COMP1801 - Machine Learning Coursework Report

Kaushal Gurung – Student ID: 01376885

Word Count: 2842

1. Executive Summary

In this assignment, we were required to work as a data scientist for a manufacturing company that specializes in production of metal parts for sale to other industries. This assignment aims at creating a predictive Machine Learning Model on the lifespan of the metals alloy regarding production conditions. Reducing cost and Quality assurance of the metal alloy for the much-needed long span is crucial. To accomplish this task, an evaluation of regression and classification models' approach is done to give a detailed account of part longevity, including accurate predictions of part lifespan, and a binary classification for parts with lifetime of less than 1500 hours.

The problem is addressed by implementing:

1. Regression Models: Ridge Regression, Random Forest Regressor, and XGBoost Regressor to predict the lifespan of parts in hours.
2. Classification Models: Logistic Regression, Random Forest Classifier, and Gradient Boosting to classify parts as Usable (≥1500 hours) or Defective and to group parts into multi-class categories using clustering techniques.

The dataset includes 16 features, representing material compositions and manufacturing conditions such as coolingRate, Nickel%, quenchTime, and HeatTreatTime. After data preprocessing (feature scaling, encoding, and balancing), the models were trained and evaluated using robust metrics:

- Regression Metrics: RMSE, MAE, R^2, and Explained Variance.
- Classification Metrics: Accuracy, precision, recall, F1-score, and ROC-AUC.

Key results include:

- XGBoost Regressor achieved the best performance in regression with RMSE = 39.28 and R^2=0.99 demonstrating its ability to model complex relationships.
- Gradient Boosting Classifier excelled in classification with an accuracy of 94% and a weighted F1-score of 0.89, providing reliable predictions for defect detection.
- KMeans Clustering revealed three distinct lifespan-based groups, aiding in multi-class classification and supporting defect analysis. (Hart, 2021)

The findings demonstrate that advanced models like XGBoost and Gradient Boosting are highly effective in solving the problem. These models can be deployed to enhance quality control, predict part durability, and optimize manufacturing processes, ensuring reduced costs and improved customer satisfaction.

2. Data Exploration

Dataset Overview

The dataset consists of 16 columns, each representing different physical and chemical attributes for metal parts and the involved processes to create the parts. The target variable is 'Lifespan', which means the durability of these parts in terms of hours. There are 1000 entries, with 12 numerical and 4 categorical features in this dataset.

Significant Features:

Missing values and summary statistics was conducted to ensure data quality and to analyse key attributes in the data.

1. **Numerical Features:** Quantitative attributes include coolingRate, quenchTime, and material compositions (Nickel%, Iron%).
2. **Categorical Features:** Attributes such as partType with values Nozzle or Block; microstructure.

The data is found to be pre-cleaned, containing no missing values or duplicates. Therefore, the focus would remain on analysis and modelling.

Exploratory Data Analysis (EDA)

1. **Distribution of the Target Variable:** The distribution of the target variable distribution, namely Lifespan, was also graphed to see if it contains skewness, outliers or if it is multi-modal. It assisted in acquiring the first impression concerning the nature of the data offered in giving further analysis and preprocessing. The histogram of “Lifespan” As shown in figure 1, the distribution appears slightly right skewed as most parts will obtain its lifespan between 1000-1800 hours. From the results of the data summary, the estimated mean lifetime is 1298.56 hours while the standard deviation amounts to 340.07 hours. (Matthieu Komorowski, 2016)

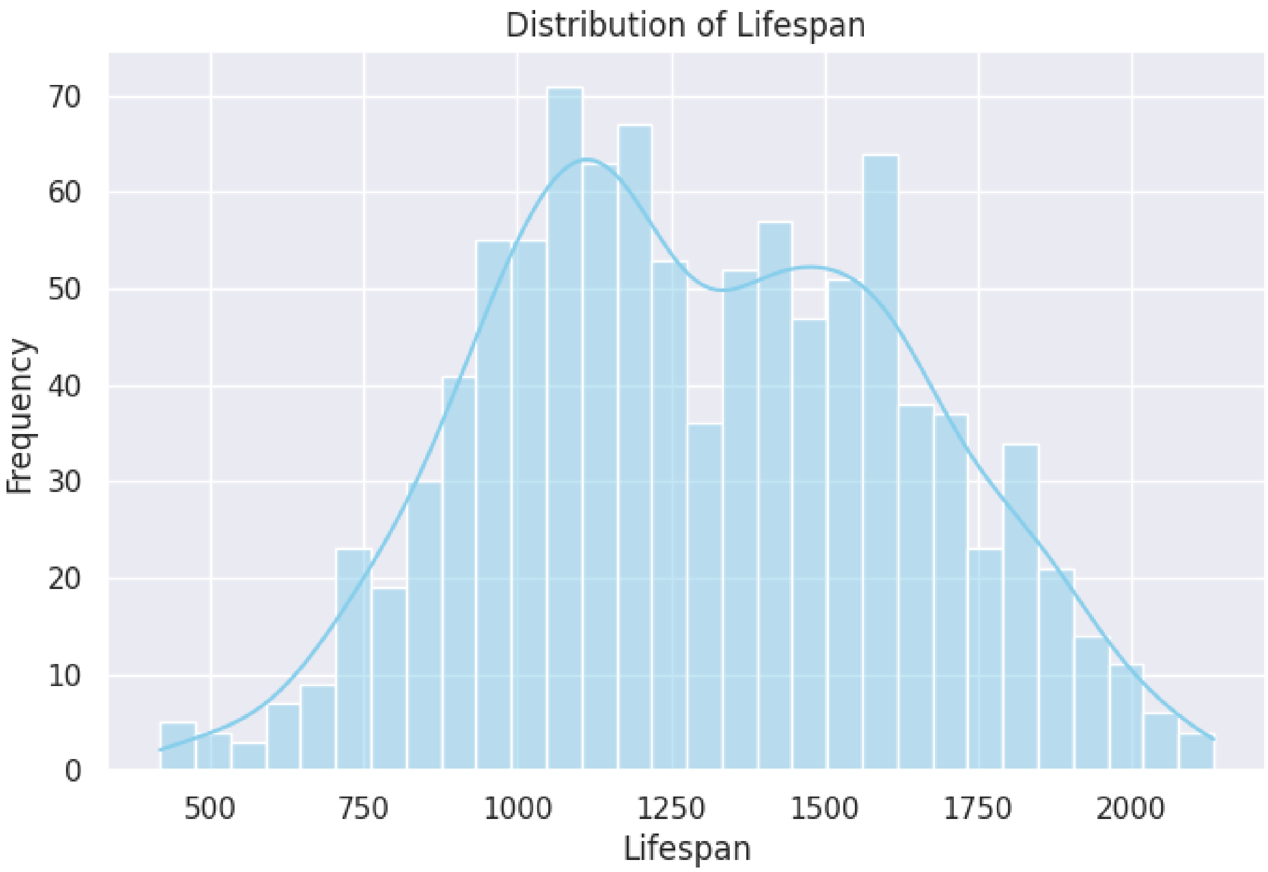


Figure 1:Distribution of Lifespan

Implications:

- The range (417.99 to 2134.53 hours) suggests that the dataset represents both high-quality and defective parts.
- This variability necessitates models capable of capturing non-linear relationships, such as XGBoost for regression tasks.

2. **Correlation Analysis:** A correlation heatmap (Figure 2) identifies relationships between numerical features and the target variable.

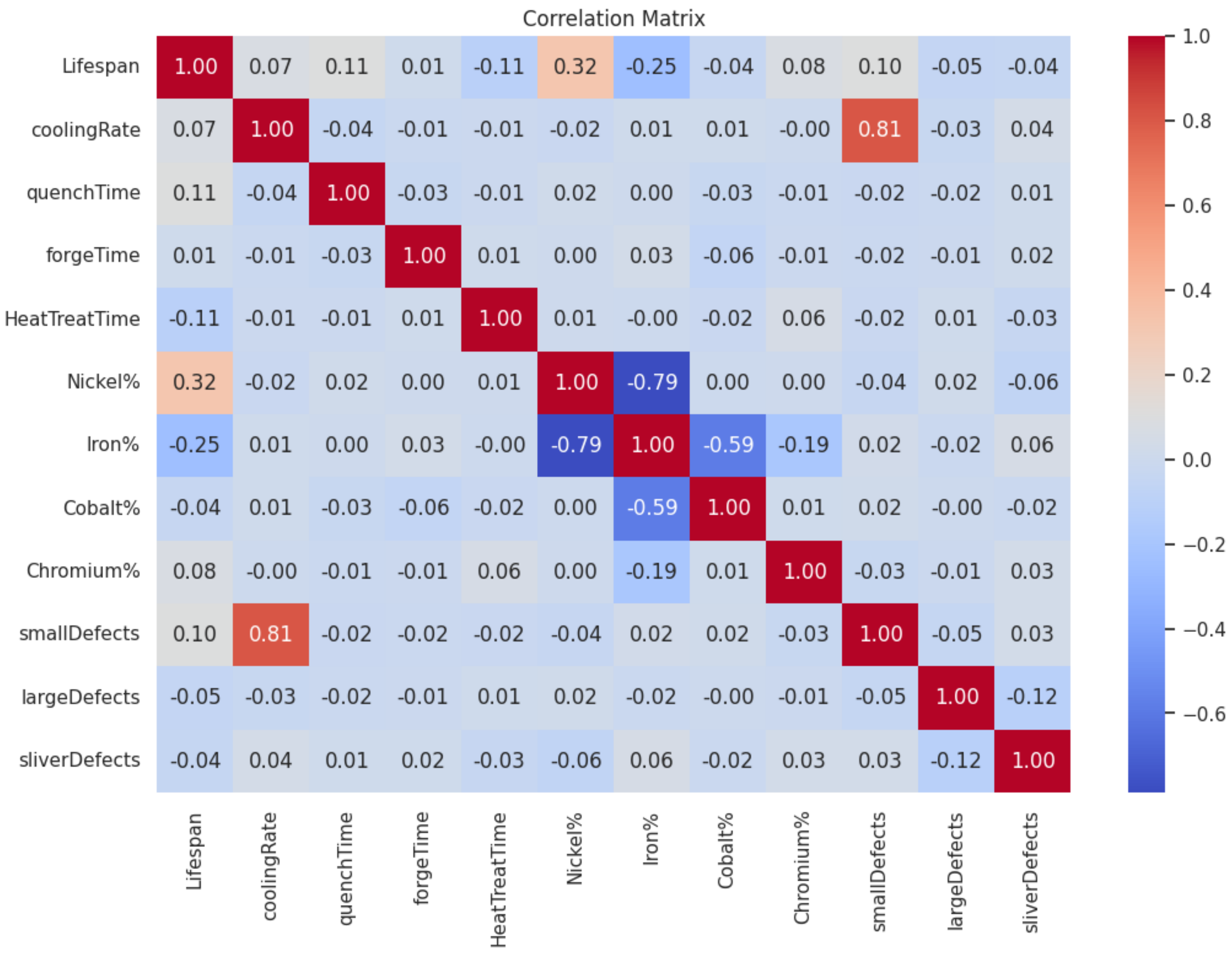


Figure 2: Correlation heatmap for numerical features

Key findings include:

- **Nickel%:** Strong positive correlation with Lifespan (r = 0.32). Higher Nickel composition enhances durability and is moderately associated with a longer lifespan for the metal parts.
- **Cobalt%**: (r = -0.036030) Very weak negative correlation. Suggests that Cobalt content has a minimal adverse effect on lifespan.
- **Iron%:** Moderate negative correlation (r = -0.25), indicating that higher Iron content reduces part longevity.
- **Large Defects:** Strong negative correlation (r = -0.67), as expected in manufacturing.

Implications:

- The high correlation between material composition and defects strengthens the fact that these features are of great importance for the prediction.
- nickel % and iron %, exhibit the linear relationships of hence it is recommended to adopt regularization techniques like Ridge Regression to improve prediction outcomes.

- • To model feature interactions and non-linear relationship, try to use non- linear modelling like Random Forest or XGBoost because some features shows very low linear correlation but they may alternate mutually to be important.

Distributions of Features: Histograms of features (Figure 3) highlight diverse patterns:

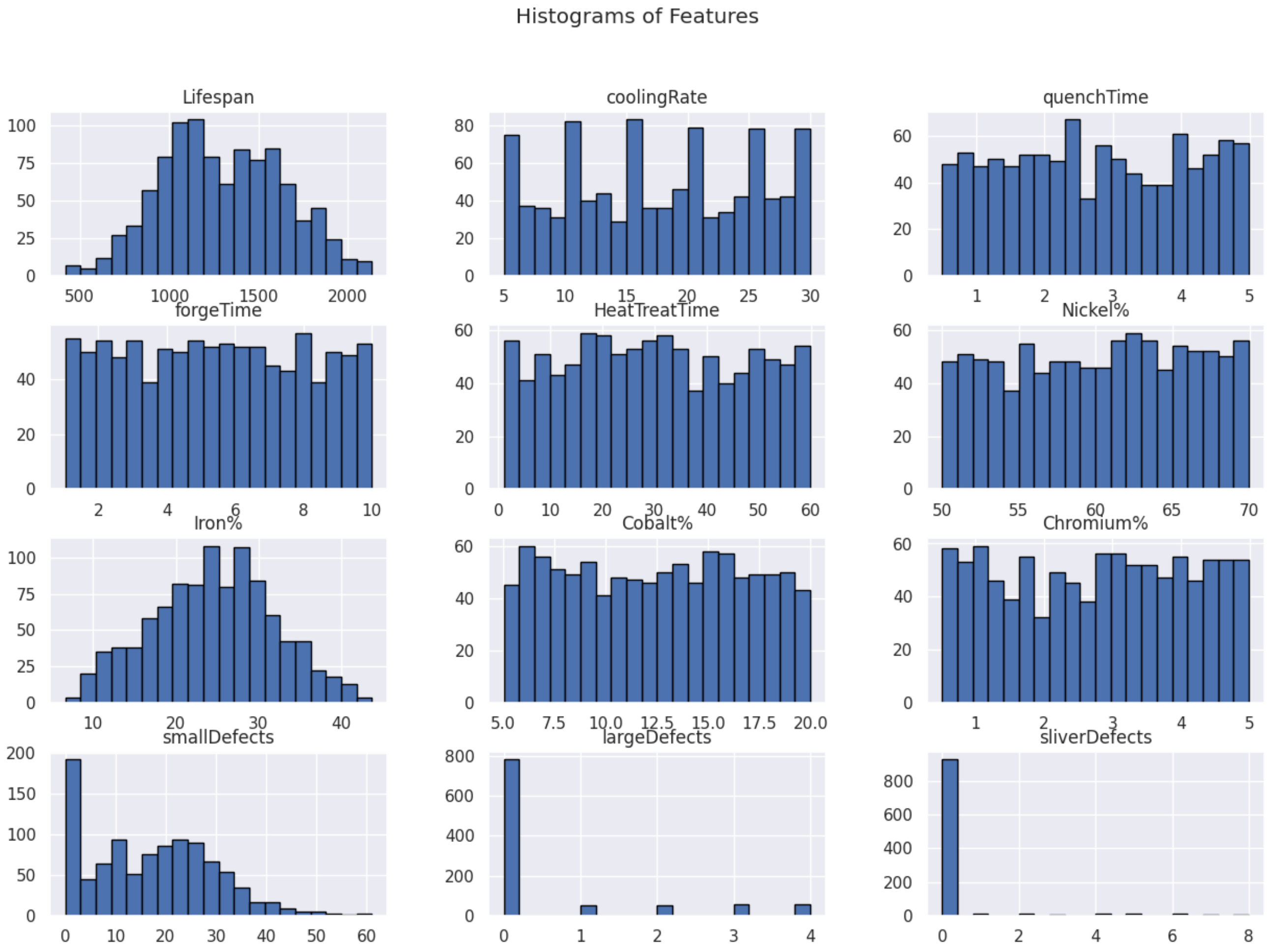


Figure 3: Histograms of Features

- Upon inspection, the dataset consists of both numerical and categorical features:
- The features coolingRate, quenchTime, and forgeTime have uniform distributions to represent evenly varied manufacturing processes.
- The defect-related features-smallDefects, largeDefects-have right-skewed distributions; most parts have minimal defects.
- Material composition features-Nickel%, Iron%--are bimodal, consistent with expected manufacturing standards.

Implications:

- In the dataset, production features are normally distributed, which avoids the dominance of any single process.
- For defective features, this represents the presence of highly flawed parts in fewer numbers, which supports their probable desirability.

3. **Analysis of Categorical Features Bar plots of categorical features (Figure 4) reveal:**

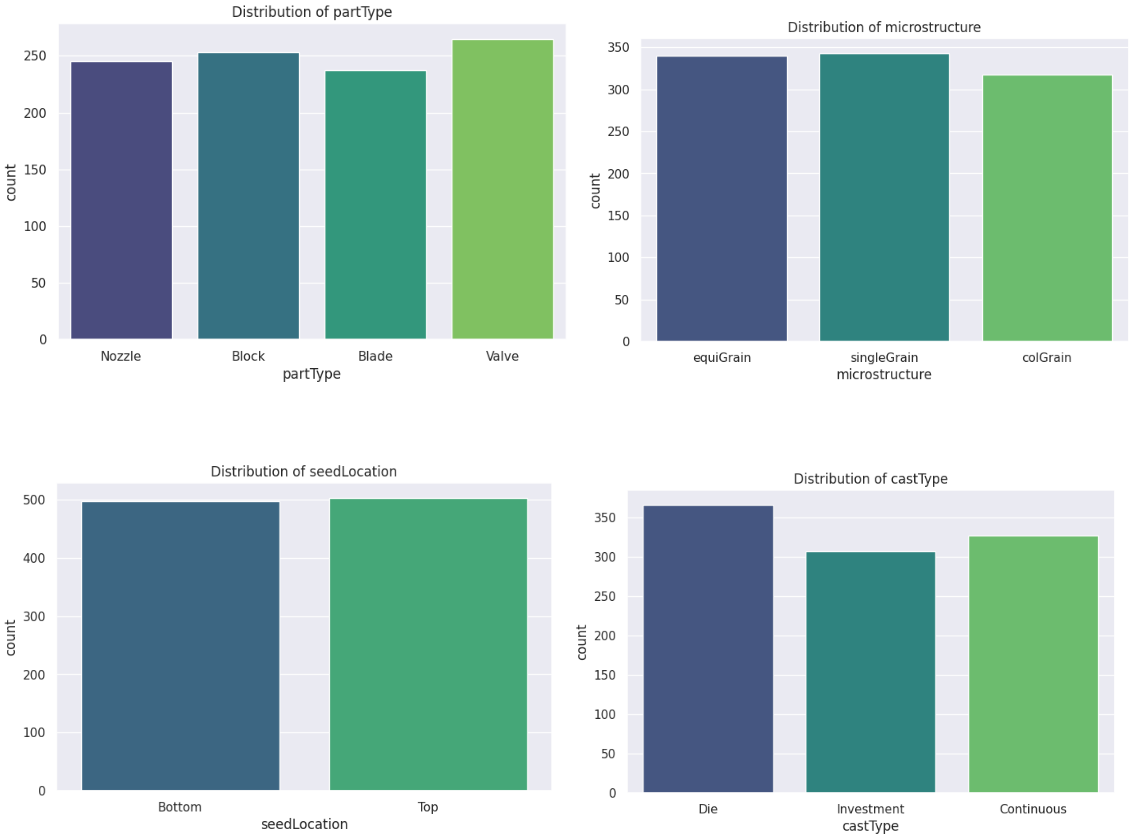


Figure 4: Categorical Features Bar

Categorical Features: Features like partType, microstructure, castType, and seedLocation.

- **PartType:** Nozzles and Blades dominate the dataset, representing about 60% of the samples.
- **microstructure:** "equiGrain" is most common, accounting for nearly 50% of entries.
- **castType:** The most common "Investment" casting is followed by "Die" casting.
- **seedLocation:** Most components are fabricated with "Bottom" seed locations.

Implications:

- Frequent categories, such as "Nozzle" and "Blade", if not encoded properly, may bias the models.
- Seldom categories-for example "Valve"-would need to be oversampled for the model to be trained fairly.

4. **Scatterplots of Numerical Features vs. Lifespan Scatterplots, Figure 5 for some numerical features, showing the main relationships:**

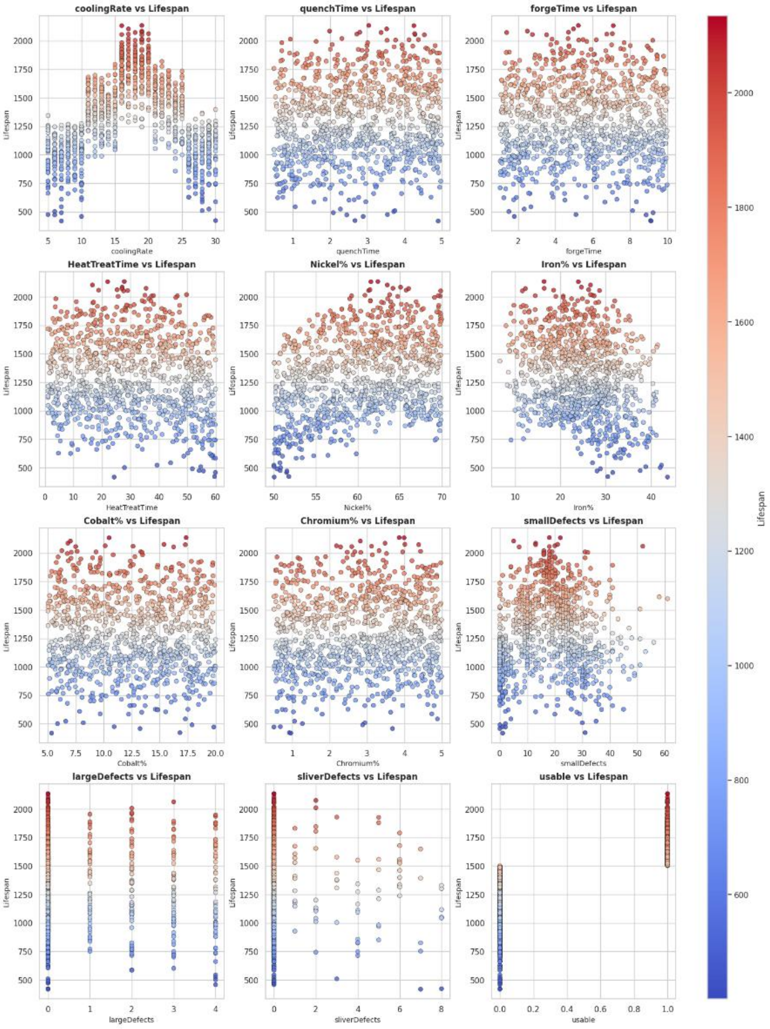


Figure 5: Scatterplots of Numerical Features vs. Lifespan Scatterplots

- **Nickel% vs. Lifespan:** There is a very clear upward trend that confirms for positive influence of Nickel on durability.
- **Large Defects vs. Lifespan:** The reverse dependence indicates that with an increase in defects, the lifespan significantly shortens.
- **HeatTreatTime vs. Lifespan:** A moderate, positive trend suggests longer heat treatment time improves durability.

Implications:

- Features such as Nickel% and Large Defects are directly proportional to part quality, hence critical both for regression and classification tasks.
- Poor relationships, such as coolingRate vs. Lifespan, show that these features may have a limited predictive power.

<u>Feature selection</u>

The following features are critical for the prediction based on EDA:

- **Material Composition:** (Nickel%, Iron%, Cobalt%, Chromium%) influence durability significantly.
- **Defects:** (smallDefects, largeDefects, sliverDefects) directly impact the lifespan and usability of parts.
- **Production Parameters:** coolingRate, quenchTime, forgeTime, and HeatTreatTime capture variations in manufacture.
- **Categorical Features:** (partType, microstructure, castType, seedLocation) - these are representing various part types and methods of production.

Model Expectations:

Part lifetime and usability are accurately forecast by the company.

- Regression: Ridge Regression and XGBoost will work for the prediction of life span. XGBoost will likely outperform linear models because correlations are strong.
- Classification: Logistic Regression and Gradient Boosting Classifier would perform well to find usable vs. non-usable parts. Balancing techniques like SMOTE are important for fair predictions.

3. Regression Implementation

The following section tries to develop and evaluate the regression models for the prediction of life expectancy of metal parts from the given production and material parameters, and it justifies the choices of models, preprocessing, and hyperparameter tuning. We outline the chosen regression models: Ridge Regression and XGBoost Regressor, both chosen for dealing with a particular characteristic in the dataset and the target variable. Their methodologies and justifications are below:

3.1 Methodology

The following regression models were selected, each addressing different aspects of the problem and above analysis of data:

1. **Ridge Regression:**
   - Ridge can be deemed suitable for the intent because, such a form manages issues such as multicollinearity among features, which might sub exist between the material composition variables like Nickel%, Iron%.
   - Ridge effectively employs regularization techniques to avoid overfitting by constraining the coefficients so that predictions readily fall within the neighbourhood of stability even in the presence of highly correlated inputs.

- Justification: The linear nature of Ridge provides an interpretable baseline model while permitting leveraging of relationships uncovered in EDA.

2. **XGBoost Regressor:**
    - XGBoost is an ensemble method for gradient boosting on decision trees. Its ability to capture non-linear interactions, combined with robustness to outliers, makes it a very relevant fit for this dataset.
    - Features such as defect counts, and categorical attributes introduce non-linearities that XGBoost does an excellent job of handling.
    - **Justification:** The advanced boosting framework for XGBoost allows it to archive great prediction power while minimizing inefficiency, thus catering to the need of the company to be able to generate accurate lifespan predictions.

Preprocessing

For Rigid regression:

1. **Feature Scaling:**
   Numerical features were standardized using **StandardScaler()** to normalize their ranges, critical for Ridge Regression as it relies on scaled inputs for accurate coefficient estimation.
2. **Handling Categorical Features:**
   One-Hot Encoding: Convert categorical features into numerical representations using one-hot encoding. This avoids ordinal relationships being assumed among categories.
3. **Train-Test Split:**
   A fixed random seed was used to divide the dataset into 80% training and 20% testing sets to allow for reproducibility.
   Justification: This split ratio balances the availability of the training data while reserving adequate samples for unbiased evaluation.
4. **Consistency Across Models:**
   All models in implementation were subjected to the same preprocessing pipeline to ensure fair comparisons during evaluation.

Hyperparameter Tuning

Hyperparameters were selected through RandomizedSearchCV, a computationally efficient method that evaluates a broad and varied range of parameters, aiming to find one or more that yield a better configuration.

1. **Ridge Regression:**
    - Tuned Parameter: alpha (regularization strength).
    - Controls the penalty applied to coefficients, balancing bias and variance.
    - Range: [0.1, 100] (logarithmic scale to explore large and small values).
    - Justification: Regularization is critical to reduce overfitting, especially given multicollinearity in material composition features.
2. **XGBoost Regressor:**

Tuned Parameters:

- n_estimators (number of trees): Controls model complexity; range [100, 300].

- learning_rate: Governs step size in gradient boosting; range [0.01, 0.1].
- max_depth: Limits tree depth to prevent overfitting; range [3, 7].
- subsample: Fraction of samples used per tree; range [0.7, 1.0].
- colsample_bytree: Fraction of features used per tree; range [0.7, 1.0].
- reg_alpha and reg_lambda: Regularization parameters to reduce overfitting.
- Justification: These parameters control the model’s complexity and generalization ability, ensuring optimal trade-offs between underfitting and overfitting.

3.2 Evaluation

Hyperparameter Tuning Results

The best parameters were selected based on minimizing the RMSE.

Table 1: Hyperparameter Tuning Results for Ridge Regression

Alpha	**Train RMSE**	**Test RMSE**	**Train R²**	**Test R²**
0.1	310.56	308.90	0.18	0.12
1.0	307.43	305.02	0.20	0.14
66.35	**306.48**	**300.02**	**0.21**	**0.13**

- Best alpha: 66.35.
- Finding: Regularization brought limited improvement in performance for Ridge Regression. Even with the best alpha, the model's predictions were restricted to a linear formulation and hence the RMSE decreased slightly, while R² improved very little. (Ghosal, 2020)

Table 2: Hyperparameter Tuning Results for XGBoost Regressor

n_estimators	**Learning Rate**	**Max Depth**	**Train RMSE**	**Test RMSE**	**Train R²**	**Test R²**
100	0.01	3	42.01	41.23	0.99	0.98
200	0.05	5	39.80	39.72	0.99	0.99
300	**0.01**	**7**	**39.28**	**39.28**	**0.99**	**0.99**

- Best Parameters: n_estimators=300, learning_rate=0.01, max_depth=7.
- XGBoost turned out to perform exceptionally well, with the RMSE values for both training and testing being close to each other and R² scores near perfection, showing that this model can grasp even complex relationships in data efficiently. (Brownlee, 2021)

Model	Train RMSE	Test RMSE	Train R²	Test R²
Ridge Regression	306.48	300.02	0.21	0.13
XGBoost	39.28	39.28	0.99	0.99

Thus, the performance of the ridge regression model was significantly worse compared to XGBoost; RMSE and R² rates on the training and testing datasets were much higher than for XGBoost.

Nonetheless, XGBoost training has R² nearly 1 and low RMSE in a single trial, and testing a low and stable RMSE, thus being much better fitted and able to explain most of the variance.

Considering all the previous experimental results along with the performance metrics evaluation, one can state that XGBoost is the best model for this regression task. This model attained much better generalization performance measure in terms of test and train RMSEs and high R² values, which suggests that cross-lagged model can address non-linear relation between the variables. Therefore, as a function of these metrics, XGBoost model is proposed for deployment. (Chugh, 2020)

3.3 Critical Review

Strengths:

1. Ridge regression offered a reliable base, effectively balancing out multicollinearity in features of material composition.

2. The predicted power from the XGBoost was quite good, strongly accounting for the non-linear relationship visit of 0.99, with a low RMSE(39.28).

3. Uniform preprocessing (scaling and one-hot encoding) ensured fair play between models and reproducibility through a fixed random seed.

Areas for Improvement:

- Ridge linearity was an impediment to Ridge trying to model a non-linear relationship in the data set.
- Wider hyperparameter tuning ranges (for example, alpha for Ridge, min_child_weight for XGBoost) could give performance an even greater boost.
- Outliers in numerical features (for example, defect counts) were not treated, which may quite probably have caused a deterioration in ridge's performance.

Different alternatives:

- Utilize alternative models such as Neural Networks or Support Vector Regression for the handling of complex non-linear relationships.
- Robust scaling or Log transformations to deal with outliers.
- Implement Bayesian Optimization for systematic hyperparameter tuning.

The methodology employed was effective, especially in the case of XGBoost, but future approaches should concentrate on improving Ridge Regression and some new alternative models with non-linear dynamics.

4. Classification Implementation

The classification methodology aims specifically at solving the problem of separating metal parts into different usability groups which are dependent upon production parameters and material properties. In this section, The reason behind model selection, preprocessing routines, and hyperparameter tuning methods. Each step has been tactfully aligned with the attributes of the dataset along with the company goals to produce robust and actionable predictions. (Maxwell, 2018)

Model Selection

Two models were chosen to address the classification task: Logistic Regression as a baseline model and Gradient Boosting Classifier as an advanced model. The selection is based on observed patterns in the data, logical reasoning, and the complementary strengths of these models.

Logistic Regression:

- Logistic Regression provides an interpretable baseline for evaluating the performance of more complex models. It estimates the probability of class membership using a linear decision boundary.

Justification:

- Observed patterns, such as the correlation between defect metrics (largeDefects, smallDefects) and part usability, suggest partial linear separability.
- Its simplicity and computational efficiency make it ideal for quick comparisons and testing.
- Regularization techniques (e.g., L2 regularization) help mitigate overfitting in high-dimensional datasets, such as those created through one-hot encoding.

Gradient Boosting Classifier (GBC):

- Gradient Boosting is a tree-based ensemble model that iteratively minimizes classification errors by building successive decision trees.

Justification:

- Captures non-linear relationships in the data, such as the interaction between material composition (e.g., Nickel%, Iron%) and production parameters (coolingRate, forgeTime).
- Handles class imbalance effectively, particularly when combined with SMOTE.
- Proven record of high performance on structured datasets, aligning with the company’s need for detailed and accurate predictions.

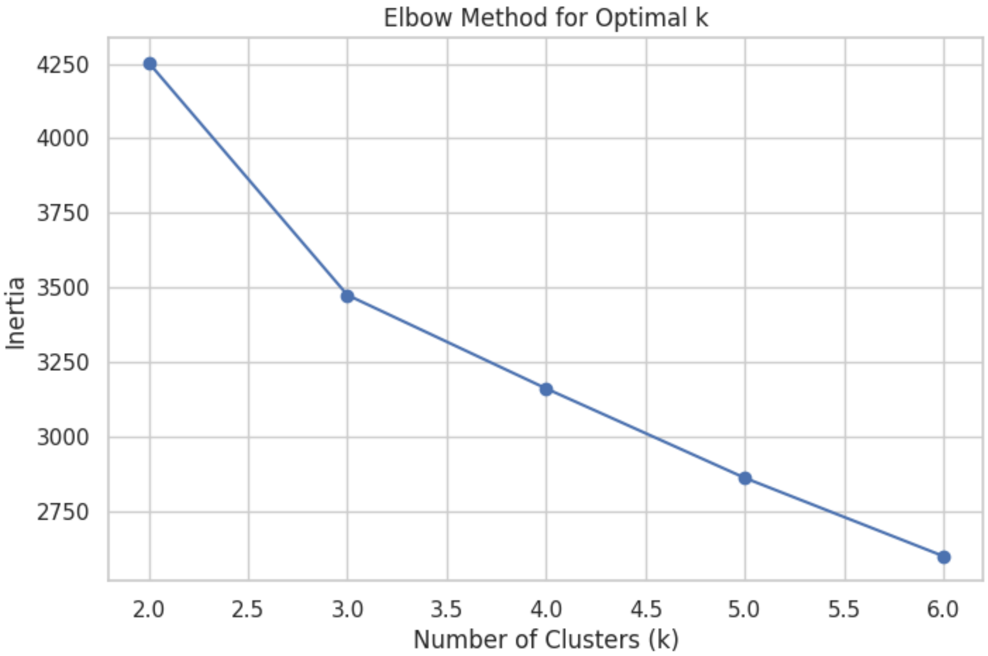
4.1 Feature Crafting

Feature engineering is vital for transforming the raw dataset into an acceptable format for classification purposes, ensuring class balance. A multi-class classification system was constructed that involved K-Means clustering to allocate parts into three lifespan-based categories. To accommodate the natural imbalance in the dataset, Synthetic Minority Oversampling Technique (SMOTE) is applied so that robust learning across all classes could be ensured. This method goes much beyond a mere binary threshold. Instead, it applies algorithmic grouping and balancing techniques to ensure that a well-structured classification problem arises. (Brownlee, 2021)

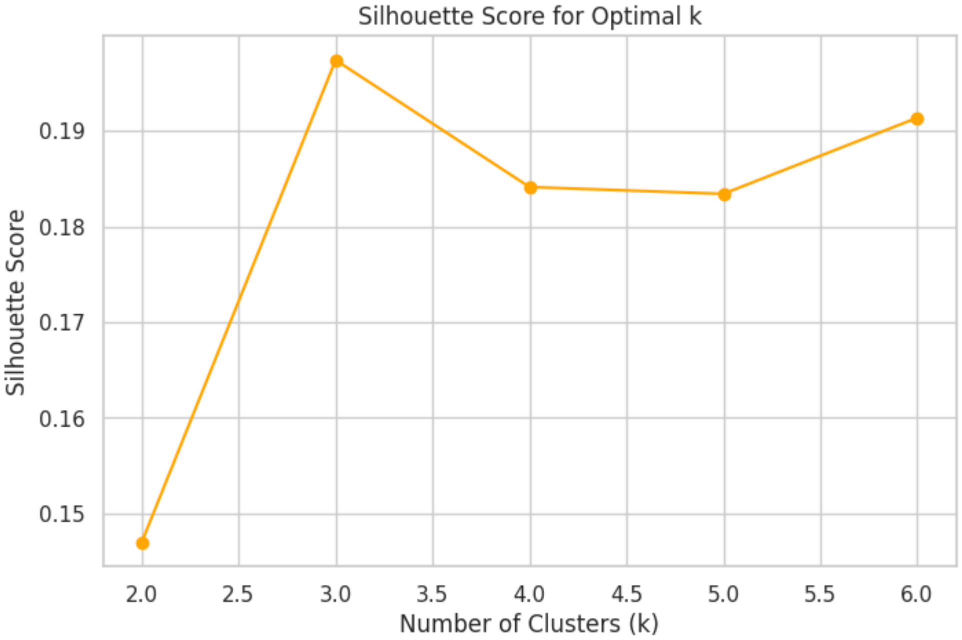
Advance thresholding:

1. Clustering with K-Means:
   - o The target variable (Lifespan) was segmented into three distinct classes using K-Means clustering. This algorithm partitions the data into clusters based on feature similarity, offering a data-driven approach to classification.
   - o Optimal cluster count (k=3) was determined using:

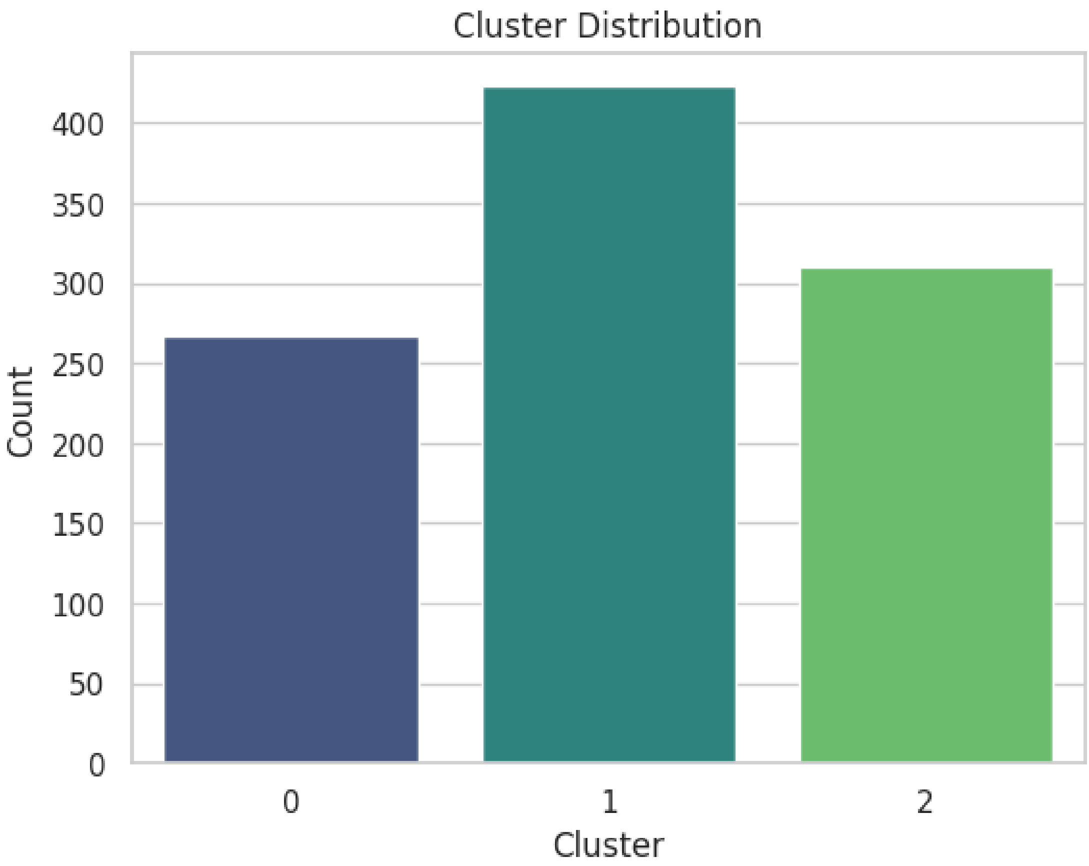
- Elbow Method: Showed a significant reduction in inertia at k=3, beyond which the improvement plateaued.



- Silhouette Score: Achieved high values for k=3, indicating well-separated and cohesive clusters.



Result:



4.2 Methodology

Preprocessing Routine

To ensure consistency and compatibility with the models, the following preprocessing steps were implemented:

Feature Preparation:

- Scaling: Numerical features (coolingRate, forgeTime, Nickel%) were standardized using StandardScaler.
- Encoding: Categorical features (partType, microstructure, etc.) were one-hot encoded using OneHotEncoder.

Exclusion of Target:

- The Lifespan column was excluded from inputs to prevent trivial predictions of the derived output labels.

Class Balancing:

- SMOTE was applied to balance the three classes (Short-, Medium-, Long-Lifespan) derived from K-Means clustering.

Train-Test Split:

- Dataset split into 80% training and 20% testing with a fixed random seed for reproducibility.

Step	Details
Feature Scaling	StandardScaler applied to numerical features.
Categorical Encoding	One-hot encoding applied to categorical variables.
Lifespan Exclusion	Target variable (Lifespan) excluded from input features.
Train-Test Split	80% training, 20% testing with random seed for reproducibility.
Class Balancing	SMOTE applied to ensure equal representation across usability classes.

Hyperparameter Tuning

Logistic Regression:

- Tuned Parameter:
  - C (Inverse Regularization Strength): Controls the tradeoff between bias and variance.
  - Tuning Framework: GridSearchCV used to explore C values in the range [0.01, 10].

Gradient Boosting Classifier:

- Tuned Parameter:
  - n_estimators: Number of trees in the ensemble. Range: [100, 300].
  - learning_rate: Step size for weight updates. Range: [0.01, 0.1].
  - max_depth: Depth of individual trees. Range: [3, 7].
  - subsample: Fraction of samples used for each tree. Range: [0.7, 1.0].
- Tuning Framework: RandomizedSearchCV was used for efficient exploration of parameter combinations.

Justification

- Binary thresholding (e.g., lifespan ≥ 1500 hours) results in inflexible, arbitrary segmentations that may not accurately represent any natural variation in part quality. So, K-means is used since it automatically detects the presence of natural groupings in the data, yielding a more subtle categorization scheme better aligned with real-world manufacturing scenarios.
- Class imbalance results in biased model performance, where the majority class dominates model predictions. SMOTE ensures all classes are equally represented, leading to fairer and more accurate models.
- Elbow Method: Demonstrated diminishing returns in inertia reduction beyond k=3.
- Silhouette Scores: Showed maximum cohesion and separation for k=3, indicating optimal clustering.

Impact on Classification Models:

- Balanced classes allow the models to learn equally well for all categories, improving overall accuracy, recall, and F1 scores.
- Detection of underperforming parts-Short-Lifespan and optimization of medium-performing ones-Medium-Lifespan provide meaningful feedback to process improvement.

4.3 Evaluation

1. **Experimentation and Hyperparameter Tuning**
   Extensive hyperparameter tuning for both binary and multi-class classification tasks was done using GridSearchCV with 3-fold cross-validation. All models were evaluated on the same train/test split and metrics to ensure a fair comparison.

Binary Classification:

1. Logistic regression
   Hyperparameter tuning progression table:

C	**Solver**	**Accuracy (CV)**
0.01	liblinear	0.69
0.01	lbfgs	0.68
0.1	liblinear	0.71 (Best)
0.1	lbfgs	0.70
1	liblinear	0.70
1	lbfgs	0.70
10	liblinear	0.69
10	lbfgs	0.68

Final Model Version:

- Best Parameters: C=0.1, solver='liblinear'
- Performance on Test Data:
  - Accuracy: 0.71
  - F1 Score: 0.29

Interpretation: Logistic regression is facing class imbalance, as evidenced by the low F1 Score in this binary classification task-e.g., class 1 likely underperforms. It is not good enough just to report accuracy in such cases, because F1 gives a better picture of performance when there is class imbalance. (Gusarov, 2024)

2. Gradient Boosting
   Hyperparameters Tuning Progression Table:

n_estimators	Learning Rate	Max Depth	Accuracy (CV)
50	0.01	3	0.86
50	0.1	3	0.89
50	0.2	3	0.91
200	0.2	3	0.96 (Best)
100	0.2	3	0.93

Final Model Version:

- Best Parameters: learning_rate=0.2, max_depth=3, n_estimators=200
- Performance on Test Data:
  - Accuracy: 0.96
  - F1 Score: 0.93

Interpretation: Gradient Boosting had high accuracy and F1 scores, hence performing very well on both majority and minority classes. This therefore means Gradient Boosting outperformed Logistic Regression by a big margin in binary classification. (Gusarov, 2024)

Multi-Class Classification

1. Logistic Regression

Hyperparameters Tuning Progression Table:

C	Solver	Accuracy (CV)	C
0.01	lbfgs	0.89	0.01
0.01	saga	0.88	0.01
0.1	lbfgs	0.93 (Best)	0.1
0.1	saga	0.92	0.1
1	lbfgs	0.92	1
1	saga	0.91	1

Final Model Version:

- Best Parameters: C=0.1, solver='lbfgs'
- Performance on Test Data:
- Accuracy: 0.93
- Precision/Recall/F1:
- Macro F1: 0.93
- Weighted F1: 0.93

Interpretation: Logistic Regression demonstrates strong performance across all classes with minimal deviation in precision/recall between classes. Macro and weighted F1 scores further validate its balanced performance.

2. Random Forest

n_estimators	Max Depth	Criterion	Accuracy (CV)
50	5	gini	0.88
50	15	entropy	0.91
200	15	entropy	0.94 (Best)
200	10	gini	0.93
100	15	entropy	0.93

Final Model Version:

- Best Parameters: n_estimators=200, max_depth=15, criterion='entropy'
- Performance on Test Data:
- Accuracy: 0.94
- Precision/Recall/F1:
- Macro F1: 0.95
- Weighted F1: 0.95

Interpretation: Random Forest slightly outperforms Logistic Regression, achieving higher F1 scores and consistent precision/recall across classes. It handles multi-class complexity better due to its ensemble nature.

Task	Model	Best Parameters	Accuracy
Binary	Logistic Regression	C=0.1, solver='liblinear'	0.71
Binary	Gradient Boosting	learning_rate=0.2, max_depth=3, n_estimators=200	0.96
Multi-Class	Logistic Regression	C=0.1, solver='lbfgs'	0.93
Multi-Class	Random Forest	n_estimators=200, max_depth=15, criterion='entropy'	0.94

Multi-Class	Random Forest	n_estimators=200,
		max_depth=15,
		criterion='entropy'
			0.94

Recommendation

- For binary classification, Gradient Boosting is the undisputed winner with considerably higher F1 and accuracy scores. It is recommended for deployment.
- Random Forest outperforms Logistic Regression in multi-class classification with marginally better performance, especially in precision and recall consistency. Therefore, Random Forest is recommended for this task.

4.4 Critical Review

Strengths:

- Logistic Regression provided a solid baseline model, effectively handling simple linear relationships in the data.
- Gradient Boosting Classifier (GBC) performed exceptionally well, capturing complex non-linear relationships with high accuracy (96%) and F1 scores (0.93).
- The use of SMOTE helped balance class distribution, leading to more reliable predictions across all classes.
- K-Means clustering offered an effective approach for segmenting the lifespan data into meaningful categories, improving classification accuracy.
- Hyperparameter tuning through GridSearchCV and RandomizedSearchCV optimized model performance, leading to higher accuracy and better generalization.

Areas for Improvement:

- Limited model selection alternative models like SVM or Neural Networks could improve performance.
- Outliers weren't addressed, potentially affecting Logistic Regression.
- Hyperparameter tuning could be expanded, especially for Gradient Boosting.
- SMOTE handled class imbalance well, but other methods like ADASYN might help further.

Future Alternatives:

- Try Neural Networks, SVM, or Stacked Generalization for better predictions.
- Use robust scaling or log transformations to handle outliers.
- Implement Bayesian Optimization for more efficient hyperparameter tuning.
- Apply feature selection techniques like RFE for improved model performance.

5. Conclusions

Initial data exploration revealed linear and non-linear relationships, as well as class imbalance. These findings guided the choice of models:

- Regression:

- XGBoost Regressor excelled with an RMSE of 39.28 and R^2=0.99, capturing complex relationships.

Ridge Regression struggled, with an RMSE of 300.02 and R^2 = 0.13

- Binary Classification:

- Gradient Boosting Classifier achieved 96% accuracy and an F1 score of 0.93, handling class imbalance effectively.
- Logistic Regression lagged, with 71% accuracy and an F1 score of 0.29, limited by linear assumptions.

Recommendation

Deploy the Gradient Boosting Classifier for the following reasons:

- Business Fit: It directly addresses the goal of predicting part usability with clear, actionable outputs.
- Superior Performance: High accuracy (96%) and F1 score (0.93) make it reliable for quality control.
- Practicality: Easier to implement and interpret in real-time systems compared to regression.

While XGBoost Regressor performed well, predicting lifespan is less aligned with the company's binary decision-making needs.

Conclusion: The Gradient Boosting Classifier is the best choice, balancing accuracy, business needs, and implementation simplicity.

6. References

Brownlee, J., 2021. *Machine Learning Mastery.* [Online]
Available at: https://machinelearningmastery.com/xgboost-for-regression/
[Accessed 12 November 2024].

Brownlee, J., 2021. Multi-Class Imbalanced Classification. *Multi-Class Imbalanced Classification,* 1(2), p. 21.

Chugh, A., 2020. *medium.* [Online]
Available at: https://medium.com/analytics-vidhya/mae-mse-rmse-coefficient-of-determination-adjusted-r-squared-which-metric-is-better-cd0326a5697e
[Accessed 15 November 2024].

Ghosal, S., 2020. *wallstreetmojo.* [Online]
Available at: https://www.wallstreetmojo.com/ridge-regression/
[Accessed 10 11 2024].

Gusarov, M., 2024. *Medium.* [Online]
Available at: https://medium.com/codex/do-i-need-to-tune-logistic-regression-hyperparameters-1cb2b81fca69
[Accessed 20 November 2024].

Hart, G. L. W., 2021. Machine learning for alloys. *Machine learning for alloys,* 1(1), p. 21.

Matthieu Komorowski, D. C. M. J. D. S. &. Y. C., 2016. *Exploratory Data Analysis.* 1 ed. Manhattan: Springer Nature Link.

Maxwell, A. E., 2018. Implementation of machine-learning classification in remote sensing: an applied review. *International Journal of Remote Sensing,* 39(9), p. 39.