Master Project Management and Data Science (MPMD)

Advanced Computational Data Analytics (SoSe2024)

Secom Case Study (Team 5)

Submitted by:

Dhruvi Jay Patel (s0592755)

Naman Mathpal (s0590500)

Bhoomika Jagadeesha (s0590573)

Jui Prasad Kulkarni (s0590496)

Ankit Satish Gupta (s0590516)

Under the guidance of:

Professor Dr. Tilo Wendler

Date: **26.07.2024**

Contents

List of figures

List of tables

Preface

This thesis represents an in-depth exploration of managing complex datasets with numerous variables and developing professional scripts using R or Python. The primary objective is to employ data cleansing and feature selection or reduction methods before applying data mining techniques to predict defaults, focusing on the SECOM dataset from the Machine Learning Repository (2008). This dataset, originating from the semiconductor industry, presents a unique challenge due to the vast number of sensors used to monitor the production process, resulting in a high-dimensional dataset.

The semiconductor manufacturing process is closely monitored by various sensors to ensure quality and efficiency, given that defects identified in the later stages of production can lead to significant financial losses. Thus, the ability to predict defaults accurately is paramount. This research aims to create parsimonious statistical models with robust predictive capabilities, based on the minimal necessary variables.

Throughout this thesis, guidance was derived from notable works by Munirathinam and Ramadoss (2016), and Kerdprasop (2003), which provided foundational steps for the analysis. These sources offered a structured approach but required the application of comprehensive knowledge and skills to achieve optimal predictive models. The complexity and intricacy of the SECOM dataset necessitated meticulous data cleansing and strategic feature selection to enhance the model's predictive accuracy.

The process involved in this case study underscores the importance of dealing with high-dimensional data in practical applications. It reflects on the critical balance between model simplicity and explanatory power, emphasizing the necessity for efficient data preprocessing techniques and sophisticated data mining methods.

This thesis documents the journey from data acquisition to the implementation of predictive models, capturing the challenges and solutions encountered along the way. It aims to contribute valuable insights into the field of data science, particularly in handling complex datasets in industrial applications. By following the structured yet flexible

approach outlined in this study, future researchers and practitioners can develop effective predictive models tailored to their specific needs and datasets.

I would like to extend my gratitude to my advisors, colleagues, and family for their unwavering support and guidance throughout this project. Their encouragement has been instrumental in the completion of this work. This thesis is dedicated to those who seek to harness the power of data to drive innovation and efficiency in various domains.

1.Introduction

The rapid advancement of technology in the semiconductor industry has necessitated the development of sophisticated methods for monitoring and optimizing the production process. Given the high cost associated with defects identified in later stages of production, the ability to accurately predict defaults is crucial. This thesis focuses on utilizing the SECOM dataset from the Machine Learning Repository (2008), which contains extensive information collected from numerous sensors monitoring the semiconductor manufacturing process. The objective is to manage this complex dataset effectively and develop predictive models with high accuracy using R or Python.

In high-dimensional datasets, such as the SECOM dataset, the presence of a large number of variables poses significant challenges. Effective data cleansing and feature selection or reduction methods are essential to enhance the performance of predictive models. This study aims to address these challenges by employing advanced data preprocessing techniques to reduce dimensionality and improve model performance. The ultimate goal is to create parsimonious models that retain strong explanatory power while minimizing the number of variables used.

The case study approach adopted in this thesis is guided by the methodologies outlined in the works of Munirathinam and Ramadoss (2016), and Kerdprasop and Kerdprasop (2003). These sources provide a structured framework for data analysis but require the application of comprehensive knowledge and critical thinking to achieve optimal results. By following these recommendations, the study seeks to reproduce some of the results and extend them to create models with superior predictive power.

1.1 Aim

The primary aim of this thesis is to learn how to handle complex datasets with an overwhelming number of variables and to write professional scripts in R or Python for predictive modeling. Specifically, the focus is on the SECOM dataset from the semiconductor manufacturing industry. The study aims to utilize data cleansing and feature selection or reduction methods to develop parsimonious predictive models. These

models should have strong explanatory power and the ability to accurately predict defaults in the production process.

1.2Methodology

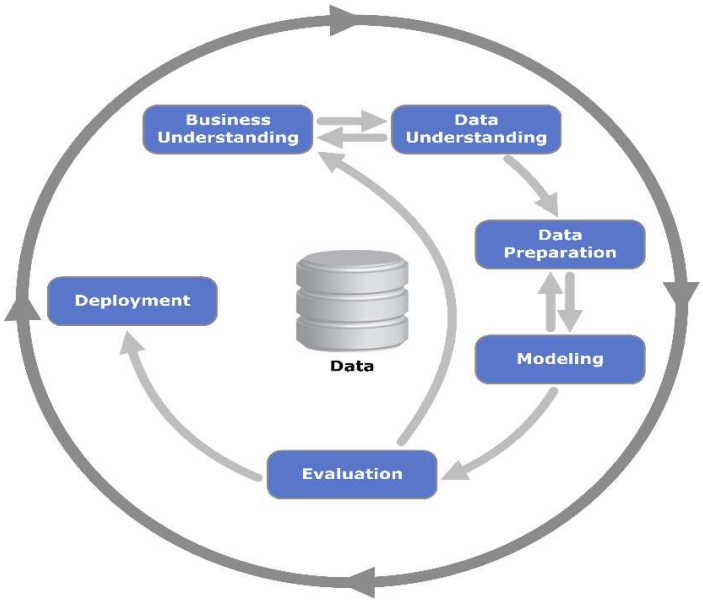

Figure 1 CrispDM Lifecycle

To achieve these objectives, this thesis employs the CRISP-DM (Cross-Industry Standard Process for Data Mining) methodology, a widely recognized framework for data mining projects. The CRISP-DM methodology involves six major phases:

1. **Business Understanding**: Gaining a clear understanding of the business objectives and requirements. For this study, it involves understanding the significance of predicting defaults in the semiconductor manufacturing process.

2. **Data Understanding**: Collecting and familiarizing oneself with the SECOM dataset. This phase includes initial data exploration and identifying data quality issues.

3. **Data Preparation**: Preprocessing the data through cleansing and feature selection or reduction methods. This step is crucial for handling the high dimensionality of the

SECOM dataset and involves techniques to reduce the number of variables while retaining essential information.

4. **Modeling**: Applying various data mining techniques to build predictive models. This phase involves selecting appropriate algorithms, training models, and fine-tuning them to achieve the best predictive performance.

5. **Evaluation**: Assessing the models to ensure they meet the business objectives. This involves evaluating the models' performance and their ability to accurately predict defaults in the production process.

6. **Deployment**: Implementing the model in a real-world setting where it can be tested and used. Although this thesis focuses on developing the models, the final step in a practical scenario would involve deploying the models for actual use.

This structured approach ensures a thorough understanding of the problem domain and a systematic process for developing effective predictive models. By following the CRISP-DM methodology, this thesis aims to provide a comprehensive analysis and develop models with superior predictive power for the semiconductor manufacturing industry.

Conclusion

This thesis addresses the critical challenge of managing and analysing complex datasets with numerous variables to predict defaults in the semiconductor manufacturing process. Through meticulous data preprocessing and the application of advanced data mining techniques, guided by the CRISP-DM methodology, the study aims to develop robust predictive models that enhance production quality and efficiency. The findings and methodologies presented here will serve as a valuable resource for researchers and practitioners in the field of data science and beyond.

2.Business Understanding

In the semiconductor manufacturing industry, particularly in wafer production, the production line is a highly complex and sensitive system monitored by a multitude of sensors. These sensors continuously collect data on various parameters of the manufacturing process to ensure the detection of defects as early as possible. Early detection is crucial because defects identified at later stages can be exceedingly expensive to correct, involving extensive rework or even scrapping of products. Wafer manufacturing, which involves intricate processes like photolithography, etching, and deposition, is especially sensitive to defects and variations in process parameters. By leveraging sensor data, predictive models can play a vital role in identifying potential faults early, allowing for timely interventions that save costs and enhance product quality. This proactive approach not only minimizes downtime and waste but also optimizes the overall efficiency of the manufacturing process. Furthermore, predictive modeling based on sensor data enables manufacturers to implement a more robust quality control system, ensuring that only high-quality wafers proceed to the next stages of chip production. The integration of advanced analytics and machine learning techniques in semiconductor and wafer manufacturing thus transforms raw data into actionable insights, driving continuous improvement and innovation in the industry.

3.Data Understanding

3.1 Data Understanding Phase of CRISP-DM

In this study, the Data Understanding phase was conducted using Python, leveraging popular libraries and frameworks such as Scikit-Learn, Matplotlib, and Pandas. These tools were instrumental in conducting descriptive data analysis, including examining missing values, volatility, duplicates, and outliers in the SECOM dataset. Insights gained from these analyses will inform subsequent data preparation and modeling phases, aiming to build robust predictive models for identifying defaults in semiconductor manufacturing. This approach ensures that the findings are based on rigorous data exploration techniques, facilitating informed decisions and actionable insights in line with the project's objectives.

3.2 Data Files

In this study, two data files were provided:

1. SECOM Data: This file contains 590 features (predictors/columns) and 1567 entries (wafer productions/rows).
2. Label Data: This file includes timestamps in date and time format and a target column with binary entries: 1 (fail) and -1 (pass). The distribution of the target column is 1463 passes and 104 fails.

3.3 **Descriptive Data Analysis**

To understand the data comprehensively, several analyses were conducted based on following methods.

1. Histogram of Missing Values
2. Histogram of Volatility
3. Heatmap
4. Duplicate Analysis
5. Outlier Analysis

3.4 Histogram of Missing Values

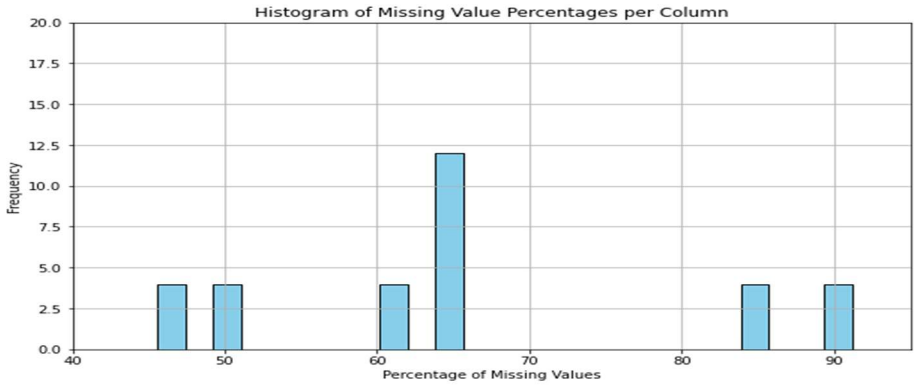

Figure 2 Histogram of Missing Values

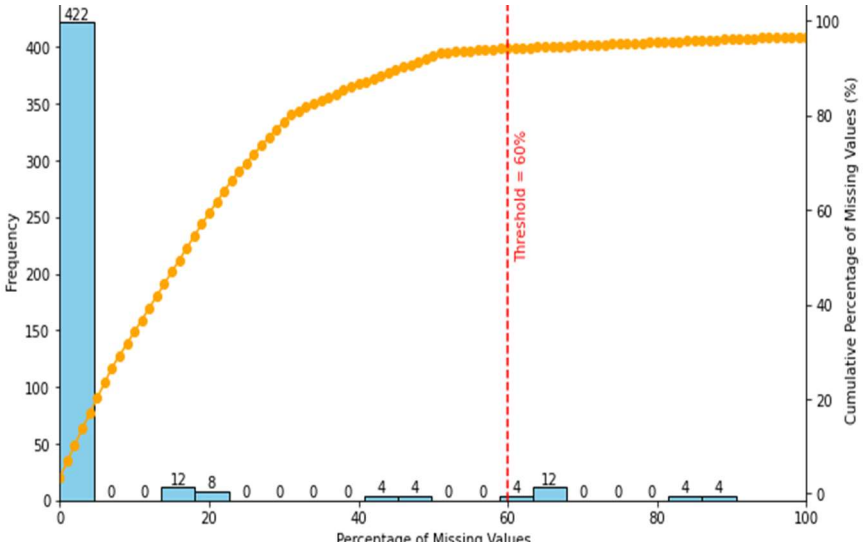

Figure 3 Pareto chart with percentage of missing values

Using Python and Jupyter Notebook, we loaded the data and created a histogram to examine the number of missing values in each column. Features with a high number of missing values can reduce the model's performance, as they introduce noise and do not contribute meaningful information. This histogram provides a quick assessment of data quality by highlighting the extent of missing data, which is essential for deciding how to handle these values—whether through imputation, deletion, or other techniques.

Understanding the distribution of missing values helps analyse their potential impact on the analysis or models. Visualizing missing values also helps identify patterns or randomness in the data, indicating whether the missing data is systematic or due to random errors. Further analysis using a Pareto chart deduced that a 60-65% threshold would be the optimal choice for handling missing values.

3.5. Histogram of Volatility

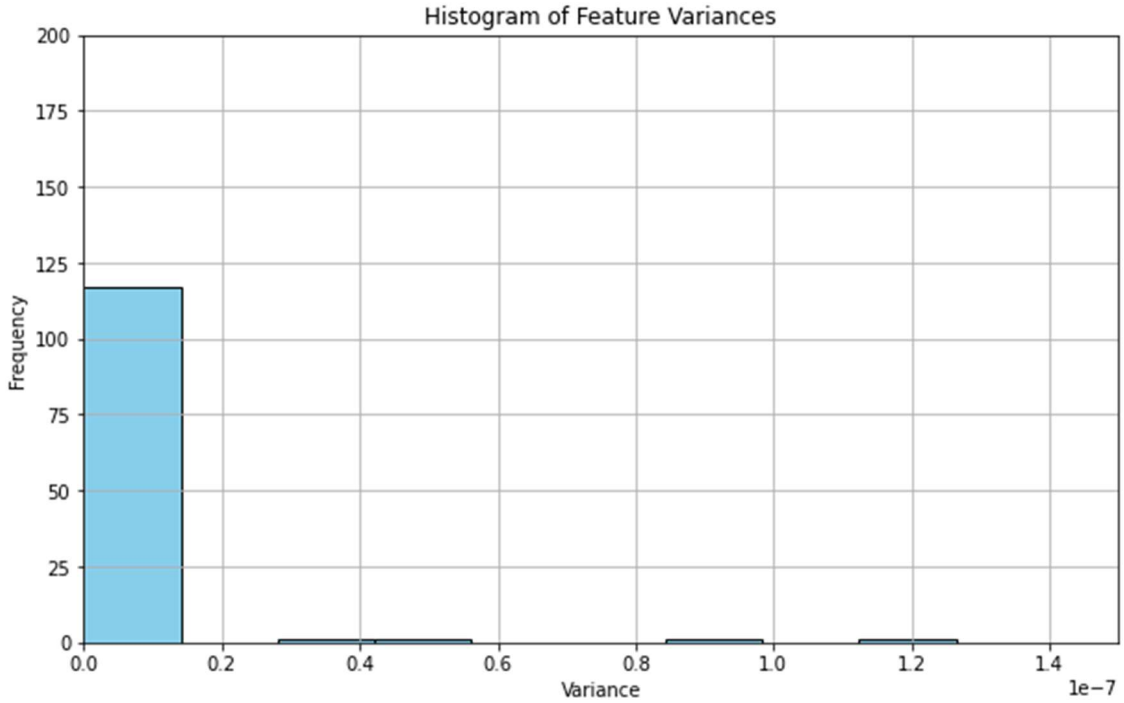

Figure 4 Histogram of Volatility of features

A histogram was plotted to measure the volatility of features. Features with high volatility, like those with numerous missing values, may not add significant value to the model. Volatility histograms can help identify outliers or extreme values in your data, which might require special handling or could indicate significant events or errors. These histograms can reveal whether most data points exhibit low volatility, high volatility, or if the distribution is skewed. Understanding the distribution of volatility is crucial for modeling, particularly in time series forecasting, financial modeling, and risk

management. It aids in selecting appropriate models and parameters, ensuring more robust and accurate analysis. After analysing histogram, 115 features were identified as having 0 variance.

3.6. Correlation Heatmap

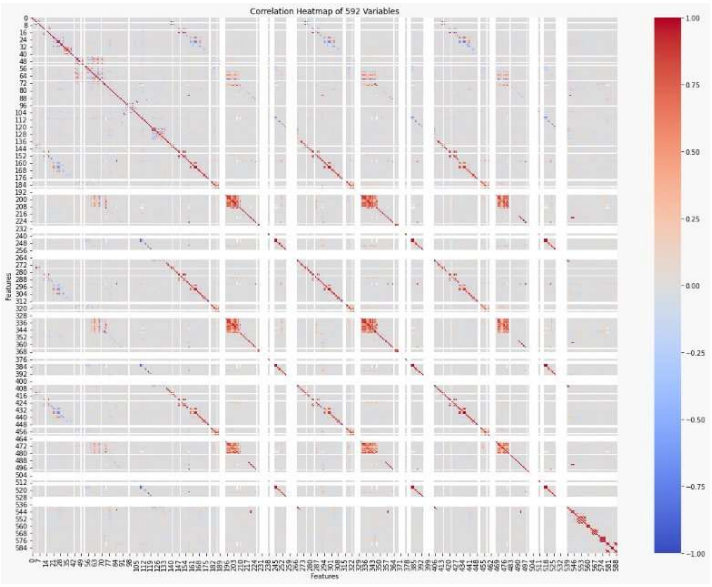

Figure 5 Correlation Heatmap

A heatmap was generated to visualize correlations between different features. Understanding the correlation structure helps in identifying multicollinearity and selecting the most informative features. However, as shown in Figure 4, the correlation heatmap did not showcase any relevant information, leading to the decision to drop this analysis technique.

3.7. Duplicate Analysis

The examination of the dataset for duplicates has highlighted critical issues affecting data integrity and analysis accuracy. Specifically, 31 duplicate timestamps were identified in label data, which could distort temporal analysis and lead to misleading insights. The solution is to merge both data files, Secom and Label. Additionally, 104 duplicate features were detected, all of which exhibited zero variance. The presence of these duplicate features suggests that they do not contribute any meaningful information and may

unnecessarily complicate the dataset. Addressing these duplicates is essential for ensuring the reliability of subsequent analyses and models. Removing redundant features will streamline the dataset, enhance its quality, and improve the robustness of any data-driven decisions or predictive models.

3.8 Outlier Analysis

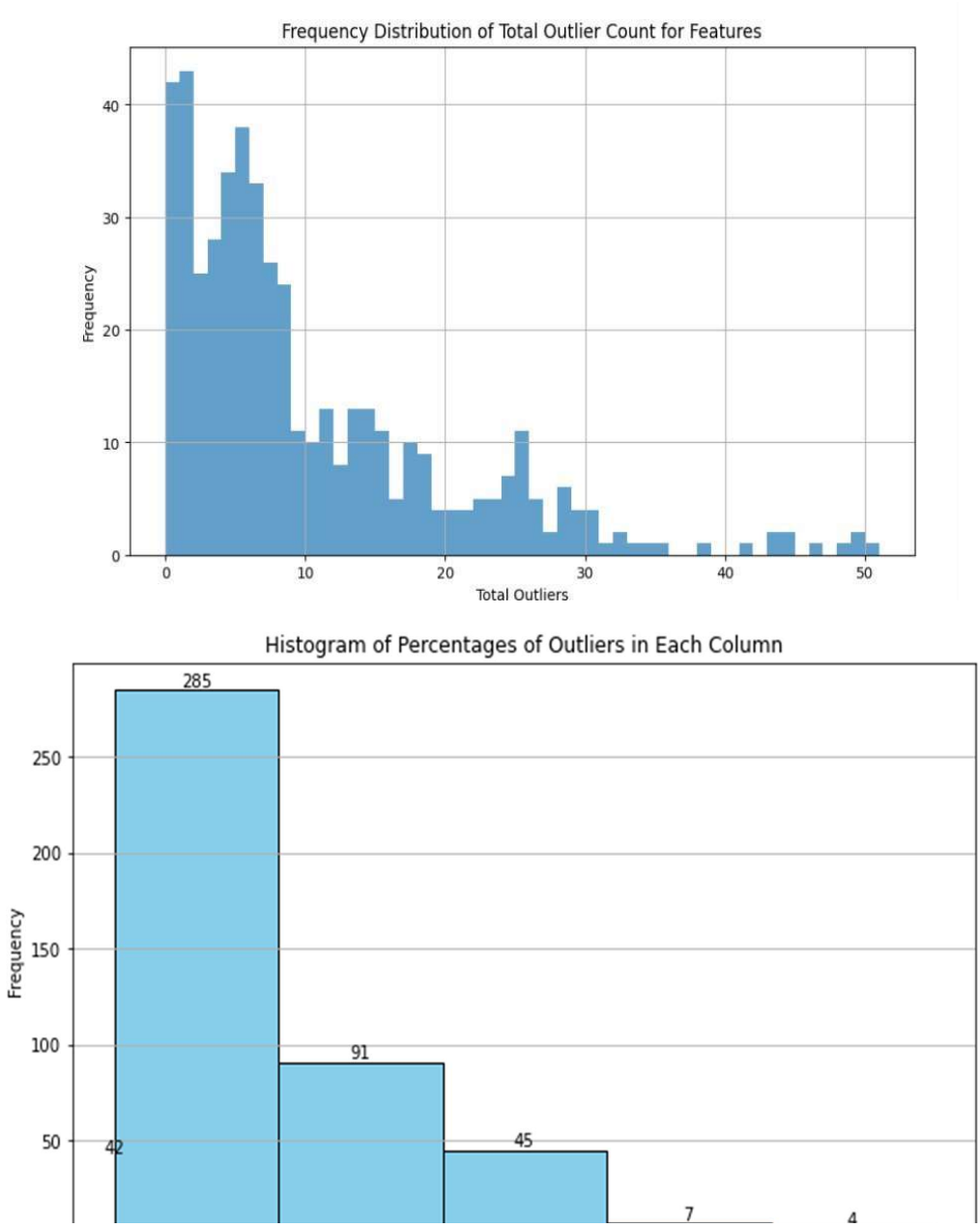

Figure 6 Outlier Frequency Distribution and Histogram

Observations from Figure 5 identified 432 features with outliers ranging from 50 to 1 (4.26% to 0.09%). The Z-score method was used to detect outliers, relying on normality assumptions (such as those used in PCA) and statistical measures. The impact of outliers includes potential violations of normality assumptions required for methods like PCA, skewing measures of central tendency and dispersion, and causing sensitivity in ML algorithms like KNN, K-Means, and SVM, which can result in misleading predictions and overfitting. Possible action points include taking no action if the proportion of outliers is negligible, modifying the outliers using statistical boundaries (e.g., replacing with mean or median), or removing the outliers from the dataset, though this could increase missing values and potentially result in the loss of important information.

The data understanding phase provided crucial insights into the SECOM dataset, after performing EDA (Exploratory data analysis), highlights data quality issues such as missing values, duplicates, and outliers. These findings will guide the subsequent data preparation and modeling phases, ensuring that the predictive models built are robust and reliable. By addressing the identified data quality issues, we aim to enhance the accuracy and performance of the models, ultimately achieving the business objectives of reducing production costs, improving product quality, and increasing efficiency in semiconductor manufacturing.

4.Data Preparation

In the next step of the Cross-industry standard process for data mining data preparation consist of the most pivotal role in the as This section offers a thorough examination of this stage, encompassing the gathering, depiction, investigation, and validation of data quality.

Within the semiconductor manufacturing sector, where production systems are complex and extremely delicate, utilising sensor data for predictive modelling is crucial for detecting defects early on. This section will delve into the intricate process of comprehending the data, laying the groundwork for subsequent modelling and analysis.

When developing and evaluating machine learning models, it's crucial to split the available dataset into training and testing subsets. This practice serves several important purposes. First, it enables performance estimation on unseen data, allowing us to evaluate the model's performance on data it hasn't seen before and providing a realistic estimate of its predictive power in real-world scenarios. Second, it helps avoid overfitting by ensuring the model's evaluation is based on its ability to generalize, not just memorize the training data, thus leading to better performance on unseen data. Lastly, it reduces evaluation bias, as using a separate test set ensures the performance metrics are not overly optimistic, providing a more accurate and reliable assessment of the model's performance.

In case of SECOM We divide the dataset into a training set (75%) and a test set (25%). This ratio is chosen to ensure that we have enough data to effectively train the model while also having a sufficient amount to evaluate its performance accurately. The model is trained using the training set, where it learns the underlying patterns and relationships in the data. After training, the model is evaluated on the test set. This assessment provides an indication of how well the model generalizes to new, unseen data. Ensuring Same Proportion of Pass and Fail Cases Using Stratified Sampling: To maintain the integrity of the dataset's class distribution as we are dealing with highly imbalanced dataset, we use stratified sampling to split the data. This means the training and test sets will have the same proportion of pass and fail cases as the original dataset. Stratified sampling ensures

that both subsets represent the original dataset's distribution, avoiding any class imbalance that could bias the model's performance evaluation.

Table 1 Details of Splitting

	Size	**Pass Cases**	**Fail Cases**	**Pass to Fail Ratio**
Total Dataset	1567	1462	103	14:1
Training Set	1175	1096	78	14:1
Test Set	392	365	25	14:1

Using stratified sampling ensures that both the training and testing sets maintain the same pass to fail ratio as the original dataset, which is 14:1. This method helps in maintaining the dataset's class distribution, allowing the model to learn, and be evaluated accurately without bias towards any class.

By using both a standard train-test stratified split, we can achieve reliable performance estimation, avoid overfitting, reduce evaluation bias, and improve the robustness of our model's evaluation. These methods ensure that the model is trained and tested on representative samples, leading to more accurate and generalizable performance metrics and 6.6% fail cases were constant all over.

In the process of data preparation, several measures were taken to optimize computational efficiency and model accuracy. Duplicate entries were eliminated to ensure unique data points, merging the datasets to have redundant entries. Features with zero variance, which add no value to the model, were excluded, reducing dimensionality by 115 features. To prevent overfitting, features like timestamp values, which can introduce period-specific noise, were removed. Additionally, a threshold for missing values was set, eliminating

features exceeding this limit while imputing those within the acceptable range. These strategies ensure a clean, pertinent, and manageable dataset, enhancing the efficiency and performance of predictive models.

4.1Outlier Handling

In semiconductor manufacturing, numerous sensors generate extensive data, where outliers can impact predictive models and indicate process anomalies. Addressing outliers involves several approaches. Retaining outliers preserves all original data but may reduce model reliability due to skewed results. Replacing outliers with values at the 3σ boundaries can mitigate their impact but might distort data distribution if outliers are numerous or not well-separated. A more advanced technique involves removing outliers and imputing their values using imputation methods like k-Nearest Neighbors (kNN), which maintains dataset integrity and skewness to a certain level but can be computationally expensive.

Based on Figure 6, for Feature 36 (has highest outlier values), The histogram shows the overall implication of the different approaches in which Keeping the outliers as they are can lead to a highly skewed distribution and Replacing outliers with values within 3 standard deviations significantly alters the data distribution and may not be the most effective method. On the other hand, Imputing outliers using the KNN method results in a distribution that closely can follow a normal curve, suggesting it is a more balanced approach and maintaining the original skewness of the data to a certain level for handling outliers. It is crucial to perform imputation after addressing outliers because certain outlier treatment methods, such as method 3, can introduce additional missing values into the dataset. By handling outliers first, we ensure that imputation is conducted on a more accurate dataset, thereby minimizing the risk of exacerbating missing data issues and achieving more reliable results.

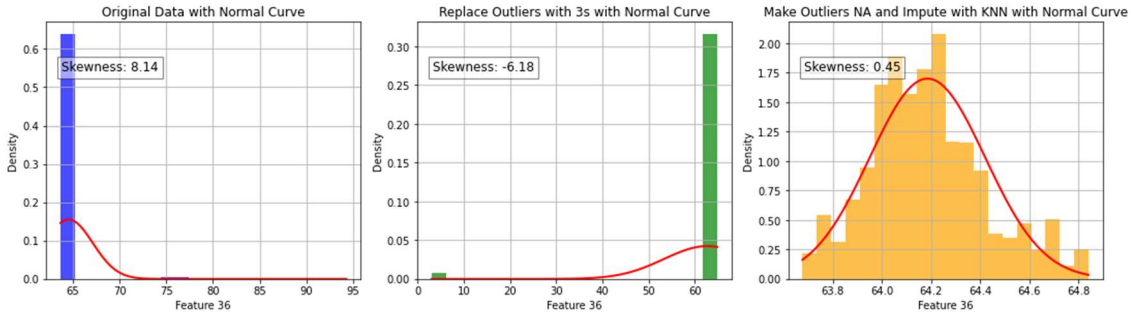

Figure 7 Outlier handling

4.2Approaches handling to missing values:

Handling missing values is also crucial, with imputation methods addressing Missing at Random (MAR), Missing Completely at Random (MCAR), and Missing Not at Random (MNAR) scenarios. Effective imputation maintains data accuracy and reliability, essential for precise analytics and optimal performance in semiconductor manufacturing.

The semiconductor manufacturing sector heavily depends on precise and accurate data for overseeing and managing intricate production procedures. The significance of imputation methods in this field can be underscored through multiple critical facets. Diverse imputation methods are crucial in managing absent data, guaranteeing the precision and dependability of datasets in the semiconductor sector. By upholding data integrity, improving predictive maintenance, refining production standards, cutting down expenses, and backing advanced analytics, these methods are essential to the sector's endeavours to attain elevated efficiency and product excellence.

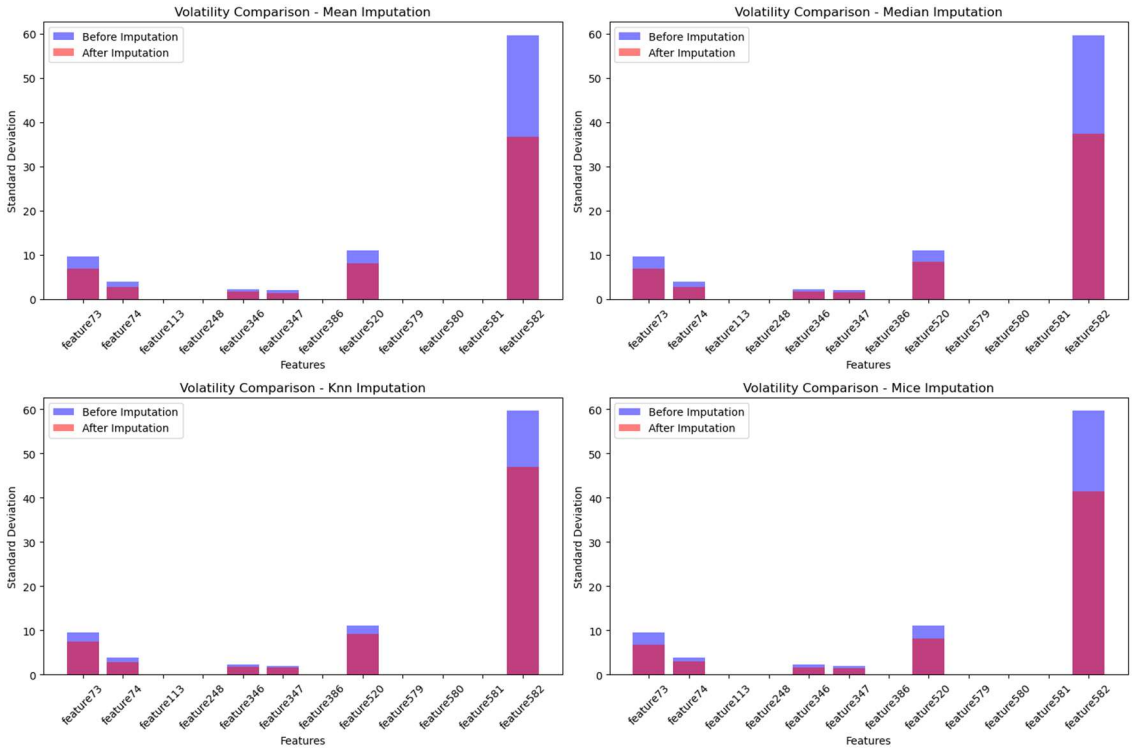

Figure 8 Volatility Comparison for Imputation

4.2.1 Mean Imputation:

Mean imputation replaces missing values in a dataset with the feature's mean from available data points. This method is simple and retains all data but can introduce bias. As shown in figure 8, standard deviation changes moderately across most features after imputation (red bars), with a noticeable reduction in the last feature. (feature 12).

4.2.2 Median imputation:

Median imputation replaces missing values with the feature's median value. It is easy to implement and less sensitive to outliers but can reduce variance and introduce bias. In the graph, changes in standard deviation are less pronounced compared to KNN imputation, with some features showing little to no change.

4.2.3 Regression Imputation:

Regression imputation predicts missing data using a regression model, leveraging relationships between the dependent variable and other independent variables in the dataset. While effective, it risks overfitting if relationships between features are strong and does not account for volatility.

4.2.4 KNN imputation:

K-Nearest Neighbors (KNN) imputation is a sophisticated technique employed to handle missing data by identifying the K most similar instances (neighbours) to the instance with missing values and using their values to estimate the missing data. This method, a type of hot deck imputation, leverages multivariate information and preserves the intrinsic relationships within the dataset, offering a robust approach to data imputation. However, KNN imputation can be computationally intensive, particularly when applied to large datasets, due to the complexity of identifying and processing the nearest neighbours.

As illustrated in the graph below, the standard deviation of several features shows notable changes post-imputation. This is especially evident in the figure, where a substantial decrease in standard deviation is observed. This indicates that KNN imputation has significantly affected the variability, highlighting its impact on the dataset's statistical properties.

4.2.5 MICE imputation:

MICE (Multiple Imputation by Chained Equations) is a complex iterative method for imputing multivariate missing data. It generates multiple imputations under the assumption that data is Missing At Random (MAR). Compared to simpler methods, MICE is more intricate to implement and interpret. In the graph below, MICE Imputation significantly alters the standard deviation of several features, notably decreasing it for feature12.

As the KNN imputation is crucial for the semiconductor industry as it effectively addresses missing data from various sensors and processes, preserving the relationships between variables that are essential for quality control and process optimization. Its ability to adapt to local patterns in the data enhances predictive accuracy, which is vital

in maintaining product consistency. Additionally, a lower change in volatility in the data is important in data mining because it leads to more stable and reliable predictions, allowing for clearer identification of trends and patterns, ultimately supporting better decision-making and more robust model performance in complex manufacturing environments hence, KNN imputation is considered for the further imputation.

4.3 Optimal parameters -

Optimal parameters are essential in the semiconductor industry for ensuring high-quality production, maximising output efficiency, and maintaining precision in manufacturing processes. In semiconductor fabrication, even minute deviations from optimal parameters can significantly impact device behaviour. Optimal parameters are crucial for process control, productivity enhancement, device performance, quality assurance, cost reduction, and driving innovation. Optimal parameters in data mining refer to the best configuration of hyperparameters for a given model or algorithm that achieves the highest performance on a specific task. These parameters are crucial for maximising the effectiveness of data mining models and are typically found through a process of hyperparameter optimization. According to the optimal parameter method, we have researched various combinations of techniques, which will be detailed in the following sections.

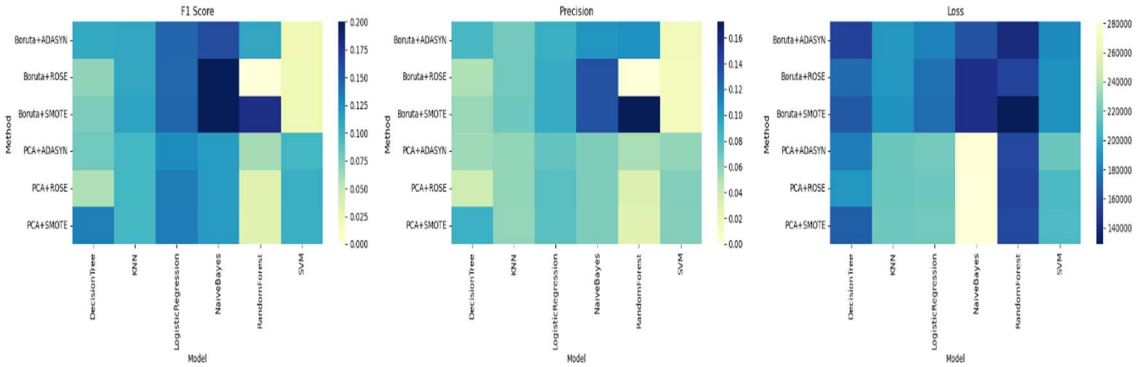

Figure 9 Heatmap for multiple combination comparison

4.4 Feature selection and Feature reduction:

Feature selection and reduction are critical processes in data preprocessing that aim to improve model performance by identifying and retaining only the most relevant features while eliminating redundant or irrelevant ones. These techniques are applied after imputation to ensure that the dataset used for analysis or modelling is both complete and optimised, thereby enhancing the accuracy and efficiency of subsequent analyses. In the semiconductor industry, where data from various sensors and processes can be voluminous and complex, effective feature selection and reduction are essential for improving output, enhancing quality control, and facilitating more efficient manufacturing processes. By focusing on the most significant features, companies can reduce computational costs, improve model interpretability, and make more informfigureed decisions based on the insights derived from their data.

4.5 Feature selection:

Feature selection is the process of identifying and selecting a subset of relevant features from a larger dataset to improve model performance by reducing overfitting, enhancing interpretability, and decreasing computational costs. Boruta is a superior feature selection method because it uses a random forest classifier to evaluate feature importance against randomly permuted versions of the features, ensuring that only statistically significant features are retained. This robust approach accounts for interactions between features, making it particularly effective in complex datasets where variable relationships are crucial for accurate predictions.

4.5.1 Boruta:

Boruta is an advanced feature selection method that enhances model performance by utilising a random forest approach to evaluate both original and shadow features, effectively capturing complex relationships within the data. It is particularly important in the semiconductor industry as it prevents the loss of crucial information by assessing feature significance, ensuring that only relevant features are retained for predicting target variables. Unlike traditional methods, Boruta does not require a direct strong correlation

with the target variable, making it versatile for various datasets. Additionally, it adeptly handles multicollinearity and non-linear relationships, making it a powerful tool for identifying and ranking important features in datasets characterised by intricate interactions, which is essential for optimising manufacturing processes and improving decision-making in the semiconductor sector.

Based on the optimal parameter analysis that was conducted as mentioned in the previous section, suggests that Boruta is a suitable method for the further processes as it indicates a lower loss cost. After implementing this, the sorted Boruta feature rankings highlight the importance of various features identified through the Boruta algorithm. Features such as feature60, feature65, feature66, feature342, feature351, feature478, feature540, and feature563 have been assigned the highest ranking of 1, indicating their critical importance in the model. These top-ranked features are considered highly relevant for predicting target variables and capturing complex relationships within the data. Following the top-tier features, several others such as feature157, feature268, feature292, feature427, and feature430 are ranked 2, denoting their substantial significance but slightly lower relevance compared to the top-ranked features. Features like feature154 and feature206 fall into ranks 3 and 4, respectively, showcasing their moderate importance in the model. Less critical features, such as feature153 and feature426, are ranked 5 and 6, respectively. Finally, feature171 is ranked 7, indicating it has the least influence among the listed features but still retains some relevance in the model.

4.6 Feature reduction:

Feature reduction is the process of decreasing the number of input variables in a dataset to simplify models, enhance computational efficiency, and mitigate overfitting and noise. Principal Component Analysis (PCA) is a leading method in this domain, as it transforms original features into a new set of uncorrelated variables (principal components) that capture the maximum variance in the data. This capability allows PCA to retain significant information while effectively reducing dimensionality, thus removing multicollinearity, and improving model performance. Additionally, PCA enables faster computations, making it particularly valuable for high-dimensional datasets.

4.6.1 PCA:

Principal Component Analysis (PCA) is a statistical technique used for dimensionality reduction that transforms a dataset into a new coordinate system, where the greatest variance in the data is captured by the first few principal components. This process simplifies the dataset by reducing the number of features while retaining the most important information, making it easier to analyse and visualise complex data structures. While it is a powerful tool for feature reduction, it has some limitations. One significant drawback is the loss of interpretability, as the principal components are linear combinations of the original features, making it difficult to understand their individual contributions. It also assumes linear relationships among features, which may not hold true in all datasets. Additionally, it requires mean centering of the data, which can be a limitation in certain contexts. It is best suited for unsupervised learning scenarios where the target variable is not the primary focus, as its goal is to reduce dimensionality without considering the target variable.

The optimal parameter analysis carried as mentioned above (figure 8) shows significantly higher loss cost than Boruta. This higher loss cost indicates that PCA may lead to less accurate predictions and may not be economically reliable compared to other methods such as Boruta.

The analysis presented below in the scree plot shows no clear elbow or break point where

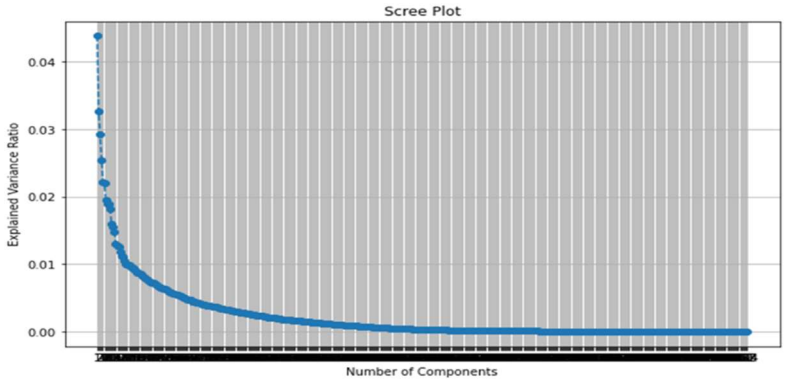

Figure 10 Scree plot for PCA

the eigenvalues begin to level off, indicating uncertainty in the number of components to retain. The KMO statistic, which measures sampling adequacy, is 0.65 after removing highly collinear features; however, multicollinearity in the original data can lead to computational issues, potentially resulting in NaN values in the KMO statistic. This suggests that important features may be discarded during the process. Overall, while PCA is somewhat suitable for factor analysis, it is not the ideal method due to these limitations.

4.7 Balancing:

Balancing adjusts the class distribution in a dataset to ensure equitable representation, which is crucial after feature selection, especially with imbalanced cases like 6.6% fail and 93.3% pass. In the semiconductor industry, balancing improves model performance by creating a more accurate training dataset and preventing overfitting, thus avoiding bias towards the majority class. This leads to better decision-making and reliable outcomes in cost-sensitive applications, enhancing the model's generalization to unseen data.

4.7.1 Oversampling

To address class imbalance in the overall sample distribution, which includes Feature 1 and Feature 2 with 1,097 majority (pass) and 1,097 minority (fail) cases, we implemented an Over-Sampling approach. This technique duplicates instances of the minority class to create a balanced dataset. While easy to implement and potentially improving model performance, Over-Sampling carries a high risk of overfitting, as duplication may lead to models that do not generalize well to real-world scenarios. Consequently, improved

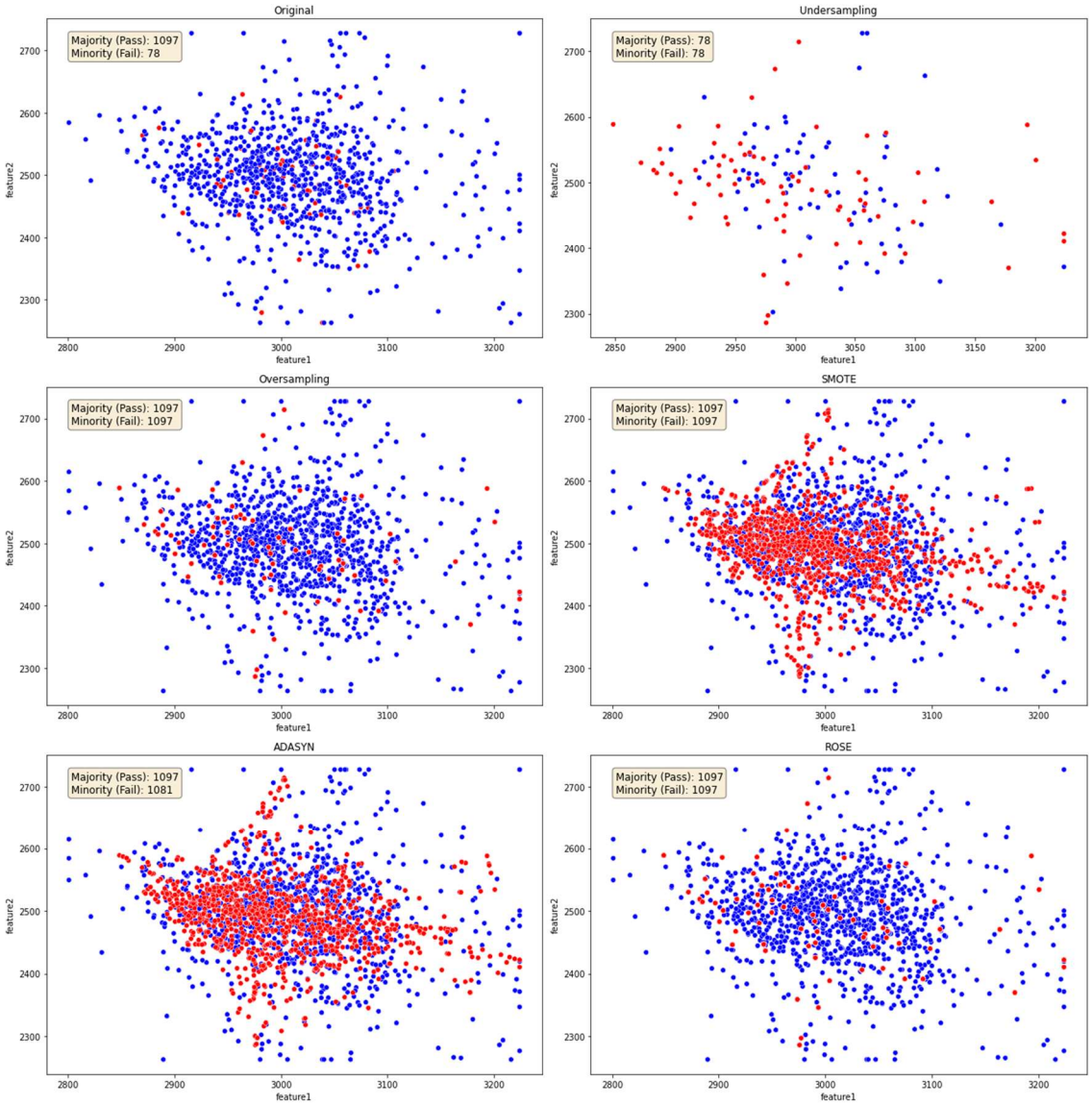

Figure 11 Sampling

accuracy may not reflect true predictive performance. Therefore, despite its advantages in specific situations, Over-Sampling is considered a lower priority in the overall modelling strategy due to its limitations in accurately representing realistic data distributions.

4.7.2Undersampling-

To address class imbalance in the sample distribution, which includes Feature 1 and Feature 2 with 78 majority (pass) and 78 minority (fail) cases, we employed the Under-sampling approach. This method removes instances from the majority class to achieve a balanced dataset. While it reduces dataset size and improves computational efficiency, it risks losing important information from the majority class, leading to underfitting and poor generalization. Despite satisfactory model accuracy, it may not reflect real-world data complexities. Therefore, under sampling is considered a lower priority in the overall modelling strategy due to the potential loss of critical information.

4.7.3Approaches to deal with imbalance data-

Smote -

1)The SMOTE (Synthetic Minority Over-sampling Technique) approach generates synthetic samples by creating new instances between existing minority class instances and their K nearest neighbors, resulting in a balanced dataset. This method is advantageous due to its uniformity and flexibility; however, it lacks an adaptive focus, which can limit its effectiveness in certain contexts. The impact of SMOTE is often assessed through cross-validation, yielding estimated results that can enhance model performance. Given the low cost associated with its implementation and the overall sample distribution, which includes 1,097 majority (pass) cases and 1,097 minority (fail) cases. As per the image (Figure 8) Smote shows low loss cost thus, the SMOTE approach is considered a high priority for addressing class imbalance in the dataset.

2)Rose-

The ROSE (Random Over-Sampling Examples) approach generates new synthetic data points by adding random noise to existing instances within the minority class, employing a smoothed bootstrap methodology. This technique helps maintain the underlying distribution of the data while reducing the risk of overfitting compared to simple duplication methods. However, it can introduce noise and requires careful parameter

tuning to optimise performance. The effect of using ROSE is to reduce bias in the model, making it a medium-priority option for addressing class imbalance. Within the overall sample distribution, which includes feature 1 and feature 2 with 1,097 majority (pass) cases and 1,097 minority (fail) cases, ROSE provides a balanced dataset that enhances model robustness and reliability.

3)ADASYN-

The ADASYN (Adaptive Synthetic Sampling) approach focuses on generating synthetic samples for more difficult-to-learn instances within the minority class, thereby creating a better-balanced dataset. While this method enhances the representation of challenging cases, it can introduce noise, and the newly created instances may not be identical to the original data points. This variability can sometimes impact the precision of model evaluation. Given these considerations, the ADASYN approach is assigned a low priority for addressing class imbalance. Within the overall sample distribution, which includes feature 1 and feature 2 with 1,097 majority (pass) cases and 1,081 minority (fail) cases, ADASYN aims to improve model training by focusing on less easily classified instances.

4.8 Decision hierarchy

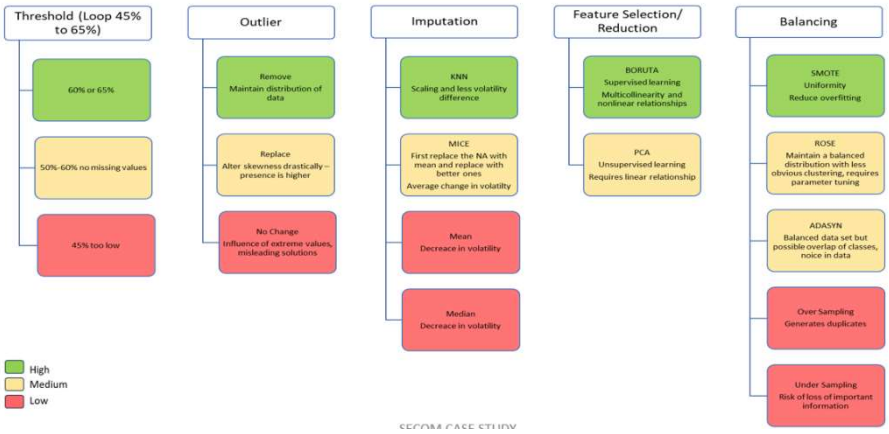

Figure 12 Decision hierarchy

To summarize, the decision hierarchy outlines the critical steps for model building and evaluation. The threshold for these steps was set between 60% and 65%. Initially, outliers were removed, followed by the application of KNN for computation. Feature selection was performed using the Boruta method, and SMOTE analysis was employed to balance the data, which were prioritized in the process. Based on these approaches, Model 1 and the customized Boruta model demonstrated the lowest loss cost, indicating their effectiveness in the model-building process.

4.9 Scaling:

Feature scaling is crucial in the semiconductor industry, particularly after model building, as it ensures uniformity across different parameters and improves the performance of algorithms like SVM and KNN, which are sensitive to feature scales. As KNN is a distant based approach, by normalising features to a common range, scaling reduces biases that can arise from variables with larger magnitudes dominating the model's decision-making process. This is especially important in semiconductor manufacturing, where various measurements and parameters can have vastly different scales. Without scaling, features with higher ranges are more likely to be given undue importance by the model, potentially leading to suboptimal or inaccurate predictions. The choice of min-max scaling method in this context is particularly apt, as it effectively brings all features to a comparable scale while preserving the relative relationships between data points, which is critical for the precise control and optimization required in semiconductor production processes.

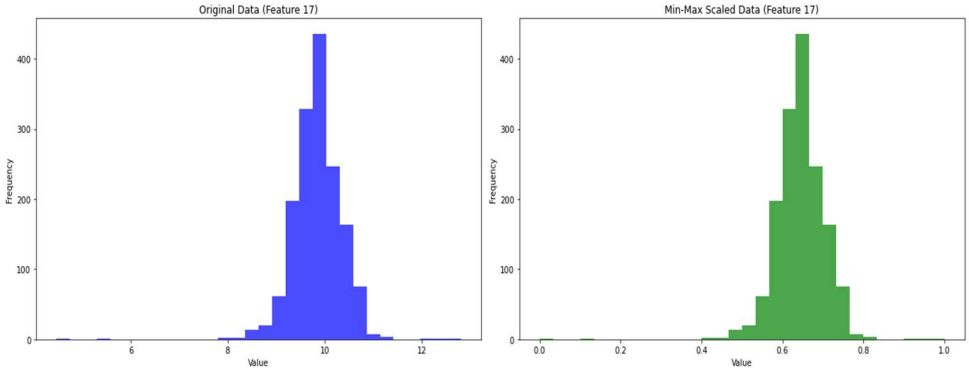

Figure 13 Min Max Scaling

5. Modeling & Evaluation

Utilizing the iterative nature of CRISP-DM, we build our models in three iterations, alternating between the modeling and evaluation phases.

5.1 Iteration 1

Based on the decision hierarchy, the Random Forest model was built using high-priority decisions and evaluated through confusion matrix analysis, loss cost, and accuracy metrics. This approach ensures that key factors influencing model performance are carefully assessed, aligning with strategic priorities for optimizing the model's effectiveness and reliability.

A confusion matrix is a table used to evaluate the performance of a classification model by comparing predicted outcomes against actual outcomes. It typically contains four key values: True Positives (TP), True Negatives (TN), False Positives (FP), and False Negatives (FN). The loss cost is a metric that quantifies the financial or operational impact of classification errors, considering the costs associated with false positives and false negatives.

A loss cost calculation framework with 1,000 false positives and 5,000 false negatives to capture the impact of misclassifications on operational efficiency and financial outcomes in the semiconductor industry. False positives, indicating non-defective wafers flagged as defective, can lead to unnecessary rework and higher costs. Conversely, false negatives represent defective wafers going undetected, causing severe quality issues and customer dissatisfaction. By assigning greater weight to false negatives, main focus is to highlight the critical need for accurate defect detection, ensuring product quality and optimizing model performance while balancing the cost implications of misclassifications.

For the first iterations, as mentioned in section decision hierarchy, for building the model 1 and Boruta no 1 feature for the customized model, the confusion matrix for Model 1 shows 351 true positives, 4 true negatives, 15 false positives, and 22 false negatives. For the Custom Boruta Model, there are 348 true positives, 6 true negatives, 18 false positives, and 20 false negatives. Model 1 has a total loss cost of $125,000, while the Custom Boruta Model shows a slightly lower loss cost of $124,000.

These metrics indicate that the Custom Boruta Model performs marginally better in terms of reducing costly misclassifications. The FP price represents the cost of incorrectly classifying a negative case as positive, while the FN price is the cost of missing a positive case. Overall, both models demonstrate high accuracy, but the Custom Boruta Model shows a slight edge in minimizing expensive errors.

Model 1 – Confusion matrix - {351 15 Loss cost: 125000
22 4}

Boruta Customized model – Confusion matrix - {345 18 Loss cost: 124000
20 6}

To ensure the correctness of our data processing steps, utilization of a pipeline approach to identify the most cost-effective predictive model is being used. This method systematically preprocesses the data and integrates various models into a cohesive pipeline. Each model was evaluated using a custom cost-sensitive loss function tailored to our specific business needs. Remarkably, the results from this pipeline approach aligned with our predefined decision hierarchy, validating the methodology's effectiveness. This consistency reinforces our confidence in the selected model's ability to minimize costs and enhance predictive accuracy, thereby supporting better decision-making and strategic planning.

As per this approach and previously mentioned section, optimal parameter which signifies smote+boruta method custom boruta model has been chosen for the further model building.

5.2 Iteration 2

Based on the models derived from phase 1 and according to section optimal parameter, for the next steps new model techniques Support vector classification (SVC), GaussianNB, Random Forest classifier is considered for further model building and evaluation.

5.2.1 Hyperparameter tuning -

Hyperparameter tuning is a critical process in data mining that involves finding the best configuration of hyperparameters to maximise model performance. In the semiconductor industry, hyperparameter tuning is essential for optimising complex manufacturing processes, ensuring high-quality control, and enhancing overall efficiency. After scaling, hyperparameter tuning becomes particularly important because scaling changes the relative importance of features, necessitating a re-optimization of parameters to ensure the model performs optimally.

5.2.2Grid search-

Grid search for hyperparameter tuning is beneficial due to its automation and scalability, allowing for systematic exploration of the hyperparameter space. This method enhances process optimization by identifying the best parameter settings, improves quality control by ensuring consistent model performance, and increases efficiency by reducing the need for manual tuning. Grid search's ability to parallelize tasks further supports large-scale semiconductor manufacturing environments, making it a valuable tool for maintaining high standards and competitiveness. As per the optimal parameter (Figure 8) when the scores arechecked for the model SVC, GaussianNB, Random Forest. After the hyperparameter tuning in the considered models for GaussianNB, Random Forest the F1 score, accuracy or the loss cost shows no differentiation after the tuning. On the other hand SVC shows slight rise in Accuracy and loss cost but slight dip in F1 score. This

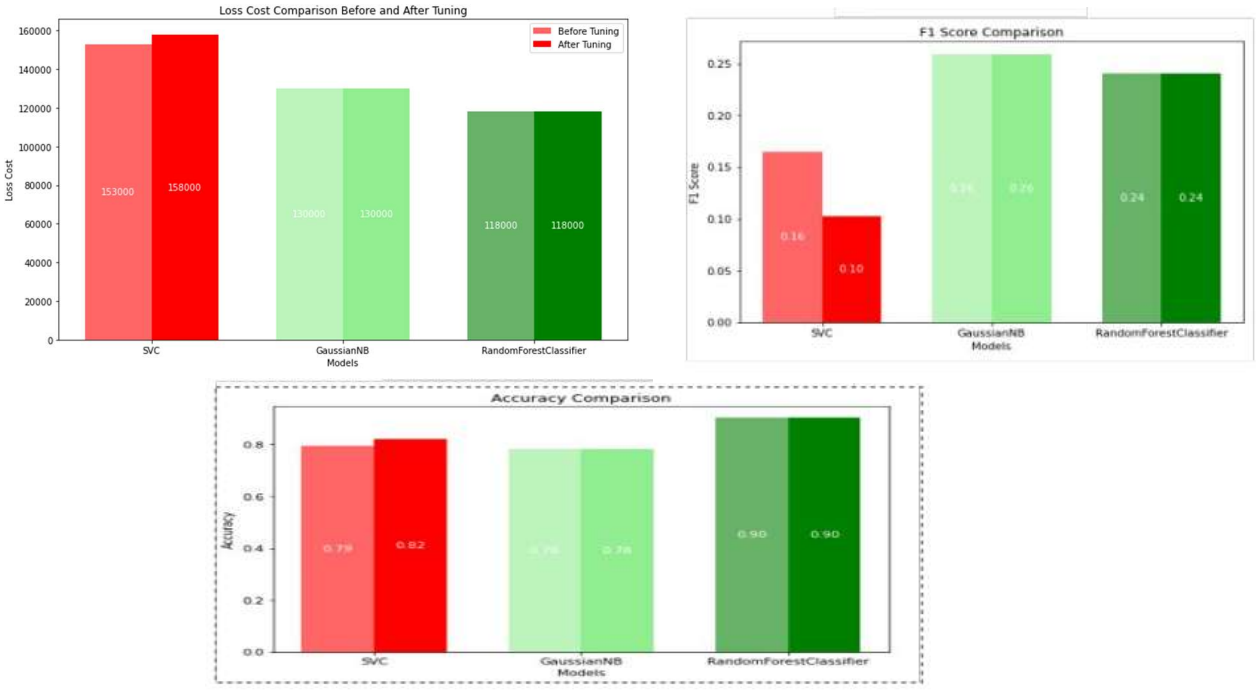

Figure 14 Hyperparameters

pattern often occurs when dealing with imbalanced datasets. To delve deeper into the performance and behaviour of these models, let's examine the learning curves and model complexity. As per the lines graphs given below, the scores given as before and after tuning shows overfitting where it shows larger gap in between scores for random forest although for Support vector classification (here after will be referred as 'SVC') and Guassian Naïve bayse (here after referred as 'NB') models the gap is less the training and validation score is less. Hence, models are not optimum.

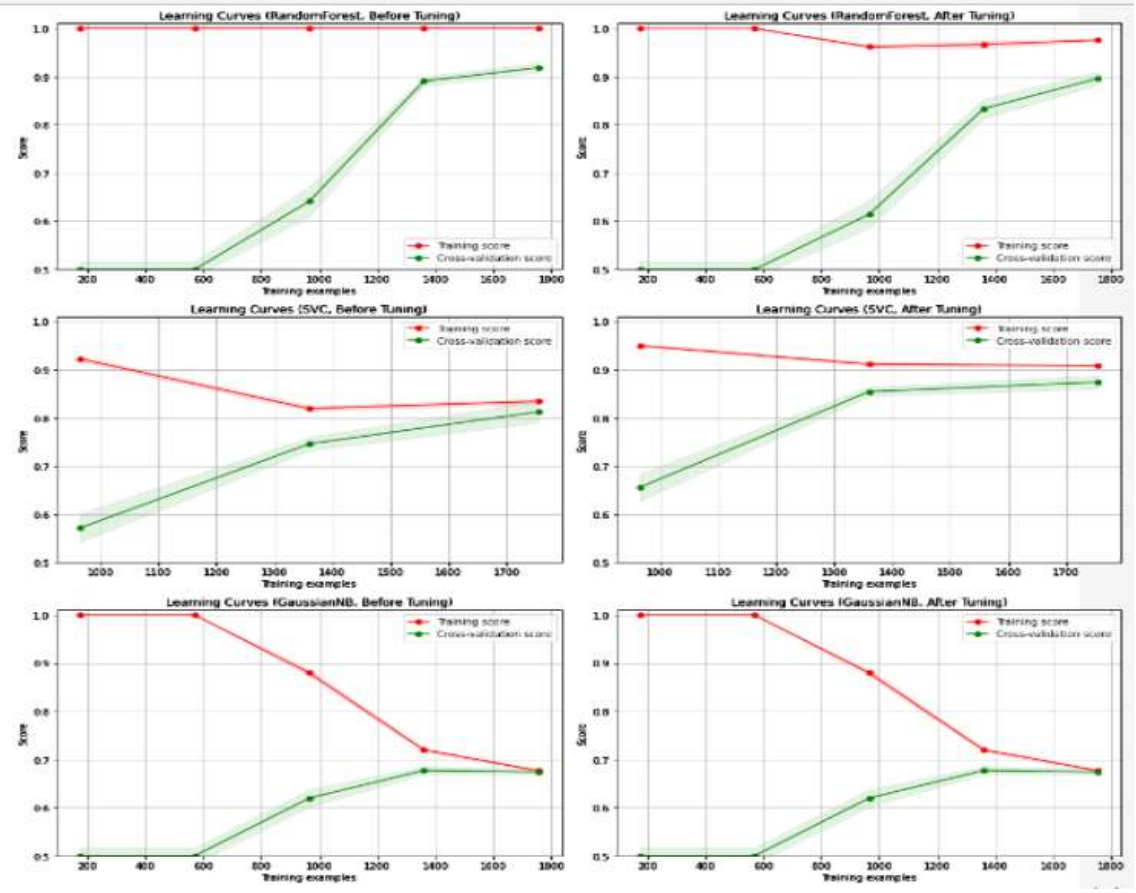

Figure 15 Learning Curves

In order to mitigate the overfitting evident in the learning curves, it is necessary to investigate alternative strategies and revisit CRISP DM model and previous decisions. Upon reviewing the progress made thus far, it is advisable to reconsider the feature timestamp for feature engineering.

5.2.3 Feature engineering -

Feature engineering is the process of creating new, more informative variables from existing data to improve model performance. When models are overfitting, feature engineering can help by reducing complexity and emphasising more relevant patterns in the data. This process can lead to models that perform well on both training and unseen data, effectively mitigating overfitting while maintaining predictive power.

In feature engineering, the timestamp feature can be determined by analysing the time intervals between each wafer production. This can be achieved by monitoring the production flow and calculating the difference from the first wafer production to second and first wafer production to third so on until the last wafer, by dividing first date and time and then dividing hours, minutes, and seconds. This process will ultimately help in understanding the change in wafer production which can affect the next wafer production. We named this feature elapsed time as it calculates the interval between each wafer production. This elapsed time provides new features which can be considered for further evaluation. The image below illustrates this feature engineering process. The significance of tracking elapsed time is that any delay in the production of a single wafer or issues with sensors can disrupt the production of subsequent wafers.

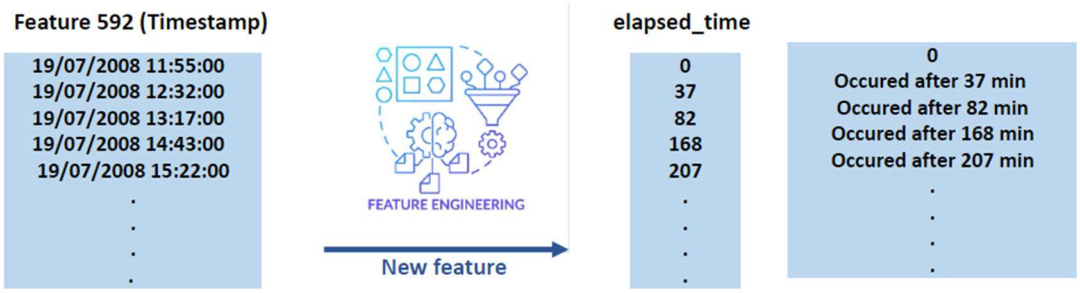

Figure 16 Feature Engineering

5.3 Iteration 3

This was our final iteration where we got our final modelmodel.

5.3.1 Modeling

Three models were developed with specific preprocessing steps and algorithms. Model 1 involved removing 60% of outliers, applying KNN imputation, conducting Boruta feature selection post-feature engineering, using SMOTE for class balancing, and applying min-max scaling, with a Random Forest algorithm. Model 2 had similar preprocessing steps:

60% outlier removal, KNN imputation, Boruta feature selection, SMOTE for class balancing, and min-max scaling, also utilizing the Random Forest algorithm. Model 3 followed the same preprocessing steps—60% outlier removal, KNN imputation, Boruta feature selection, SMOTE, and min-max scaling—but employed the Naïve Bayes algorithm.

When evaluating the performance of above models, we considered multiple metrics to gain a comprehensive understanding of their effectiveness. In this analysis, we evaluate three models based on their F1 score, loss cost, and confusion matrix (True Negatives (TN), False Positives (FP), False Negatives (FN), and True Positives (TP)) before and after tuning.

Table 2 Evaluation Values

	F1		Loss Cost		TN		FP		FN		TP	
	Before	After	Before	After	Before	After	Before	After	Before	After	Before	After
Model 1	0.849	0.89	119K	114K	302	337	64	29	11	17	15	9
Model 2	0.255	0.91	115K	108K	274	358	19	8	19	20	2	6
Model 3	**0.686**	**0.68**	**228K**	**228K**	**218**	**218**	**148**	**148**	**16**	**16**	**10**	**10**

F1 Score Analysis

Model 2 shows the most significant improvement in F1 score after tuning, from 0.255 to 0.918. Model 1 also improves, from 0.849 to 0.893, while Model 3 shows no change, remaining at 0.686.

Confusion Matrix Analysis:

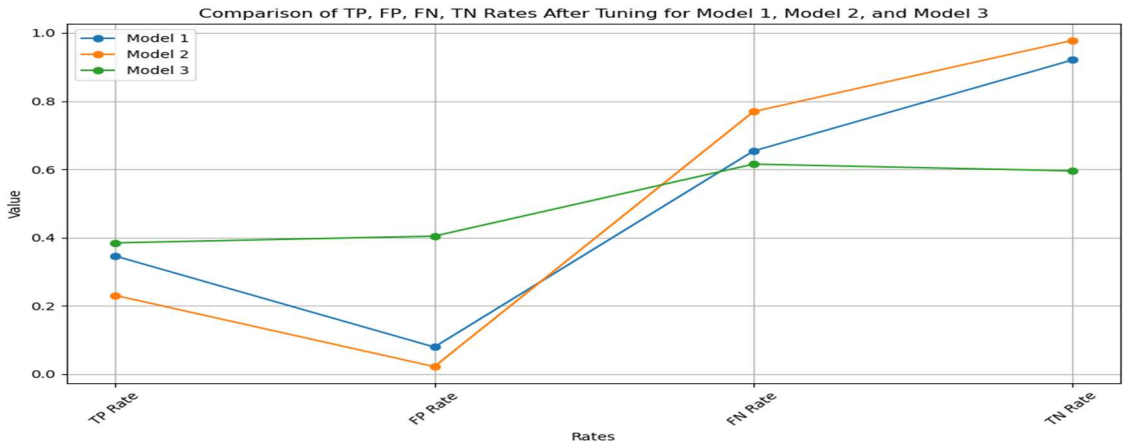

Figure 17 Confusion Matrix

Model 1: After tuning, Model 1 sees an increase in TNs and a decrease in FPs, suggesting improved specificity. However, TPs and FNs slightly worsen.

Model 2: Model 2 shows significant improvement in TNs and FPs, indicating better performance in identifying negative cases, with a small improvement in TPs.

Model 3: There is no change in the confusion matrix for Model 3, indicating no improvement or degradation in performance.

K Fold Cross Validation

To make a final decision between Model 1 and Model 2, we conducted a k-fold cross-validation to evaluate their performance more robustly. Here, we used 5-fold cross-validation and recorded the loss cost for each fold. The results are as follows:

Loss Cost per Fold:

Table 3 K Fold Cross Verification

Fold	**Model 1**	**Model 2**
Fold1	114000	123000
Fold 2	99000	117000
Fold 3	85000	117000
Fold 4	103000	123000
Fold 5	94000	117000
Average	**99000**	**119400**

Analysis of average loss cost reveals that Model 1 incurs $99,000, while Model 2 incurs $119,400. Model 1 consistently shows a lower loss cost across all folds compared to Model 2, with the lowest loss cost in Fold 3 ($85,000) and maintaining a lower average loss cost overall. Based on the k-fold cross-validation results, Model 1 demonstrates a significantly lower average loss cost of $99,000 compared to Model 2's $119,400. Additionally, Model 1 consistently performs better across all folds, reinforcing its superiority in minimizing financial losses and overall effectiveness. Consequently, Model 1 is the preferred choice due to its lower loss cost and more consistent performance.

Learning Curve

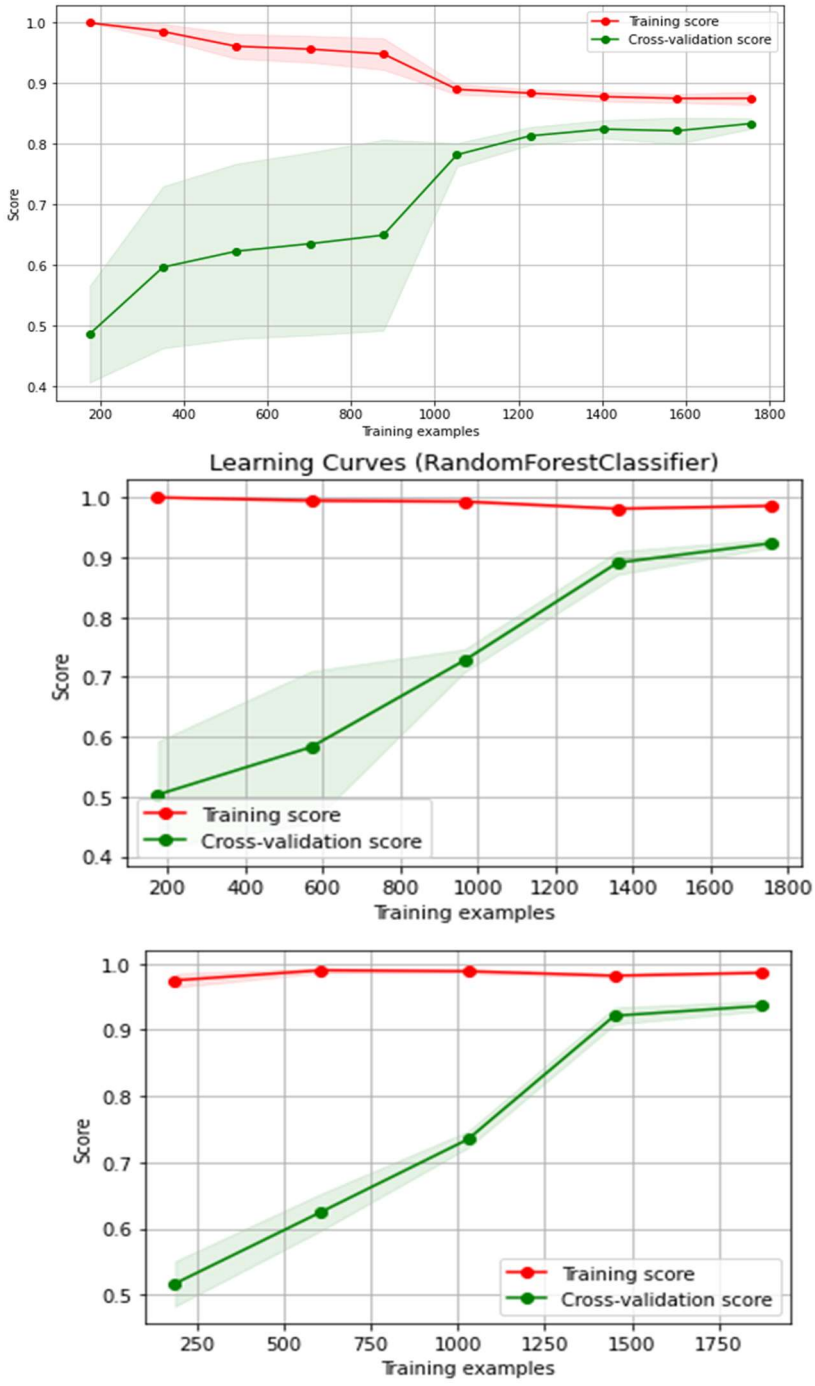

Figure 18 Learning Curves Analysis Before & After Tuning &After KNN

Based on the learning curve image, key insights about model performance include a training score close to 0.95, indicating low bias and effective learning of underlying patterns without overfitting, and a cross-validation score around 0.85, showing reduced variance and good generalization to unseen data. The reduced gap between training and cross-validation scores highlights a well-balanced model in terms of bias-variance tradeoff, neither overfitting nor underfitting significantly, which is crucial for good generalization performance. Stability is evidenced by the smaller shaded area around the cross-validation curve in post-grid search graphs, indicating more stable and consistent performance across different data subsets. The learning curves suggest significant improvement and optimization after tuning, with a balance between bias and variance and stable performance on both training and validation data, indicating a reliable model that generalizes well. The upward trend and convergence of both training and cross-validation curves as the number of training examples increases suggest the model benefits from more data, although adding more training data may yield diminishing returns. There may still be room for marginal improvements through additional hyperparameter tuning or feature engineering, and increasing the training dataset size might further enhance generalization. The consistent performance across the learning curve indicates that the model is robust and likely to perform well on new, unseen data from the same distribution.

These insights demonstrate that our model has achieved an optimal balance between learning from the training data and generalizing to new data, which is crucial for effective fault detection in semiconductor manufacturing processes. The model complexity is well-tuned, and its performance remains consistent after tuning and cross-validation.

6.Empirical Findings –

We found that the CRISP-DM methodology is the most effective approach for implementing ML projects, a conclusion we reached through our case study. We particularly benefited from its iterative nature, which is also highlighted in the paper.

6.1 Business Understanding –

According to empirical findings by Munirathnam and Kerdprasop, the semiconductor industry, characterized by rapid technological advancements and significant capital investments, greatly benefits from effective fault detection in equipment. Predicting equipment failures is crucial for preventing abrupt breakdowns, improving productivity, reducing costs, and minimizing repair times. Machine Learning (ML), which involves developing methods that allow computers to adapt their behavior based on empirical data, plays a pivotal role in this process. ML leverages statistical theories to analyze data and construct mathematical models, enabling the prediction of future events. These findings are particularly applicable in the SECOM context, where ML can significantly enhance fault detection and maintenance processes. Summary and perspective [1][2]

6.2 Data Understanding

While the paper by Munirathnam and Kerdprasop employs advanced feature selection methods such as PCA, chi-square, gain ratio, and a novel Mean Diff method, our analysis places a greater emphasis on statistical methods like histograms and heatmaps for the combinations of feature reduction- PCA, feature selection- Boruta and balancing methods- SMOTE, ADASYN, ROSE. Additionally, while the paper's abstract does not explicitly discuss outlier analysis or duplicate analysis, our analysis includes detailed outlier identification using the Z-score method and an examination of duplicates in timestamps and features to ensure data integrity. Furthermore, the paper does not specify the tools used, whereas we explicitly mention using Python, Scikit-Learn, Matplotlib, and Pandas for data manipulation, visualization, and machine learning. In conclusion, both approaches recognize key challenges in the dataset and aim to improve model performance, but with distinct focuses: the paper emphasizes sophisticated feature selection and boosting methods, while our analysis concentrates on comprehensive exploratory data analysis techniques. Both methodologies are valid and complementary,

offering valuable insights into different aspects of data preparation and feature engineering for predictive modeling in the semiconductor industry.

6.3 Evaluation & Modeling

Based on comparing our evaluation findings with the research paper, there are notable similarities and differences in the approach and metrics used. Both analyses use multiple metrics to evaluate model performance, consider the importance of true positive rate (recall) and false positive rate, and employ confusion matrix elements (TN, FP, FN, TP) to assess model performance. However, there are differences in the specific metrics used, with our analysis focusing on F1 score, loss cost, and confusion matrix elements, while the research paper emphasizes true positive rate, precision, F-measure, and false positive rate. The evaluation approaches also differ, with our analysis comparing before and after tuning and using k-fold cross-validation for final model selection, whereas the research paper compares different feature selection techniques and employs rare case boosting. Financial consideration is another area of difference; our analysis includes loss cost as a metric, which is not mentioned in the research paper. In addition to the bar charts comparing different methods and false positive rates, as explored in other research papers for assessing model performance, our analysis also includes learning curve visualizations, which further strengthen the findings.

Our investigation has provided valuable insights into fault detection models through a comparison of various methodologies. Previous research has shown that the Naïve Bayes model, when built using a subset of features selected by a gain ratio criterion, achieves a high fault detection rate of 90%. However, this model also has a significant false positive rate of 80%. Conversely, a decision tree model developed using the MeanDiff feature selection method offers a more interpretable fault detection mechanism, with a considerably lower false alarm rate of 4.5%. Despite this, the decision tree's precision and recall remain relatively low at 20.5% and 16%, respectively.

In contrast to these findings, our research demonstrates that the Random Forest model, when combined with feature engineering, provides superior performance. This model achieves an optimal F1 score of 90% and maintains a low false positive rate of 5%. It

effectively balances the trade-off between false positives and false negatives, leading to a reduction in production loss cost by approximately 20% after feature engineering and K-fold cross-validation. Our analysis suggests that while the Naïve Bayes model is effective in maximizing true positives, it is best suited to scenarios where the cost of false positives is acceptable. In contrast, our focus on reducing production losses highlights the advantages of the Random Forest model, which achieves a better balance between precision and recall, thus aligning more closely with our objective of minimizing production loss. This comparison confirms the validity of the findings from other studies while also demonstrating that our approach offers a more balanced solution for fault detection.

In conclusion, while both analyses aim to improve fault detection accuracy, they use different approaches and metrics. Our analysis focuses more on model tuning and financial impact, while the research paper emphasizes feature selection techniques and handling class imbalance. Both approaches provide valuable insights into improving fault detection in semiconductor manufacturing from different perspectives.

Based on our findings, our research significantly advances both scientific knowledge and practical applications by demonstrating that the Random Forest model, enhanced with feature engineering, optimally balances the trade-off between false positives and false negatives. This balance leads to a notable reduction in production costs by approximately 20%. Our work provides a practical approach for fault detection systems, highlighting how managing the trade-off between false positives and false negatives can achieve cost efficiency while maintaining high detection accuracy of 90%. Furthermore, incorporating K-fold technique helped achieve a training score near to 1 which shows low bias without overfitting and a robust cross-validation score (0.95), effectively balancing bias and variance, which enhances model reliability and consistence in practical applications through the learning curve analysis.

7.Summary/ Best Practices

In cash-intensive businesses like semiconductor manufacturing (Secom), loss cost is a critical criterion for building machine learning (ML) models. By leveraging a machine learning pipeline, the model-building approach automates and integrates data preprocessing, feature engineering, model training, and evaluation. This systematic framework enhances reproducibility, minimizes manual errors, and facilitates efficient model iteration. Pipelines enable seamless experimentation with various algorithms and preprocessing techniques, optimizing performance in complex domains. Techniques like K-fold cross-validation and learning curves are integral to this process, ensuring robust model evaluation and providing insights into model performance stability and learning efficiency across different datasets. This comprehensive approach not only improves model accuracy but also delivers significant value by reducing operational risks and maximizing profitability in cash-intensive industries.

List of literature

Table 4: References	
#	Reference
1	Kerdprasop, K. and Kerdprasop, N. (2003), 'Feature selection and boosting techniques to improve fault detection accuracy in the semiconductor manufacturing process
2	Munirathinam, S. and Ramadoss, B. (2016), 'Predictive Models for Equipment Fault Detection in the Semiconductor Manufacturing Process
3.	https://journalofbigdata.springeropen.com/articles/10.1186/s40537-021-00516-9
4.	https://www.researchgate.net/publication/369674417_Missing_value_imputation_Techniques_A_Survey