Seismic Bumps and
Predicting Seismic Hazards

Introduction

The dangers associated with coal mining are myriad; black lung, flammable gas pockets, rock-bursts, 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.

Problem Statement

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

shift – input – binary

variable	Cardinality	Nulls	Total	Uniqueness	Distinctness
class	2	0	2584	0.0007739938	0.0007739938
energy	242	0	2584	0.0936532508	0.0936532508
gdenergy	334	0	2584	0.1292569659	0.1292569659
gdpuls	292	0	2584	0.1130030960	0.1130030960
genergy	2212	0	2584	0.8560371517	0.8560371517
ghazard	3	0	2584	0.0011609907	0.0011609907
gpuls	1128	0	2584	0.4365325077	0.4365325077
maxenergy	33	0	2584	0.0127708978	0.0127708978
nbumps	10	0	2584	0.0038699690	0.0038699690
nbumps2	7	0	2584	0.0027089783	0.0027089783
nbumps3	7	0	2584	0.0027089783	0.0027089783
nbumps4	4	0	2584	0.0015479876	0.0015479876
nbumps5	2	0	2584	0.0007739938	0.0007739938
nbumps6	1	0	2584	0.0003869969	0.0003869969
nbumps7	1	0	2584	0.0003869969	0.0003869969
nbumps89	1	0	2584	0.0003869969	0.0003869969
seismic	2	0	2584	0.0007739938	0.0007739938
seismoacoustic	3	0	2584	0.0011609907	0.0011609907
shift	2	0	2584	0.0007739938	0.0007739938

Figure 1: Summary of Variables

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 main effects for this study are considered to be the all numeric variables, plus ghazard, seismoacoustic and shift. This stands to reason, since numerical energy readings and shift activity type all seem like they would impact the number of hazardous seismic events in

the next shift. The nbumps class of variables are left out for more advanced models, since the resonance and frequency ranges could have a multitude of confounding variables that we, without significant mining expertise, would miss. To test that nbumps isn't necessarily the largest effect, we looked at a side-by-side histogram of nbump records for each of the two output classes:

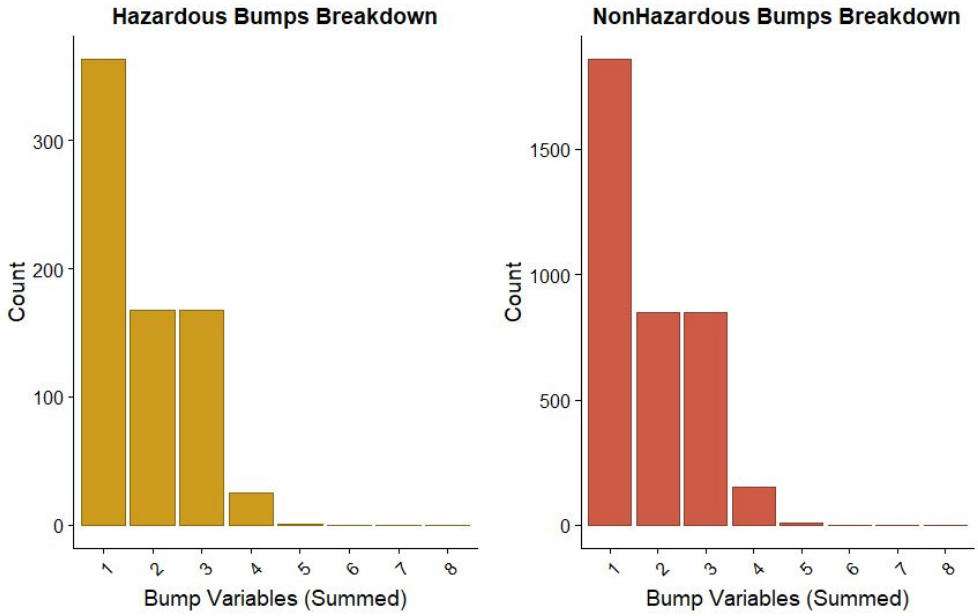

Figure 2 comparison of nbumps ratios across class

The pattern of frequency distributions appears to be consistent, regardless of class. Therefore, we will not be making 'nbumps' one of our main effects variables.

Logistic Regression Assumptions

Output Variable

Logistic regression requires a categorical output variable. In this case, our output variable (class) is a binary categorical variable.

Independence of Observations

We are making the assumption that each measurement is an independent measurement, taken at different times, from the same mine. This is based on the data set description at the UCI Machine Learning repository, and the fact that you can't take multiple simultaneous readings from the same instrument.

Multi-collinearity

We used the following chart to assess the correlation of variables with each other. The most highly correlated variables are energy and maxenergy. It is also interesting that the gpuls:genergy and gdpuls:gdenergy are somewhat correlated, but we would expect that an increase in the count of pulses per shift would raise the calculated energy per shift. We would also expect the change in variance in the number of pulses per shift (gdpuls) to correlate somewhat with the variance in the energy measured (gdenergy). For this model, we decided to leave all the main effects variables intact and address any multicollinearity issues after glmnet's automatic feature selection, if necessary (spoiler: it wasn't).

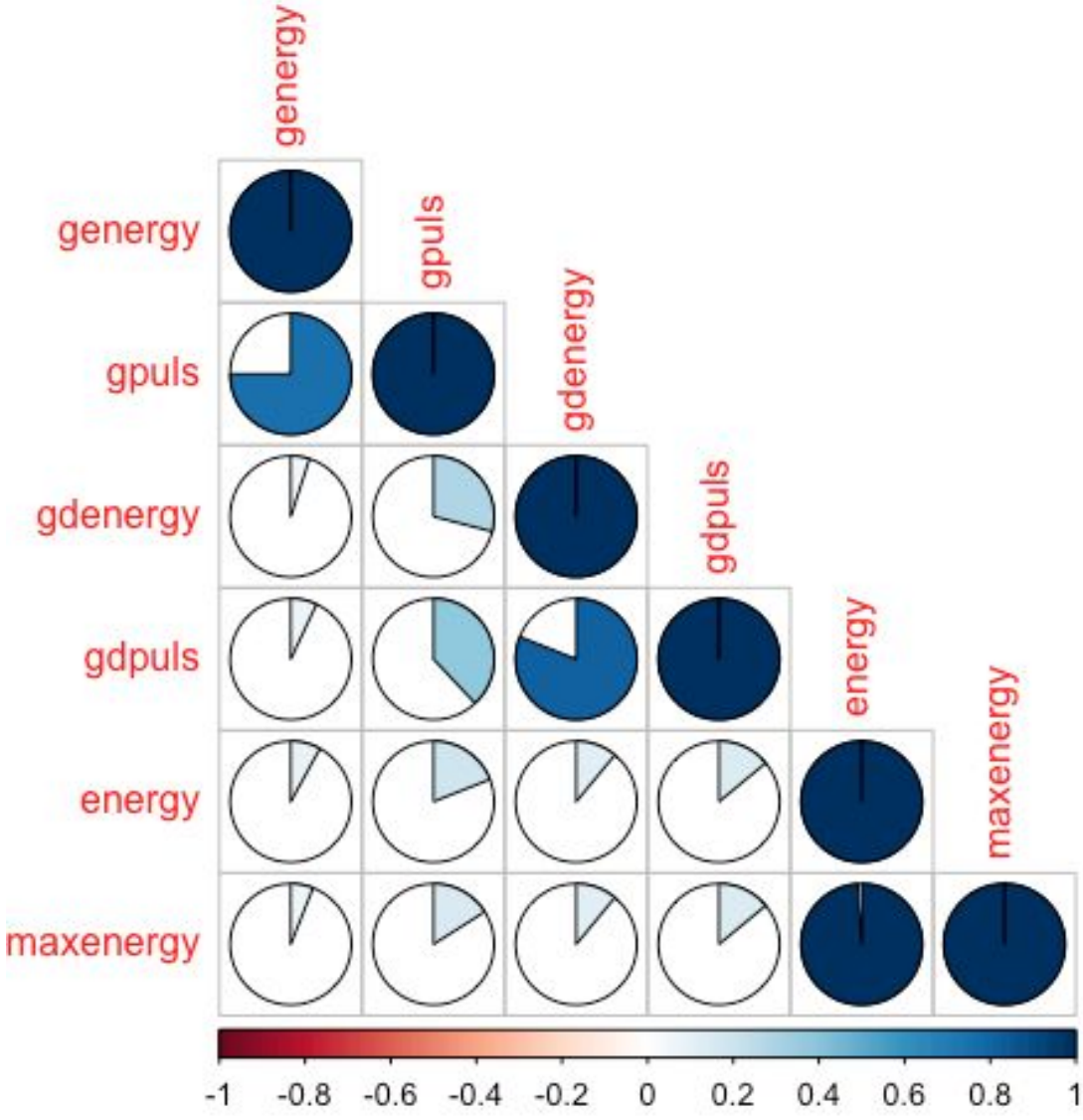

Figure 3 Correlation Matrix

Problem with unbalanced class variable

The dependent variable in our model is quite unevenly balanced with a 14:1 ratio for non-hazardous to hazardous. To illustrate the problem with having an unbalanced class variable we compare accuracy of the logistic regression full model with the null model.

Full Model:

We first built a logistic regression model taking all the observations and variables into account. glm(class ~ . , data = sb, family = "binomial"). Next we predict using this full model and calculate the probability of getting a hazardous bump. To convert these probabilities to classes, we define a threshold. A good value for threshold is the mean of the original response variable. Probabilities greater than this threshold are categorized into hazardous bump class and probabilities lesser than the threshold are categorized into non-hazardous bump class. Model accuracy is then calculated by comparing with the actual class variable.

Our calculation show that the logistic regression model with all the observations and variables made a correct prediction 79% of the time.

Null Model:

What if our model always predicted class == 0, then what is the accuracy ? The null model accuracy is 93.42%. With our full model accuracy of 79% 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 one outcome is very rare predicting the opposite can result in very high accuracy. We 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 aggressive and one that is overly passive. In the next sections we will create a ROC curve and compute the area under the curve (AUC) to evaluate the logistic regression model.

Balancing the class variable

When balancing the class variable, a key question to ask ourselves is by how much should we shrink the number of observations for the class with most values. To aid in answering this question, we can use cross validation to figure out the optimal value to choose. We developed a function in R called *cVLogisticRegression(n, k, size)* - where n is the number of observations of the majority class, k is the number of folds and size is the number of observations per fold to include in test and train data. The function returns a vector of “Area Under the Curve” AUC values. A histogram of these AUC values for n = 170, n = 500, n = 1000 is shown in the figure.

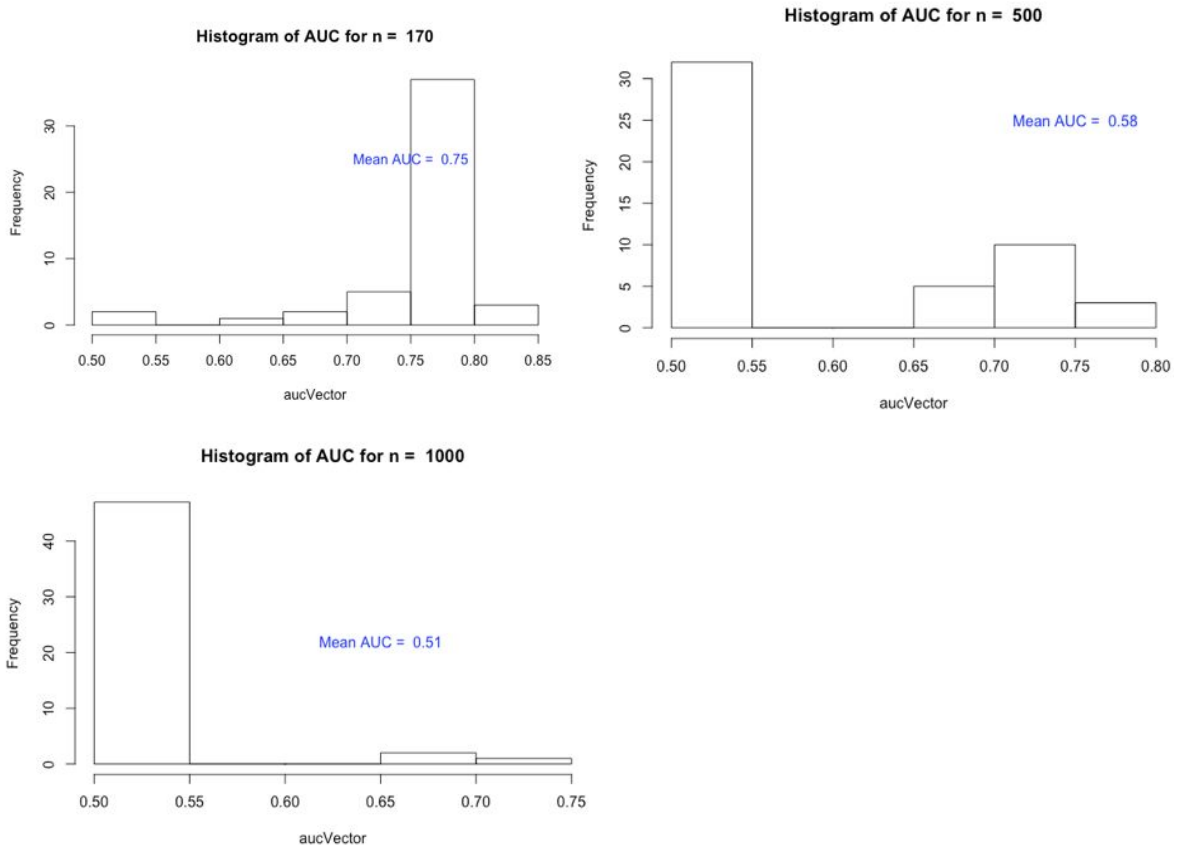

Figure 4 Cross Validation n-selection AUC Results

As we can see, as the n value increases (i.e imbalance increases) our model performance is decreasing. This suggests that we should carefully choose the n value while building our model.

The effect of balancing the class variable can be observed in the below table:

170 -> AUC was close to 80%

500 -> AUC was close to 60%

1000 -> AUC was close to 50%

This shows that logistic regression model's performance is largely dependent on having a balanced class variable. We chose to use n = 200 which gives us a fairly balanced dataset with 200 non-hazardous cases and 170 hazardous cases. In the next section, we dive into the details of model building using train/test split and n-fold cross validation.

Logistic Regression using cv.glmnet with balanced dataset with train/test split

Using the balanced dataset with approximately equal number of hazardous and nonhazardous cases we split the balanced dataset into two new datasets, one for training(75%) and one for testing(25%). Using these new balanced datasets we are able to construct and test our model. glmnet package allows us to fit a generalized linear model via penalized maximum likelihood. The regularization path is computed for the lasso or elasticnet penalty at a grid of values for the regularization parameter lambda. The function cv.glmnet automatically performs a grid search to find the optimal value of lambda.

Below is a plot of the model. The plot shows that the log of the optimal value of lambda (i.e the one that minimizes the misclassification error) is approximately -2.3. The objective of regularisation is to balance accuracy and simplicity. In this context, this means a model with the smallest number of coefficients that also gives a good accuracy. To this end, the cv.glmnet function finds the value of lambda that gives the simplest model. In this case there are 4 variables which are contributing the most.

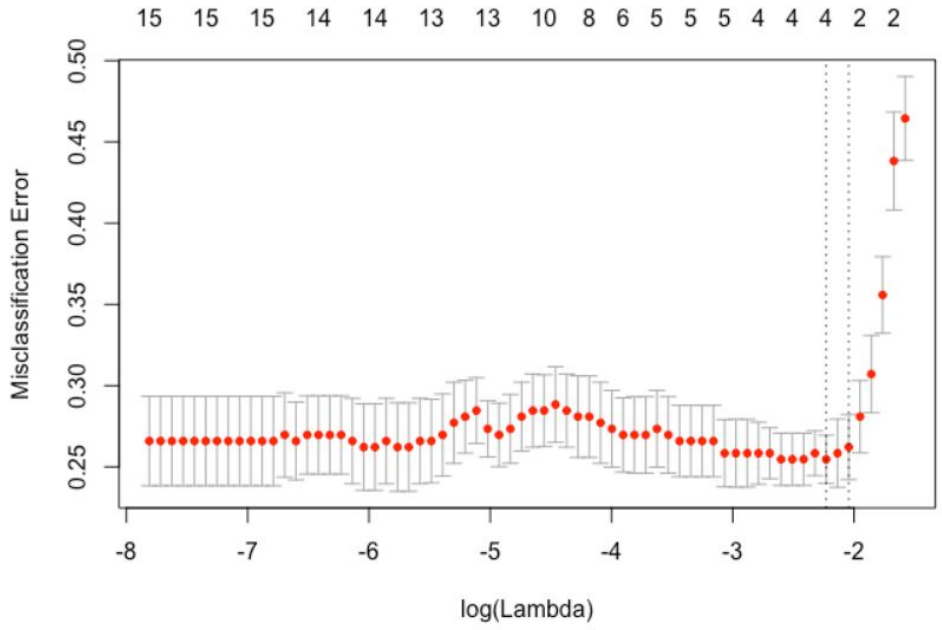

Figure 5 Plot of the Model

Looking at the figure 6, we can see that `seismicb, shiftW, gpuls and nbumps` are part of the final model. The coefficients of all other variables have been set to zero by the algorithm. Lasso has reduced the complexity of the fitting function from 18 predictors to 4 predictors.

```
## 22 x 1 sparse Matrix of class "dgCMatrix"
##                               1
## (Intercept)     -0.6877404393
## (Intercept)      .
## seismicb         0.0412656255
## seismoacousticb  .
## seismoacousticc  .
## shiftW           0.0011580912
## genergy          .
## gpuls            0.0003433553
## gdenergy         .
## gdpuls           .
## ghazardb         .
## ghazardc         .
## nbumps           0.2048022200
## nbumps2          .
## nbumps3          .
## nbumps4          .
## nbumps5          .
## nbumps6          .
## nbumps7          .
## nbumps89         .
## energy           .
## maxenergy        .
```

Figure 6 Coefficients

Using the above model we can now predict on the test dataset. The figure7 below shows the ROC curve. The AUC value is a bit less compared to the complex model (AUC for full model was 79%) but the really nice thing about this is that using a simpler function we are able to do a good job in fitting the signal in the data. Due to the simplicity of this model, it is less prone to overfitting and we can rely on the prediction accuracy.

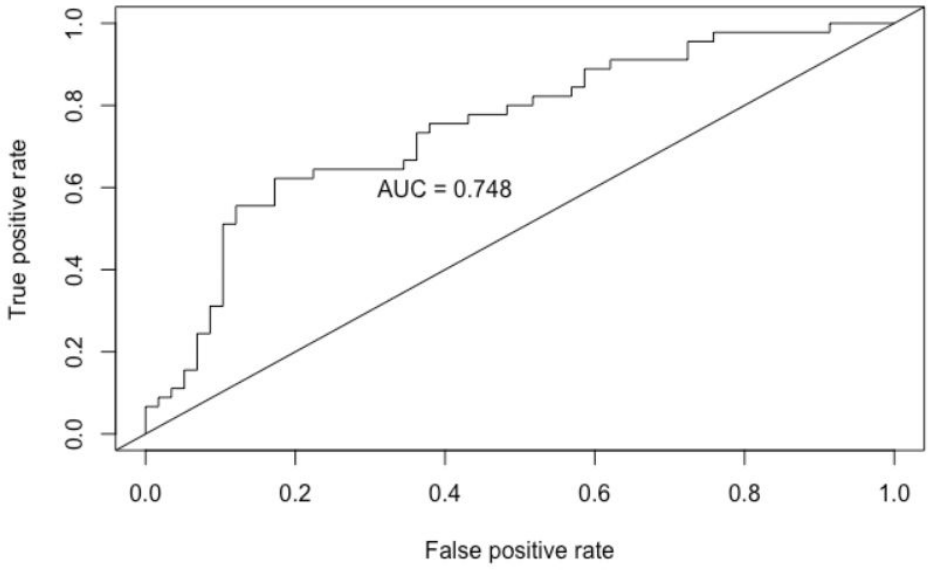

Figure 7 ROC Curve

Evaluating model performance using k-fold Cross Validation

Using cross validation we can assess how well our model building process works. The idea is that we can know how well our model will perform on new data not yet collected. We will use AUC as the performance metric. A 50-fold cross validation was performed with a training set size of 60 and test set size of 60 over a balanced dataset of 200 observations. After running the cross validation we created a histogram to show the results.

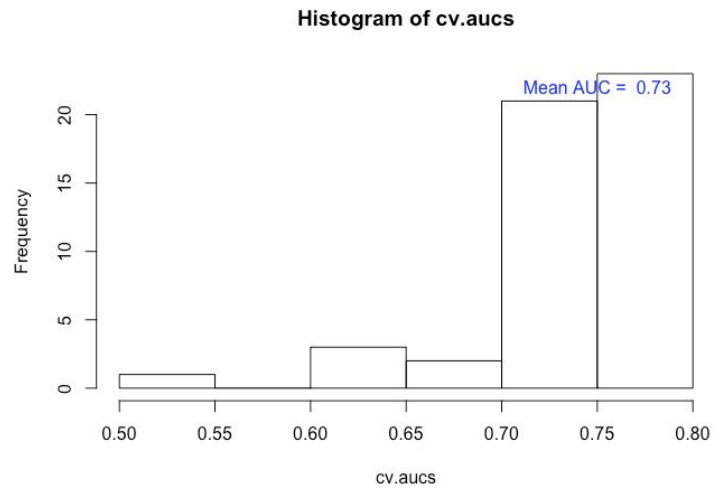

Figure 8 Histogram of Cross Validation

This indicates that a majority of time our model prediction performance lies between 70 to 80%.

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

Figure 10 Comparison

Conclusion

This data set has been used in many, many papers and theories. This seismic bump problem is an interesting problem, an important problem, and has a fascinating data set. However, it is challenging due in no small part to the unbalanced data. Like the infamous Challenger O-Ring data set, the lack of a hazardous result is not a 'null' case but instead informs the model about situations where nothing bad happens... a class = 0 event. Unlike the Challenger data set, we have many more parameters to choose from, and myriad ways of dealing with the imbalanced data. We chose to Randomly Under-Sample, but we could also do Synthetic Minority Over-Sampling, Cluster-Based Over-Sampling, Random Over-Sampling, Algorithmic Ensemble Sampling, Bagging, Boosting, and probably many other techniques which aren't covered in a graduate level statistics course. We chose a pretty straightforward solution, which definitely impacts the performance of our model.

Once the imbalance is dealt with, there are innumerable approaches to predicting an output class. In the table above, the best model (SimpleCart) achieved 93.4% Accuracy prediction. The null model alone (i.e. predicting that all bumps are non-hazardous) yields 93.4% Accuracy. But the Balanced prediction is closer to 50%. Our logistic regression model predicts the test set with around 70 to 75% accuracy, after random under-sampling.

While the methods for calculating the results of these various methods aren't clearly documented in the data set, we can assume that we understand a few of them through inference. Accuracy (Acc.) is the percentage of times our model correctly predicted the class in the test set, and Balanced Accuracy (BAcc.) is the percentage of times our model correctly predicted the class in the balanced data set.

Sensitivity and Specificity are well known, and size is the number of leaves / trees /rules in the algorithm (i.e. does not apply for our model). Considering that, our line would look something like this.

Algorithm	**Acc.**	**BAcc.**	**Acc.0.spec**	**Acc.1.sense**	**Size**
Logistic Regression	93.4	73.4	52.3	89.4	NA

Where FP = false positive, FN = false negative, TP = true positive, and TN = true negative, the following analysis is given:

This is a little disconcerting because the sensitivity (Ratio of TPs to Sum(TP, FN)) is much higher than the above models, and the specificity (Ratio of TNs to Sum(TN+FP)) is much lower than the above models.

When tuning these models, the idea that this specificity is more important than sensitivity makes sense. We want TPs (That is, something was hazardous and it WAS hazardous) is the most important thing. We also want to minimize FNs (that is, something was hazardous, but we didn't predict it as such). So specificity is at a premium. In this light, our model isn't too great.

Likewise, sensitivity should be low because we don't mind false positives (we say it's hazardous and it's not) nearly as much as false negatives. Since FPs are on the denominator, we are okay with sensitivity being kinda low with high false positives. Again, in this case, our model isn't great at predicting the things we want, just good at general accuracy -- it's better than the other models at getting the right answer, just not with the desirable lean. Another method of efficacy is the diagnostic odds ratio:

The diagnostic odds ratio (DOR) is calculated by comparing the following formula:

DOR = $\frac{TP/FN}{FP/TN}$

Another way of stating this is in terms of sensitivity and specificity:

DOR <- (sens * spec) / ((1-sens)*(1-spec))

In that case, our diagnostic odds ratio is 9.30 for this model. In other words, the ratio of the odds that the class is predicted as hazardous with respect to the odds of being non-hazardous, is about 9.30.

References

- Application of rule induction algorithms for analysis of data collected by seismic hazard monitoring systems in coal mines.
- A Study of Rockburst Hazard Evaluation Method in Coal Mine

- Classification: Basic concepts, decision trees and model evaluation
- Our R code is located here