.1					12.6	57.8

Interval	CN2					C4.5-rules					
		(a)	(b)	(c)	Error %		(a)	(b)	(c)	Error %	Majority Class Error rate %
<.025 [.025,0.75] >0.75	(a)	62	6	0	8.8	(a)	61	7	0	10.2	
	(b)	8	44	6	24.1	(b)	0	53	5	8.6	
	(c)	0	7	26	21.2	(c)	0	7	26	21.2	
TOTAL					17					11.9	57.8

Interval	CN2					C4.5-rules					
		(a)	(b)	(c)	Error %		(a)	(b)	(c)	Error %	Majority Class Error rate %
<.035 [.035,0.65] >0.65	(a)	65	6	0	8.4	(a)	61	10	0	14	
	(b)	7	40	5	23	(b)	0	46	6	11.5	
	(c)	1	7	28	22.2	(c)	0	7	29	19.4	
TOTAL					16.4					14.5	56.5

Interval	CN2					C4.5-rules					
		(a)	(b)	(c)	Error %		(a)	(b)	(c)	Error %	Majority Class Error rate %
<.035 [.035,0.75] >0.75	(a)	65	6	0	8.4	(a)	61	10	0	14	
	(b)	5	45	5	18.1	(b)	0	54	1	1.8	
	(c)	0	7	26	21.2	(c)	0	12	21	36.3	
TOTAL					14.5					14.5	56.5

Appendix II. Lowest error rates for the C4.5-rules algorithm.

Interval	CN2					C4.5-rules					
<.015 [.015,1.75] >1.75		(a)	(b)	(c)	Error %		(a)	(b)	(c)	Error %	Majority Class Error rate %
	(a)	62	4	0	6	(a)	60	6	0	9	
	(b)	11	63	5	20.2	(b)	1	76	2	3.7	
	(c)	1	1	12	14.2	(c)	0	2	12	14.2	
TOTAL					13.8					6.9	59.9

Interval	CN2					C4.5-rules					
<.025 [.025,1.75] >1.75		(a)	(b)	(c)	Error %		(a)	(b)	(c)	Error %	Majority Class Error rate %
	(a)	62	6	0	8.8	(a)	61	7	0	10.2	
	(b)	17	54	6	29.8	(b)	0	75	2	2.5	
	(c)	1	1	12	14.2	(c)	0	2	12	14.2	
TOTAL					19.5					6.9	57.8

Interval	CN2					C4.5-rules					
<.035 [.035,1.75] >1.75		(a)	(b)	(c)	Error %		(a)	(b)	(c)	Error %	Majority Class Error rate %
	(a)	63	8	0	11.2	(a)	61	10	0	14	
	(b)	16	52	6	29.7	(b)	0	72	2	2.7	
	(c)	0	2	12	14.2	(c)	0	2	12	14.2	
TOTAL					20.1					8.8	56.5

Appendix III. Confusion Matrices using the discretization "0.015 and 0.75" and "0.015 and 1.75".

Confusion Matrices using the cut points "0.015 and 0.75" in all the scenarios.

Scenario	CN2					C4.5-rules					
		(a)	(b)	(c)	Error %		(a)	(b)	(c)	Error %	Majority Class Error rate %
1	(a)	15	5	2	31.8	(a)	14	7	1	36.3	
	(b)	10	23	18	54.9	(b)	2	32	17	37.2	
	(c)	4	10	46	23.3	(c)	0	14	46	23.3	
TOTAL					36.8					30.8	54.4

Scenario	CN2					C4.5-rules					
		(a)	(b)	(c)	Error %		(a)	(b)	(c)	Error %	Majority Class Error rate %
2 B	(a)	78	0	0	0	(a)	77	1	0	1.2	
	(b)	5	35	4	20.4	(b)	1	39	4	11.3	
	(c)	2	2	3	57.1	(c)	0	5	2	71.4	
TOTAL					10.1					8.5	59.9

Scenario	CN2					C4.5-rules					
		(a)	(b)	(c)	Error %		(a)	(b)	(c)	Error %	Majority Class Error rate %
3	(a)	62	4	0	6	(a)	62	4	0	6	
	(b)	7	46	7	23.3	(b)	1	58	1	3.3	
	(c)	0	7	26	21.2	(c)	0	12	21	36.3	
TOTAL					15.7					11.3	56.7

Confusion matrices using the cut points "0.015 and 1.75" in all the scenarios.

Scenario	CN2					C4.5-rules					
		(a)	(b)	(c)	Error %		(a)	(b)	(c)	Error %	Majority Class Error rate %
1	(a)	17	4	1	22.7	(a)	16	2	4	27.2	
	(b)	14	30	16	50	(b)	8	21	31	65	
	(c)	1	15	35	31.3	(c)	0	8	43	15.6	
TOTAL					38.3					39.8	54.4

Scenario	CN2					C4.5-rules					
		(a)	(b)	(c)	Error %		(a)	(b)	(c)	Error %	Majority Class Error rate %
2 B	(a)	77	1	0	1.2	(a)	77	1	0	1.2	
	(b)	1	47	0	2	(b)	1	47	0	2	
	(c)	1	2	0	0	(c)	0	2	1	66.6	
TOTAL					3.9					3.1	59.9

Scenario	CN2					C4.5-rules					
		(a)	(b)	(c)	Error %		(a)	(b)	(c)	Error %	Majority Class Error rate %
3	(a)	62	4	0	6	(a)	62	4	0	6	
	(b)	6	72	1	8.8	(b)	1	76	2	3.7	
	(c)	0	2	12	14.2	(c)	0	2	12	14.2	
TOTAL					8.2					5.7	56.7

Appendix IV. Rules generated by the C4.5-rules algorithm

C4.5 [release 8] rule generator Thu Jul 9 03:04:38 1998

Options:

Rulesets evaluated on unseen cases

File stem </var/tmp/AAAa000ox>

Read 694 cases (2 attributes) from /var/tmp/AAAa000ox

Processing tree 0

Final rules from tree 0:

Rule 20:

porosity > 15.7

-> class 1.75+ [86.4%]

Rule 18:

depth > 15599

porosity > 3.9

-> class 1.75+ [79.4%]

Rule 12:

depth > 15426

depth <= 15496

porosity > 10.8

-> class 1.75+ [78.7%]

Rule 17:

depth > 15582

porosity > 11.1

-> class 1.75+ [64.5%]

Rule 1:

porosity <= 2.9

-> class - 0.015 [99.3%]

Rule 4:

depth > 15412

porosity <= 3.9

-> class - 0.015 [96.3%]

Rule 16:

depth > 15582

depth <= 15599

porosity <= 11.1

-> class - 0.015 [86.7%]

Rule 14:

depth > 15515

porosity <= 10.3

-> class - 0.015 [77.2%]

Rule 19:

porosity > 3.9
porosity <= 15.7
-> class 0.015-1.75 [65.3%]

Rule 3:

porosity > 2.9
porosity <= 3.2
-> class 0.015-1.75 [54.6%]

Default class: 0.015-1.75

Evaluation on training data (694 items):

Rule	Size	Error	Used	Wrong	Advantage	Class
20	1	13.6%	126	14 (11.1%)	53 (64|11)	>1.75+
18	2	20.6%	4	0 (0.0%)	4 (4|0)	>1.75+
12	3	21.3%	33	11 (33.3%)	11 (22|11)	>1.75+
17	2	35.5%	6	2 (33.3%)	3 (4|1)	>1.75+
1	1	0.7%	203	0 (0.0%)	56 (56|0)	< 0.015
4	2	3.7%	22	4 (18.2%)	1 (4|3)	< 0.015
16	3	13.3%	8	1 (12.5%)	3 (3|0)	< 0.015
14	2	22.8%	60	27 (45.0%)	6 (33|27)	< 0.015
19	2	34.7%	228	40 (17.5%)	0 (0|0)	[0.015,1.75[
3	2	45.4%	2	0 (0.0%)	0 (0|0)	[0.015,1.75]

Tested 694, errors 99 (14.3%) <<

(a)	(b)	(c)	<-classified as
261	16	1	(a): class - 0.015
32	192	26	(b): class 0.015-1.75
0	24	142	(c): class 1.75+

Evaluation on test data (159 items):

Rule	**Size**	**Error**	**Used**	**Wrong**	**Advantage**	**Class**
1	1	0.7%	61	1 (1.6%)	59 (60|1)	< 0.015
19	2	34.7%	82	6 (7.3%)	0 (0|0)	[0.015,1.75]
20	1	13.6%	14	2 (14.3%)	10 (12|2)	> 0.75

Tested 159, errors 11 (6.9%) <<

(a)	(b)	(c)	<-classified as
60	6	0	(a): class < 0.015
1	76	2	(b): class [0.015,1.75]
0	2	12	(c): class > 1.75