AGH

AKADEMIA GÓRNICZO-HUTNICZA IM. STANISŁAWA STASZICA W KRAKOWIE

Python in Scientific Research

Estimating Stability of Underground Excavations with Python and Machine Learning

Author: *Miguel Jorquera*
Professor: prof. dr hab. inż.Beata Hejmanowska
dr inż.Mariusz Twardowski

Kraków, July 2020

Table of contents

1. Introduction

Hard rock mining is an industry that has as a main goal to extract minerals out of mineral deposits located below the surface. Given the location of the deposit, the extraction process could be classified as either surface mining or underground mining. Given the current depth of the most recent geological deposits discovered and the geotechnical condition of those deposits, it is getting more important to develop easily accessible tools that would allow a better identification of risks in underground mining. The sublevel stopping mining method (SLS) being studied on this research suffers greatly from low stability on the walls of the excavations, which it is the main focus of the investigation.

One of the major concerns in SLS is the collapse of cavities during the exploitation, which could cause harm to workers, damage to equipment and major loss of mineral resources. There are some theoretical methods currently being used in the industry that provides some information on the expected condition of the walls after exploitation (Mathews wall stability graph for example), but this methods are greatly influenced by the experience and criteria used by the engineer, and can be heavily biased by a great array of factors found in each particular mining site. Given the present condition, it may be a great addition to the industry to make use of machine learning algorithms to develop predictive tools that could be feed data from a wide array of mining sites that share similar characteristics, to obtain the expected condition the wall after the extraction of the stope.

A first approach of the predictive tool would be based specifically to the SLS mining method, but with more information available from mines on different regions of the world it would be possible to expand such a tool, and be able to predict the condition of excavations in a more general approach.

2. Theoretical background

The theoretical background necessary to develop the research is from 2 major “Categories”:

- Underground mining
- Machine learning

The specifics topics that need to be understood in order to analyze the steps taken during research are shown in this section.

2.1. Underground mining

Underground mining is the activity that has as its objective to extract minerals from deposits located beneath the surface, where it is necessary to excavate tunnels and cavities underground. During the exploitation the mineral deposits, it is possible to create instabilities in the walls of the excavations. These instabilities can have many different factors such as: low rock quality, high stresses, presence of water and major fault zones in the area, among many others.

There are many different theoretical models that try to estimate the stability of the walls during and after extraction of the mineral, one of the most used in the industry is the Mathews stability graph which is mainly used for the Sublevel open stopping mining method.

2.1.1.Sublevel open stopping

Sublevel Stopping (SLS) is a large-scale open stopping method, which is usually applied too strong ore bodies that require minimal support and are surround by strong rock mass. The mineralization should be fairly regular in shape, and the dip should be enough to allow the broken ore to flow freely.

The SLS method requires the following developments before being able to extract the mineral:

- Transport drift: level below the ore body, this is the level where the blasted mineral will be collected by the hauling equipment.
- Draw point: this is the opening where the hauling equipment will access below the ore body in order to collect the broken ore.
- Drill Access: this are the levels where the drilling equipment will make boreholes for the explosives to be introduced into the ore body in order to break the rock mass.

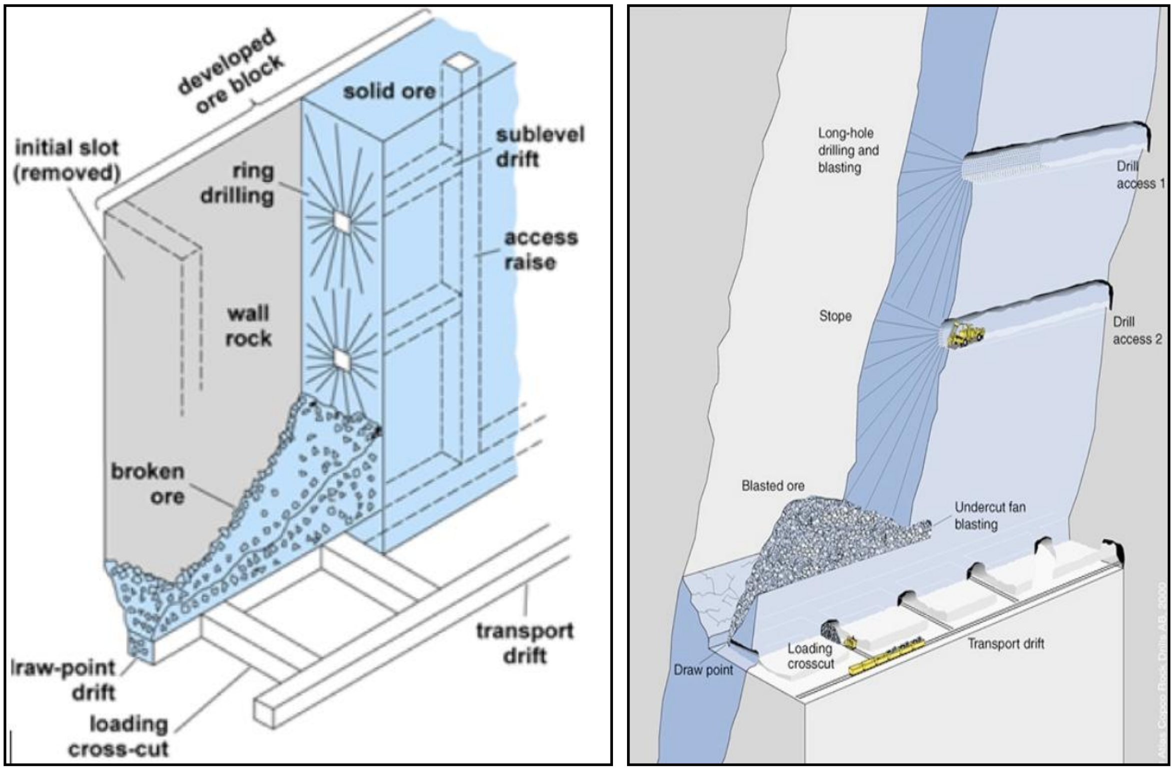


FIGURE 1: SUBLEVEL OPEN STOPPING

The main advantages and disadvantages of the mining method are the following:

Advantages

- Pillars that are left in place can be recovered after using backfilling on adjacent stopes.
- Low blasting costs.
- Development needed for extraction can be done in ore.
- It’s possible to use large scale equipment.
- High productivity and efficiency.
- Little exposure to hazardous conditions.
- Low Dilution (around 20%).
- High recovery (75%-90%).

Disadvantages

- Early production is low.
- Method is not selective.
- Requires high capital expenditure due to early development.
- Inflexible mining plan.
- Drilling requires precision.

2.1.2.Matthews stability graph

This method was developed by Matthews (1980), later modified by Potvin (1989), and in the most recent version, it was actualized by C. Maedesley and R. Trueman (2000). The analysis is based on over 400 historic cases from underground mines located in Australia and Canada. This method allows estimating the probability of failure for a given excavation.

The method is based on the relation between the index N' and the hydraulic radius of the excavation. The two parameters are then located into the graph (Figure 2) and the probability of failure is estimated (probability of caving).

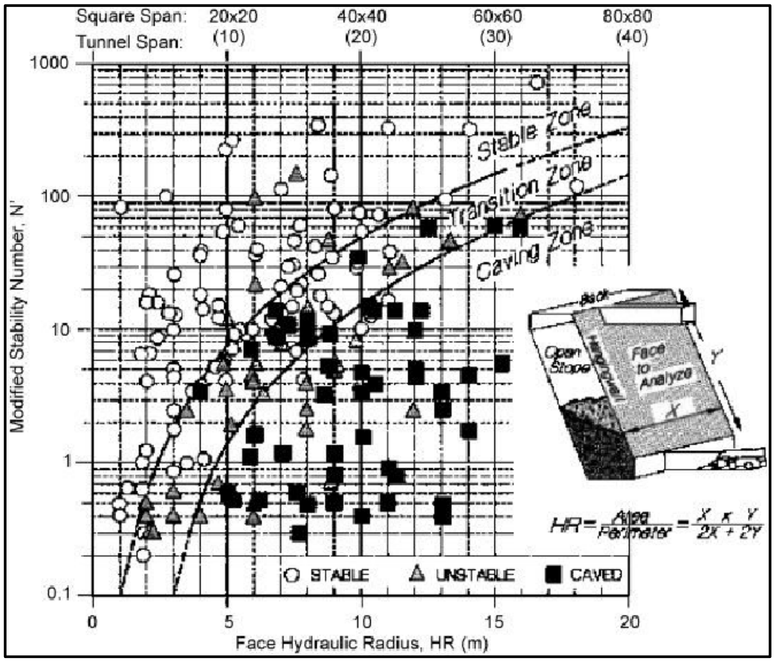


FIGURE 2: MATHEWS STABILITY GRAPH

The main parameters that are used in this method are the following:

- Rh: Hydraulic radius

  The hydraulic radius is a measure of the shape of the wall. It is defined as the cross sectional area of the wall divided by the perimeter.

$$\boxed{Rh = Area / Perimeter}$$

- Q': Rock tunneling quality index

  This is an index that represents the strength of the rock mass, is calculated from the following equation.

$$Q' = (RQD/Jn)x(Jr/Ja)$$

  Where:

  RQD/Jn = is the measure of block size for a jointed rock mass.

  Jr / Ja = is the measure of join surface strength and stiffness.

- A: Condition of stresses factor

  This is a factor that takes into account the ratio between intact rock strength and the induced stress on the wall.

- B: Structural condition of walls factor

  This factor is the measure between the orientation of the walls and the orientation of the dominant joints.

- C: Gravity factor

  This is the measure of how the gravity influences the stability of the wall.

- N': Stability Number

  This is the Matthews stability number, which represents the competency of the rock mass for a given stress condition.

- Expected Damage on the wall

  The expected damage on the wall can be classified on 3 different categories, depending on how likely it is to experience instability: Stable (not likely to become instable), transition zone (walls that can see some form of failure), caving zone (walls that are expected to collapse after or during extraction).

2.2. Machine learning

Machine learning is a form of Artificial Intelligence that enables a system to learn from the available data instead than the usual method (explicit programming). Machine learning makes use of a variety of algorithms that allow the model to learn, improve, describe the data and even predict possible outcomes.

A machine learning model has to be trained before being able to provide a reliable output. The main difficulty for this type of models resides on the training phase, choosing the most adequate classifier and setting the parameters that would allow for the sought output.

The classifiers used during the research are the following:

2.2.1.Random forest (RF)

The random forest classifier consists of a collection of decision tress, where each one of those trees is constructed by applying an algorithm on the training set (Figure 3). Later, the prediction for the output is obtained by majority vote over the predictions of each individual tree created.

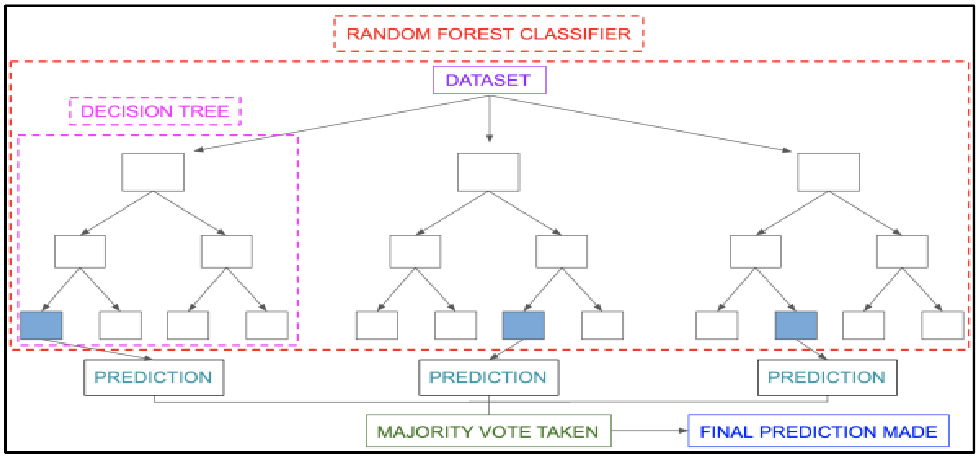


FIGURE 3: RANDOM FOREST ALGORITHM REPRESENTATION

The main parameters used on the research regarding random forest are the following:

- Number of estimators: the number of the trees in the forest.
- Criterion: function to measure the quality of a split. The criteria could either be ‘Gini” or ‘Entropy’.
- Maximum depth: this is the maximum depth of the tree.
- Minimum samples leaf: the minimum number of samples required to be at a leaf node.
- Minimum samples split: minimum number of samples required to split an internal node of the tree.
- Maximum features: the number of features to consider when looking for the best split of the node.

2.2.2.Support vector machines (SVM)

SVM are a set of supervised learning methods that are used for classification, regression or outlier detection. This algorithm looks to find a hyperplane in an N-Dimensional space that is able to classify the data points into different categories (Figure 4). In order to obtain better results on the output, the hyperplane has to have the maximum possible margin with the data points.

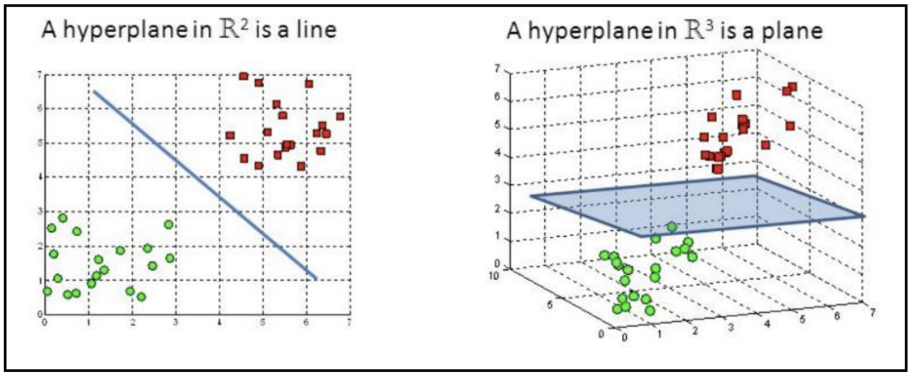


FIGURE 4: SUPPORT VECTOR MACHINE ALGORITHM REPRESENTATION

The main advantages of this classifier are:

- Effective in high dimensional spaces.
- Still effective in cases where number of dimensions is greater than the number of samples.
- Uses a subset of training points in the decision function (called support vectors), so it is also memory efficient.
- Versatile: different Kernel functions can be specified for the decision function. Common kernels are provided, but it is also possible to specify custom kernels.

The disadvantages of support vector machines include:

- If the number of features is much greater than the number of samples, avoid over-fitting in choosing Kernel functions and regularization term is crucial.
- SVMs do not directly provide probability estimates, these are calculated using an expensive five-fold cross-validation (see Scores and probabilities, below).

The main parameters used on the research regarding random forest are the following:

- C: Regularization parameter. The strength of the regularization is inversely proportional to C. Must be strictly positive.
- Kernel: Specifies the kernel type to be used in the algorithm. It must be one of ‘linear’, ‘poly’, ‘rbf’, ‘sigmoid’, ‘precomputed’ or a callable.
- Cache size: this is the size of the kernel cache to be used.
- Degree: depgree of the polynomial kernel. Ignored by other kernels

- Gamma: Kernel coefficient for ‘rbf’, ‘poly’ and ‘sigmoid’
- Ceof 0: Independent term in the kernel function, used in ‘poly and ‘sigmoid’
- Tol: tolerance for stopping criterion.
- Probability: enabling the probability estimates.

2.2.3.Gaussian naïve Bayes (GNB)

Naive Bayes methods are a set of supervised learning algorithms based on applying Bayes’ theorem with the “naive” assumption of conditional independence between every pair of features given the value of the class variable. There are different types of naive Bayes classifiers, mainly differing on the assumptions the make on the distribution of the data.

- Advantages:
  - Works well in multiclass predictions.
  - When the Naïve assumption holds true, the classifier performs better than other model.
  - Needs less data.

- Disadvantages:
  - In case of “Zero Frequency”
  - Assumption of independent predictor.

The main parameters used on the research regarding random forest are the following:

- Variance smoothing: portion of the largest variance of all features that is added to variances for calculation stability.

2.2.4.K nearest neighbors (KNN)

Nearest neighbor algorithms are among the simplest of all machine learning algorithms. The idea is to memorize the training set and then to predict the label of any new instance on the basis of the labels of its closest neighbors in the training set. The rationale behind such a method is based on the assumption that the features that are used to describe the domain points are relevant to their labeling in a way that makes close-by points likely to have the same label. Furthermore, in some situations, even when the training set is immense, finding a nearest neighbor can be done extremely fast (for example, when the training set is the entire Web and distances are based on links).

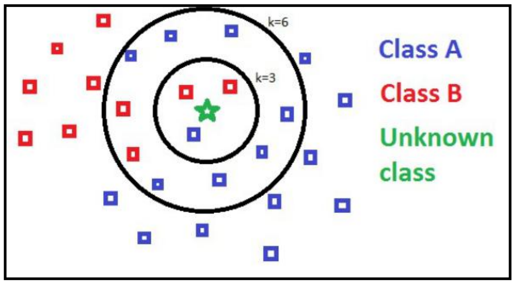


FIGURE 5: REPRESENTATION OF K NEAREST NEIGHBORS

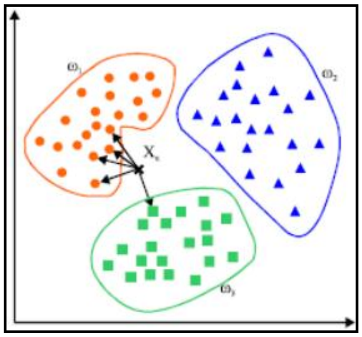


FIGURE 6: K NEAREST NEIGHBORS

- Advantages:
  - Robust to noisy training data.
  - Effective if training data is large.
  - Learns complex models easily.

- Disadvantages:
  - Need to determine value of parameter K.
  - Hard to apply on high dimensional data.
  - Not clear which type of distance metric to use.
  - High computation cost.

The main parameters used on the research regarding random forest are the following:

- Number of neighbors: number of neighboring samples to use for queries.
- Weights: Weight function used in the prediction.
- Algorithm: algorithm used to compute nearest neighbors.
- Leaf size: leaf size passed to BallTree or KDTree algorithms. Can affect speed of computation and memory required.
- P: parameter for the Minkowski metric.
- Metric: the distance metric to be used for the tree.

3. Data

The data used in the research was collected both directly from mining companies located in Chile and by analyzing previous studies done in underground mines in Australia and Canada. The summary of the collected data can be found in Table 1 (detailed information on the database can be found in the Annex I).

TABLE 1: SUMMARY OF COLLECTED DATA

Source	Number of Cases	Percentage
Mine Alcaparrosa - Chile	41	6%
Mine Candelaria – Chile	103	16%
Mine Santos – Chile	23	4%
Previous Studies	465	74%

The data contains 647 sets, and each set describes the main stability characteristics for Open Cavities. The main features to be analyzed in the research are the following:

- Rh: Hydraulic radius
- Q': Rock tunneling quality index
- A: Condition of stresses factor
- B: Structural condition of walls factor
- C: Gravity factor
- N': Stability Number
- Expected Percentage of Failure
- Expected Damage on the wall
- Observed wall condition

Taking into account that the objective of the research is to estimate the final condition of the cavity after extraction, it's is possible to make the following classification on the database (see Table 2):

TABLE 2: CATEGORIZATION OF DATA ACCORDING TO WALL CONDITION

Wall Condition	Number of Cases	Percentage
Stable	339	52%
Failure	211	33%
Major Failure	97	15%

The database is divided in two major categories, Training Data and Testing Data. The training data are the data sets that are going to be used in order to prepare the machine learning algorithm, adjusting the parameters of the classifiers in order to obtain a better fit to the data. On the other hand, testing data is not seen by the classifiers until the final step of the research, where the data is going to be used in order to determine how accurate the algorithm is to predict the expected results on new data. The details on the training and testing data can be found in Table 3:

TABLE 3: SUMMARY OF TRAINING AND TESTING DATA

Data	Wall Condition	Number of Cases	Total number of cases
Training Data	Stable	295	547
	Failure	173	
	Major Failure	79	
Testing Data	Stable	44	100
	Failure	38	
	Major Failure	18	

4. Methodology

This section will present the methodology used during the research, describing each one of the steps taken to achieve the desired result. The methods used are from statistical analysis and machine learning.

4.1. Import and preparation of data

Before any other step, it is necessary to import the collected data from the Excel file into arrays in order to be able to work with them on the next steps. The data is in an excel file, and the training data (547 data sets) and the test data (100 data sets) are in different worksheets. The data sets will be read from the file by making use of the function 'read_excel( )' from the Panda package. The complete process is as follows:

- Read Training data from the Excel Database: pd.read_excel( ).
- Replace condition values (String) to numerical values (Int): {'STABLE': 1, 'FAILURE': 2, 'MAJOR FAILURE': 3}.
- Replace damage values (String) to numerical values (Int): {'STABLE': 1, 'VERY LOW': 2, 'LOW': 3, 'HIGH': 4, 'VERY HIGH': 4}.
- Drop the unused columns: 'Mine' (Name of the mine), 'Name' (Name of cavity), 'Method' (Mining Method), 'Wall' (Type of Wall), 'Origin' (Source of the data).
- The steps are repeated for the testing data.
- The last step on preparation of data is to shuffle the training and testing datasets. This is done because some of the classifiers.

4.2. Correlation matrix

The first step on data analysis is to determine if there is high correlation between any of the features to be used in the research. This will be done by making use of the correlation matrix, where each cell on the table shows the correlation between two variables.

The correlation matrix is obtained from the data by using 'dataframe.corr( )' from the 'Pandas' package.

4.1. Feature importance

The second step on the study is to determine if the features to be used have enough importance in order to be relevant for the prediction algorithm. To determine the importance of the features, the data will be fitted to the Random Forest classifier (Sickit-Learning package) and then, the ".feature_importance_" parameter will be used.

The results are returned as an array, where each parameter is the importance of the corresponding feature (in percentage) and they add up to 1 (100%). Features with an importance lower than 5% can be considered as not relevant, and dropped from the study.

4.2. Machine learning estimation

After having analyzed the database and made sure that the data sets have the necessary features, the implementation of the machine learning algorithm can begin. This step starts with the shuffling of the training data; this is done in order to avoid any kind of bias on the results that could be produced by the order of the database.

4.2.1.Preliminary classifier scores

This step begins by using cross validation for classifiers of interest in order to obtain some preliminary results on their accuracy, precision and the correlation matrix. The cross validation is done by making use of the ‘cross_val_score’ function (from the ‘sklearn.model_selection’ package). The classifiers and some of the parameters to be used on this step are the following:

- **Random forest:**
  This classifier is part of the ‘sklearn.ensemble’ package, it is initiated with a random_state equal to zero, and the cross validation splitting strategy will be 10-fold.

- **Gaussian naïve Bayes:**
  This classifier is part of the ‘sklearn.naive_bayes’ package, it is initiated without any specific parameters, and the cross validation splitting strategy will be 10-fold.

- **Support vector machines:**
  This classifier is part of the ‘sklearn.svm’ package, it is initiated without any specific parameters, and the cross validation splitting strategy will be 10-fold.

- **K nearest neighbors:**
  This classifier is part of the ‘sklearn.neighbors’ package, it is initiated without any specific parameters, and the cross validation splitting strategy will be 10-fold.

After initialization and fitting the classifiers to the training data set, the accuracy, precision and correlation matrix are obtained by using the following functions:

- Cross validation score: ‘cross_val_score’ (from the ‘sklearn.model_selection’ package).
- Accuracy Score: ‘accuracy_score’ (from sklearn.metrics package).

- Precision score: ‘classification_report’ (from sklearn.metrics package).
- Confusion matrix: ‘plot_confusion_matrix’ (from sklearn.metrics package).

This step is repeated 50 times. The results for the Cross validation score and Accuracy Score are saved, and then some basic statistical analysis is performed (Average, Maximum value and Variance) for each one of the classifiers.

4.2.2.Determination of classifier scores and hyper parameter tuning

After having obtained preliminary results for accuracy, precision and the correlation matrix, it is necessary to improve the results obtained by tuning the hyper parameters of the classifiers. The tuning will be done by using the BayesSearchCV function (from ‘skopt’ package). This function uses Bayes optimization to model the search space and obtain good parameters for the classifiers as soon as possible. The parameters to be used in this step for each classifier are the following (see Table 4 to Table 7):

TABLE 4: RANDOM FOREST - HYPER PARAMETER TUNING PARAMETERS

Parameter	Values
Number of estimators	[100, 150, 200]
Criterion	[gini, entropy]
Max depth	[5, 10, 20]
Min samples leaf	[2, 5, 10]
Min samples split	[10, 15, 20]
Max features	[2, 3, 4, 5, 7]
Cross validation splitting	5-fold

TABLE 5: GAUSSIAN NAIVE-BAYES - PARAMETER TUNING PARAMETERS

Parameter	Values
Var smoothing	[1e-11, 1e-10, 1e-9, 1e-8, 1e-7, 1e-6, 1e-5, 1e-3, 1e-2, 1e-1]
Cross validation splitting	5-fold

TABLE 6: SVM - PARAMETER TUNING PARAMETERS

Parameter	Values
C	np.arange(1, 52, 10)
Kernel	['sigmoid', 'rbf']
Cache size	[200, 300, 400]
Degree	np.arange(3, 8)
Gamma	['scale', 'auto']
Coef 0	np.arange(0.001, 10, 0.5)
Tol	[1e-1, 1e-3, 1e-5]
Probability	[True, False]
Cross validation splitting	5-fold

TABLE 7: KNN - PARAMETER TUNING PARAMETERS

Parameter	Values
N neighbors	[2, 5, 7, 10]
Weights	['uniform', 'distance']
Algorithm	['ball_tree', 'kd_tree', 'brute']
Leaf Size	[20, 30, 40]
P	[1, 2, 3, 4]
Metric	'metric': ['euclidean', 'manhattan', 'minkowski']
Cross validation splitting	5-fold

After performing the hyper parameter tuning and fitting the classifiers to the training data set, the scores, best parameters and correlation matrix are obtained by using the following functions:

- BayesSearchCV best scores: ‘best_scores_’ (from ‘skopt’ package).
- BayesSearchCV best parameters: ‘best_params_’ (from ‘skopt’ package).
- Accuracy Score: ‘accuracy_score’ (from sklearn.metrics package).
- Precision score: ‘classification_report’ (from sklearn.metrics package).
- Confusion matrix: ‘plot_confusion_matrix’ (from sklearn.metrics package).

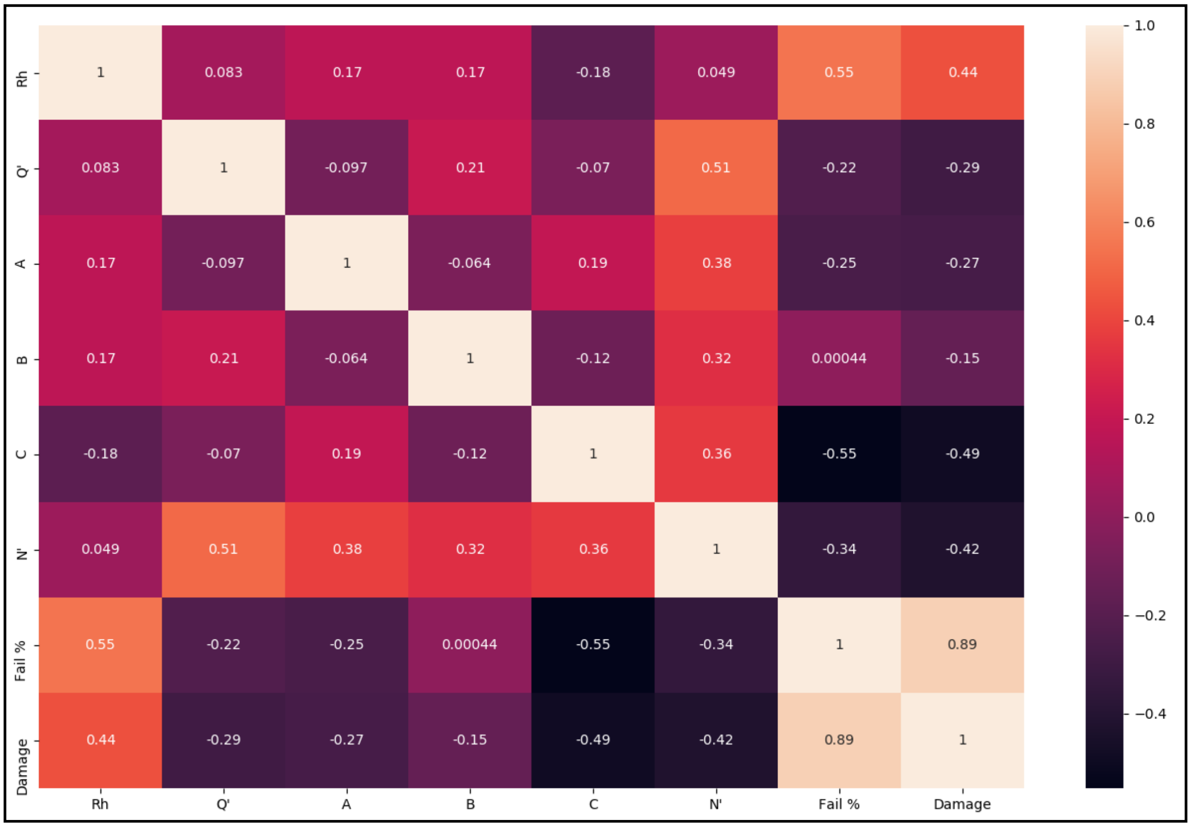
The hyper parameter tuning step is repeated 50 times. The results for the best score and best parameters on the BayesSearchCV are saved, and then some basic statistical analysis is performed (Average, Maximum value and Variance) for each one of the classifiers.

5. Results

The results obtained during the research and a brief discussion on the values can be found in this section.

5.1. Correlation matrix

By making use of the correlation matrix function of the Pandas package, the correlation between the analyzed features is obtained (Figure 7).



	Rh	Q'	A	B	C	N'	Fail %	Damage
Rh	1	0.083	0.17	0.17	-0.18	0.049	0.55	0.44
Q'	0.083	1	-0.097	0.21	-0.07	0.51	-0.22	-0.29
A	0.17	-0.097	1	-0.064	0.19	0.38	-0.25	-0.27
B	0.17	0.21	-0.064	1	-0.12	0.32	0.00044	-0.15
C	-0.18	-0.07	0.19	-0.12	1	0.36	-0.55	-0.49
N'	0.049	0.51	0.38	0.32	0.36	1	-0.34	-0.42
Fail %	0.55	-0.22	-0.25	0.00044	-0.55	-0.34	1	0.89
Damage	0.44	-0.29	-0.27	-0.15	-0.49	-0.42	0.89	1

FIGURE 7: CORRELATION MATRIX

From the obtained matrix, it is possible to observe that two of the features (wall damage and fail percentage) have a high correlation between them (0.89). This means that one of the features should be removed from the next steps, it is decided that wall damage will be eliminated.

5.2. Feature importance

As previously mentioned on the last section, the next result obtained is the importance of the features analyzed. The results are shown in the following table (Table 8), where the total sum of the features must be equal to one.

TABLE 8: IMPORTANCE OF FEATURES

Feature	Relative Importance
Rh	0.155
Q'	0.120
A	0.102
B	0.057
C	0.087
N'	0.154
Fail %	0.323

From the table, is it possible to see that all the features have a relative importance over 5%. According to expertise criteria, all of the features have enough importance on the research.

5.3. Preliminary classifiers scores

The following step after having the data prepared is to obtain the preliminary classifiers scores:

- Cross validation scores on the training data.
- Precision score on the testing data.

The results are obtained from 50 tests on each one of the classifiers (with data being shuffle before each test). The main results obtained are shown in the following table (Detailed results can be found in Annex II).

TABLE 9: SUMMARY OF PRELIMINARY CLASSIFIERS SCORES

	Cross Validation Score			Precision Score		
Algorithm	**Average**	**Max**	**Std Dev**	**Average**	**Max**	**Std Dev**
RF	0.8485	0.8648	0.0061	0.8232	0.8400	0.0084
GNB	0.6989	0.7093	0.0061	0.6500	0.6500	0.0000
SVM	0.7074	0.7166	0.0042	0.6200	0.6200	0.0000
KNN	0.7780	0.7897	0.0055	0.7900	0.7900	0.0000

The confusion matrixes that show the highest accuracy obtained for each classifier are shown in the following figures:

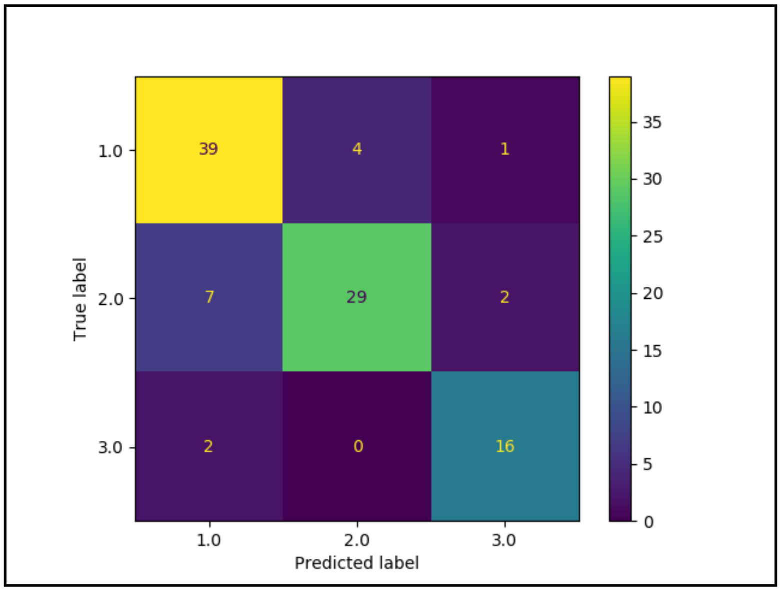


FIGURE 8: RANDOM FOREST CONFUSION MATRIX

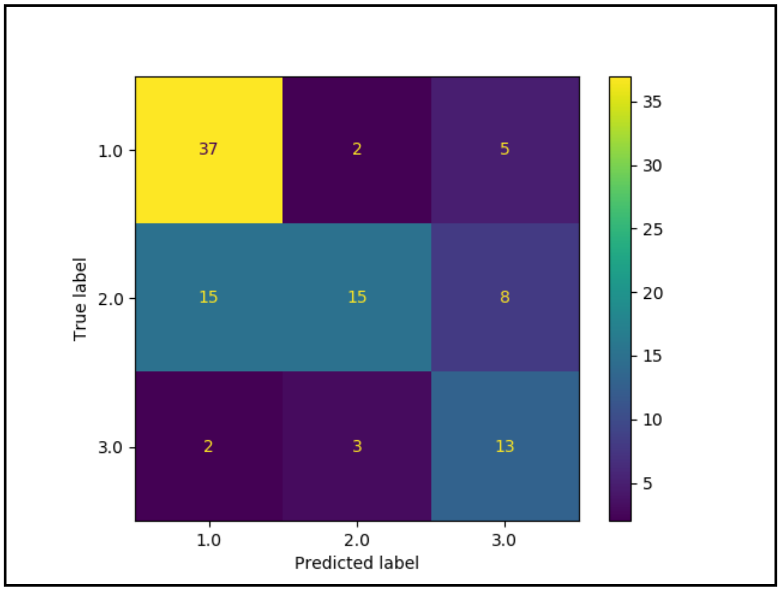


FIGURE 9: GAUSSIAN NAIVE BAYES CONFUSION MATRIX

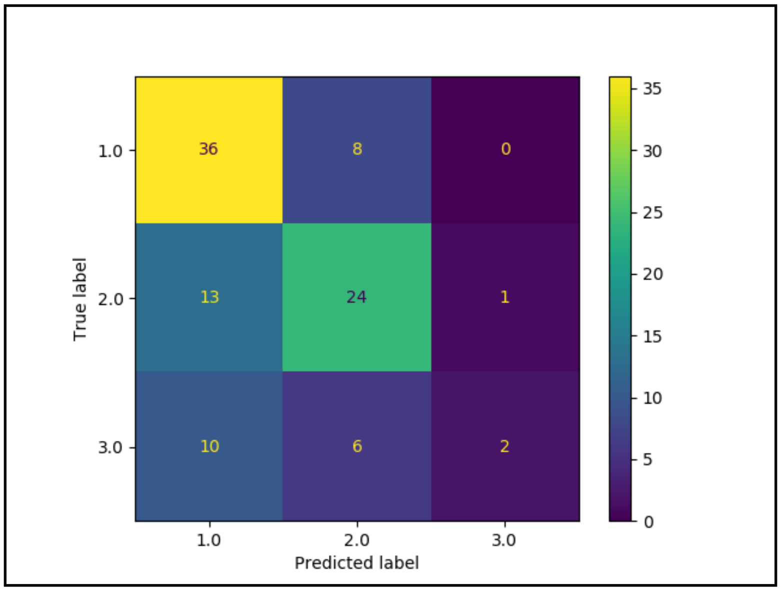


FIGURE 10: SUPPORT VECTOR MACHINE CONFUSION MATRIX

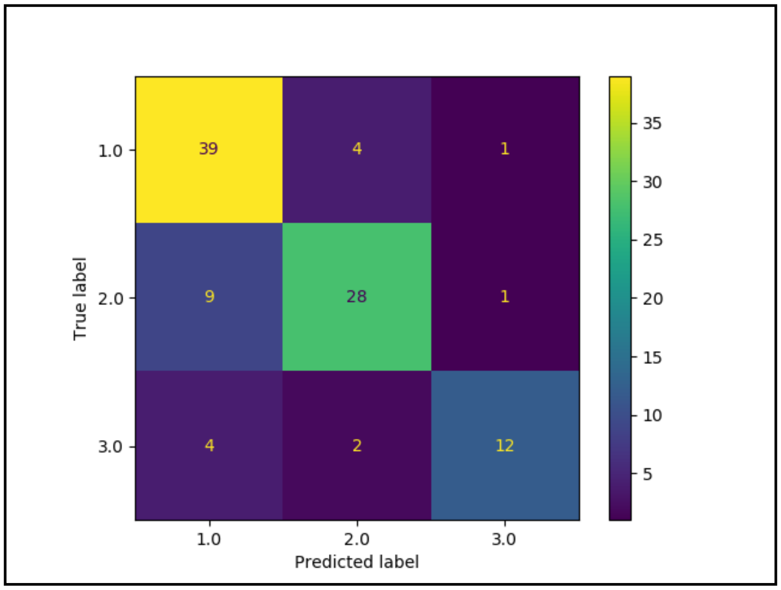


FIGURE 11: K NEAREST NEIGHBORS CONFUSION MATRIX

From the preliminary results, it is seen that the random forest algorithm is giving a better results in cross validation score and precision score, with the K Nearest Neighbors coming second.

5.4. Classifiers scores and hyper parameter tuning

After checking the preliminary results and having an idea on how the classifiers rank among themselves it is necessary to do hyper parameter tuning and then scoring the classifiers once again. This is done by using the BayesSearchCV function. The results obtained are shown in the following table (Table 10):

TABLE 10: SUMMARY OF CLASSIFIERS SCORES

	Cross Validation Score			**Precision Score**		
Algorithm	**Average**	**Max**	**Std Dev**	**Average**	**Max**	**Std Dev**
RF	0.8624	0.8720	0.0048	0.8268	0.8400	0.0077
GNB	0.7347	0.7459	0.0056	0.7200	0.7200	0.0000
SVM	0.7908	0.8062	0.0106	0.7382	0.7600	0.0052
KNN	0.8112	0.8227	0.0066	0.8000	0.8100	0.0064

After checking the results, it is necessary to determine which hyper parameters provided the best performance for each of the classifiers. The hyper parameters will be shown for the best 3 scores on each classifier on Table 11 (detailed results can be found in Annex III).

TABLE 11: HYPER PARAMETERS FOR BEST SCORES

Algorithm	Test	BayesSearch CV Score	Precision Score	Hyper Parameters
RF	9	0.8720	0.84	('criterion', 'entropy'), ('max_depth', 20), ('max_features', 7), ('min_samples_leaf', 2), ('min_samples_split', 10), ('n_estimators', 100)
RF	24	0.8647	0.84	('criterion', 'gini'), ('max_depth', 20), ('max_features', 4), ('min_samples_leaf', 2), ('min_samples_split', 10), ('n_estimators', 200)
RF	30	0.8647	0.84	('criterion', 'entropy'), ('max_depth', 20), ('max_features', 4), ('min_samples_leaf', 2), ('min_samples_split', 10), ('n_estimators', 150)
GNB	8	0.7458	0.72	('var_smoothing', 1e-05)
GNB	3	0.7440	0.72	('var_smoothing', 1e-05)
GNB	42	0.7440	0.72	('var_smoothing', 1e-05)
SVM	10	0.7367	0.76	('C', 11), ('cache_size', 400), ('coef0', 2.501), ('degree', 4), ('gamma', 'auto'), ('kernel', 'rbf'), ('probability', True), ('tol', 0.1)
SVM	45	0.8062	0.74	('C', 51), ('cache_size', 300), ('coef0', 4.001), ('degree', 7), ('gamma', 'scale'), ('kernel', 'rbf'), ('probability', True), ('tol', 0.001)
SVM	34	0.8025	0.74	('C', 51), ('cache_size', 200), ('coef0', 9.001), ('degree', 6), ('gamma', 'scale'), ('kernel', 'rbf'), ('probability', True), ('tol', 1e-05)
KNN	19	0.8062	0.81	('algorithm', 'ball_tree'), ('leaf_size', 20), ('metric', 'manhattan'), ('n_neighbors', 7), ('p', 2), ('weights', 'distance')
KNN	21	0.8043	0.81	('algorithm', 'kd_tree'), ('leaf_size', 20), ('metric', 'manhattan'), ('n_neighbors', 7), ('p', 2), ('weights', 'distance')
KNN	27	0.8226	0.8	('algorithm', 'ball_tree'), ('leaf_size', 30), ('metric', 'manhattan'), ('n_neighbors', 10), ('p', 3), ('weights', 'distance')

The confusion matrixes obtained for each of the shown tests are the following:

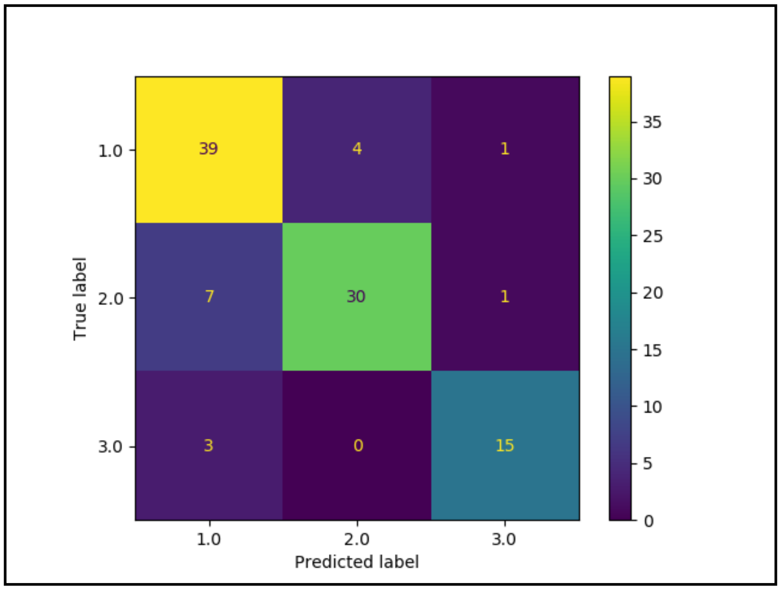


FIGURE 12: RANDOM FOREST CONFUSION MATRIX – TEST 09

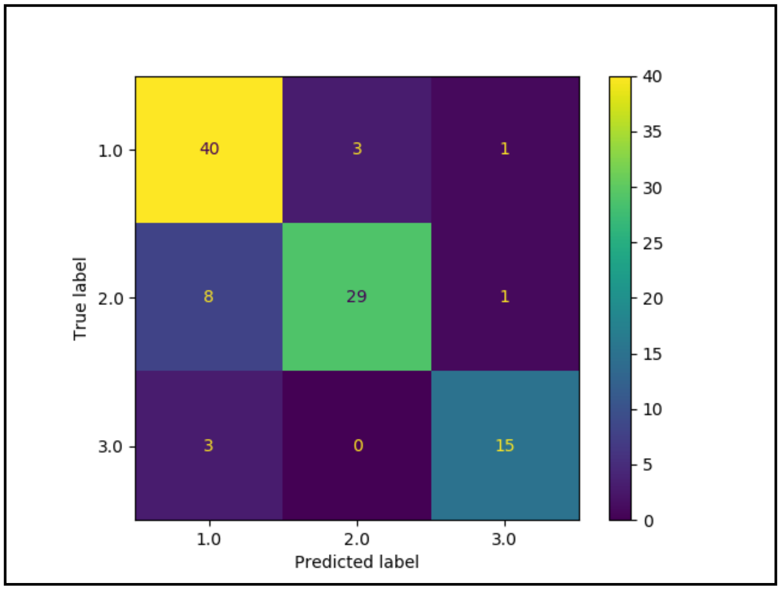


FIGURE 13: RANDOM FOREST CONFUSION MATRIX – TEST 24

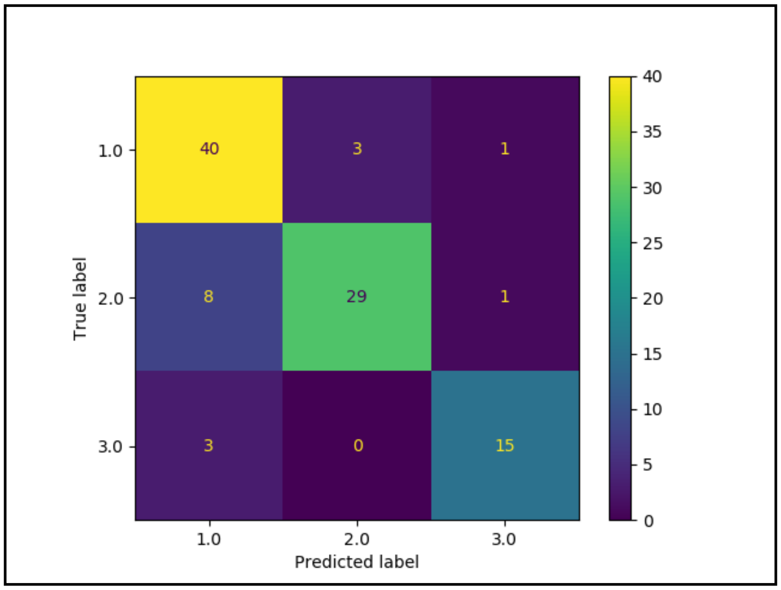


FIGURE 14: RANDOM FOREST CONFUSION MATRIX – TEST 30

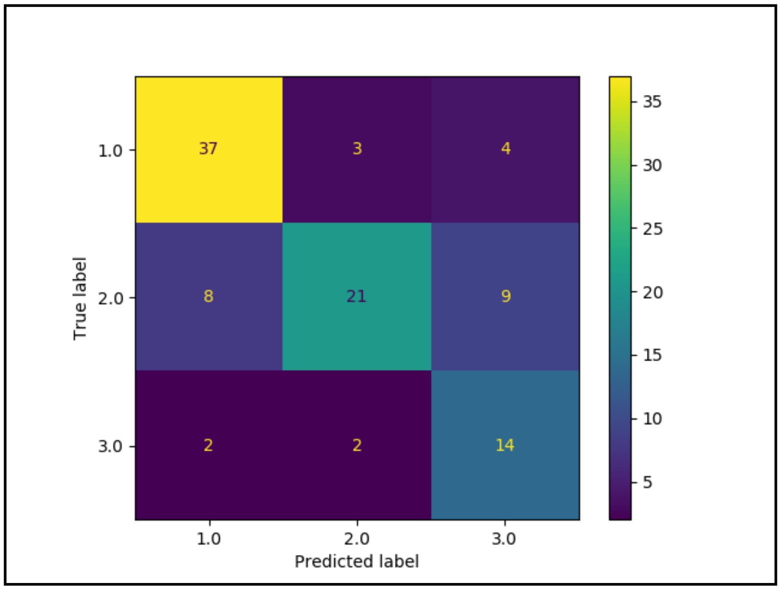


FIGURE 15: GAUSSIAN NAIVE BAYES CONFUSION MATRIX – TEST 08

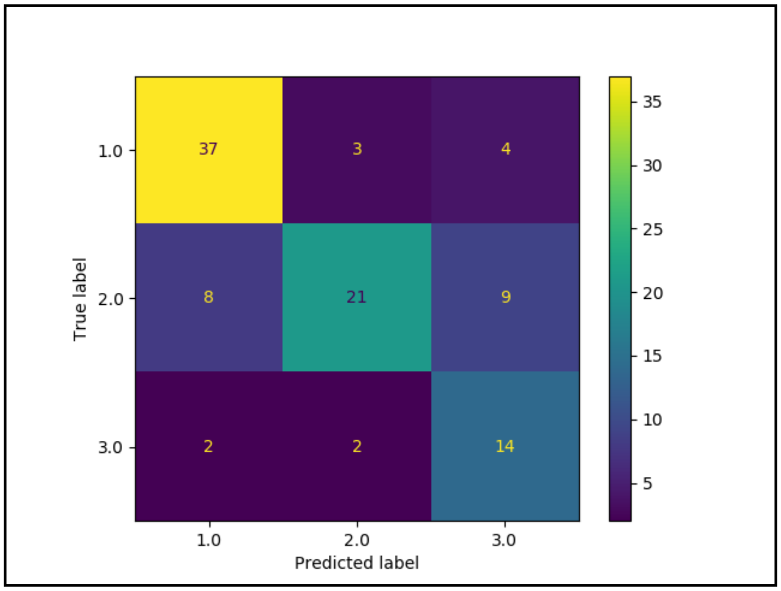


FIGURE 16: GAUSSIAN NAIVE BAYES CONFUSION MATRIX – TEST 03

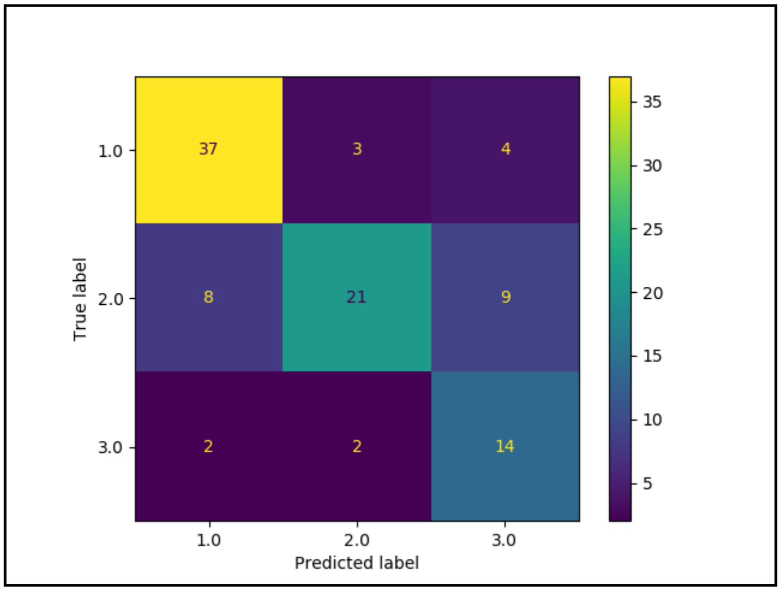


FIGURE 17: GAUSSIAN NAIVE BAYES CONFUSION MATRIX – TEST 42

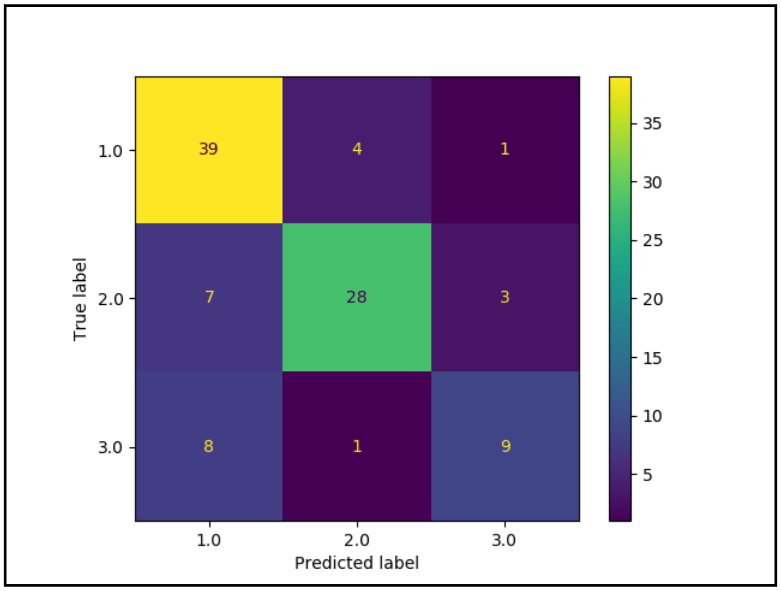


FIGURE 18: SVM CONFUSION MATRIX – TEST 10

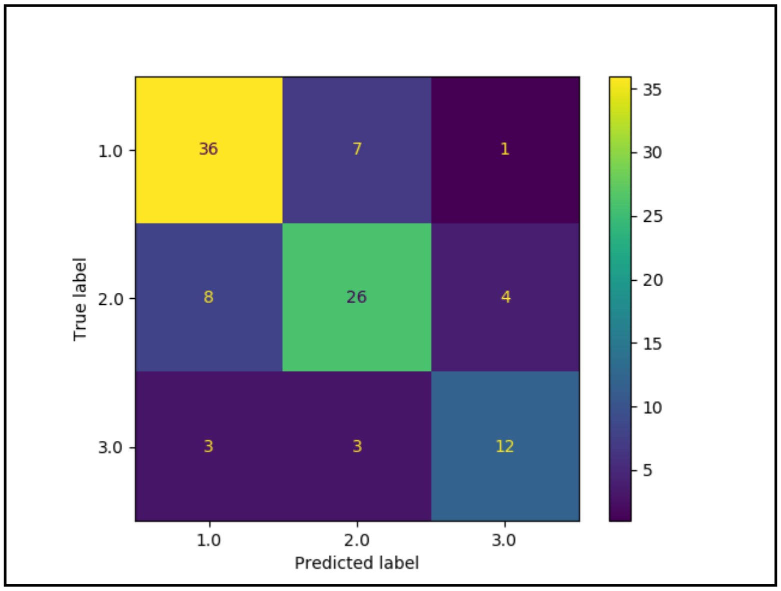


FIGURE 19: SVM CONFUSION MATRIX – TEST 45

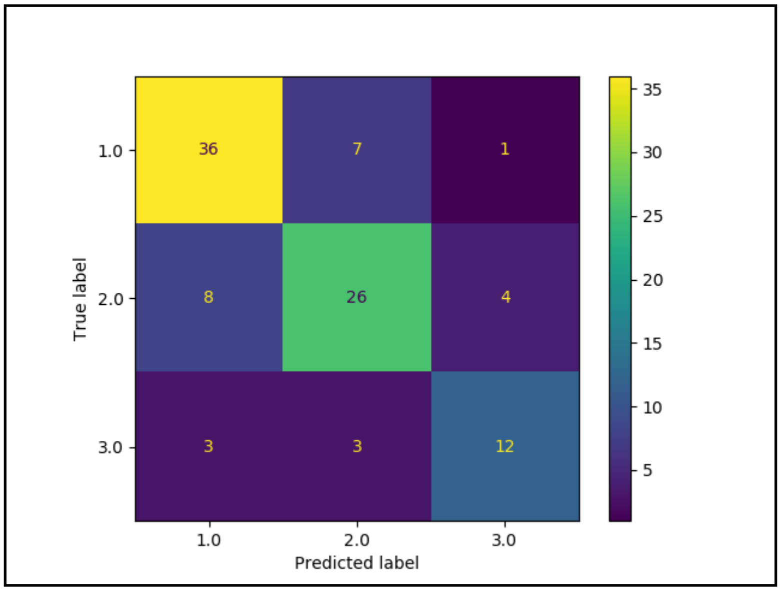


FIGURE 20: SVM CONFUSION MATRIX – TEST 34

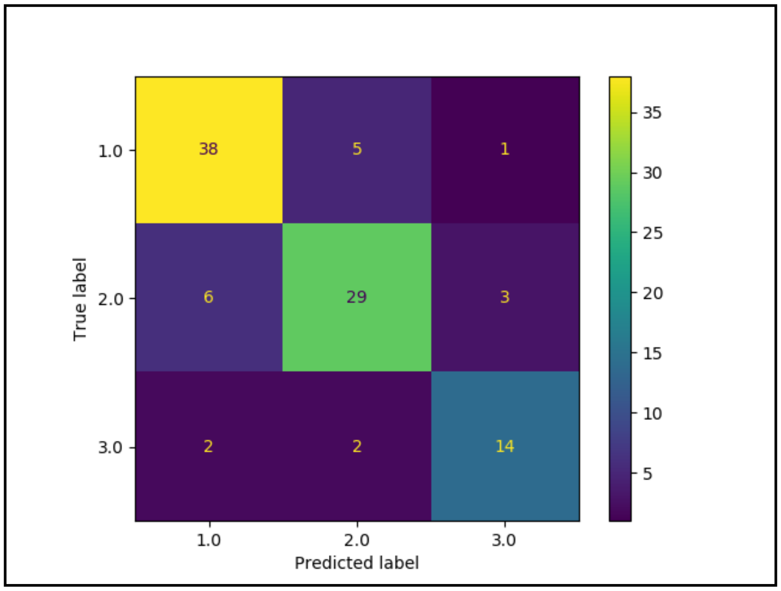


FIGURE 21: KNN CONFUSION MATRIX – TEST 19

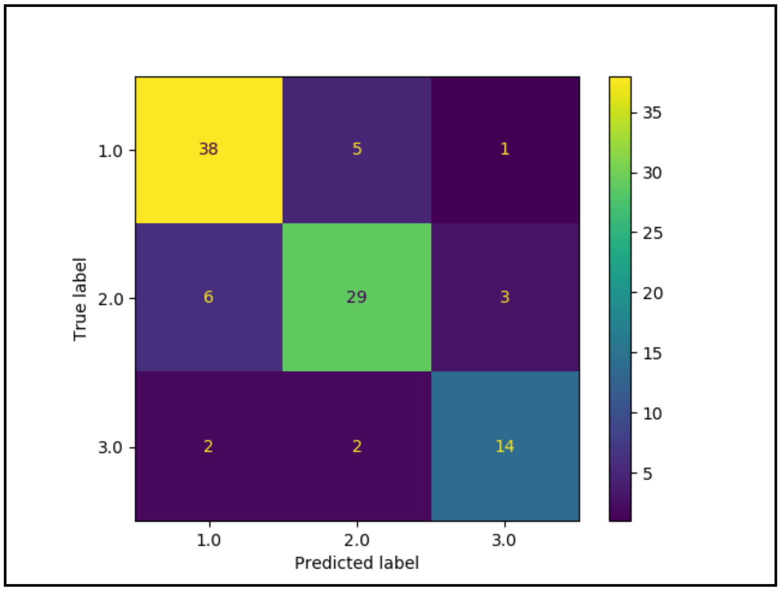


FIGURE 22: KNN CONFUSION MATRIX – TEST 19

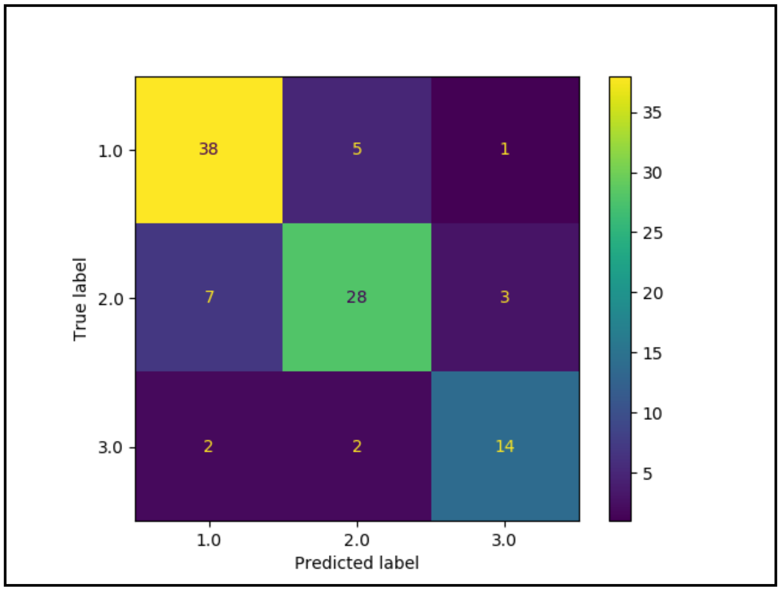


FIGURE 23: KNN CONFUSION MATRIX – TEST 27

5.5. Results discussion

The results obtained after both testing phases of the experiment show that the random forest classifier has a higher cross validation score and is the most precise among all the classifiers tested. After hyper parameter tuning, random forest had a cross validation score of 0.87 and precision of 0.84. The second best performing classifier was the KNN classifier with a cross validation score of 0.82 and a precision of 0.81. The rest of the classifiers weren't able to obtain over 0.8 in both of the analyzed scoring categories which makes them unfit to be utilized in following stages of the research.

It is important to take extra care into the analysis of the confusion matrix, and see where the highest amount of error occurs. Given that the tool is going to be used in the designing stage of stopes, and the high impact that wall failure can cause (accidents to personnel, damage to equipment and loss of resources) it is important to reduce as much as possible the false positives (cases that are flagged as stable when in reality they are failures). When we looked at the obtained results, it is possible to see that both KNN and RF showed the best results among all the classifiers (between 10 and 11 false positives).

6. Conclusion

The results obtained from the different machine learning classifiers let us see from pretty interesting behaviors in the analyzed data, for example we can see that the SVM classifier experienced the biggest improvement on the cross validation score after performing hyper-parameter tuning (about 0.9), on the other hand the best performing algorithm (Random Forest) didn’t see much of an increase on the scores after hyperparameter tuning (0.1 – 0.2). This information could let us conclude that the default parameters of some of the classifiers are better suited for some type of data (distribution, amount of data sets and number of parameters).

When we start to look into the precision results, we can see a similar trend, where the worst performing classifiers on the preliminary testing experienced a bigger improvement on their scores after the tuning.

Another interesting aspect of the study was the data shuffling, and how it affected the results for the different classifiers. We can see from the results that some of the classifiers didn’t see much variance on the distribution of the results (precision score for GNB), while other classifiers experienced really high variance (Cross validation score for SVM). This difference can be attributed to how each classifier predicts the results, which on itself is a really interesting area of study.

It is important to discuss how the machine learning classifiers compare to the theoretical methods currently being used in the mining industry. In my experience, the Mathews stability graph can be trusted to have up to 80% - 85% precision when it is used with an specific mine site. This is pretty close to the accuracy obtained with the highest performing classifiers (Random Forest and KNN). Taking into account that the machine learning algorithm being used in this research is just a first approach to the problem, and that it could be adjusted for a specific mine, it may be possible to increase the precision of the machine learning algorithm and outperform the current used theoretical method by making use of local data for an specific mine.

But a better result of the algorithms is not everything that needs to be taken into account. It is important to analyze the cost associated with developing such a tool (both time and monetary cost), and it is also primordial to consider the stigma that the machine learning algorithms have in mining. Most of the mining sites are reluctant to try out such a solution, because the theoretical methods are able to provide a good enough answer to the current problem.

In order to improve the success of this approach to analyze the stability of cavities, it is necessary to get in contact with different mine sites, and to test the results obtained with some of their excavations so that the industry as a whole would be able to trust this type of solution to the stability analysis issue.

7. Bibliography

1. **Brown, B. H. G. BradyE. T.** *Rock Mechanics for Underground Mining.* s.l. : Springer Dordrecht, 1985.

2. *Extending the Mathews stability graph for open-stope design.* **C. Mawdesley, R. Trueman and W. J. Whiten.** s.l. : Transactions of the Institution of Mining and Metallurgy, Section A: Mining Technology, 2001, Vol. 110.

3. *Applicability of the Mathews Stability Method to Open Stope Stability Assessment at Olympic Dam Mine.* **Sharp, Jacqueline Emily U'Ren.** s.l. : University of Canterbury, 2011. Thesis for Master of Science in Engineering Geology.

4. **Ben-David, Shai Shalev-Shwartz and Shai.** *Understanding Machine Learning: From Theory to Algorithms.* s.l. : Cambridge University Press, 2014.

5. *Evaluacion Estadisitica de los Factore de Ajuste de la Metodologia de Mathews y su Impacto en el Diseño Empirico de Caserones.* **Navarrete, Jose Ignacio Fuenzalida.** s.l. : Universidad de Chile, 2014. Thesis for Mining Engineering Degree.

6. *An Overview of Opportunities for Machine Learning Methods in Underground Rock Engineering Design.* **Josephine Morgenroth, Usman T. Khan and Matthew A. Perras.** s.l. : Geosciences, 2019, Vol. 9. 504.

7. **Scikit-Learn.** scikit-learn - Machine Learning in Python. *Scikit-Learn.* [En línea] Sckit-Learn Consortium, July 2020. https://scikit-learn.org/stable/.