# Condition-Based Maintenance of Naval Propulsion Systems with Supervised Data Analysis.

Archit Gupta(10319011921) AI-DS B2

*University School of Automation and Robotics (Surajmal vihar east Delhi 110092)*

**Abstract**

Considering the many variables affecting the main components of Ship Propulsion Systems, it is difficult to model their behavior and interactions a priori with physical k nowledge. Data-Driven Models (DDM's) misuse advanced statistical models to create models based on the huge amount of data scraped by systems without any theoretical knowledge. They are very useful when monitoring the propulsion based system and take decisions based on propulsion plant. This report is made to study the performance of condition based maintenance through the use of DDM's. The study includes use of supervised learning technique which require sensor data in order to be executed. A naval vessel , with diesel-electric and gas propulsion plant, has been used to collect data and show the efficiency of the approach used, because of security concerns the data that has been used for analysis here is taken from the freely available UCI repository.

**Keywords:** Data-Driven Model , Naval Propulsion System, Condition based maintenance , Data analysis , Supervised Learning.

## 1. Introduction

In the area of naval operations , the maintenance and reliability of propulsion systems plays an important role in ensuring safety and efficiency of those vessels. Condition based maintenance (CBM) is an excellent approach for monitoring critical components and guessing their useful life. The availability of huge datasets has enabled the use of supervised learning to develop accurate predictive models for CBM.

The purpose of this report is to present deep analysis of the dataset obtained from UCI repository , which is focused on the condition based maintenance of naval propulsion systems. The dataset has a rich collection of parameters that can be used a valuable resource for conducting predictive maintenance analysis.

The goal of this analysis was twofold. To begin, determine the most important aspects influencing the health of naval propulsion systems. Second, based on the identified parameters, create a trustworthy predictive model capable of reliably forecasting the remaining useful life of these systems.

Throughout the report we will talk about the steps involved in data analysis process which includes data processing , exploratory data analysis , feature selection , model development and performance evaluation. The insights derived form this analysis is worthy in enhancement of effectiveness and reliability of naval propulsion systems. By combining supervised learning techniques and the dataset obtained from UCI repository , this analysis will contribute in the field of condition based maintenance of naval propulsion systems.

## 2. Proposed Methodology

In this study , we used a supervised learning strategy, specifically the Multiple Linear Regression (MLR) model, in this study to investigate the correlations between various input parameters and the health of naval propulsion systems. MLR is a strong statistical technique that allows us to forecast system health based on several independent inputs.

### *2.1 Dataset*

The tests have been carried out by implies of a numerical test system of a maritime vessel (Frigate) characterized by a Gas Turbine (GT) drive plant. The distinctive pieces shaping the total test system (Propeller, Body, GT, Adapt Box and Controller) have been created and fine tuned over the year on a few comparative genuine drive plants. In see of these perceptions the accessible information are in understanding with a conceivable genuine vessel. In this discharge of the test system it is additionally possible to require under consideration the execution rot over time of the GT components such as GT compressor and turbines. The impetus framework conduct has been portrayed with this parameters: Transport speed (direct work of the lever position lp). - Compressor debasement coefficient kMc. - Turbine debasement coefficient kMt. so that each conceivable debasement state can be portrayed by a combination of this triple (lp,kMt,kMc). The run of rot of compressor and turbine has been inspected with an uniform lattice of accuracy 0.001 so to have a great granularity of representation. In particular for the compressor rot state discretization the kMc coefficient has been examined within the space [1; 0.95], and the turbine coefficient within the space [1; 0.975]. Transport speed has been examined examining the run of doable speed from 3 hitches to 27 hitches with a granularity of representation break even with to tree hitches. A arrangement of measures (16 highlights) which by implication speaks to of the state of the framework subject to execution rot has been procured and put away within the dataset over the parameter's space. For each record a 16-feature vector containing the GT measures at steady state of the physical asset:

Lever position (lp) [ ] , Ship speed (v) [knots] , Gas Turbine (GT) shaft torque (GTT) [kN m], GT rate of revolutions (GTn) [rpm] , Gas Generator rate of revolutions (GGn) [rpm], Starboard Propeller Torque (Ts) [kN], Port Propeller Torque (Tp) [kN], Hight Pressure (HP) Turbine exit temperature (T48) [C], GT Compressor inlet air temperature (T1) [C], GT Compressor outlet air temperature (T2) [C], HP Turbine exit pressure (P48) [bar], GT Compressor inlet air pressure (P1) [bar], GT Compressor outlet air pressure (P2) [bar], GT exhaust gas pressure (Pexh) [bar], Turbine Injecton Control (TIC) [%], Fuel flow (mf) [kg/s], GT Compressor decay state coefficient and GT Turbine decay state coefficient.

### *2.2 Data Pre Processing*

Data pre processing is done on the Dataset that we obtained from UCI repository , it has been done to make the data ready for machine learning model. This is one of the mandatory steps as it plays a very crucial role in making of robust predictive model for naval propulsion systems.

#### *2.2.1 Importing Libraries*

In order to perform we need to import some python libraries which can be used to perform specific tasks. The libraries that are used by the author for the pre processing of this dataset are :

1. Numpy : used for mathematical operations and scientific calculations in dataset.
2. Matplotlib: used for 2D plotting
3. Pandas: used for importing and management of datasets.
4. Seaborn: used for making statistical plotting.

5. Scikit learn : used for regression models and r2_score for data analysis.

### 2.2.2 *Importing Dataset*

To analyse the dataset we have to import it for the machine learning model to be applied on it. To import the dataset we used read_csv() function of pandas library by giving a specified path as an argument.

### 2.2.3 *Feature extraction*

There are 2 types of variables that are present in the dataset. Firstly independent variables that counts to 16 in the datasets which are Lever position (lp) [ ] , Ship speed (v) [knots] , Gas Turbine (GT) shaft torque (GTT) [kN m], GT rate of revolutions (GTn) [rpm] , Gas Generator rate of revolutions (GGn) [rpm], Starboard Propeller Torque (Ts) [kN], Port Propeller Torque (Tp) [kN], Hight Pressure (HP) Turbine exit temperature (T48) [C], GT Compressor inlet air temperature (T1) [C], GT Compressor outlet air temperature (T2) [C], HP Turbine exit pressure (P48) [bar], GT Compressor inlet air pressure (P1) [bar], GT Compressor outlet air pressure (P2) [bar], GT exhaust gas pressure (Pexh) [bar], Turbine Injecton Control (TIC) [%], Fuel flow (mf) [kg/s]. Secondly , dependant variables that counts to 2 in the dataset which are GT Compressor decay state coefficient and GT Turbine decay state coefficient. These features are extracted using pandas library and the method use can be seen in the figure(2.2.3.1). These features have a little correlation so there is no need for dropping the features.

```
x = dataset.drop(['17 - GT Compressor decay state coefficient.','18 - GT Turbine decay state coefficient.'],axis=1).values
y = dataset[['17 - GT Compressor decay state coefficient.','18 - GT Turbine decay state coefficient.']].values
```

The x variable contains all the independent features and y contains all dependent features.

### 2.2.4 *Handling Missing Data and Scaling of Data*

The next step of pre processing is handling of missing data , since it may cause problems for our machine learning model . Since the UCI repository contained the data which was cleaned therefore there is not need for handling as missing values as none were found. Also the distribution of data does not contains a high variance so there is no for scaling of data. The data used is not scaled for the further analysis.

### 2.2.5 *Split Dataset into Training and Testing Data*

The dataset is divided into training and testing data this will enhance the performance of our machine learning model. Training set is the one on which the model is trained so that it can be become well versed with the type of data it is going to handle for prediction, and testing set is the one which is used for testing of model so that we can figure out the reliability of our machine learning model. The task is accomplished by using a python library named as scikit-learn from which we import train_test_split function. The working can be demonstrated as

```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test = train_test_split(x,y,test_size = 0.3,random_state=0)
```

The following code shows that 70 % of data is used for training and the rest is used for testing.

## *2.3 Applying Multiple Linear Regression(MLR), Lasso , Ridge , ElasticNet and Decision Tree Regression*

Multiple linear regression (MLR), often known as multiple regression, is a statistical technique that predicts the result of a response variable using several explanatory variables. Multiple linear regression attempts to represent the linear relationship between explanatory (independent) and response (dependent) variables.

Formula and Calculation:

$$y_i = \beta_0 + \beta_1 x_{i1} + \beta_2 x_{i2} + \ldots + \beta_p x_{ip} + \epsilon$$

Where ,

$y_i$ = dependent variable,

$x_i$ = independent variable,

$\beta_0$ = y-intercept,

$\beta_p$ = slope coefficients for each independent variable,

$\epsilon$ = error term

Lasso regression is a linear regression approach that integrates a penalty component into the least squares objective function. The penalty term is calculated by multiplying the absolute values of the coefficients by a tuning parameter. This penalty promotes model sparsity by reducing some coefficients to zero. As a consequence, Lasso regression does feature selection, which means that it automatically picks important characteristics while removing irrelevant ones from the model. It is especially beneficial when dealing with high-dimensional datasets that may contain a large number of unnecessary or duplicated characteristics.

Ridge regression, also known as Tikhonov regularisation, is yet another linear regression approach for dealing with overfitting. It does this by introducing a penalty component into the least squares objective function, which is the total of the squared coefficient values multiplied by a tuning parameter. This penalty term pushes the coefficients to be modest but does not completely remove them. Ridge regression provides a trade-off between model simplicity and accuracy by reducing the coefficients towards zero but not precisely zero. It is especially useful for dealing with multicollinearity, which occurs when the predictors are highly linked.

Elastic Net regression is a hybrid of the Lasso and Ridge regression methods. It introduces a penalty term that is a linear mixture of the absolute and squared coefficient values, with two tuning parameters determining the intensity of each penalty. This combination enables Elastic Net regression to benefit from both Lasso's feature selection capacity and Ridge's multicollinearity handling. The tuning settings may be adjusted to regulate the balance between sparsity and model stability. Elastic Net regression is very beneficial when dealing with datasets with a high number of features and predictors that are multicollinear.

A decision tree regressor is a non-linear regression approach that predicts the target variable using a tree-like model. Each internal node of the tree represents a feature, and the tree divides the data depending on their values. Splitting is done in such a way that the variance within each resultant subgroup is minimised. The procedure is repeated until a stopping requirement, such as reaching a maximum depth or a minimum quantity of samples per leaf, is fulfilled. The projected values for the target variable are stored in the tree's leaf nodes. Decision tree regressors are useful for capturing complicated connections and interactions between features, and they can handle both numerical and categorical predictors. They are, however, prone to overfitting if not adequately trimmed or regularised.

The MLR, Lasso, Ridge, ElasticNet, Decision Tree regressions are applied on the dataset to create a machine learning model that can be used for prediction. This is achieved by using a library named scikit-learn and using Regression models with fit function for training of dataset.

The prediction is done using the predict function by giving testing dataset as an argument.

## 3. Experimental Result

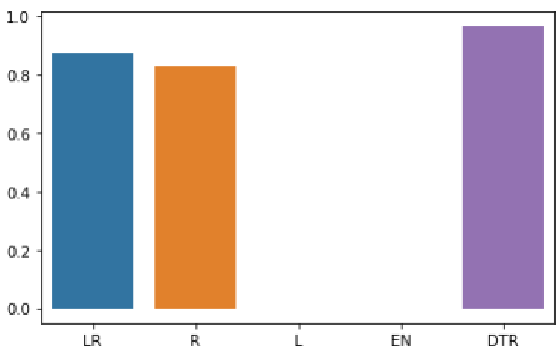
In this section the author will report the results obtained by applying regression model to the condition based maintenance of the naval propulsion system based on the regression framework and data used as described in section 2. The dataset was divided into training and testing dataset as reported in the above section. The performance of MLR, Ridge, Lasso, Elastic Net, Decision Tree Regressor were measured according to the $R^2$ score measured by the author. As a result 96% confidence interval was reported.

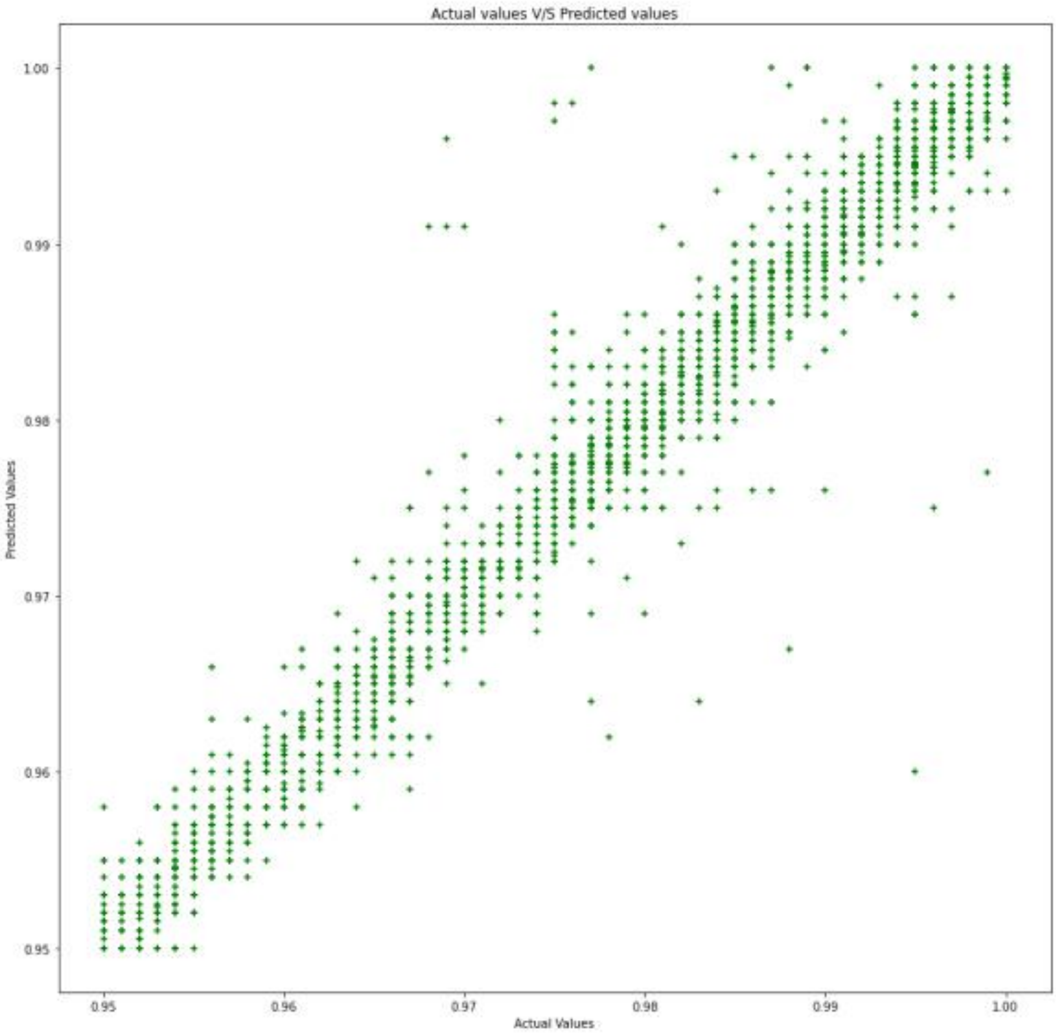
The results can be seen in the following figure.

```
R2_SCORES
LR :- 0.873276199651367
R :- 0.8294719901685492
L :- -0.0007405519833493246
EN :- -0.0007496498732133539
DTR :- 0.9668534237127835
dict_keys(['LR', 'R', 'L', 'EN', 'DTR'])
dict_values([0.873276199651367, 0.8294719901685492, -0.000740551983349324
6, -0.0007496498732133539, 0.9668534237127835])

Out[64]:

<AxesSubplot:>
```






For $R^2$ score scikit-learn library and metrices sub library was exploited. For plotting the graph matplotlib library was exploited.

## 4. Conclusion

The behavior and interaction of the main components of naval propulsion system cannot be effectively modeled with a priori knowledge , considering large number of variables influence them. In reality, in this work, author showed that by using strategies and the huge amount of authentic information collected by the current on-board mechanization frameworks it is possible to construct successful Data-Driven Models which don't require any a priori information. In conclusion, this study proved that it is possible to treat a Condition Based Maintenance problem by using regression techniques. These models can be used for real-time applications to easily and quickly identify maintenance necessities. The proposed model is very compelling and solid, and, for this reason, the author's approach can accurately distinguish the state of the system and can clearly identify when maintenance is required. This can reduce the operational and maintenance cost. To begin with, directed MLR require a huge sum of named information to proficiently screen the naval propulsion system components state, which would suggest the operational halt or indeed the drydocking of the vessel. Moreover, in the future, author will focus on the development of supervised and unsupervised methods which could solve this problem and allow a simplification of the data collection procedure.

## References

[2] M. Altosole, G. Benvenuto, M. Figari, U. Campora, Real-time simulation of a cogag naval ship propulsion system, Proceedings of the Institution of Mechanical Engineers, Part M: Journal of Engineering for the Maritime Environment 223 (1) (2009) 47-62.

UCI machine learning repository.

F. Cipollini *et al.* Condition-based maintenance of naval propulsion systems with supervised data analysis

G. Huang *et al.* Trends in extreme learning machines: a review

N.P. Kyrtatos *et al.* Shipboard engine performance assessment by comparing actual measured data to nominal values produced by detailed engine simulations

https://scikit-learn.org/stable/supervised_learning.html#supervised-learning

https://matplotlib.org/stable/plot_types/basic/scatter_plot.html

https://scikit-learn.org/stable/model_selection.html

https://scikit-learn.org/stable/modules/generated/sklearn.metrics.r2_score.html#sklearn.metrics.r2_score

https://pandas.pydata.org/docs/user_guide/io.html#io-read-csv-table

V.N. Vapnik Statistical learning theory

A. Coraddu *et al.* Machine learning approaches for improving condition-based maintenance of naval propulsion plants

R.K. Mobley An introduction to predictive maintenance

Y. Peng *et al.* Current status of machine prognostics in condition-based maintenance: a review

https://numpy.org/doc/stable/user/whatisnumpy.html

https://www.investopedia.com/terms/m/mlr.asp

A.H.C. Tsang Condition-based maintenance: tools and decision making

Friedman, J., Hastie, T., & Tibshirani, R. (2001). The elements of statistical learning. Springer series in statistics Springer, Berlin.