# Statistical Learning Methods Assignment 2021/22

Tanmay Kacker - 372465
tanmay.kacker.465@cranfield.ac.uk
Applied Artificial Intelligence MSc

Autumn 2021

# 1 Introduction

## 1.1 Problem Brief

Predictive maintenance refers to the set of techniques used to determine the operational condition of a system and estimate the maintenance time frame using predictive models. This helps reduce the operational costs of maintenance and prevent untimely failures which could lead to extended downtimes.

An important aspect of predictive maintenance is failure prediction using historical data. Analysis of sensor and telemetry data can be used to predict the Time-To-Failure (TTF) of a system which allows for the planning of a maintenance schedule. This is particularly useful for aircraft manufacturers and operators.

The aim of this report is to describe the usage of statistical learning techniques to historical sensor measurement data to predict the future failures of engines. Machine learning techniques can be used to establish relationships between sensor measurements and historical failures. The goal is to make the following predictions:

1. The Time-To-Failure (TTF) of an engine.
2. Classify whether an engine will fail in a given time period.

## 1.2 Data

The data used for this report is a subset of a larger dataset generated by Microsoft and consists of run-to-failure scenarios for a number of aircraft engines. The following data files were provided for building the predictive maintenance models:

1. `train_selected.csv` - contains the historical sensor measurements and failure data for multiple engines over cycles of operation
2. `test_selected_ttf.csv` - contains the historical sensor measurements and failure data for multiple engines at a randomly selected cycle of operation. This will be used to quantify the accuracy of the models.

A detailed analysis of the data will be provided as a part of the data pre-processing and exploratory data analysis (EDA) sections of the report.

## 1.3 Problem Formulation

In order to achieve the goals of the report supervised learning techniques will be used to establish relationships between sensor measurements and historical failures. The two requirements of the problem will be formulated as follows:

1. Regression models to predict the continuous dependent variable `TTF`
2. Binary classification models to predict whether an engine will fail in a given time period.

## 1.4 Development Environment

1. Integrated Development Environment with `Visual Studio Code (1.62.3)` and `Python (3.9.1)`
2. Data processing with `Pandas (1.3.4)` and `Numpy(1.21.3)`
3. Data visualisation with `Seaborn (0.11.2)` and `Yellowbrick (1.3.post1)`
4. Machine learning with `Scikit_learn (1.1.dev0)`

# 2 Data pre-processing

## 2.1 Data Summary

The training and testing dataset, provided as CSV files, were imported into a dataframe using the `Pandas` library. Printing the samples of the dataset using `head` and `tail` functions makes it easier to understand the structure of the data. The columns from the original dataset were renamed to make the data more readable.

| | ID | CYCLE | S1 | S2 | S3 | S4 | TTF | TTF_LABEL |
|---|---|---|---|---|---|---|---|---|
| 0 | 1 | 1 | 1400.60 | 554.36 | 47.47 | 521.66 | 191 | 0 |
| 1 | 1 | 2 | 1403.14 | 553.75 | 47.49 | 522.28 | 190 | 0 |
| 2 | 1 | 3 | 1404.20 | 554.26 | 47.27 | 522.42 | 189 | 0 |
| 20628 | 100 | 198 | 1428.18 | 550.94 | 48.09 | 520.01 | 2 | 1 |
| 20629 | 100 | 199 | 1426.53 | 550.68 | 48.39 | 519.67 | 1 | 1 |
| 20630 | 100 | 200 | 1432.14 | 550.79 | 48.20 | 519.30 | 0 | 1 |

Figure 1: First and last 3 rows in the training dataset

The dataset has 20,631 rows and comprises the following columns:

- `ID` - the unique identifier for each engine ranging from 1 to 100.
- `CYCLE` - The cycle of operation for the engine ranging from 1 to the cycle where failure happened.
- `S1, S2, S3 and S4` - the sensor readings for the engine in the cycle.
- `TTF` - the number of remaining cycles before the engine fails.
- `TTF_LABEL` - 1 if the engine will fail in the cycle, 0 otherwise. An engine is flagged as 1 if the number of remaining cycles is less than or equal to 30.

### 2.1.1 Test Data

The test dataset was created as a hold out set and consists of 100 rows where each row consists of sensor measurements randomly selected from the engines in the training dataset. The expected regression values `TTF` and classification labels `label_bnc` are also provided.

## 2.2 Missing values

Missing values in the dataset can hinder in the analysis of the data and the operation of the predictive models. If there are missing values in the dataset, either the data can be imputed or the data points can be dropped if the count is below a certain threshold. The dataset was checked for missing values using the `info` function from the `Pandas` library.

As can be seen from the output in Figure 2, there are no missing values in the dataset.

```
Data columns (total 8 columns):
 #   Column     Non-Null Count  Dtype
---  ------     --------------  -----
 0   ID         20631 non-null  int64
 1   Cycle      20631 non-null  int64
 2   S1         20631 non-null  float64
 3   S2         20631 non-null  float64
 4   S3         20631 non-null  float64
 5   S4         20631 non-null  float64
 6   TTF        20631 non-null  int64
 7   TTF_LABEL  20631 non-null  int64
```

Figure 2: Count of missing values in the dataset

## 2.3 Outlier Detection

Outlier are points in the dataset that significantly distant from the rest of the data or the inliers. They can greatly hamper the accuracy of the predictive models. In order to check if the data has outlier, the `boxplot` function from the `Seaborn` library was used. The box and whisker plot are a non-parametric method to visualize the distribution of the data where the box starts at the first quartile and ends at the third quartile. The middle line represent the median or the 50th percentile. The whisker extend to 1.5 times the interquartile range (IQR) on each sides of the box. If points are outside the box, they are plotted individually and are considered as outliers.

It can be seen from the box plots in Figure 3 that each of the sensor readings have a certain amount of outliers. These will have to be identified and removed to improve the accuracy of the predictive models.

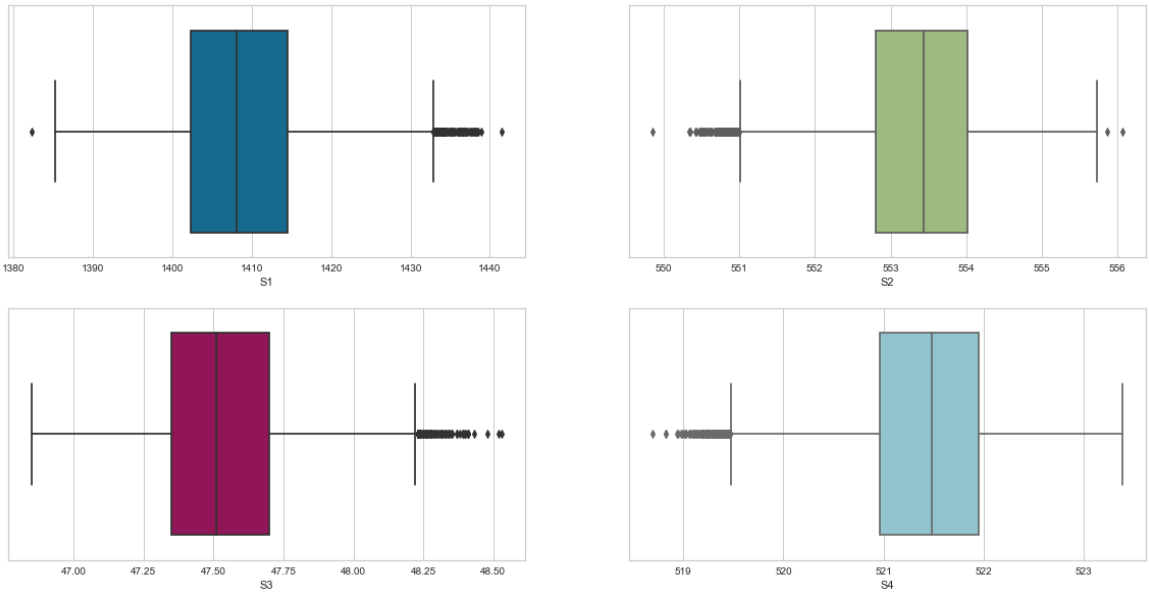



Figure 3: Box plots of sensors showing distribution of sensor readings and outliers

To quantitatively identify the outliers, the following univariate and multivariate statistical tests were used:

- `Numeric Outlier Technique` - A non-parametric technique which quantifies the approach presented in the box and whisker plot above. Any data point outside the sum of upper or lower quartile plus or minues 1.5 times the interquartile range is considered an outlier. Here, 1.5 is used as the IQR multiplier. 427 rows were identified as outliers which is around 2% of the dataset.
- `Z-score` - The Z-score is a parametric method that indicates the number of standard deviations a data point is away from the mean. The absolute value of the Z score should be greater than a certain threshold for a data point to be deemed as an outlier. Here, the threshold was set to 3. 115 rows were identified as outliers which is around 0.6% of the dataset.
- `Mahalanobis Distance` - A multivariate generalization of calculating the distance of a point from its mean in standard deviations using the covariance matrix of the component variables. A brief description of the outlier detection technique can be found in the next subsection.
- `Cook's distance` - Cook's distance is especially relevant when doing ordinary least squares regression and measures the influence of a data point to the results using the residuals and the mean squared error. A brief description of the outlier detection technique can be found in the last subsection.

### 2.3.1 Mahalanobis Distance

Add details and plot for MD or remove subsection and add to the list.

### 2.3.2 Cook's Distance

As mentioned above, Cook's distance is a measure of the influence of a data point on the estimated OLS regression coefficients. It is a function of the standardized residuals and the leverage associated with each data point. Once the influence is computed, outliers are identified by setting the threshold as $\frac{4}{n}$ where `n` is the number of data points. In order to visualize the influence, the function `CooksDistance` was used from the `Yellowbrick` library. The function plots as a stem plot where the x-axis is each data point and the y-axis is the corresponding influence or the Cook's distance as shown in Figure 4.

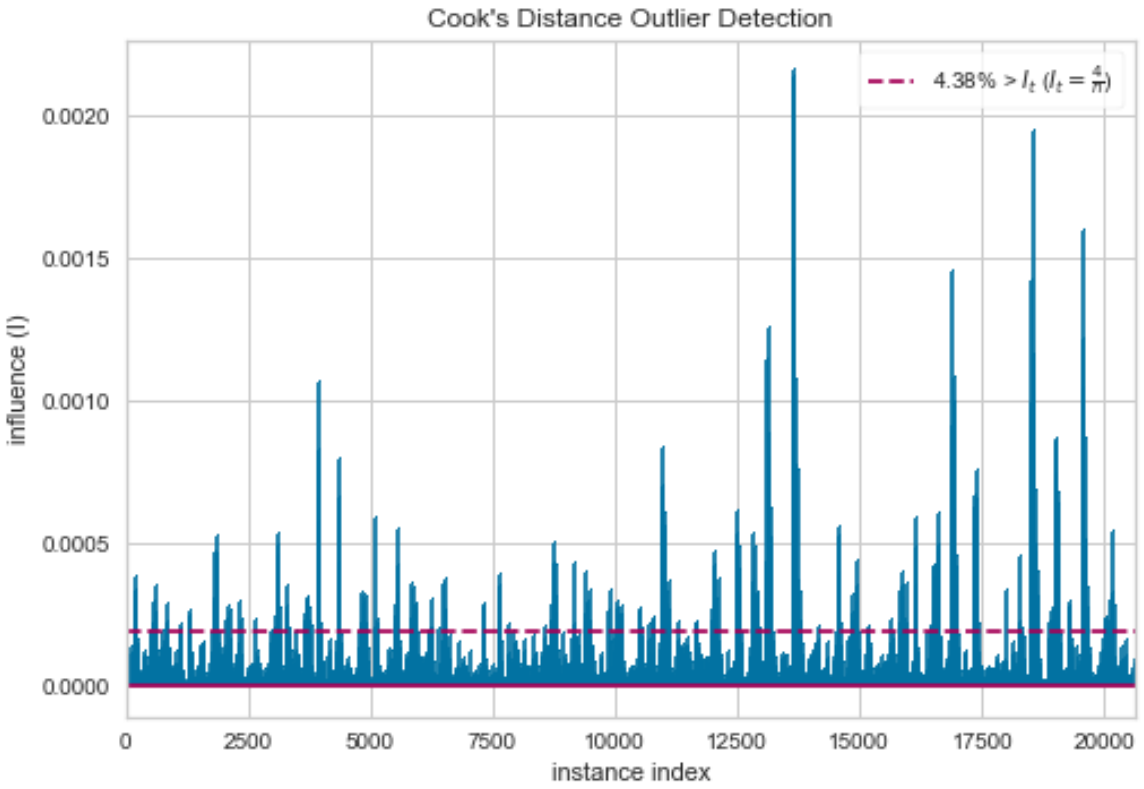



Figure 4: Detecting outliers in the dataset using Cook's distance

As seen above, 4.38% of the data points are outliers which corresponds to 904 rows having more than $1.94 \times 10^4$ influence where the latter is the threshold for identifying outliers. Removing the outliers from the dataset will improve the accuracy of the linear regression model which will be seen in the regression section. The dataset was pruned to remove the outliers using the `drop` function from the `Pandas` library and stored as a CSV file for further usage.

## 2.4 Feature extraction

Test rolling mean and standard deviation

## 2.5 Summary

- The dataset has four columns with sensor measurements - `S1, S2, S3 and S4` along with engine identifier `ID` and current cycle of operation `CYCLE`.
- There are also regression values for the engine failure time `TTF` and the classification labels for the engine failure `TTF_LABEL`.
- There are 100 engines with 20,631 and 100 rows in the training and test datasets respectively.
- The dataset has no missing values.

- Outliers were detected in the dataset and subsequently removed.
- 4.38% of the data points were identified as outlier by Cook's distance and were specifically removed for OLS regression.

# 3 Exploratory Data Analysis

## 3.1 Descriptive Statistics

It was previously identified that the training dataset has 20.631 rows with 8 columns corresponding to the data of 100 engines. The sensor measurements are continuous values while the dataset does not contain any categorical values except for the classification labels. The columns `ID and CYCLE` serve as the indexing columns and their usage in training the model could lead to target leakage. We shall avoid using these columns unless the hypothesis is invalidated.

### 3.1.1 Central Tendency and Dispersion

To generate the summary of central tendency and variability of the data, the `describe` function from the `Pandas` library was used. Since for each engine there is a row for each cycle of operation, the dataset was grouped by the engine identifier and the maximum value of cycle was extracted for generating the summary statistics. This corresponds to the maximum cycle of operation for each engine or the `TTF`. The following key observations were made about the `ID and CYCLE` columns:

- The values of `ID` range from 1 to 100 but the mean and quartiles do not line up indicating each engine has a variable maximum number of cycles.
- The above disparity is elucidated by the statistics of the aggregated cycle column where the average engine failure is around 199 - 206 cycles.
- Minimum number of cycles that an engine ran for before failure is 128 and the maximum is 306 with a standard deviation of around 46.34 cycles.

The above statistics can be further verified by plotting the histogram of the aggregated cycle column using the `histplot` function from the `Seaborn` library.

Try wrap figure

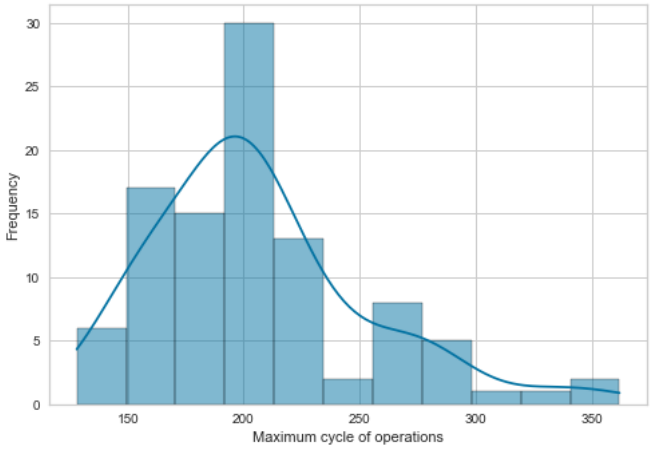
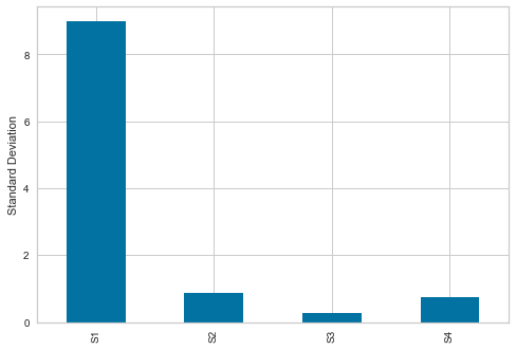


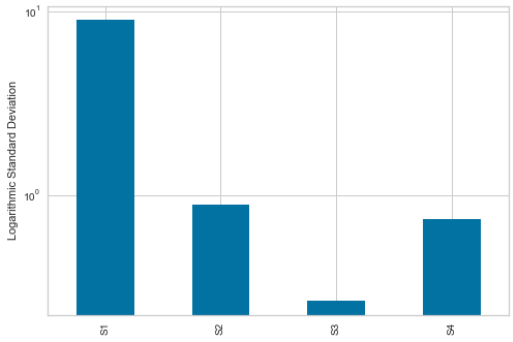

Figure 5: Histogram of the maximum cycle of operation for each engine

Understanding the variability of the sensor measurements is extremely important to identify the features to be selected. If a feature does not show any significant variability, it is likely to not affect the dependent variable. The standard deviation of the sensor measurement was calculated using the `std` function from the `Pandas` library and plotted using the `plot` function as shown in Figure 6.

`S1` has a significant standard deviation of around 9 while `S2` and `S4` have a standard deviation of around 0.89 and 0.74 respectively. `S3` has the least standard deviation of around 0.26. Analysis of the correlation of the sensor measurements with the dependent variable `TTF` would lead to the selected set of features.




(a) Standard deviation of the sensor variables



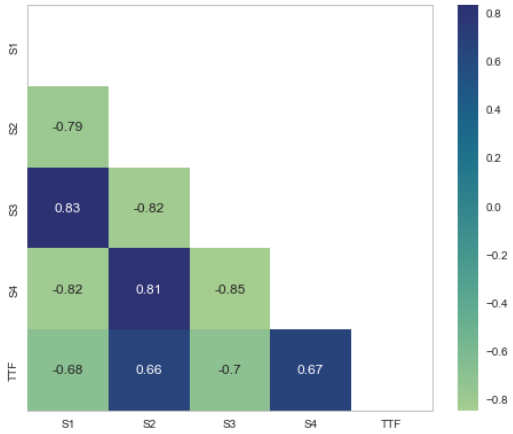

(b) Logarithmic standard deviation of the sensor variables

Figure 6: Variability of sensor measurements

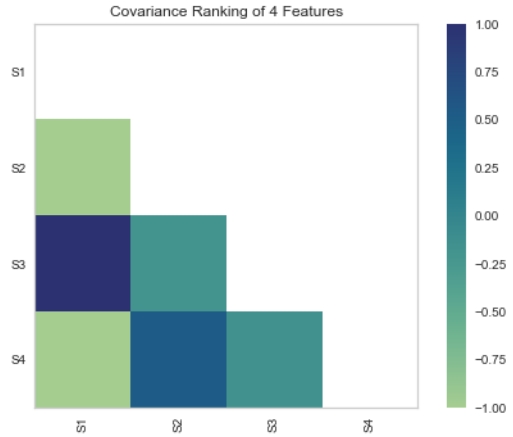
### 3.1.2 Correlation

Correlation is a measure of the linear dependence between two variables. A greater correlation between dependent and independent variables indicates a strong predictive capability. Whereas multicollinearity between the independent variables is not desirable as it can lead to overfitting and increase the dimensionality of the model. Regression models are specifically sensitive to correlated variables.

To analyse the correlation between the variables, Pearson's correlation coefficient was used from the `Pandas` library and a heat map was generated using the `heatmap` function from the `Seaborn` library. Pearson's R ranges from -1 to 1 with a value of 0 indicating no correlation. Other measures such as Spearman's R and Kendall's Tau yielded similar results. The covariance of the independent variables was also calculated using the `cov` function from the `Pandas` library and plotted.




(a) 1a




(b) 1b

Figure 7: plots of....

The following are the major observations that are clearly visible from the correlation and covariance matrix in Figure 6:

- `S2` and `S4` are positively correlated with `TTF` with scores of 0.66 and 0.67 respectively.
- `S1` and `S3` are negatively correlated with `TTF` with scores of -0.68 and -0.7 respectively.
- (`S1`, `S3`) and (`S2`, `S4`) have significant positive correlations within each other with scores of 0.83 and 0.81 respectively.
- (`S1`, `S4`) and (`S3`, `S4`) have significant negative correlations within each other with scores of -0.82 and -0.85 respectively.

The correlation with the dependent variable is indicative of good predictive capability and is desirable. However, the correlation between the independent variables can hamper the performance of the model and thus will have

to be resolved. These features will either have to be removed during the feature selection process or the redundancy could be reduced using principal component analysis (PCA) where the features are assimilated and dimensions are reduced.

## 3.2 Visual EDA

The pairwise relationships were visualized using the `pairplot` function from the `Seaborn` library. The diagonal shows the univariate marginal distributions of the variables.

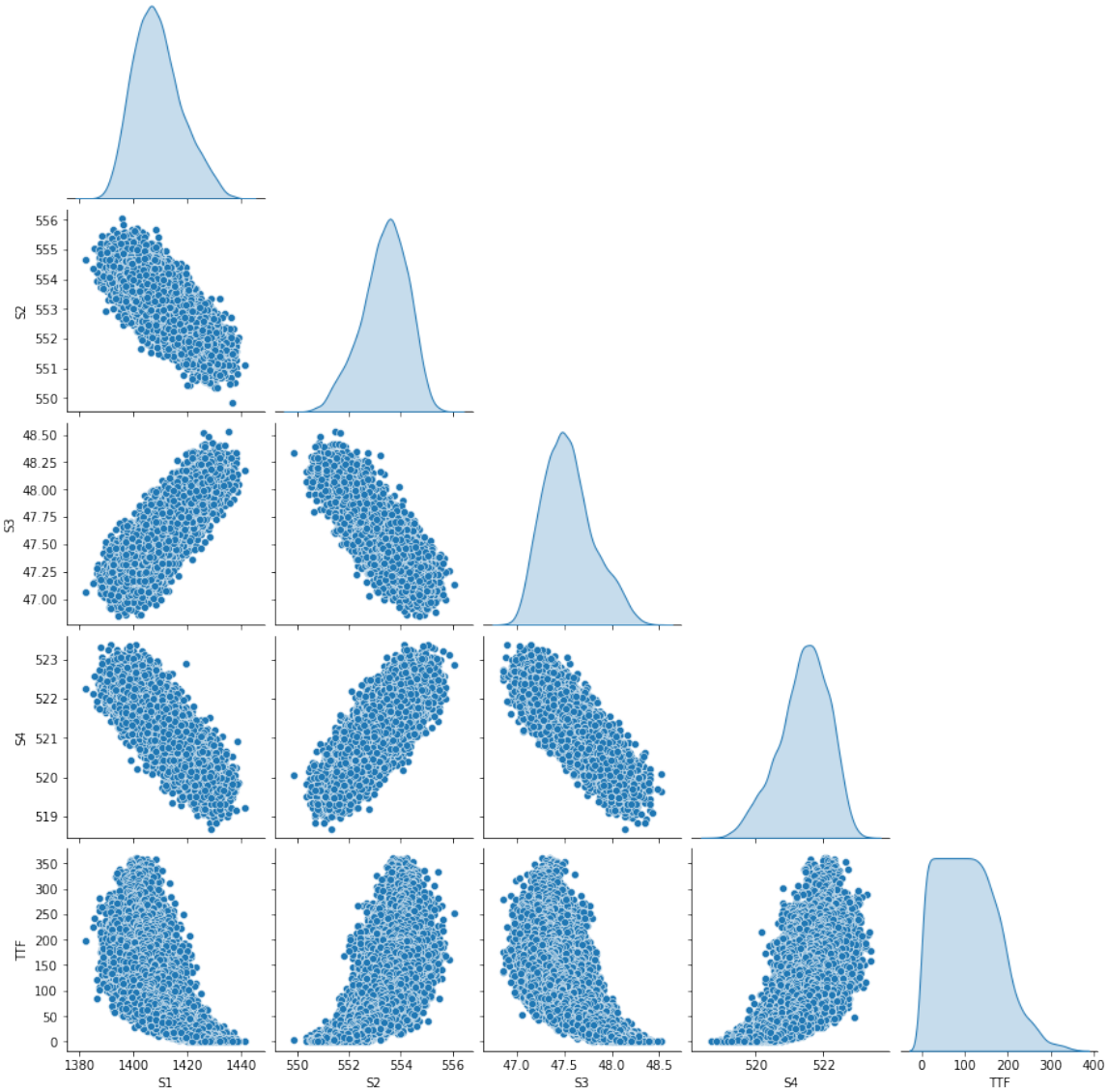
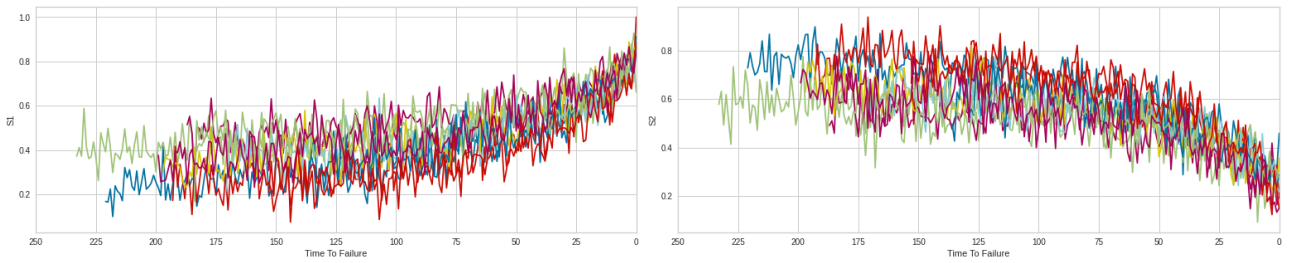



Figure 8: Pairwise relationships between dependent and independent variables

The significant observations that can be made from the plot in Figure 8 are:

- All the dependent variables display to have a normal distribution which is desirable due to the assumption of normality in parametric methods.
- The correlation between the sensor measurements and `TTF` is apparent as computed in the correlation matrix.
- The correlation between the sensor measurements and `TTF` not linear but of the polynomial nature.
- The linear correlation between the independent variables is apparent as computed in the correlation matrix.




(a) Measurements of S1

(b) Measurements of S2

Figure 9: Plots of signals from Sensor 1 and 2 sampled from 10 random engines

The correlation between sensor measurements and `TTF` is also visible in the plot of the measurements from Sensor 1 and 2 from 10 randomly sampled engines. As the engines move towards failure, measurements from `S1` spike whereas `S2` tends to have lower values near failure.

Another interesting observation is that initially the measurements from the sensor tend to be constant and beyond a certain threshold is when the slope gets steeper. This could be used to improve model accuracy via pruning the values before the threshold, which seems like somewhere between 150 and 125 cycles.

### 3.2.1 Skewness and Kurtosis

Since, the distribution of the dependent variables is normal, but there is some visible skewness, we need to analyse the extent of it and see if corrective measures need to be taken. The skewness and kurtosis were calculated using the `skew` and `kurt` functions from the `Pandas`. The following was observed from the results:

- `S1` and `S3` are positively skewed with scores of 0.44 and 0.47 respectively.
- `S2` and `S4` are negatively skewed with scores of -0.39 and -0.44 respectively.
- All the variables are mildly platykurtic with kurtosis in the range of -0.14 to -0.16.

Skewness can be fixed by transformations namely the box-cox transformation. The skewness of the features is not significant enough to be corrected, and thus the features are not transformed.

## 3.3 Classification

The target variable `TTF_LABEL` is a categorical variable with two possible values - `0` and `1`. The target variable is used to classify if the engine will fail in a given period. Checking the support for each class, we see that the number of instances of `0` is much higher than the number of instances of `1`. The value `0` of the classification label nearly accounts for 85% of the dataset which can be seen in Figure 10. This is known as class imbalance. This cane be corrected by performing rebalancing or stratified sampling. A consequence of this is that accuracy will not be a good measure of the model performance and hence will not be used for model selection and evaluation.

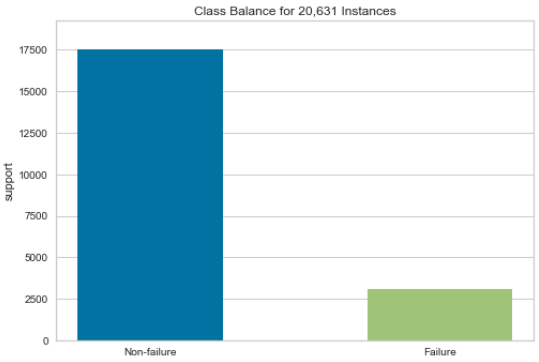
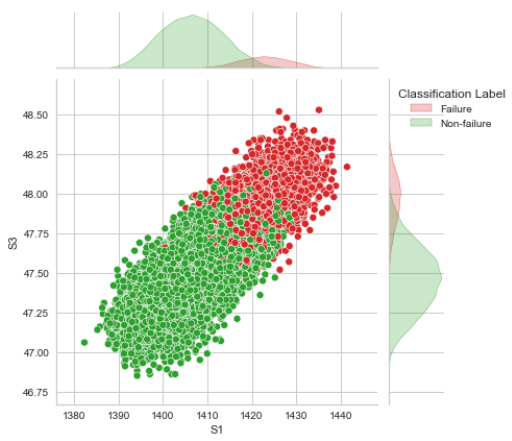
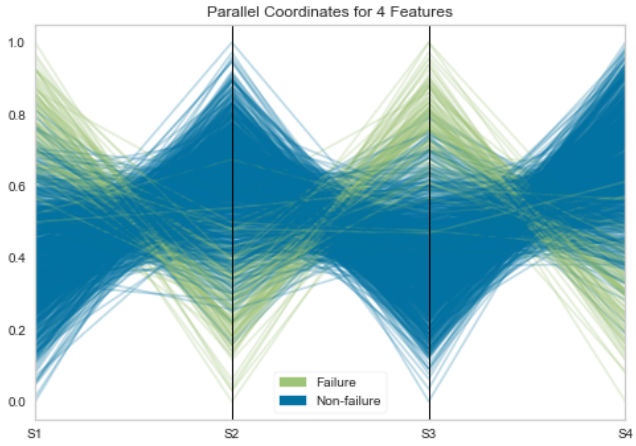



Figure 10: Visualization of classification imbalance in the training data

Describe classificat little bit




(a) Scatter plot of S1 and S3 with TTF_LABEL




(b) Parallel coordinates of sensors with TTF_LABEL

Figure 11: Plots showing separability among the classes

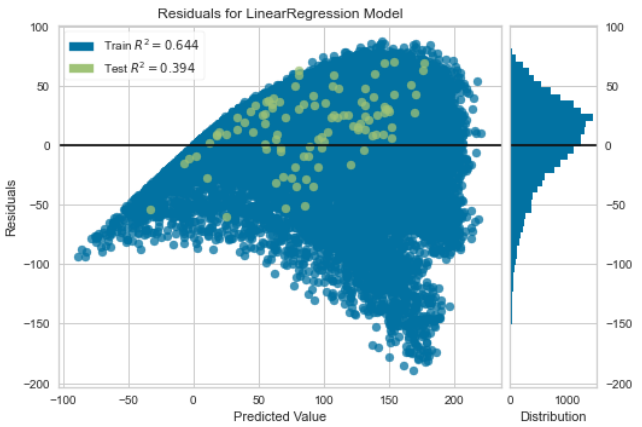
# 4 Regression Models - Estimating the Time-To-Failure (TTF)

## 4.1 Linear Regression

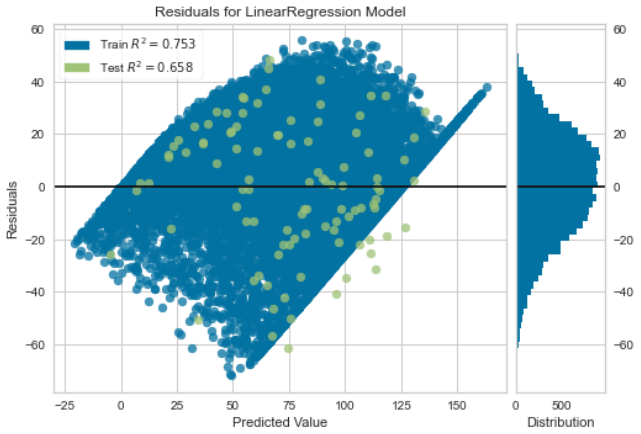
Standard ordinary least squares (OLS) regression was used to set up the baseline model. The initial model was trained on the original training dataset and then on the dataset which was normalized and had the outliers removed. The model was evaluated using the `R2` and `root mean squared error (RMSE)` metrics.

| | Linear Regression Base (Training) | Linear Regression Base (Test) | Linear Regression (Training) | Linear Regression (Test) |
|---|---|---|---|---|
| R2 | 0.644452 | 0.393569 | 0.753062 | 0.657937 |
| RMSE | 41.071267 | 32.360897 | 20.714950 | 24.304287 |

Figure 12: First and last 3 rows in the training dataset




(a) Scatter plot of S1 and S3 with TTF_LABEL




(b) Parallel coordinates of sensors with TTF_LABEL

Figure 13: Plots showing separability among the classes

Describe the linear regression residuals

## 4.2 Polynomial Regression

As mentioned in Section 3.2, the relationship between sensor measurements and `TTF` is non-linear and hence a polynomial regression model would perform better than linear models. In order to test this hypothesis, polynomial features of the dependent variables were created and the model was trained on the dataset. Hyperparameter tuning was done through Grid search to identify the best polynomial degree to use. In order to evaluate each candidate cross validation across 5 folds was used.

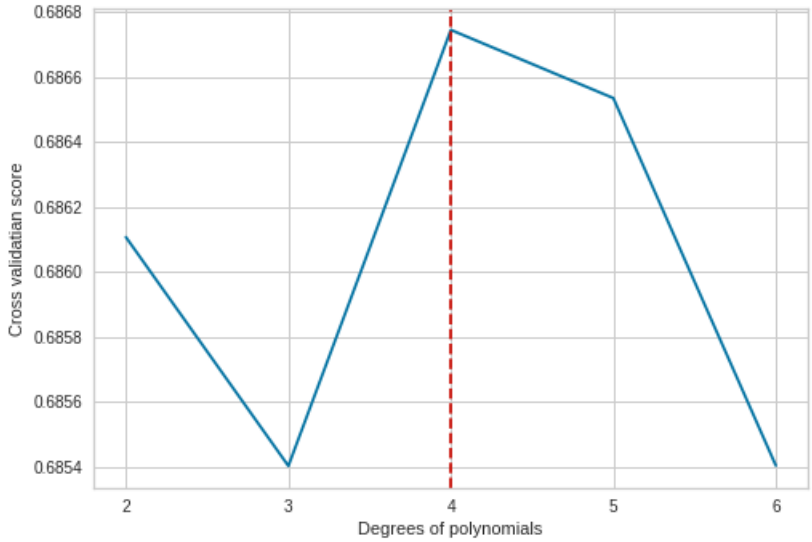
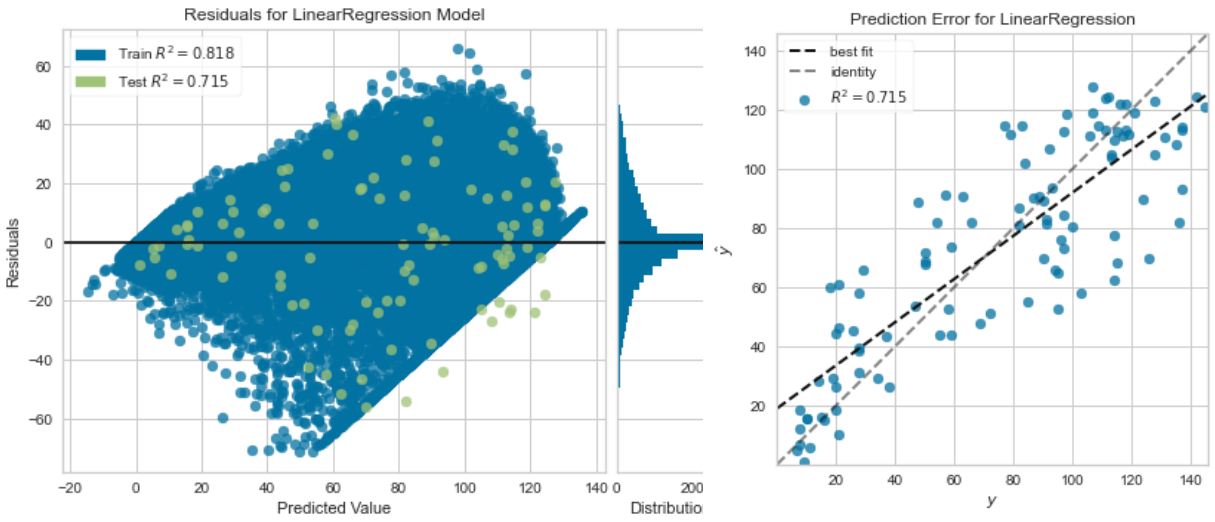



Figure 14: Grid search for finding the best polynomial degree

As seen in Figure 14, the best polynomial degree is 4. The polynomial features were created using the `PolynomialFeatures` class from the `sklearn` library. The model was trained and evaluated using the `R2` and `root mean squared error (RMSE)` metrics.




(a) Scatter plot of S1 and S3 with TTF_LABEL

(b) Parallel coordinates of sensors with TTF_LABEL

Figure 15: Plots showing separability among the classes

The following are the key observations from the result metrics and plots:

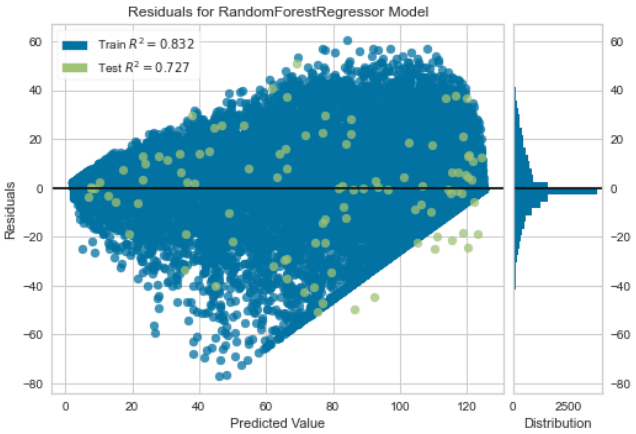
- The model had a R2 score of 0.812 and an RMSE of 18.099 on the training dataset.
- The model had a R2 score of 0.735 and an RMSE of 21.401 on the test dataset which is 14% better than the baseline model.
- The residuals are randomly spread around the horizontal axis and are fairly normally distributed.

Describe t polynomia regression residuals

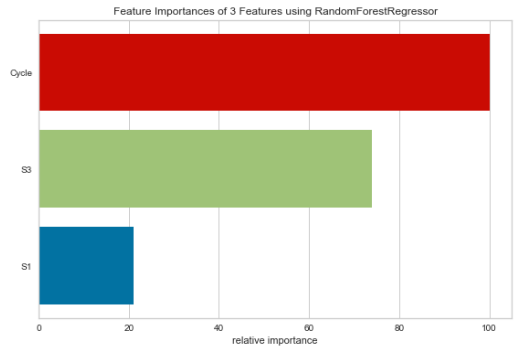
## 4.3 Random Forest Regression

As mentioned in Section 3.1.2, the sensor measurements exhibit collinearity which can hamper the performance of predictive models. Random forest regression is a non-parametric method which is quite robust to correlated dependent variables and even outliers. `RandomForestRegressor` from the `sklearn` library as the model of choice

and hyperparameter tuning was done through Grid search with cross-validation across 5 folds. Random forest has quite a few hyperparameters that need to be tuned to avoid overfitting where the effect of `n_estimators`, `max_features` and `max_depth` is quite significant. These determine the structure of the decision trees and the splitting criterion. The following are the key observations from the result metrics and plots:




(a) Scatter plot of S1 and S3 with TTF_LABEL




(b) Parallel coordinates of sensors with TTF_LABEL

Figure 16: Plots showing separability among the classes

- The model had a R2 score of 0.832 and an RMSE of 17.069 on the training dataset.
- The model had a R2 score of 0.727 and an RMSE of 21.717 on the test dataset which is 14% better than the baseline model.
- The residuals are randomly spread around the horizontal axis and are fairly normally distributed.

## 4.4 Conclusions

# 5 Classification Models - Predicting whether an engine will fail in a given time period

# 6 Conclusion

## 6.1 Summary

## 6.2 Next Steps

# Appendix I