## Introduction

The dangers associated with coal mining are myriad; black lung, flammable gas pockets, rockbursts, and tunnel collapses are all very real dangers that mining companies must consider when attempting to provide safe working conditions for miners. One class of mining hazard, commonly called 'seismic hazards', are notoriously difficult to protect against and even more difficult to predict with certainty. Therefore, predicting these hazards has become a well-known problem for machine learning and predictive analytics. The UCI Machine Learning Repository (https://archive.ics.uci.edu) provides a 'seismic bumps' data set that contains many records of combined categorical and numeric variables that could be used to predict seismic hazards. This 'seismic bumps' data set can be found at https://archive.ics.uci.edu/ml/datasets/seismic-bumps.

Our analysis attempts to use logistic regression techniques to predict whether a seismic 'bump' is predictive of a notable seismic hazard. We attempt to characterize our prediction accuracy and compare the results against the state of the art results from other statistical and machine learning techniques, that are included within the data set.

## Data set description

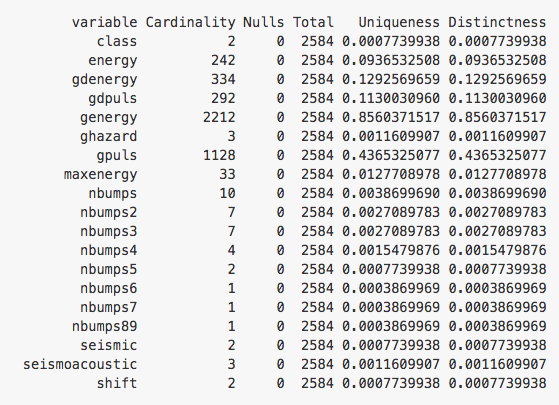
The data were taken from instruments in the Zabrze-Bielszowice coal mine, in Poland. There are 2,584 records, with only 170 class = 1 variables, so the data are significantly skewed towards non-hazardous training data. Field descriptions are below, but essentially energy readings and bump counts during one work shift are used to predict a 'hazardous' bump during the next shift. From the data description, a 'hazardous bump' is a seismic event with > 10,000 Joules, and a 'shift' is a period of 8 hours. For the sake of reference, a practical example of 10,000 Joules would be the approximate energy required to lift 10,000 tomatoes 1m above the ground. A class = 1 variable result signifies that a hazardous bump did, indeed, occur in the following shift to the measured data. Here is an example of the fields in the data set.

From the UCI Machine Learning Repository, these are the field descriptions:

* **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);
* **seismoacoustic**: result of shift seismic hazard assessment in the mine working obtained by the seismoacoustic method;
* **shift**: information about type of a shift (W - coal-getting, N -preparation shift);
* **genergy**: seismic energy recorded within previous shift by the most active geophone (GMax) out of geophones monitoring the longwall;
* **gpuls**: a number of pulses recorded within previous shift by GMax;
* **gdenergy**: a deviation of energy recorded within previous shift by GMax from average energy recorded during eight previous shifts;
* **gdpuls**: a deviation of a number of pulses recorded within previous shift by GMax from average number of pulses recorded during eight previous shifts;
* **ghazard**: result of shift seismic hazard assessment in the mine working obtained by the seismoacoustic method based on registration coming form GMax only;
* **nbumps**: the number of seismic bumps recorded within previous shift;
* **nbumps2**: the number of seismic bumps (in energy range [10^2,10^3)) registered within previous shift;
* **nbumps3**: the number of seismic bumps (in energy range [10^3,10^4)) registered within previous shift;
* **nbumps4**: the number of seismic bumps (in energy range [10^4,10^5)) registered within previous shift;
* **nbumps5**: the number of seismic bumps (in energy range [10^5,10^6)) registered within the last shift;
* **nbumps6**: the number of seismic bumps (in energy range [10^6,10^7)) registered within previous shift;
* **nbumps7**: the number of seismic bumps (in energy range [10^7,10^8)) registered within previous shift;
* **nbumps89**: the number of seismic bumps (in energy range [10^8,10^10)) registered within previous shift;
* **energy**: total energy of seismic bumps registered within previous shift;
* **maxenergy**: the maximum energy of the seismic bumps registered within previous shift;
* **class**: the decision attribute - '1' means that high energy seismic bump occurred in the next shift ('hazardous state'), '0' means that no high energy seismic bumps occurred in the next shift ('non-hazardous state').

## Variable Types and Cardinality

There are 18 input variables and one binary output variable ("class"). The data are mostly numeric with 4 categorical input variables. However, some of the numeric values only contain a handful of discrete values which can be viewed as coded categorical variables. In particular, maxenergy and the 'nbumps(n)' variables can be treated as categorical. So, in short, we see the following breakdown in variable types:

class – output – binary

\*energy – input – numeric

g\*puls – input – numeric

ghazard – input – categorical

nbumps(n) – input – categorical

seismoacoustic – input - categorical

## Exploratory Data Analysis

It is important to understand how many observations are "hazardous state (class = 1)" and "non-hazardous state (class = 0)". There are 2414 records with a non-hazardous state response and 170 records with a hazardous state response. As mentioned above, the data set output variable is highly skewed.

### Main Effects

The ‘nbumps’

### Logistic Regression Assumptions

#### Linearity

#### Independence of Errors

#### Multi-collinearity

Collect all the numeric variables and check for multi-collinearity:

```{r}

seismicDataNumeric <- seismicData %>%

select(genergy, gpuls, gdenergy, gdpuls, energy, maxenergy)

```

```{r}

# Create the correlation matrix

M <- round(cor(seismicDataNumeric), 2)

# Create corrplot

corrplot(M, method="pie", type = "lower")

```

## Logistic Regression Model

Fit a logistic regression model with what you think could be contributing to the seismic hazard.

```{r}

seismic\_model <- glm(class ~ seismic + seismoacoustic + shift + ghazard,

data = seismicData, family = "binomial")

summary(seismic\_model)

```

Making a binary prediction:

We used the glm() function to build a logistic regression model of the `class` variable. As with many of R's machine learning methods, you can apply the `predict()` function to the model object to forecast future behavior. By default, predict() outputs predictions in terms of log odds unless `type = "response"` is specified. This converts the log odds to probabilities.

Because a logistic regression model estimates the probability of the outcome, it is up to you to determine the threshold at which the probability implies action. One must balance the extremes of being too cautious versus being too aggressive. For example, if we classify an observation which has a probability of being in class 1 as 99% or greater, then we may miss out on some observations that may indeed be class 1 but were classified as class 0. This balance is particularly important to consider for severely imbalanced outcomes, such as in this dataset where class 1 are relatively rare.

```{r}

# make a copy

seismicDataPredictions <- seismicData

# Estimate the probability of class 1

seismicDataPredictions$prob <- predict(seismic\_model, type = "response")

```

Find the actual probability of an observation to be in class 1.

```{r}

mean(as.numeric(as.character(seismicData$class)))

```

We will use this as our cut-off threshold.

```{r}

seismicDataPredictions$pred <- ifelse(seismicDataPredictions$prob > 0.0657, 1, 0)

```

Now calculate the model accuracy:

```{r}

mean(seismicDataPredictions$pred == seismicDataPredictions$class)

```

This shows that the logistic regression model with all the factor variables made a correct prediction 44% of the time.

What would be the accuracy of the model if a model had simply predicted class 0 for each observation ?

```{r}

seismicDataPredictions$predNull <- 0

mean(seismicDataPredictions$predNull == seismicDataPredictions$class)

```

With an accuracy of 44% the model is actually performing worse than if it were to predict class 0 for every record.

This illustrates that "rare events" create challenges for classification models. When 1 outcome is very rare predicting the opposite can result in very high accuracy.

Calculate ROC Curves and AUC:

The previous exercises have demonstrated that accuracy is a very misleading measure of model performance on imbalanced datasets. Graphing the model's performance better illustrates the tradeoff between a model that is overly agressive and one that is overly passive. Here we will create a ROC curve and compute the area under the curve (AUC) to evaluate the logistic regression model that we created above.

```{r}

ROC <- roc(seismicDataPredictions$class, seismicDataPredictions$prob)

plot(ROC, col = "blue")

text(x = .42, y = .6,paste("AUC = ", round(auc(ROC), 2), sep = ""))

```

Dummy variables, missing data and interactions:

```{r}

seismic\_model <- glm(class ~ . , data = seismicData, family = "binomial")

summary(seismic\_model)

```

## Evaluating the model performance.

## Comparing the performance of classification techniques.

Within the ARFF file is a large table of various classification techniques and their results.

Classification results using stratified 10-fold cross-validation repeated 10 times

| Algorithm | Acc. | BAcc. | Acc.0 spec | Acc.1 sense | Size |

------------------------------|-----------|-----------|-------------|-------------|-----|

| q-ModLEM(entropy-RSS) (1) | 80.2(5.1) | 69.1(6.2) | 81.90 | 56.35 | 27.5 |

| q-ModLEM(entropy-Corr.) (1) | 82.9(4.5) | 67.9(7.2) | 85.15 | 50.65 | 45.5 |

| MODLEM (2) | 92.5(0.8) | 52.6(2.8) | 98.58 | 6.65 | 145.5 |

| MLRules(-M 30) (3) | 93.2(0.3) | 50.5(1.3) | 99.69 | 1.29 | 30 |

| MLRules(-M 100) (3) | 92.9(0.6) | 52.0(2.2) | 99.10 | 4.88 | 100 |

| MLRules(-M 500) (3) | 92.3(0.6) | 52.9(2.8) | 98.27 | 7.59 | 500 |

| BRACID (4) | 87.5(0.4) | 62.0(2.6) | 91.38 | 32.71 | - |

| Jrip (Weka) | 93.0(0.6) | 51.4(2.4) | 99.35 | 3.47 | 1.8 |

| PART (Weka) | 92.1(0.8) | 52.7(3.5) | 98.09 | 7.35 | 34 |

| J48 (Weka) | 93.1(0.8) | 50.2(0.9) | 99.64 | 0.82 | 5.6 |

| SimpleCart (Weka) | 93.4(0.0) | 50.0(0.0) | 100 | 0.00 | 1.0 |

| NaiveBayes (Weka) | 86.7(2.0) | 64.7(5.8) | 90.08 | 39.41 | - |

| IB1 (Weka) | 89.4(1.6) | 55.3(4.8) | 94.54 | 16.06 | - |

| RandomForest(-I 100) (Weka) | 93.1(0.6) | 52.1(2.5) | 99.31 | 4.88 | 100 |

---------------------------------------------------------------------------------------------

## References

\* [Application of rule induction algorithms for analysis of data collected by seismic hazard monitoring systems in coal mines.](https://actamont.tuke.sk/pdf/2013/n4/7sikora.pdf)

\* [A Study of Rockburst Hazard Evaluation Method in Coal Mine](https://www.hindawi.com/journals/sv/2016/8740868/#B13)

\* [Classification: Basic concepts, decision trees and model evaluation](https://www-users.cs.umn.edu/~kumar001/dmbook/ch4.pdf)