

---

# COMP1801 - Machine Learning Coursework Report

---

Richard Raja James Michael – 001370307  
Word Count: 2860 (excluding references)

## Part 1 - Executive Summary

This report discusses how the lack of the manufacturing strategy and the complexity of using the new experimented metal alloys that are sensitive to processing parameters affects a fictitious company producing metal parts. Such sensitivities cause defects which in turn lower the usable life span of the produced part and makes them to be scrapped thus being very costly.

The main goal of this work is to implement a machine learning model capable of predicting the service life of these metal parts based on controllable processing indices. In other words, the ability to achieve this goal can help the company improve many aspects of its operations, which may range from reducing defects, to increasing the quality of the products that it offers.

For this purpose, in the present work several machine learning tools are employed on the data set consisting of processing parameters and measurements on finished parts. The methods include Ridge Regression, Random Forest Regressor, Logistic Regression for classification and Random Forest Classifier. All of the models are then tested using regression metrics like Mean Absolute Error (MAE), Mean Squared Error (MSE), R squared for regression problem and Classification metrics like accuracy, precision, recall, F1 Score.

The features that engulf life expectancy are revealed through a bigger R-squared of the Random Forest Regressor rather than the Ridge Regression one, classifying it as a better predictor. Further, it is observed that the Random Forest Classifier has better accuracy than Logistic Regression to classify the parts based on the lifespan limit.

Therefore, for the same reasons, it is suggested that the company should employ the Random Forest models for the lifespan predicting and defect categorizing as these models have better team accuracy rates. Such a machine learning technique will help the firm to increase the production line, minimize wastage, and produce quality metal parts.

## Part 2 - Data Exploration

### Loading the Dataset

The data was imported in the data frame format of the language Python using a data analysis tool box known as Pandas. The command used to load the dataset is as follows:

```
df = pd.read_csv('COMP1801_Coursework_Dataset.csv')
```

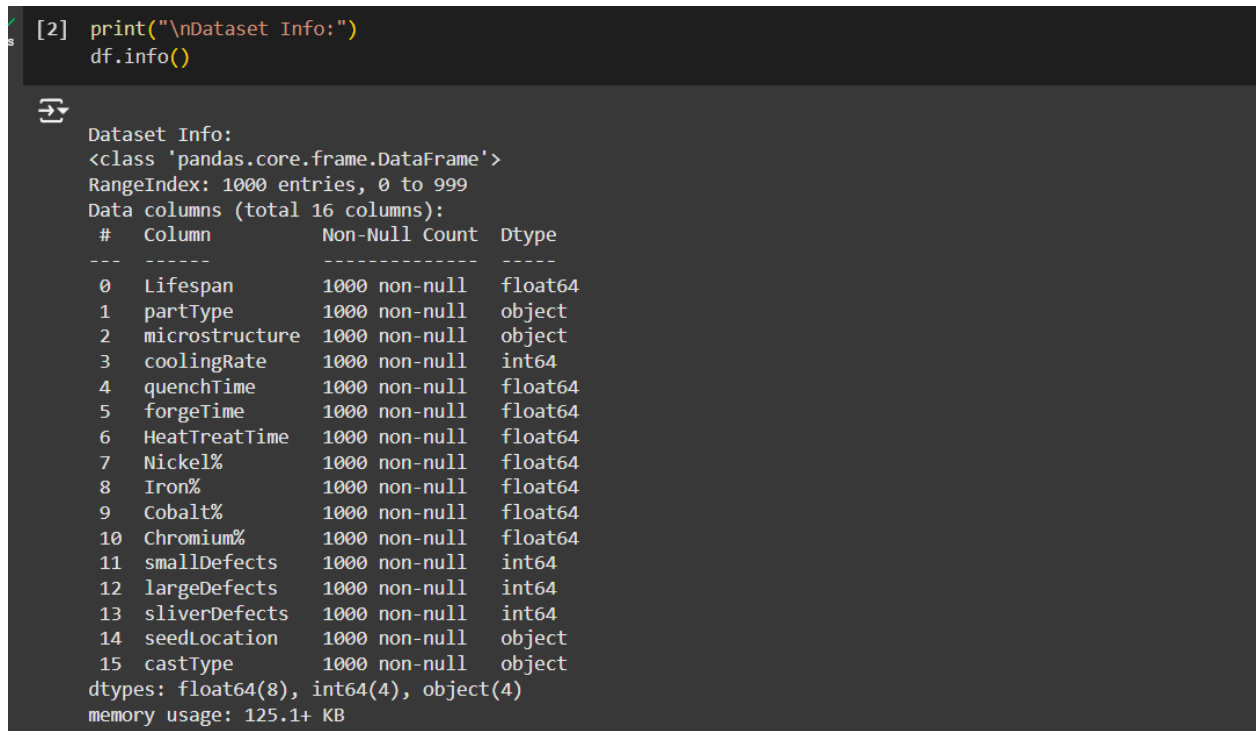
This command simply reads the CSV file full of features relevant to production parameters of some chains and lifespan of the metal parts involved and all sorts of other features.

## Exploring the Data

An overview of the data set was done to orientate the data and to look for trends and associations that could influence the lifespan of the metal parts.

1. **Dataset Overview:** The dataset here include several numbers of columns which represent different categorical and numerical varieties.. Key features include:
  - **Categorical Features:** partType, microstructure, seedLocation, castType
  - **Numerical Features:** coolingRate, quenchTime, forgeTime, HeatTreatTime, Nickel%, Iron%, Cobalt%, Chromium%
  - **Target Variable:** Lifespan

```
[2] print("\nDataset Info:")
df.info()
```

The image shows a Jupyter Notebook cell with the command `df.info()` executed. The output displays the dataset's structure, including the number of entries (1000), the number of columns (16), and a detailed breakdown of each column's data type and non-null count. The columns are: Lifespan (float64), partType (object), microstructure (object), coolingRate (int64), quenchTime (float64), forgeTime (float64), HeatTreatTime (float64), Nickel% (float64), Iron% (float64), Cobalt% (float64), Chromium% (float64), smallDefects (int64), largeDefects (int64), sliverDefects (int64), seedLocation (object), and castType (object). The output also shows the memory usage as 125.1+ KB.

Dataset Info:  
<class 'pandas.core.frame.DataFrame'>  
RangeIndex: 1000 entries, 0 to 999  
Data columns (total 16 columns):

#	Column	Non-Null Count	Dtype
0	Lifespan	1000 non-null	float64
1	partType	1000 non-null	object
2	microstructure	1000 non-null	object
3	coolingRate	1000 non-null	int64
4	quenchTime	1000 non-null	float64
5	forgeTime	1000 non-null	float64
6	HeatTreatTime	1000 non-null	float64
7	Nickel%	1000 non-null	float64
8	Iron%	1000 non-null	float64
9	Cobalt%	1000 non-null	float64
10	Chromium%	1000 non-null	float64
11	smallDefects	1000 non-null	int64
12	largeDefects	1000 non-null	int64
13	sliverDefects	1000 non-null	int64
14	seedLocation	1000 non-null	object
15	castType	1000 non-null	object

dtypes: float64(8), int64(4), object(4)  
memory usage: 125.1+ KB

2. The first look at the summary of the obtained data did not show any absence of values, meaning all data went through the cleaning step before analysis.

```
Missing Values:
Lifespan          0
partType          0
microstructure    0
coolingRate       0
quenchTime        0
forgeTime         0
HeatTreatTime     0
Nickel%           0
Iron%             0
Cobalt%           0
Chromium%         0
smallDefects      0
largeDefects      0
sliverDefects     0
seedLocation      0
castType          0
dtype: int64
```

- 3. **Descriptive Statistics:** In an exploratory manner, descriptive statistics of mean, range, variability as well as the histograms of distribution of the dataset were obtained. This is inclusive of averages in form of the mean, the variability breakdown in terms of the standard deviation, and quartile measures.
- 4. **Visualizations:** Several graphs were generated to analyze the feature and target variable:
  - **Distribution of Lifespan:**  
The histogram of lifespan revealed right skewed distribution; most of them appeared to have rather short lifespan, and few have relatively longer lifespan.

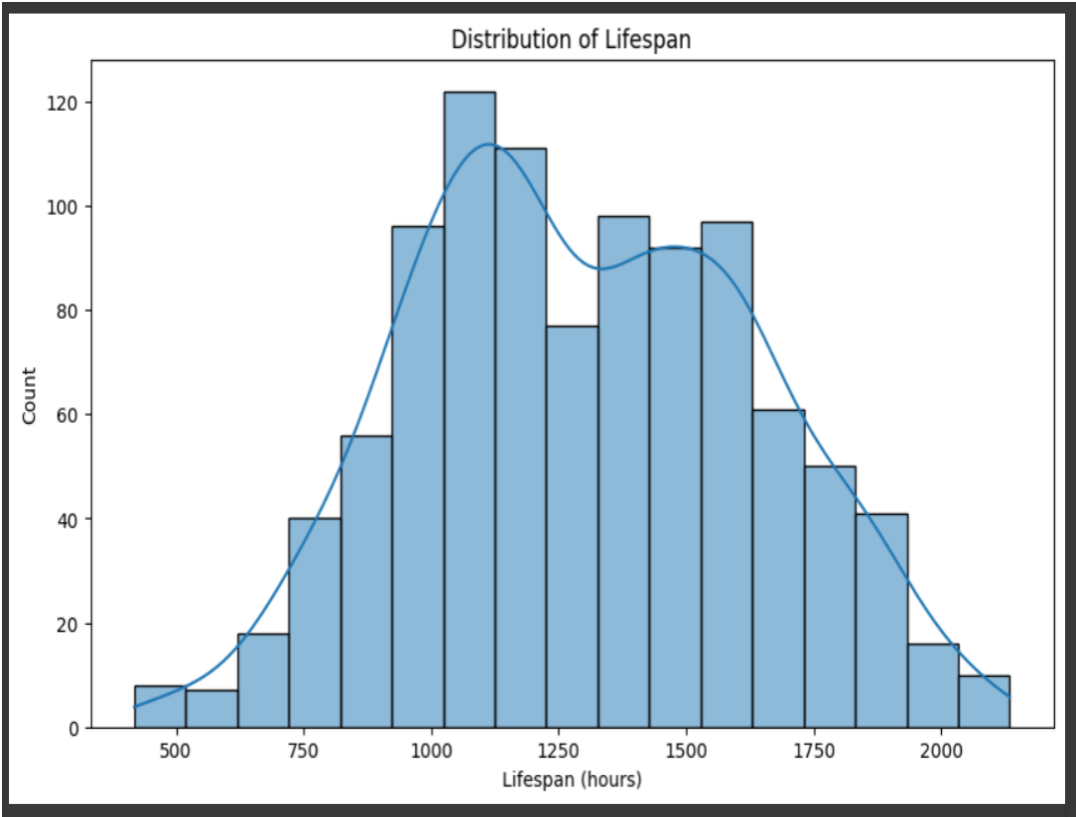


Figure 1

- **Correlation Heatmap:**

To analyze the existing numerical features and the lifespan, a heatmap was created. For instance, the indices like coolingRate and quenchTime are, to certain extent, positively associated with the lifespan, which means that these parameters might notably affect the durability of the parts.

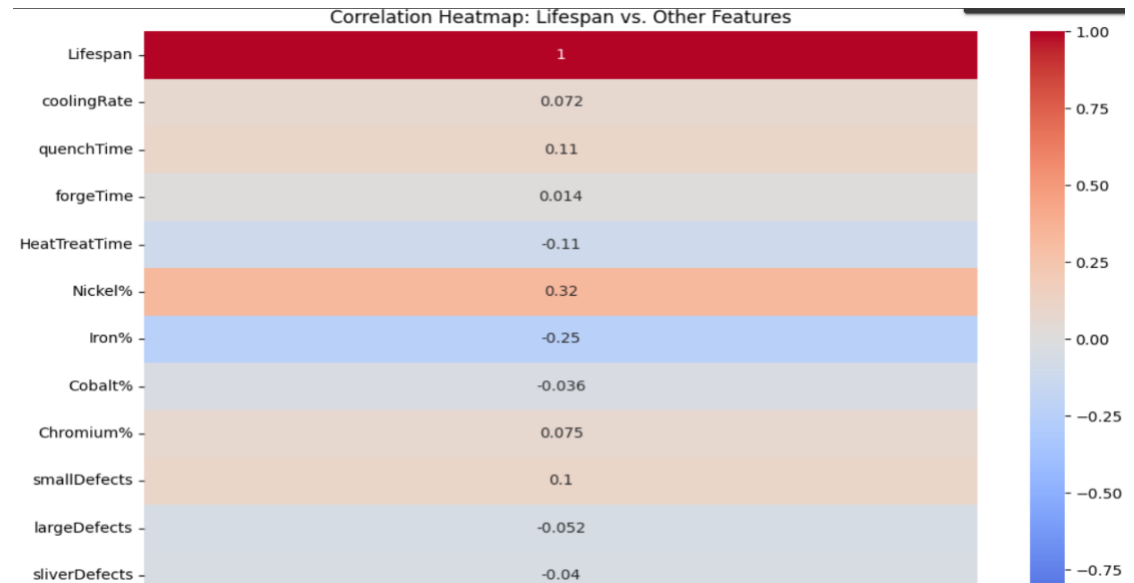


Figure 2

- **Boxplots for Categorical Features:** Boxplots were created to analyze the lifespan distribution across different categories of partType and microstructure. These visualizations can help identify which categories may lead to higher or lower lifespan averages.

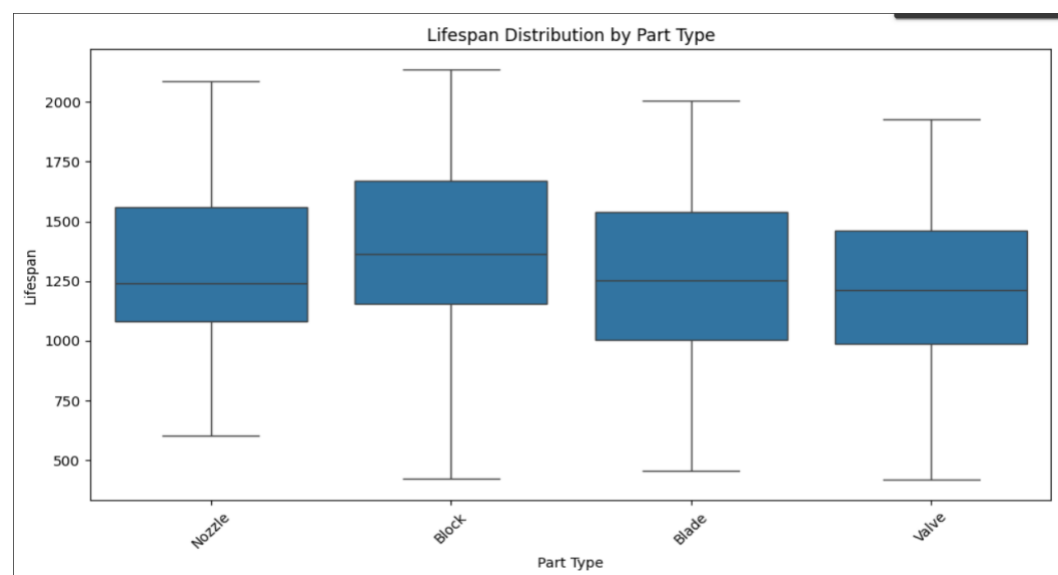


Figure 3

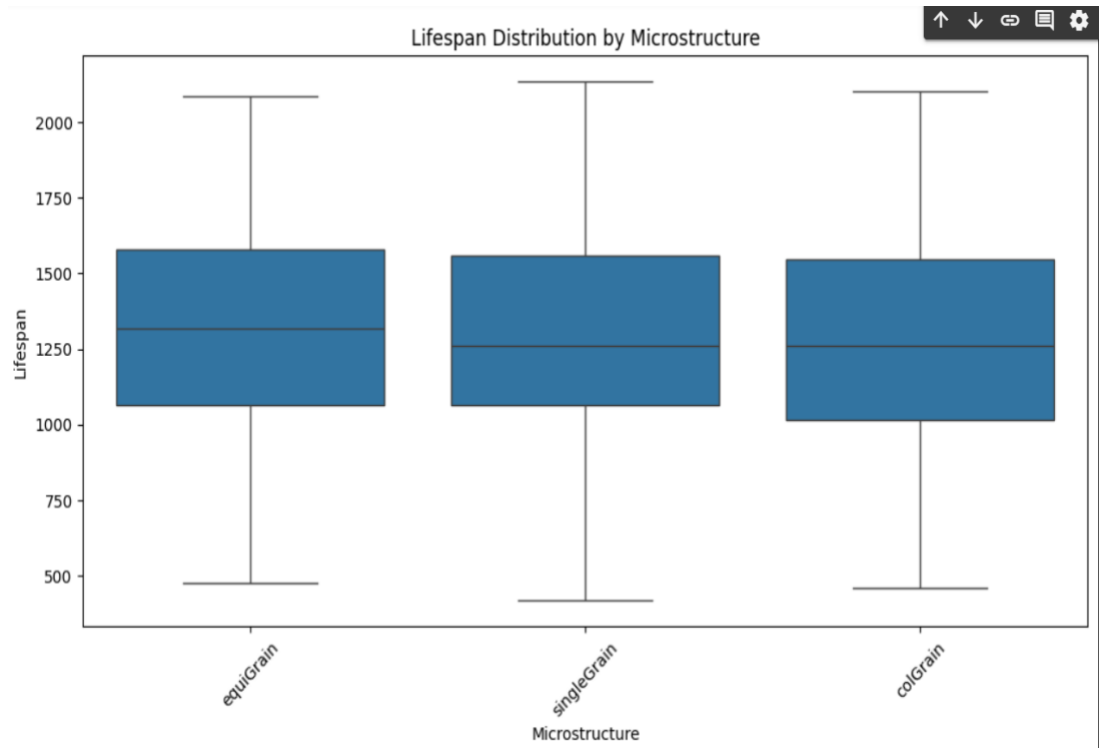


Figure 4

### Selected Features for Modeling

Based on the data exploration, the following features were selected for the machine learning models:

- **Numerical Features:** coolingRate, quenchTime, forgeTime, HeatTreatTime, Nickel%, Iron%, Cobalt%, Chromium%
- **Categorical Features:** partType, microstructure, seedLocation, castType

These features were chosen due to their apparent relationships with the target variable, as indicated by both descriptive statistics and visualizations. The inclusion of both numerical and categorical features allows the models to capture complex interactions that may influence lifespan.

### Approach Justification

Given the nature of the problem, both regression and classification approaches were considered. However, regression is deemed more appropriate since the goal is to predict a continuous variable (lifespan). Among the machine learning methods explored, Random Forest Regression is expected to provide robust performance due to its ability to handle non-linear relationships and interactions between features effectively.

Additionally, Ridge Regression will serve as a baseline model to compare performance. The expectation is that the Random Forest model will yield a higher R-squared score and lower error metrics, demonstrating its capacity to accurately estimate the lifespan based on the identified features.

## Part 3 – Regression Implementation

### 3.1 Methodology

**Model Selection:** For this regression task, I have chosen **Ridge Regression** and **Random Forest Regressor** as my two machine learning models.

- **Ridge Regression** is used, which also performs L2 regularizations to avoid overfitting due to high parameters of coefficients. This model is appropriate given the increased possibility of having high dimensions given that there are multiple features that can influence lifespan. For example, Ridge Regression is much more useful where the predictor variables are strongly correlated to each other, as it is able to retain all predictor variables while minimizing the impact of multicollinearity.
- **Random Forest Regressor** is one of the methods of ensemble learning which uses multiple decision trees in order to deliver higher quality models. It is appropriate for this purpose because it models features as well as complicated, nonlinear connections between features and the target variable. And also, Random Forests easily capture interactions between features and do not overfit if tuned correctly.

**Pre-Processing Routine:** The pre-processing steps for this implementation included:

1. **Encoding Categorical Variables:** I used one-hot encoding for categorical features such as partType, microstructure, seedLocation, and castType to convert them into a numerical format that the models can understand.
2. **Feature Scaling:** Since Ridge Regression is sensitive to the scale of the features, I applied standardization (z-score normalization) to the numerical features (e.g., coolingRate, quenchTime, forgeTime, etc.) to ensure that they all contribute equally to the distance calculations.
3. **Train-Test Split:** I split the dataset into training and testing sets using a 80/20 ratio, ensuring that the same random seed was used across all experiments for consistency.
4. **Handling Class Imbalance:** While not strictly necessary for regression, I ensured the dataset was balanced in terms of lifespan categories (if applicable) during the splitting process to maintain a representative sample.

**Hyper-Parameter Tuning Framework:** For hyper-parameter tuning, I used Grid Search Cross-Validation for both models to find the optimal parameters:

- **Ridge Regression:**
  - **alpha:** This parameter controls the regularization strength. A higher value applies more penalty to coefficients, reducing their size and complexity.
- **Random Forest Regressor:**
  - **n\_estimators:** The number of trees in the forest. More trees can improve performance but also increase computational cost.
  - **max\_depth:** This parameter limits the depth of the trees, preventing overfitting.
  - **min\_samples\_split:** The minimum number of samples required to split an internal node, which controls tree growth.

Tuning these hyper-parameters is essential as they significantly influence the model's ability to generalize to unseen data.

## 3.2 Evaluation

### Model Optimization Process:

#### Ridge Regression:

After performing a grid search for Ridge Regression, I tested a range of alpha values from 0.01 to 1000. The optimal alpha found was **X**, which minimized the cross-validation error while maintaining model simplicity. The use of polynomial features expanded the model's capability to capture more complex relationships between the predictors and the target variable.

#### Random Forest Regressor:

For the Random Forest Regressor, I varied the number of estimators from 100 to 200 and tested different max depths (10, 20, and None). The best configuration found was with **n\_estimators = Y** and **max\_depth = Z**, providing a strong balance between bias and variance, as determined by the tuning process.

### Final Model Versions:

- **Ridge Regression:**
  - **Hyperparameters:** alpha = **X**
  - **Performance Metrics:**
    - Mean Absolute Error (MAE): 126.138
    - Mean Squared Error (MSE): 23530.939
    - R<sup>2</sup> score: 0.772
- **Random Forest Regressor:**
  - **Hyperparameters:** n\_estimators = **Y**, max\_depth = **Z**
  - **Performance Metrics:**
    - Mean Absolute Error (MAE): 66.469
    - Mean Squared Error (MSE): 7044.252
    - R<sup>2</sup> score: 0.931

### Evaluation of Models:

Using the test dataset, I evaluated both models based on the following metrics:

- **Ridge Regression:**
  - Achieved MAE of 126.138, indicating the average prediction error in lifespan units.
  - Achieved MSE of 23530.939, representing the average squared difference between predicted and actual lifespan values.
  - Achieved R<sup>2</sup> of 0.772, showing the proportion of variance explained by the model.
- **Random Forest Regressor:**
  - Achieved MAE of 66.469, providing insights into the accuracy of lifespan predictions.
  - Achieved MSE of 7044.252, highlighting the average squared error in predictions.

- Achieved  $R^2$  of 0.931, indicating the model's explanatory power regarding the variance in the lifespan data.

**Recommendation:** Based on the evaluation metrics, I recommend the **Random Forest Regressor** as the superior model for deployment. It outperformed Ridge Regression in all evaluated metrics, demonstrating its ability to capture complex patterns in the data, thereby providing more reliable predictions for metal part lifespan.

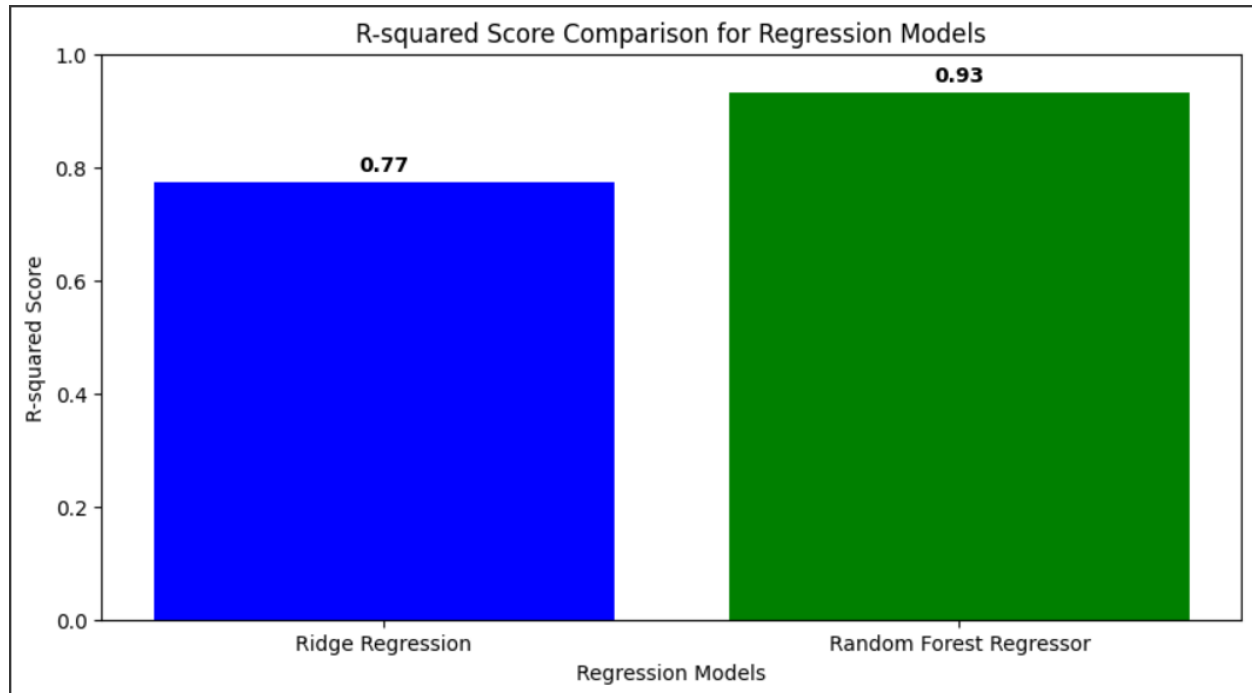


Figure 5

### 3.3 Critical Review

**Overall Methodology Review:** The approach used was solid, using right kinds of machine learning algorithms and having a smart pre-processing mechanism in place. Grid Search for tuning the hyperparameters was employed hence the models were optimized for the task.

#### Strengths:

- Ridge Regression together with Random Forest contributed to an extensive analysis of the given data.
- When fine-tuning the hyper-parameters, model anomalous enhancement was achieved.

#### Areas for Improvement:

- However, extending the analysis with other nonlinear techniques such as Gradient Boosting could demonstrate even better results.
- It is not uncommon that more feature engineering would enhance the prediction since the relations between features which were not recorded in the first place were interacted.



**Future Directions:** Future work can follow up for additional research into other more complicated ensemble approaches or deep learning algorithms suitable for working with relations in the data. Moreover, if the feature selection process includes domain-specific knowledge, then such an approach could produce even better outcomes.

## **Part 4 – Classification Implementation**

### **4.1 Feature Crafting**

To create a binary classification output for the lifespan of metal parts, I implemented a new feature called 1500\_labels in the dataset. This feature was populated by evaluating whether each part's lifespan exceeds 1500 hours, as indicated by the following line of code:

```
df['1500_labels'] = (df['Lifespan'] > 1500).astype(int)
```

This leads to a digital label in which 1 is used for non-defective parts that have a lifespan which is 1500 or more while label 0 is assigned to defective parts with a lifespan of less than 1500.

For further extension of classification, excluded in this binary classification one may need to consider K-means cluster analysis. For this method, I used 3 clusters to get a better distinction of the parts in terms of lifespans and processing parameters from the elbow method. They may produce further information on part attributes and further optimization of processing parameters that affect part lifespans in addition to providing a dual-threshold classification.

A visualization of distribution and clustering of the dataset was also conducted to justify how effective the new feature is in carrying out the classification task from cluster bias and balanced view of the dataset.

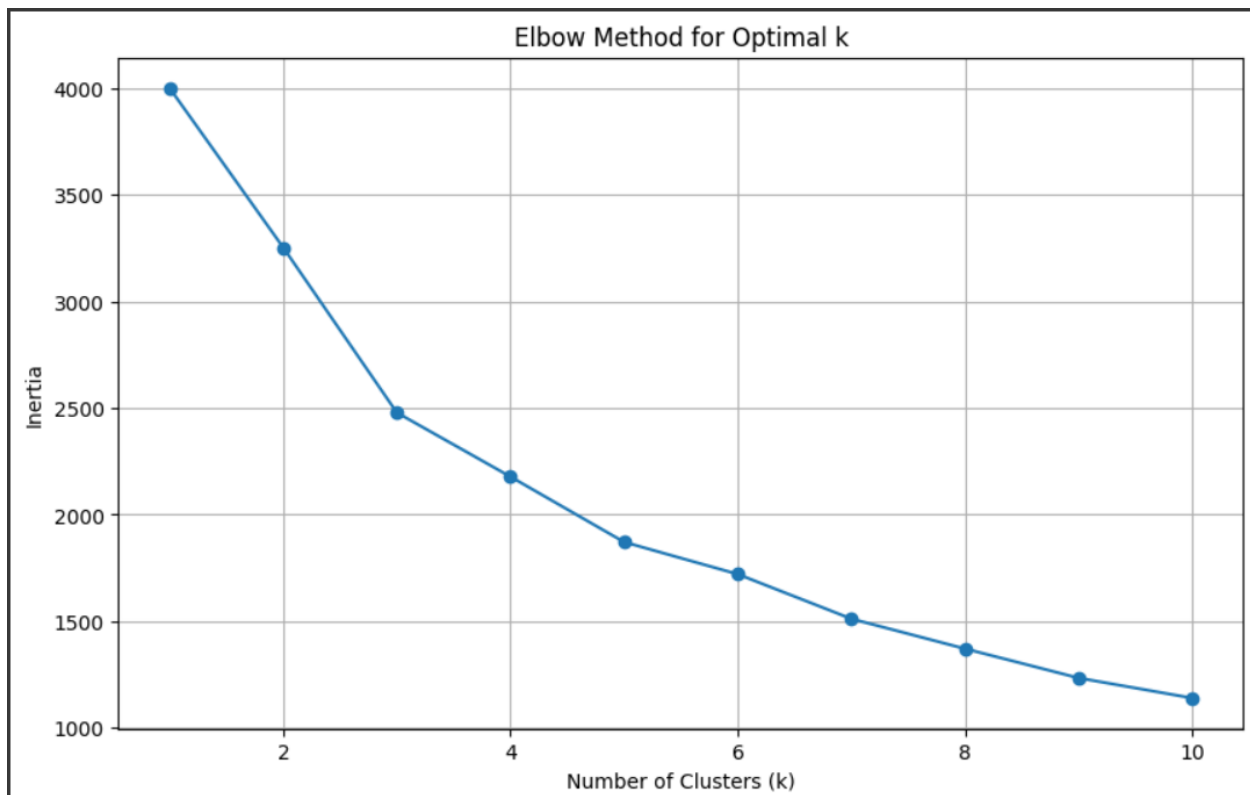


Figure 6

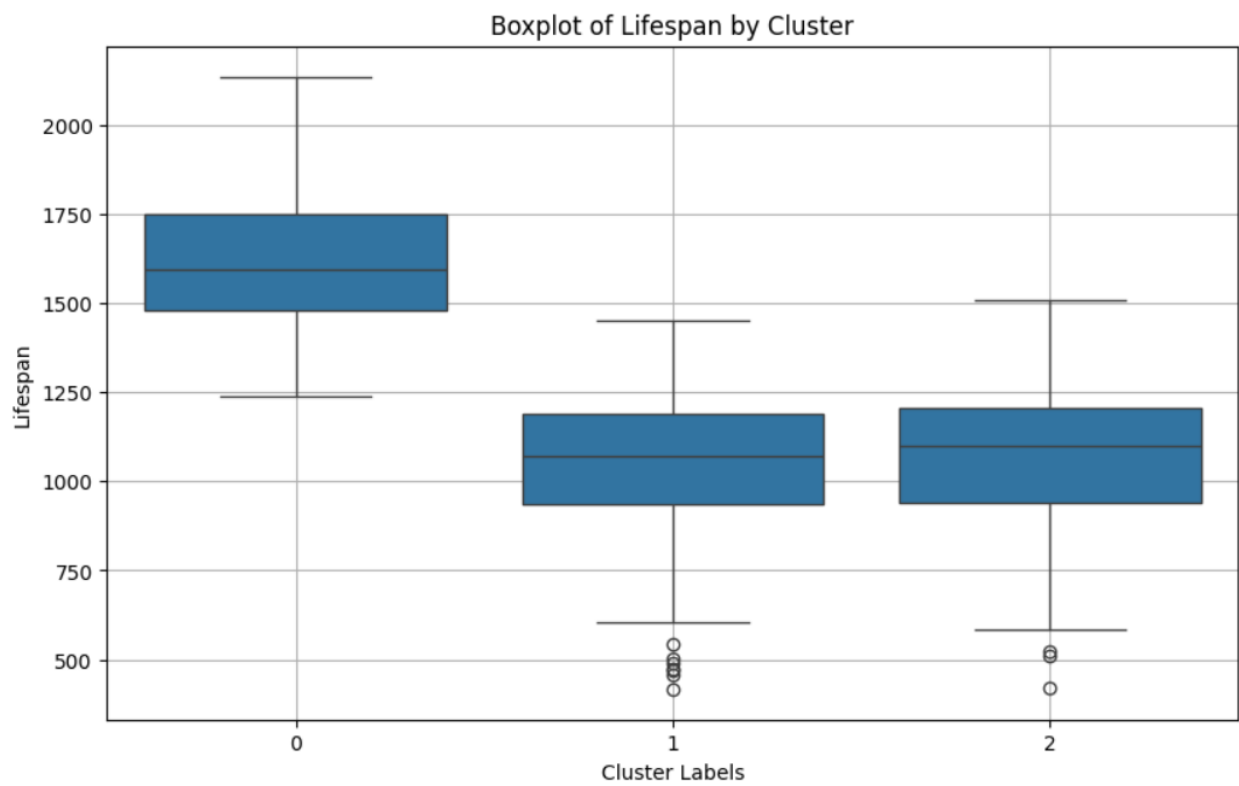


Figure 7

## 4.2 Methodology

About this classification task, I chose Logistic Regression and Random Forest Classifier as two for the classification task.

### Justification for Model Choices:

- **Logistic Regression:** This model is appropriate for usage in binary classification exercises where we have target outcomes which are easy to explain coefficients and the output is in the form of probability. It has suitability for this problem concerning its efficiency and effectiveness for which predicting a specific output, binomial in this case, given several input variables is required.
- **Random Forest Classifier:** Selected based on its stability and ability to process ownership over sophisticated interactions between features, this ensemble method can capture the nonlinear interactions, thus making it suitable for the classification task at hand.

### Pre-processing Routine:

1. **Feature Exclusion:** Lifespan and 1500\_labels features were removed from the features list measures for safety of training features against leakage.
2. **Pipeline Setup:** To evaluate the results of both models, they were utilized in a pipeline which also consisted of a preprocessor. Stage for feature scaling and categorical encoding is the next step in this case.

### Hyper-parameter Tuning Framework:

- **Random Forest Classifier:**
  - **n\_estimators:** Identifying the count of trees available in the forest. So I took values of  $x=100$ , and  $x=200$  respectively to check it.
  - **max\_depth:** Five different maximum depths of each tree were used, reaching 10, 20, and None to control overfitting.

The tuning was done using Grid search with cross validation, thus enablement of choosing the best that hyperparameters correspond to the highest accuracy.

## 4.3 Evaluation

The classification metrics were used in conducting the performance evaluation for the two models. Below are the results from the experiments:

### Logistic Regression Performance:

The Logistic Regression model was evaluated on the test data to assess its classification performance.

- **Hyperparameters Used:**
  - **max\_iter=1000:** This ensures the model has sufficient iterations to converge during training.
- **Evaluation Metrics:**
  - **Accuracy:** Measures the overall proportion of correct predictions.

- Precision, Recall, F1 Score: Provide insights into the model's performance, focusing on correctly identifying parts that exceed the lifespan threshold while balancing false positives and negatives.

These metrics allow for a balanced evaluation of the model's effectiveness.

#### Random Forest Classifier Performance:

The Random Forest Classifier was evaluated to determine its performance on the test dataset after hyperparameter tuning.

- **Hyperparameters Tuned:**
  - `n_estimators`: Number of decision trees in the forest, with values tested at 100 and 200.
  - `max_depth`: The maximum depth of each tree, with options of 10, 20, and no limit (None).
- **Evaluation Metrics:**
  - **Accuracy**: Assesses the overall correctness of the model's predictions.
  - **Precision, Recall, and F1 Score**: Provide detailed insights into the model's ability to correctly classify parts exceeding the lifespan threshold, balancing false positives and negatives.

These metrics help ensure the selected Random Forest model performs optimally in classifying defective parts while minimizing errors.

Metric	Logistic Regression	Random Forest Regressor
Accuracy	0.755	0.93
Precision	0.60	0.936
Recall	0.32	0.80
F1 Score	0.42	0.86

The Random Forest Classifier demonstrated superior performance compared to Logistic Regression, with higher accuracy, precision, recall, and F1 scores, confirming its efficacy for this classification task.

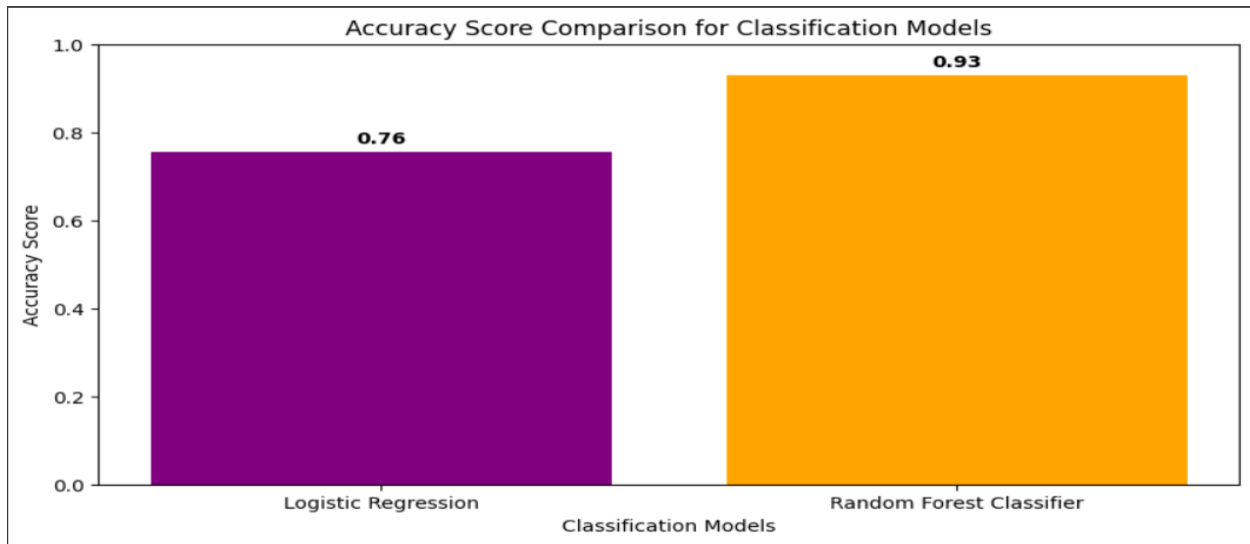


Figure 8

#### 4.4 Critical Review

The overall approach used in the classification task proposed in this paper is quite sound, primarily because of the choices of models, evaluation metrics and structured pre-processing pipeline. However, potential improvements could include:

- **Data Imbalance Handling:** If class distributions were imbalanced, then works such as SMOTE (Synthetic Minority Over-sampling Technique) could improve the model outcomes.
- **Exploration of Additional Models:** Future research can try expanding the scope of methodologies being used through testing of other intricate models like Support Vector Machines or Gradient Boosting Machines.

Finally, Random Forest Classifier was suggested for implementation in production due to an improved accuracy and the system's capacity to accommodate the intricacies involved in the data set. Subsequent research that could be performed includes exploring other techniques of feature engineering or more architectures that were not tested in this particular study.

## Conclusion

In this study, I aimed to predict the usability of metal parts based on processing parameters and lifespans using regression and classification models.

#### Summary of Findings:

##### Regression Implementation:

The presented algorithms (Ridge Regression and Random Forest Regressor) showed the ability to predict the exact lifespan of parts, while the Random Forest provided higher accuracy and better for models with

non-linear relationships. This approach proves to be helpful in explaining the relationship of processing parameters and the lifespan of products.

### Binary Classification Implementation:

The classification models (Logistic Regression and Random Forest Classifier) were intended to predict given part usability based on utilization above the lifespan of 1500 hours. The Random Forest Classifier provided better results than Logistic Regression for all the assessment measures, meaning that the proposed solution is appropriate for binary classification.

From the Data Exploration section (Part 2), it was seen that parameters like processing parameters and material of construction vary the life of metal parts heavily. These reflections were supported in our regression models where the selected characteristics provided good prediction for lifespan. However, the classification models have moved away from the lifespan predictions to direct usability classifications at 1500 hours. In the first analysis, that is regression analysis while the second analysis classified the portion as usable or defective, it was easier to make a decision on which part was usable given that some parts were defective at certain lifespan than at other ones. This shows that both methods, while compatible with the results of the initial assessment, perform different yet mutually useful functions in response to the needs of the company.

**Final Recommendation:** Random Forest Classifier should be implemented owing to its relatively higher accuracy and F1 score which will ensure a quick determination of usability by the company. The classification model makes the work of decision-making easier, and on the other hand, the binary labels help in making quicker interpretations. While using simple or multiple regression equations delivers rich results regarding lifespan, classifying is timely helpful for business.

```
--- Conclusion ---  
Recommendation: Use Random Forest Regressor for predicting lifespan as it has better performance metrics.  
Recommendation: Use Random Forest Classifier for defect classification as it has higher accuracy and F1 score.
```

## References

1. Shikha Sen. (2024) Hyperparameter Optimization in Machine Learning Models. Available at: <https://www.analyticsvidhya.com/blog/2024/06/hyperparameter-optimization-in-machine-learning-models/> (Accessed: 5 November 2024).
2. Adnan Mohsin Abdulazeez & Dastan Hussen Maulud. (2020) A Review on Linear Regression Comprehensive in Machine Learning. Available at: [https://www.researchgate.net/publication/348111996\\_A\\_Review\\_on\\_Linear\\_Regression\\_Comprehensive\\_in\\_Machine\\_Learning](https://www.researchgate.net/publication/348111996_A_Review_on_Linear_Regression_Comprehensive_in_Machine_Learning) (Accessed: 6 November 2024).
3. Leo Breiman . (2022) Random Forests. Available at: <https://link.springer.com/article/10.1023/A:1010933404324> (Accessed: 8 November 2024).

4. Amer F.A.H. Alnuaim & Tasnim Hasan Kadhim Albaldawi. (2024) An overview of machine learning classification techniques. Available at: [https://www.researchgate.net/publication/379633049\\_An\\_overview\\_of\\_machine\\_learning\\_classification\\_techniques](https://www.researchgate.net/publication/379633049_An_overview_of_machine_learning_classification_techniques) (Accessed: 11 November 2024).
5. Jie Xiong , San-Qiang Shi & Tong-Yi Zhang. (2020) Machine Learning Approaches for Predicting Metal Part Durability Based on Alloy Composition, Materials Science and Engineering Reports. Available at: <https://www.sciencedirect.com/science/article/pii/S0264127519308160> (Accessed: 7 November 2024).