Proceedings of the 33rd European Safety and Reliability Conference (ESREL 2023)
Edited by Mário P. Brito, Terje Aven, Piero Baraldi, Marko Čepin and Enrico Zio
©2023 ESREL2023 Organizers. *Published by* Research Publishing, Singapore.
doi: 10.3850/978-981-18-8071-1_P050-cd

Predictive Modelling for Asset Availability using Artificial Intelligence

Kok Ping Hun
Staff Reliability Engineer, Petroleum Nasional Berhad, Malaysia.
E-mail: kok_pinghun@petronas.com

Khairul Nizam Baharim
Data Scientist, PETRONAS Digital Sdn Bhd, Malaysia.
E-mail: khairulnizam.bahari@petronas.com

Reliability, Availability and Maintainability (RAM) studies have been performed on equipment, systems, and oil and gas production fields to predict availability targets using reliability block diagrams and equipment runtime statistical analysis. However, no integration of RAM analysis involving multiple fields can be found; multi-field data need to be **updated manually**, and there is **no live** data updating feature available, resulting in data inaccuracy and longer duration to complete RAM analysis. In this study—anchoring on the theme of integration and automation—the authors aim to improve completion time, **increase the visibility of availability** data for the production field and **improve flexibility** in data update. The main objective is to allow for **accurate prediction** of field availability for the following month and to **expedite** the correct intervention actions identification to meet the required target. The predictive model was developed utilizing Microsoft Azure Machine Learning and R Programming by utilizing availability data of field with Mean Absolute Error less than one percent. As part of machine learning improvement, it is recommended for the model to be expanded to include other fields with integration of more live data.

Keywords: Asset Availability, Predictive Model, Machine Learning, Artificial Intelligence.

1. Introduction

Upstream RAM studies are used widely to determine projected availability or anticipated production of a gas/oil production field via development of reliability block diagrams and runtime analysis of equipment making up the field. Typically, the simulated availability/production numbers for individual fields are used to predict performance of the entire supply chain. Currently, RAM studies are being carried out individually on systems, equipment, or at field level using different type RAM software. While all the fields are linked to each other via pipelines and supported by gas compressor or oil pumping stations and storage facilities, however there is **no integrated** RAM has been done to study and model the entire supply chain, integrating data from equipment level. Besides that, each individual RAM software is unique in feature and **unable to interface with each other's**. Each time when full system (or supply chain) performance is needed, data must be **re-entered manually** resulting in **high likelihood of data-entry error and longer time required for completion as there is no live data updating.**

A typical upstream field network is illustrated in Figure 1, with each of field details is illustrated in reliability block diagrams shown in figure 2.

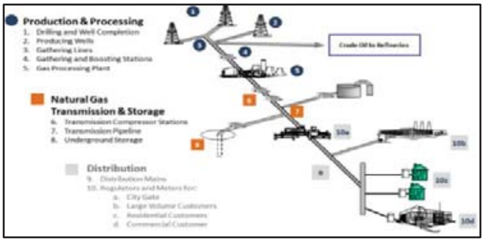


Fig. 1. Typical upstream field network.

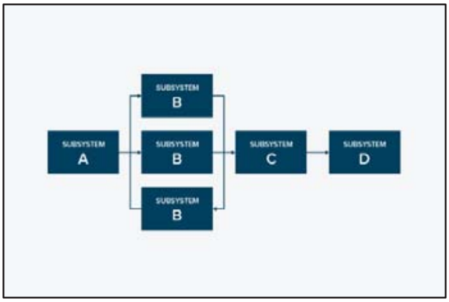
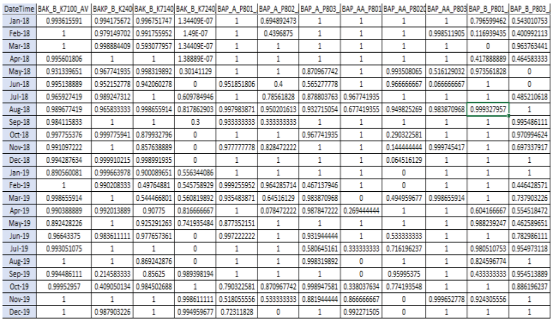


Fig. 2. Typical Reliability Block Diagrams

In this study which aim to develop the baseline Model via machine learning tools using monthly field Availability data from 2018 to 2020, with full utilization of Machine Learning Studio and R Programming (which can replicate reliability block diagram development in programming). The study involved data extraction from daily operation reports, data correlation, Exploratory Data Analysis (EDA), and feature importance which is portrayed in Power BI. Under machine learning methodology, it involves with feature creation, feature transformation, feature reduction and feature selection. Twenty-four sets of testing and combination of different features are used before a final combination that constitutes the model is selected based on lowest Mean Absolute Error (MAE) with the lowest difference between test and validation data. The model is validated via Model Validation Strategy using hyperparameter tuning and cross validation which resulted in generation of validated MAE results. For the continuation of the model and as part of machine learning improvement, it is recommended for the model is expanded to include other fields with more data and continuous validation using other programming software such as Phyton or even improvement of existing R programming coding.



DateTime	BAK_B_K7100_AV	BAKP_B_K240	BAK_B_K7140	BAK_B_K7240	BAP_A_P801_	BAP_A_P802_	BAP_A_P803_	BAP_AA_P801	BAP_AA_P802	BAP_AA_P803	BAP_B_P801_	BAP_B_P803_
Jan-18	0.993615591	0.994175672	0.996751747	1.34409E-07	1	0.694892473	1	1	1	1	0.796599462	0.543010753
Feb-18	1	0.979149702	0.991755952	1.49E-07	1	0.4396875	1	1	1	0.998511905	0.116939435	0.400992113
Mar-18	1	0.998884409	0.593077957	1.34409E-07	1	1	1	1	1	1	0	0.963763441
Apr-18	0.995601806	1	1	1.38889E-07	1	1	1	1	1	1	0.417888889	0.464583333
May-18	0.931339651	0.967741935	0.998319892	0.30141129	1	1	0.870967742	1	0.993508065	0.516129032	0.973561828	0
Jun-18	0.995138889	0.952152778	0.942060278	0	0.951851806	0.4	0.565277778	1	0.966666667	0.066666667	1	0
Jul-18	0.965927419	0.989247312	1	0.609784946	1	0.78561828	0.878803763	0.967741935	1	1	1	0.485210618
Aug-18	0.989677419	0.965833333	0.998655914	0.817862903	0.997983871	0.950201613	0.932715054	0.677419355	0.949825269	0.983870968	0.999327957	1
Sep-18	0.984115833	1	1	0.3	0.933333333	0.333333333	1	1	1	1	1	0.995486111
Oct-18	0.997755376	0.999775941	0.879932796	0	1	1	0.967741935	1	0.290322581	1	1	0.970994624
Nov-18	0.991097222	1	0.857638889	0	0.977777778	0.828472222	1	1	0.144444444	0.999745417	1	0.697337917
Dec-18	0.994287634	0.999910215	0.998991935	0	1	1	1	1	0.064516129	1	1	1
Jan-19	0.890560081	0.999663978	0.900089651	0.556344086	1	1	1	1	0	1	1	1
Feb-19	1	0.990208333	0.49764881	0.545758929	0.999255952	0.964285714	0.467137946	1	0	1	1	0.446428571
Mar-19	0.998655914	1	0.544466801	0.560819892	0.935483871	0.64516129	0.983870968	0	0.494959677	0.998655914	1	0.737903226
Apr-19	0.990388889	0.992013889	0.90775	0.816666667	1	0.078472222	0.987847222	0.269444444	1	1	0.604166667	0.554518472
May-19	0.892428226	1	0.925291263	0.741935484	0.877352151	1	1	1	1	1	0.988239247	0.462589651
Jun-19	0.96643375	0.983611111	0.977657361	0	0.997222222	1	0.931944444	1	0.533333333	1	1	0.782986111
Jul-19	0.993051075	1	1	0	1	1	0.580645161	0.333333333	0.716196237	1	0.980510753	0.954973118
Aug-19	1	1	0.869242876	0	1	1	0.998319892	0	1	1	0.824596774	1
Sep-19	0.994486111	0.214583333	0.85625	0.989398194	1	1	1	0	0.95995375	1	0.433333333	0.954513889
Oct-19	0.99952957	0.409050134	0.984502688	1	0.790322581	0.870967742	0.998947581	0.338037634	0.774193548	1	1	0.886196237
Nov-19	1	1	1	0.998611111	0.518055556	0.533333333	0.881944444	0.866666667	0	0.999652778	0.924305556	1
Dec-19	1	0.987903226	1	0.994959677	0.72311828	0	1	0.992271505	0	1	1	1

Fig. 3. 2018- 2020 Equipment Availability Tracking

2. Methodology

Eight years of availability (Ref.1) data from selected equipment which had direct production impact are collected between 2013 and 2020. Data quality is manually analysed to ensure consistency throughout the eight-year period. However, from 2013 to 2017, there were changes on equipment configuration and new equipment were added to the system because of changes in maintenance and operation philosophy. Thus only 2018 to 2020 Availability data are finally used in this machine learning experiment. Figure 3 shows Equipment Availability tracking from 2018- 2020 while Figure 4 outlines the keys steps in the methodology. Once the range of data is selected, Exploratory Data Analysis is performed to further understand the characteristics of data in terms of proportion of missing values, data outliers' existence, correlation among the features in the data, and the most important features that could highly contribute to higher accuracy in predictive models.

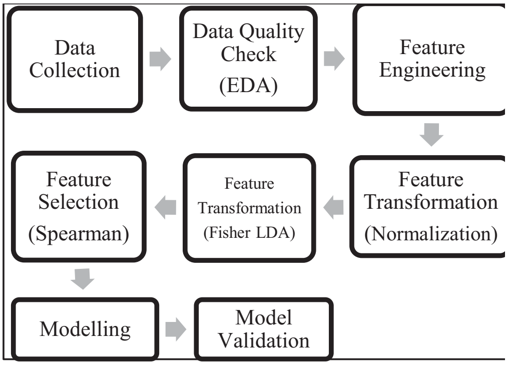

Data Collection
Data Quality Check (EDA)
Feature Engineering
Feature Transformation (Normalization)
Feature Transformation (Fisher LDA)
Feature Selection (Spearman)
Modelling
Model Validation

Fig. 4. Key steps in the methodology.

2.1. *Exploratory Data Analysis (EDA) (Ref -6)*

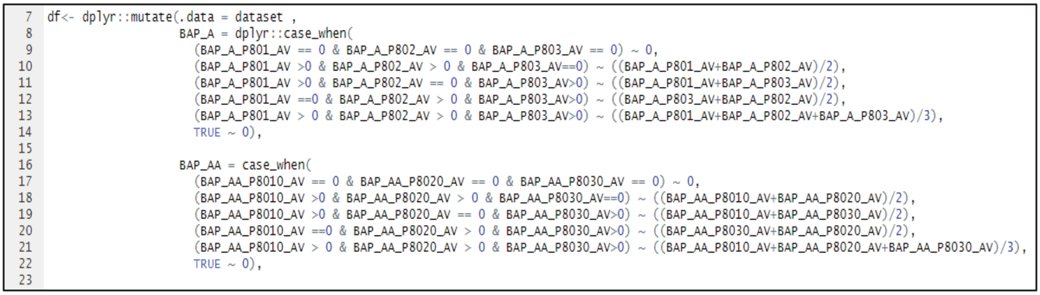
Equipment’s’ Availability data with missing values are manually analysed to impute the possible values for them. The objective of treatment of these missing values is to prevent problems caused by missing data that could arise when training the model. Outliers are handled by removing a few points of lowest and highest data using threshold approach. In Machine Learning Studio, we use the clip values component to identify and optionally replace data values that were above or below a specified threshold with a mean, a constant, or other substitute value. Outlier data are observations that appear far away and diverge from an overall pattern in the collected data. It could skew and mislead the training process of machine learning algorithms, resulting in longer training times, and less accurate models.

In this experiment, we use 99% upper threshold and 1% lower threshold.

The feature correlation and feature importance approach are used to identify important features that have most relationship to the target variable, *i.e.,* Availability variable. The results could be one or multiple features depending on another feature or a cause for another feature or it could be one or multiple features associated with other features. This problem called multicollinearity. When it happens, one feature in a machine learning model can be linearly predicted from the other features (not the availability target variable) in the dataset.

2.2. *Feature Engineering*

In the Feature Engineering phase, the collected data from various systems are aggregated using engineering reliability knowledge by replicating reliability block diagram in R programming. The value of Availability (AV) ranges from 0 to 1, where 1 represents full Availability of the system. Sample snippet codes for the aggregation is depicted in Figure 5; lines 8 to 14, show that three equipment which located to location BAPA, *i.e.*, BAP_A_P801_AV, BAP_A_P802_AV, BAP_A_P803_AV are used to calculate the Availability of BAP_A in which an aggregated new feature was created in the data.



```
 7  df<- dplyr::mutate(.data = dataset ,
 8                     BAP_A = dplyr::case_when(
 9                       (BAP_A_P801_AV == 0 & BAP_A_P802_AV == 0 & BAP_A_P803_AV == 0) ~ 0,
10                       (BAP_A_P801_AV >0 & BAP_A_P802_AV > 0 & BAP_A_P803_AV==0) ~ ((BAP_A_P801_AV+BAP_A_P802_AV)/2),
11                       (BAP_A_P801_AV >0 & BAP_A_P802_AV == 0 & BAP_A_P803_AV>0) ~ ((BAP_A_P801_AV+BAP_A_P803_AV)/2),
12                       (BAP_A_P801_AV ==0 & BAP_A_P802_AV > 0 & BAP_A_P803_AV>0) ~ ((BAP_A_P803_AV+BAP_A_P802_AV)/2),
13                       (BAP_A_P801_AV > 0 & BAP_A_P802_AV > 0 & BAP_A_P803_AV>0) ~ ((BAP_A_P801_AV+BAP_A_P802_AV+BAP_A_P803_AV)/3),
14                       TRUE ~ 0),
15
16                     BAP_AA = case_when(
17                       (BAP_AA_P8010_AV == 0 & BAP_AA_P8020_AV == 0 & BAP_AA_P8030_AV == 0) ~ 0,
18                       (BAP_AA_P8010_AV >0 & BAP_AA_P8020_AV > 0 & BAP_AA_P8030_AV==0) ~ ((BAP_AA_P8010_AV+BAP_AA_P8020_AV)/2),
19                       (BAP_AA_P8010_AV >0 & BAP_AA_P8020_AV == 0 & BAP_AA_P8030_AV>0) ~ ((BAP_AA_P8010_AV+BAP_AA_P8030_AV)/2),
20                       (BAP_AA_P8010_AV ==0 & BAP_AA_P8020_AV > 0 & BAP_AA_P8030_AV>0) ~ ((BAP_AA_P8030_AV+BAP_AA_P8020_AV)/2),
21                       (BAP_AA_P8010_AV > 0 & BAP_AA_P8020_AV > 0 & BAP_AA_P8030_AV>0) ~ ((BAP_AA_P8010_AV+BAP_AA_P8020_AV+BAP_AA_P8030_AV)/3),
22                       TRUE ~ 0),
23
```

Fig 5 Availability Aggregation using R programming.

Feature Transformation (Ref-8) such Principal Component Analysis (PCA) and Fisher Linear Discriminant Analysis (LDA) is used in this experiment to project the data onto weights vector that represent the data. In general, the method analyses the data and creates a reduced features that capture all the information contained in the dataset, but in a smaller number of features. Principal component analysis (PCA) is widely used for dimension reduction and embedding of real data in social network analysis, information retrieval, and natural language processing. The LDA method is often used for dimensionality reduction, because it projects a set of features onto a smaller feature space while preserving the information that discriminates between classes of the target variable, *i.e.*, Availability. This method not only reduces computational costs for a given classification task but can help prevent overfitting in machine learning model. Thus, both methods, PCA and LDA were chosen for this experiment.

2.3. Machine Learning Models (Ref-2,3,5& 7)

The model development and experimentation are performed in Azure Machine Learning platform; in particular, Linear Regression (LR), Boosted Decision Tree (BDT) and Neural Network (NN) are selected for this experiment.

Theoretically, LR attempts to establish a linear relationship between one or more independent variables to a numeric outcome, in this case Availability. If there is a linear relationship between selected features and the numeric outcome such as in Figure 6, it could churn out a very accurate model. LR is a common statistical method, which has been adopted in machine learning and enhanced with many new methods for fitting the line and measuring error. It also tends to work well on high-dimensional, sparse data sets lacking complexity. In Azure Machine Learning Studio, the module supports two methods to measure error and fit the regression line: ordinary least squares (OLS) method, and gradient descent. Gradient descent is a method that minimizes the amount of error at each step of the model training process. OLS refers to the loss function, which computes error as the sum of the square of distance from the actual value to the predicted line and fits the model by minimizing the squared error. Both methods have been applied to identify the best method that can produce the most accurate model.

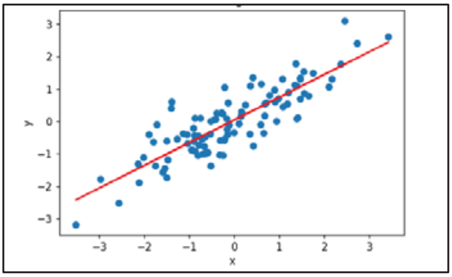


Fig. 6. Example of linear relationship between feature (x) and numeric outcome (y)

Another approach, BDT is a kind of ensemble of regression trees using boosting method. Boosting means that each tree is dependent on prior trees. The algorithm learns by fitting the residual of the trees that preceded it. Thus, boosting in a decision tree ensemble tends to improve accuracy with some small risk of less coverage. In Azure Machine Learning Studio, BDT uses an efficient implementation of the MART gradient boosting algorithm. Gradient boosting is a machine learning technique for regression problems. It builds each regression tree in a stepwise fashion, using a predefined loss function to measure the error in each step and correct for it in the next. Thus, the prediction model is an ensemble of weaker prediction models. Figure 7 shows a simple process flow of BDT.

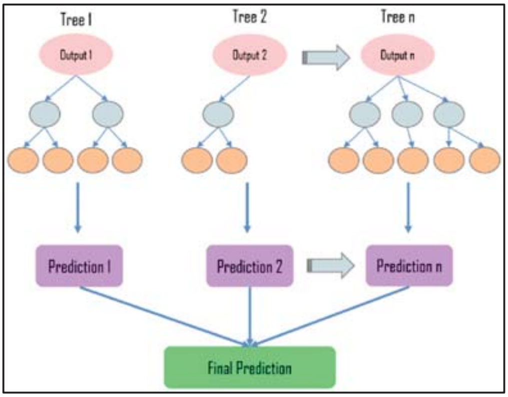


Fig. 7. Boosted Decision Tree Process Flow

Another approach in this experiment, NN, is widely known for use in deep learning and modelling complex problems such as image recognition, they are easily adapted to regression problems. Any class of statistical models can be termed a neural network if they use adaptive weights and can approximate non-linear functions of their inputs. Thus, neural network regression is suited to problems where a more traditional regression model cannot fit a solution. Figure 8 shows the NN configuration in Azure Machine Learning Studio (top image) which is represented in the NN architecture (bottom image).

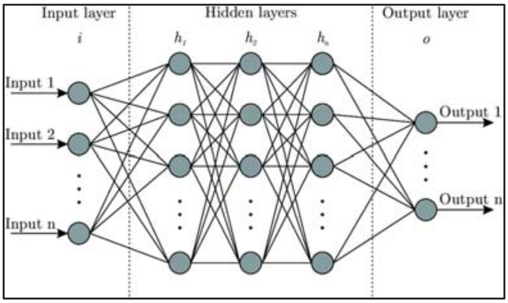
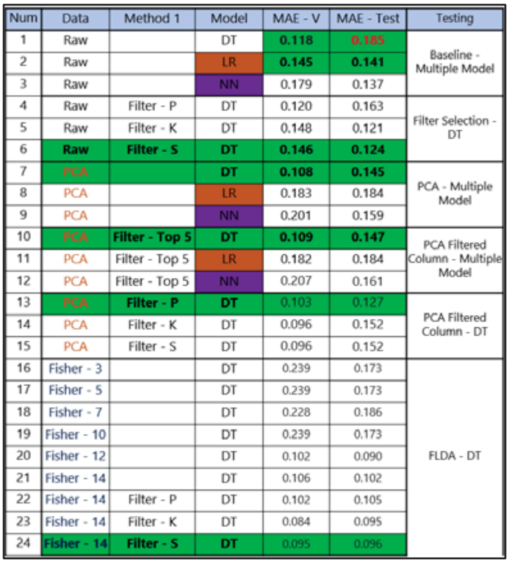


Fig. 8. NN configurations in Machine Learning Studio and basic NN architecture

In summary, these three types of Machine Learning approaches are selected from easy to medium and complex models such Neural Network is used to identify the best algorithm that can fit in the use case of asset availability.



Num	Data	Method 1	Model	MAE - V	MAE - Test	Testing
1	Raw		DT	**0.118**	**0.185**	Baseline - Multiple Model
2	Raw		LR	**0.145**	**0.141**	
3	Raw		NN	0.179	0.137	
4	Raw	Filter - P	DT	0.120	0.163	Filter Selection - DT
5	Raw	Filter - K	DT	0.148	0.121	
6	**Raw**	**Filter - S**	**DT**	**0.146**	**0.124**	
7	**PCA**		**DT**	**0.108**	**0.145**	PCA - Multiple Model
8	PCA		LR	0.183	0.184	
9	PCA		NN	0.201	0.159	
10	**PCA**	**Filter - Top 5**	**DT**	**0.109**	**0.147**	PCA Filtered Column - Multiple Model
11	PCA	Filter - Top 5	LR	0.182	0.184	
12	PCA	Filter - Top 5	NN	0.207	0.161	
13	**PCA**	**Filter - P**	**DT**	0.103	0.127	PCA Filtered Column - DT
14	PCA	Filter - K	DT	0.096	0.152	
15	PCA	Filter - S	DT	0.096	0.152	
16	Fisher - 3		DT	0.239	0.173	FLDA - DT
17	Fisher - 5		DT	0.239	0.173	
18	Fisher - 7		DT	0.228	0.186	
19	Fisher - 10		DT	0.239	0.173	
20	Fisher - 12		DT	0.102	0.090	
21	Fisher - 14		DT	0.106	0.102	
22	Fisher - 14	Filter - P	DT	0.102	0.105	
23	Fisher - 14	Filter - K	DT	0.084	0.095	
24	**Fisher - 14**	**Filter - S**	**DT**	0.095	0.096	

Fig. 9. Summary of Model Experimentation Results

2.4. *Model Validation*

Model validation consists of hyperparameter tuning, cross validation strategy and model testing to ensure its accuracy. The goal of hyperparameter tuning is to determine the optimum configurations for a machine learning model. The component builds and tests multiple models by using different combinations of settings. It compares metrics over all models to get the combinations of settings. There are two methods available: entire grid, and random sweep. We use entire grid method to search all over a grid of predefined configuration. This method will try different combinations of configurations to identify the best learner.

Cross validation strategy is a technique used in machine learning to assess both the variability of a dataset and the reliability of any model trained through that data. It divides the dataset into some number of subsets or folds. Then a model is built on each fold and returns a set of accuracy statistics for each fold. By comparing the accuracy statistics for all the folds, we can interpret the quality of the dataset and understand whether the model is susceptible to variations in the data. In this experiment, we used 10 folds. The usage of three (3) methods is to ensure the validation processes are done more balanced and holistically.

3. Experimental Results and Discussions

The success criteria of model acceptance are:

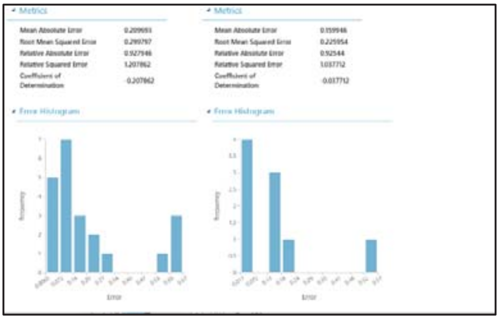
a) MAE result for both validation and test data should be less than 10%.
b) The different between MAE validation and MAE test data should be less than 5%

From various experimentation results (a total of 24 combinations of data, methods, and models), it is found that combination of Fisher LDA with 14 sets of data, correlation of spearman method and boosted decision tree model yield the best result with MAE validation data of 0.095 (9.5% error) and MAE test data of 0.096 (9.6%) while the difference between both validation and test data is 0.1 %. Figures 9 and 10 show the various test results, Machine Learning experimentation model and simulation results. From the various combination of data, method, and model, it took 23 set of different type of combination to finally meet the success criteria. The closest final combination which are Fisher LDA with 14 sets of data, correlation of kindle method and boosted decision tree model yield the best result of MAE validation data of 0.084 (8.4% error) and MAE test data of 0.095 (9.6%), yet it is not been selected as the best model despite having lower MAE error rate , both different MAE validation and test error rate is higher 0.1 (1% different error) compare to selected final model which has MAE different of 0.1%.

Besides that, as it is the experimental and baseline modelling also shown several crucial insights as below,

a) Model combination with raw data without undergoing feature transformation will yield significantly high MAE error for both validation and test data of more than 10%.
b) Among three (3) model combination with raw data, different between both MAE validation and test are more 1%
c) Under the next three (3) model combination with raw data with data filtering, the MAE test and validation are lower.
d) For data which undergoes feature transformation, the MAE test and validation also show lower error compared to without feature transformed data.
e) For data which undergoes feature transformation with machine learning, it is shown there is increase in MAE test and validation error ranging between 10% to 20%
f) With the inclusion of BDT in model combination, there is decreasing trend in MAE test and validation error.
g) Combination of Fisher LDA and BDT, it further reduces the MAE validation and test data error while improving the difference between both MAE validation and test data.
h) With more set of data been aligned with fisher LDA and BDT, it yields the best result in both low MAE and its difference between both MAE validation and test data.

As the summary of the result, it shown that with advancement of technology, operation and maintenance data become more readily accessible, voluminous in nature and available in various format and structure, machine learning increasing become fundamental and more useful tools in ensuring faster data processing/structuring, increasing process cycle efficiency and lower error which resulted in much better accuracy in predictive analysis.



Metrics

Mean Absolute Error	0.209693
Root Mean Squared Error	0.299797
Relative Absolute Error	0.927946
Relative Squared Error	1.207862
Coefficient of Determination	-0.207862

Metrics

Mean Absolute Error	0.159946
Root Mean Squared Error	0.225954
Relative Absolute Error	0.92544
Relative Squared Error	1.037712
Coefficient of Determination	-0.037712

Fig. 10. Microsoft Azure Machine Learning – Model Evaluation Result

4. Conclusion

As the organization embraces digital as part of its business strategy, it is crucial for more collaborative effort between engineering/operation and digital engineering. With the voluminous data readily available and with the expectation of pace and efficiency, this study proved to be a great testament of such, with company's desired target (availability/production efficiency) derived from the result of RAM study at faster pace and better accuracy. With the greater and deepest embracement of technology and artificial intelligence technology, it amplifies data transparency while increasing work efficiency.

The current experimentation model is a baseline model, and it can be further enhanced and able to be scaled up by:

1) Improving analytics dashboard visibility which includes predictive analysis incorporating well and production deferment/estimation performance while providing more analysis insights
2) Enhancing R programming coding by including sensitivity analysis various operating and maintenance philosophies that including production dynamics performances
3) Validating R programming accuracy with other programming language (*e.g.*, Phyton) with the aim of automating in updating programming code, and
4) Expanding the model to include other operating fields and assets with the end of goal having an integrated model of all operating assets within organization for better and live monitoring of asset performance
5) Enhancing the model accuracy by testing different type of model combination ranging from different feature engineering method and machine learning tools as it increases the model robustness and expedite on the machine learning curve.
6) Validating model accuracy with other machine learning algorithm as part of continuous improvement

References

(1) International Standard, ISO 14224, 2006. Petroleum and natural gas industries — Collection and exchange of reliability and maintenance data for equipment. IHS.

(2) Drucker, H., & Cortes, C. (1995). Boosting decision trees. Advances in neural information processing systems, 8.

(3) Poole, M. A., & O'Farrell, P. N. (1971). The Assumptions of the Linear Regression Model. Transactions of the Institute of British Geographers, 52, 145–158.

(4) PETRONAS. (2021). Citizen Analytics Programme Azure Machine Learning Training

(5) Lawrence, J. (1993). Introduction to neural networks. California Scientific Software.

(6) Microsoft. (2023). Algorithm & component reference for Azure Machine Learning designer.

(7) Drucker, H., & Cortes, C. (1995). Boosting decision trees. Advances in neural information processing systems, 8.

(8) Wold, S., Esbensen, K., & Geladi, P. (1987). Principal component analysis. Chemometrics and intelligent laboratory systems, 2(1-3), 37-52.

(9) Li, C., & Wang, B. (2014). Fisher linear discriminant analysis. CCIS Northeastern University, 6.