Università di Pisa

Project Report

Seismic Bumps Dataset

Data Mining: Fundamentals

Edited By:

Xhenis Jaku	[639180]	x.jaku@studenti.unipi.it
Marco D'Arrigo	[563441]	m.darrigo@studenti.unipi.it
Davide Chen	[544795]	d.chen@studenti.unipi.it
Alberto La Piccirella	[636680]	albertolapicc@gmail.com

Accademic year 2021/2022

Abstract

In this report we analyze with data mining tools the problem of forecasting high seismic bumps in a coal mine.

The data was taken by the polish "Institute of Computer Science, Silesian University of Technology" and the " Institute of Innovative Technologies", from a Polish coal mine.

"Mining activity was and is always connected with the occurrence of dangers which are commonly called mining hazards. A special case of such threat is a seismic hazard which frequently occurs in many underground mines. Seismic hazard is the hardest detectable and predictable of natural hazards and in this respect it is comparable to an earthquake. More and more advanced seismic and seismoacoustic monitoring systems allow a better understanding rock mass processes and definition of seismic hazard prediction methods."

Contents

1 Data Understanding & Preparation 1

1.1 Data Semantics 1

1.2 Distribution of the variables and statistics 2

1.3 Assessing data quality 4

1.4 Variable transformations 5

1.5 Pairwise correlations and eventual elimination of variables 5

2 Clustering 7

2.1 Clustering analysis by K-Means 7

2.2 Analysis by density-based clustering 9

2.3 Analysis by hierarchical clustering 10

2.4 Final discussion 11

3 Classification 12

3.1 Classification by Decision Tree 12

3.1.1 Applying classification 12

3.1.2 Identify the best parameter configuration 12

3.1.3 Decision tree 13

3.1.4 Performance evaluation 14

3.1.5 Confusion matrix 15

3.1.6 Precision-recall curve 15

3.2 Classification by K-NN 15

3.2.1 Confusion matrix 16

3.2.2 Precision Recall curve 16

3.3 Final discussion 16

4 Pattern Mining 17

4.1 Frequent Pattern extraction 17

4.2 Discuss Frequent Pattern 17

4.3 Association Rules extraction 18

4.4 Discuss Association rules 18

4.5 Exploit the most useful extracted rules 19

4.6 Final discussion 19

1. Data Understanding & Preparation

1.1 Data Semantics

In this section, we introduce the variables with their meaning and characteristics.
Dataset description: The data was taken by the polish "Institute of Computer Science, Silesian University of Technology" and the " Institute of Innovative Technologies", from a Polish coal mine. The dataset contains 2,584 records and 19 columns. The data is in different formats, so we divided the attributes in types as shown in Table 1.1.

Table 1.1: Variable Table

Attribute	**Description**	**Type**
seismic	Result of shift seismic hazard assessment in the mine working obtained by the seismic method (a - lack of hazard, b - low hazard, c - high hazard, d - danger state)	Categorical
seismoacoustic	Result of shift seismic hazard assessment in the mine working obtained by the seismoacoustic method (a - lack of hazard, b - low hazard, c - high hazard, d - danger state)	Categorical
shift	Information about type of a shift (W - coal-getting, N -preparation shift)	Categorical
hazard	Result of shift seismic hazard assessment in the mine working obtained by the seismoacoustic method based on registration coming form GMax only (a - lack of hazard, b - lowhazard, c - high hazard, d - danger state)	Categorical
genergy	Seismic energy recorded within previous shift by the most active geophone (GMax) out of geophones monitoring the longwall	Numerical
gpuls	A number of pulses recorded within previous shift by GMax	Numerical
gdenergy	A deviation of energy recorded within previous shift by GMax from average energyrecorded during eight previous shifts	Numerical
gdpuls	A deviation of a number of pulses recorded within previous shift by GMax from average number of pulses recorded during eight previous shifts	Numerical
energy	Total energy of seismic bumps registered within previous shift	Numerical
maxenergy	The maximum energy of the seismic bumps registered within previous shift	Numerical
nbumps	The number of seismic bumps recorded within previous shift	Discrete
nbumps i, for i ∈{2,...,9}	The number of seismic bumps in energy range $(10^i, 10^{i+1})$ [J] registered within previous shift	Discrete
class	–'1' high energy seismic bump occurred in the next shift ('hazardous state') –'0' no high energy seismic bumps occurred in the next shift ('nonhazardousstate')	Discrete/Binary

1.2 Distribution of the variables and statistics

In this section, we explore the variables quantitatively. We analyzed each attribute singularly, by making the appropriate plot, and we will show the plots of variables that we consider more relevant for our analysis.

Class: Firstly we decided to visualize the variable *class* which is a discrete-binary data (Figure 1.1). We create a bar-chart that shows the following result:

- "Hazardous state" (class 1) : 170 (6.6%)
- "Non-hazardous state" (class 0): 2414 (93.4%)

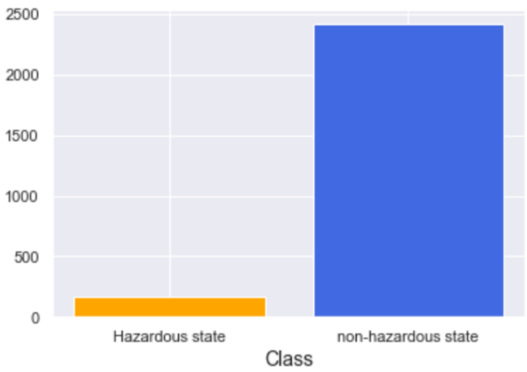


Figure 1.1: Class Bar-chart

Numerical variables:

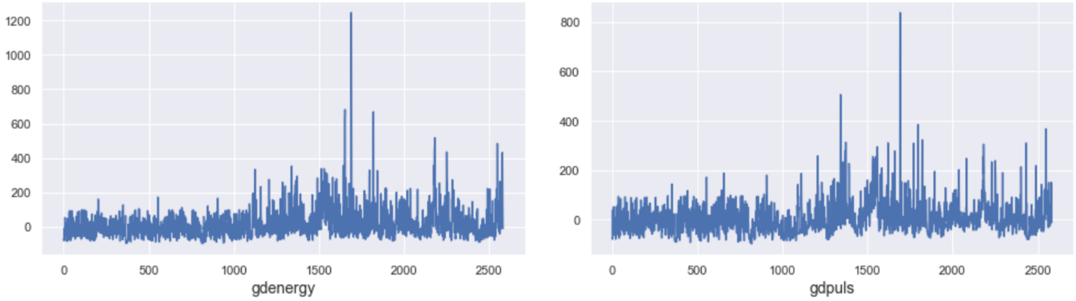
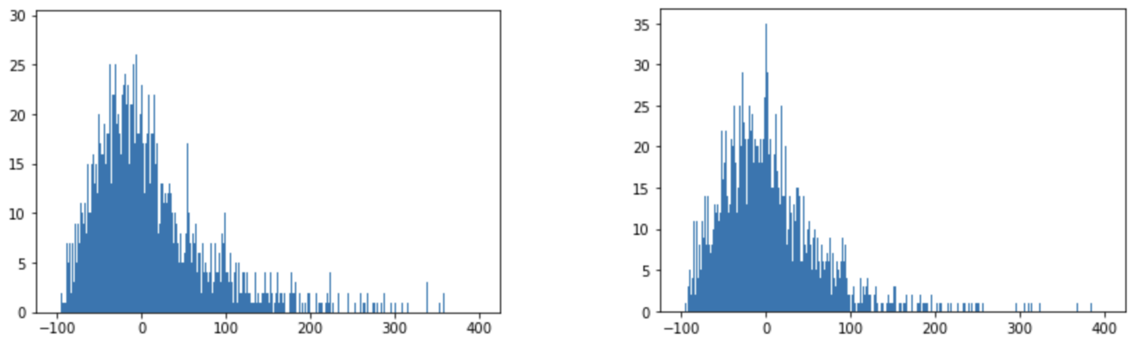


Figure 1.2: Time series of gdpuls and gdenergy

We created plots for the numerical variables in time series (see example in Figure 1.2). We can see similarities in the pair of plots of *gdenergy, gdpuls* and *maxenergy, energy* in the range 0-500 in the plots and a flat interval around 1000. We will analyse these similarities and eventual correlations in the section 1.5 of the report.

We created histograms for the numerical variables *gdenergy* and *gdpuls* in order to visualize the distribution which considers the energy recorded within previous 8 shifts by GMax (Figure 1.3).



(a) gdenergy: mean=12.3, median=-6

(b) gdpuls: mean=4.5, median=-6

Figure 1.3: histograms

Categorical variables: Looking at the bar-charts of categorical variables in Figure 1.4, we can see that the majority of hazard states enter in the category of lack of hazard, danger state is almost never present in the plots of *seismic, seismoacoustic, ghazard*. *Seismoacustic* is the only one that shows a relevant amount of "high hazard" states. While in the plot *shift* we see that there are more preparation shifts.

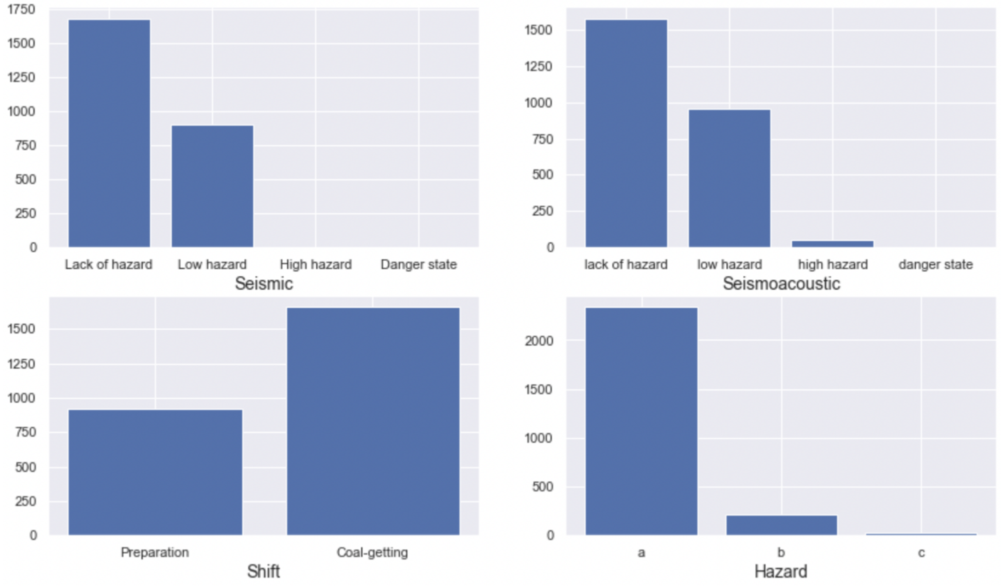
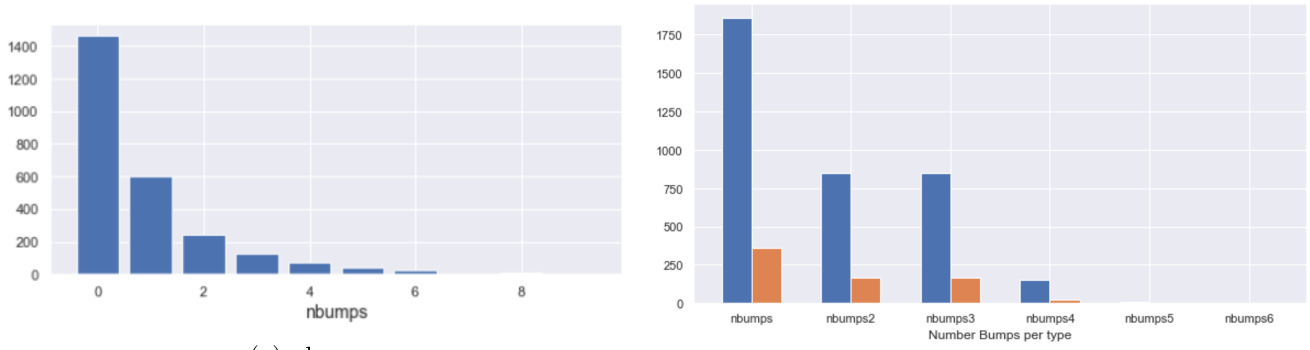


Figure 1.4: Categorical variables Bar-charts

Discrete variables: In order to visualize the discrete variables we created bar-charts for every *nbumps* column. In the plots we see that these data represents an exponential distribution.
Here, in figure 1.5a we display just the first of the nbumps column series.
In figure 1.5b we counted the total number of bumps from each intensity divided in the two classes 0 and 1. It is important to notice that a significant part of the bumps happens in shifts of class 1, despite they are only 170/2584 of the total. The ratio between the number of bumps of class 0 and 1, for each type of bumps is more less constant ∼5% .



(a) nbumps

(b) bumps vs class

Figure 1.5: Discrete variables Bar-charts

Contingency tables

We compiled contingency tables in order to understand if there is a relationship bias between the categorical features and the label categories.

In the tables below (Figure 1.6) we can see that the variables *seismic* and *shift* contain some distribution bias. The category 'b' in *seismic* feature contains a greater fraction of hazardous seismic bumps while that of *shift*'s category "W" contains more seismic bumps than the category "N".

class	0	1
seismic		
a	1599	83
b	815	87

(a) Table class-seismic

class	0	1
shift		
N	904	17
W	1510	153

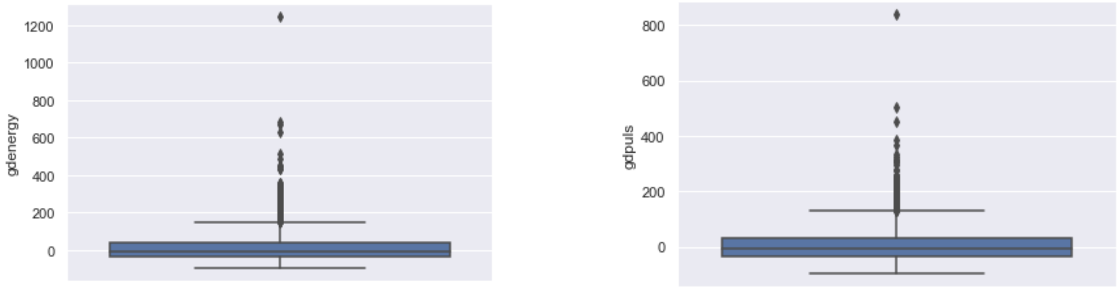
(b) Table class-shift

Figure 1.6: Contingency Tables

1.3 Assessing data quality

In the following section we will analyze the quality of the data in our dataset. In order to check if there are any missing values we analyzed the dataset with the pandas dataframe.info method. The result showed no missing values. We noticed that the columns *nbumps6, nbumps7* and *nbumps89* have no useful value for our analysis, therefore we remove them.

The outliers, which are points extremely far from the concentration of the values, were detected through the use of a BoxPlot, as shown in Figure 1.7 for **gdenergy** and **gdpuls**. We chose those attributes to show how extreme these outliers are and that they could influence our further analysis.



(a) gdenergy

(b) gdpuls

Figure 1.7: BoxPlot

1.4 Variable transformations

We have analyzed all the attributes of the dataset, except for those impractical in their scale. Due to a large range in order of magnitude of values being present in the data, we use a logarithmic scale in the attributes *genergy*, *gpuls*, *energy* and *maxenergy* in order to clearly present the data.
In order to create the correlation maps, we transformed the variables of some categorical attributes,(*seismic*, *seismoacustic*, *hazard* and *shift*) from strings to integers: $a, b, c, d \longrightarrow 0, 1, 2, 3$ and $W, N \longrightarrow 0, 1$.

1.5 Pairwise correlations and eventual elimination of variables

In this section, we evaluated the correlations between variables. In order to do so we used Pearson and Spearman methods. We decided that Spearman correlation (Figure 1.8) is the best method for our data because it considers the time series of the recordings, moreover the values are far from the mean. Spearman correlation is less sensitive than Pearson correlation to strong outliers.

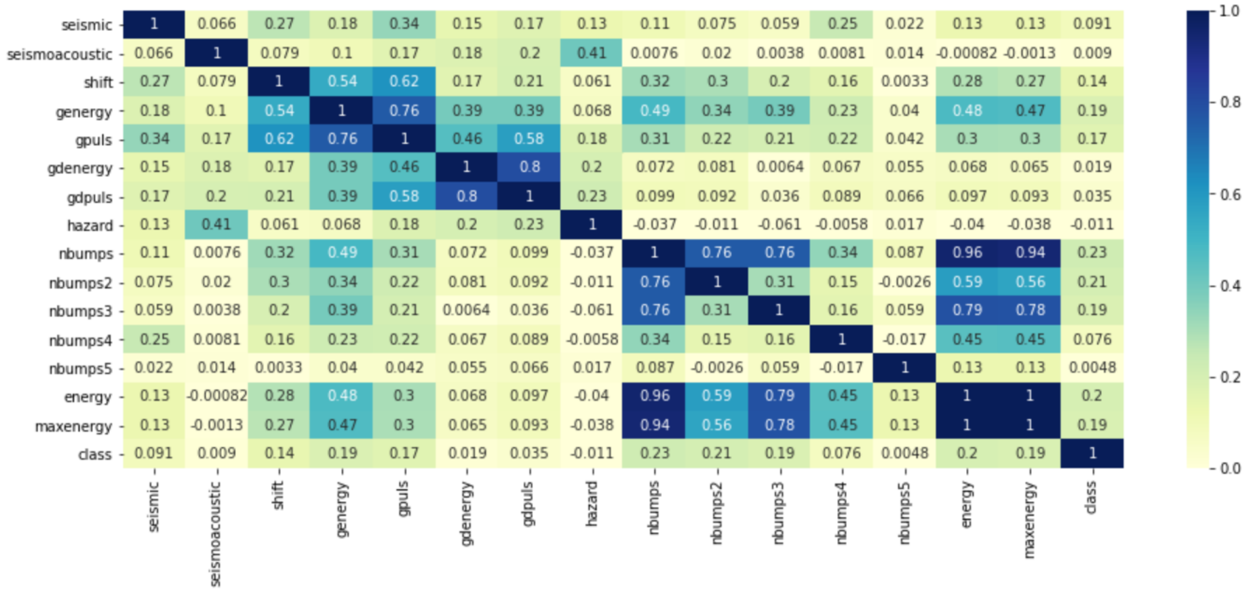


Figure 1.8: Spearman Heatmap

Perfect correlation (0,96 - 1)	energy	maxenergy	nbumps	
High correlation	nbumps	nbumps2	nbumps3	
	gdenergy	gdpuls	gpuls	
	genergy	gpuls	shift	
Other correlations	seismoacustic	hazard		
	shift →	genergy	gpuls	
	seismic →	shift	gpuls	
	genergy →	energy	gdpuls	gdenergy

Table 1.2: Spearman correlation

Correlations with class:
As shown in the Spearman correlation index, the attribute class does not have a strong correlations with the other attributes. The highest correlation is only 0.23 with attribute *nbumps*.

After analyzing the correlations between variables, we concluded that *energy* and *maxenergy* have a strong linear relationship, and have almost a perfect correlation, therefore we will drop the variable *energy*. *Seismoacustic, gdenergy, gdpuls, hazard* are irrelevant because not correlated to class. From now on, we will consider only *nbumps* instead of the other *nbumps(i)* attributes, in case of further analysis related to the number of seismic bumps.

The last step of the data understanding and pre-processing includes the *normalisation* of the values which is fundamental for the clustering. The dataset we are analyzing contains attributes that have different ranges, and extreme outliers (as shown previously in Figure 1.7).

Therefore, we concluded that the best option is to normalize values with *Robust scaler*. Robust scaling is a normalization technique that ignores outliers and calculates the median, the first and third quartiles. The median of the values then is subtracted and divided by the interquartile range which is the difference between the third and first quartile. After the normalization the outliers are present with the same relative relationships to other values.

2. Clustering

Generally all clustering methods have the same goal, which is calculating similarities and then use that to cluster the data into groups. But there are slight differences between the algorithms. In our analysis we used two clustering analysis: clustering analysis by k-means and by density based clustering (DBSCAN).
Cluster analysis requires to use a selected group of variables as described above, in our analysis we chose to use the numerical variables that have a relevant correlation with the class.

2.1 Clustering analysis by K-Means

The clustering analysis by k-means is a technique that produces its final clustering based on the number of clusters defined by the user (which is represented by the variable K). When the value of K is defined, consequently, the number of centroids is decided. K-means then allocates the data point to the closest centroid, creating in this way a cluster. After this point, it recalculates the position of centroids by averaging the data points of the centroid's cluster. Now we will explain how we used this algorithm with our data.

The next step we made was transform all the distribution of the remaining attributes with the logarithmic scale. (We made a comparison between the plot with and without this scale.What we have without the logarithm transformation was a flattened plot.)
Therefore, in our analysis the best attributes for K-means were: *gpuls*, *genergy* and *maxenergy*. Using these attributes we obtained the best combination of SSE and silhouette score. We selected $K = 5$ for our analysis, according to the elbow method (Figure 2.1).

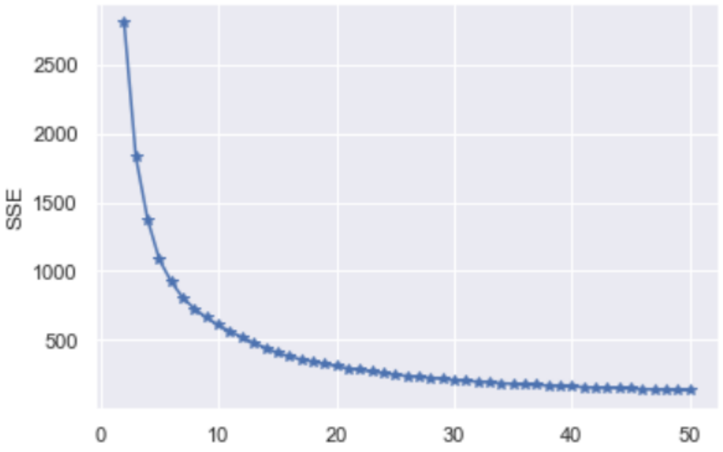
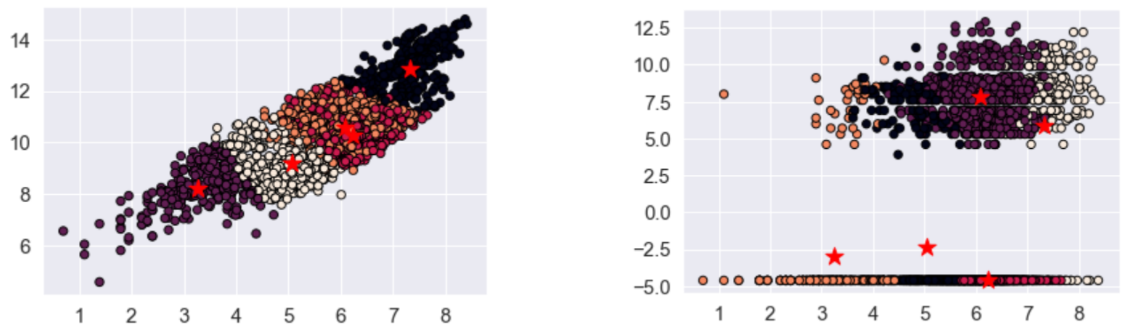


Figure 2.1: SSE for K in range (2, 52)

Therefore, we computed the K-means with $k = 5$ for the three attributes selected. We obtained a Sum of Squared Error (SSE) = 1081 and a Silhouette score of 0.39. In order to better visualize the distributions and the just formed clusters and centroids (figure 2.2), we plotted the scatterplots.

We created a parallel plot of the clusters' centroids too (Figure 2.3), in order to visualize the result in another way. This plot allows us to see what attributes contributed to differentiate the clusters



(a) kmeans: gpuls vs genergy

(b) kmeans: gpuls vs maxenergy

Figure 2.2: Scatterplots

the most.

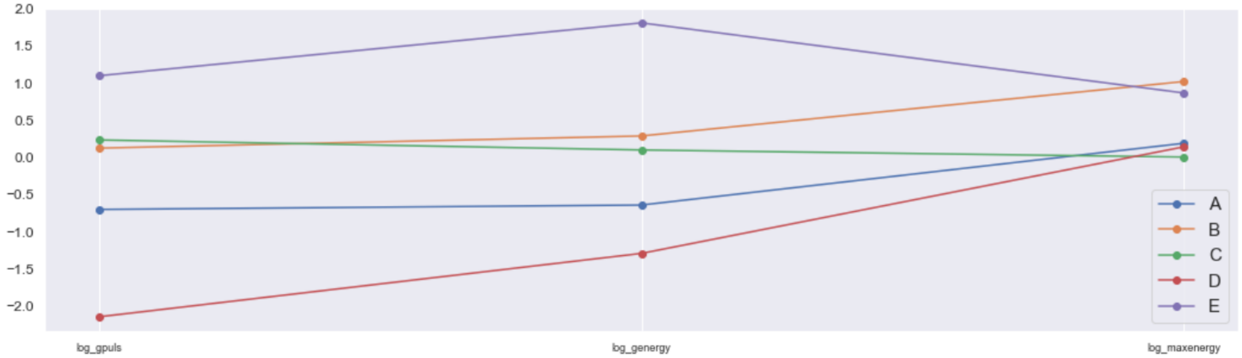


Figure 2.3: Parallel plot: "log_gpuls" ,"log_genergy" ,"log_maxenergy"

At this point, we created a bar-plot of each cluster (with normalized values) (Figure 2.4) in order to show data points from class "0" and "1" and identify which cluster represents clearly the "hazardous state".

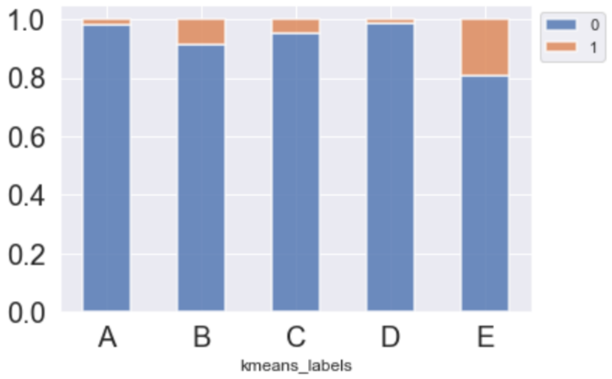


Figure 2.4: kmeans clusters with data points divided into –'0' 'non-hazardous state'(blue) –'1' 'hazardous state'(orange)

2.2 Analysis by density-based clustering

In this section we will analyse our data with the DBSCAN algorithm using the same attributes, used in the previous clustering analysis. In the density-based clustering we do not have to specify the number of clusters to use. This algorithm uses two parameters: the minimum number of points clustered together(*min_ samples*), and epsilon (ε) which is the distance measure we use to locate the points. The algorithm starts by picking up a point in the data set (and visits all the points), it considers parts of the same cluster the minimum points within the radius of ε. Then it repeats the calculations.

We will discuss the results obtained by applying the DBSCAN algorithm on the same set of attributes used for the KMeans.

First of all, we have performed several tests and attempts on ε and *min_ samples* in order to identify the best configuration of parameters, also with the help of *elbow method* (Figure 2.5).

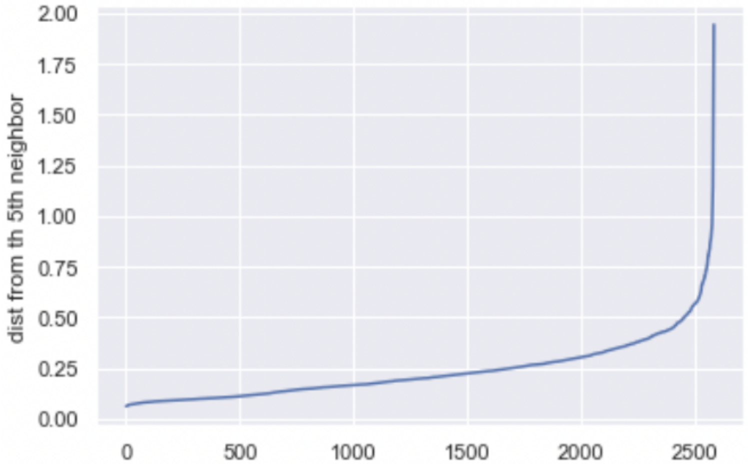
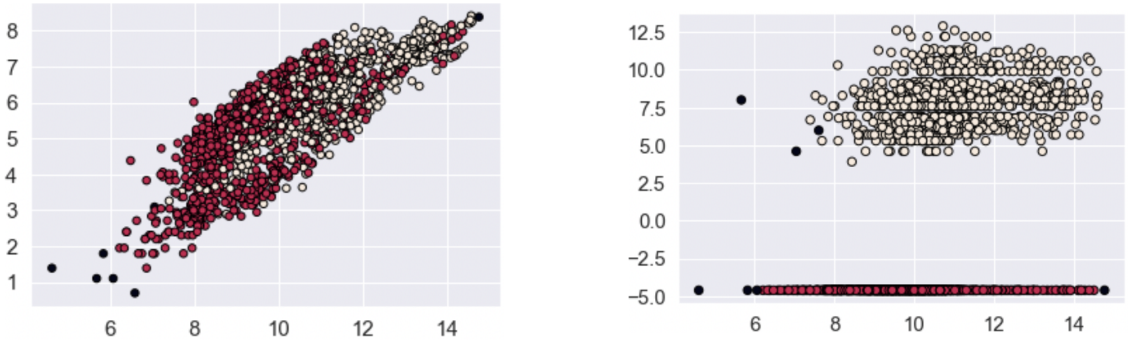


Figure 2.5: Elbow method for ε

The best combination of values we achieved are $\varepsilon = 0.66$ and *min_ samples* $= 24$, obtaining a Silhouette value ≈ 0.27.



(a) gpuls vs genergy

(b) genergy vs maxenergy

Figure 2.6: Scatter plot of DB clusters with ε = *0.66* and *min_ samples* = *24*

Even if we could have got a higher silhouette by increasing the parameter values, we decided to keep it lower silhouette. That is because with a higher silhouette we would get only one cluster and less outliers which is not the expected output.

At this point, as for the K-means analysis, we created a bar-plot of each cluster, with normalized values (see Figure 2.7) in order to show data points from class "0" and "1" and identify which cluster represents clearly the "hazardous state".

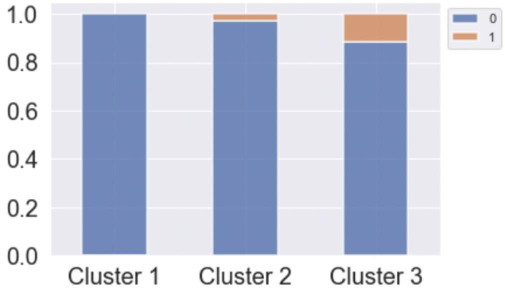


Figure 2.7: DBSCAN clusters with data points divided into –'0' 'non-hazardous state'(blue) –'1' 'hazardous state'(orange)

2.3 Analysis by hierarchical clustering

We analyzed the data set with four hierarchical clustering methods: centroid, single, Max and average. The method *Average* resulted to be the most fitting for our case, because it is robust to noise and outliers but biased towards globular clusters (look Figure 2.8). The method Max is not adequate because it tends to divide big clusters, and we have two classes of which class "0" is the major one.

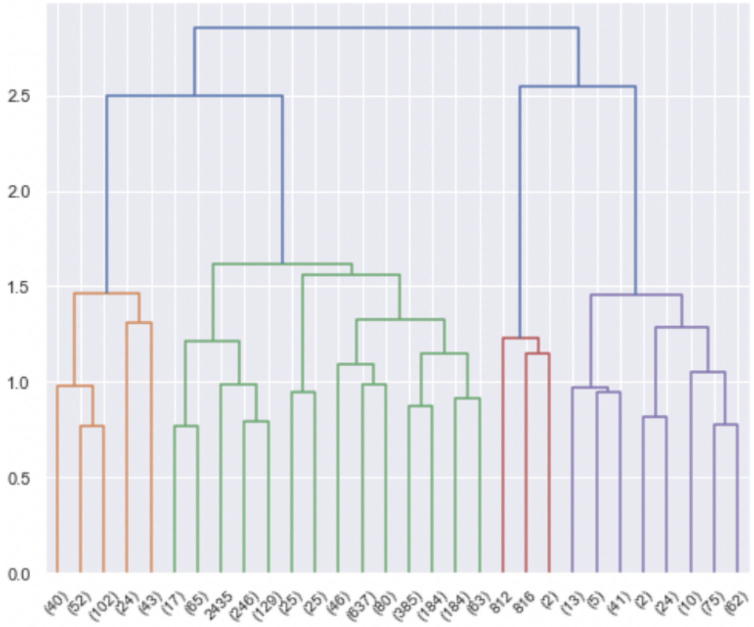


Figure 2.8: Dendogram of the dataset using *AVERAGE* method

Observing the Figure 2.8, we can see that the algorithm assembled the data in four clusters, but the red one is negligible because is composed by just four elements.

Then, we created a bar-plot of each cluster, with normalized values (see Figure 2.9) in order to show data points from class "0" and "1" and identify which cluster represents clearly the "hazardous state". We obtained a silhouette score is 0.373.

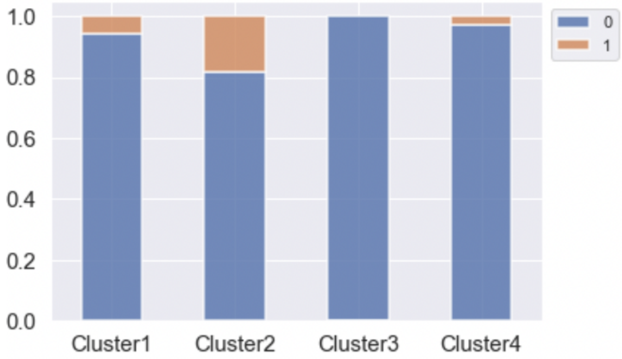


Figure 2.9: hierarchical clusters with data points divided into –'0' 'non-hazardous state'(blue) –'1' 'hazardous state'(orange)

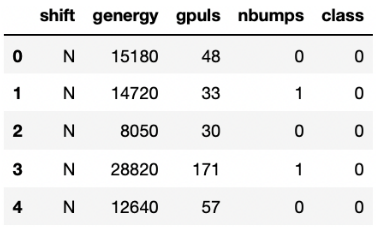
2.4 Final discussion

We analyzed our data set with three types of clustering methods: clustering by K-Means, DBSCAN and by hierarchical clustering.
The clustering by K-Means was not able to cluster the class efficiently. The results obtained by the DBSCAN method were better than the clustering by K-means, but still not optimal.
Finally, we analyzed the data sets with four hierarchical methods which are: centroid, completed, single and average. The method average was the most fitting for our case, because it is robust to noise and outliers.

In conclusion, none of the clustering methods managed to efficiently cluster the class, DBSCAN and hierarchical worked better than k-means, the hierarchical method seems to fit better since it has a better silhouette score.

3. Classification

Choice of attributes:
We chose the attributes shown in Figure 3.1 because they were the most relevant during the phase of *Data Understanding and Preparation.*



	shift	genergy	gpuls	nbumps	class
0	N	15180	48	0	0
1	N	14720	33	1	0
2	N	8050	30	0	0
3	N	28820	171	1	0
4	N	12640	57	0	0

Figure 3.1: Attributes chosen for the classification tasks

Numerical: ‘gpuls’, ‘genergy’
Discrete: ‘nbumps’
Categorical: ‘shift’

3.1 Classification by Decision Tree

The decision tree algorithm work differently with the two type of quantitative and categorical variables. Usually, it is better to not consider categorical variables when using Decision Tree algorithm implemented in Scikit-learn library. However, we realized that the attribute *shift* is correlated to the numerical attributes, therefore it is important to keep it.

3.1.1 Applying classification

We split the data set into training-set and test-set using *HOLD-OUT* method. We use *stratify* parameter so that each set contains approximately the same percentage of samples of each target class as the complete data set.

- Development set: it is 80% of the whole data set and it will be further divided into training set (75%) and validation set (25%) for the algorithm training and validation

- Test set: it will be used to determine the accuracy of the model and it is 20% if the data set.

Since our data set is unbalanced, we applied the the *RandomOverSampler* technique in order to supplement the training data set with copies of some of the minority classes which in our case is class 1. The result is a more balanced training set. Next step is applying the classification algorithm.

3.1.2 Identify the best parameter configuration

We chose as a gain criterion the *GINI index* which measures the inequality among values of a variable, the degree of probability of a particular variable being wrongly classified when it is randomly chosen. We used *gridsearch* algorithm to evaluate the best parameters for node splitting we obtained: ’min_samples_split’: 10, ’min_samples_leaf’: 5.

We realized that with this procedure, the decision tree model would be overfitted, as the performance on the test data was very bad. Therefore, the model would be too close or linked to a particular set of data, then, it may fail to fit additional data or predict future observations in a reliable way.

At this point we opted for another way to find the optimal configuration of the parameters, that is to take a metric for evaluating the performance of a classifier, such as the Precision-Recall Curve and see how the area under this graph varies, as the number of nodes in the model construction increases. We do this for both the training set and the validation set, in order to choose the value that maximizes the area for both. (look at Figure 3.2).

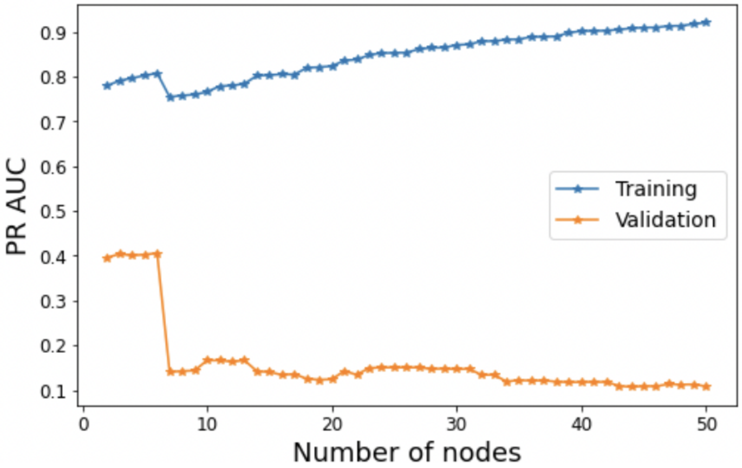


Figure 3.2: Plot used to find which max_tree_nodes works best

From the plot we understand that the optimal peak of nodes is 6. After that, the value of AUC of the training set improves while the one of the validation set declines, leading again to the phenomenon of Overfitting.

Below are the values of the feature space importance used for the construction of the model according to the GAIN of the impurity measure used (in our case the GINI index):

1. nbumps → 0.605
2. gpuls → 0.229
3. shift → 0.086
4. genergy → 0.080

3.1.3 Decision tree

The test condition chosen in the decision tree depends on shift and continuous attributes, each node is divided in a 2-way split (look at Figure 3.3). In order to find the best split we measure the node impurity with the *GINI index.*

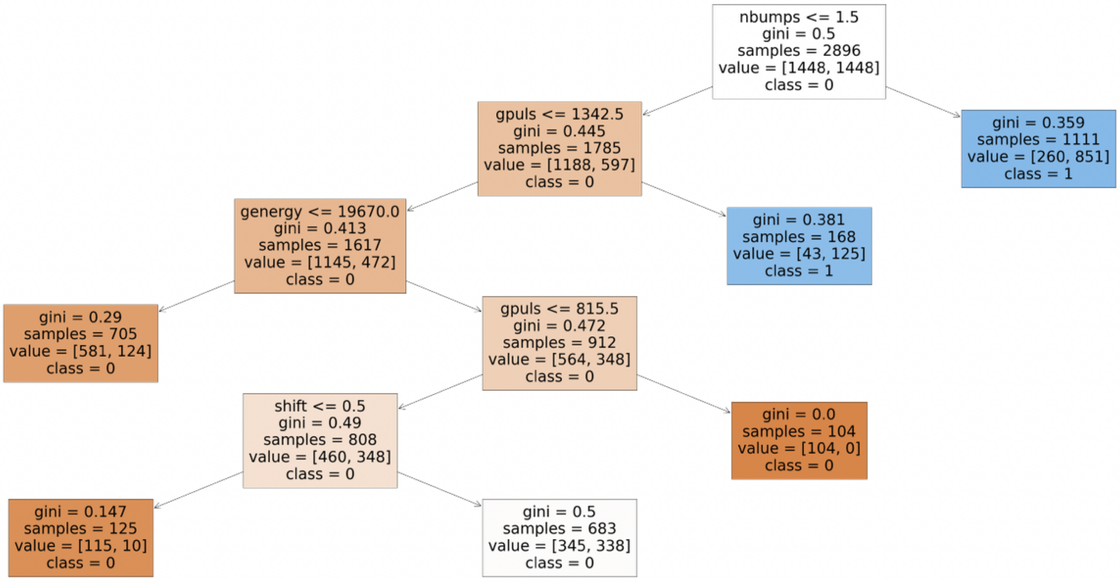
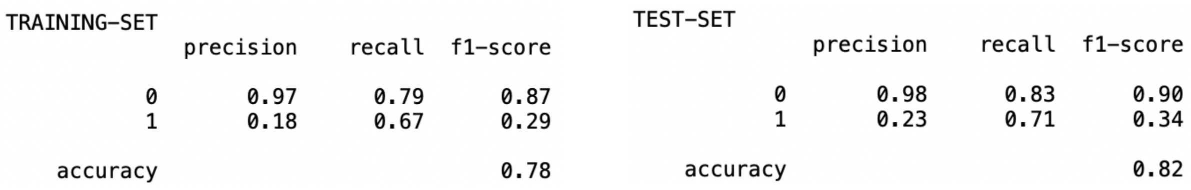


Figure 3.3: decision tree

3.1.4 Performance evaluation

In this section, we are going to compare the accuracy of the results between the training set and test set. We apply the decision tree model to the training set and test set. Then, we have to check the *Precision, Accuracy, F1, recall* values to evaluate the quality of the prediction (look Figure 3.4).



```
TRAINING-SET                                         TEST-SET
              precision    recall  f1-score                        precision    recall  f1-score

           0       0.97      0.79      0.87                   0       0.98      0.83      0.90
           1       0.18      0.67      0.29                   1       0.23      0.71      0.34

    accuracy                           0.78             accuracy                          0.82
```

Figure 3.4: Indices values

- **Accuracy**: $\frac{TP+TN}{TP+TN+FP+FN}$ [1] It is a metric for evaluating how a model is predicting well.
- **Precision**:$\frac{TP}{TP+FP}$ The ability of a classification model to identify only the relevant data points. Otherwise, quantifies the number of positive class predictions that actually belong to the positive class.
- **Recall**: $\frac{TP}{TP+FN}$ The ability of a model to find all the relevant cases within a data set. Otherwise, quantifies the number of positive class predictions made out of all positive examples in the dataset.
- **F1 score**: $\frac{2TP}{2TP+FN+FP}$ It provides a single score that balances both the concerns of precision and recall in one number.

[1] where TP, TN, FP and FN are in order: True Positive, True Negative, False Positive and False Negative

As a performance measure, accuracy is inappropriate for imbalanced classification problems. The main reason is that the number of examples from class "0" overwhelms the number of examples in the class "1", meaning that even simple models can achieve very high accuracy scores depending on how severe the class imbalance happens to be.

We are analyzing a seismic bumps data set of a coal mine. In this context, false negative values would mean that an hazardous state would not be predicted correctly, therefore, it is very dangerous and the miners could die. We want to prevent that, so we need a high recall.

3.1.5 Confusion matrix

We computed the confusion matrix:

		predicted class	
		Positive	Negative
actual class	Positive	68	34
	Negative	303	1145

	predicted class	
	Positive	Negative
Positive	24	10
Negative	82	401

Table 3.1: Confusion matrix on training (*left*) and test set (*right*)

3.1.6 Precision-recall curve

The Precision-Recall curve is more informative then the ROC curve when analyzing imbalanced data-sets as ours. The PR curve is composed by the recall or true positive rate on the x-axis and the precision on the y-axis (Figure 3.5).

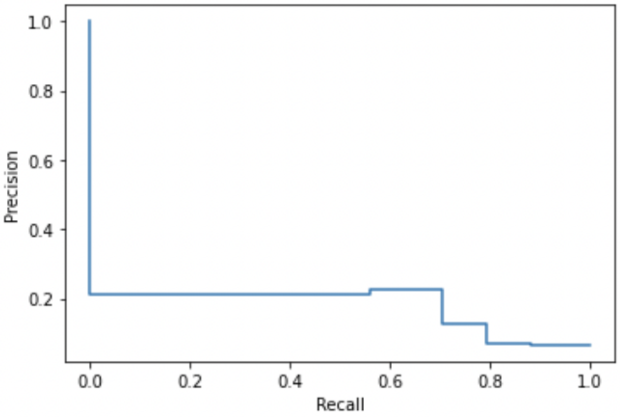
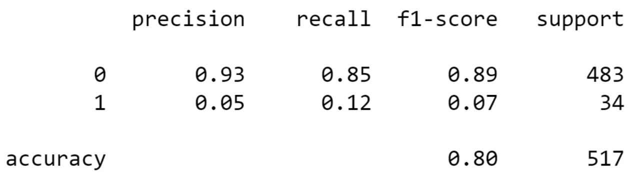


Figure 3.5: Precision-Recall Curve

The precision-recall curve also confirms what was said above. As we expected, we have the precision that is always low, but the positive note is that as the threshold decreases and the recall increases, the precision remains constant up to a certain point. We even have a rise before degenerating completely. Area Under Curve (AUC) obtained is around 0.404.

3.2 Classification by K-NN

In this section, we analyze training set and test set with K-Nearest Neighbour algorithm. Again, the dataset is partitioned using the HOLD-OUT method, so we move directly to model analysis.



```
              precision    recall  f1-score   support

           0       0.93      0.85      0.89       483
           1       0.05      0.12      0.07        34

    accuracy                           0.80       517
```

Figure 3.6: Performance evaluation

The K-NN model performance is way worse than the previous decision tree model as shown in Figure 3.6. Both the precision and recall metrics are drastically different for the positive and the negative class.

3.2.1 Confusion matrix

For the confusion matrix the number of False Negative and False Positive predictions are extremely high and the number of True Positive are almost 0.

		predicted class	
		Positive	Negative
actual class	Positive	4	30
	Negative	74	409

3.2.2 Precision Recall curve

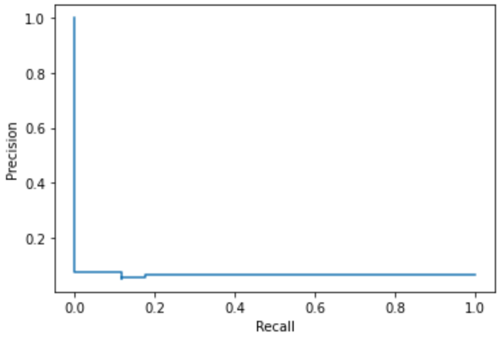


Figure 3.7: PR curve AUC = 0.094

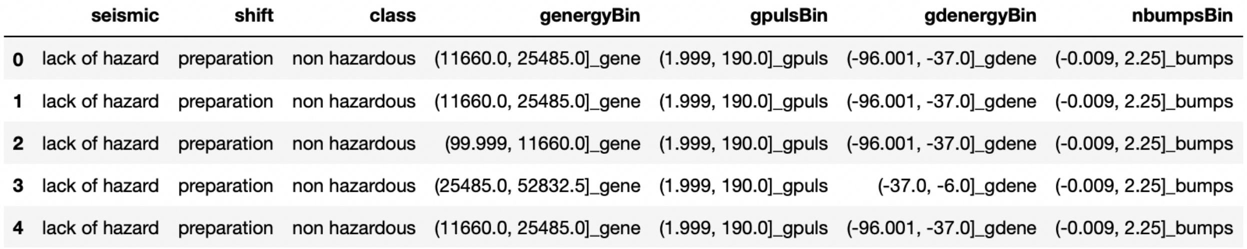
3.3 Final discussion

We classified our data with the methods: the decision tree and the K-NN. On one hand, the output obtained by the decision tree is positive, the model had an overall good performance, even though the precision-recall curve does not have an high AUC.
On the other hand, the classification by K-NN did not perform as good as the first one. We can understand this by comparing the precision-recall curves and the AUC of the two models.
In fact, the decision tree's precision-recall curve fully subtends the K-NN one, proving to be the most efficient.

4. Pattern Mining

The aim of pattern mining is to find the most frequent item sets, then generate associated rules, and extract the ones with greater confidence. In order to do the Pattern mining tasks, we have make to make a light feature pre-processing.

Pattern Mining Pre-processing It consists in the discretization of the continuous variables which in our case are: *genergy*, *gpuls*, *gdenergy*, *nbumps*. The first three variables were divided into bins with a quantile-based discretization function, while the variable *nbumps* was discretized into equal sized bins of intervals. The categorical attributes, (*seismic, shift*) were simply renamed (look Figure 4.1). We also changed the binary and numerical value of the variable "Class" into the "hazardous" and "Non-hazardous" for a better comprehension.

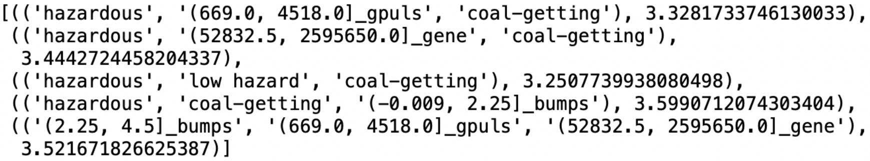


	seismic	shift	class	genergyBin	gpulsBin	gdenergyBin	nbumpsBin
0	lack of hazard	preparation	non hazardous	(11660.0, 25485.0]_gene	(1.999, 190.0]_gpuls	(-96.001, -37.0]_gdene	(-0.009, 2.25]_bumps
1	lack of hazard	preparation	non hazardous	(11660.0, 25485.0]_gene	(1.999, 190.0]_gpuls	(-96.001, -37.0]_gdene	(-0.009, 2.25]_bumps
2	lack of hazard	preparation	non hazardous	(99.999, 11660.0]_gene	(1.999, 190.0]_gpuls	(-96.001, -37.0]_gdene	(-0.009, 2.25]_bumps
3	lack of hazard	preparation	non hazardous	(25485.0, 52832.5]_gene	(1.999, 190.0]_gpuls	(-37.0, -6.0]_gdene	(-0.009, 2.25]_bumps
4	lack of hazard	preparation	non hazardous	(11660.0, 25485.0]_gene	(1.999, 190.0]_gpuls	(-96.001, -37.0]_gdene	(-0.009, 2.25]_bumps

Figure 4.1: Variables chosen and transformed during pattern mining pre-processing phase

4.1 Frequent Pattern extraction

We extract all the frequent item sets from the data set with the function *Apriori*. The minimum support value specified for the frequent item sets is 0.03 and the length of the items sets is 3. We chose a low level of support because the data set analyzed is very unbalanced, therefore a high support level would not consider any frequent patterns that includes "hazardous" class.



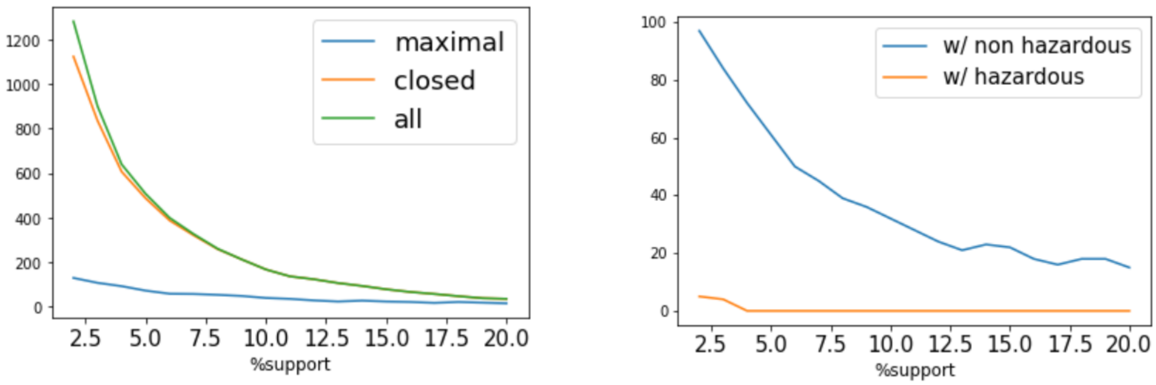
```
[(('hazardous', '(669.0, 4518.0]_gpuls', 'coal-getting'), 3.3281733746130033),
 (('hazardous', '(52832.5, 2595650.0]_gene', 'coal-getting'),
  3.4442724458204337),
 (('hazardous', 'low hazard', 'coal-getting'), 3.2507739938080498),
 (('hazardous', 'coal-getting', '(-0.009, 2.25]_bumps'), 3.5990712074303404),
 (('(2.25, 4.5]_bumps', '(669.0, 4518.0]_gpuls', '(52832.5, 2595650.0]_gene'),
  3.521671826625387)]
```

Figure 4.2: First five frequent item sets

We can understand that the most frequent item set is a hazardous seismic bump that happens after the preparation shift and the number of pulses recorded within previous shift by GMax is in the bin [669,4518]. The support of the first element is 3.328.

4.2 Discuss Frequent Pattern

We applied the *Apriori* algorithm to extract the maximal frequent item sets, using the same parameters as used previously. At this point, we can make a global overview of the characteristics in function of the support and we can compare different targets. First, we count all the frequent patterns, the number of maximal frequent patterns and the number of closed frequent patterns, in function of the support value. The result is shown in the Figure 4.3a.



(a) Maximal, closed, all

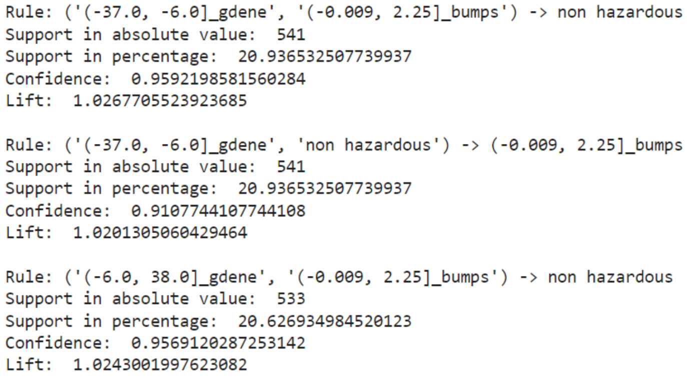
(b) Hazardous and non hazardous

Figure 4.3: Plots of number of frequent item sets

From the plot, we understand that the maximal frequent item sets is a subset of the closed ones. While the support value increases, we obtain less frequent items sets.
We analyze the number of item sets of "hazardous" and "non hazardous" in the whole data set (look Figure 4.3b).We understand that we have more "non hazardous" frequent patterns than "hazardous" ones. After a support value of 3%, we notice that the "hazardous" patterns do not appear anymore.

4.3 Association Rules extraction

We follow a similar process used in the section before, regarding the association rules. We extract the association rules, with the *Apriori* algorithm , the support value decided is 0.2, with a value of confidence of 30 (look Figure 4.4). The decision of the parameters is based on the fact that we wanted to obtain as a result association rules that would imply hazardous bumps. The support shows how actually an association rule is relevant in the whole data set.



```
Rule: ('(-37.0, -6.0]_gdene', '(-0.009, 2.25]_bumps') -> non hazardous
Support in absolute value:  541
Support in percentage:  20.936532507739937
Confidence:  0.9592198581560284
Lift:  1.0267705523923685

Rule: ('(-37.0, -6.0]_gdene', 'non hazardous') -> (-0.009, 2.25]_bumps
Support in absolute value:  541
Support in percentage:  20.936532507739937
Confidence:  0.9107744107744108
Lift:  1.0201305060429464

Rule: ('(-6.0, 38.0]_gdene', '(-0.009, 2.25]_bumps') -> non hazardous
Support in absolute value:  533
Support in percentage:  20.626934984520123
Confidence:  0.9569120287253142
Lift:  1.0243001997623082
```

Figure 4.4: First three association rules

4.4 Discuss Association rules

We used a heat-map to visualize better the number of rules in function of the support and confidence (look Figure 4.5) . We can see that with a higher confidence value we get less association rules and with the increase of support the association rules decrease.

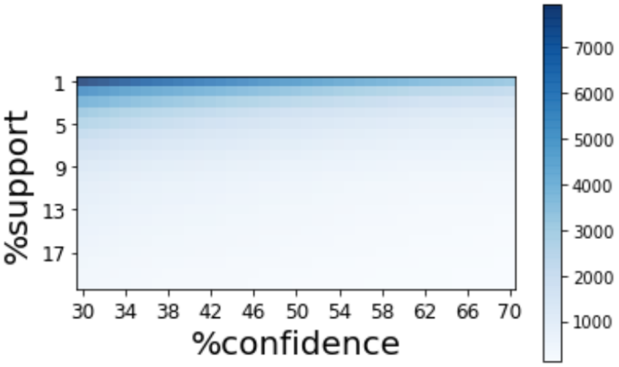


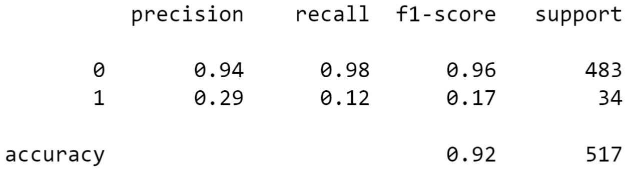
Figure 4.5: Heatmap

4.5 Exploit the most useful extracted rules

In this section, we used the association rules in order to classify the transactions in the variable *class*. The association rules selected, were the ones that implied "hazardous" in the class.
The chosen association rule is:

[3, 4]*nbumps*', '(669, 4518]*gpuls*', '(52832, 2595650]g*ene*', '*low hazard*', '*coal-getting*' → *hazardous*

This rule has been applied to the same test-set used for the Classification part. Then, we used the same method evaluation of chapter 3



```
              precision    recall  f1-score   support

           0       0.94      0.98      0.96       483
           1       0.29      0.12      0.17        34

    accuracy                           0.92       517
```

Figure 4.6: Performance Evaluation of Association Rules

Confusion matrix

		predicted class	
		Positive	Negative
actual class	Positive	4	30
	Negative	10	473

4.6 Final discussion

The results obtained from the pattern mining task is not different from what we were expecting. For example, we found frequent patterns between variables that were already analyzed earlier during the data understanding phase (Figure 1.8). It resulted difficult to find association rules containing the hazardous class, because of the nature of our data set, which is unbalanced.

The association rule found, which contains the hazardous class, is implied by parameters that we

could expect, for instance high *genergy* and *gpuls*, high number of *nbumps* and a "low hazard" in the seismic attribute.
Since we found just one "good" association rule, it is hard to find an high recall. The overall performance is not optimal, especially if compared to the decision tree classification model, in particular.

Conclusion

The Clustering and Pattern Mining techniques resulted to be weak in clustering groups or predict the target *class*, for our unbalanced data-set. The most efficient method for targeting the *class* was the Decision tree classifier. We were looking for a model that would balance the precision and the recall, in order to obtain a not excessive number of false negatives without obtaining a big number of false positives.
In conclusion, we can consider all the methods used valid for the analysis of our data set combined all together, but the most efficient one was the decision tree classifier.
We are satisfied by the results obtained, since with this basic analysis we still can extract useful information and eventually predict part of the hazardous bumps so the miners can avoid dangerous situations.