COMP1801 - Machine Learning Coursework Report

Name: MIZANUR RAHMAN

Student ID: 001359488

Word Count: 2963

Contents

Part1 - Executive Summary

This report examines the prediction of metal part lifespan using manufacturing and material features. Two regression models were evaluated: Polynomial Ridge Regression and Random Forest Regression. Polynomial Ridge Regression, with a degree of 2 and alpha of 1e-10, performed better than Random Forest, achieving a Mean Squared Error (MSE) is 24,066.54 and an R2 score is 0.81, while Random Forest produced an MSE is 40,010.09 and an R2 is 0.69. For classification, K-Means clustering was employed to segment lifespan into six categories, with Logistic Regression and Artificial Neural Networks (ANN) used for modeling. The ANN, which incorporated dropout regularization, achieved 56% accuracy, substantially outperforming Logistic Regression, which only reached 24%. The results indicate that Polynomial Ridge Regression is the most effective model for lifespan prediction, as it captures non-linear relationships well, while ANN shows promise for classification tasks. Based on these findings, Polynomial Ridge Regression is recommended for deployment.

Part2 - Data Exploration

Data Loading and Overview

The given dataset ("COMP1801_Coursework_Dataset") is a CSV file. Python library pandas is used to read this CSV file and create a DataFrame. The first few rows of the DataFrame have been given below:

	Lifespan	partType	microstructure	coolingRate	quenchTime	forgeTime	HeatTreatTime	Nickel%	Iron%	Cobalt%	Chromium%	smallDefects	largeDefects	sliverDefects	seedLocation	castType
0	1469.17	Nozzle	equiGrain	13	3.84	6.47	46.87	65.73	16.52	16.82	0.93	10	0	0	Bottom	Die
1	1793.64	Block	singleGrain	19	2.62	3.48	44.70	54.22	35.38	6.14	4.26	19	0	0	Bottom	Investment
2	700.60	Blade	equiGrain	28	0.76	1.34	9.54	51.83	35.95	8.81	3.41	35	3	0	Bottom	Investment
3	1082.10	Nozzle	colGrain	9	2.01	2.19	20.29	57.03	23.33	16.86	2.78	0	1	0	Top	Continuous
4	1838.83	Blade	colGrain	16	4.13	3.87	16.13	59.62	27.37	11.45	1.56	10	0	0	Top	Die

Fig-1: DataFrame

The DataFrame contains 1000 rows, each representing a metal part with various manufacturing and 16 features/columns, with "Lifespan" as the target feature. Columns are a mix of numerical and categorical features.

	Lifespan	coolingRate	quenchTime	forgeTime	HeatTreatTime	Nickel%	Iron%	Cobalt%	Chromium%	smallDefects	largeDefects	sliverDefects
count	1000.000000	1000.000000	1000.000000	1000.000000	1000.000000	1000.000000	1000.000000	1000.000000	1000.000000	1000.000000	1000.000000	1000.000000
mean	1298.556320	17.639000	2.764230	5.464600	30.194510	60.243080	24.553580	12.434690	2.768650	17.311000	0.550000	0.292000
std	340.071434	7.491783	1.316979	2.604513	16.889415	5.790475	7.371737	4.333197	1.326496	12.268365	1.163982	1.199239
min	417.990000	5.000000	0.500000	1.030000	1.030000	50.020000	6.660000	5.020000	0.510000	0.000000	0.000000	0.000000
25%	1047.257500	11.000000	1.640000	3.170000	16.185000	55.287500	19.387500	8.597500	1.590000	7.000000	0.000000	0.000000
50%	1266.040000	18.000000	2.755000	5.475000	29.365000	60.615000	24.690000	12.585000	2.865000	18.000000	0.000000	0.000000
75%	1563.050000	24.000000	3.970000	7.740000	44.955000	65.220000	29.882500	16.080000	3.922500	26.000000	0.000000	0.000000
max	2134.530000	30.000000	4.990000	10.000000	59.910000	69.950000	43.650000	19.990000	4.990000	61.000000	4.000000	8.000000

Fig-2: Summary Statistics

The target feature 'Lifespans' varies significantly from 418 to 2135, with an average of about 1300. Manufacturing processes, such as 'Heat Treat Time' and alloy components (Nickel% and Iron%), show considerable variability, highlighting the diversity in part production and material composition.

Visual Exploration

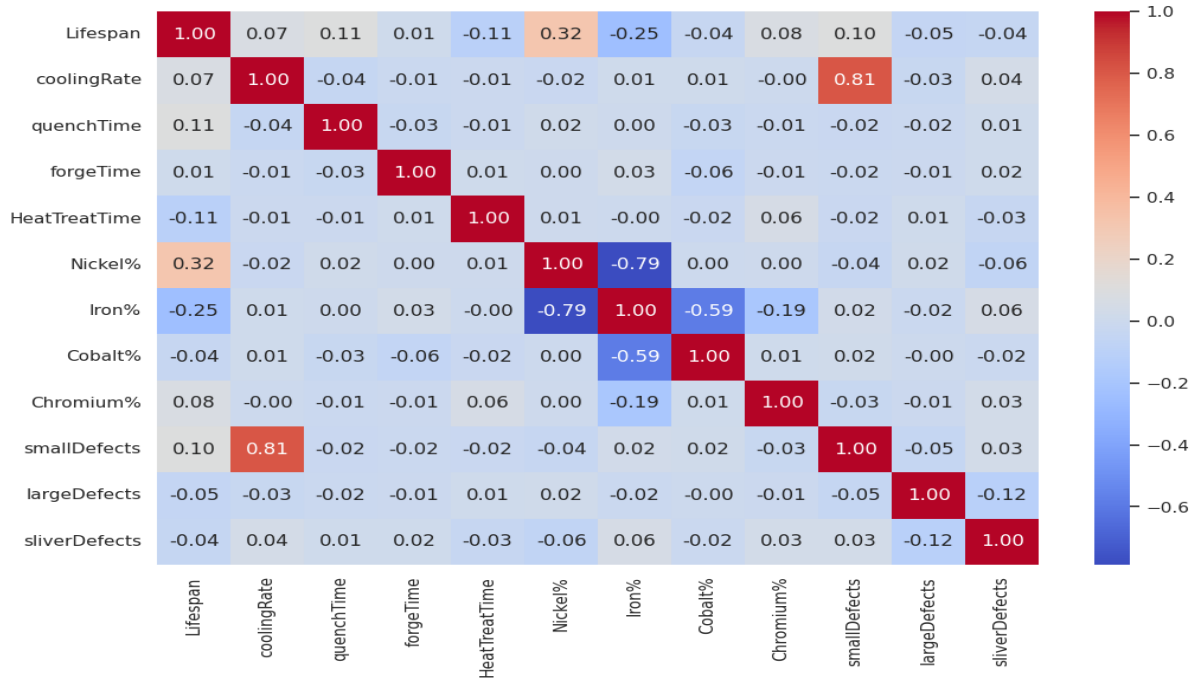

Fig-3: Heatmap

A correlation heatmap is used to identify linear relationships among numerical features. The heatmap reveals that the cooling rate and the presence of small defects are positively correlated. Additionally, the nickel percentage and iron percentage are negatively correlated. However, there is no strong linear correlation between lifespan and any other features.

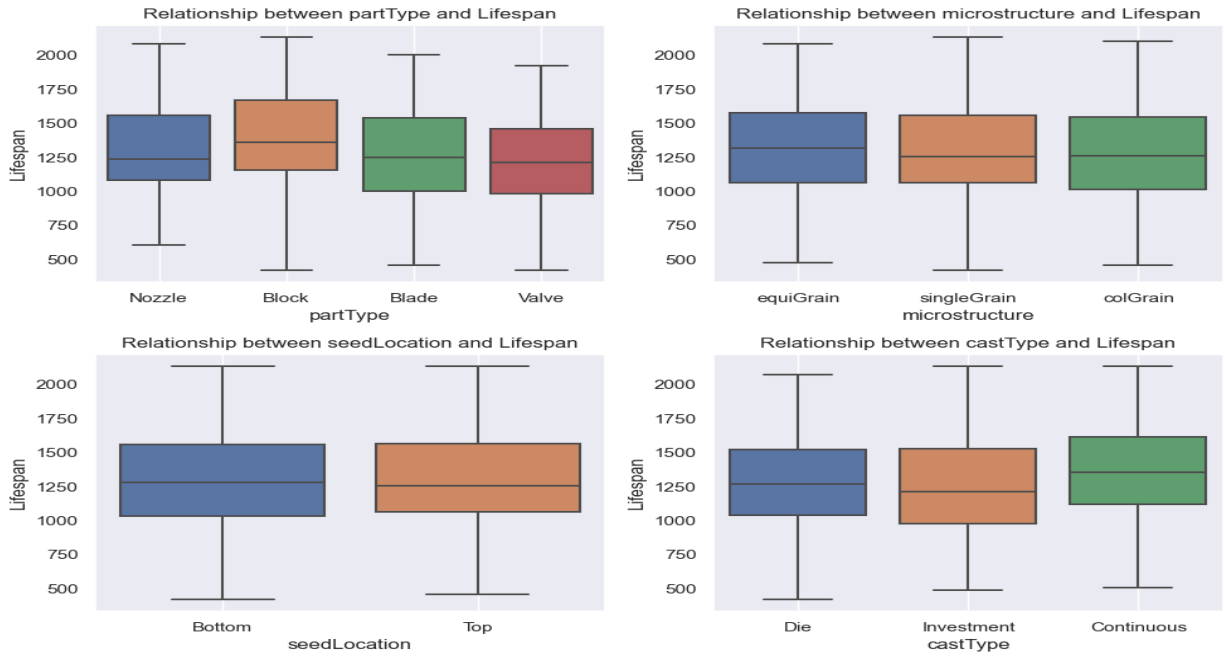

Fig-4: Box plots of the Lifespan feature displayed alongside all categorical features.

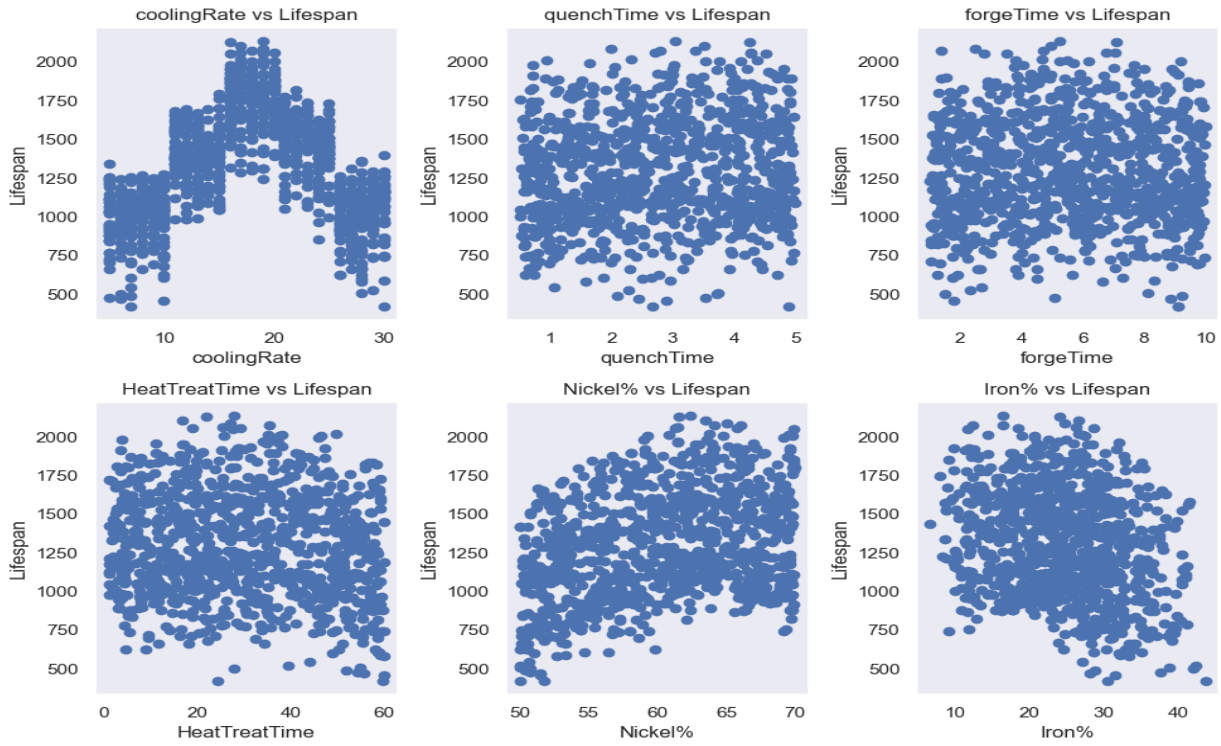

Fig-5.1: Scatter plots of the Lifespan feature displayed alongside all numerical features.

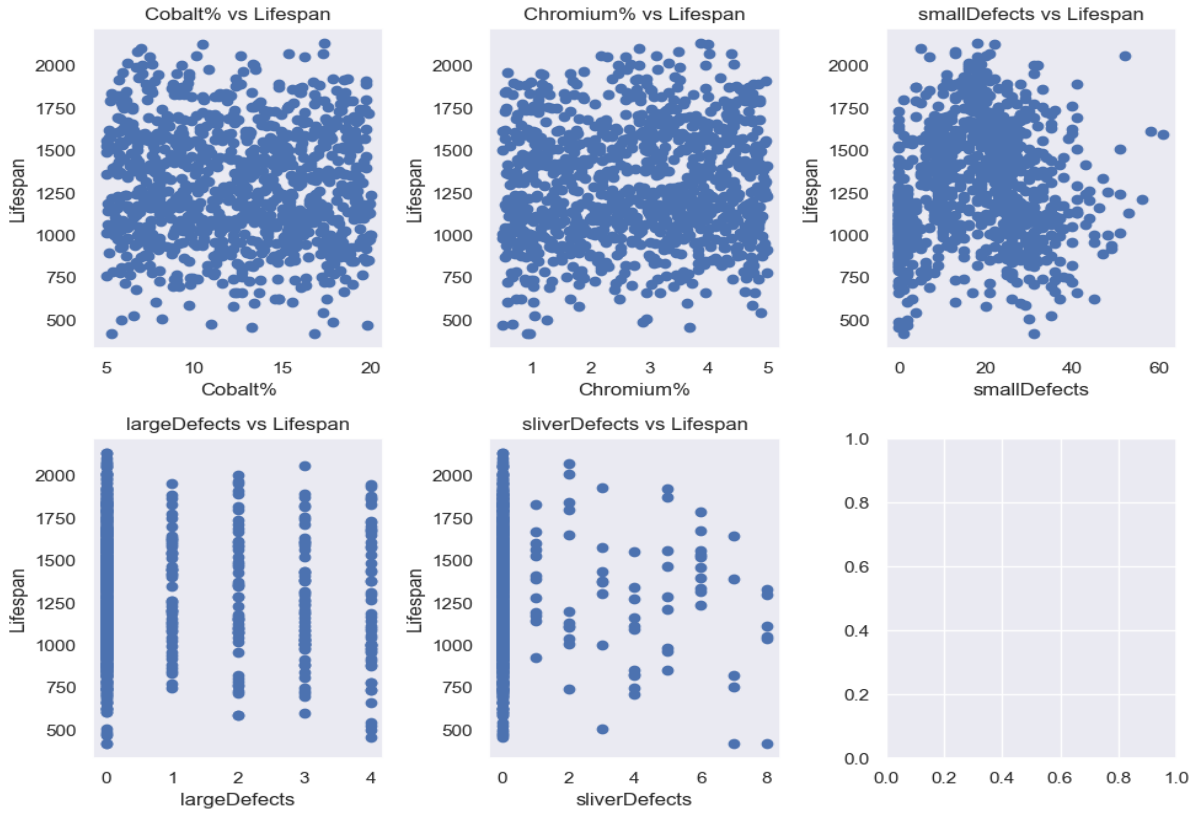

Fig-5.2: Scatter plots of the Lifespan feature displayed alongside all numerical features.

Variability in lifespan across categorical features like ‘partType’ and ‘castType’ was observed in the box plot. For example, casting methods categorized as Continuous seemed to be associated with longer lifespans, indicating potential differences in casting quality. Additionally, the scatter plots depicting 'coolingRate' and alloy components (Nickel% and Iron%) reveal complex and possibly non-linear relationships with lifespan.

Feature Selection for Model

The visual exploration indicates that it is challenging to identify specific numerical features, as there is no linear relationship between lifespan and other numerical attributes. Therefore, all numerical features will be included in the machine learning model. Additionally, categorical features such as ‘partType’ and ‘castType’ seem to influence lifespan differently, likely due to structural or compositional differences. Consequently, both the ‘partType’ and ‘castType’ features will be utilized in the model.

Model Approach and Expectations

Since the target variable, 'Lifespan,' is continuous, a regression-based approach is the best practice for predicting the longevity of metal parts. Polynomial Regression stands out as an effective option for model because there are few linear relationships between the features and Lifespan. This polynomial approach enables us to capture non-linear interactions and complex patterns that linear regression might overlook.

However, there is a risk of overfitting, particularly because of the many features that exhibit diverse relationships with Lifespan. To address this complexity and mitigate overfitting, Ridge and Lasso Regression are recommended to achieve accurate results.

In summary, combining Polynomial Regression with Ridge or Lasso regularization is expected to offer an effective balance between capturing complex relationships and preventing overfitting, making it a robust approach for accurately predicting Lifespan.

Part3 - Regression Implementation

3.1 Methodology

Based on the characteristics of the dataset and the nature of the problem, two models have been chosen: Polynomial Ridge Regression and Random Forest Regression.

Polynomial Ridge Regression: Polynomial Ridge Regression enhances linear regression by introducing polynomial terms to account for non-linear relationships between the

features and the target variable. By incorporating polynomial terms, this model can more effectively represent these non-linear interactions (Ostertagová, 2012). Additionally, the ridge regularization technique helps reduce the risk of overfitting, which can occur when polynomial features are included. This approach is particularly beneficial for datasets where it is crucial to consider feature interactions while also managing the potential for overfitting (Anthony Miller, 2022).

Random Forest Regression: Random Forest is an ensemble method that uses multiple decision trees during training and generates predictions by averaging the outputs of these trees. This technique is particularly effective for regression tasks involving non-linear and complex relationships, as it does not depend on linear associations between features and the target variable. This model is especially useful when there is limited prior knowledge about feature interactions, as it can uncover complex, non-linear relationships without the necessity for extensive model tuning (Yile Ao, 2019).

Data Preprocessing

Feature Scaling: Polynomial Ridge Regression requires scaled data to ensure that the regularization terms function effectively, while Random Forest models can perform slightly better when the features are scaled, especially in cases with high-dimensional data. Therefore, standard scaling is applied, which normalizes the features to have a mean of zero and a standard deviation of one.

Train-Valid-Test Split: A train-valid-test split is used, where 80% of the data is allocated for training, 10% for validation, and the remaining 10% for testing. All the data are allocated randomly. The validation set is used to optimize hyperparameters, while the test set remains separate until the final evaluation stage.

Feature Encoding: Since two categorical features of the dataset will be used in the model, they need to be transformed into a numerical format to be compatible with machine learning algorithms. One-Hot-Encoding is utilized for this purpose; it is a popular technique that converts categorical variables into binary vectors. This method is suitable for both Polynomial Ridge Regression and Random Forest Regression.

Hyperparameter Tuning Framework

Hyperparameter tuning is an important part of finding more accurate results from model. Therefore, essential hyperparameter tuning will be applied to both models.

Hyperparameter Tuning of Polynomial Ridge Regression

Degree of Polynomial: This parameter controls the complexity of the model by incorporating polynomial terms up to a specified degree. While higher degrees can

improve the model's fit, they also increase the risk of overfitting. On the other hand, lower degrees may result in underfitting.

Regularization Strength (alpha): The regularization term helps prevent overfitting by penalizing large coefficients. Adjusting the regularization strength, alpha, is crucial for achieving a balance between bias and variance in the model.

Hyperparameter Tuning of Random Forest Regression

Number of Trees (n_estimators): This parameter influences both the accuracy of the model and the duration of training. Increasing the number of trees improves stability but also increases computational time.

Maximum Depth of Trees (max_depth): This setting controls the growth of the trees and helps prevent overfitting. Shallower trees may underfit the data, while deeper trees are at risk of overfitting.

Minimum Samples Split: This parameter specifies the minimum number of samples required to split a node, impacting both the model's performance and its ability to generalize.

3.2 Evaluation

Polynomial Ridge Regression

A grid search was conducted on polynomial degrees and regularization strengths, with the best model found to have a polynomial degree of **2** and an alpha of **1e-10**. The following table summarizes the tuning results:

Polynomial Degree	MSE (Train)	MSE(Valid)	R2 Score (Train)	R2 Score (Valid)
1	95229.07	81280.32	0.19	0.04
2	18565.29	22533.37	0.84	0.73
3	18062.49	83763.35	0.84	0.009
4	845.77	1.640	0.99	-1939860320586

Table 1: MSE and R2 score of Polynomial Model (different degree)

Alpha Value	MSE(Valid)
1e-10	22532.58
0.0001	22532.64
1.0	23284.13
10000.0	79367.54

Table 2: MSE according to Alpha values

Random Forest Regression

Using randomized search, the optimal parameters were determined to be **200 trees (n_estimators)**, a **maximum depth of 20**, and a **minimum samples split of None**. Selected results on the validation set are shown below:

n estimators	Max depth	Max features	MSE(Train)	R2 (Train)	MSE (Valid)	R2 (Valid)
50	None	None	988	0.9916	12331	0.8542
50	None	Sqrt	3170	0.9731	21670	0.7438
100	None	None	936	0.9921	12090	0.8571
100	20	None	1264	0.9893	12306	0.8545
200	None	None	917	0.9922	11954	0.8587
200	20	None	917	0.9922	11948	0.8588
300	None	Sqrt	2584	0.9781	19814	0.7658

Table 3: Hyperparameter of Random Forest Regression Model

Model Performance and Comparison

The final model versions for each regression model are evaluated on a test set using Mean Squared Error (MSE) and R2 score:

Model	MSE	R2 Score
Polynomial Ridge Regression	24066.54	0.81
Random Forest Regression	40010.09	0.69

Table 4: Model performance

Interpretation: Polynomial Ridge Regression shows superior performance on both MSE and R2 Score, suggesting it captures the complexities of the dataset better than Random Forest Regression.

3.3 Critical Review

Polynomial Ridge Regression was a good choice because it captures non-linear patterns when linear relationships are weak. Ridge regularization helps manage complexity from polynomial terms and reduce overfitting. Additionally, Random Forest is a strong complementary model due to its robustness with non-linear relationships and automatic handling of feature importance. Moreover, regularization through Ridge Regression and careful hyperparameter tuning improved model performance and reduced overfitting. A grid search for Polynomial Ridge and a randomized search for Random Forest efficiently identified optimal settings. The data was split into training, validation, and test sets to ensure unbiased performance assessment. Feature scaling also enhanced the effectiveness of Polynomial Ridge Regression in capturing relationships among features.

Although Polynomial Ridge Regression outperformed Random Forest in MSE and R2 score, there are signs of potential overfitting, as indicated by the sharp performance drop at higher polynomial degrees (especially degrees 3 and 4). This suggests exploring Lasso Regression for regularization and feature selection could stabilize the model while reducing complexity. While the hyperparameter tuning strategy was effective, expanding

it to include Bayesian optimization or cross-validation-based grid search could lead to better selections, especially for the computationally intensive Random Forest model.

However, Neural Network model could serve as a valuable alternative, capturing complex patterns without the need for manual polynomial specifications. Combining neural networks with regularization techniques like dropout can further reduce overfitting risks. Moreover, future work could implement k-fold cross-validation during hyperparameter tuning to ensure robust generalization and minimize performance biases from the train-validation split.

Part4 - Classification Implementation

4.1 Feature Crafting

An unsupervised clustering technique, K-Means, was used to categorize lifespan hours into multiple classes based on their lifespan and related features. The Elbow Method was applied to analyze the sum of squared distances (inertia) for k values ranging from 1 to 10. This analysis revealed an optimal k value of 6 clusters, as indicated by the elbow plot, which showed a significant reduction in inertia up to this point. This approach resulted in six distinct clusters based on lifespan, reflecting subtle variations in manufacturing quality and process parameters.

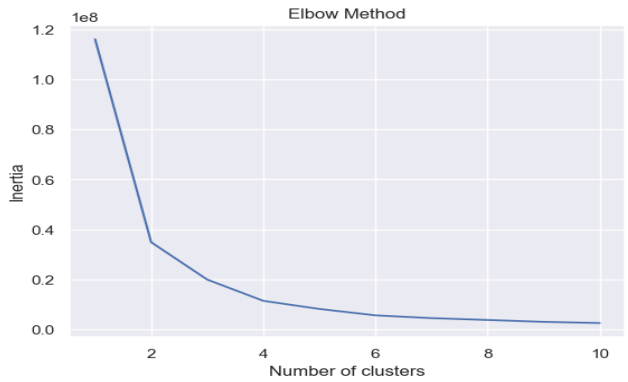

Fig-6: Number of clusters vs Inertia

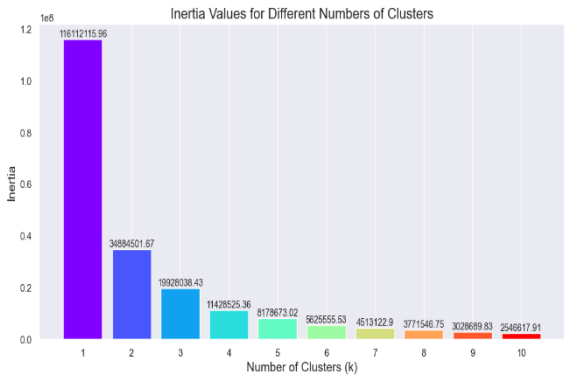

Fig-7: Inertia Values for Different Clusters

According to these results, the values of the Lifespan feature are distributed into six categories under a new feature named ‘Target Hour’. Furthermore, only six clusters maintain the threshold of 1500 hours. In the case of fewer than six clusters, one category has a range of 1400 to 1600 hours, which violates the threshold rule. Here each category containing a range of hours as shown below:

Cluster	Lifespan Range (Hours)
0	1300.66 – 1527.35
1	850.00 – 1078.50
2	1082.10 – 1299.90
3	1774.38 – 2134.53
4	417.99 – 845.40
5	1529.60 – 1761.96

Table 5: Cluster range

4.2 Methodology

There are many models available for classification. For this task, I will use Logistic Regression and an Artificial Neural Network (ANN).

Logistic Regression: This model is chosen for its simplicity and interpretability, making it an effective baseline for classification tasks. Logistic regression is suitable for both binary and multi-class problems where the classes can be somewhat linearly separated. It generates probability-based outputs, which are useful for adjusting decision thresholds (Sperandei, 2014).

Artificial Neural Network (ANN): An ANN is selected as a second model because of its ability to capture complex, non-linear relationships among manufacturing parameters. ANNs can be adapted for both binary and multi-class classification and are particularly powerful when working with high-dimensional or intricate datasets (Tu, 1996).

Data Preprocessing

The 'Lifespan' feature has been removed and replaced with the 'Target_hour' feature, which contains categorical data. Additionally, all the categories in the target feature are numerical, ranging from 0 to 5. Therefore, there was no need to apply categorical encoding to this column.

For data splitting, feature scaling, and categorical encoding, the methods used here are the same as those described in section 3.2 for linear regression. First, randomly split the data into training (80%), validation (10%), and test (10%) sets. Next, apply standard scaling to all numerical data and use one-hot encoding for categorical data.

Hyperparameter Tuning

Logistic Regression Tuning: The regularization will be used in logistic regression. The regularization parameter C governs the inverse strength of regularization in the logistic regression model. Regularization is important for preventing overfitting by penalizing large coefficients that could lead the model to capture noise instead of true patterns in

the data. Specifically, lower values of C result in stronger regularization, while higher values of C decrease the strength of regularization.

Artificial Neural Network: In this model, the learning rate has a major effect in the case of accuracy. The learning rate controls the step size with which the optimizer updates the model's weights during training. Moreover, to get a better result dropout regularization method will be applied as this method randomly disables a portion of neurons during training to prevent them from overly depending on each other. The dropout rate specifies the percentage of neurons that are dropped out.

4.3 Evaluation

After data processing, two different models (Logistic Regression and ANN) were applied. To achieve better results and minimize errors, appropriate hyperparameter tuning was also performed.

Logistic Regression Result

Before Tuning: The initial logistic regression model performed poorly, with a validation accuracy of only 18.00%. The confusion matrix showed that the model struggled with all class predictions, especially the minority classes.

After Tuning: After adjustments, the logistic regression model's accuracy increased to 24.00% with the best regularization parameter (C=0.01). Although the accuracy improved, the model still made many mistakes, particularly with the minority classes.

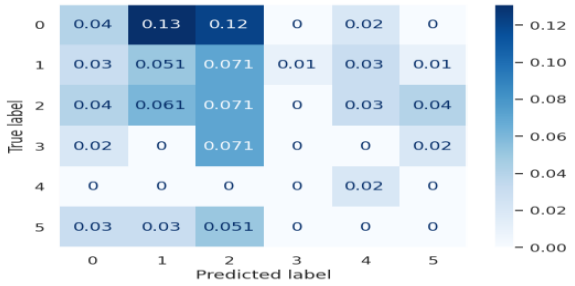

Fig-8: Confusion Matrix (Before Tuning)

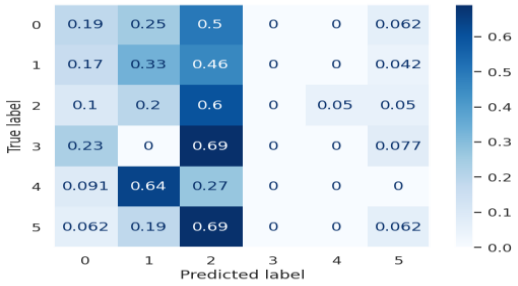

Fig-9: Confusion Matrix (After Tuning)

Metric	Before Tuning	After Tuning
Accuracy	0.18	0.24
Precision (Macro)	0.14	0.15
Recall (Macro)	0.28	0.19
F1 Score (Macro)	0.16	0.16

Table 6: Matrices values before and after tuning

Artificial Neural Network Model Performance

Initially, the artificial neural network (ANN) was tested without dropout regularization, achieving an accuracy of 32% and an F1 score of 0.28. While this performance was slightly better than that of the logistic regression model, it exhibited signs of overfitting, primarily due to the model's high capacity relative to the size of the dataset. After implementing dropout, the model's generalization improved significantly, resulting in an accuracy of 56% and an F1 score of 0.56. Both precision and recall also increased, indicating that dropout allowed the model to learn more robust features and effectively reduce overfitting.

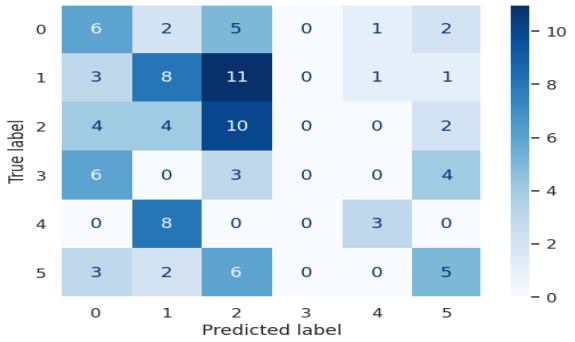

Fig-8: Confusion Matrix (Without Regularization)

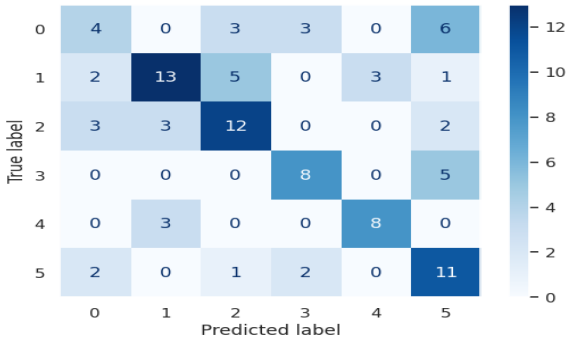

Fig-9: Confusion Matrix (With Regularization)

Metric	With Regularization	With Regularization
Accuracy	0.32	0.56
Precision (Macro)	0.30	0.56
Recall (Macro)	0.29	0.57
F1 Score (Macro)	0.28	0.56

Table 7: Matrices values before and after regulation

Model Performance and Comparison

Metric	Logistic Regression	ANN
Accuracy	0.24	0.56
Precision (Macro)	0.15	0.56
Recall (Macro)	0.19	0.57
F1 Score (Macro)	0.16	0.56

Table 8: Matrices values of Logistic Regression and ANN model

The artificial neural network (ANN) with dropout outperformed logistic regression across all metrics, particularly in the F1 score, which is crucial for imbalanced multi-class classification tasks. This improvement in the ANN's performance with dropout suggests that it is more effective at capturing complex patterns in the data while reducing the risk of overfitting.

4.4 Critical Review

A comparison was made between Logistic Regression, a simple interpretable model, and an Artificial Neural Network (ANN), a more complex non-linear model. Both models were optimized by adjusting hyperparameters, such as the regularization parameter for Logistic Regression and the learning rate and dropout for the ANN. The ANN's

performance improved significantly with dropout regularization, increasing accuracy from 32% to 56%, and enhancing precision, recall, and F1 score by reducing overfitting.

Despite tuning, Logistic Regression achieved only 24% accuracy, struggling with imbalanced datasets and complex, non-linear data. ANN faced overfitting without regularization, though dropout improved generalization. Both models performed poorly on minority classes, and techniques like oversampling or class weights could help. Feature scaling and encoding were applied, but advanced methods like PCA might improve performance. Given the imbalanced dataset, metrics like F1 score, precision, and recall are more informative than accuracy.

To improve model performance, other classification models like Random Forests, Gradient Boosting Machines (XGBoost, LightGBM), and Support Vector Machines (SVM) should be explored. Handling class imbalance through SMOTE or class weight adjustments, along with advanced regularization (e.g., L2, early stopping), can help. Additionally, techniques like PCA, feature selection, and k-fold cross-validation could enhance performance and generalization.

Part5 - Conclusion

Polynomial Ridge Regression outperformed Random Forest Regression with a lower MSE is 24,066.54 and a higher R2 score is 0.81, effectively capturing non-linear relationships in the Lifespan data. In classification, Logistic Regression struggled with 24% accuracy, while an Artificial Neural Network (ANN) improved accuracy to 56%, though it still faced overfitting and class imbalance issues. As I mentioned in part 2 (Data Exploration Section) since the target feature is a continuous numerical value, linear regression—especially Ridge Regression with polynomial features—could be more effective in capturing non-linear relationships and mitigating overfitting. Ultimately, the Ridge Regression model using a 2-degree polynomial produced the best results.

Polynomial Ridge Regression is the recommended model for deployment. Its ability to handle complex non-linear relationships, along with its superior performance in predicting the Lifespan of metal parts, makes it the best choice. This approach will provide better accuracy, more reliable predictions, and a stronger basis for manufacturing decisions than attempting to classify the parts directly.

Part6- References

Anthony Miller, J. P., 2022. A review of regression and classification techniques for analysis of common and rare variants and gene-environmental factors. *Neurocomputing,* 7 June, Volume 489, pp. 466-485.

Available at: https://www.sciencedirect.com/science/article/pii/S092523122101907X
Accessed (11 November 2024)

Ostertagová, E., 2012. Modelling using Polynomial Regression. *Procedia Engineering,* pp. 500-506.
Available at: https://www.sciencedirect.com/science/article/pii/S1877705812046085#section-cited-by
Accessed (11 November 2024)

Sperandei, S., 2014. Understanding Logistic Regression Analysis. *Biochemia Medica,* pp. 12-18.
Available at:
https://www.researchgate.net/publication/260810482_Understanding_logistic_regression_analysis
Accessed (11 November 2024)

Tu, J. V., 1996. Advantages and disadvantages of using artificial neural networks versus logistic regression for predicting medical outcomes. *Journal of Clinical Epidemiology,* pp. 1225-1231.
Available at: https://www.sciencedirect.com/science/article/pii/S0895435696000029
Accessed (11 November 2024)

Yile Ao, H. L. S. A., 2019. The linear random forest algorithm and its advantages in machine learning assisted logging regression modeling. *Journal of Petroleum Science and Engineering,* Volume 147, pp. 776-789.
Available at: https://www.sciencedirect.com/science/article/pii/S0920410518310635
Accessed (11 November 2024)