Statistical Assessment of Operational Risks for Induced Seismicity from Hydraulic Fracturing in the Montney, Northeast BC

Geoscience BC Report 2020-12
(Geoscience BC Project 2019-008)

Final Report

Appendix B – Methodology Details

by Scott McKean and Amy Fox

Enlighten Geoscience Ltd.

October 2020

Contents

Magnitude of Completeness 3
Feature Selection 3
Filter-Based Feature Selection 3
Wrapper-Based Feature Selection 3
Models and Hyperparameters 4
Modeling 4
Classification 4
Regression 5
Tuning, Training, and Testing 5
Generalized Linear Model (GLM) 6
Classification and Regression Trees (CART) 6
Multivariate Adaptive Regression Splines (MARS) 6
Extreme Gradient Boosting (XGBoost) 7
Global Model Interpretation 8
Permutation-Based Feature Importance 8
Feature Interaction 8
Partial Dependence Plots & Individual Conditional Expectation Curves 9
Local Model Interpretation 9
Local Interpretable Model-Agnostic Explanations 10
Shapley Additive Explanations 10
References 11

Magnitude of Completeness

Three methods are used to estimate the b-value and magnitude of the dataset: the maximum-likelihood technique (Aki, 1965; Bender, 1983), the maximum curvature method, and the goodness-of-fit method (Wiemer and Wyss, 2000). The Maximum-Likelihood technique determines the b-value using the Bender (1983) approach. A binned histogram is then used to create an empirical cumulative distribution function (ECDF), from which the maximum likelihood magnitude is determined. The b-value is then determined. The maximum curvature method fits a spline to the ECDF and determines the location of the maximum curvature (i.e. the second derivative). The ECDF is then filtered to only include events exceeding the magnitude of completion and a linear model fit to the remaining data.

Feature Selection

Feature selection is essential because of the low number of observations (in the order of 1,500 for classification and 500 for regression) relative to a large number of features (approximately 80). The outcomes of this study are more sensitive to feature selection than any other analysis factor (model type, hyperparameter tuning, and cross-validation strategy for example). For this reason, an extensive feature selection workflow was used, consisting of filter-based and wrapper-based feature selection. The results of a filter-based approach, a sequential wrapper-based approach, and a random wrapper-based approach were aggregated to provide a final feature ranking that was used to select a final feature set for modeling.

Filter-Based Feature Selection

Filter-based feature selection is calculated using the FSelector package in R with two tests: information gain and chi-squared testing. Information gain is a mutual information, or entropy based, approach, meaning that variables that share information will display a high entropy value and have shared information. The chi-square test is a statistical test that determines the dependency of two variables. The chi-square test is only applicable to categorical or nominal data, but it can be extended to continuous variables after binning. Since our study has several categorical and logical features, a binned chi-square evaluation is necessary. The chi-square statistic is calculated between each feature variable and the target variable. In both approaches, high statistic values indicate that a feature and variable are dependent, and by deduction, the feature is important. In both tests, the correlation and variance of the features is used to determine each feature's relative influence on potential model results. This makes filter-based feature selection model agnostic; however, it also means that confounding variables and feature interactions are not generally accounted for.

Wrapper-Based Feature Selection

In an attempt to characterize feature interactions and confounding variables to some degree, wrapper-based feature selection methods use model performance to rank features. Two methods are used for this process with several models.

The first method is a sequential floating backwards search (SFBS), where features are eliminated in a stepwise fashion until the complexity penalized model performance stabilizes. The SFBS process begins

with a model that contains all features. we start with a model built using all features. In each step, the feature decreasing the performance measure the most is removed from the model. The 'floating' portion of the algorithm creates an additional step to add a feature back in randomly after it has been removed in order to sample a larger number of feature combinations. We repeat this process 20 times, since the random nature of each search generates varied results. This produces approximately 50,000 to 100,000 models. We select the features based on the features in the top 1,000 models with the best performance.

The second method is a random search, where we iterate the model 100,000 times and probabilistically select the feature importance based on the top 1,000 models with the best performance.

Models and Hyperparameters

A total of six models with four data/feature sets are used for wrapper-based methods – 12 using all features and 12 using completion only features. BCOGC and WCFD datasets are used with each model. In each case, the algorithm determines the performance after five-fold cross-validated training on the train set. During the feature selection, the default hyperparameters are used to provide a consistent training process and produce a consistent model for feature comparison. The models are summarized below, with a description of their default hyperparameters provided in the modeling section.

Modeling

The following section describes the modeling workflow used in the study. It begins by providing an overview to the seismogenic classification and magnitude regression problems, followed by a description of the models and their default hyperparameters.

Classification

The classification model provides a likelihood of a well being seismogenic. In this study, a seismogenic well is defined by a True or 1 value, whereas a non-seismogenic well is defined by a False or 0 value. The objective of the modelling is to predict as many classes correctly as possible. A perfect model would have no false positives or false negatives and make these predictions with high probabilities. In order to evaluate the performance of classification models, we use the mean of three metrics: the log loss, the F1 score, and mean misclassification error (MMCE). Each metric measures the classification accuracy (predicted vs actual class), quantifying the balance between false positives and false negatives across decision thresholds.

The log loss score is a good metric for the evaluation of problems with uneven class distributions, such as this study. The log loss measures the confidence of predictions in a classification model, adding the probability of a correct prediction versus the probability of an incorrect prediction.

$$LL = -\frac{1}{N}\sum_{i=0}^{N} y_i \, log(p(y_i)) + (1 - y_i) log(1 - p(y_i)))$$

The F1 Score is used as the main performance measure for the reporting the seismogenic classification workflow performance across different thresholds since the log loss takes into account all thresholds to

provide a summary statistic. The F1 Score is a harmonic mean of the precision and recall of a classification problem, which is a factor of the true positive (TP), true negative (TN), false positive (FP), and false negative (FN) rates. It takes into account both false positives and false negatives.

$$F1 = 2\frac{precision \cdot recall}{precision + recall}$$

$$precision = \frac{TP}{TP + FP}$$

$$recall = \frac{TP}{TP + FN}$$

The MMCE, which is also known as the misclassification rate, provides an easier to understand metric, quantifying the percentage of misclassified predictions (FP + FN) against the total number of predictions. It is also equal to unity minus the accuracy of the prediction.

$$MMCE = \frac{FP + FN}{FP + FN + TP + TN}$$

Regression

The regression model predicts the magnitude of the maximum earthquake for a seismogenic well. The average of Root Mean Square Error (RMSE) and Maximum Absolute Error (MAE) is used as a metric for evaluating the model and hyperparameter tuning. These two metrics balance how outliers are handled, since the RMSE (L2 norm) minimizes the contribution of outliers whereas the MAE (L1 norm) maximizes the outlier contribution to the metric. The two metrics provide a bound for the accuracy of model predictions.

$$MAE = \sum_{i=0}^{n} |\, y_i - h(x_i)|$$

$$RMSE = \sqrt{\sum_{i=0}^{n} (\, y_i - h(x_i))^2}$$

Tuning, Training, and Testing

In order to provide a rough idea of how a model will perform on unseen data, we first hold out 10% of the available dataset from training and hyperparameter tuning. This test set is deterministic, meaning that we set a random seed so that the same rows of the dataset are selected across each model run. The holdout sets are inconsistent between classification and regression since the datasets vary. This set provides an estimate of the generalizability of the model but should not be considered a robust estimate of overall model accuracy due to its small size. The performance of each model on unseen future data will depend on numerous factors, namely the similarity of the feature distributions to those used to train the model, and ultimately the variance of the chosen model itself.

The remaining 90% of the dataset is used for model training and hyperparameter tuning. A five-fold cross validation is used for hyperparameter tuning. This helps quantify the out-of-sample accuracy and uncertainty in the model's prediction and allows for balanced hyperparameter tuning across data folds. Bayesian model-based optimization (Bischl et al., 2017) is used to hypertune parameters. A budget of 100 iterations of the optimization algorithm is used. The performance of the hypertuned model is evaluated against the training and holdout set.

The hypertuned model performance is also evaluated against 500 partitions of the entire dataset (training and holdout set), generated using bootstrap resampling. The purpose of resampling evaluation is the quantify the bias and variance of the hypertuned model. The bias is calculated as the mean difference between the predicted and actual value of the target across all partitions. In the case of regression, this is the predicted magnitude versus the observed magnitude. In the case of classification, the bias is 1 in the case of a misclassification and 0 in the case of a correct classification. The variance is calculated as the difference between the prediction and the mean of all predictions for each observation. In the case of binomial classification – it is the mean probability that the predicted label does not match the majority prediction.

Generalized Linear Model (GLM)

Generalized linear models (McCullagh and Nelder, 1989) are an extension of multivariate linear regression that allow for combinations of variables and for variables to have non-normal error distributions. In the case of classification, it also allows for a logistic link function which transforms a continuous linear prediction into a logistic (or sigmoidal) probability function that can be used for binomial classification. There are no hyperparameters in generalized linear models and a gaussian distribution is assumed in this study so that the model can be analytically evaluated. The logistic link function is used to transform the model in a binomial classifier.

Classification and Regression Trees (CART)

Classification and regression trees, which are also known as recursive partitioning and regression trees, make prediction using a rule-based approach. A decision tree is used to recursively split observations based on an impurity metric, or the rule that will split the observations as equally as possible. The significant hyperparameters include:

- minsplit = 20 (The minimum number of samples required to split a node)
- cp = 0.01 (The complexity parameter for minimal cost-complexity pruning)
- maxdepth = 30 (The maximum depth of any node of the final results)

Multivariate Adaptive Regression Splines (MARS)

Multivariate adaptive regression splines (MARS) models (Friedman 1991) create piecewise multivariate linear models. This provides a logical step between highly non-linear models such as tree-based or neural network-based methods and standard multivariate linear regression. In a MARS model, piecewise linear functions are fit between knots in the data. To avoid overfitting, the number of knots is determined through complexity penalized cross-validation. The significant hyperparameters include:

- degree = 1 (The maximum interaction degree)
- penalty = 2 (The cost per degree of freedom added)
- thresh = 0.001 (The forward stepwise stopping threshold)

Extreme Gradient Boosting (XGBoost)

We use an optimized and regularized version of Friedman's gradient boosting machine for the classification and regression problem (Friedman, 2001). This technique, called XGBoost, is popular for a wide range of machine learning problems. A gradient boosted machine is a tree-based ensemble model (similar to a random forest but using classification and regression tree (CART) models, which provide a numerical interpretation of tree decisions). The objective function for the model, which is optimized to determine model parameters is:

$$obj(\theta) = \sum_{i}^{n} l\,(y_i, \hat{y}_i) + \sum_{k=1}^{K} \Omega\,(f_k)$$

where θ is a set of model parameters, y_i are the predicted training data, $\hat{y}_i$ are the labelled training data, l is the training loss term, and Ω is the regularization term of the CART function. The regularization term is important when using tree ensembles to avoid overfitting, since the number of trees often exceeds 1,000, with multiple decisions per tree. Applying a penalty to this complexity is essential for achieving the right bias-variance trade-off during model creation. In XGBoost, the regularization term is defined as:

$$\Omega(f) = \gamma T + 1/2\lambda \sum_{j=1}^{T} w_j^2$$

where T is the number of leaves in the model, w_j^2 is the vector scores on the leaves, λ is the L2 regularization term, and γ is the minimum loss reduction required to make a further partition on a new tree. Increasing the λ and γ values makes the model more conservative in terms of penalizing complexity. Gradient boosted machines are sometimes called additive boosted models because they proceed through the optimization problem by adding CARTs successively. The objective function is minimized with every CART added to the model through gradient descent. This gradient descent function is relatively flexible. In addition to supervised regression problems, it can be applied to logistic regression for binary classification problems. We use it for both of the cases in this study for that reason. The tuning and implementation of the model is relatively complex, but the problems in this study are equally complex and the balance of regularization and complexity offered by XGBoost are therefore attractive.0020There are quite a few default hyperparameters in an XGBoost model, but the most significant ones include:

- nrounds = 100 (the maximum number of iterations for training the model)
- eta = 0.3 (the learning rate of the XGBoost model)
- gamma = 0 (no complexity penalty is applied)

- max_depth = 6 (a maximum tree depth of 6 leaves)
- minchildweight = 1 (minimum trees in a child node to reduce feature interaction)
- subsample = 1 (all samples are available to each tree)
- colsample_bytree = 1 (all features are available to each tree)
- lambda = 0 (no L2 regularization is applied)
- alpha = 1 (L1 regularization is applied)

Global Model Interpretation

The IML package is used to interrogate and interpret the results of the model. We use several techniques including feature interaction (Friedman and Popescu, 2008), partial dependence plots (Friedman, 2001), individual conditional expectation plots (Goldstein et al., 2015), and permutation-based feature importance (Breiman, 2001).

Permutation-Based Feature Importance

Permutation-based feature importance is used to interpret the global importance of each input feature on the target outcome. Feature importance measures the effect of removing (also termed permuting) a feature's information on the model error. The feature importance is quantified by the ratio of the permuted model error to the original model error, with higher importance represented by importance values above unity and the error specified by the model loss metric. The permutation-based feature importance also takes feature interaction, since holding x_c constant while permuting x_s eliminates the interaction between x_s and x_c. Feature importance was originally designed for random forest models (Breiman, 2001) but was extended into a model agnostic framework by (Fisher et al., 2018). It is somewhat analogous to causal inference (Pearl, 2009). It is worth noting that permutation feature importance results are stochastic due to sampling of the features while permuting, and thus the results may vary considerably for each realization and a resampling approach is required. In both PDPs and feature importance, strongly correlated features can both decrease the importance of each feature and create bias. For this reason, it is important to highlight strongly correlated features and eliminate strongly correlated sets when possible. In this study, we compute the feature importance for the training data set and strive to eliminate strongly correlated variables.

Feature Interaction

Strongly interacting features can create issues with statistical models (Hall, 1999). This interaction can be quantified by the amount of model variance that is explained by the interaction of two features (analogous to the covariance of two features). The measure varies between 0 and 1, with 0 representing no interaction and 1 meaning that 100% of the model variance is explained by the interaction between two features. Our objective in quantifying feature interaction is to demonstrate low feature interaction, in support of our feature selection process described above.

Partial Dependence Plots & Individual Conditional Expectation Curves

Partial Dependence Plots (PDPs) compare the change in the average predicted value as specified feature(s) vary over their marginal distribution. This is done by holding all variables constant for each observation in our training data set but then apply the unique values of one feature for each observation. Individual Conditional Expectation (ICE) curves are an extension of PDP plots but, rather than plot the average marginal effect on the response variable, we graph the change in the predicted response variable for each observation as we vary each predictor variable. Below shows the regular ICE curve plot (left) and the centered ICE curves (right). When the curves have a wide range of intercepts and are consequently "stacked" on each other, heterogeneity in the response variable values due to marginal changes in the predictor variable of interest can be difficult to discern. The centred ICE can help draw these inferences out and can highlight any strong heterogeneity in our results.

Partial dependence plots (Friedman, 2001; Molnar et al., 2019) are a global method used to visualize the relationship between the target and a feature. Partial dependence plots (PDPs) estimate the marginal effect of a feature on the predicted outcome of a model through Monte Carlo simulation. Equation 1 describes the 1D PDP estimator, where x_s is the feature of the PDP, x_c are all other features in the model, $\hat{f}$ is the machine learning model, and $dP(x_c)$ represents the marginal distribution of the model output over feature set x_c. This results in a function that depends solely on the feature x_s and its indirect interactions with feature set x_c. In the discrete Monte Carlo portion of Equation 1, x_c^i represents the actual feature value of the dataset. A rug plot is used to show the distribution of feature set x_c for each plot. This technique assumes that each feature x_s is independent from other features in x_c, so can provide misleading interpretations when the features are in fact highly correlated.

$$\hat{f}_{x_s}(x_s) = \int \hat{f}(x_s, x_c) dP(x_c) = \frac{1}{n}\sum_{i=1}^{n} \hat{f}\,(x_s, x_c^i)$$

The ICE curves extend PDPs to investigate heterogeneous effects, or the differing effect of x_s on individual observations due to natural response variability or differing relationships between those observations and x_c. Relative to a PDP, an ICE curve represents the prediction on a feature for each realization of that feature in the model. The value is computed by keeping all other features (x_c^i) constant for a single observation and varying the realization of feature (x_s^i), as opposed to the average relationship described by a PDP. Generally, the plots are centered at an anchor point to better describe the individual relationships and heterogeneous, since individual predictions can vary significantly. It is worth reiterating that ICE curves also assume independent features.

Local Model Interpretation

A selection of large magnitude events is investigated using local interpretability techniques, including local interpretable model-agnostic explanations (LIME, Ribeiro et al., 2016) and SHapley Additive exPlanations (SHAP, Shapley, 1953, Lundberg and Lee, 2017). These techniques provide an explanation of which features contribute to the result of an individual event. Two techniques are employed since the interpretation results can vary considerably, both when comparing LIME and SHAP, and when the local point being investigated is compared with other similar observations (Alvarez-Melis and Jaakkola, 2018).

Local Interpretable Model-Agnostic Explanations

Local interpretable model-agnostic explanations use a local surrogate model to explain individual predictions (Ribeiro 2016). The surrogate model is trained on permutations of the training data and the corresponding predictions of the actual machine learning model, which is treated as a black box. An interpretable model is used for the surrogate such that the influence of each parameter can be quantified relative to the model response. A fidelity measure is used to explain how well the interpretable model approximates the global model.

The IML package is used to implement a modified approach to LIME. A weighted generalized linear model (glm) is fit with the desired observation as the target. A local neighbourhood is used to determine the weights for the glm, using Gower's (1971) proximity measure and L1-regularization. The original dataset is used for resampling (instead of the normal distribution proposed in the original LIME approach). This approach is applicable to both continuous and categorical features.

The visualization for LIME shows the effect of each feature, which is the weight of the feature times the feature value when a linear regression (i.e. glm) is employed.

Shapley Additive Explanations

SHAP computes feature contributions for single predictions with the Shapley value, an approach from cooperative game theory.

The Shapley value is the average marginal contribution (i.e. fair value) for each feature to the difference of the instance's prediction and the mean of the dataset target. These values are determined by using random perturbations of individual features, replacing the actual value with a random value, and determining how the target changes. The possible combinations of features are analyzed, with a weighted average used to determine the final Shapley value.

For example, in a linear model $\hat{f}(x)$,

$$\hat{f}(x) = \beta_0 + \beta_1 x_1 + \ldots + \beta_n x_n$$

The contribution of a feature j on the prediction is,

$$\phi_j(\hat{f}) = \beta_j x_j - E(\beta_j X_j)$$

where $E(\beta_j X_j)$ is the mean effect estimate for feature j, or the difference between the feature effect and the average. For non-linear models, the game-theory approach proposed by Shapley (1953) is used to provide a 'fair payout' of feature effect. That is, a balance of feature effects that is efficient, symmetric, dummy-variable proof, and additive is determined. In practice, the Shapley value is determined through Monte Carlo resampling since it is very computationally intensive (Štrumbelj and Kononenko, 2014). This 'fair payout' makes the SHAP value attractive, since LIME is based on the explanations of a locally retrained model, which provides little to no robust theoretical background, nor the promise of a balanced solution.

SHAP is implemented through a Kernel-based approach (Lundberg and Lee, 2017) or Tree-based approach (Lundberg et al., 2018). The tree-based approach (TreeSHAP) is used in this study because we use an XGBoost model.

References

Aki, K., Maximum likelihood estimate of bin the formula logN=a−bm and its confidence limits. Bulletin of Earthquake Research Institute, Tokyo Univ., 43, 237–238, 1965.

Alvarez-Melis, D. and T. S. Jaakkola, On the robustness of interpretability methods, arXiv preprint arXiv:1806.08049, 2018.

Bender, B., Maximum likelihood estimation of b values for magnitude grouped data, Bulletin of the Seismological Society of America, 73 (3): 831–851, 1983.

Bischl, B., Lang, M., Kotthoff, L., Schiffner, J., Richter, Studerus, E., Casalicchio, G. and Z. M. Jones, mlr: Machine Learning in R, Journal of Machine Learning Research, 17(1), 2016.

Bischl, B., Richter, J., Bossek, J., Horn, D., Thomas, J. and M. Lang, mlrMBO: A modular framework for model-based optimization of expensive black-box functions, arXiv preprint arXiv:1703.03373, 2017.

Breiman, L., Random Forests, Machine Learning, 45(1), 2001.

Fisher, A., Rudin, C. and F. Dominici, All models are wrong but many are useful: Variable importance for black-box, proprietary, or misspecified prediction models, using model class reliance. arXiv preprint arXiv:1801.01489, 2018.

Friedman, J. H., Greedy function approximation: A gradient boosting machine, Annals of Statistics, 2001.

Friedman, J. H. and B. E. Popescu, Predictive learning via rule ensembles, The Annals of Applied Statistics, 2(3), pp.916-954, 2008.

Goldstein, A., Kapelner, A., Bleich, J. and E. Pitkin, Peeking Inside the Black Box: Visualizing Statistical Learning with Plots of Individual Conditional Expectation, Journal of Computational and Graphical Statistics, 24(1), 2015.

Gower, J. C. "A general coefficient of similarity and some of its properties." Biometrics: 857-871, 1971.

Hall, M. A., 1999. Correlation-based feature selection for machine learning.

Lundberg, S. M. and S. I. Lee, A unified approach to interpreting model predictions, In Advances in neural information processing systems (pp. 4765-4774), 2017.

Lundberg, S.M., Erion, G.G. and S. I. Lee, Consistent individualized feature attribution for tree ensembles, arXiv preprint arXiv:1802.03888, 2018.

McCullagh P. and Nelder, J. A. (1989) Generalized Linear Models. London: Chapman and Hall.

Molnar, C., Interpretable machine learning, https://christophm.github.io/interpretable-ml-book, 2019.

Molnar, C., Casalicchio, G. and B. Bischl, Quantifying Interpretability of Arbitrary Machine Learning Models Through Functional Decomposition, arXiv preprint arXiv:1904.03867, 2019.

Pearl, J., Causal Inference in Statistics: An Overview, Statistics Surveys 3, 2009.

Ribeiro, M. T., Singh, S. and C. Guestrin, Why should I trust you?: Explaining the predictions of any classifier, Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2016.

Shapley, L. S., A value for n-person games, Contributions to the Theory of Games, 2(28), 1953.

Štrumbelj, E. and I. Kononenko, Explaining prediction models and individual predictions with feature contributions. Knowledge and information systems, 41(3), pp.647-665, 2014.

Wiemer, S. and M. Wyss, Minimum magnitude of completeness in earthquake catalogs: Examples from Alaska, the western United States, and Japan, Bulletin of the Seismological Society of America, 90(4), pp.859-869, 2000.

