Università di Pisa

Department of Computer Science

Master program in Data Science and Business Informatics

DATA MINING REPORT

Analysis of Seismic Bumps in a coal mine

A.A 2021/2022

Authors

Alessandra Cicciari, Giorgia D'Antoni
Alessia Simone, Francesca Maria Palazzotto

Contents

Introduction

Mining activity was and is always connected with the occurrence of dangers which are commonly called *mining hazards*. A special case of such a threat is a *seismic hazard* which frequently occurs in many underground mines.
The data set we are going to analyse comes from the **UCI - Machine Learning Repository** and describes the problem of high energy seismic bumps forecasting in a coal mine located in Poland.
The purpose of this research is to determine factors that could be useful to predict the occurence of dangerous seismic hazards.

1 Data Understanding & Preparation

In this section, we will analyse the records that come from a data set from the UCI-Machine Learning Repository. This data set contains 2584 instances (rows) and 19 attributes (columns), out of which 4 are categorical features, 8 discrete features, 6 numeric features, and 1 label column. This last one represents the *decision attribute* which contains the binary value “1” for ‘hazardous state’ with high energy seismic bump occurring in the next shift and “0” for ‘non-hazardous state’ with no high energy seismic bumps occurring in the next shift.

First of all, we take a look at the feature *Class*. It is easy to see that we are dealing with an unbalanced distribution: there are only 6.58% of records (170) classified as 1 and 93.42% (2414) as 0.

Our aim is to try to predict this value based on other features and the possible correlations between them.

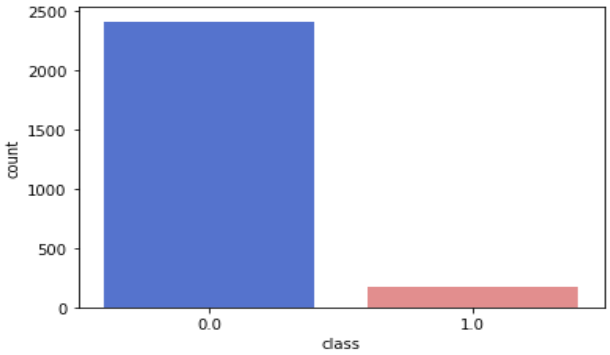

Figure 1: decision variable distribution

Since the data set, according to this attribute, is very unbalanced, we are going to examine and analyse in depth each feature. We are going to start with a brief introduction of them and then with our research.

1.1 Data Semantics

Let introduce the semantic meanings of the features presented in the complete data set:

Feature name	*Definition*	*Type*
Seismic	*Result of shift seismic hazard assessment in the mine working obtained by the seismic method*	*Categorical*
Seismacoustic	*Result of shift seismic hazard assessment in the mine working obtained by the seismoacoustic method*	*Categorical*
Shift	*Information about type of a shift*	*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 energy recorded 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*
Ghazard	*Result of shift seismic hazard assessment in the mine working obtained by the seismoacoustic method based on registration coming from GMax only*	*Categorical*
Nbumps	*The number of seismic bumps recorded within previous shift*	*Discrete numerical*
Nbumps2	*The number of seismic bumps (in energy range $[10^2, 10^3)$) registered within previous shift*	*Discrete numerical*
Nbumps3	*The number of seismic bumps (in energy range $[10^3,10^4)$) registered within previous shift*	*Discrete numerical*

Nbumps4	*The number of seismic bumps (in energy range [10^4,10^5)) registered within previous shift*	*Discrete numerical*
Nbumps5	*The number of seismic bumps (in energy range [10^5,10^6)) registered within the last shift*	*Discrete numerical*
Nbumps6	*The number of seismic bumps (in energy range [10^6,10^7)) registered within previous shift*	*Discrete numerical*
Nbumps7	*The number of seismic bumps (in energy range [10^7,10^8)) registered within previous shift*	*Discrete numerical*
Nbumps89	*The number of seismic bumps (in energy range [10^8,10^{10})) registered within previous shift*	*Discrete 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*
Class	*The decision attribute*	*Binary - label*

Regarding the categorical features, *seismic*, *seismoacoustic* and *hazard* take the values 'a', which means 'lack of hazard', 'b' which means 'low hazard', 'c' which means 'high hazard' and 'd' which means 'danger state'. *Shift,* instead, takes the values 'W', which means 'coal getting', and 'N' which means 'preparation shift'.

1.2 Distribution of the Variables and Statistics

Figure 2 shows the distribution of the categorical variables of our set: *seismic, seismacoustic, hazard* and *shift*. Using a bar chart, it is easy to see the frequencies of the values of these features.

At first, we noticed there is not the presence of a *'d - danger state'* value and there is a low probability of a *'c - high hazard*' value, evidenced by the low frequencies of it which occurs in 0% (0), 1.86% (48) and 1.16% (30) respectively for the first three features. On the other hand, the bar charts are showing the most frequent values for each attribute and it is evident that the *'a - lack of hazard'* value is the one in greater percentage assumed by our features, respectively, we see a 65.10% (1682) for *Seismic*, 61.15% (1580) for *Seismacoustic* and 90.63% (2342) for *Hazard*, in contrast of the percentage of '*b - low hazard'* value. Observing the fourth variable *Shift,* we note how the value '*W - coal getting*' (1663, 64.36%) is the most frequent.

In this first analysis, we can deduce that there have been no records that lead us to think of catastrophic situations inside the coal mine.

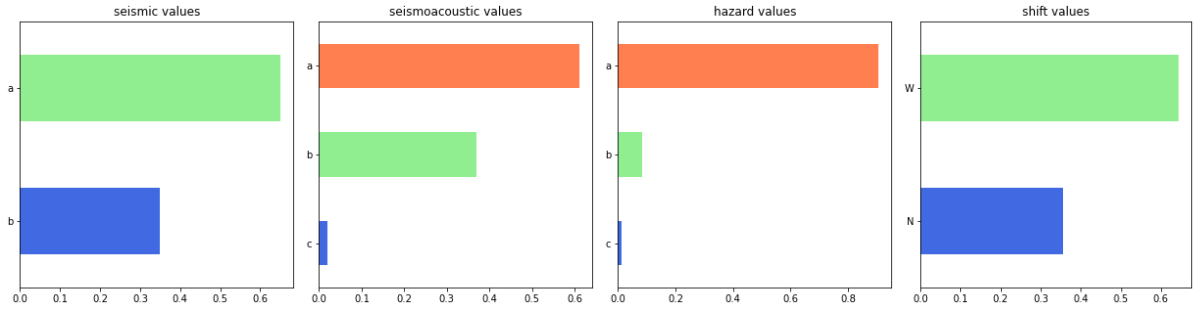

Figure 2: categorical variables distribution

For numerical continuous attributes, we realised a graphical representation that is shown in Figure 3. We used the density distribution to see how the features *genergy*, *gpuls*, *gdenergy*, *gdpuls*, *energy* and *maxenergy* are distributed in each record. As we can see, the variables assume a right skewed distribution. This means that the majority of values are located in the right part, around zero, and presenting an unbalanced distribution.

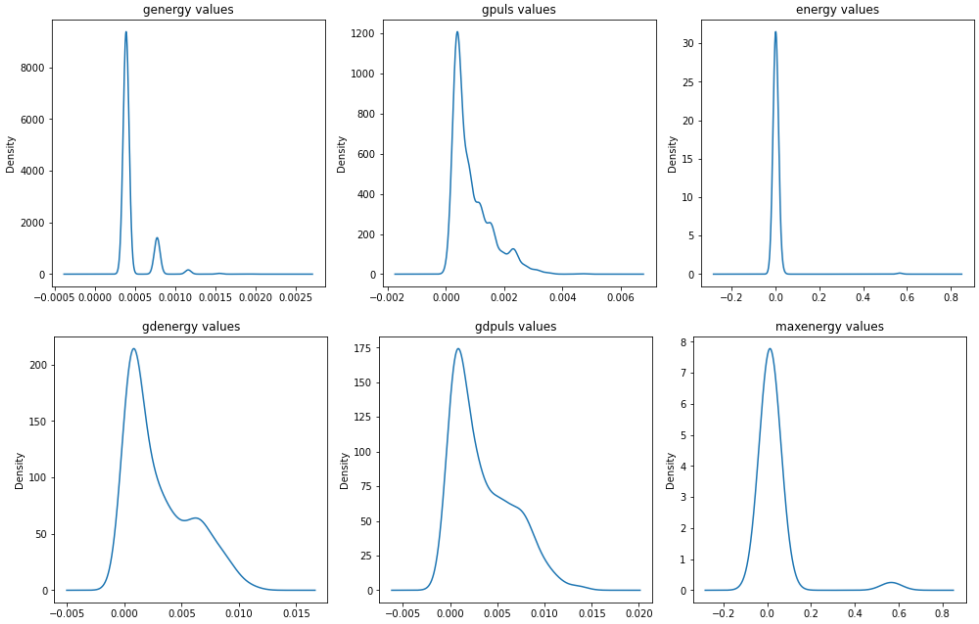

Figure 3: numerical continuous variables distribution

The attributes *nbumps, nbumps2, nbumps3, nbumps4, nbumps5, nbumps6, nbumps7* and *nbumps89* represent the numerical discrete attributes, of which distribution we studied using histograms (Figure 4). The latter were created using a number of bins equal to the one suggested by the **Sturges' rule**, that is *k =12.*

nbumps6, *nbumps7* and *nbumps89* do not contribute in any way to the classification of the hazardous state and, consequently, they represent *irrelevant features*: as we can see, they present only 0 values. Based on that, we decided to remove them from our analysis.

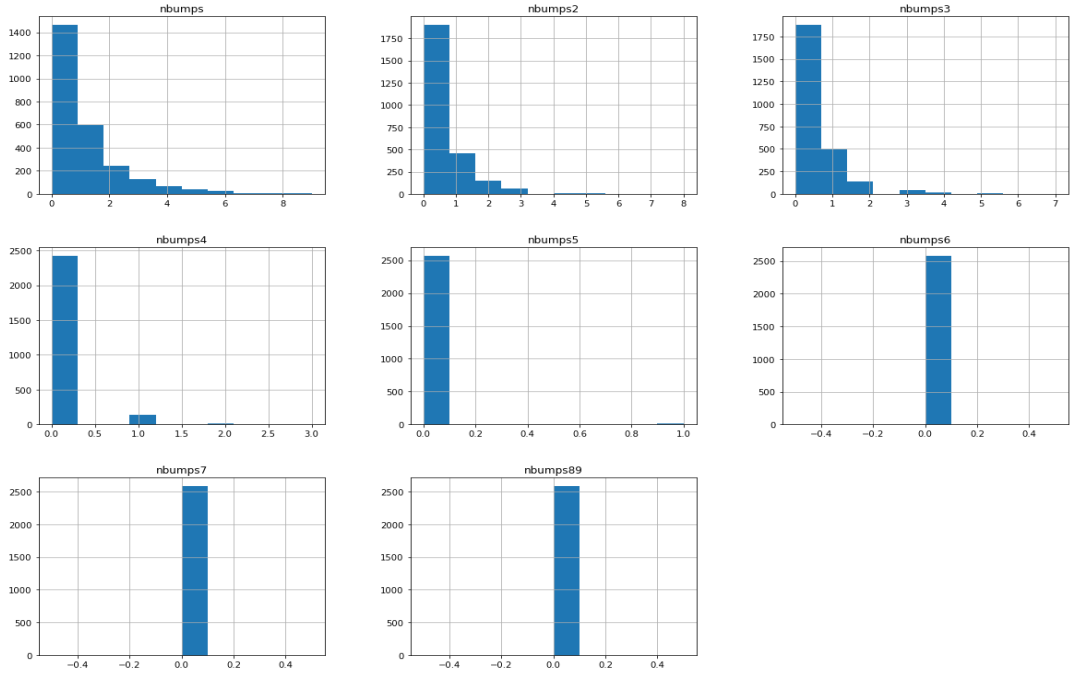

Figure 4: numerical discrete variables distribution

The remaining variables, *nbumps, nbumps2, nbumps3, nbumps4* and *nbumps5*, have an unbalanced distribution with a high frequency of values equal to zero. Furthermore, we have noticed some outliers. We will delve into this aspect during the data quality detection phase.

1.3 Assessing Data Quality

To assess the data quality test, we have first checked if there were any missing values on the complete data set and we haven't found any. Then we started to detect hidden missing values.

In order to obtain this information, we decided to remove all the rows in which the value of *nbumps* is equal to 0 and the value of *genergy* is greater than the second quartile; this because, since the seismic energy is directly proportional to the amplitude of the oscillations of seismic waves, the greater is the energy the more the number of bumps increase. According to this, we obtained 486 records.

In addition, since the features *gpuls* and *genergy* are strictly connected, as before, we decided to remove all the rows with the value of *nbumps* equal to 0 and the value of *gpuls* greater than the second quartile: here, we got 222 records. Hence, we have scaled our data set into a new one composed of a total of 1876 records.

After that, we checked for possible inconsistencies. Therefore, we determined the presence of two inconsistent rows in which the records were distorted: the total number of bumps within the previous shift did not match the sum of the number of bumps of each energy range. By dropping them, we have reduced the data set to 1874 records (rows).

In order to analyse outliers, in a first step, we have used a graphical representation by box plots both for the numerical discrete and continuous features, as we can see from Figure 5.

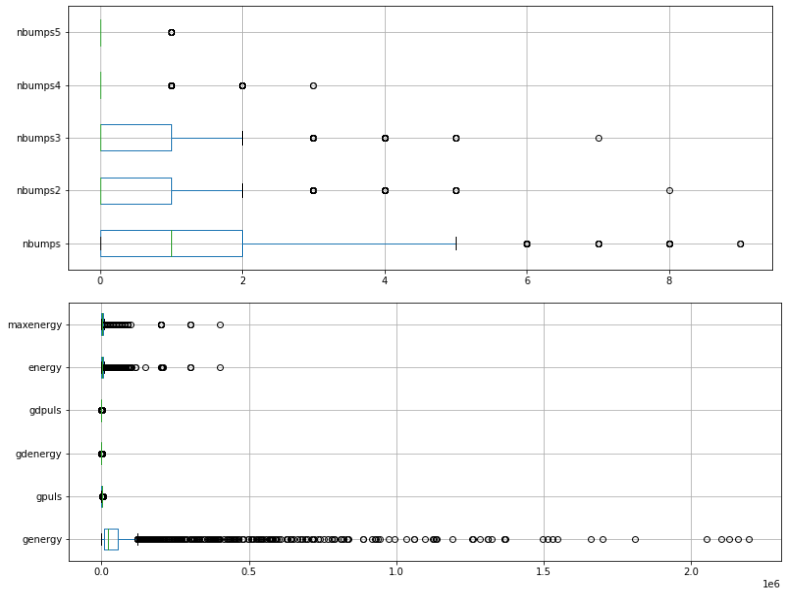

Figure 5: box plot distributions for numerical discrete and continuous variables

Then, we used the **IQR function** to determine the number of outliers in our data set: we detected 249 outliers for *genergy*, 100 outliers for *gpuls*, 70 outliers for *gdenergy*, 26 outliers for *gdpuls*, 148 outliers for *energy* and 193 outliers for *maxenergy.* So, we removed a total of 786 outliers in our records to obtain a reduction of our data set to 1088 records (rows).

At last, about the numerical discrete features, we discovered that every record not equal to 0 emerged as an outlier. So, if we would have dropped them, we would have obtained the entire data set with 0s in *nbumps* feature.

Eliminating the outliers, we checked the correlation between the variables, determining the presence of NaN values for the features *nbumps4* and *nbumps5*. We dropped them and continued our analysis using 14 features (columns).

Moving on with the analysis, we detected the entity of our categorical variables with the reduced data set. All of them are divided in the same amount of values in the two classes: 1059 (97.33%) fall in class '0' and 29 (1.83%) fall in class '1'. So, they do not give us any information about the hazard state of the coal mine.

After that, since bumps are the number of seismic bumps recorded within the previous shift, it makes sense to check their entity: in other words, in which *class* they fall ('0' = non-hazardous state, '1' = hazardous state). We checked it using crosstab (Figure 6).

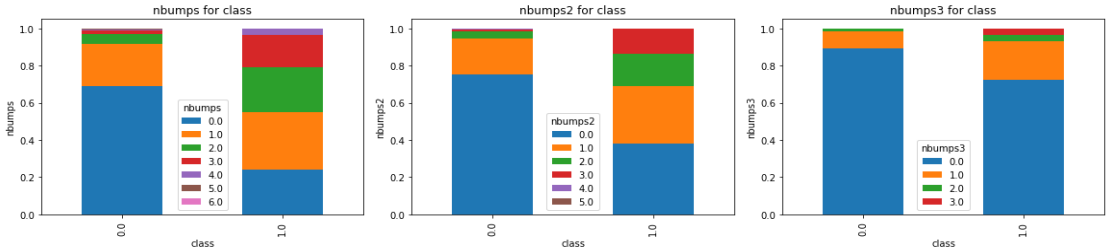

Figure 6: crosstabs

The crosstabs highlight that the majority of bumps that fall in the class "0" are equal to zero, meaning "non-hazardous state". On the other hand, class 1 is composed of various levels of bumps. Also the ones which have registered an energy intensity in the upper bound of the range, have been classified as *'non-hazardous state*', belonging to the class '0'. Moreover, they do not lead us to detect any kind of range or limit that we can consider to detect in which cases we obtain class '0' or '1'.

1.4 Variable Transformations

Due to the fact that the variables *genergy* and *gpuls* present a right skewed distribution, we transformed them using the **logarithmic transformation** to give them a gaussian shape and make the distribution more normal improving the linearity between our variables, shown by Figure 7, using histograms.

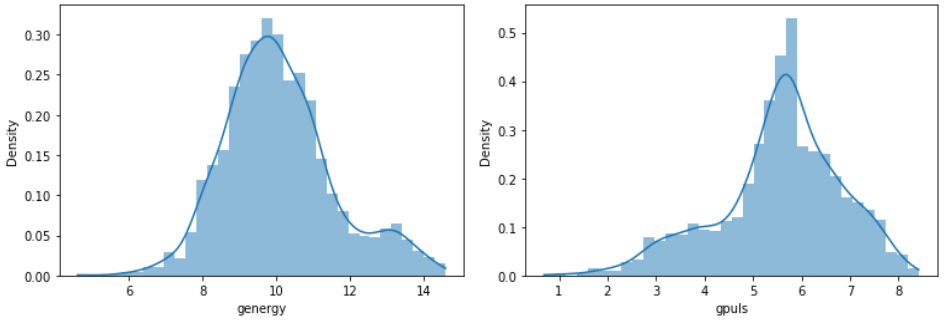

Figure 7: log-transformation hist plots

Having done this, we reanalysed the data set introducing the transformed variables *log_gpuls* and *log_genergy*.

Moreover, we transformed the variables *gdenergy* and *gdpuls* using the **z-score method**, this because they presented a lot of values equal to 0 and we couldn't use the log-transformation.

At the same time, not being able to use the log-transformation on negative values, we transformed the variables *energy* and *maxenergy* adding them the value 1 to shift them and apply the log-transformation.

1.5 Pairwise Correlations

To investigate the entity of the correlation between the variables we firstly plotted a heatmap, as shown in Figure. As it is possible to see, most of them are weakly correlated. Between few variables there is a strong correlation, this is the case for '*gpuls'* and '*genergy'* or '*energy' and 'maxenergy'.*

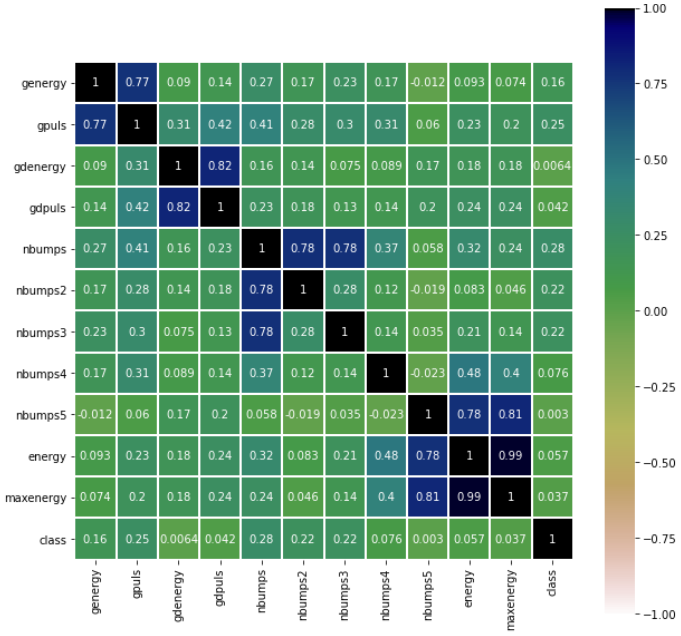

Figure 8: correlation matrix

To better visualize the correlation between these variables we plotted the scatter plots - grouping the variables by the decision variable *'class'* - firstly between *'genergy'* and *'gpuls'*. From the scatterplot between *'genergy'* and *'gpulse'*, the correlation is not clearly visible, so we repeated the plotting, using the **logarithmic transformed variables,** since the variables considered have right-tailed distribution. Thus, it is clear the positive correlation. Intuitively, when the seismic energy recorded in a shift is high, the number of recorded puls should be, in turn, high, and vice versa.

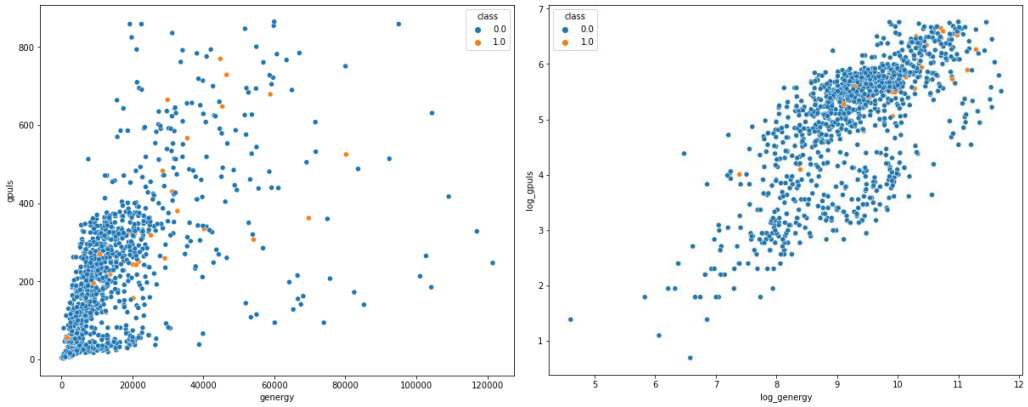

Figure 9: Right - scatterplot genergy and gpuls. Left - scatter plot log_genergy and log_gpuls

Similarly, we plotted the scatterplot between the variables *'energy'* and *'maxenergy'*, from which it is clear a strong positive correlation. On the other hand, since the two variables have several zero values, we applied a logarithmic transformation, and added a constant (equals to 1), both to energy (which began shifted_log_energy) and maxenergy (which began shifted_log_maxenergy). Indeed, the following scatter plot is definitely clearer.

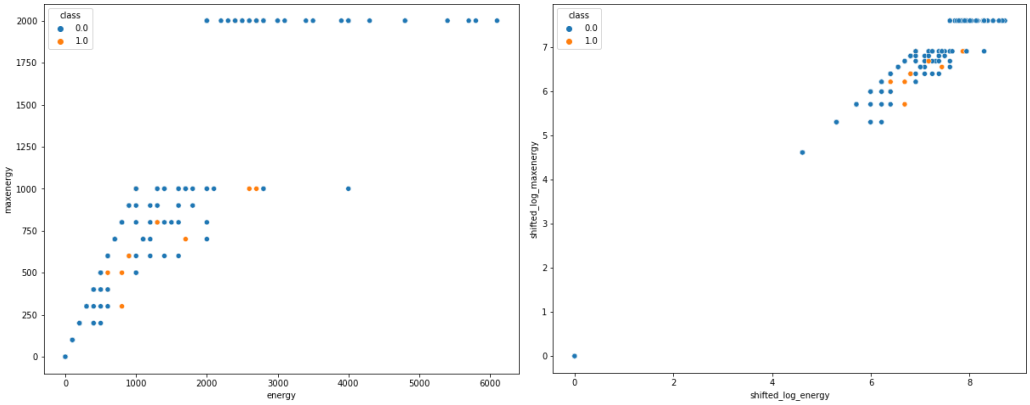

Figure 10: Right - Scatterplot energy and maxenergy. Left - Scatter plot shifted_log_energy and shifted_log_maxenergy

2 Clustering

2.1 Clustering analysis by K-Means

For the clusterization part, we used an algorithm of unsupervised learning, in this case the *Kmeans* - to find clusters in our dataset, which have not been labeled in it. Our attempt was to find groups clusterized by the target feature '*class*'. In order to find out the best number of clusters for this data set, we also observed each Silhouette value obtained increasing the number of clusters. The highest Silhouette was with the number of clusters equal to 2. Thus, the k value we provided is equal to 2.

We applied k means several times, each time giving as input different values. In particular, we provided the algorithm the columns *'genergy', 'gpuls', 'gdenergy', 'gdpuls', 'nbumps', 'nbumps2', 'nbumps3', 'energy', 'maxenergy', 'class',* once **without any logarithmic transformation** applied on them. In this case, the value of the ***Silhouette*** was equal to ***0.74*** and the algorithm seems to distinguish two clusters when the variables genergy and gpuls are compared. The bar plot of the kmeans label highlights how the clusterization does not return great results on this dataset: the label 0 is completely formed by values labeled as zero, but the label 1 is almost entirely formed by values labeled as zero. This could be caused by the imbalanced initial distribution of 'class'. As seen in the previous paragraph, the majority of variables are labeled as 0, and we have much less values labeled as class 1.

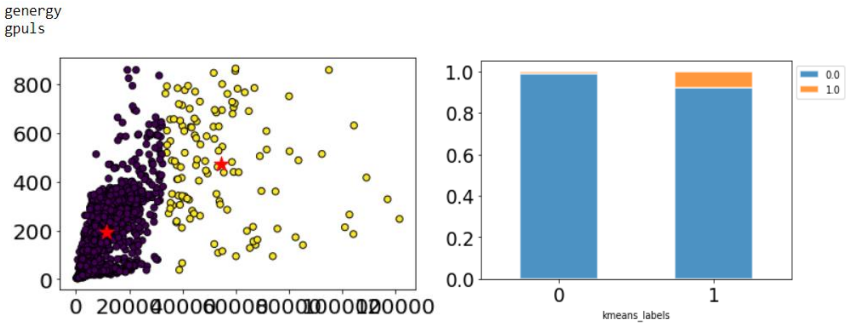

Figure 11: Right - Scatterplot genergy and gpuls with k-means, Non transformed dataset. Left - K-means labels obtained with k-means

Then, to the same columns, so to the non-transformed dataset we applied the ***MinMaxScaler*** from *sklearn.* The value of Silhouette decreases to **0.48**. The clusters on the same two variables are less distant and distinguished.

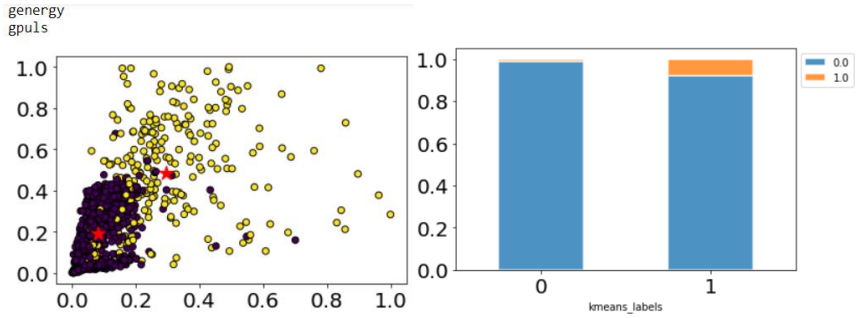

Figure 12: Right - Scatterplot genergy and gpuls with k-means,MinMaxScaler. Left - K-means labels obtained with k-means

Since most features had a skewed distribution, we applied **log transformation or z score** (where zero values appeared), and then, to scale them we applied the ***MinMaxScaler*** and ***standardard* scaler**. So we applied k-means also on **transformed and scaled data**. With the MinMaxScaler we obtained a Silhouette score of **0.61**; the labels are not better assigned, and the clusters are not better distinguished.

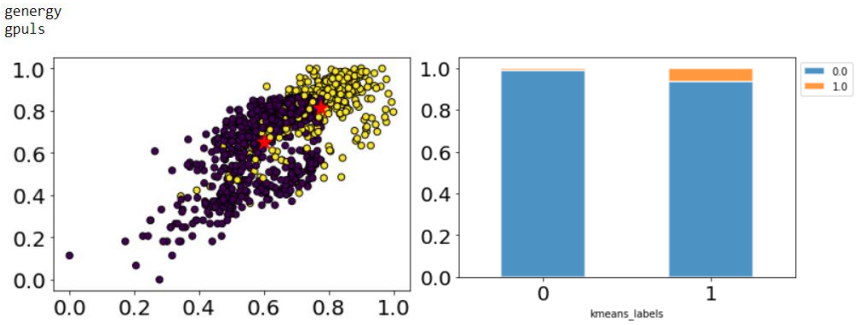

Figure 13: Right - Scatterplot genergy and gpuls with k-means, LogTransformation, z-score and MinMaxScaler. Left - K-means labels obtained with k-means

Changing the scaler method, so applying **StandardScaler**, the Silhouette decreases to **0.45**. The situation doesn't change much.

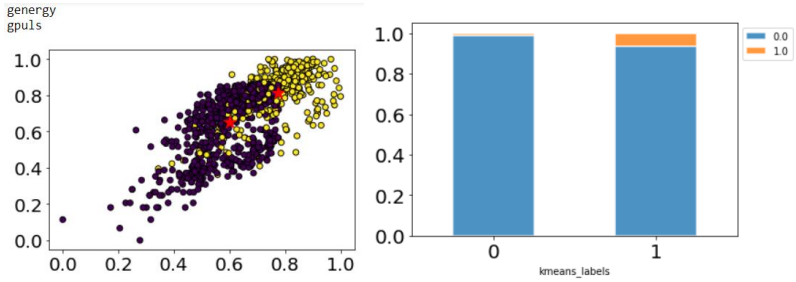

Figure 14: Right - Scatterplot genergy and gpuls with k-means, Standardcaler. Left - K-means labels obtained with k-means

It is possible to see that the higher value of Silhouette is obtained when applying k-means to the non-transformed and non-scaled dataset, also the clusters seem to be better defined.

2.2 Analysis by Density-Based clustering

For the analysis by Density-Based clustering we utilised the density based clustering algorithm **DBSCAN.** To find the optimal combination of the two hyper parameters of dbscan - **ε** (region within looking for a close point) and **minPts** (minimum number of points required to form a dense region) - we used the **grid search** and runned the algorithm providing a range of values for both hyperparameters. For eps we provided a range equal to (0.1, 2, 0.1), instead for minPts a range of (5, 50, 5). Analysing the resulting heatmap we chose the exact value of **ε** and **minPts** to give to the algorithm, and finally plotted a pairplot to see how the chosen model

performed. This procedure has been applied to different types of dataframe, so the one without the logarithmic transformation and normalization and the ones with the data transformations.

Non transformed and non scaled dataset

Based on grid search, we chose an **eps=0.3** and **MinPts=5.** On this dataset we didn't obtain a proper result, as it can be seen from the Figures below: the clusters are not detected.

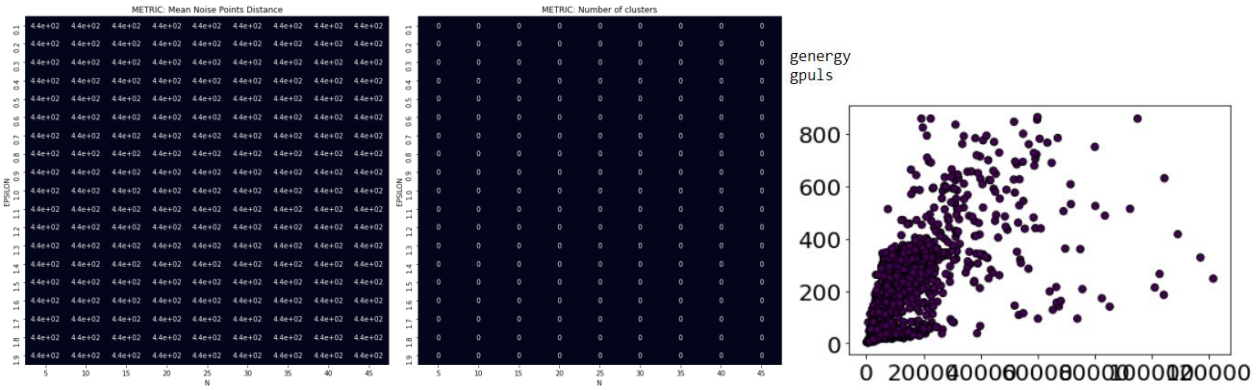

Figure 15: Right - HeatMap Grid Search Non transformed and non scaled dataset. Left - Scatterplot genergy vs gpuls

Scaled dataset (DataScaler) without transformations

Based on grid search, we chose an **ε = 1.4** and a **MinPts = 30**

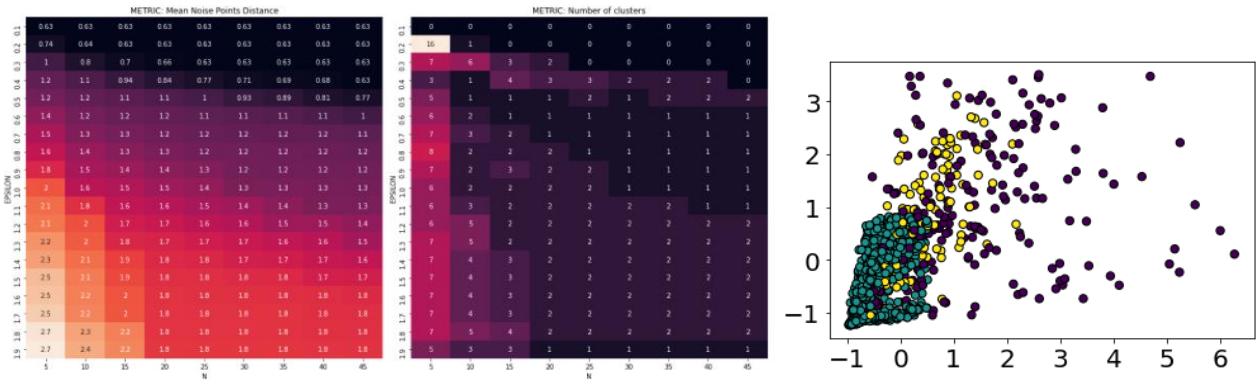

Figure 16: Right - HeatMap Grid SearchScaled dataset (DataScaler) without transformations. Left - Scatterplot genergy vs gpuls

Scaled dataset (MinMaxScaler) without transformations

ε = 0.8 and a **MinPts = 15**

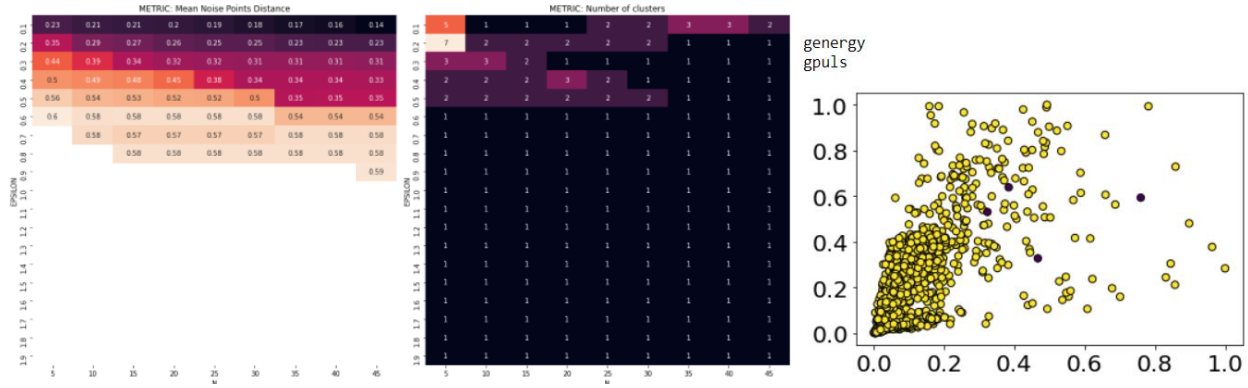

Figure 17: Right - HeatMap Grid Search
Scaled dataset (MinMaxScaler) without transformations. Left - Scatterplot genergy vs gpuls

Dataset with logarithmic and z-score transformation, without data scaling
Based on grid search, we chose an **ε = 1.3** and a **MinPts = 35**

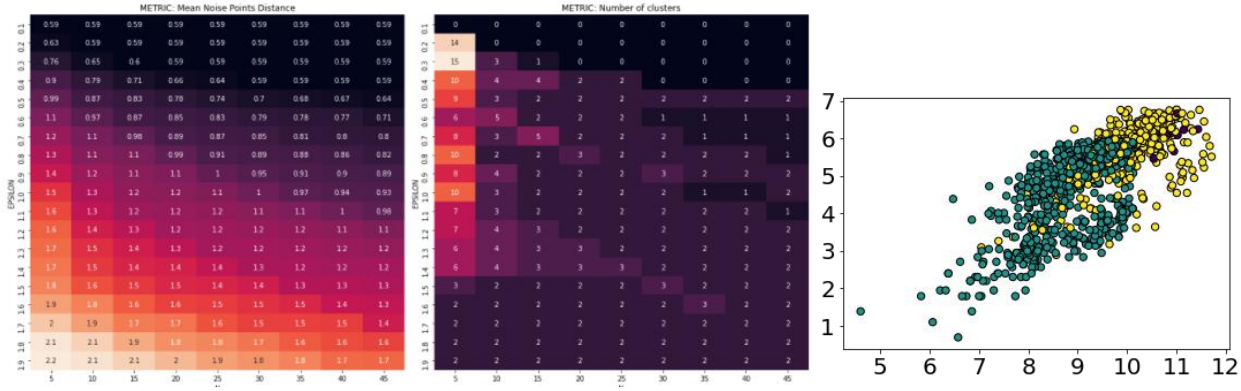

Figure 18: Right - HeatMap Grid Search
Dataset with logarithmic and z-score transformation, without data scaling. Left - Scatterplot genergy vs gpuls

Dataset with logarithmic and z-score transformation and scaled data (DataScaler)
eps=1.3 and **minPts=40**

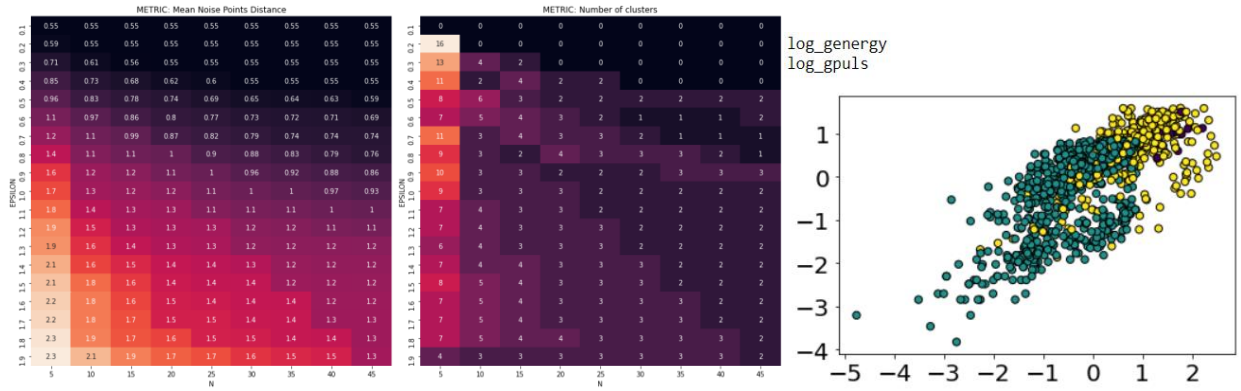

Figure 19: Right - HeatMap Grid Search Dataset with logarithmic and z-score transformation and scaled data (DataScaler). Left - Scatterplot genergy vs gpuls

Dataset with Logarithmic transformation and z-score and MinMax Scaled
eps = 0.5 and **MinPts = 35**

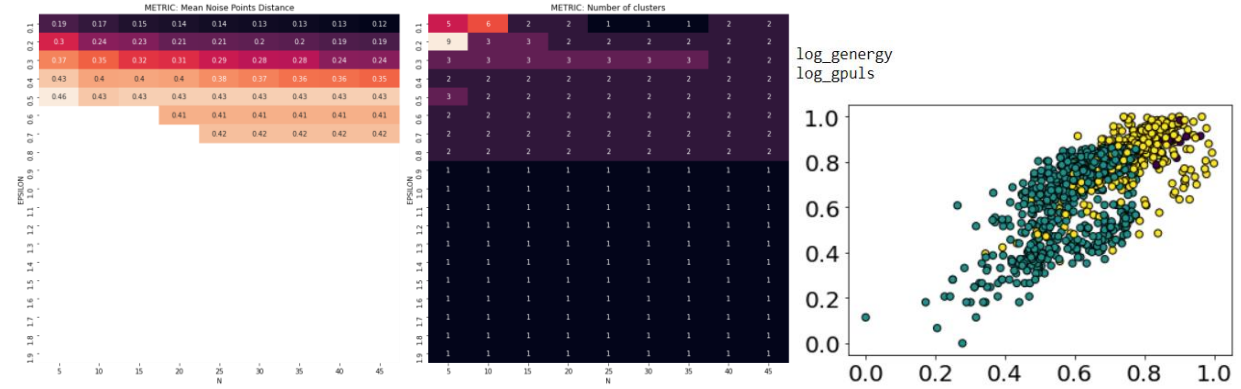

Figure 20: Right - HeatMap Grid SearchDataset with Logarithmic transformation and z-score and MinMax Scaled. Left - Scatterplot genergy vs gpuls

To conclude, giving DBSCAN the Dataset with Logarithmic transformation and z-score and MinMax Scaled, it seems to produce slightly better separated clusters, in contrast to the other ones, as shown in the figures. By the way there is not a consistent difference.

2.3 Analysis by hierarchical clustering

We decided to perform hierarchical clustering using the Euclidean distance and the same dataset and attributes we have used in the previous clustering techniques that gave us the best results (dataset with Logarithmic transformation and z-score and MinMax Scaled and non transformed and non scaled dataset).

Hierarchical clustering allows us to not specify the number of clusters in advance, as we can get any number of them simply by cutting the resulting dendrogram at the desired height.

We started to use the **complete linkage clustering** (Figure 21 - a). The algorithm generates three macro sub-clusters, two of them are unified in the same cluster. Because of that, we noticed the small distance between these two sub-clusters that converge to each other, then we chose to cut at level 2, in order to obtain 2 clusters.

Then, we used the **single-linkage clustering** which resulted to be not useful because it is unable to uniformly distribute the elements in the clusters. As we can see (Figure 21 - b), it creates several unbalanced groupings and a single cluster containing most of the data. So, we decided to discard this method.

After that, using the **group-average clustering** (Figure 22 - c), looking at the dendrogram, the algorithm created three sub-clusters that are not equally distributed, as we seen before: it produces one larger cluster (red) containing a great number of values and two little clusters (orange, green) that have fewer values. Nevertheless, if we cut at level 1.3, we obtain two clusters, one with the majority of data and another with a little sample. Consequently, this method is not very useful for our analysis and we did not consider it.

Finally, we tried the **Ward's method** (Figure 22 - d) obtaining such an optimal situation. We can immediately distinguish two main clusters that are generated by the algorithm and data is not unbalanced at all. By deciding to cut at level 10, we have found 2 clusters.

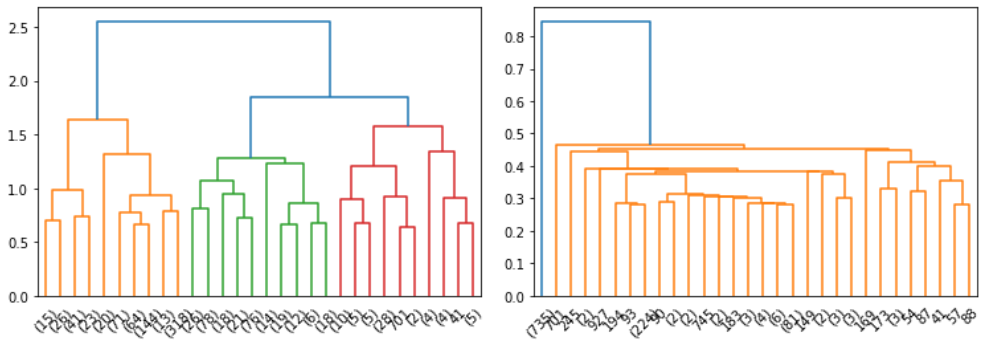

Figure 21: Complete linkage clustering (a) - Single linkage clustering (b)

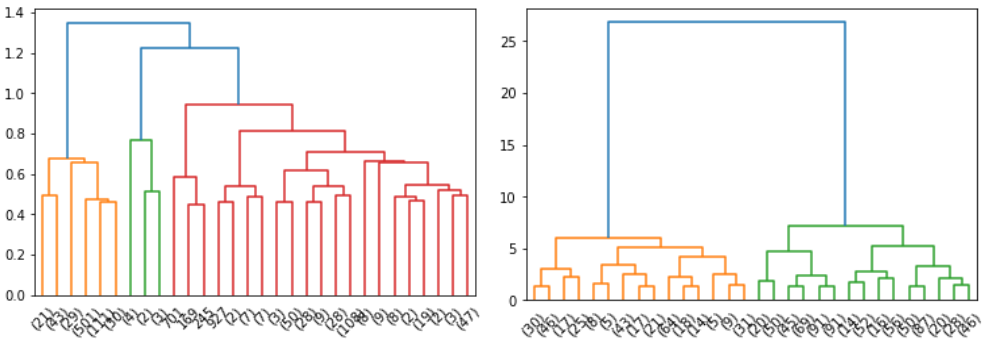

Figure 22: Group-average clustering (c) - Ward's method (d)

In conclusion, the final result of cluster analysis using hierarchical algorithms is a total number of 2 clusters, confirming the result obtained by K-means analysis. Indeed, as it is possible to see from the figures

above, there is evidence that a number of clusters equal to two emerges, since the distance from the two higher clusters to the others is high enough to distinguish them.

2.4 Final discussion

At the end of all the analysis through the different types of clustering algorithms, we can say that the optimal number of clusters in which it is possible to divide our data set is equal to 2.

Particularly, we have seen that DBSCAN is the algorithm that worst represents the data set. Based on the examinations we have done, it does not give us any useful information about the data.

On the other hand, we have seen how well both K-means and Hierarchical algorithms work, either with transformed data or not. As regards the first, despite the high Silhouette coefficient (0.74), clusters obtained are not equally dense and not well separated. This is because K-means works better with spherical shapes. About the second, we have seen how defining the number of clusters is simpler unlike other methods and it gives us a better representation and distinction of clusters, even if it is more expensive in computational cost terms.

3 Classification

In this chapter, we are going to apply classification techniques in order to predict what leads to the attribute *Class*. In order to do that, we have used Decision Tree model, KNN and Random Forest algorithms.

3.1 Classification by Decision Trees

We have analyzed both dataset with and without transformations reaching the same results. Because of that, we are going to present and analyze the classification technique using the dataset without transformation.

First of all, we split the dataset into a training set and test set, with a proportion of respectively 70% (761 rows) and 30% (327 rows).

Then we determine the importance of the features to decide which feature is the root node and the subsequent features to split with. We discovered that the features *nbumps*, *nbumps2*, *nbumps3* and *maxenergy* have not a particular importance, so we built it with only the following features: *genergy*, *gpuls*, *gdenergy*, *gdpuls* and *energy*. In particular, the most important feature is *gpuls,* indeed, as we can see in *Figure 23*, it becomes the root node.

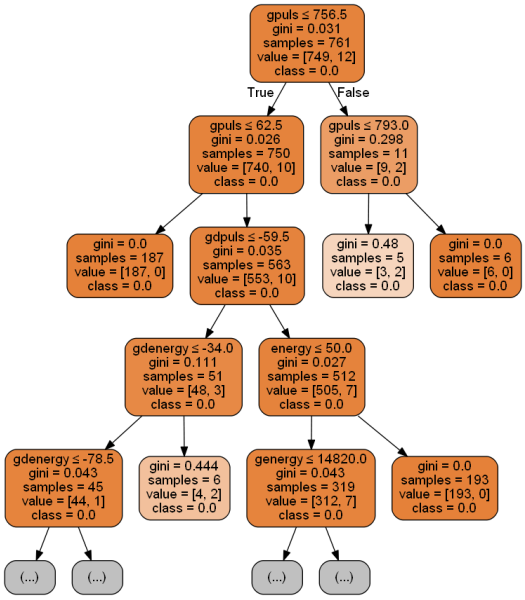

Figure 23: Decision Tree

To build the Decision Tree, we used the *Gini Index* as a criterion for impurity measure that has a minor impact on computational cost. Analyzing it, we can easily see that the probability to have 'Yes' as a

result is higher than to obtain 'No'. This is due to the fact that our features are not much correlated with respect to the *Class* attribute, that has an unbalanced starting distribution, and, in many cases, we have a non-hazardous state - 0.

To evaluate the performance of the prediction, we computed various performance metrics: *accuracy, precision, recall, f1-score, support* (Figure 24). The most suitable metric for our purpose is f1-score, which results to be equal to 0.99 for class 0 (non-hazardous state) and 0.00 for class 1 (hazardous state). Then we applied the decision tree on the test set to evaluate the performance of the model, and we obtained the same value for f1-score.

As we can see in the two tables in Figure 24, the *Accuracy* is high in both cases, almost the same and very close to one, this means that the fraction of the records is classified well by the model.

Accuracy 0.9842312746386334
F1 [0.99205298 0.   ]

	precision	recall	f1-score	support
0.0	0.98	1.00	0.99	749
1.0	0.00	0.00	0.00	12
accuracy			0.98	761
macro avg	0.49	0.50	0.50	761
weighted avg	0.97	0.98	0.98	761

(a)

Accuracy 0.9847094801223242
F1-score [0.99229584 0.   ]

	precision	recall	f1-score	support
0.0	0.98	1.00	0.99	322
1.0	0.00	0.00	0.00	5
accuracy			0.98	327
macro avg	0.49	0.50	0.50	327
weighted avg	0.97	0.98	0.98	327

(b)

Figure 24: classification metrics used on training set-(a) and test set-(b)

Then, we studied the confusion matrix that allows us to visualize the performance of the algorithm in both sets. We obtained almost the same percentage of the results: 0% for *True Positive* (*TP*) and *False Positive* (*FP*), 1.58% for *False Negative* (FN) and 98.45% for *True Negative* (*TN*). This leads us to say that our model classifies a new registration correctly. Having a low rate of FN is important to avoid the model classifying an hazardous bump as non-hazardous.

To analyze the predictive power of the classifier, we plotted the **Receiver Operating Characteristic Curve (ROC)** (Figure 25), that shows the trade-off between sensitivity (True Positive Rate - TPR) and specificity (False Positive Rate - FPR). Since it lays mostly under the diagonal, the curve obtained is not very accurate; although it almost lays on it, meaning that the model is a random classifier. So it randomly predicts the positive and the negative classes: TPR is equal to FPR at any threshold.

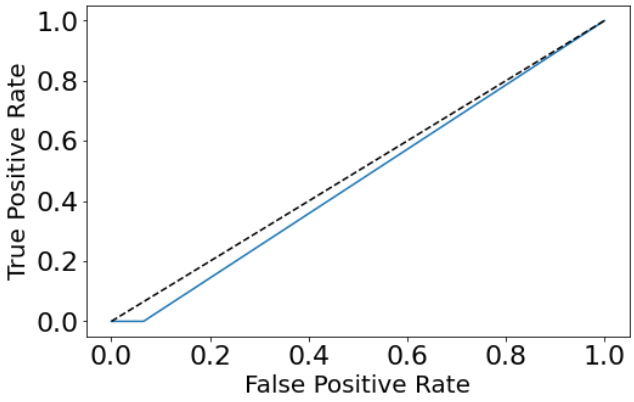

Figure 25: ROC curve on test set

3.2 Classification by Other Algorithms or Baselines

In order to do a comparison, we decided to use a Dummy Classifier, K-Nearest Neighbor and Random Forest algorithms and compare our model performances to the baseline model. The tool we have used to do this is a confusion matrix.

The fact that the distribution of *'class'* is unbalanced causes the model to correctly predict the negative class 0 (322 TN over 0 FP), but it doesn't manage to correctly classify the positive class 1 (Figure 24 - left).

More specifically, analyzing the different algorithms and looking at their confusion matrices, we can conclude that both Dummy Classifier and K-Nearest Neighbor (KNN) work in the same way: taking a look at Figure 24 - left and Figure 25 - right, we obtained a 98.47% (322 TN) of success about classifying correctly the attribute *class* versus a 1.53% (5 FN) of misclassification.

Moreover, checking at Figure 24 - right, the baseline algorithm has a percentage of success of 94.50% (309 TN) and a percentage of failure of 5.50% (5 FN and 13 FP). Although the latter is much less than the probability of success, for our data set it affects a lot, since when an hazardous state is classified as a non-hazardous state, it can cause serious consequences.

At last, as we can see at Figure 25 - left, the Random Forest algorithm has the 100% (5 TP and 322 TN) of success to predict the right class without falling into error on the classification.

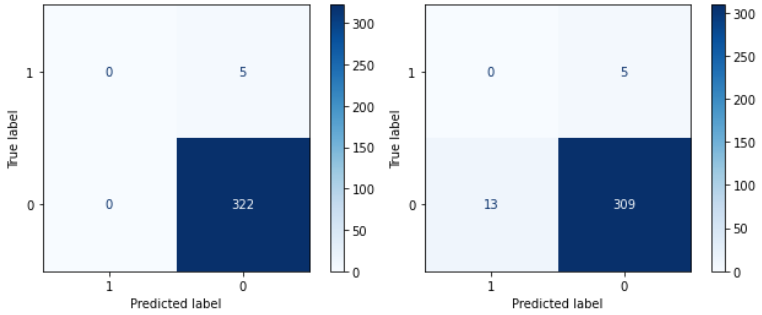

Figure 24: Dummy Classifier (left) and Decision Tree (right) confusion matrices

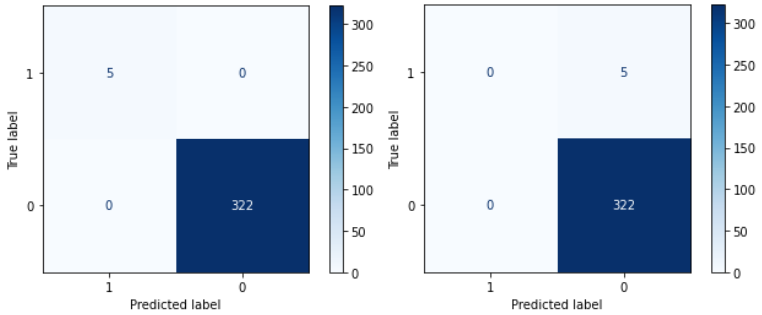

Figure 25: Random forest (left) and KNN (right) confusion matrices

In the end, since we wanted to evaluate their performance from a different point of view, we have studied their *f1-score* index to obtain a better vision.

```
Accuracy: 0.9844 (+/- 0.01) Accuracy: 0.9844 (+/- 0.01) Accuracy: 0.9844 (+/- 0.01)
F1-score: 0.4961 (+/- 0.00) F1-score: 0.4961 (+/- 0.00) F1-score: 0.4961 (+/- 0.00)
```

Figure 26: accuracy and f1-score indexes of Dummy Classifier, Random Forest and KNN algorithm

3.3 Final discussion

Concluding our Classification analysis, looking at the *f1-score* indexes of the algorithms we have used, we can state that the Decision Tree algorithm is the best algorithm with a value closer to one versus a low rate of the other ones.

Their *accuracy* index is high for all, so we are confident that the fraction of the records is classified well in each case. However, as we can see also in the previous algorithms, the model is not able to classify anything that belongs to class 1 (hazardous state), even though class 1 values are not equal to zero.This happens because, as we've seen during the whole process, the dataset is very unbalanced.

For this reason, while using the Dummy Classifier, we oversampled the data using the Smote function. However the results, as shown in figure 24, didn't improve much.

4 Pattern Mining

In this part of our analysis, we have used association rules and frequent itemsets to determine interesting patterns among the data. To do that, we have used the cleaned data set that we have created during the data understanding process, more precisely we have taken in consideration the most informative attributes: *genergy, gpuls, gdenergy, gdpuls, nbumps, nbumps2, nbumps3, energy, maxenergy* and *class*.

In order to apply the Apriori Algorithm and extract Frequent Pattern before, and Association Rules after, we have discretized continuous variables using the binning technique: we have specified the number of bins, which corresponds to our categorical values, and then dropped the transformed columns.

4.1 Frequent Pattern extraction

We applied the Apriori algorithm on the list of discrete variables and extracted the frequent itemsets by trying different values of support threshold. With a support equal to 0.02, the frequent itemset results to be a non-hazard bump with an *energy* between (700.0, 6100.0) and a *max_energy* between (700.0, 2000.0), with a support equal to 2.57 as shown in the first five results below.

```
1. (('3-hazard',  '(700.0, 2000.0]_maxenergy',  'nothaz-state', '(700.0,
   6100.0]_energy'),  2.5735294117647056),
2. (('3-hazard', '(700.0, 2000.0]_maxenergy', 'nothaz-state'),
   2.5735294117647056),
3. (('3-hazard', '(700.0, 2000.0]_maxenergy', '(700.0, 6100.0]_energy'),
   2.5735294117647056),
4. (('3-hazard', '(700.0, 6100.0]_energy', 'nothaz-state'), 3.0330882352941178),
5. (('3-hazard', '(700.0, 6100.0]_energy', '(25885.0, 121520.0]_genergy',
   'nothaz-state'), 2.297794117647059)
```

After that, we studied the difference between all frequent itemsets and the maximal only: we obtained that the maximal frequent itemset is the same. When increasing the level of support, for example to 0.06, the number of frequent itemsets decreases and changes as we can see in the first five results below.

```
1. (('2-hazard', '(700.0, 6100.0]_energy', 'nothaz-state'), 6.341911764705882),
2.  (('(-42.0, -28.0]_gdpuls', '(-0.001, 700.0]_energy', '(-0.001,
   700.0]_maxenergy', 'Not-hazardous'), 13.878676470588236),
3.  (('(-42.0, -28.0]_gdpuls', '(-0.001, 700.0]_energy', '(-0.001,
   700.0]_maxenergy'), 13.878676470588236),
4.  (('(-42.0, -28.0]_gdpuls', '(-0.001, 700.0]_energy', 'Not-hazardous'),
   13.878676470588236),
5.  (('(-42.0, -28.0]_gdpuls', '(-0.001, 700.0]_energy', 'nothaz-state', '(-
   0.001, 700.0]_maxenergy', 'Not-hazardous'), 13.602941176470587)
```

4.2 Discuss Frequent Pattern

To analyze the frequent pattern, we looked at sintactic of the first five frequent itemsets. From this first analysis, we could see that inside the frequent itemsets, we can find a level of medium-high energy level (2-hazard or 3-hazard), with a level of energy between 700.0 and 6100.0 or 700.0 and 2000.0, together with a not-hazard classification of the bump. These patterns could be expected looking at the distribution of the class, the energy feature and the bumps ones.

Then, we filtered the maximal frequent itemsets for the number of hazardous-state frequent itemsets and not hazardous-state ones. As expected, the number of maximal not hazardous frequent itemsets is significantly higher than the hazardous ones. Also, increasing the support, the not hazardous itemsets decrease. We also observed that maximal frequent itemsets are a subgroup of closed frequent itemsets (Figure 27 - b).

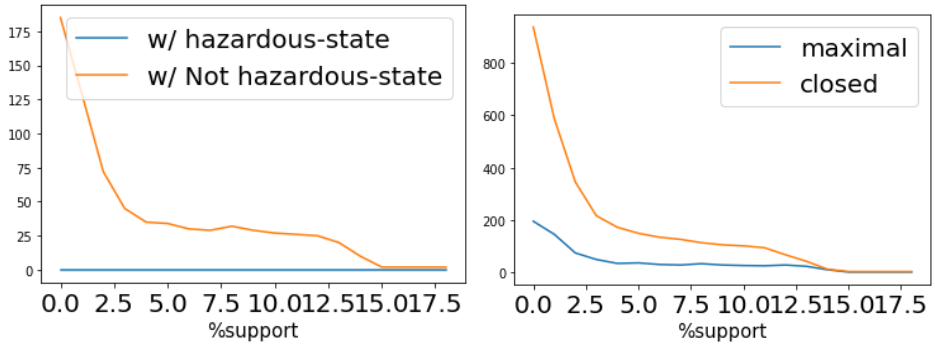

Figure 27: relation between support and frequent itemsets (a) - support and maximal/closed frequent itemset (b)

4.3 Association Rules extraction

We extracted association rules by specifying different thresholds of confidence. With a support value of 0.10% and a confidence value of 0.60% and with this setting we found that gdpuls' (-42.0, -28.0], 'energy' (-0.001, 700.0], 'maxenergy' (-0.001, 700.0], and 'Not-hazardous' bump state, implies a not-hazardous classified bump, we obtained a support of 13.6%, a confidence of 0.98 and a lift of 0.99. Increasing the confidence to 0.90% didn't change the results.

When we decrease the threshold of min_conf to 50%, the confidence decreases to 0.57.

4.4 Discuss Association rules

From this heat map we can see that if we have less support we will have more rules with less confidence. Also, by decreasing the level of min_conf we obtain more rules in comparison to the number of rules obtained with a higher min_conf.

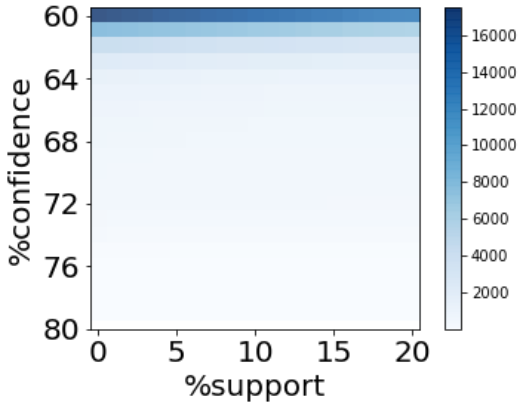

Figure 28: heatmap support-confidence

The rules below are the five most important rules we found in the dataset, so the ones that have the highest confidence value.

1. `('nothaz-state', ('(-42.0, -28.0]_gdpuls', '(-0.001, 700.0]_energy', '(-0.001, 700.0]_maxenergy', 'Not-hazardous'), 148, 13.602941176470587, 0.9801324503311258, 0.9956901082728897)`
2. `('nothaz-state', ('(-42.0, -28.0]_gdpuls', '(-0.001, 700.0]_energy', '(-0.001, 700.0]_maxenergy'), 148, 13.602941176470587, 0.9801324503311258, 0.9956901082728897)`
3. `('nothaz-state', ('(-42.0, -28.0]_gdpuls', '(-0.001, 700.0]_energy', 'Not-hazardous'), 148, 13.602941176470587, 0.9801324503311258, 0.9956901082728897)`

4. ('nothaz-state', ('(-42.0, -28.0]_gdpuls', '(-0.001, 700.0]_energy'), 148, 13.602941176470587, 0.9801324503311258, 0.9956901082728897)
5. ('nothaz-state', ('(-42.0, -28.0]_gdpuls', '(-0.001, 700.0]_maxenergy', 'Not-hazardous'), 151, 13.878676470588236, 0.9805194805194806, 0.9960832817975676)

For example, the first rule tells us that the most confident situation to predict the not-hazardous state is having the following values in the attributes : ('(-42.0, -28.0]_gdpuls', '(-0.001, 700.0]_energy', '(-0.001, 700.0]_maxenergy', 'Not-hazardous').

After analyzing the different amount of rules using different *min_confidence* thresholds, we have decided to use the *min_confidence*=0.60 and *support* = 0.10. Then, we plotted the histogram of the confidence and lift of not-hazardous state rules. In both cases, as we can see in the figure 29, the confidence and the lift of the majority of the rules is high, so we are confident that they are valid.

In particular, looking at the lift distribution, it is easy to see how in most rules the value is greater than one, meaning that the occurrence of the rule body has a positive effect on the occurrence of the rule head. On the other rules, the lift value is near one, which means that the occurrence of the rule body has almost no effect on the occurrence of the rule head.

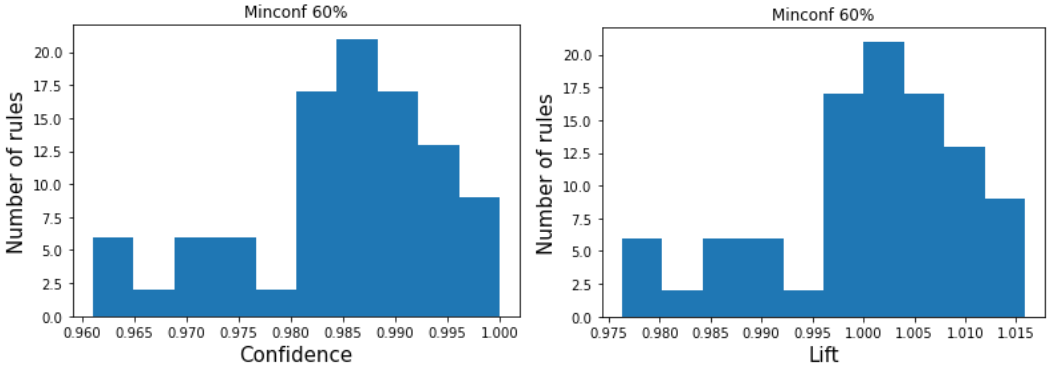

Figure 29: Confidence and lift distributions

5 Final discussion

For the analysis of the *class*, we have used different data mining techniques, trying to understand whether the class of the bump is 0 (not-hazardous state) or 1 (hazardous-state).

In the **Data Understanding** section, we addressed issues related to data semantics and data quality: detecting outliers, deleting redundant and not relevant variables observing the correlation. Our main goals were to clean up the variables, to reduce the dimensionality of the dataset and to have a basic understanding of variable distributions.

In the **Clustering** section, we focused on numeric attributes (*genergy, gpuls, gdenergy, gdpuls*) to find groups of similar points in the dataset. We found, as an obstacle, clusters not separated very well. We found the best performances with K-means with k = 2, and Hierarchical (in particular with the Ward's method).

In the **Classification** phase, we tried to predict the value of the target variable *Class*. We performed the Decision Tree, Random Forest, Dummy Classifier and the KNN algorithms. The accuracy and the F1 score are very high, but the classifier does not obtain good results, due to the unbalance of our dataset.

To conclude, in the **Pattern Mining** phase, we obtained some interesting rules by discarding highly unbalanced categorical attributes.