COMP1801 - Machine Learning Coursework Report

Hitesh Taneja – 001406939

Word Count:2523

1. Executive Summary

This report addresses the problem of predicting the lifespan of metal parts and classifying parts as defective or non-defective based on their lifespan and manufacturing parameters. Accurate prediction and classification are crucial for optimizing manufacturing processes and ensuring quality control.

For the regression task, Linear Regression and XGBoost were implemented to predict the lifespan of metal parts. XGBoost outperformed Linear Regression, achieving a significantly higher R^2 score of 0.96, demonstrating its capability to capture non-linear relationships and feature interactions.

For the classification task, Random Forest and Logistic Regression were employed to determine whether a part meets the minimum lifespan threshold of 1500 hours. Random Forest emerged as the superior model, achieving an accuracy of 90% and an F1-score of 0.84, effectively handling the imbalanced dataset.

The report concludes by recommending XGBoost for regression tasks due to its predictive power and Random Forest for classification tasks due to its robust handling of imbalanced data and superior classification performance. These models provide actionable insights to enhance manufacturing processes and part quality.

2. Data Exploration

Dataset Loading

The dataset was loaded using the Python pandas library. The CSV file, containing 1000 rows and 16 columns, was imported and inspected using .info() and .describe(). No missing or duplicate values were identified, ensuring the dataset was clean and ready for analysis.

Key Patterns and Relationships

To better understand the relationships among numerical features, a correlation matrix was generated (refer to Figure 2.2). The matrix highlights that Nickel% has a moderate positive correlation with Lifespan (+0.32), while Iron% exhibits a moderate negative correlation (-0.25). Additionally, smallDefects and coolingRate show a strong positive correlation (+0.81), suggesting a potential link between these variables. These insights guided the feature selection for subsequent regression modeling.

The data exploration revealed several important patterns and relationships relevant to predicting the lifespan of metal parts:

1. Target Variable Distribution:
    - The target variable, Lifespan, ranges from 418 to 2135 hours, with a slightly skewed near-normal distribution. (Refer to the KDE plot of Lifespan in Figure 2.1.)

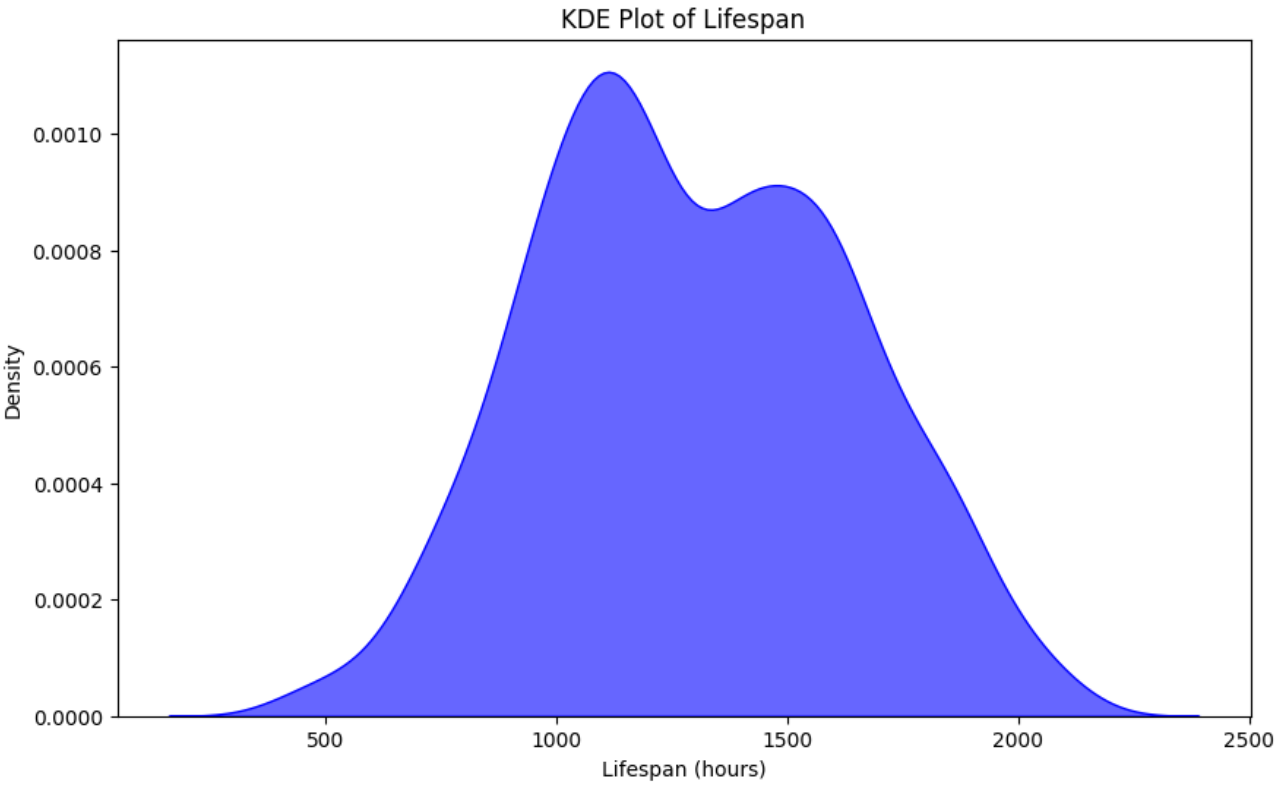

Figure 2.1: KDE Plot of Lifespan showing the distribution of the target variable, indicating a slightly skewed near-normal distribution.

2. To better understand the relationships among numerical features, a correlation matrix was generated (refer to Figure 2.2). The matrix highlights that Nickel% has a moderate positive correlation with Lifespan (+0.32), while Iron% exhibits a moderate negative correlation (-0.25). Additionally, smallDefects and coolingRate show a strong positive correlation (+0.81), suggesting a potential link between these variables. These insights guided the feature selection for subsequent regression modeling.

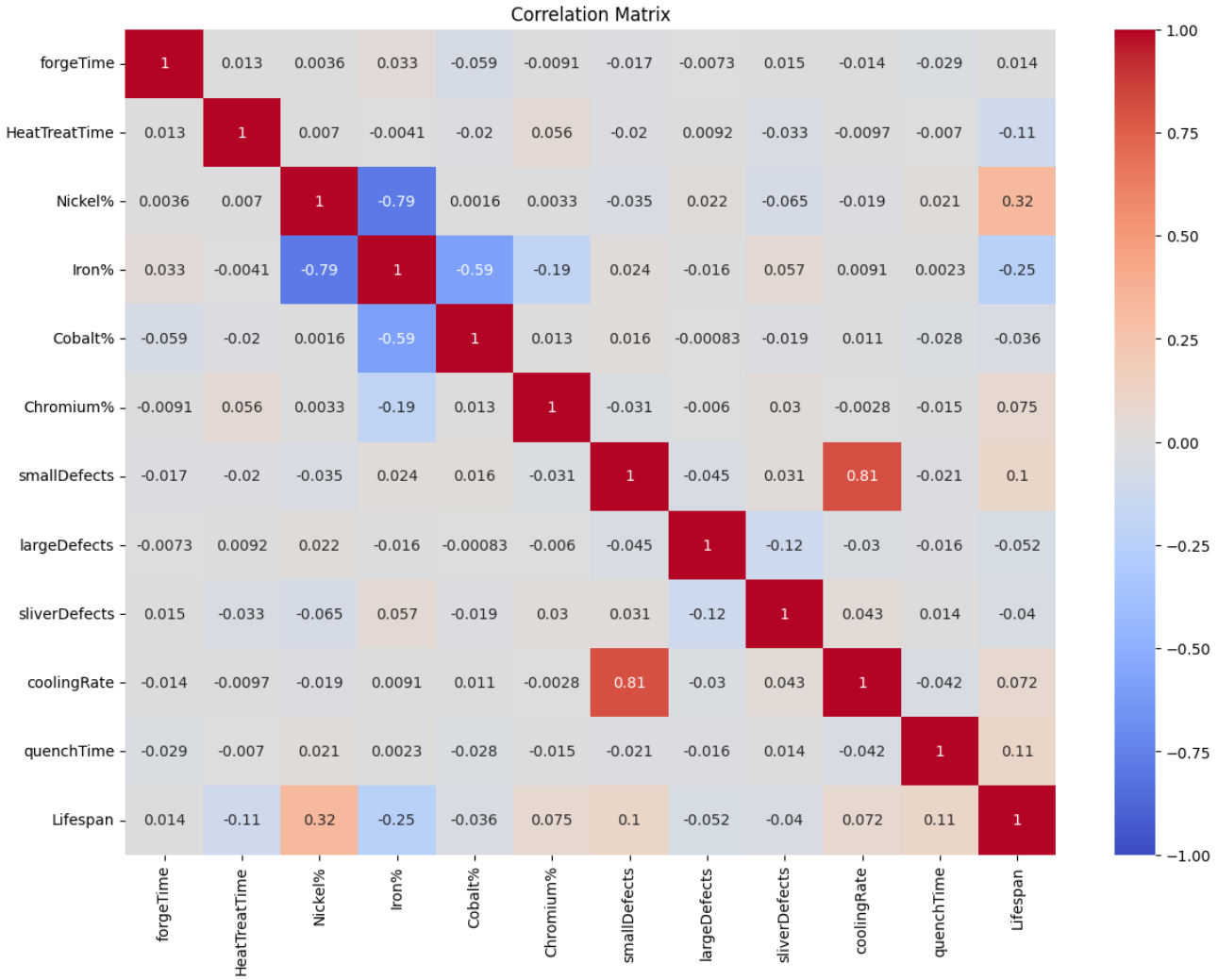

Figure 2.2: Correlation matrix showing relationships between numerical features, highlighting key correlations relevant to feature selection.

3. Strongly Correlated Features:
    - Nickel% has a moderate positive correlation (+0.32) with Lifespan, suggesting that higher nickel content enhances durability. (Refer the scatter plot of Nickel% vs. Lifespan in Figure 2.3)

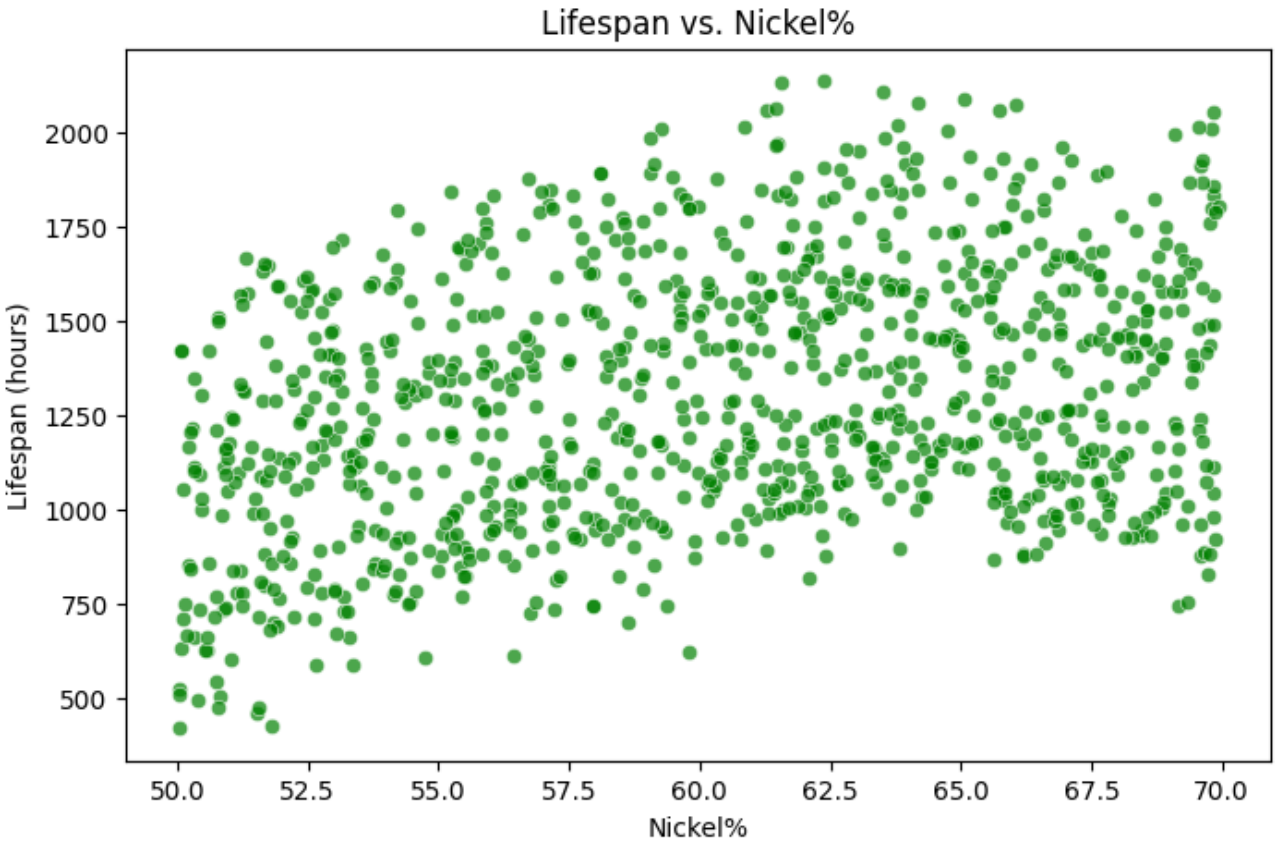

Figure 1.3: Scatter plot of Nickel% vs. Lifespan, demonstrating a moderate positive correlation.

- Iron% has a moderate negative correlation (-0.25) with Lifespan, indicating that increased iron content reduces longevity, likely due to brittleness. (Refer scatter plot of Iron% vs. Lifespan in Figure 2.4)

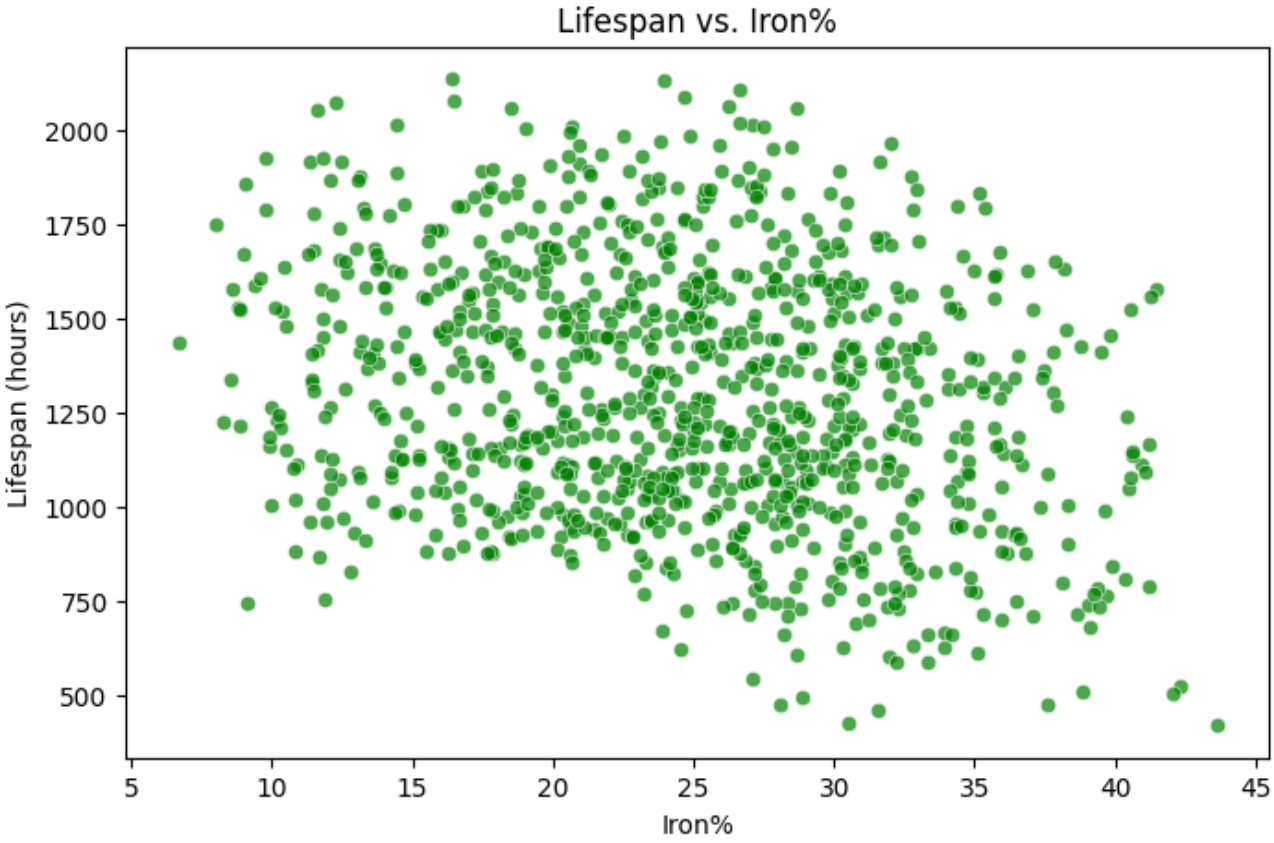

Figure 2.4: Scatter plot of Iron% vs. Lifespan, illustrating a moderate negative correlation.

4. Feature Interactions:
    - coolingRate and smallDefects are strongly correlated (+0.81). Faster cooling rates result in more small defects, which can indirectly affect lifespan. (Refer to smallDefects vs. coolingRate plot in Figure 2.5)

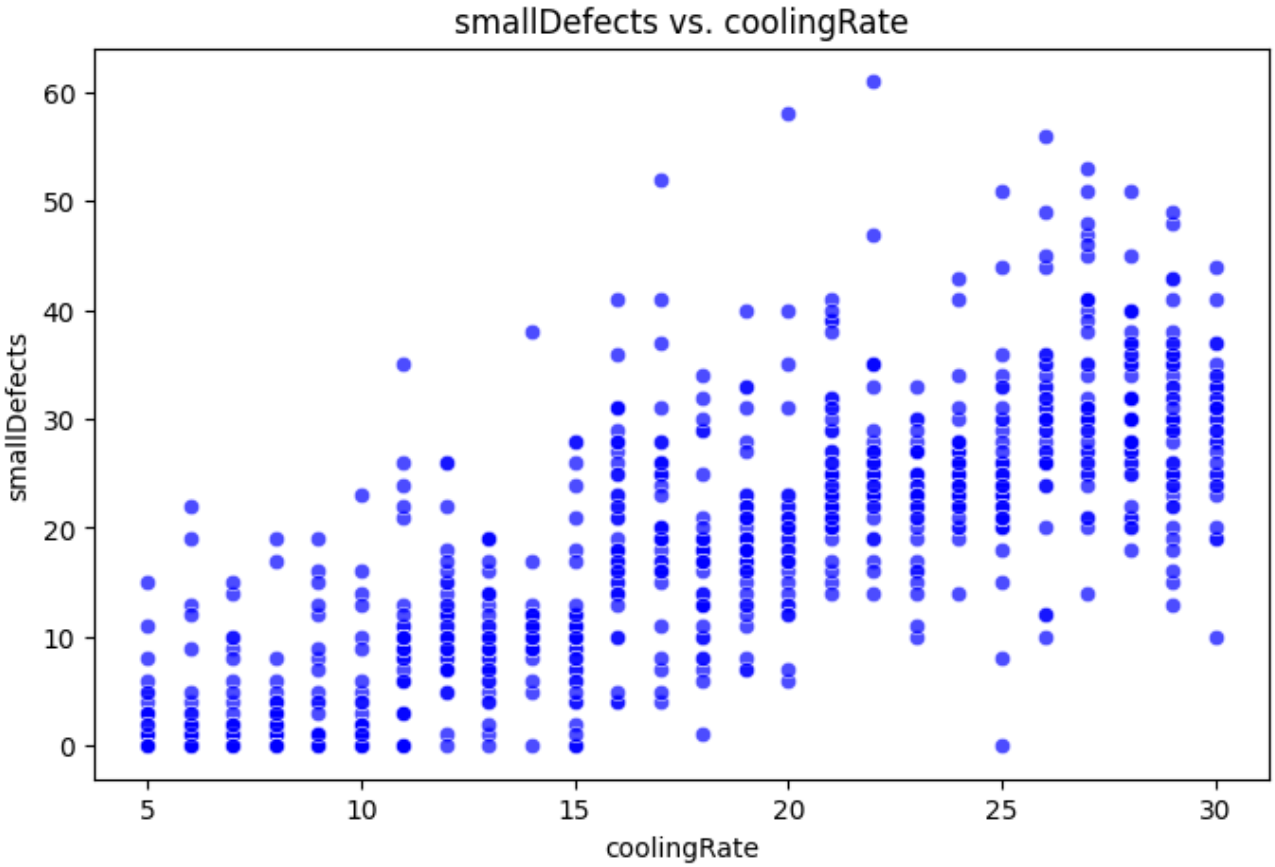

Figure 2.5: Scatter plot of smallDefects vs. coolingRate, indicating a strong positive correlation between cooling rate and small defects..

- Nickel% and Iron% exhibit a strong negative correlation (-0.79), reflecting material trade-offs during manufacturing. (refer Figure2.6)

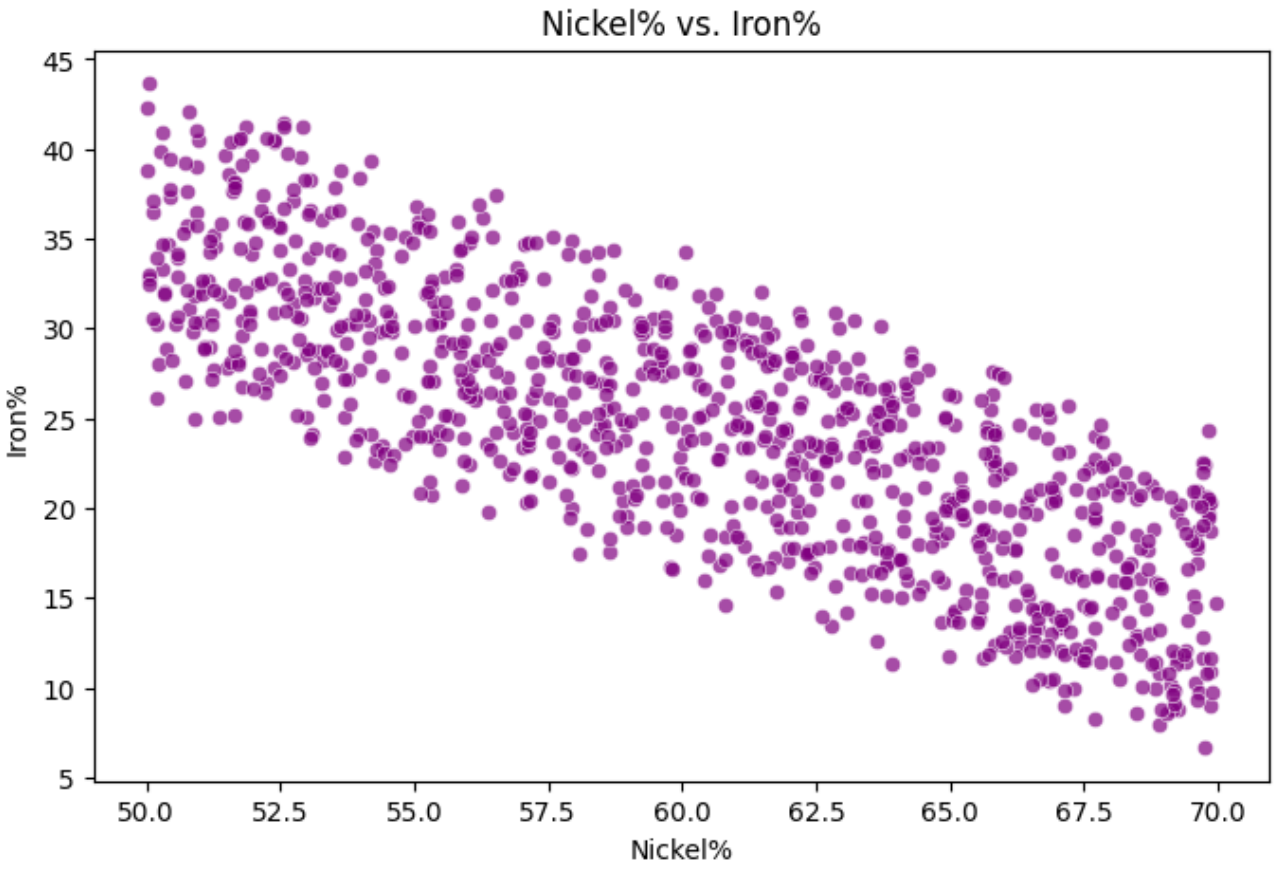

Figure 2.6: Scatter plot of Nickel% vs. Iron%, highlighting a strong negative correlation between these two features.

5. Weak Relationships:
    - Features like HeatTreatTime, quenchTime, and sliverDefects showed weak correlations with Lifespan, indicating limited direct impact.

Selected Features for Modeling

The following features were chosen based on their correlation with Lifespan and their relevance:

- Key Predictors: Nickel%, Iron%, smallDefects, largeDefects, coolingRate.
- Categorical Features: microstructure, castType—representing material and manufacturing process characteristics.

These features capture material composition, defects, and process parameters, which are critical for predicting part longevity.

Modeling Approach

The problem requires predicting a continuous target variable (Lifespan), aligning it with a regression approach. Among machine learning models, the following are expected to perform well:

1. Gradient Boosting Regressor (XGBoost):
    - Handles non-linear relationships and feature interactions effectively.
    - Provides robust predictions on small to medium-sized datasets.
2. Linear Regression:
    - Serves as a baseline for comparison.
    - Offers simplicity and interpretability, especially for features with linear trends.

Given the data's small size and the presence of mixed feature types, Gradient Boosting is expected to provide the most accurate predictions while offering feature importance insights to refine future models.

3. Regression Implementation

3.1 Methodology

Why Linear Regression and XGBoost?

Linear Regression is chosen because is a simple method that helps us understand if there's a direct relationship between Target Variable (Lifespan) and features, as identified in the data exploration phase (e.g., moderate correlations with Nickel% and Iron%). XGBoost is a more powerful method that can find complex patterns and relationships that the simpler method might miss. XGBoost's success in similar regression problems has been widely reported in the literature (e.g., Chen & Guestrin, 2016), making it an appropriate choice for this dataset. The exploratory analysis indicated significant non-linearities and feature interactions (e.g., smallDefects vs. coolingRate), further justifying the use of XGBoost.

Preprocessing Routine

Preprocessing was performed to ensure the data was suitable for regression modeling.

- One-Hot Encoding: Applied to categorical features (partType, microstructure, castType) to convert them into numeric format without introducing ordinal relationships.
- Feature Scaling: Standardization was applied to numerical features (Nickel%, Iron%, etc.) to ensure consistent ranges across all inputs, improving model convergence for Linear Regression. While XGBoost is robust to unscaled features, scaling aids interpretability.
- Train-Test Split: Data was split into 80% training and 20% testing to evaluate model generalization. A random state was fixed for reproducibility.

These steps ensured the data was clean, balanced, and prepared for accurate modeling.

Hyperparameter Tuning Framework

- Linear Regression: We applied Ridge Regression (L2 regularization) to prevent overfitting by penalizing large coefficients. The regularization strength (alpha) was optimized via Grid Search.
- XGBoost: Randomized Search was used to tune essential hyperparameters:
    - Learning Rate (eta): Controls the step size during training, balancing speed and stability.
    - Number of Trees (n_estimators): Optimized to balance bias and variance.
    - Tree Depth (max_depth): Prevents overfitting by limiting tree complexity.
    - Subsampling (subsample, colsample_bytree): Introduced randomness to reduce overfitting.
    - Minimum Child Weight (min_child_weight): Prevents overly specific splits.
    - Hyperparameter tuning improved model performance significantly, as observed in the evaluation results.

3.2 Evaluation

Experimental Process

We trained baseline models for both Linear Regression and XGBoost to establish initial performance. Hyperparameter tuning was conducted as outlined above, and the optimized models were evaluated.

Figure 3.3 highlights the feature importance as determined by the XGBoost model. smallDefects and Nickel% emerge as the most critical predictors of Lifespan, aligning with insights from the correlation analysis in Section 2.

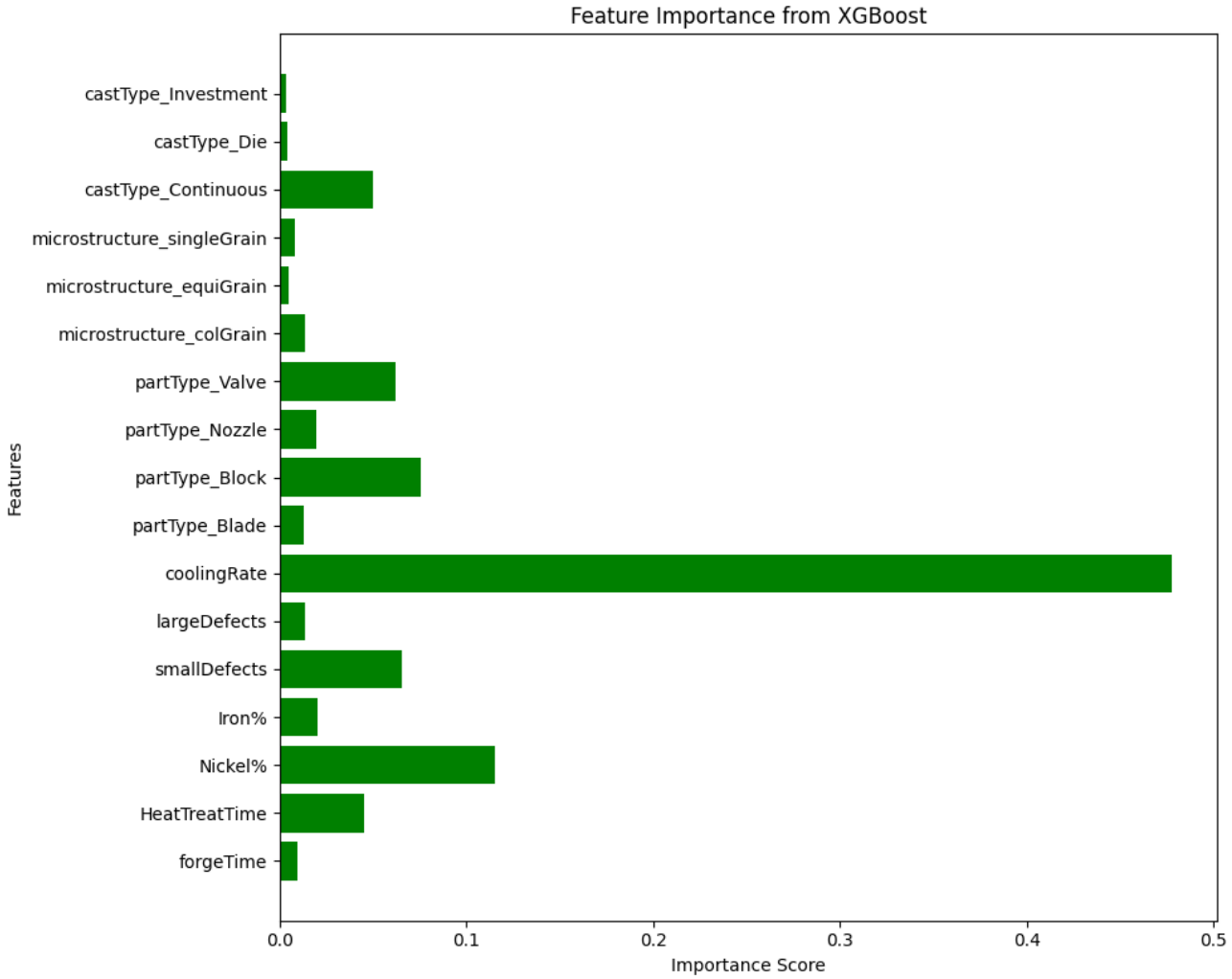

Figure 3.1: Feature importance plot for XGBoost, identifying the most influential features for predicting Lifespan, with smallDefects and Nickel% contributing the most.

The table below summarizes the results:

Model	MAE(Baseline)	MSE(Baseline)	R^2(Baseline)	MAE(Tuned)	MSE(Tuned)	R^2(Tuned)
Linear Regression	252.05	92,497.68	0.11	251.25	92,167.73	0.11
XGBoost	59.88	6,149.63	0.94	54.23	4,570.21	0.96

The scatter plots (Figures 3.2 and 3.3) compare predictions vs. actual values for Linear Regression and XGBoost models, respectively. The XGBoost model demonstrates a better fit, with predictions closely aligning with actual values.

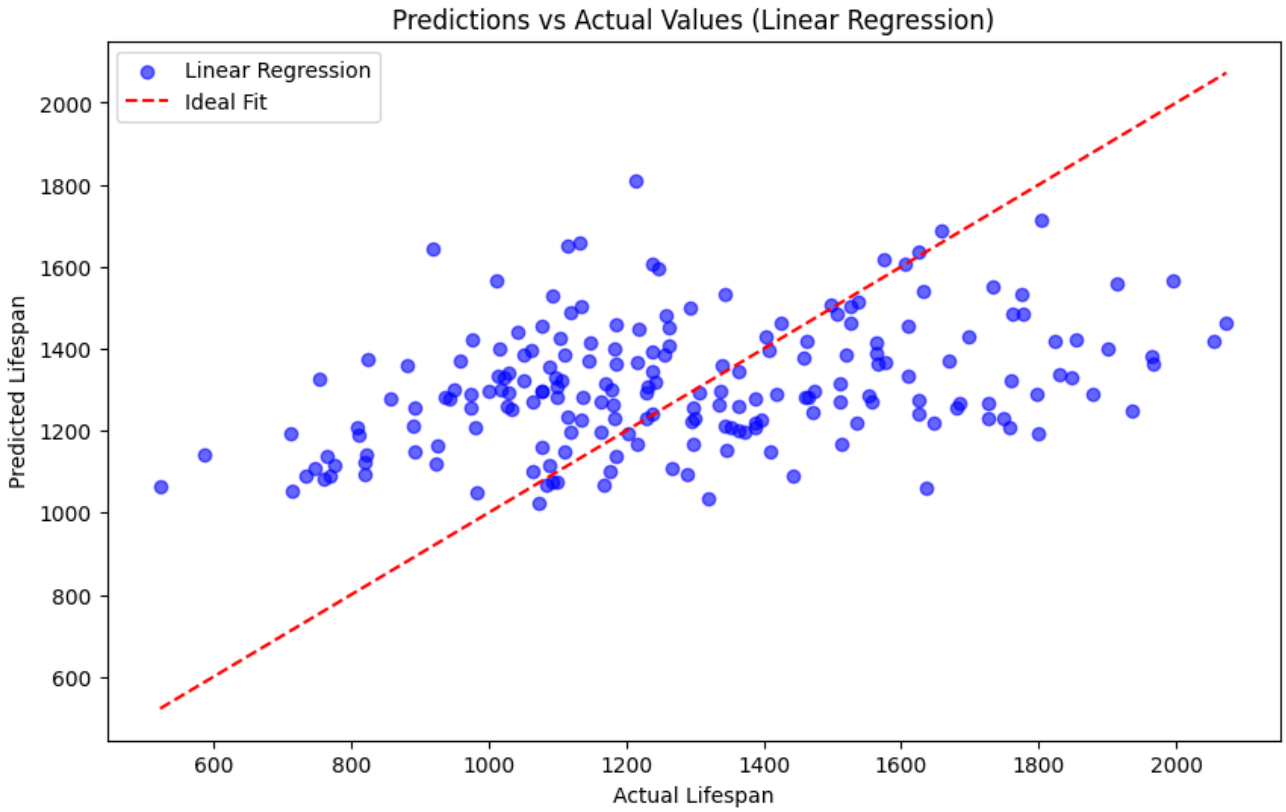

Figure 3.2: Scatter plot showing predictions vs. actual values for the Linear Regression model, illustrating its limited predictive accuracy with points scattered away from the diagonal.

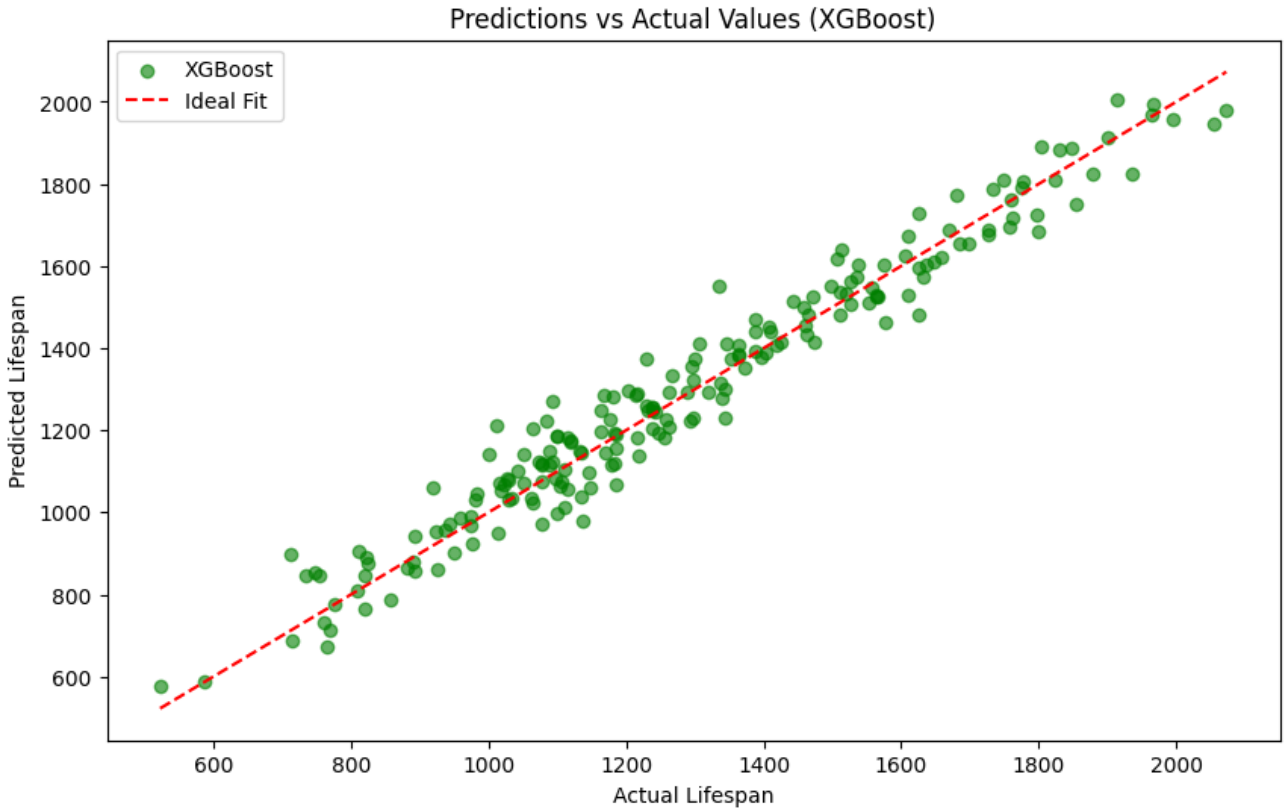

Figure 3.3: Scatter plot showing predictions vs. actual values for the XGBoost model, demonstrating higher predictive accuracy with points closely aligned to the diagonal.

Final Model Details

- Linear Regression (Tuned): Ridge Regression with alpha=1.0.
- XGBoost (Tuned): n_estimators=200, learning_rate=0.05, max_depth=5, subsample=0.8, colsample_bytree=0.8, min_child_weight=3.

Evaluation Metrics

- MAE (Mean Absolute Error): Measures average prediction error; low values indicate better predictions.
- MSE (Mean Squared Error): Penalizes large errors more heavily; lower MSE indicates better model fit.
- R^2 Score: Explains the proportion of variance captured by the model; higher values indicate better performance.

XGBoost outperformed Linear Regression on all metrics, achieving low MAE and MSE with an R^2 of 0.96, indicating excellent predictive performance.

Comparison and Recommendation

XGBoost is the superior model due to its ability to handle non-linear relationships and feature interactions. Linear Regression is unsuitable for this dataset due to its simplicity and inability to capture complex patterns. We recommend deploying XGBoost for predicting Lifespan.

3.3 Critical Review

Strengths

- XGBoost effectively handled the dataset's complexity, delivering high accuracy.
- The preprocessing routine ensured clean, scaled, and encoded data, which contributed to model success.
- Rigorous hyperparameter tuning optimized the models for better performance.

Areas for Improvement

- Linear Regression could have been enhanced with feature engineering or polynomial terms to capture non-linear trends.
- The dataset's limited size may have constrained model generalization; cross-validation could be extended to improve robustness.

Alternative Approaches

1. Untried Models:
   - Neural Networks: Could capture non-linear patterns but may require more data to perform effectively.
   - Random Forest Regressor: Less prone to overfitting than XGBoost and may provide similar results with simpler tuning.
2. Alternate Preprocessing:
   - Polynomial Features: Expanding feature space to capture non-linearities.
   - Dimensionality Reduction (e.g., PCA): Reducing noise and improving model efficiency.
3. Tuning Scheme:
   - Bayesian Optimization: A more efficient tuning method for high-dimensional hyperparameter spaces.

4. Classification Implementation

4.1 Feature Crafting

We chose binary classification instead of multi-class classification due to several reasons:

- Simpler Decision Boundary: The client's primary objective is to determine whether a part is defective (lifetime < 1500 hours) or not defective (lifetime ≥ 1500 hours). This binary threshold aligns with operational requirements.
- K-Means Clustering Results:
    - The Elbow Method (Figure 4.1) suggested an optimal k=4, while the Silhouette Score (Figure 4.2) peaked at k=6. However, visual inspection with k=6 (Figure 4.3) of clusters showed significant overlap between groups, reducing the effectiveness of multi-class classification.

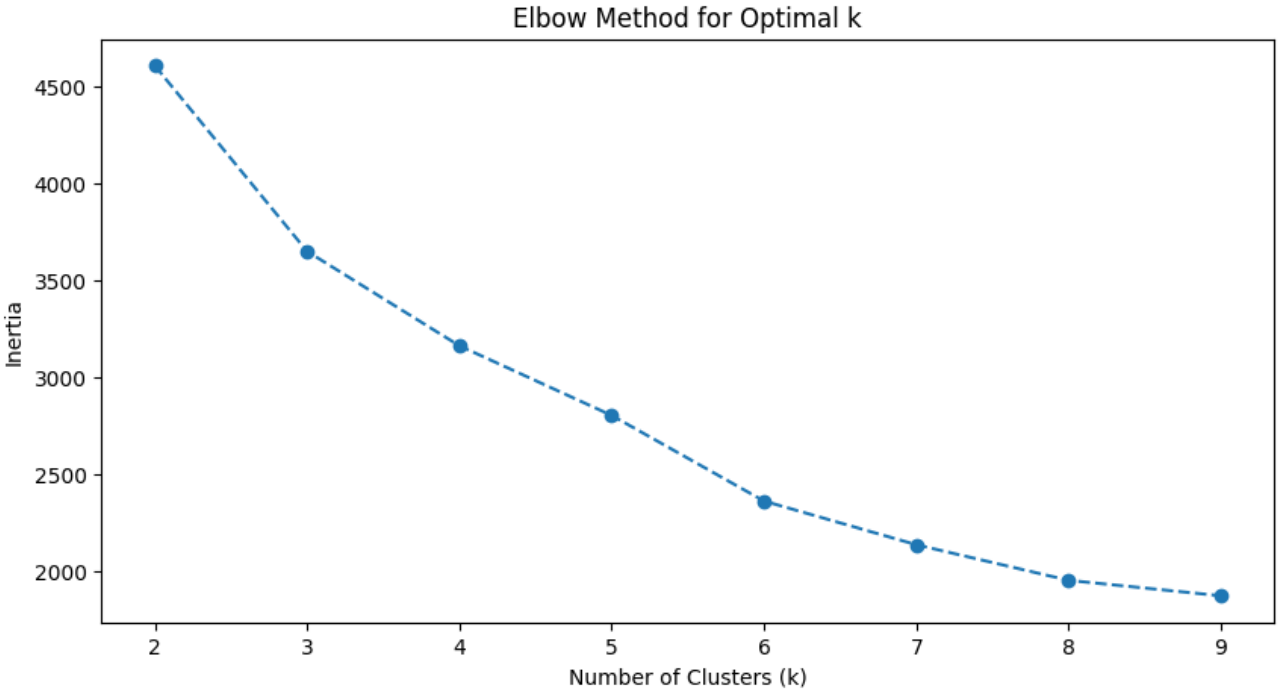

Figure 4.1: Elbow Method plot to identify the optimal number of clusters (k) for k-means clustering.

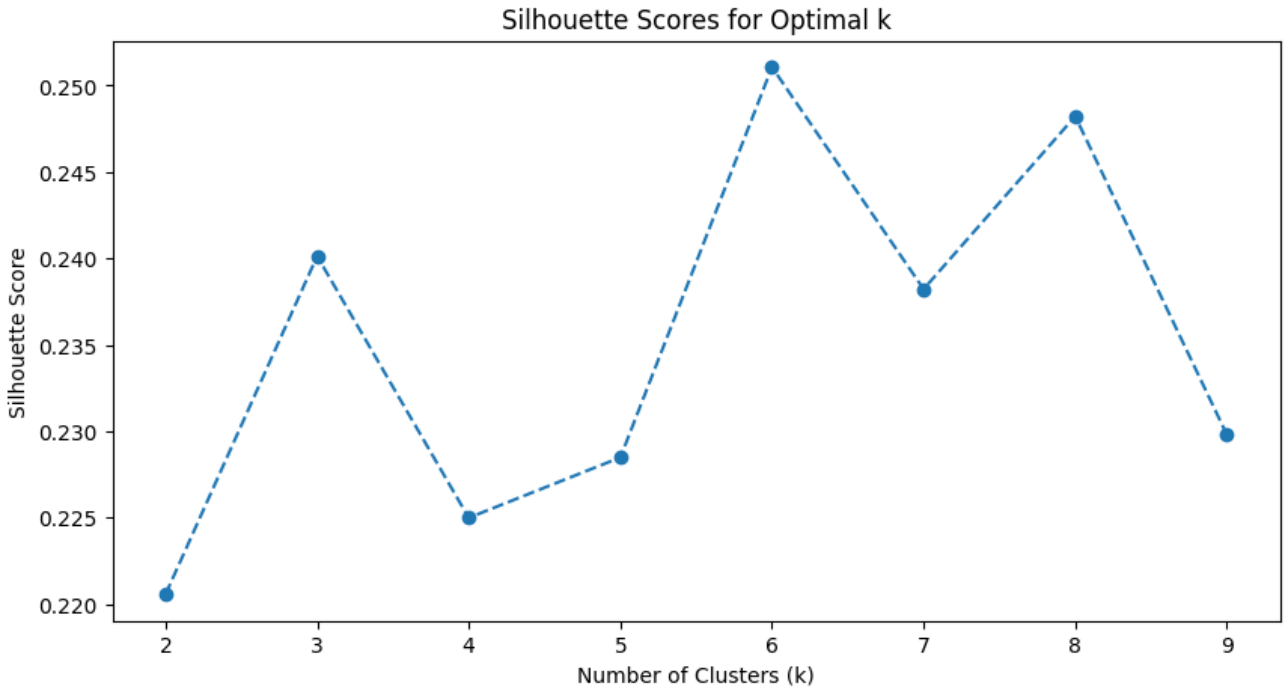

Figure 4.2: Silhouette scores for different k values, indicating clustering quality, with a peak observed at k=6.

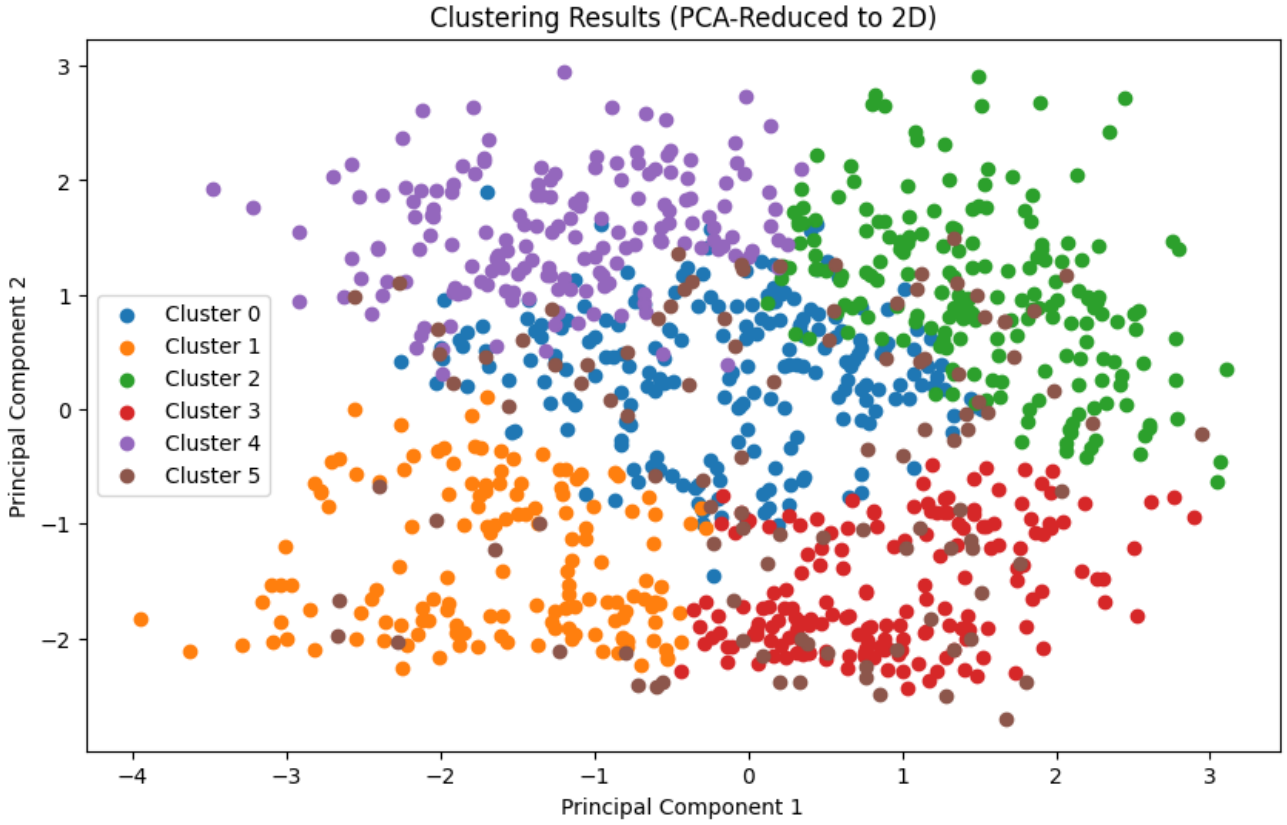

Figure 4.3: PCA-reduced 2D scatter plot of k-means clusters, visualizing the separation between groups.

    - Binary classification simplifies the task without sacrificing actionable insights.
- Balanced Dataset: After grouping, the binary target variable (is_not_defective) resulted in the following distribution:
    - Non-defective (1): 69.5%

- Defective (0): 30.5% This distribution is imbalanced but manageable using techniques like SMOTE for class balancing.

Target Class	Count	Percentage
Non-Defective	695	69.5%
Defective	305	30.5%

The chosen approach provides clarity and supports the client's requirements while maintaining dataset balance.

4.2 Methodology

Model Choices:

1. Random Forest Classifier:
   - Robust to non-linear feature interactions and imbalanced data due to its ability to handle categorical and numerical features.
   - Provides feature importance metrics, aiding interpretability.
2. Logistic Regression:
   - A baseline model with interpretable results and strong theoretical foundations.
   - Regularized versions (e.g., L1) reduce overfitting and improve performance on imbalanced datasets.

These models complement each other, offering both interpretability and performance.

Preprocessing Routine:

- Scaling: Standardized numerical features using StandardScaler to ensure consistency and improve convergence in Logistic Regression.
- Encoding: Applied OneHotEncoder to convert categorical variables into dummy variables.
- Balancing: Used SMOTE (Figure 4.4) to oversample the minority class (defective parts), improving the models' ability to predict rare events.

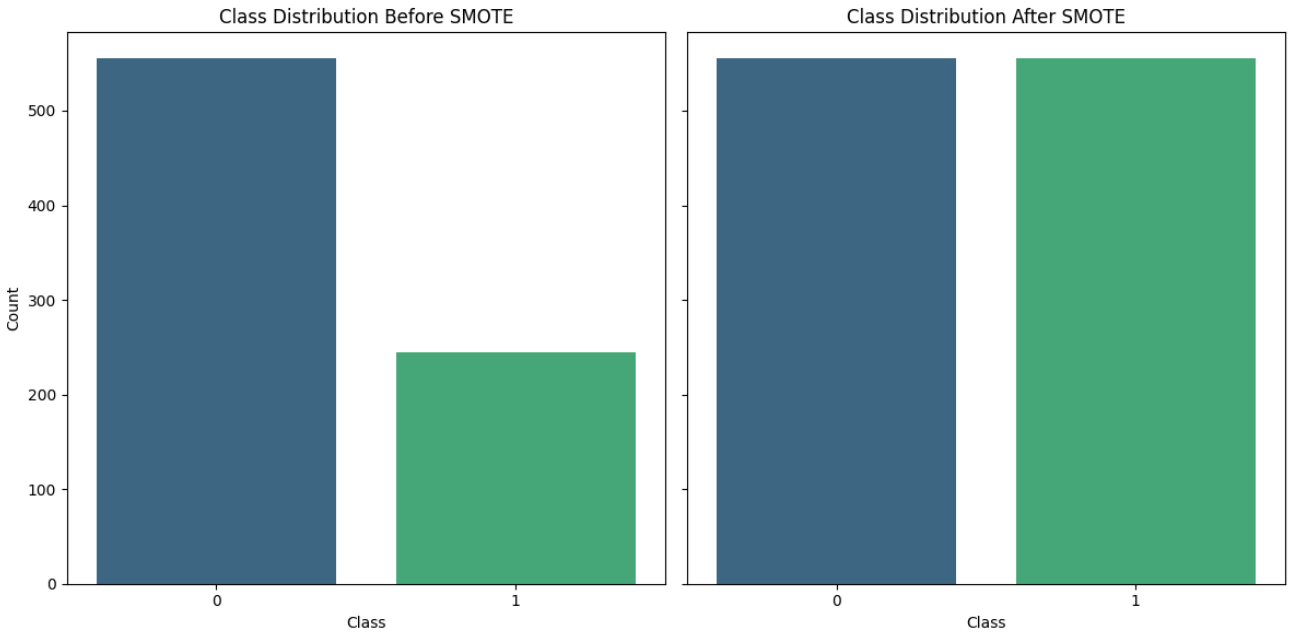

Figure 4.4: Class distribution before and after SMOTE, showing effective balancing of the minority class (Defective: 0) and majority class (Non-Defective: 1).

- Splitting: Data was split into training (80%) and testing (20%) sets, stratified to preserve the class distribution.

Hyperparameter Tuning Framework:

- Random Forest:
  - Tuned parameters such as n_estimators, max_depth, and class_weight to balance bias-variance trade-off and handle class imbalance.
- Logistic Regression:
  - Optimized C (regularization strength), penalty (L1 or L2), and solver for improved generalization.
- Tuning Approach:
  - Applied RandomizedSearchCV for broader exploration of hyperparameter space, followed by GridSearchCV for fine-tuning.
  - Focused on F1-Score to balance precision and recall, especially for the minority class.

4.3 Evaluation

Experiment Summary: Baseline models were trained and evaluated. Hyperparameter tuning significantly improved performance, especially for Random Forest. The results are summarized below:

Model	F1-Score (Baseline)	F1-Score (Tuned)	Key Hyperparameters
Random Forest	0.81	0.84	n_estimators=200, max_depth=20, class_weight=' balanced_subsample'
Logistic Regression	0.50	0.52	C=0.1, penalty='l1', solver='liblinear'

The confusion matrices (Figures 4.2 and 4.3) reveal that the Random Forest model performs significantly better than Logistic Regression in predicting the minority class, with fewer false negatives for defective parts.

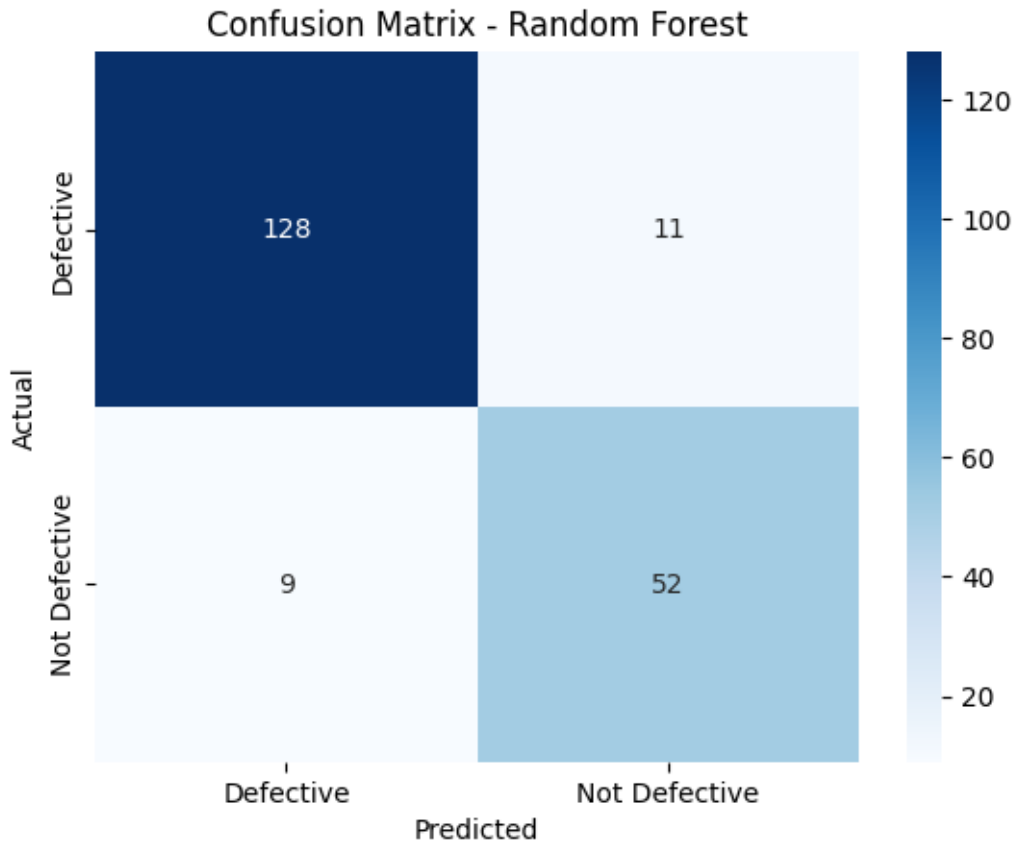

Figure 4.5: Confusion matrix for the Random Forest Classifier, highlighting its performance in predicting defective and non-defective parts.

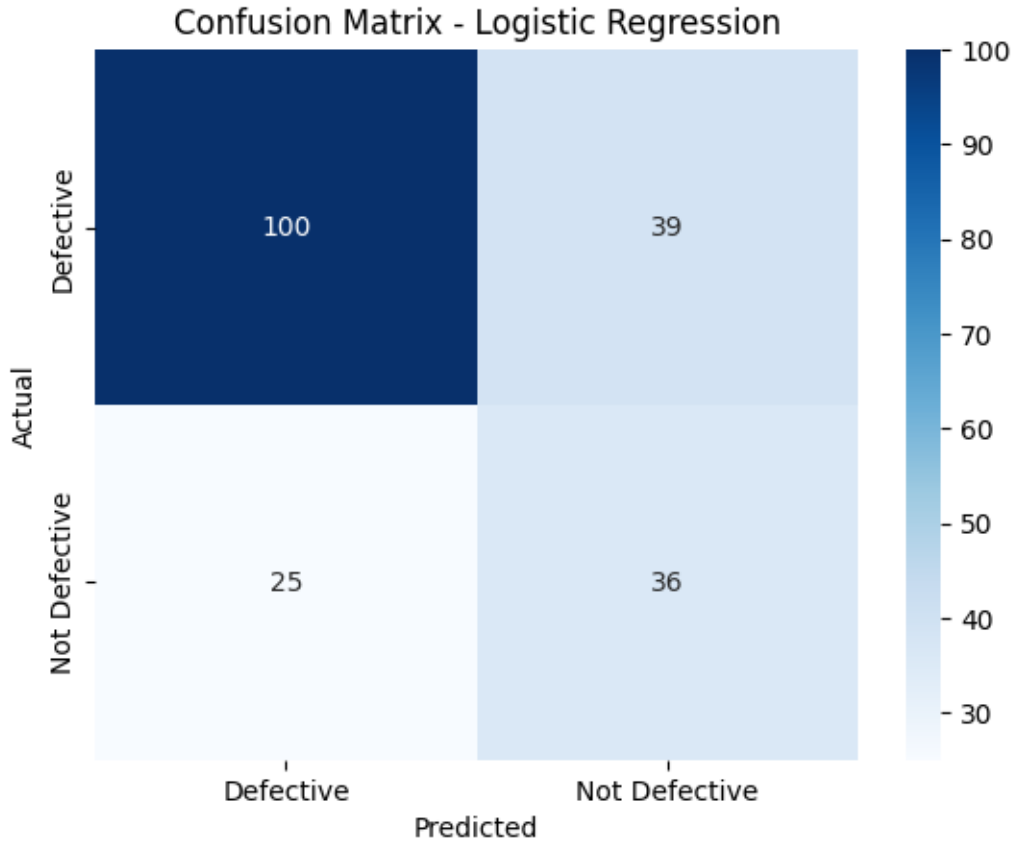

Figure 4.6: Confusion matrix for the Logistic Regression Classifier, illustrating its classification performance and challenges with the minority class.

Chosen Metrics:

- Accuracy: Measures overall performance.
- F1-Score: Prioritized due to class imbalance, balancing precision and recall.
- Confusion Matrix: Analyzed true positives and negatives versus misclassifications.

Final Model Performance:

- Random Forest:
  - Accuracy: 0.90
  - F1-Score: 0.84 (significant improvement post-tuning).
  - Precision and recall improvements, particularly in identifying defective parts
- Logistic Regression:
  - Accuracy: 0.68
  - F1-Score: 0.52.
  - Struggles with non-linear relationships and imbalanced classes.

4.4 Critical Review

Strengths:

- Random Forest: Outstanding performance, balancing precision and recall, and offering robust handling of class imbalance.
- Logistic Regression: Provides interpretable coefficients and works well as a baseline.

Areas of Improvement:

- Logistic Regression could be enhanced by feature engineering or adding interaction terms.
- More advanced class balancing techniques could be explored, such as cost-sensitive learning.

Alternative Approaches:

- Untried Models:
  - XGBoost or LightGBM: Gradient boosting models are known for superior performance on imbalanced data.
  - SVM with RBF Kernel: Effective for non-linear decision boundaries.
- Alternate Preprocessing:
  - Feature selection or dimensionality reduction (e.g., PCA) to reduce noise.
- Different Tuning Framework:
  - Bayesian Optimization for more efficient hyperparameter search.

5. Conclusions

The experiments conducted in this study demonstrated that XGBoost is the most effective model for predicting the lifespan of metal parts, achieving an R^2 of 0.96 and excelling in capturing non-linear relationships and feature interactions. On the other hand, Random Forest emerged as the superior model for binary classification, achieving an accuracy of 90% and an F1-score of 0.84, effectively identifying defective parts despite class imbalance.

The data exploration phase highlighted significant relationships between features like Nickel% and Iron% with lifespan, which were key drivers of the models' success. Both models relied on feature crafting, such as balancing the dataset using SMOTE for classification, which significantly improved Random Forest's ability to predict the minority class.

For the company's primary task of predicting whether a part is usable or defective, the classification model (Random Forest) is recommended for deployment. While the regression model provides detailed lifespan predictions, its practical utility in deciding part usability is less direct. The Random Forest classifier offers a clear binary decision aligned with the company’s operational needs, allowing immediate action to improve production efficiency and reduce defects.

This recommendation balances model accuracy, simplicity, and business applicability, ensuring the model delivers actionable insights tailored to the company's objectives.

6. References

- Chen, T. and Guestrin, C. (2016). XGBoost: A Scalable Tree Boosting System. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. [Online] Available at: https://arxiv.org/abs/1603.02754 [Accessed:19/11/2024]