COMP1801 - Machine Learning Coursework Report

Shourya Negi – 001480606

Wordcount: 2939

1) Executive Summary (5 Marks)

This coursework investigates how machine learning can support a manufacturing company that produces safety-critical metal parts from a newly developed alloy. Currently, the company relies on destructive tests: parts are loaded until failure and then scrapped. This is slow, expensive and wastes potentially good components. The goal is to build a data-driven system that estimates both the expected lifespan of each part and whether it is safe for deployment, using measurements already collected on the production line.

The dataset contains 1,000 rows and 16 columns. Each row represents one part and includes process parameters (cooling rate, quench time, forge time, heat-treat time), alloy composition percentages (Nickel%, Iron%, Cobalt%, Chromium%), defect counts and several categorical descriptors such as part type and microstructure. The main target variable is Lifespan in hours. For safety decisions a second target is created: parts with lifespans greater than 1,500 hours are labelled “safe” and the rest “scrap”.

Two predictive tasks are addressed. The first is a **regression task**, which predicts continuous lifespan. The second is a **classification task**, which answers the simpler question of whether a part is safe or not. For regression, a **Linear Regression** model provides a baseline and is compared with a **Random Forest Regressor** that can capture non-linear patterns. For classification, **Logistic Regression** is used as a baseline and a **Gradient Boosting Classifier** is adopted as a more flexible ensemble method. All four models are implemented as scikit-learn pipelines with a shared preprocessing step: numerical features are passed through unchanged and categorical features are one-hot encoded. An 80/20 train–test split with a fixed random seed is used so that results are reproducible and comparisons are fair.

Performance is assessed on the held-out test set. For regression, mean absolute error (MAE), mean squared error (MSE) and R^2 are reported. For classification, accuracy, precision, recall, F1-score, confusion matrices and ROC curves are used. In both tasks the tree-based ensembles clearly outperform the linear baselines. The tuned Random Forest Regressor with 200 trees and maximum depth 15 achieves a test MAE of **64.60** hours, MSE of **6,756.20** and R^2 of **0.9432**, explaining about 94% of the variance in lifespan. The Gradient Boosting Classifier reaches **97%** test

accuracy, with precision and recall above 90% for both classes and only a few misclassified parts.

Throughout the report the models are treated as **decision-support tools**, not replacements for engineers. Their assumptions and limitations are discussed and final responsibility for safety-critical decisions remains with humans. Overall, the **Gradient Boosting Classifier** is recommended for the safe/scrap decision, with the **Random Forest Regressor** used alongside it to provide approximate lifespan estimates and to support process-tuning analyses.

---

2) Data Exploration (10 Marks)

Before training models, the data is explored to check structure and quality and to understand the relationships between variables.

The df.info() output confirms that the dataset has **1,000 observations** and **16 columns**, with **no missing values**. The numerical predictors are process parameters (coolingRate, quenchTime, forgeTime, HeatTreatTime), composition percentages (Nickel%, Iron%, Cobalt%, Chromium%) and defect counts (smallDefects, largeDefects, sliverDefects). The categorical predictors are partType, microstructure, seedLocation and castType. These details are summarised in **Table 1**.

COLUMN NAME	DATA TYPE	NON-NULL COUNT
LIFESPAN	float64	1000
COOLING RATE	float64	1000
QUENCH TIME	float64	1000
FORGE TIME	float64	1000
HEAT TREAT TIME	float64	1000
NICKEL%	float64	1000
IRON%	float64	1000
COBALT%	float64	1000
CHROMIUM%	float64	1000
SMALL DEFECTS	float64	1000
LARGE DEFECTS	float64	1000
SLIVER DEFECTS	float64	1000
PART TYPE	object	1000
MICROSTRUCTURE	object	1000

SEED LOCATION	object	1000
CAST TYPE	object	1000

Table 1. Dataset structure from df.info()

Descriptive statistics for Lifespan show a mean of **1,281.81** hours, median **1,254.99** hours and standard deviation **341.14** hours. Observed lifespans range from **359.71** to **2,046.41** hours (Table 2). In other words, parts last a little over 1,200 hours on average, but performance varies widely.

STATISTIC	**VALUE (HOURS)**
MEAN	1281.81
MEDIAN	1254.99
STD DEV	341.14
MIN	359.71
MAX	2046.41

Table 2. Lifespan Variable Analysis

A histogram with kernel density estimate (Figure 1) reveals a **slightly right-skewed** distribution. Most parts fall between **1,000 and 1,500 hours**, with a thinner tail of robust parts lasting beyond 1,800–2,000 hours. This suggests the process can produce very durable parts but also that there is room to reduce variability.

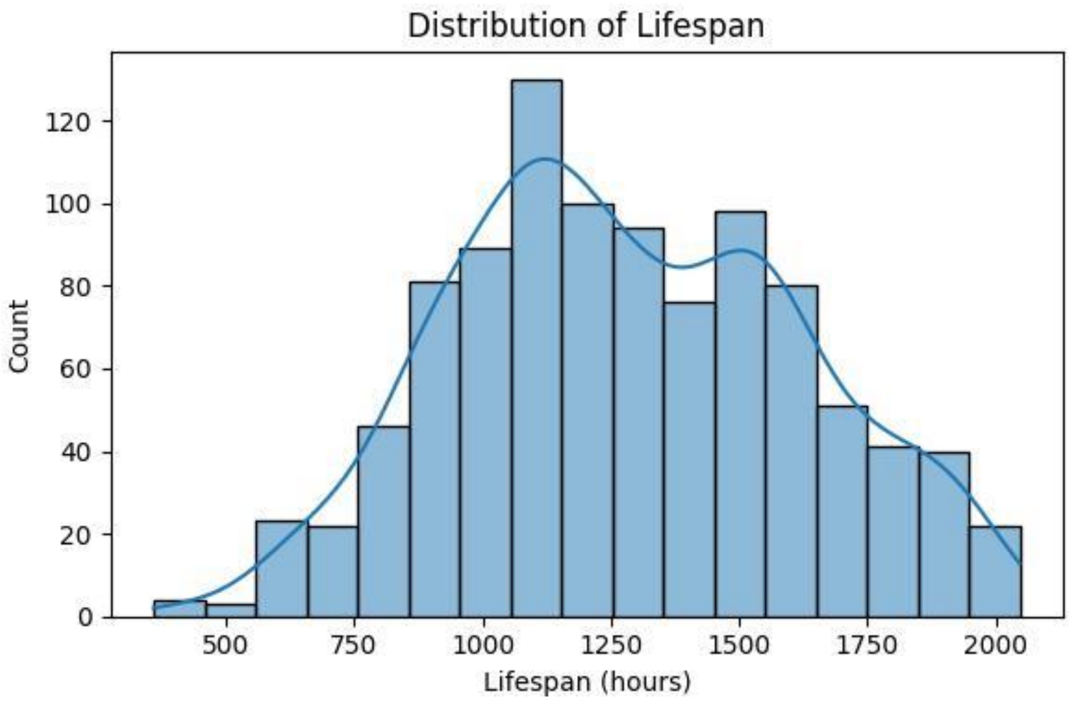

Figure 1. Distribution of Lifespan.

To examine linear relationships with Lifespan, correlation coefficients are computed. **Nickel%** has the strongest positive correlation (≈ **0.35**), while **Iron%** has a moderately strong negative correlation (≈ **−0.28**). Cooling rate and small defect counts have weaker positive correlations; other variables show only small linear effects. These values are summarised in **Table 3**. A full correlation heatmap (Figure 2) confirms that most pairwise correlations are modest, which fits with a picture of multi-dimensional, non-linear relationships.

FEATURE	**CORRELATION WITH LIFESPAN**
NICKEL%	0.3475
COOLINGRATE	0.1374
IRON%	−0.2845

Table 3. Correlation of numerical features with Lifespan

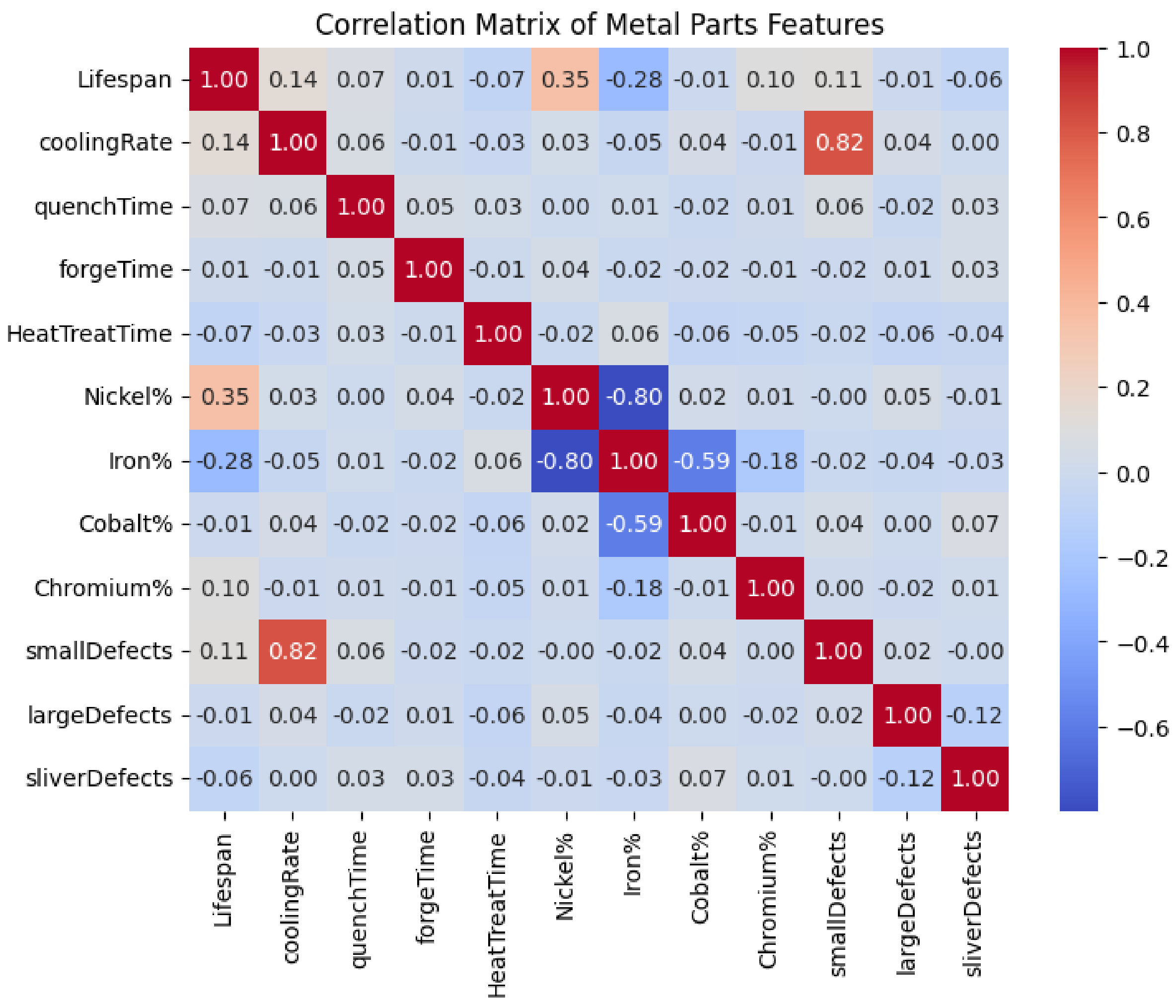

Figure 2. Correlation heatmap of numerical features.

To visualise possible non-linear structure, scatter plots are drawn for Lifespan against selected features such as Nickel%, Iron%, cooling rate and defect counts (Figure 3). Curved and fan-shaped patterns appear in several plots, reinforcing the expectation that linear models will struggle to capture all important structure.

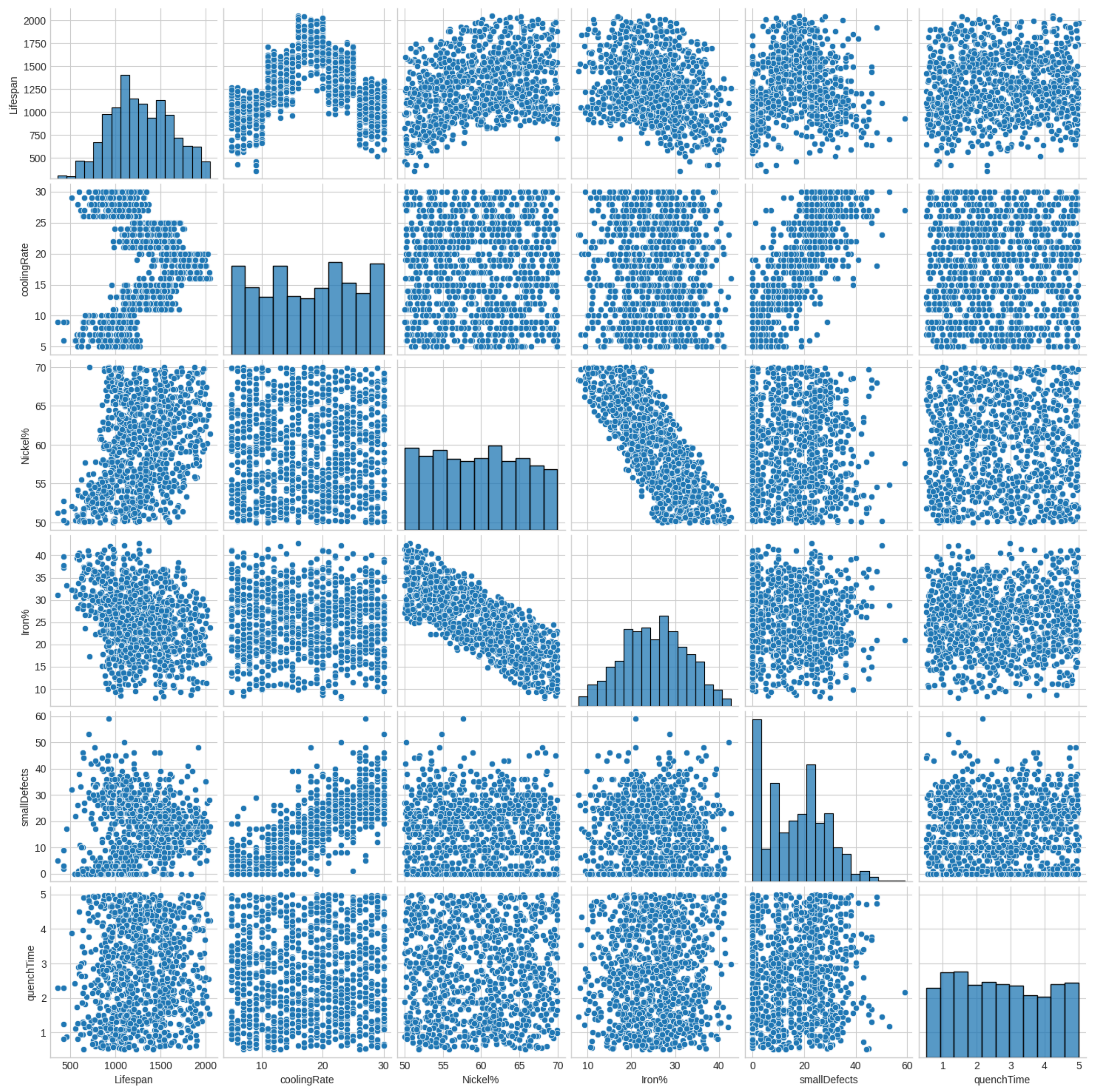

Figure 3. Scatter plot: Lifespan vs features

The exploratory phase also includes basic sanity checks. For example, the code checks for negative defect counts, which would be physically impossible. None are found, but any such values would be clipped to zero. This small step illustrates the use of domain knowledge to protect against data problems.

Overall, exploration leads to three conclusions. The data is clean and ready for modelling. Alloy composition and some process conditions clearly influence lifespan, with Nickel and Iron especially important. Finally, relationships are complex and partly non-linear, which motivates using tree-based ensembles in addition to linear baselines.

3) Regression Implementation (30 Marks)

3.1) Methodology (10 Marks)

The regression task predicts each part's **lifespan in hours**. The feature matrix **X** contains all columns except Lifespan; the target vector **y_reg** is the Lifespan column. An **80/20 train–test split** with random_state=42 is used.

A ColumnTransformer builds a unified **preprocessing pipeline**. Numerical features are passed through unchanged; categorical features are one-hot encoded. This preprocessor is included inside scikit-learn Pipeline objects so that all models share identical preprocessing and data leakage is avoided.

Two regression models are evaluated:

1. **Linear Regression**, a multivariate linear model that fits a single straight-line relationship between features and lifespan. It is simple and interpretable but relies on strong linearity assumptions.
2. **Random Forest Regressor**, an ensemble of decision trees. Each tree is trained on a bootstrap sample and uses a subset of features at each split. Predictions are averaged across trees. Random Forests can capture non-linear effects and interactions at the cost of lower transparency.

To tune the Random Forest, the number of trees (n_estimators) is varied from **10 to 290 in steps of 20**, with max_depth=15 to limit overfitting. For each setting, the model is fit on the training data and the training and test MSE are recorded. These values are reported in **Table 4** and plotted in **Figure 4**.

N_ESTIMATORS	TRAINING MSE	TEST MSE
10	1545.66	7630.60
30	1169.55	7260.59
50	1106.47	7295.25
70	1047.01	7168.06
90	1025.25	6974.70
110	1018.43	6830.82
130	1006.97	6842.74
150	1001.19	6753.01
170	1000.24	6735.47
190	997.17	6745.98
210	998.93	6734.98
230	996.75	6728.90
250	988.18	6735.29
270	985.35	6708.14
290	983.43	6706.50

Table 4. Random Forest n_estimators vs MSE

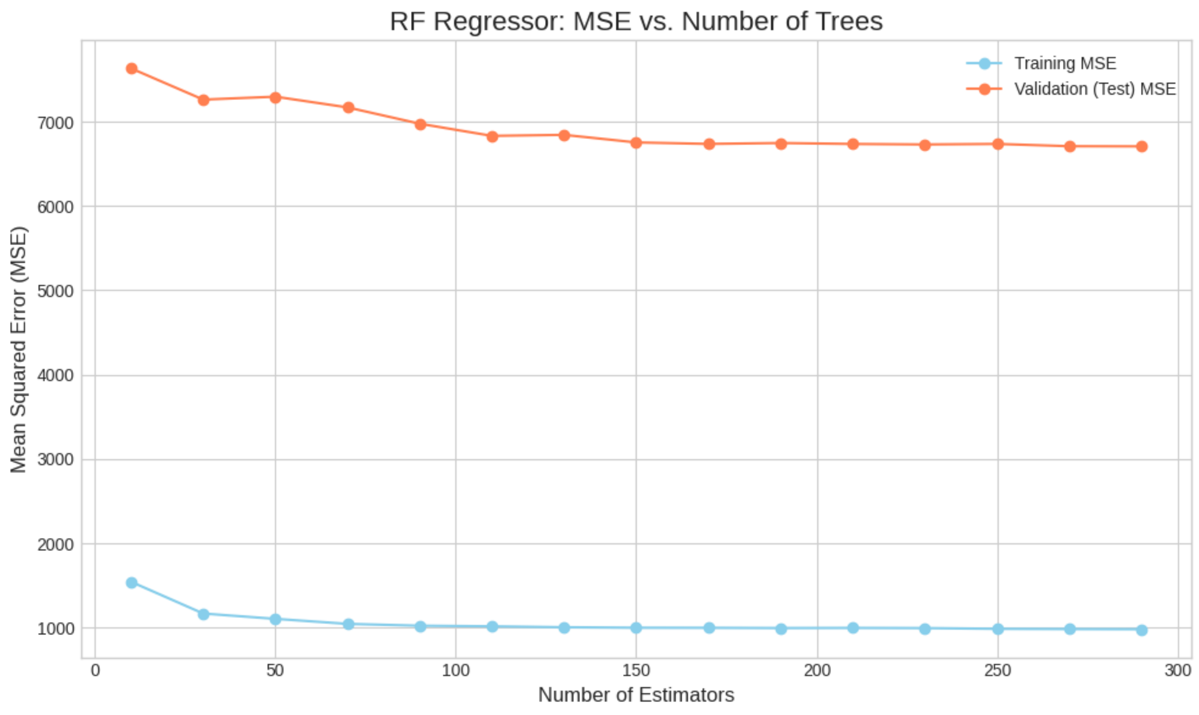

Figure 4. RF Regressor: MSE vs. Number of Trees

The curve behaves as expected. With 10 trees, training MSE is about **1,545.66** and test MSE around **7,630.60**. As more trees are added, training MSE drops below 1,000 and test MSE falls to around **6,700–6,750**, after which gains become very small. A configuration with **200 trees** is therefore chosen as a good balance between accuracy and complexity.

Two diagnostic plots are then created. A **predicted-versus-actual** plot for the Random Forest (Figure 5) shows how closely predictions follow the 45-degree line. A **feature-importance** chart (Figure 6) ranks the most informative features based on the Random Forest's impurity-based importance scores.

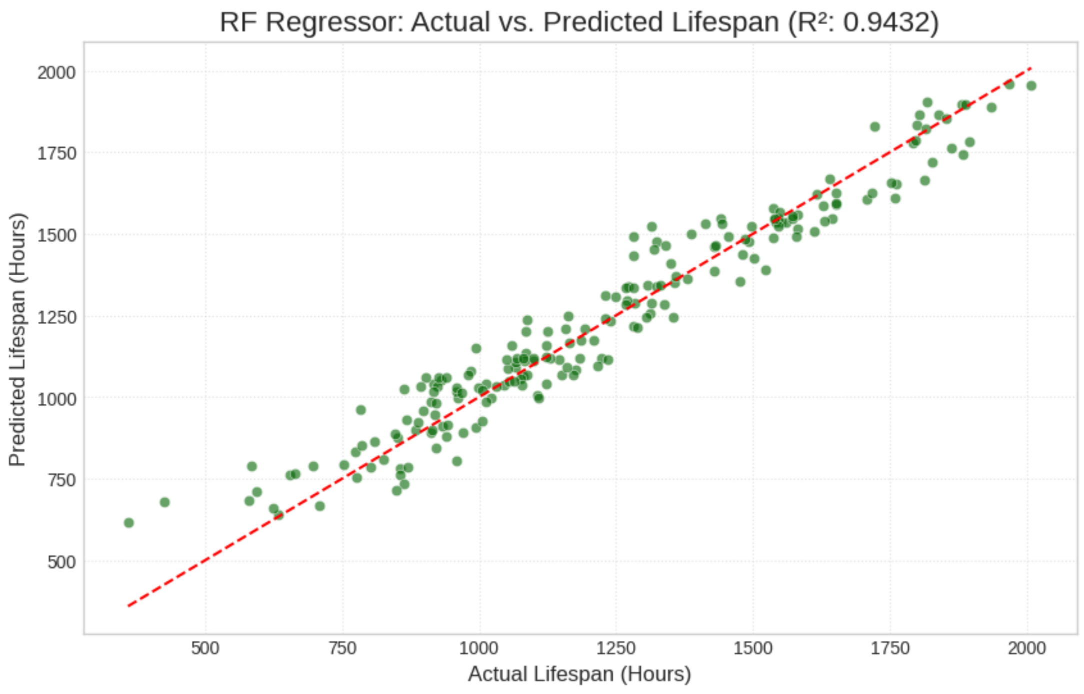

Figure 5. RF Regressor: Actual vs Predicted Value

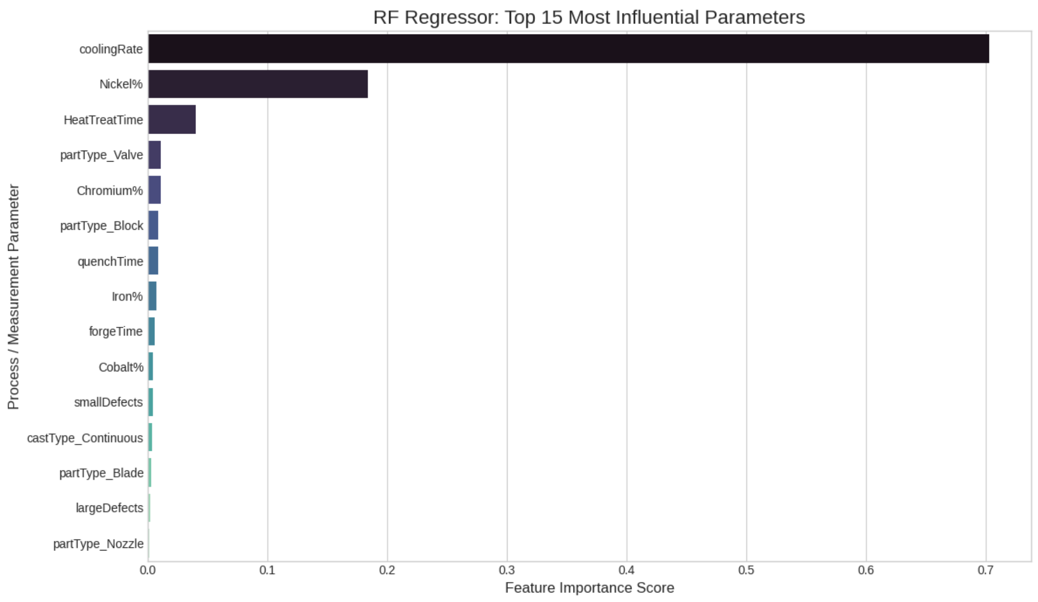

Figure 6. Random Forest Regressor's most important features

3.2) Evaluation (15 Marks)

Both regression models are evaluated on the test set using MAE, MSE and R^2.

For **Linear Regression** the test metrics are:

- MAE = **271.37** hours
- MSE = **98,856.45**
- R^2 = **0.1686**

Thus the linear model explains only about 17% of the variance in lifespans and is typically off by more than 270 hours, which is large relative to the target scale.

For the **Random Forest Regressor** the metrics are:

- MAE = **64.60** hours
- MSE = **6,756.20**
- R^2 = **0.9432**

These results, summarised in **Table 5**, show a dramatic improvement. The Random Forest reduces MSE by more than an order of magnitude and increases R^2 from 0.17 to 0.94. An average error of about 65 hours is far more acceptable when parts are expected to last over 1,000 hours.

MODEL	MAE (HOURS)	MSE	R^2
LINEAR REGRESSION	271.37	98,856.45	0.1686
RANDOM FOREST REGRESSOR	64.60	6,756.20	0.9432

Table 5. Regression model performance on the test set.

The predicted-versus-actual plot (Figure 5) supports this: Random Forest predictions cluster closely around the diagonal line, indicating accurate, unbiased predictions. The feature-importance plot (Figure 6) shows that the model relies heavily on composition variables—especially Nickel% and Iron%—alongside key process parameters and defect counts. This matches the earlier exploratory findings and provides engineers with a clear sense of which levers matter most.

3.3) Critical Review (5 Marks)

Despite strong performance, the regression approach has limitations.

First, evaluation is based on a single **train–test split**. While the split is fixed, results may still depend somewhat on which examples fall into the test set. K-fold cross-validation would provide a more robust estimate of generalisation performance.

Second, the **hyperparameter search** is narrow. Only the number of trees and a fixed depth are explored. Tuning additional parameters such as min_samples_leaf, max_features or alternative depths could yield a more efficient or slightly more accurate model.

Third, the analysis treats each part as an **independent observation**. In practice there may be batch or time effects—for example, changes in raw material quality or equipment calibration—that are not explicitly modelled. These could be captured using extra features or by monitoring model performance over time.

Finally, while Random Forests offer feature importances, they are still less transparent than a simple linear model. Tools such as partial-dependence plots or SHAP values could be used in future work to provide more detailed explanations and to increase trust among engineers.

---

4) Classification Implementation (40 Marks)

4.1) Feature Crafting (10 Marks)

For day-to-day operations the key question is whether each part is **safe to use** or should be **scrapped**. To support this, the continuous Lifespan is converted into a binary label using a **1,500-hour** threshold:

- Lifespan > 1500 → 1 (Safe)
- Lifespan ≤ 1500 → 0 (Scrap)

The Lifespan column is then removed from the feature matrix so that classifiers rely only on the process and inspection features that would be available at decision time.

Using the same 80/20 split, class counts on the test set are inspected. There are **150 scrap** and **50 safe** parts (Table 6), so the data is moderately **imbalanced**. A trivial model that always predicts scrap would already achieve 75% accuracy, so recall and the confusion matrix are crucial for judging performance.

CLASS	LABEL	COUNT
SCRAP	0	150
SAFE	1	50

Table 6. The test set's class distribution.

4.2) Methodology (10 Marks)

The classification workflow mirrors the regression setup. The same ColumnTransformer is reused inside two new pipelines:

1. **Logistic Regression (baseline).**
   This model assumes a linear relationship between the transformed features and the log-odds of the safe class. It is relatively interpretable and outputs class probabilities.
2. **Gradient Boosting Classifier (ensemble).**
   This model builds an ensemble of shallow trees in sequence, each correcting the errors of the previous ensemble. In this coursework it uses **100 trees** with **learning rate 0.2** and a fixed random seed. Gradient boosting is well suited to structured tabular data and can capture complex non-linear boundaries.

Both models are trained on the training data and evaluated on the test data. For each, a classification report, confusion matrix and ROC curve are produced. Accuracy is reported, but more weight is given to precision and recall for each class and to the confusion matrices, which show the actual error patterns.

4.3) Evaluation (15 Marks)

For **Logistic Regression**, the test accuracy is **0.79**. Class-wise metrics (Table 7) show that for **scrap** parts (class 0) precision is **0.79** and recall **0.99**, while for **safe** parts (class 1) precision is **0.83** but recall only **0.20**. The confusion matrix (Table 8 and Figure 7) reveals that out of 50 genuinely safe parts, only 10 are correctly classified; the remaining 40 are incorrectly labelled scrap.

CLASS	PRECISION	RECALL	F1-SCORE	SUPPORT
0 (SCRAP)	0.79	0.99	0.88	150
1 (SAFE)	0.83	0.20	0.32	50
ACCURACY	-	-	0.79	200

Table 7. Classification metrics for Logistic Regression (test set)

	PREDICTED 0 (SCRAP)	PREDICTED 1 (SAFE)
ACTUAL 0 (SCRAP)	148	2
ACTUAL 1 (SAFE)	40	10

Table 8. Confusion matrix for Logistic Regression (test set)

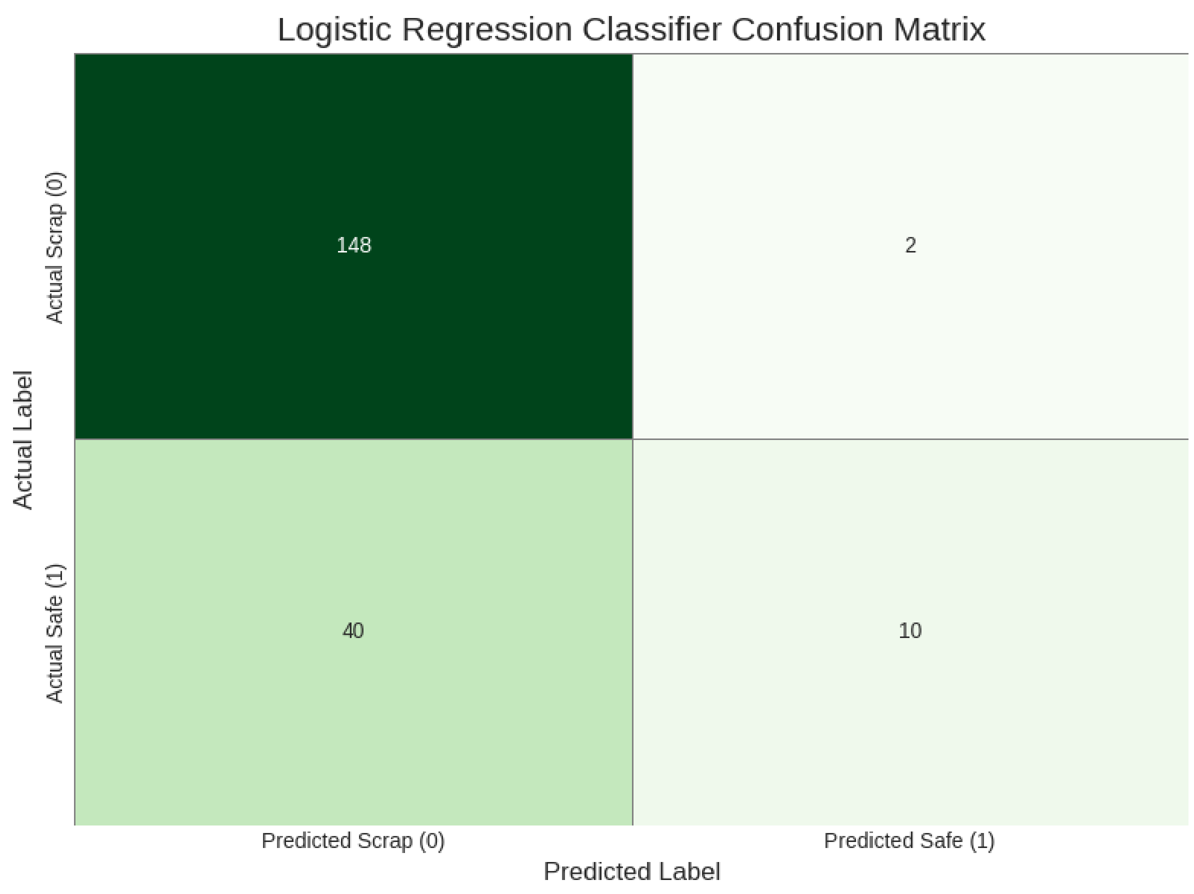

Figure 7. Confusion matrix for Logistic Regression

This makes the model extremely **conservative**. It almost never passes a scrap part as safe, but discards the majority of safe parts. While this is cautious from a safety perspective, it would be very costly in practice because many good parts would be scrapped.

The **Gradient Boosting Classifier** performs much better. On the same test set it achieves **0.97** accuracy. For scrap parts, precision is **0.97** and recall **0.99**; for safe parts, precision is **0.96** and recall **0.92** (Table 9). The confusion matrix (Table 10 and Figure 8) shows only **2 scrap parts** wrongly labelled safe and **4 safe parts** wrongly scrapped; 194 of 200 parts are correct.

CLASS	PRECISION	RECALL	F1-SCORE	SUPPORT
0 (SCRAP)	0.97	0.99	0.98	150
1 (SAFE)	0.96	0.92	0.94	50
ACCURACY	-	-	0.97	200

Table 9. Metrics for classifying the Gradient Boosting Classifier (test set)

	PREDICTED 0 (SCRAP)	**PREDICTED 1 (SAFE)**
ACTUAL 0 (SCRAP)	148	2
ACTUAL 1 (SAFE)	4	46

Table 10. Confusion matrix for the Gradient Boosting Classifier (test set)

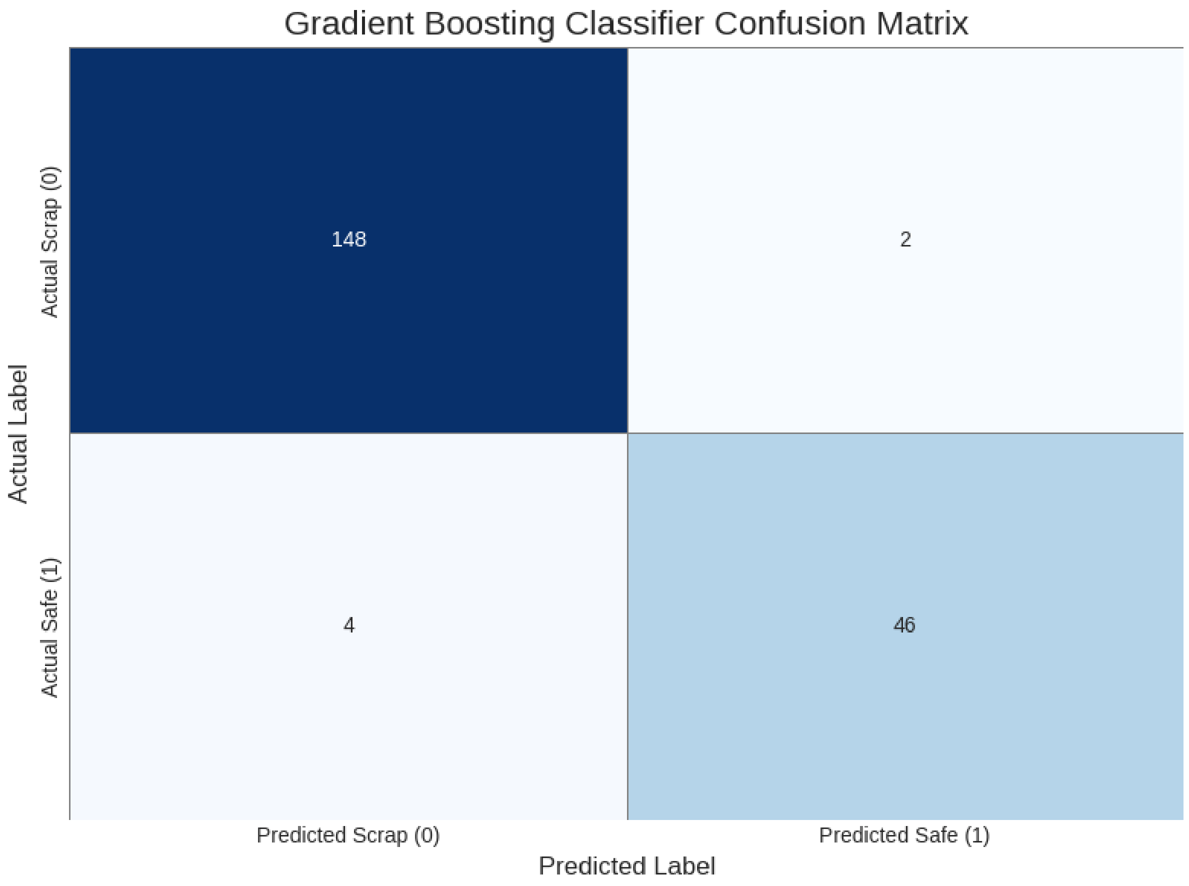

Figure 8. Confusion matrix for the Gradient Boosting Classifier

ROC curves (Figure 9) summarise performance across thresholds. The Gradient Boosting curve lies above the Logistic Regression curve, and its area under the curve is higher, confirming superior discriminative ability.

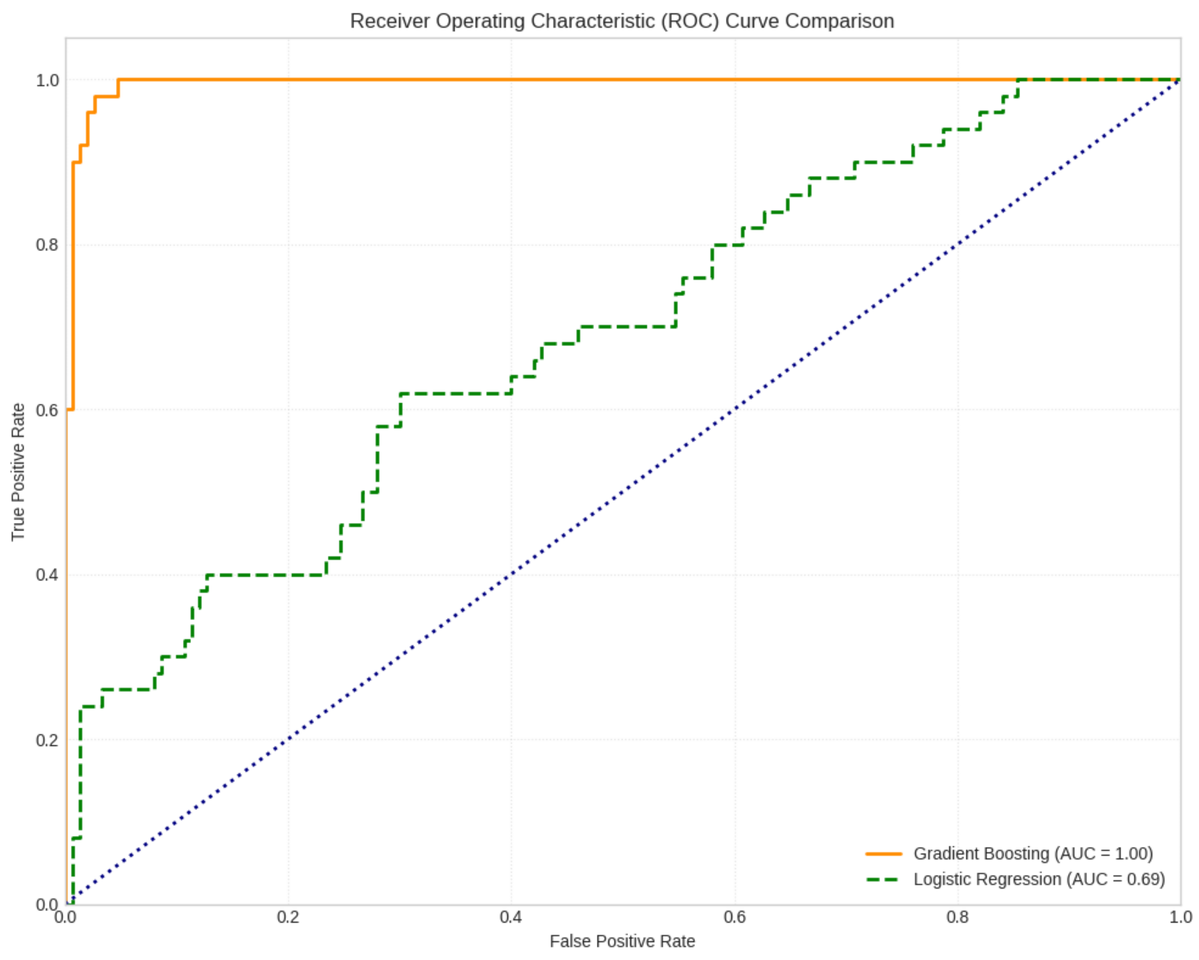

Figure 9. ROC curves for the Gradient Boosting Classifier and Logistic Regression

In practical terms Gradient Boosting offers a much better **trade-off**. It maintains almost perfect recall for scrap parts, keeping safety risks low, while also correctly recovering most safe parts that Logistic Regression would throw away. It is therefore both safer and more efficient.

4.4) Critical Review (5 Marks)

Even with strong performance, there are caveats.

The dataset is **imbalanced**. The current approach relies on the algorithms and loss functions to handle this implicitly. If class proportions changed, explicit techniques such as class weighting or resampling might be needed to maintain good performance, particularly for the minority class.

Hyperparameter tuning for Gradient Boosting is limited. Only one configuration with 100 trees and learning rate 0.2 is used. A broader search over depth, learning rate and regularisation parameters could produce an even more robust model or reduce complexity without sacrificing accuracy.

Evaluation uses mostly **symmetric metrics**. In reality, misclassifying a scrap part as safe is more serious than misclassifying a safe part as scrap. Incorporating explicit cost matrices or tuning the decision threshold to reflect asymmetric costs would make the evaluation more realistic.

Finally, once deployed, the classifier would need **ongoing monitoring**. Manufacturing processes change over time, and data distributions can drift. Regular checks, periodic retraining and the ability for engineers to inspect and override predictions are important aspects of a responsible, human-centred deployment.

---

5) Conclusion (5 Marks)

This coursework has developed a complete machine-learning pipeline to support quality control for metal parts made from a sensitive new alloy. Data exploration showed that the dataset is clean, that lifespans are slightly right-skewed, and that alloy composition—especially Nickel and Iron—plus key process parameters and defect counts are important drivers of durability. Relationships appear complex and non-linear, motivating the use of tree-based ensembles alongside linear baselines.

For the **regression task**, Linear Regression served as a baseline but captured only a small fraction of the variance and produced large average errors. The **Random Forest Regressor**, tuned mainly via the number of trees and depth, delivered much stronger performance with R^2 of 0.9432 and MAE around 64.60 hours, making it a useful tool for estimating lifespans and exploring how changes in process settings might affect durability.

For the **classification task**, the continuous lifespan was converted into a binary safe/scrap label using a 1,500-hour threshold. Logistic Regression turned out to be overly conservative, discarding most safe parts. The **Gradient Boosting Classifier** achieved both high safety and high efficiency, with 97% accuracy and high recall for both classes, and only a handful of misclassifications in the confusion matrix.

Throughout, the models are framed as **tools to support human experts** rather than as autonomous decision makers. Their limitations are acknowledged, and suggestions are made for stronger hyperparameter tuning, cost-sensitive evaluation and ongoing monitoring.

The final recommendation is two-fold. For the operational decision of whether each part should be shipped or scrapped, the company should deploy the **Gradient Boosting Classifier**. For estimating expected lifespans and understanding how process changes influence durability, it should use the **Random Forest Regressor** as a complementary tool. Together, these models could reduce reliance on destructive testing, cut costs and help engineers improve the manufacturing process in a controlled and responsible way.