Capstone Project

Coal Prediction

Leah Pettigrew
DATA SCIENTIST

Table of Contents

Problem Statement

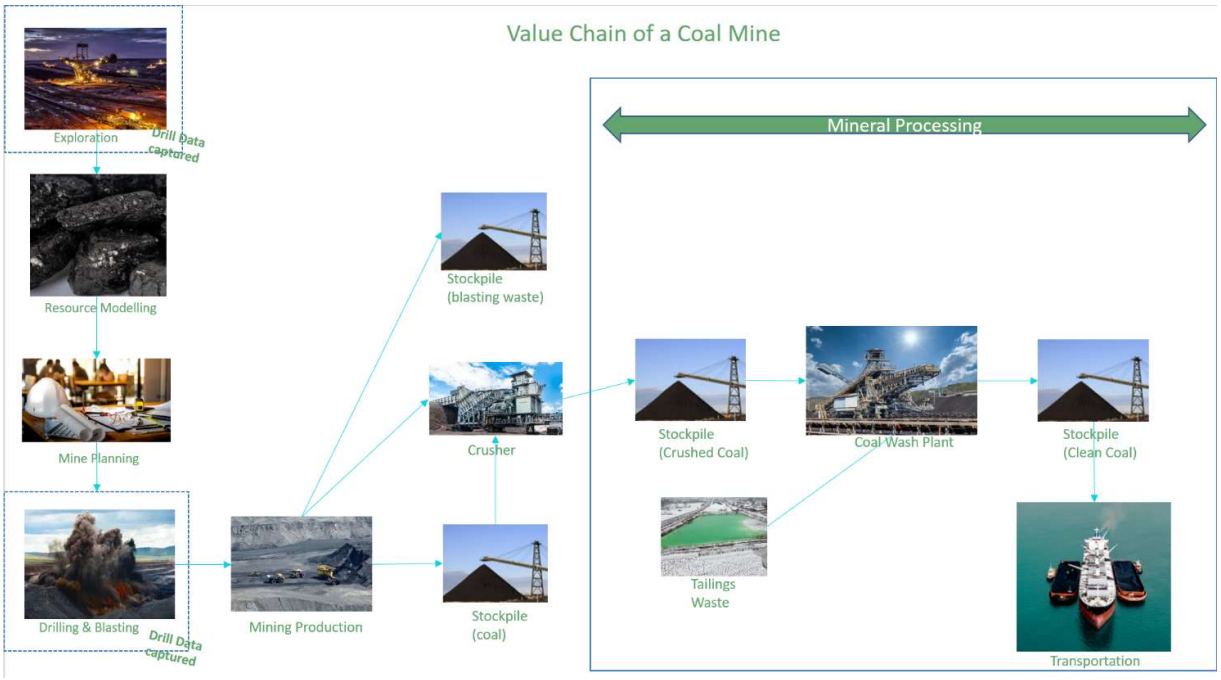
Exploration drilling is performed to determine the location and characteristics of the different rock types (i.e. where is the coal and how good is it?), it is expensive, but necessary, and it is the industry standard. Blasting is used to break up the rock that covers up the coal. Production drilling for drill and blast purposes is used to create a hole which explosives can be loaded into; this means that even with good exploration drilling you still need to go through the production drilling process. There is an opportunity to supplement or even substitute the results of exploration with information from production drilling.

Some of the difficulties faced with production drilling is that the data captured is noisy, and the time between the drilling taking place and the actual loading of explosives is a very small window.

Industry/Domain

The mining industry is $527.2bn business in Australia and has a growth rate of 13%. QLD's mining industry is amongst the most successful in the world and has over 50 major coal mines and 100 metalliferous mines. Drill and blast costs are around 20% of the cost of a total mining operation. Within drill and blast, the cost of drilling is about 20% of the drill & blast costs.

The value chain of a coal mine is shown below. The parts of the value chain being focused on in this design are the exploration, resource modelling and the drill & blast areas. The focus is to utilise information from the drilling process to supplement the exploration information. By doing so, we can prove up the resource model with high resolution information without incurring any additional exploration costs.



This design can be further investigated and possibly enhanced to predict not only if it's coal or not coal, but to also predict the exact type of material as the drill passes through it.

Stakeholders

Drill and blast employees are the first immediate stakeholders. Second is the mine planning engineers who can optimise the delivery of the Contract requirements. Other stakeholders are the shareholders of the business as they will expect to have improved revenue and reduced costs. It comes back to simply meeting Contract requirements as efficiently as possible.

By implementing this model, drill and blast costs wouldn't decrease as we would still be utilising this part of the value chain as is, if not more. The benefit is that there is improved optimisation of the mine planning, which means there would be more revenue produced overall, and less demurrage costs incurred. Another benefit to adopting this model is to avoid damaging coal – if blast limits can be optimised to avoid coal damage, then coal loss and dilution is limited which is one of the major reasons for mining companies not achieving Contractual obligations

Business Question

How can I get a high-resolution model of my geology quickly and accurately to determine if the rock being drilled through is coal or not, without relying on expensive exploration drilling that can take weeks to get results?

Data Question

Can I use the Measurements While Drilling (MWD) data to determine if the rock being drilled through is coal or not coal?

Data

The data was sourced from actual production data that was deidentified. The data consists of 1,952 drilling events over a 1 month period. The data provided included:

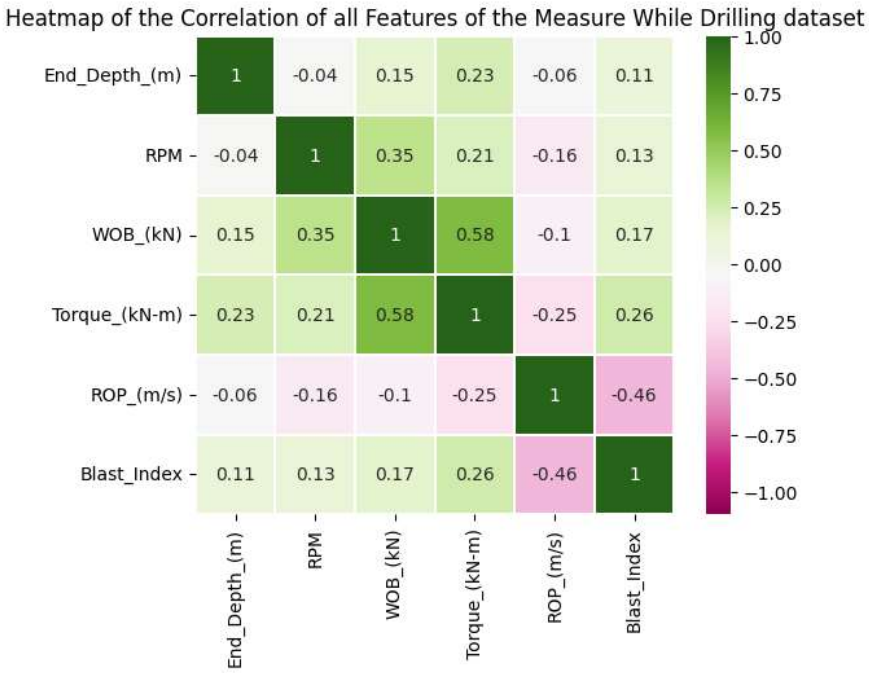
- Hole ID: contains a unique number for each drilling event.
- Drill ID: contains a unique reference for each drill.
- Start_Time
- End_Time
- End_Depth_(m): how deep the hole was drilled
- Metrics captured while drilling is taking place
    - RPM:
    - WOB_(kN)
    - Torque_(kN-m)
    - ROP_(m/s)
    - Blast Index
- Rock ID: a number from 1 – 5 to identify the type of rock, with coal being 2

Data Science Process

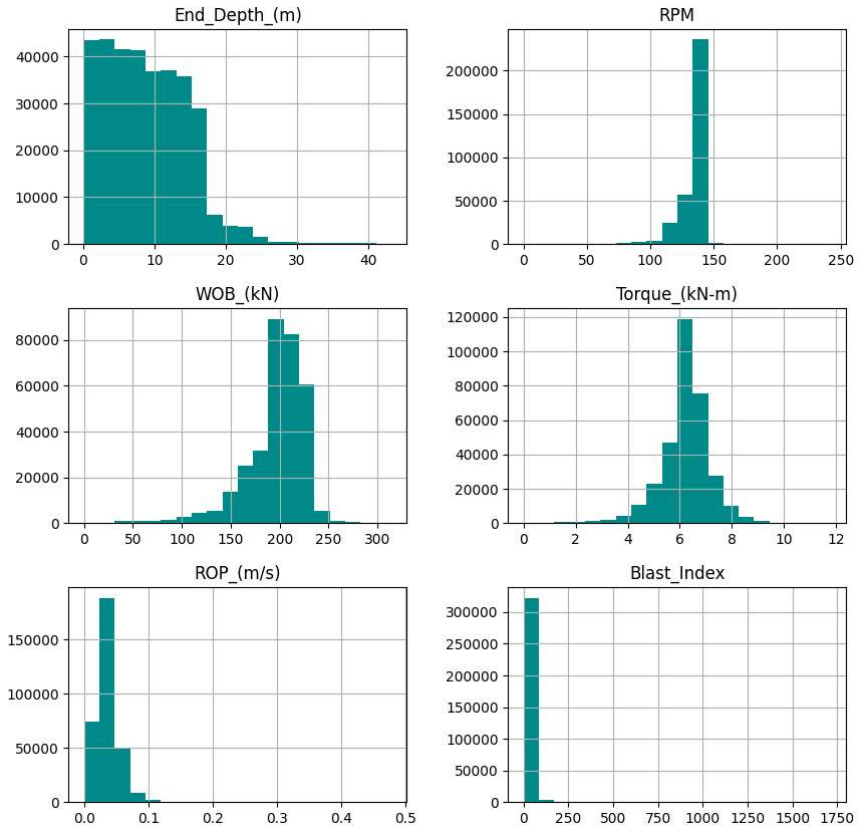
Exploratory Data Analysis (EDA)

The data was loaded into Jupyter notebook as a dataframe. There were 326,436 rows and 11 features when first loaded, and there were no null values. The min of most numeric columns was zero which was found to be correct. The features were renamed to make them easier to work with.

A Pearson Correlation heatmap shows that the most highly correlated Features are WOB_(kN) and Torque_(KN-m) but these are only 0.58 so all features remained in the data frame for initial modelling.



A histogram of each feature shows differently distributed data for all features, with End Depth being left skewed and WOB being right skewed and Torque the most normally distributed of them all.

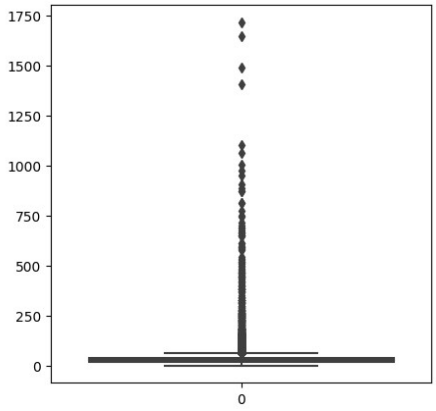


The target value is the rock ID, which initially determines what type of rock the source data was. There were 5 different types of rock, but for this model only 2 categories were necessary – coal or not coal. This field was manipulated to show only coal or not coal, however in the future it's noted that this model could be further developed to show specific types of rocks instead of just focusing on coal. The target value ended up having 263,869 lines for "not coal" and 62,567 rows for "coal" indicating an imbalanced data set.

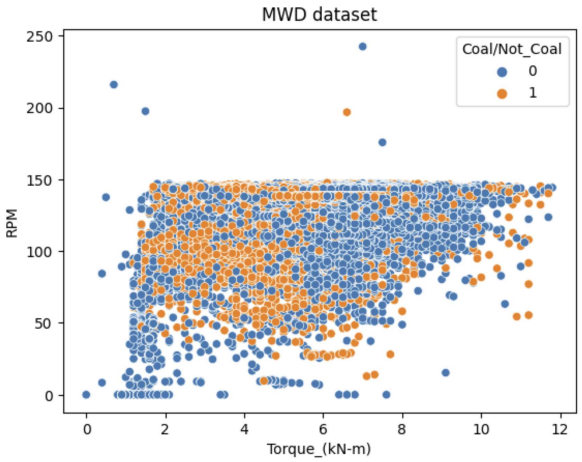
Forward feature selection showed that it was best to use the 4 features of ROP, Torque, WOB and RPM, and not to include End Depth.

Outliers

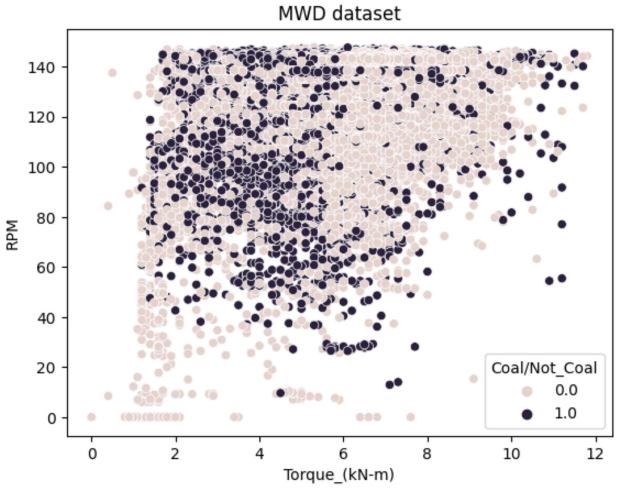
An initial boxplot showed a number of outliers in the Blast_Index feature, and after discussions with the business that provided the data it was discovered that this is a derived field and the calculations are incorrect. This feature was removed from the model.



There are a number of other outliers in the data – this can be seen in the scatterplot below of RPM vs Torque.



Large outliers were removed after business consultation with the updated data set looking like this:



Modelling

Initially (as part of mini project 3) the models Logistic Regression, Naïve Bayes and Support Vector Machine were run. Logistic regression performed best when run with all 4 features and newton-cg hyperparameter tuning. The following Naïve Bayes had poor accuracy regardless of the number of features and no further Hyperparameter tuning was completed. Lastly, the Support Vector machine performed better with only 2 features rather than 3, however it took 2 days to run and due to the computational cost and time impacts no further models or hyperparameter tuning was complete.

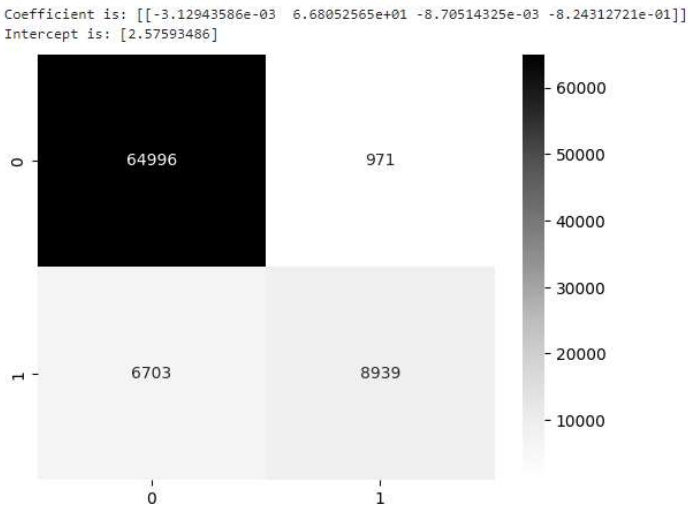
The best performing from each of these models is summarised into the below table.

Model	**Coal or Not Coal?**	**Precision**	**Recall**	**F1-score**	**Accuracy**	**# of Features**
Logistic Regression (model 3, newton-cg)	0 = Coal	0.91	0.99	0.94	0.91	4
	1 = Not Coal	0.90	0.57	0.70		
Naïve Bayes	0 = Coal	0.87	0.97	0.92	0.86	4
	1 = Not Coal	0.77	0.39	0.52		
Support Vector Machine	0 = Coal	0.96	0.97	0.96	0.94	2
	1 = Not Coal	0.85	0.84	0.85		

The best model was the logistic regression which had accuracy of 90.53 with recall of 0.99 for coal and 0.57 for not coal. The false negatives are an issue and further models will be run to try and increase the accuracy for the "not coal" data. Following this, a further 4 models were run to attempt to bring the Recall rate above 84% for “not coal.”

Logistic Regression Model – with Outliers (Model 1)

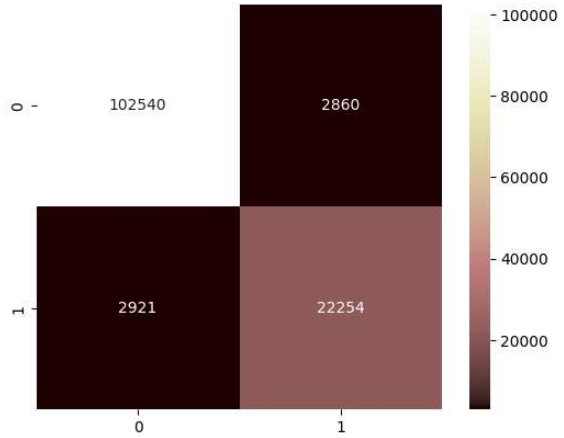
The logistic regression model was run again but this time including the outliers to determine the validity of removing the outliers and the impact this had on the model. The logistic regression model with outliers actually ended up being more accurate than the model that had the outliers removed – the accuracy is 90.534, but recall is still only 0.57 which is too low to be a viable model.



Decision Tree – with Outliers (Model 2)

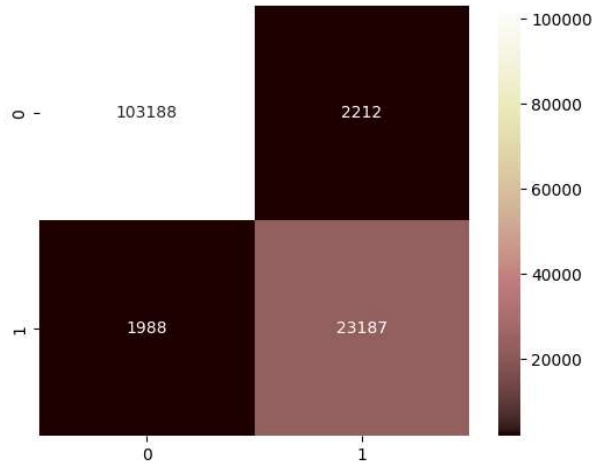
Next a decision tree was run without any hyperparameters. A decision tree is a type of machine learning that makes predictions on how a previous set of questions were answered. When used on the MWD data, the decision tree resulted in training accuracy of 99.9 and test accuracy of 95.57 with the false negatives significantly reducing.

```
Confusion Matrix on Default decision tree with no hyperparameters

Train Accuracy : 0.9999285207366448
Train Confusion Matrix:
[[158469      0]
 [    14  37378]]

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

Test Scores & Confusion Matrix:

Accuracy Score =
0.9557265939115451
```



Following this, numerous hyperparameters were applied to further refine the model and improve the recall score (Appendix 1) Finally, gridsearch was used to determine the best parameters, which were max_depth of 10, min_samples_leaf of 5 and gini rather than entropy. This resulted in:

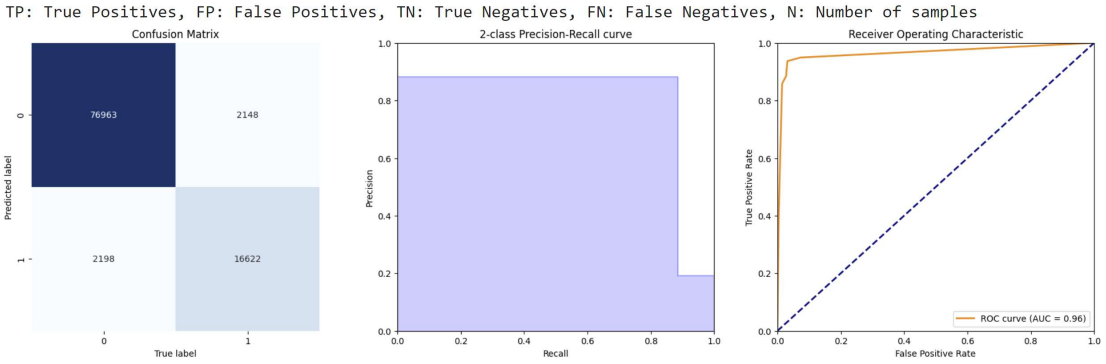
```
Confusion Matrix on Decision Tree using best parameters based on gridsearch

Train Accuracy : 0.973389291385217
Train Confusion Matrix:
[[155680   2789]
 [  2423  34969]]

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

Test Scores & Confusion Matrix:

Accuracy Score =
0.967834577828834
```



Here it can be seen that the recall can get up to 92% using the decision tree and the right hyperparameter tuning.

Stacking – with outliers (Model 3)

Stacking is an ensemble method which uses a machine learning model to learn how to combine predictions from contributing members of the ensemble. A stacking model was run using newston-cg in logistic regression as that was the best logistic regression model in mini project 3. It was run using KNeighbours, Random Forest and Decision Tree, using imbalanced data and including outliers. Due to the use of imbalance data, stratify was set to 'y'.

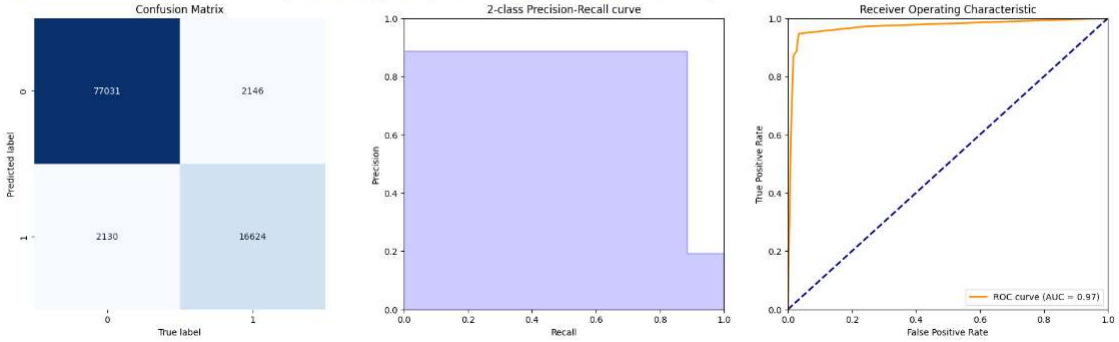


	Model	Accuracy	Precision	Recall	ROC_AUC
0	K-NN	0.886409	0.801378	0.541590	0.932140
1	Random Forest	0.997291	0.996071	0.989771	0.999950
2	Decision Tree	0.999891	1.000000	0.999429	1.000000
3	Logistic Regression	0.903354	0.897601	0.559605	0.975254
4	Stacking	0.999891	1.000000	0.999429	0.999949
5	Stacking (with test)	0.955622	0.883209	0.885562	0.964882

This model was run 5 times using different random states, and using Kfold cross validation to ensure that the data wasn't overfit (Appendix 2). Stacking increased the recall to 88%, however it was still more accurate using Decision Tree alone rather than stacking.

The stacking model was run again, however this time using random over sampler to generate a more balanced data set. This gave a result of:

```
Accuracy : 0.9563 [TP / N] Proportion of predicted labels that match the true labels. Best: 1, Worst: 0
Precision: 0.8864 [TP / (TP + FP)] Not to label a negative sample as positive.       Best: 1, Worst: 0
Recall   : 0.8857 [TP / (TP + FN)] Find all the positive samples.                    Best: 1, Worst: 0
ROC AUC  : 0.9705                                                                    Best: 1, Worst: < 0.5
----------------------------------------------------------------------------------------------------
TP: True Positives, FP: False Positives, TN: True Negatives, FN: False Negatives, N: Number of samples
```



	Model	Accuracy	Precision	Recall	ROC_AUC
0	K-NN	0.905754	0.853372	0.979871	0.982524
1	Random Forest	0.999334	0.998735	0.999935	0.999999
2	Decision Tree	0.999938	0.999881	0.999995	1.000000
3	Logistic Regression	0.958805	0.949399	0.969270	0.981201
4	Stacking	0.999938	0.999881	0.999995	0.999939
5	Stacking (with test)	0.956337	0.886424	0.885669	0.970543

Here you can see that the recall has slightly increased again to 88.57

Neural Network – with Outliers (4th Model)

A neural network (or perceptron) works similarly to the way a nerve cell makes a decision. Inputs are given weights, the weighted signals are summed and the summed signal is transformed by an activation function to produce an output. The model here is a binary classification neural network. Initial attempts at running this model failed to reach a conclusive result due to the loss score remaining the same on all attempts; meaning the model was not learning. I then ran the neural network with different optimisers, batch sizes and regulizers, and had the below results. Model 5 with Adagrad Optimiser and 8 layers had the best loss score. The neural network models will be further developed and tweaked to determine the most appropriate model.

Neural Network Model	Optimiser	Regulizer	Batches	Layers	Loss Score
1	SGD	N/A	N/A	1	12.33
2	SGD	N/A	N/A	8	6.98
3	Adamax	N/A	N/A	8	6.47
4	SGD	N/A	N/A	12	6.42
5	Adagrad	N/A	N/A	8	5.8
6	SGD	L2	32		6.47

Outcomes

The decision tree was the most accurate model in terms of recall and overall accuracy. This confirms that it is possible to predict if the rock being drilled through during production is coal or not.

Models (*exc Neural Network)	Coal or Not Coal?	Precision	Recall	Accuracy
Logistic Regression	0 = Coal	0.91	0.99	0.91
	1 = Not Coal	0.90	0.57	
Decision Trees	0 = Coal	0.98	0.98	0.97
	1 = Not Coal	0.91	0.92	
Stacking		0.88	0.88	0.96

**The Neural Network needs further work to improve the performance of this model.*

Implementation

The next steps in this process will be to work closely with a mine in developing this to a point that it can be used in Production.

Data answer

The Measurements While Drilling (MWD) data can be used to determine if the rock being drilled through is coal or not coal during production, with accuracy of 97%.

Business answer

The business question was answered satisfactorily. The time to run the model is only a few minutes and the accuracy and recall scores are 97% and 92% respectively. However it is still to be determined through stakeholder engagement if 92% is an acceptable recall rate.

Response to Stakeholders

Discuss results with stakeholders and determine what the appropriate range for recall is – if recall of 92% and precision of 98% fits within this acceptable range, then this model will be further developed into a marketable product.

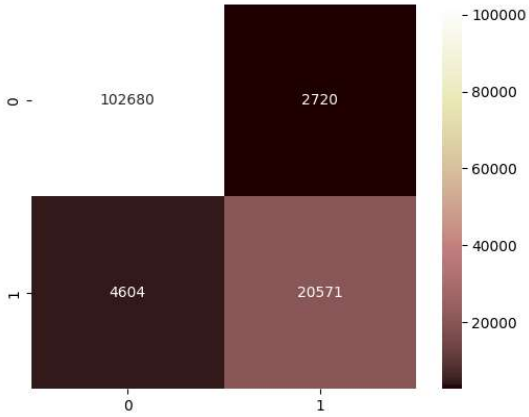
End to End Solution

This model will be developed into a product that can be marketed to mining companies, this will involve the setting up of a data business model. The end-to-end solution for the mining companies will be for them to upload their data as it is drilled, and get real time information on what rock they are drilling into

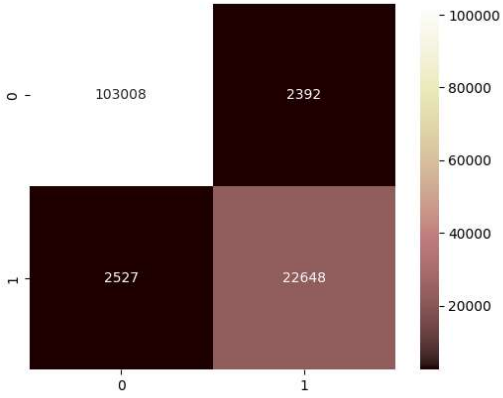
Appendix 1

Decision Tree Iterations

```
Confusion Matrix on Decision Tree with a max depth of 3

Train Accuracy : 0.9443023368613456
Train Confusion Matrix:
[[154448   4021]
 [  6888  30504]]

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

Test Scores & Confusion Matrix:

Accuracy Score =
0.9439096304805668
```

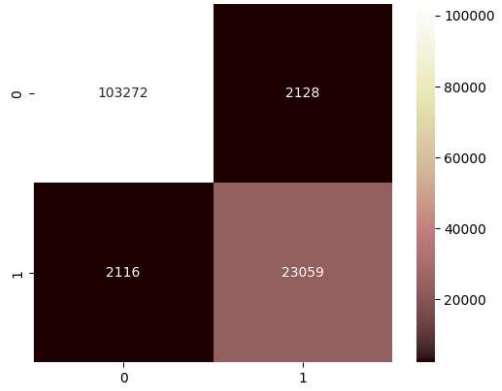



```
Confusion Matrix on Decision Tree with minimum samples of 20 before split

Train Accuracy : 0.9810324669025482
Train Confusion Matrix:
[[156668   1801]
 [  1914  35478]]

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

Test Scores & Confusion Matrix:

Accuracy Score =
0.9623281638904844
```

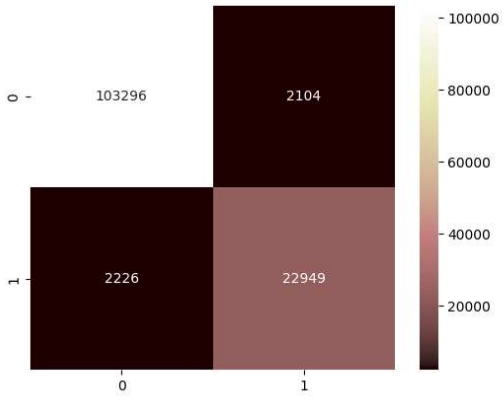


Controlling the depth of the tree gives 94% accuracy. try with another layer?

```
Confusion Matrix on Decision Tree with minimum value in leaf nodes of 20

Train Accuracy : 0.9733535517535395
Train Confusion Matrix:
[[155811   2658]
 [  2561  34831]]

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

Test Scores & Confusion Matrix:

Accuracy Score =
0.9674976067394218
```




```
Confusion Matrix on Decision Tree using entropy instead of Gini

Train Accuracy : 0.9733280234451984
Train Confusion Matrix:
[[155905   2564]
 [  2660  34732]]

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

Test Scores & Confusion Matrix:

Accuracy Score =
0.9668389814282979
```



Appendix 2

Stacking Iterations

```
[70]: results
```

[70]:

	Model	Accuracy	Precision	Recall	ROC_AUC
0	K-NN	0.886409	0.801378	0.541590	0.932140
1	Random Forest	0.997291	0.996071	0.989771	0.999950
2	Decision Tree	0.999891	1.000000	0.999429	1.000000
3	Logistic Regression	0.903354	0.897601	0.559605	0.975254
4	Stacking	0.999891	1.000000	0.999429	0.999949
5	Stacking (with test)	0.955622	0.883209	0.885562	0.964882

[99]:

	Model	Accuracy	Precision	Recall	ROC_AUC
0	K-NN	0.887276	0.803574	0.545129	0.932422
1	Random Forest	0.997230	0.996457	0.989063	0.999948
2	Decision Tree	0.999908	1.000000	0.999521	1.000000
3	Logistic Regression	0.903915	0.898660	0.562070	0.975298
4	Stacking	0.999908	1.000000	0.999521	0.999951
5	Stacking (with test)	0.955867	0.886186	0.883165	0.964230

[109]:

	Model	Accuracy	Precision	Recall	ROC_AUC
0	K-NN	0.886550	0.800902	0.543097	0.932388
1	Random Forest	0.997287	0.996481	0.989337	0.999948
2	Decision Tree	0.999912	1.000000	0.999543	1.000000
3	Logistic Regression	0.905608	0.900793	0.570336	0.975820
4	Stacking	0.999912	1.000000	0.999543	0.999986
5	Stacking (with test)	0.956480	0.887790	0.884763	0.965627

	Model	Accuracy	Precision	Recall	ROC_AUC
0	K-NN	0.886344	0.801890	0.540562	0.932024
1	Random Forest	0.997313	0.996413	0.989543	0.999954
2	Decision Tree	0.999904	1.000000	0.999498	1.000000
3	Logistic Regression	0.904532	0.900226	0.564468	0.975536
4	Stacking	0.999904	1.000000	0.999498	0.999951
5	Stacking (with test)	0.956510	0.886857	0.886148	0.965277

[119]:

	Model	Accuracy	Precision	Recall	ROC_AUC
0	K-NN	0.886094	0.799899	0.541064	0.931640
1	Random Forest	0.997015	0.995541	0.988858	0.999946
2	Decision Tree	0.999895	1.000000	0.999452	1.000000
3	Logistic Regression	0.904037	0.899200	0.562367	0.975522
4	Stacking	0.999895	1.000000	0.999452	0.999963
5	Stacking (with test)	0.956122	0.885926	0.885029	0.965616

References

https://www.ibisworld.com/au/market-size/mining/

https://www.business.qld.gov.au/

https://www.mastersindatascience.org/learning/machine-learning-algorithms/decision-tree/#:~:text=A%20decision%20tree%20is%20a,that%20contains%20the%20desired%20categorization.