# **Waste Cost Prediction Model**

**1. AIMS AND OBJECTIVES:**

**Aim:** The primary aim of this project is to develop a predictive model to estimate the cost of waste, particularly focusing on scrap and failed batches, based on various input features. This model will assist in optimizing waste management processes, reducing costs, and improving overall efficiency within the plate production process.

**Objectives:**

1. **Data Collection**: The dataset has been provided by the technical team at NIMS. It includes data\_batch.csv - scrap data from the plate production process and fail\_data.csv -fails data from quality control and production issues.
2. **Data Cleansing**: Process the data to remove inconsistencies, handle missing values, and ensure uniformity.
3. **Data Transformation**: Transform the cleansed data into a format suitable for analysis and visualization.
4. **Exploratory Data Analysis (EDA)**: Perform EDA to understand the underlying patterns, trends, and relationships within the data, particularly focusing on scrap percentages post-pouring on the production line and failed batches due to quality control or production issues.
5. **Feature Engineering**: Create new features or modify existing ones to enhance the predictive power of the model.
6. **Model Development**: Develop and train a predictive model using the prepared dataset to estimate waste costs accurately.
7. **Model Evaluation**: Evaluate the performance of the model using appropriate metrics to ensure its accuracy and reliability.
8. **Visualization**: Create visualizations to illustrate the data insights and model results effectively.

*Visuals would be included as part of other sections and there is not particular section for Visualisation.*

**2. DATASETS:**

The dataset has been provided by the technical team at NIMS. It includes data\_batch.csv - and fail\_data.csv.

**Batch Data Description**

The batch data primarily represents the various stages and parameters involved in the plate production process. This data includes information on production lines, pouring operations, and the resultant scrap percentages, all of which contribute to the overall waste cost. Key attributes in this dataset are:

Batch ID: Unique identifier for each production batch.

Production Line: The specific line on which the batch is processed.

Pouring Start Time: The timestamp when the pouring process begins.

Pouring End Time: The timestamp when the pouring process ends.

Record Scrap %: The percentage of the batch that is recorded as scrap post-pouring, typically due to issues like crashes, bubbles, equipment malfunctions, or cracked dishes.

Material Type: Type of material used in the batch.

Operator ID: Identifier for the operator overseeing the batch process.

Production Duration: The total time taken for the batch production.

Waste Cost: The cost associated with the waste generated from the batch, calculated based on the scrap percentage and material costs.

The aim of analyzing this data is to understand the factors leading to scrap generation, predict the associated waste cost, and identify patterns or anomalies that can help in reducing the scrap percentage and optimizing production efficiency.

**Fail Data Description**

The fail data pertains to batches that did not pass quality control (QC) testing or encountered significant production issues leading to complete batch scrapping, contributing to the waste cost. This dataset includes information about the nature and cause of the failure. Key attributes in this dataset are:

Batch ID: Unique identifier for each failed batch.

Failure Type: Description of the failure, such as QC failure or production issue.

Failure Cause: Specific reason for the batch failure, such as vessel sterilization failure or boiler fault.

Failure Time: The timestamp when the failure was identified.

Operator ID: Identifier for the operator overseeing the batch at the time of failure.

Material Type: Type of material used in the batch.

Production Line: The specific line on which the batch was processed.

Waste Cost: The cost associated with the waste generated from the failed batch, considering the complete scrapping of the batch.

The number of rows in Batch Data: 2171

The number of columns in Batch Data: 15

The number of rows in Fail Data: 269

The number of columns in Fail Data: 18

The dataset has very few rows as seen from above results. Ensuring robust preprocessing, careful model selection, and rigorous validation are key to overcoming the challenges posed by limited data. The analysis of this data aims to identify common failure points and underlying causes, predict the associated waste cost, and enable proactive measures to prevent such failures, thereby improving overall production reliability and cost efficiency.

**3. DATA CLEANSING:**

Data cleansing is a critical step to ensure that the data is consistent, accurate, and suitable for analysis. Few steps were taken to cleanse the batch and fail datasets.

1. Handling missing values:

The datasets were checked for missing values and the code produced the below results:

A screenshot of a computer program

Description automatically generated

A screenshot of a computer

Description automatically generated

So, the below insights can be derived:

* The Batch Data is very complete and should provide reliable information for analysis.
* The Fail Data has more missing values, particularly in the production quantity and additional information fields.
* The missing vessel information in the Fail Data might limit some analyses related to equipment-specific issues.

The high number of missing values in Total Quantity Produced for failed batches might make it challenging to accurately calculate yield or loss for these cases.

Imputation is a valuable technique in data preprocessing, enabling more robust and comprehensive analysis by addressing the issue of missing data. Instead of removing records or features with missing data, which can lead to loss of valuable information, imputation fills in the missing values with substituted values based on various strategies. The goal of imputation is to provide a complete dataset for analysis and modelling without introducing significant bias.

1.First, the code separates the columns of batch\_data into two categories:

numeric\_columns: columns containing numerical data

categorical\_columns: columns containing non-numerical (categorical) data

2.For numeric columns:

It creates a SimpleImputer object with the strategy set to 'median'.

This means it will replace missing values with the median value of each column.

The fit\_transform method is then used to both calculate the median values and apply the imputation to the numeric columns.

3.For categorical columns:

It creates another SimpleImputer object, this time with the strategy set to 'most\_frequent'.

This means it will replace missing values with the most frequently occurring value in each column.

Again, fit\_transform is used to determine the most frequent values and apply the imputation to the categorical columns.

This approach ensures that missing values are handled appropriately based on the type of data in each column. Using the median for numeric data helps to minimize the impact of outliers, while using the most frequent value for categorical data is a common strategy when there's no meaningful "average" for categories.

The similar method is applied to the fail dataset. After this we check if it has worked, and we can see that there are no missing values as seen in below screenshot.

A screenshot of a computer screen

Description automatically generated

1. Removal of duplicate values in columns:

While plotting count plot for ‘SAP Fail Category/Description’ , it was noticed that there were duplicates in the values as seen in below screenshot.

A graph of a bar graph

Description automatically generated

This was corrected using a function which implements a data standardization process for two datasets, 'batch\_data' and 'fail\_data' that converts string inputs to lowercase, strips whitespace, and capitalizes the first letter. This function is applied to specific columns in both datasets, including 'SAP Fail Category / Description', 'Sub Category - Fails', 'Manufacturing Location', 'Vessel', and 'Material'. Additionally, for the 'SAP Fail Category / Description' column in the 'fail\_data' dataset, a dictionary-based mapping is used to standardize category names, ensuring consistency across various spellings and capitalizations (e.g., 'Equipment', 'EQUIPMENT', and 'equipment' are all standardized to 'Equipment'). This standardization process aims to improve data consistency and facilitate more accurate analysis and comparisons between the datasets.

**4. EXPLORATORY DATA ANALYSIS AND VISUALIZATION:**

Data analysis involves examining the dataset to uncover insights, patterns, and relationships that can help in achieving the project objectives. The following analyses were performed on the batch and fail datasets.

1. The summary statistics (mean, median, standard deviation) were calculated for the features such as Waste in ML, Theoretical Yield, Waste Total Cost.
2. Examined the distribution of these features to understand their central tendency and dispersion.
3. Computed the correlation matrix to identify relationships between features.
4. Plotted histograms, box plots, violin plots for numerical features to identify skewness, outliers, and overall distribution.
5. Used techniques such as feature selection.

**Correlation matrix:**

The correlation matrix provides a visual representation of the relationships between different variables in the dataset. Here are some key insights based on the provided heatmaps:

Strong Positive Correlations:

Theoretical Yield and G.R.Qty: A very high positive correlation (close to 1) indicates that the theoretical yield is closely related to the gross quantity.

Total Input in ML and Theoretical Yield (ML): High positive correlation, suggesting that as the total input in milliliters increases, the theoretical yield in milliliters also increases.

Waste in ML and % Waste Loss: The positive correlation between waste in milliliters and the percentage of waste loss indicates that these two measures increase together.

Strong Negative Correlations:

Yield Efficiency and Waste Total Cost: A negative correlation suggests that higher yield efficiency is associated with lower waste total costs.

% Waste Loss and Yield Efficiency: Strong negative correlation, indicating that higher waste loss percentages are associated with lower yield efficiency.

Interesting Observations:

Batch and Period: There is a high positive correlation between the batch and period, which might be due to sequential production processes within specific periods.

Order Quantity and Cost Variance: Positive correlation implies that higher order quantities are associated with higher cost variance.

Week of Manufacture and Week No.: These two variables show high positive correlation, which is expected as they both relate to the timing of production.

Predictive Insights:

Waste Total Cost and Various Features: Understanding the correlations between waste total cost and other features like % waste loss, yield efficiency, and theoretical yield can help in building a predictive model for waste costs. Features with strong correlations (positive or negative) are particularly valuable for prediction.

Multicollinearity Considerations:

Highly Correlated Features: Features that are highly correlated with each other (e.g., theoretical yield and G.R.Qty) might introduce multicollinearity issues in regression models. Techniques like Principal Component Analysis (PCA) or removing one of the correlated features could be considered.

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

**Distribution of Waste Total Cost:**

**Right-Skewed Distribution:** The distribution remains right-skewed, with a significant concentration of data points at the lower end of the Waste Total Cost spectrum and a long tail extending towards the higher end. This indicates that most observations have low waste costs, while a few have very high waste costs.

**Frequency Concentration:** The highest frequency of data points is still clustered around the lowest values of Waste Total Cost. The additional bins allow us to see that the majority of the data points are very close to zero, indicating that low waste costs are common.

**Outliers:** The histogram shows a few extreme values (outliers) with very high Waste Total Cost. These outliers can significantly impact the mean and variance of the dataset and might need further investigation to understand their causes.

**Data Granularity:** The increased number of bins provides a more granular view of the distribution, revealing more detail about the spread of the data. This helps in identifying patterns and understanding the distribution better.

**KDE Line:** The Kernel Density Estimate (KDE) line still shows a sharp peak at the start, confirming the high concentration of low waste costs. The line gradually tapers off, indicating the presence of fewer high-cost values.

A graph of a distribution of waste

Description automatically generated

**Box Plot:**

1. The plot has only one category on the x-axis labeled "Failed", indicating that all data points represent failed batches.
2. The y-axis represents the Waste Total Cost, ranging from 0 to about 550,000 units (likely in currency).
3. The box part of the plot, which typically represents the interquartile range (IQR), is extremely compressed and appears as a thin line near the bottom of the plot. This suggests that the majority of the waste costs are concentrated at the lower end of the scale.
4. There's a long vertical line (whisker) extending upwards from the box, indicating a wide range of values above the upper quartile.
5. There are two visible outlier points:
   * One outlier is slightly above the top of the whisker.
   * Another extreme outlier is plotted near the top of the chart, at around 550,000 units.
6. The presence of these outliers, especially the extreme one, suggests that while most failed batches result in relatively low waste costs, there are occasional instances of very high-cost failures.
7. The compressed nature of the box and the long whisker indicate a highly skewed distribution of waste costs, with most values clustered at the lower end but with some very high values pulling the distribution to the right.

This plot reveals that while most failures result in relatively low waste costs, there are rare but significant high-cost failures that could have a substantial impact on overall waste management and cost control strategies.

A graph of a graph

Description automatically generated with medium confidence

**Scatter Plot of Theoretical Yield vs Waste Total Cost**

High Outlier: There is an exceptional batch with a very high waste cost that needs to be investigated to understand the cause of such high wastage.

Low Waste Costs: Most batches have low waste costs, which is a positive indicator of the production process efficiency.

Concentration of Theoretical Yield: Most batches have theoretical yields below 1500, which may suggest a common production scale or capacity limit.

A graph of a graph

Description automatically generated with medium confidence

**Pair plots:**

Pair plots of selected features (Theoretical Yield', 'G.R.Qty', 'Waste in ML', 'Waste Total Cost', 'Failed) were implemented.

A graph of a graph of a variety of blue dots

Description automatically generated with medium confidence

A graph of a graph of a number of data

Description automatically generated with medium confidence

**Correlation Between Yield and Gross Quantity:** There is a strong correlation between theoretical yield and gross quantity, which is expected in most production processes.

**Low Waste Costs:** Most batches have low waste costs, but the presence of an outlier indicates that some batches can incur extremely high waste costs.

**Skewed Distributions:** Waste-related features have skewed distributions with most values near zero, indicating that waste events are relatively rare but can be severe when they do occur.

**Multimodal Yield Distribution:** Theoretical yield has a multimodal distribution, suggesting different production processes or batch sizes.

**Count Plot of SAP Fail Category:**

A graph of a bar graph

Description automatically generated

The above is a count plot of SAP Fail Categories, providing insights into the frequency of different types of failures in a manufacturing or production process. Here are the key insights from this visualization:

**Most Common Fail Categories:**

Appearance is the most frequent fail category with about 70 occurrences. Performance is the second most common, with around 62 instances. Contamination is the third, with approximately 51 cases.

**Mid-range Fail Categories:**

Equipment-related fails are moderately common, with about 37 instances. Manufacturing issues occur around 19 times.

**Less Frequent Fail Categories:**

Operator-related fails are less common, with about 17 occurrences. PH (likely referring to pH level issues) appears about 10 times. Documentation is the least frequent category, with only 3 instances.

**Distribution:**

There's a clear disparity between the most common and least common fail categories. The top three categories (Appearance, Performance, Contamination) account for a significant majority of all failures.

**Process Implications:**

Quality control processes might need more focus on appearance-related issues. Performance and contamination are also major areas of concern that may require attention. While equipment issues are significant, they're not the leading cause of failures.

**Potential Areas for Improvement:**

Addressing appearance, performance, and contamination issues could potentially reduce over half of all failures. Operator training and documentation processes seem to be working relatively well, given their lower frequency.

Further Analysis Needed:

It would be beneficial to investigate why appearance issues are so prevalent. Understanding the specific nature of performance and contamination failures could lead to targeted solutions.

**5. FEATURE ENGINEERING AND MERGING:**

Feature engineering and merging datasets are critical steps in the data preparation process that significantly influence the performance of machine learning models. Below is a detailed description of how feature engineering and merging were performed in the provided code. Feature engineering involves creating new features or modifying existing ones to improve the predictive performance of machine learning models. Here are the steps taken for feature engineering in the code:

**Creation of New Features:**

Yield Efficiency: This feature is calculated as the ratio of the actual yield (G.R.Qty) to the theoretical yield. It provides insights into how efficiently the production process is converting inputs into outputs.

Waste Percentage: This feature is calculated as the percentage of waste produced relative to the total input. It helps in understanding the extent of wastage in the production process.

**Handling Missing Values:**

Numeric Columns: Missing values in numeric columns of batch\_data were imputed using the median value.

Categorical Columns: Missing values in categorical columns were imputed using the most frequent value.

For fail\_data, specific strategies were applied based on the column type and the nature of the data. For instance, numeric columns were converted to numeric types and imputed with median or zero, while categorical columns were filled with specific values or indicators.

**Merging of Datasets**

Merging datasets involves combining multiple data sources to create a comprehensive dataset for analysis. The steps for merging in the provided code are as follows:

**Merging batch data and fail data:**

The datasets are merged on common columns (Material and Batch from batch\_data, and Product/Material and Batch Number from fail\_data) using a left join. This ensures that all records from batch\_data are retained, and matching records from fail\_data are included.

**Handling NaN Values After Merge:**

After merging, NaN values in the resulting dataset are handled based on the data type and context. For categorical columns, missing values are filled with 'No Fail' or a similar indicator. For numeric columns, missing values are filled with zero or the median/mean value.

**Column and Row Count:**

The number of rows in Merged Dataset: 2171

The number of columns in Merged Dataset: 35

**Creating a Binary Indicator for Failures:**

A new binary column Failed is created to indicate whether a batch has failed based on the presence of data in the fail\_data.

After the merge, some columns become redundant. These columns are dropped to simplify the dataset.

The resulting merged\_data dataset now contains a comprehensive set of features from both batch\_data and fail\_data, with newly engineered features and appropriately handled missing values. This prepared dataset can be used for further analysis and modeling to predict outcomes such as waste total cost and identify factors contributing to failures.

**6. MODEL DEVELOPMENT:**

The model development process involves several key steps, including data splitting, feature selection, model training, evaluation, and iteration. Here’s a detailed description of these steps based on the provided code.

**Data Split**

The first step in model development is to split the data into training and test sets. This allows us to train the model on one portion of the data and evaluate its performance on another portion that the model has not seen during training.

**Selecting Features and Target:**

Features and the target variable are selected from the merged dataset.

features = ['Theoretical Yield', 'G.R.Qty', 'QC Qty (ML)', 'Waste in ML', '% Waste loss',

'Yield\_Efficiency', 'Waste\_Percentage', 'Failed']

target = 'Waste Total Cost'

**Polynomial Feature Transformation:**

Polynomial features are generated to capture interactions between features. Polynomial features are a form of feature engineering that involves generating new features by taking combinations of existing features to a specified degree. This helps in capturing non-linear relationships in the data that linear models might miss.

**Scaling and Transforming Features:**

The features are scaled using RobustScaler to handle outliers, and the target variable is transformed using PowerTransformer to ensure normal distribution and handle outliers.

**Splitting the Data:**

The scaled features and transformed target variable are split into training and test sets.

**Feature Selection**

Feature selection is crucial for improving the model’s performance by eliminating irrelevant or redundant features. Recursive Feature Elimination (RFE) is used for feature selection in this process.

**Applying RFE:**

RFE with RandomForestRegressor as the estimator is used to select the top 20 features.

**Model Training**

Model training involves defining and training the machine learning model. A stacking regressor is used in this process, which combines multiple base models to improve predictive performance.

**Defining Base Models:**

Four base models are defined: RandomForest, GradientBoosting, XGBoost, and LightGBM.

**Defining Meta-Model:**

An MLPRegressor (Multi-Layer Perceptron Regressor) is used as the meta-model.

**Creating and Training Stacking Regressor:**

A stacking regressor is created using the base models and the meta-model, and then it is trained on the training data.

**Model Evaluation**

Evaluating the model’s performance is crucial to understand how well it generalizes to new, unseen data.

**Cross-Validation:**

Cross-validation is performed to further validate the model’s performance.

**Model Iteration and Improvement:**

The model is iterated and improved by identifying important features and addressing potential issues like outliers.

**Identifying Important Features:**

The top 10 important features are identified using RandomForestRegressor.

**Visualizing Predictions:**

Actual vs. predicted values are plotted to visually inspect the model’s performance.

**Handling Outliers:**

Outliers are removed, and the model is re-evaluated to check for improvements.

**Visualizing Improved Predictions:**

Predictions are visualized again after outlier removal to inspect any improvements.

By following these steps, the model is developed, trained, evaluated, and iteratively improved to achieve better performance in predicting waste total cost and identifying factors contributing to production failures.

**7. MODEL EVALUATION RESULTS AND INTERPRETATION:**

After training the stacking regressor model and performing cross-validation, the results provide insights into the model's performance. Below is a detailed description of the results, including the metrics, their meanings, and possible interpretations.

**Model Evaluation Metrics**

**Mean Squared Error (MSE):**

MSE measures the average squared difference between the predicted and actual values. A lower MSE indicates a better fit to the data.

MSE: 2.4017275346879488

**R-squared (R2) Score:**

The R-squared score represents the proportion of the variance in the dependent variable that is predictable from the independent variables. It ranges from 0 to 1, with higher values indicating better performance.

R2: 0.21898671943590742

**Cross-Validation Results**

Cross-validation provides a more robust evaluation by splitting the data into multiple training and validation sets and averaging the performance across these splits.

**Cross-Validation R2 Scores:**

The R2 scores from 5-fold cross-validation are as follows:

Fold 1: 0.96287192

Fold 2: 0.93218684

Fold 3: 0.83183895

Fold 4: 0.95874365

Fold 5: 0.20166846

Mean CV R2 score: 0.7774619659594698

**Interpretation of Results**

MSE Interpretation:

An MSE of 2.4017 indicates that the model's predictions deviate from the actual values by a squared difference of about 2.4017 units on average. While MSE is useful for understanding the magnitude of prediction errors, it is not always easy to interpret without context (e.g., the scale of the target variable).

R2 Score Interpretation:

The R2 score of 0.219 suggests that approximately 21.9% of the variance in the target variable (Waste Total Cost) is explained by the model. This is relatively low, indicating that the model may not be capturing all the relevant patterns in the data.

Cross-Validation R2 Scores:

The cross-validation results show high variability in the R2 scores across different folds. While four folds exhibit high R2 scores (above 0.8), indicating good performance, one fold has a significantly lower R2 score (0.2017), dragging down the average.

The mean CV R2 score of 0.7775 suggests that, on average, the model explains about 77.75% of the variance in the target variable during cross-validation. This is considerably higher than the single test set R2 score, indicating that the model performs better on different subsets of the data.

A graph with a line

Description automatically generated

**Key Insights:**

* Most of the data points are clustered around the origin (0,0), indicating that for many instances, the predicted values are close to the actual values, especially for lower values of the transformed Waste Total Cost.
* There is a clear presence of outliers, particularly the single data point far from the main cluster, with an actual value of around 35 and a corresponding predicted value of around 20. This indicates a significant deviation from the ideal prediction line for this instance.
* Most of the data points lie close to the red dashed line, suggesting that for these points, the model predictions are reasonably accurate. However, there are a few instances where the predictions deviate significantly from the actual values.
* The outliers significantly impact the model's performance, as they indicate instances where the model's predictions are far from the actual values. These outliers can skew the error metrics (such as MSE) and reduce the overall R2 score.
* The points are symmetrically distributed around the red dashed line, indicating that the model does not systematically overpredict or underpredict the values, but there is noise in the predictions, which could be due to the complexity of the data or limitations of the model.

**Understanding "Original Scale"**

In the context of the provided code and machine learning model evaluation, the term "original scale" refers to the measurement units and data distribution of the target variable (Waste Total Cost) before any transformation was applied.

**Why is the Target Variable transformed?**

Normalization: To ensure that the target variable has a normal distribution, making it easier for the model to learn and perform well.

Handling Outliers: To mitigate the impact of extreme values, which can distort the model's learning process.

Stabilizing Variance: To stabilize the variance across different values of the target variable, making the model's predictions more reliable.

Transformation Applied In the code, the PowerTransformer was used to transform the target variable.

**Model Evaluation in Transformed Scale**

During model training and initial evaluation, the target variable is in its transformed scale, which helps the model perform better. However, for interpretability, it is essential to convert the predictions back to the original scale.

Calculating Metrics in Original Scale: To understand the model’s performance in real-world terms, the predictions and actual values are converted back to their original scale.

**Importance of Evaluation in Original Scale**

Interpretability: Metrics like Mean Squared Error (MSE) and R-squared (R2) in the original scale provide insights that are easier to interpret in practical terms, as they reflect the actual measurement units (e.g., dollars, kilograms).

Real-World Relevance: Stakeholders and decision-makers often require results in the original scale to make informed decisions. For instance, understanding the cost savings or losses in actual dollar amounts is crucial.

Model Validation: It helps in validating the model's performance in terms of the business problem it aims to solve, ensuring that improvements translate to real-world benefits.

Example: Consider a scenario where the target variable is Waste Total Cost in dollars:

Transformed Scale: The target variable might be transformed to a normalized distribution with mean 0 and standard deviation 1.

Original Scale: The target variable is in dollars, and we want to evaluate the model’s predictions in this scale.

The original scale refers to the data's natural units before any transformation was applied. Evaluating the model in this scale ensures that the results are interpretable and relevant to real-world applications, providing a clear understanding of the model's effectiveness in predicting actual outcomes.

**MSE (Original Scale): 681782386.5131766** is quite high, indicating that, on average, the squared errors between the predicted waste total costs and the actual waste total costs are substantial. This suggests that the model's predictions are not very accurate in the original scale of the data.

**R2 (Original Scale): 0.0031187631223814627** is extremely low, close to zero. This implies that only about 0.31% of the variance in the waste total cost is explained by the model. Essentially, the model is barely better than a horizontal line (i.e., predicting the mean of the target variable).

Now that we know there are outliers, we remove the outliers.

After performing outlier removal, retraining the stacking regressor, and conducting cross-validation, the results provide insights into the model's performance.

A graph with a red line

Description automatically generated

 **Overall Fit:** Most of the data points lie close to the red dashed line, indicating that the model's predictions align well with the actual values for most of the dataset.

 **Outliers:** There is a significant outlier far from the main cluster with an actual value of over 500,000 and a much lower predicted value, suggesting a substantial prediction error for this instance.

 **Range of Predictions:** The plot shows that the model performs well within a lower range of Waste Total Cost, but struggles with accuracy for extreme high values, highlighting the impact of outliers on model performance.

**Results After Outlier Removal and Retraining**

After removing outliers, retraining the stacking regressor, and re-evaluating the model, the results are as follows:

Original Data Shape: (2171, 44)

Cleaned Data Shape: (2147, 44)

Mean Squared Error (MSE): 0.003661195394590265

R-squared (R2): 0.9658258023263161

**Interpretation:**

* Mean Squared Error (MSE): The MSE value is significantly lower after outlier removal. This indicates that the model's predictions are much closer to the actual values, meaning the average squared difference between the predicted and actual values has reduced substantially.
* R-squared (R2): The R2 score is approximately 0.966, indicating that the model explains about 96.58% of the variance in the target variable after removing outliers. This is a substantial improvement compared to the initial results, showing that the model now captures most of the variation in the data.

**Implications:**

* Performance Improvement: The significant reduction in MSE and the increase in R2 score demonstrate that the model's performance has greatly improved after handling outliers. This suggests that outliers were previously distorting the model's predictions.
* Model Robustness: The high R2 score indicates a strong fit, suggesting that the model is robust and can effectively predict the target variable for most data points.

After performing cross-validation, the results are as follows:

**Cross-Validation R2 Scores:**

Fold 1: 0.9616

Fold 2: 0.9563

Fold 3: 0.9418

Fold 4: 0.9567

Fold 5: 0.9488

Mean Cross-Validation R2 Score: 0.9530

**Interpretation:**

* **Consistent High Performance:** The R2 scores from cross-validation are consistently high across all folds, indicating that the model maintains strong performance across different subsets of the data. This consistency is crucial for ensuring that the model generalizes well to unseen data.
* **Mean R2 Score:** The mean R2 score of 0.9530 across all folds suggests that, on average, the model explains 95.30% of the variance in the target variable during cross-validation. This further reinforces the model’s robustness and reliability.

**Implications:**

* Generalization: The consistent and high cross-validation R2 scores imply that the model generalizes well across different data subsets, making it reliable for real-world applications.
* Validation of Improvements: The improved cross-validation scores validate the effectiveness of the steps taken (such as outlier removal and feature selection) to enhance the model.

**LIME Explanation of the Prediction**

**A screenshot of a computer

Description automatically generated**

The provided image is the output of LIME (Local Interpretable Model-agnostic Explanations) explaining the prediction of the stacking regressor for a specific instance. Here’s a detailed explanation:

**Instance Features:**

The instance being explained has feature values listed at the top. For example:

Waste in ML: 4.56

% Waste loss: 4.57

Theoretical Yield Yield\_Efficiency: 8.39

**Actual vs Predicted Value**

Actual Value: 0.45744387460756497

Predicted Value: 0.6326555879560958

The predicted value is relatively close to the actual value, suggesting that the model has made a reasonably accurate prediction for this instance.

**Positive Contributions (orange bars):**

Waste in ML > 0.66 has the highest positive contribution, indicating that higher waste in ML strongly influences the prediction to increase.

% Waste loss > 0.65 also contributes positively to the prediction.

Other features like Theoretical Yield Yield\_Efficiency, Theoretical Yield, and Waste\_Percentage have smaller but positive contributions.

**Negative Contributions (blue bars):**

G.R.Qty <= -0.60 contributes negatively, reducing the predicted value.

Other features contribute less significantly to decreasing the predicted value.

**Feature Values:**

The values of each feature for the instance are provided on the right side. For example:

Waste in ML: 4.56

% Waste loss: 4.57

Theoretical Yield Yield\_Efficiency: 8.39

G.R.Qty: -1.50

QC Qty (ML): 5.28

Waste in ML and % Waste loss are the dominant features influencing the prediction, both contributing significantly to increasing the predicted waste total cost. The LIME explanation shows that the model considers higher waste metrics as critical factors for predicting higher waste total costs, which aligns with domain knowledge. There is a balance of both positive and negative contributions from different features, indicating that the model uses a combination of factors to arrive at the final prediction. The LIME output provides a clear, interpretable explanation of how different features contribute to the model’s prediction for this specific instance. This transparency helps in understanding the model’s decision-making process and can be valuable for validating the model and identifying areas for improvement.

A graph of a graph showing a line of blue dots

Description automatically generated with medium confidence

**Key Insights:**

**Improved Fit:** Most of the data points are closely clustered around the red dashed line, indicating that the model's predictions are closely aligned with the actual values after outlier removal. This suggests that the model's accuracy has improved significantly.

**Reduced Spread:** Compared to the previous plots (before outlier removal), there is a noticeable reduction in the spread of the data points. This indicates that removing outliers has helped in reducing the prediction errors, making the predictions more consistent.

**Residuals and Variance:** While most points are tightly clustered around the line, there are still some deviations, particularly at the ends of the scale. However, these deviations are less pronounced than before, indicating a better fit overall.

**Accuracy:** The close alignment of the data points with the red dashed line indicates that the model is making more accurate predictions, with fewer large errors, compared to before outlier removal.

**Generalization:** The improved fit suggests that the model generalizes better to new data, as it can more accurately predict outcomes without being influenced by extreme outliers.

**Model Robustness:** The consistency of the predictions after outlier removal demonstrates the robustness of the model, as it performs well across a range of transformed Waste Total Cost values.

The results after outlier removal and cross-validation demonstrate a significant improvement in the model's performance. The model is now more accurate and reliable, with the ability to generalize well to new data. These improvements suggest that the steps taken to preprocess the data and refine the model were effective, resulting in a robust tool for predicting Waste Total Cost and understanding the factors that influence production efficiency.

**8. Summary:**

The code begins by importing necessary libraries for data manipulation, visualization, modeling, and explanation. It then loads the batch\_data and fail\_data datasets, handling missing values by imputing with median values for numeric columns and the most frequent values for categorical columns. Exploratory data analysis includes generating correlation heatmaps and distribution plots to identify relationships among key variables, such as yield, waste, and costs. New features like Yield\_Efficiency and Waste\_Percentage are engineered to provide additional insights into production efficiency and waste.

After preprocessing, the datasets are merged, and any remaining missing values are addressed. Outliers are detected and removed to improve model performance. The dataset is split into training and test sets, with Recursive Feature Elimination (RFE) used for feature selection. A stacking regressor model, comprising base models like RandomForest, GradientBoosting, XGBoost, and LightGBM, and an MLPRegressor as the meta-model, is trained and evaluated. Model performance is assessed using metrics like MSE and R2, both before and after outlier removal, showing significant improvement post-outlier removal.

Cross-validation is conducted to ensure the model generalizes well, and the results indicate consistent high performance. LIME (Local Interpretable Model-agnostic Explanations) is applied to explain the model's predictions, providing interpretability and insights into feature contributions. The LIME output helps identify the most influential features driving the model's predictions, enhancing the understanding of the model's decision-making process and ensuring transparency in the predictive modelling approach.