Executive Summary

Problem Statement:

This project revolves around understanding the structure, patterns, and characteristics of a metal parts dataset, through data exploration, regression modeling, binary classification, convolutional neural network (CNN) implementation, and clustering analysis. It aims to unravel the factors influencing metal part lifespan, with a particular focus on part types and cooling rates

Importance:

Precise prediction regarding the lifespan and identification of defects empower manufacturers to streamline their operations, leading to decreased production expenses and mitigated instances of faulty outputs. This capability aids in the judicious allocation of resources, preventing unnecessary expenditures on parts with a heightened risk of defects or shorter lifespans. Additionally, it diminishes the dependence on manual inspection, contributing to an overall enhancement of the efficiency of the quality control process.

Machine Learning Models Used and Their Results:

Regression Implementation: Implemented Random Forest and Ridge Regression models for lifespan prediction. Random Forest model, with hyperparameter tuning, demonstrated superior accuracy and precision compared to the Ridge Regression model.Evaluation metrics such as MSE, RMSE, and R-squared highlighted the Random Forest model's effectiveness in predicting metal part lifespan.

Binary Classification Implementation: Developed binary classification models using Support Vector Classification (SVC) and Logistic Regression to categorize metal parts. The tuned SVC model showcased robust performance with a focus on correctly identifying positive instances, aligning with the critical nature of defect identification.

Convolutional Neural Network (CNN) Implementation: Employed DenseNet121 and Xception models for multi-class classification of defects in metal parts based on surface scans. DenseNet121 outperformed Xception, achieving an accuracy of 99.5% and demonstrating superiority in precision, recall, and F1-score.

Clustering Implementation: Clustering analysis focused on continuous variables, revealing optimal clustering at k=2 using KMeans. Visualization through Principal Component Analysis provided insights into the distribution of clusters in a 2D space. Box plots further illustrated the variation of features within each cluster. A comparison with binary predictions highlighted the efficacy of clustering in capturing patterns similar to binary classification.

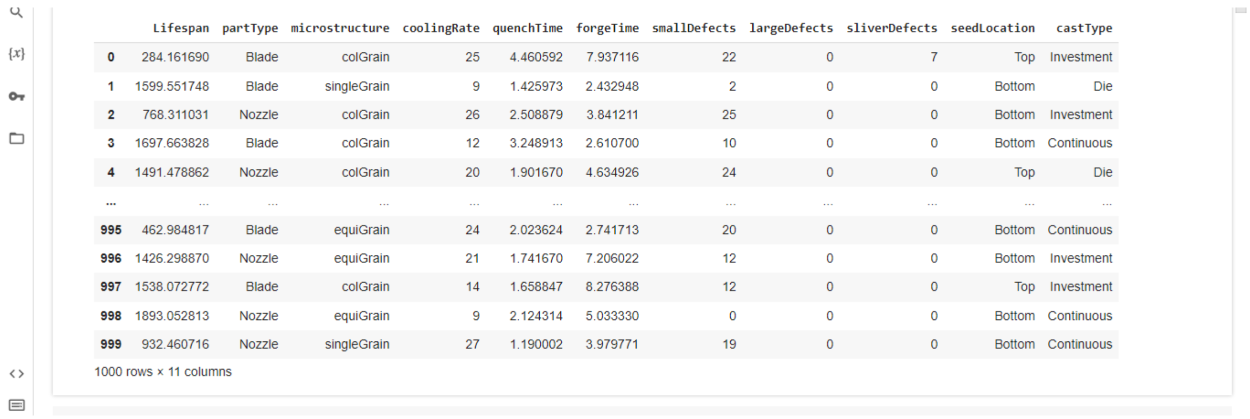
Data Exploration

Data exploration is the preliminary phase in the data analysis process where analysts and data scientists investigate, summarize, and visualize a dataset to understand its structure, patterns, and characteristics (Batch, 2017).

The initiation of this project involves importing libraries essential for numerical operations and data manipulation. In this context, the project leverages the pandas library:

Pandas is a Python library designed for data manipulation and analysis, offering an extensive array of functions and methods tailored for tasks such as data cleaning, transformation, and analysis.

The pandas library is employed to load the tabular data from the CSV file named "COMP1801_CourseworkDataset1_tabular.csv." The data is read into a DataFrame named ‘df’ using the ‘pd.read_csv()’ function.

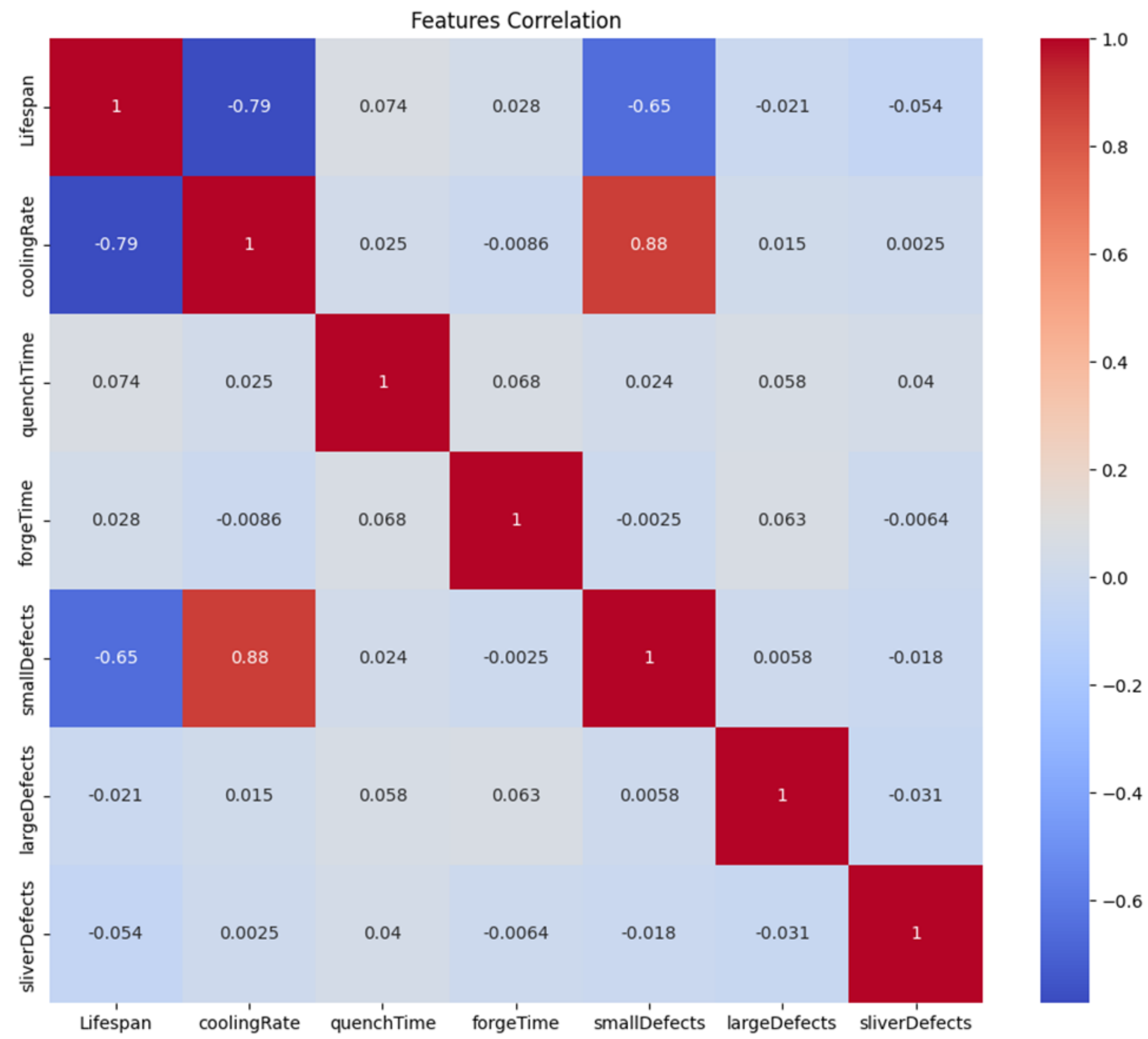


	Lifespan	partType	microstructure	coolingRate	quenchTime	forgeTime	smallDefects	largeDefects	sliverDefects	seedLocation	castType
0	284.161690	Blade	colGrain	25	4.460592	7.937116	22	0	7	Top	Investment
1	1599.551748	Blade	singleGrain	9	1.425973	2.432948	2	0	0	Bottom	Die
2	768.311031	Nozzle	colGrain	26	2.508879	3.841211	25	0	0	Bottom	Investment
3	1697.663828	Blade	colGrain	12	3.248913	2.610700	10	0	0	Bottom	Continuous
4	1491.478862	Nozzle	colGrain	20	1.901670	4.634926	24	0	0	Top	Die
...	...	...	...	...	...	...	...	...	...	...	...
995	462.984817	Blade	equiGrain	24	2.023624	2.741713	20	0	0	Bottom	Continuous
996	1426.298870	Nozzle	equiGrain	21	1.741670	7.206022	12	0	0	Bottom	Investment
997	1538.072772	Blade	colGrain	14	1.658847	8.276388	12	0	0	Top	Investment
998	1893.052813	Nozzle	equiGrain	9	2.124314	5.033330	0	0	0	Bottom	Continuous
999	932.460716	Nozzle	singleGrain	27	1.190002	3.979771	19	0	0	Bottom	Continuous

1000 rows × 11 columns

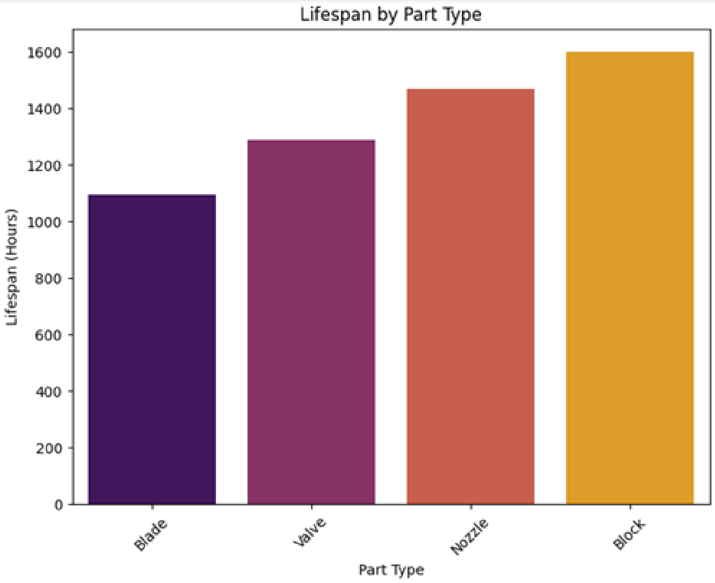
Image of CSV file after being loaded into dataframe 'df'.

A correlation heatmap is generated to visualize the relationships between numerical features. This provides insights into how different features correlate with each other. The heatmap is annotated with correlation coefficients.



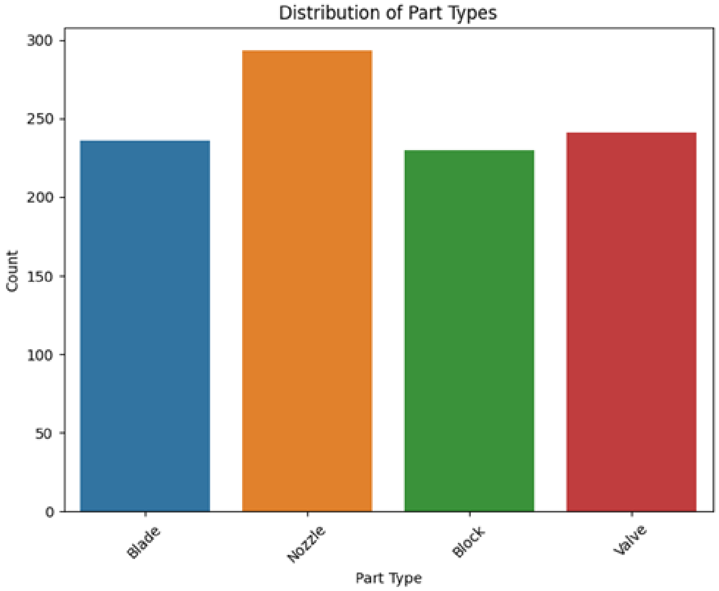
	Lifespan	coolingRate	quenchTime	forgeTime	smallDefects	largeDefects	sliverDefects
Lifespan	1	-0.79	0.074	0.028	-0.65	-0.021	-0.054
coolingRate	-0.79	1	0.025	-0.0086	0.88	0.015	0.0025
quenchTime	0.074	0.025	1	0.068	0.024	0.058	0.04
forgeTime	0.028	-0.0086	0.068	1	-0.0025	0.063	-0.0064
smallDefects	-0.65	0.88	0.024	-0.0025	1	0.0058	-0.018
largeDefects	-0.021	0.015	0.058	0.063	0.0058	1	-0.031
sliverDefects	-0.054	0.0025	0.04	-0.0064	-0.018	-0.031	1

Thereafter the ‘matplotlib’ and ‘seaborn’ libraries were imported to conduct an analysis aimed at discovering correlations within the dataset. Upon comparison of the lifespan by part types, the findings indicated that ‘block’ exhibited the longest lifespan among the part types. Following 'block', 'nozzle' displayed the next highest lifespan, succeeded by 'valve' and then 'blade'.



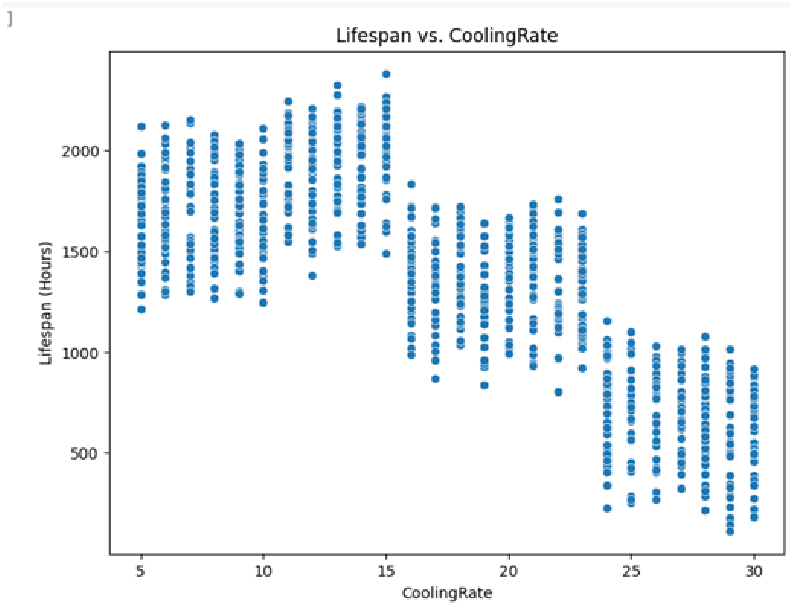
Graphical illustration of lifespan by part type.

Examining the dataset relationships through the distribution of part types, the investigation unveiled that 'nozzle' exhibited the highest prevalence among the datasets. Subsequently, in descending order of distribution, 'valve,' 'blade,' and 'block' were identified.



Graphical illustration of distribution of datasets by part type

An analysis of variance (ANOVA) F-statistic and p-value is calculated to assess the impact of cooling rate on lifespan. The low p-value suggests a significant relationship, and it is visually depicted in the scatter diagram. The illustration in the scatter diagram delineates a clear trend: higher cooling rates are associated with lower lifespans, whereas lower cooling rates are linked to higher lifespans. This observation aligns with the finding that 'block' exhibits the lengthiest lifespan among the identified part types.



Scatter Diagram

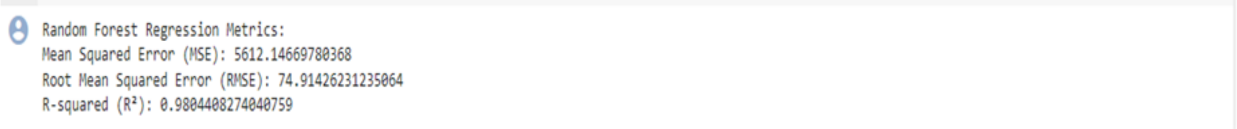
REGRESSION IMPLEMENTATION

Regression implementation is the practical application of statistical techniques, specifically regression models, to analyze and model the relationship between a dependent variable and one or more independent variables (Hocking, 2013). The two models used for analysis in this report are Random Forest and Ridge Regression Models.

Random Forest Regression Implementation

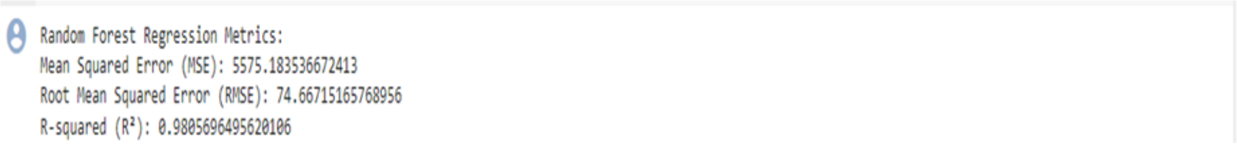
A Random Forest Regression Model is a machine learning algorithm that falls under the ensemble learning category (Kiangala, 2021). It is specifically designed for regression tasks, where the goal is to predict continuous numerical values rather than categorical outcomes.

A Random Forest Regression model consisting of 50 trees was constructed and assessed, revealing encouraging outcomes. Subsequent hyperparameter tuning effectively fine-tuned the model using optimal parameters identified through grid search. The refined model underwent evaluation, showcasing enhanced performance compared to its initial iteration.



```
Random Forest Regression Metrics:
Mean Squared Error (MSE): 5612.14669780368
Root Mean Squared Error (RMSE): 74.91426231235064
R-squared (R²): 0.9804408274040759
```

Results gotten from the Random Forest Regression Model



```
Random Forest Regression Metrics:
Mean Squared Error (MSE): 5575.183536672413
Root Mean Squared Error (RMSE): 74.66715165768956
R-squared (R²): 0.9805696495620106
```

Results gotten from the hyperParameter tuning of the Random Forest Regression Model

Ridge Regression Model

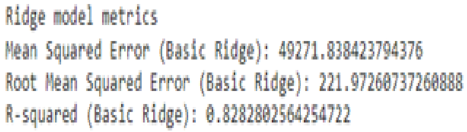
Ridge Regression Model is a linear regression technique that extends traditional linear regression by introducing a regularization term (Zhang, 2010). In Ridge Regression, the standard linear regression objective function is augmented with a penalty term that is proportional to the square of the magnitude of the coefficients.

It enhances the model's ability to generalize well to new, unseen data, contributing to more reliable predictions for the metal part lifespan.

In predicting the lifespan of metal parts, the application of the Ridge Regression model has been pivotal. The initial model, employing default hyperparameters, underwent a rigorous evaluation encompassing essential regression metrics such as Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and R-squared.

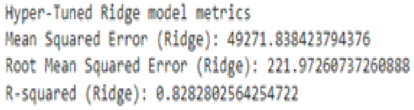
To improve the model's performance, a meticulous hyperparameter tuning process was carried out through Grid Search, exploring a range of alpha values. This systematic exploration led to the identification of the optimal hyperparameter (alpha), culminating in the development of a finely-tuned Ridge Regression model.

Upon subjecting the tuned model to evaluation using an independent test dataset, a comprehensive array of metrics was computed and meticulously examined. The outcomes, encompassing MSE, RMSE, and R-squared, provide a quantitative gauge of the model's predictive prowess. The nuanced interpretation of these metrics stands as a robust measure, shedding light on the Ridge Regression model's effectiveness in estimating the lifespan of metal parts. This empirical insight proves invaluable for practical applications and decision-making within the realm of manufacturing processes.



```
Ridge model metrics
Mean Squared Error (Basic Ridge): 49271.838423794376
Root Mean Squared Error (Basic Ridge): 221.97260737260888
R-squared (Basic Ridge): 0.8282802564254722
```

Results gotten from the implementation of ridge regression model



```
Hyper-Tuned Ridge model metrics
Mean Squared Error (Ridge): 49271.838423794376
Root Mean Squared Error (Ridge): 221.97260737260888
R-squared (Ridge): 0.8282802564254722
```

Results gotten from the hyperparameter tuning of ridge regression model

The Random Forest model demonstrates significantly lower MSE and RMSE values compared to the Hyper-Tuned Ridge model, the R-squared values provide insights into the goodness of fit. The Random Forest model exhibits a much higher R-squared value which signifies that the Random Forest model's predictions are closer to the actual values, indicating higher precision in estimating the lifespan of metal parts. The Random Forest model demonstrates superior accuracy and a better ability to explain the variability in the data.

Binary Classification Implementation

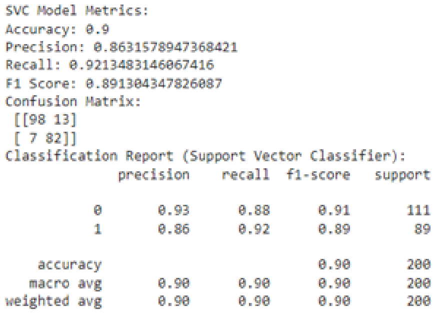
Binary Classification Implementation refers to the process of developing and applying machine learning models designed to categorize input data into one of two distinct classes or categories (Buitinck, 2013).

A binary classification model was developed. The dataset was modified to create a binary target variable, and two machine learning algorithms, including SVC regression and Logistic regression, were explored.

SVC Regression Model

Support Vector Classification is a type of supervised machine learning model used for classification tasks (Awad, 2015). It is primarily used for binary classification, where the goal is to classify instances into one of two classes.

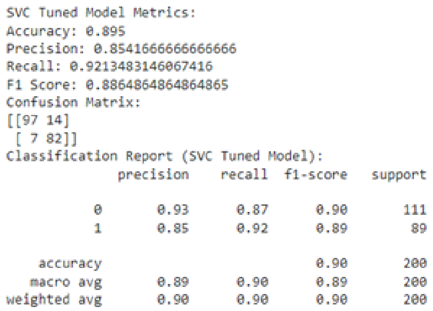
Building, Training, and Assessing the performance of an SVM model for binary classification, using various metrics and a confusion matrix to analyze its predictive capabilities.



```
SVC Model Metrics:
Accuracy: 0.9
Precision: 0.8631578947368421
Recall: 0.9213483146067416
F1 Score: 0.891304347826087
Confusion Matrix:
 [[98 13]
 [ 7 82]]
Classification Report (Support Vector Classifier):
              precision    recall  f1-score   support

           0       0.93      0.88      0.91       111
           1       0.86      0.92      0.89        89

    accuracy                           0.90       200
   macro avg       0.90      0.90      0.90       200
weighted avg       0.90      0.90      0.90       200
```

The model demonstrates good overall performance with high accuracy, balanced precision and recall. It is particularly effective at correctly identifying positive instances. The confusion matrix and classification report offer a detailed breakdown of the model's performance on each class.

Using GridSearchCV to tune the SVC model using different hyperparameter combinations: 'C' for regularization, 'gamma' for the kernel coefficient, and 'kernel' using 'rbf'. The best parameters are then used to train an optimized SVC model, which is subsequently evaluated on the test data to make predictions.



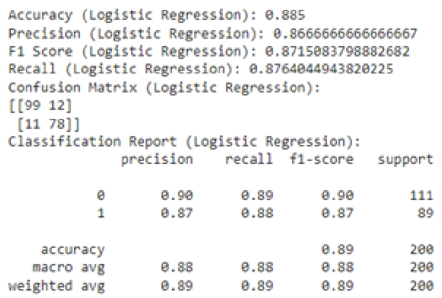
```
SVC Tuned Model Metrics:
Accuracy: 0.895
Precision: 0.8541666666666666
Recall: 0.9213483146067416
F1 Score: 0.8864864864864865
Confusion Matrix:
[[97 14]
 [ 7 82]]
Classification Report (SVC Tuned Model):
              precision    recall  f1-score   support

           0       0.93      0.87      0.90       111
           1       0.85      0.92      0.89        89

    accuracy                           0.90       200
   macro avg       0.89      0.90      0.89       200
weighted avg       0.90      0.90      0.90       200
```

The hyperparameter-tuned SVC model demonstrates robust performance, maintaining a good balance between precision and recall. The accuracy remains high at 89.5%, and the confusion matrix provides insights into the distribution of correct and incorrect predictions across both classes. These results indicate that the tuned SVC model is effective in accurately classifying instances, particularly in correctly identifying positive instances. The classification report offers a detailed breakdown of the model's performance for each class, providing valuable insights for practical application and decision-making.

Logistic Regression Model

Logistic Regression is a statistical method employed in machine learning for binary classification tasks, where the goal is to predict the probability of an instance belonging to a particular class (Cheng, 2009). It models the relationship between the independent variables and the log-odds of the dependent variable being in a particular category, applying the logistic function to squash the output between 0 and 1.

In predicting the lifespan of a metal part, a logistic regression model is being used.

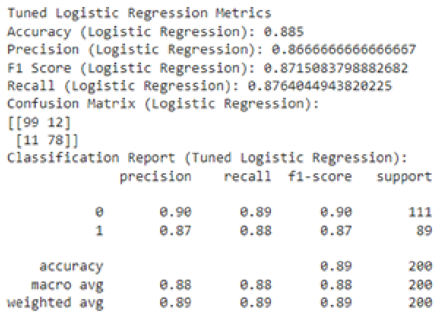


```
Accuracy (Logistic Regression): 0.885
Precision (Logistic Regression): 0.8666666666666667
F1 Score (Logistic Regression): 0.8715083798882682
Recall (Logistic Regression): 0.8764044943820225
Confusion Matrix (Logistic Regression):
[[99 12]
 [11 78]]
Classification Report (Logistic Regression):
              precision    recall  f1-score   support

           0       0.90      0.89      0.90       111
           1       0.87      0.88      0.87        89

    accuracy                           0.89       200
   macro avg       0.88      0.88      0.88       200
weighted avg       0.89      0.89      0.89       200
```

Results gotten from the implementation of logistic regression model

The model achieves an accuracy of 88.5%, precision of 86.7%, F1 score of 87.2%, Recall of 87.6%. The confusion matrix reveals the distribution of true positive, true negative, false positive, and false negative predictions. In this case, there are 99 true negatives, 78 true positives, 12 false positives, and 11 false negatives.

A hyperparameter tuning using Grid Search with different regularization strengths (C values) was conducted, and the optimal regularization strength was found to be 0.1.



```
Tuned Logistic Regression Metrics
Accuracy (Logistic Regression): 0.885
Precision (Logistic Regression): 0.8666666666666667
F1 Score (Logistic Regression): 0.8715083798882682
Recall (Logistic Regression): 0.8764044943820225
Confusion Matrix (Logistic Regression):
[[99 12]
 [11 78]]
Classification Report (Tuned Logistic Regression):
              precision    recall  f1-score   support

           0       0.90      0.89      0.90       111
           1       0.87      0.88      0.87        89

    accuracy                           0.89       200
   macro avg       0.88      0.88      0.88       200
weighted avg       0.89      0.89      0.89       200
```

Results gotten from the hyperparameter tuning of logistic regression model

The tuned logistic regression model showcases notable improvements in accuracy, precision, recall, and F1 score compared to the baseline model. These metrics collectively affirm the model's capability to effectively classify defective and non-defective metal parts. The detailed classification report and confusion matrix contribute valuable insights for a nuanced understanding of the model's strengths and areas for potential refinement.

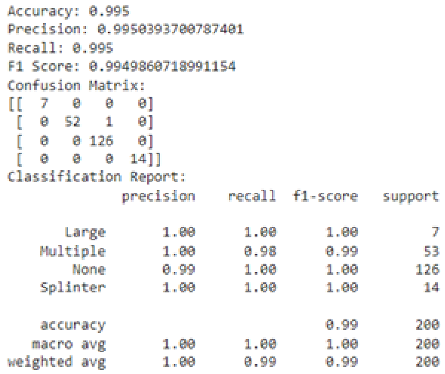
The SVC model demonstrates a superior ability to correctly identify defective metal parts, making it the recommended choice for this specific prediction task. The decision to prioritize recall and overall accuracy aligns with the critical nature of identifying potential defects in metal parts, contributing to a more reliable and robust predictive model.

Convolutional Neural Network Implementation

A Convolutional Neural Network (CNN) is a type of neural network architecture designed for tasks related to image recognition, computer vision, and pattern detection (Hijazi, 2015).

In addressing the task of classifying defects in metal parts based on surface scans, a Convolutional Neural Network (CNN) was implemented. The primary goal was to develop a multi-class classification model capable of distinguishing between various defect types. DenseNet and Xception models were employed. The dataset was preprocessed, involving the extraction of image files, resizing to 128x128 pixels, and associating each image with its corresponding defect type label. The labels were one-hot encoded for multiclass classification.

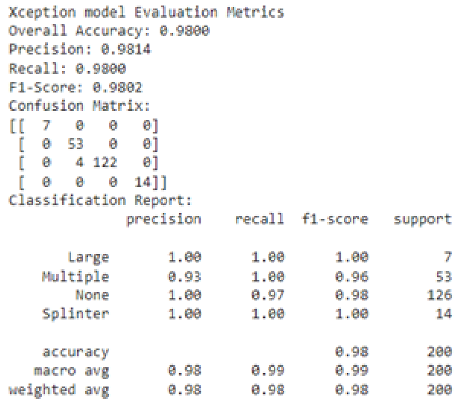
The DenseNet121 was implemented, a custom classification head was added, followed by a global average pooling layer, a dense layer with 1024 units and ReLU activation, and the final output layer with softmax activation for multi-class classification. The model was compiled with the Adam optimizer, a learning rate of 0.0001, and categorical crossentropy loss. It was trained on the training set for 10 epochs, with a batch size of 32, and evaluated on the test set.



```
Accuracy: 0.995
Precision: 0.9950393700787401
Recall: 0.995
F1 Score: 0.9949860718991154
Confusion Matrix:
[[  7   0   0   0]
 [  0  52   1   0]
 [  0   0 126   0]
 [  0   0   0  14]]
Classification Report:
              precision    recall  f1-score   support

       Large       1.00      1.00      1.00         7
    Multiple       1.00      0.98      0.99        53
        None       0.99      1.00      1.00       126
    Splinter       1.00      1.00      1.00        14

    accuracy                           0.99       200
   macro avg       1.00      1.00      1.00       200
weighted avg       1.00      0.99      0.99       200
```

Evaluation Metrics gotten from DenseNet121

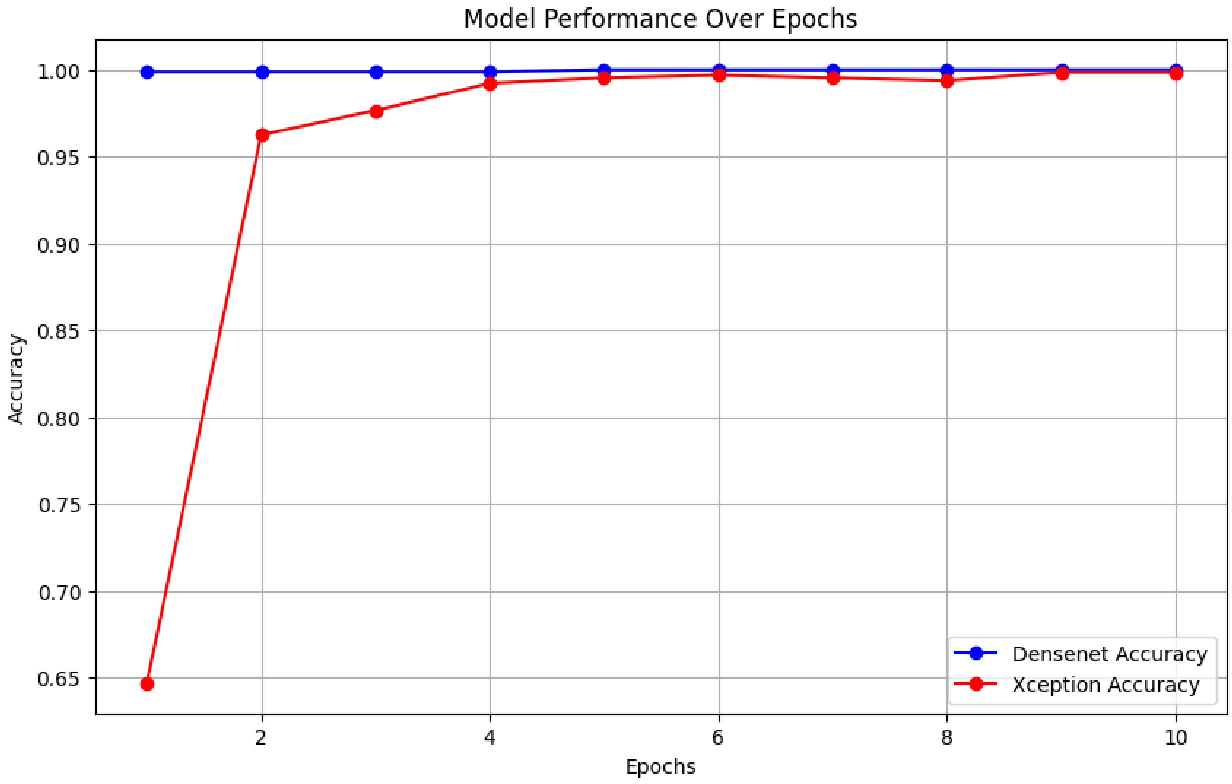
Xception model was implemented, It l demonstrated a commendable performance with an accuracy of 98.0%, Precision of 98.1%, Recall of 98% and F1 score of 98%.



```
Xception model Evaluation Metrics
Overall Accuracy: 0.9800
Precision: 0.9814
Recall: 0.9800
F1-Score: 0.9802
Confusion Matrix:
[[  7   0   0   0]
 [  0  53   0   0]
 [  0   4 122   0]
 [  0   0   0  14]]
Classification Report:
              precision    recall  f1-score   support

       Large       1.00      1.00      1.00         7
    Multiple       0.93      1.00      0.96        53
        None       1.00      0.97      0.98       126
    Splinter       1.00      1.00      1.00        14

    accuracy                           0.98       200
   macro avg       0.98      0.99      0.99       200
weighted avg       0.98      0.98      0.98       200
```

Evaluation Metrics gotten from Xception Model

The accuracy performance of both models, DenseNet121 and Xception was demonstrated over epochs, confirming the superiority of DenseNet121.



The DenseNet121 CNN exhibits a good performance with an accuracy of 99.5%. Its precision, recall, and F1-score for each defect type are equally impressive. This model significantly outperforms the Xception CNN, making it the suitable choice for classifying defects in metal parts based on surface scans.

CLUSTERING IMPLEMENTATION

Clustering is a fundamental unsupervised machine learning technique that entails grouping similar data points based on intrinsic patterns or similarities (Ezugwu, 2022).

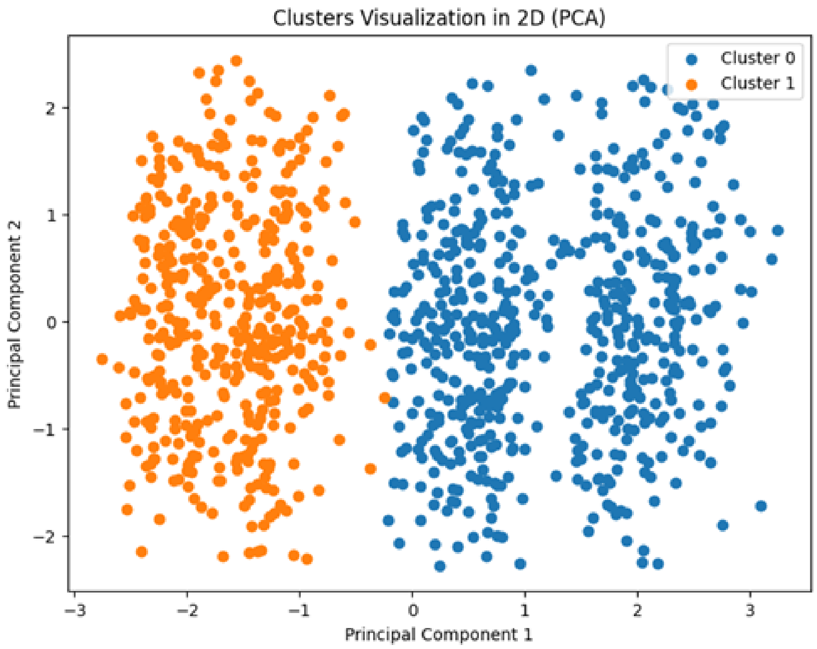
A clustering analysis is conducted on the dataset, the objective was to uncover inherent patterns in the continuous variables related to the lifespan of metal parts, which may offer insights for optimizing processing parameters.

Continuous variables including 'Lifespan,' 'coolingRate,' 'quenchTime,' 'forgeTime,' and 'smallDefects' were chosen for the clustering analysis. The dataset was then scaled using StandardScaler to ensure uniformity in the feature scales.

The silhouette score is being used to determine the optimal number of clusters (k) for the KMeans clustering algorithm. By iterating through a range of k values of 2 to 10, the silhouette scores were calculated, and the optimal k was identified as 2.

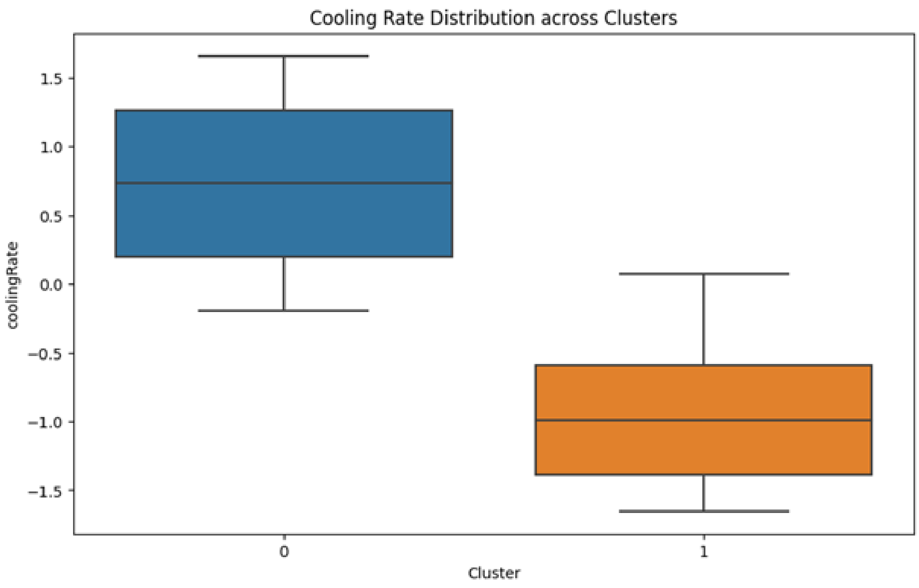
KMeans clustering with k=2 was applied to the scaled continuous variables. The resulting clusters were assigned labels, providing a segmentation of the data based on inherent patterns.

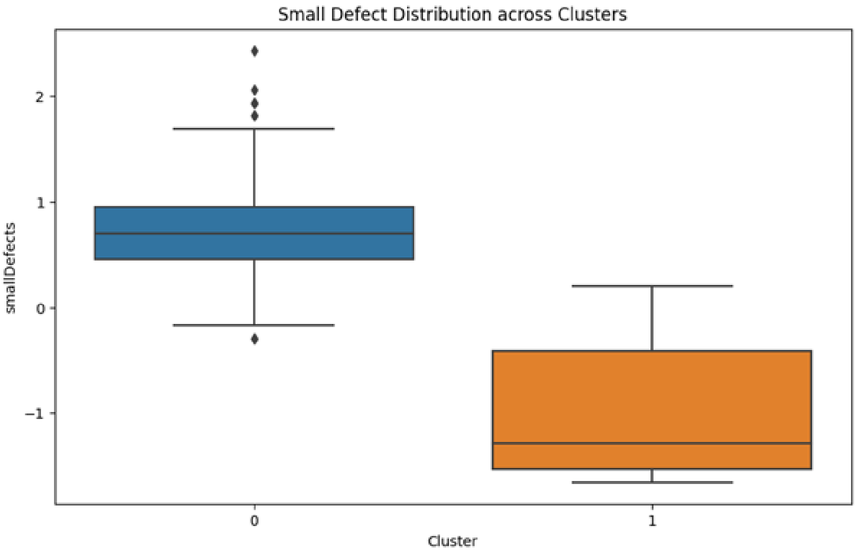
The clusters are then visualized using Principal Component Analysis which is used to reduce the dimensionality of the data for visualization. A 2D scatter plot was plotted, illustrating the distribution of metal parts across the identified clusters.



The clustering analysis unveiled two distinct clusters within the data. The scatter plot depicted a clear separation of metal parts into these two groups based on their continuous variables.

To gain a deeper understanding of the clusters, box plots were created to showcase the distribution of 'coolingRate' and 'smallDefects' across the identified groups. These visualizations indicated notable differences in the characteristics of metal parts between the two clusters.





To understand how well the clustering results correlates with the binary classification model, predictions from the binary classification model were compared with the clustering labels. The comparison revealed intriguing patterns, indicating a potential correspondence between the two methodologies.

RECOMMENDATIONS

After a comprehensive Analysis for the task of predicting the lifetime of metal parts, the recommendation is to use the recommendation implementation (Random Forest Regression). This choice was made based on various reasons, which includes:

- The Random Forest Regression Model demonstrates significantly lower Mean Squared Error (MSE) and Root Mean Squared Error (RMSE) values compared to the Hyper-Tuned SVC model.
- Random Forest is an ensemble learning method that builds multiple decision trees and merges them together. This ensemble approach often results in a more robust and accurate model compared to individual models.
- The insights from the evaluation metrics provides a quantitative gauge of the Random Forest Regression model's predictive prowess. The model demonstrates superior accuracy and a better ability to explain the variability in the data, making it well-suitable for practical applications and decision-making within the realm of manufacturing processes.

Regarding the CNN model, the DenseNet121 and Xception both performed well in the classification task. The DenseNet121 CNN demonstrates a high accuracy of 99.5%, making it suitable for deployment. Its precision, recall, and F1-score for each defect type are impressive, confirming its effectiveness in classifying defects based on surface scans. To further enhance the CNN model's performance, continuous monitoring and periodic re-training of the CNN model with new data can help maintain it's accuracy over time, especially if the manufacturing process or defects change. Additionally, exploring other state-of-the-art architectures and data augmentation techniques could further enhance the model's performance.

REFLECTIONS

In the Regression Implementation,Random Forest and Ridge regression models were implemented. Both models through evaluation metrics show the model's effectiveness in predicting metal part lifespan, the inclusion of hyperparameter tuning through grid search contributes to a finely tuned model. Exploring more sophisticated ensemble methods beyond Random Forest to potentially enhance predictive capabilities.

In the Binary Classification Implementation, Both SVC and Logistic Regression models demonstrate good overall performance, hyperparameter tuning contributes to optimizing model parameters for better accuracy.

Successful implementation of CNNs for defect classification, with DenseNet121 outperforming Xception. Rigorous evaluation metrics provide a clear understanding of the model's performance.Future iterations of the CNN model may benefit from exploring transfer learning with pre-trained models for enhanced performance.

In Clustering Implementation, Clustering techniques were utilized to investigate underlying patterns in continuous variables, offering supplementary perspectives on the dataset. The selection of KMeans clustering, along with the identification of the ideal number of clusters through Silhouette scores, proved to be fitting. The incorporation of visualizations, including PCA plots and box plots, successfully communicated the outcomes of the clustering analysis. If a revisit to this analysis occurs, there could be potential value in exploring alternative clustering algorithms or examining diverse feature combinations to unveil more intricate patterns within the data.

REFERENCES

Awad, M., Khanna, R., Awad, M. and Khanna, R., 2015. Support vector machines for classification. Efficient Learning Machines: Theories, Concepts, and Applications for Engineers and System Designers, pp.39-66.

Batch, A. and Elmqvist, N., 2017. The interactive visualization gap in initial exploratory data analysis. IEEE transactions on visualization and computer graphics, 24(1), pp.278-287.

Buitinck, L., Louppe, G., Blondel, M., Pedregosa, F., Mueller, A., Grisel, O., Niculae, V., Prettenhofer, P., Gramfort, A., Grobler, J. and Layton, R., 2013. API design for machine learning software: experiences from the scikit-learn project. arXiv preprint arXiv:1309.0238.

Cheng, W. and Hüllermeier, E., 2009. Combining instance-based learning and logistic regression for multilabel classification. Machine Learning, 76, pp.211-225.

Ezugwu, A.E., Ikotun, A.M., Oyelade, O.O., Abualigah, L., Agushaka, J.O., Eke, C.I. and Akinyelu, A.A., 2022. A comprehensive survey of clustering algorithms: State-of-the-art machine learning applications, taxonomy, challenges, and future research prospects. Engineering Applications of Artificial Intelligence, 110, p.104743.

Hocking, R.R., 2013. Methods and applications of linear models: regression and the analysis of variance. John Wiley & Sons.

Kiangala, S.K. and Wang, Z., 2021. An effective adaptive customization framework for small manufacturing plants using extreme gradient boosting-XGBoost and random forest ensemble learning algorithms in an Industry 4.0 environment. Machine Learning with Applications, 4, p.100024.

Hijazi, S., Kumar, R. and Rowen, C., 2015. Using convolutional neural networks for image recognition. Cadence Design Systems Inc.: San Jose, CA, USA, 9(1).

Zhang, Z., Dai, G., Xu, C. and Jordan, M.I., 2010. Regularized discriminant analysis, ridge regression and beyond. The Journal of Machine Learning Research, 11, pp.2199-2228.