Hochschule für Technik
und Wirtschaft Berlin
University of Applied Sciences

Building data mining model to predict faulty products - An empirical case study

Mini Thesis

eMPMD2.2 Advanced Computational Data Analytics

Master Project Management and Data Science

Faculty 3

from

Himansha Gupta

Ekin Pomay Polat

Rashmi Carol Dsouza

Quynh Dinh Hai Pham

Date:

Berlin, 05.08.2022

Supervisor: Prof. Dr. Tilo Wendler

Index

List of figures

List of tables

Index of abbreviations

3s	three times standard deviations away from the mean
ADASYN	Adaptive Synthetic Sampling Approach
AUC	Area Under Curve
CRISP-DM	Cross-industry standard process for data mining
DT	Decision Tree
FN	False Negative
FP	False Positive
KNN	K-Nearest Neighbour
ML	Machine Learning
NA	Not Applicable
NB	Naïve Bayes

RF	Random Forest
ROC	Receiver Operating Characteristic
ROSE	Random Over-Sampling Examples
SMOTE	Synthetic Minority Oversampling Technique
SVM	Support Vector Machine
TN	True Negative
TP	True Positive

1 Introduction

Semiconductor production entails very complicated and time-consuming procedures with between 300 and 500 stages and a significant number of interconnected factors. Process monitoring and fault detection are critical for the industry to fulfil the increasing demand for high-quality goods and dependable operations. This is accomplished by closely monitoring numerous features concurrently and processes to rapidly discover abnormal behaviours and assign causes in order to decrease abnormal yield loss. As a result, the sector must continually monitor operations effectively and efficiently, as well as develop corporate plans by utilising useful insights gained from data mining methods and machine learning algorithms.

The objective of this study is to examine various fault detection approaches and attempt to use the information to develop a model that can properly predict the result of the defective characteristics. The following questions were used to direct the study and experimental processes:

- What are the best data modelling methodologies and strategies for defect detection, particularly for classification problems?
- What were the critical steps in establishing the quality of our models?
- What is the most effective mix of strategies for feature selection, balance, and machine learning models?
- What are the criteria, and how to evaluate the various findings to get the best balance of error and accuracy while meeting business objectives?

To address this list of problems, this paper will concentrate on the SECOM dataset. The major goal is to find the optimal model that not only provides low error and high accuracy but also minimises business loss. This paper is structured into the following sections. Chapter 2 describes the theoretical research on methodologies used to create predictive models for defect detection. A re-experiment utilising the SECOM dataset is explained and evaluated in Chapter 3. Finally, Chapter 4 summarises findings and recommendations for future process studies.

2 Theoretical research

The Cross-Industry Standard Process for Data Mining known as CRISP DM is a process model for translating business issues into data mining tasks and carrying out data mining projects regardless of both the application field and the used technology (Wirth and Hipp, 2000, para. 2). The CRISP-DM model provides a structured roadmap for our research method. This model, which represents the machine learning process, describes the common methods implemented by experts in this field to overcome problems (Munirathinam and Ramadoss, 2016, p. 275). In order to achieve a well implemented data science project, a reliable and repeatable process must be followed. The CRISP-DM model promotes best practises and provides businesses with better and faster results from data mining and divides the data mining process into the following six steps: business understanding, data understanding, data preparation, modelling, evaluation, and deployment (Shearer, 2000, pp. 13-14).

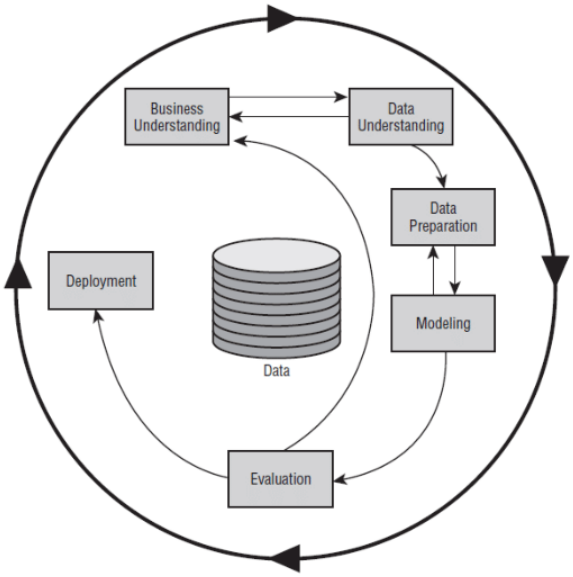

Figure 1: Visualisation of CRISP-DM (Shearer, 2000, p. 14)

2.1 Business understanding

Before starting any project, the most critical step is to understand the business, assess the project goals from their perspective, identify the business problems that need to be tackled and develop a strategic plan to accomplish project objectives. This crucial phase will greatly assist analysts later in approaching and analysing data to address business challenges (Shearer, 2000, p. 14). This paper focuses on semiconductor manufacturing with the objective of finding the most effective model to help with the monitoring process and predict imminent failure. The SECOM dataset, which has been made accessible for more study and re-experimentation, will then be used to experiment and evaluate various theoretical methods and research findings.

Semiconductor chips are the future's cornerstone. Wafer manufacturing is a time-consuming process that includes turning pure silicon into thin raw wafer discs, generating surface conductor structures on each disc, combining these layers into microchips, and evaluating the finished result. It can take up to 700 processing steps and 14 weeks to complete. In most manufacturing processes, cost, quality, and delivery time are key factors in gaining a competitive advantage. Process engineers must monitor and identify abnormalities in the production process as soon as possible. As a result, semiconductor fabrication must go through hundreds of phases that are meticulously monitored and recorded in real time with cutting-edge sensors. The collected data from these sensors is used to improve efficient control and optimization. However, the data volume is usually so large that it is tough to spot any faults throughout the manufacturing process in a timely manner (Munirathinam and Ramadoss, 2016, p. 273).

The semiconductor industry is now dealing with three significant concerns. To begin with, quality control is getting increasingly challenging as the size and dimensions of integrated circuits continue to reduce to enable nano-generation. The second issue is poor data quality, which leads to inadequate, incomplete, or incorrect process measurements. Finally, a significant amount of data makes it harder to extract relevant information and monitor and manage more complicated production processes. It is also difficult to find relationships between each feature, which increases the number of mistakes. Due to intense market competition, researchers and the industry continually look for new technical advancements to improve performance and reduce manufacturing costs.

2.2 Data understanding

The second phase of the CRISP-DM model, data understanding, begins with data collection. The analyst then proceeds to spot issues with data quality, gain a preliminary understanding of the data, or explore noteworthy subsets to generate hypotheses about hidden information (Shearer, 2000, p. 15).

The SECOM dataset consists of 2 files: Secom.data and Secom_labels.data. The first file has 1567 observations from a wafer manufacturing production line, with 590 sensor readings per observation. The second file provides the timestamp along with the classification status, consisting of 1463 pass and 104 fail cases. The ratio between pass and fail cases is 14:1. Due to a large number of observations with several sensor readings and other issues such as missing values, the imbalance between passed and failed classifications, etc., it is challenging to analyse this dataset.

2.3 Data preparation

The data preparation phase consists of all tasks that constitute the final dataset from the initial raw data in order to ready the data for further processing (Shearer, 2000, p. 16). The following data preparation stages are used to prepare the dataset in order to extract the most important features that can be included into models.

2.3.1 Splitting data into training and test datasets

The development of a more accurate model stems from the partitioning of data into training and test sets. The training set is used to identify patterns in the data, while the test set is used to show how effectively the model works (Kerdprasop and Kerdprasop, 2003, p. 114). Before performing any data processing, it is critical to separate data into a test subset to retain the data characteristics. Thus, this must always be the initial step in the process of constructing any model (Wendler and Gröttrup, 2021, p. 1199).

Data division into training and test sets aids in the training and assessment of machine learning algorithms. The model is developed using a subset of the dataset, while the remainder is used to test the model. When comparable data is used to train and test the model, it is more vulnerable to overfitting due to the model's propensity to memorize data and cannot successfully generalize to unknown data. When doing data splits, it is critical to ensure that the data is partitioned into appropriate pieces. The training data should account for the majority of the dataset (80%), while the testing data should represent approximately 20%. The goal of dividing the data set is to estimate the model's performance on the additional data (test/validation set).

2.3.2 Feature removal

The statistical power of a study can be decreased by missing data, which can also lead to biased estimates and false findings (Kang, 2013, p. 1). In order to remove features containing numerous missing values, a threshold can be defined.

2.3.3 Outlier detection and treatment

An observation that differs significantly from the other values is generally labelled as an outlier. Since variables in a dataset frequently have a partial relationship with one another, one may define an outlier as a data entry that is situated apart from the rest of the data entries (Benatti, 2019, p. 2). It is crucial to clarify that an outlier is not necessarily a mistake. There are several ways to identify outliers. One approach is if a value is 3 times standard deviations away from the mean (3s), that data point is identified as an outlier. Removing the outliers might cause a data loss. Therefore, the outliers can be treated as missing values or replaced with 3s boundaries.

2.3.4 Missing value imputation

One of the most frequent issues in data mining research is missing value. It causes issues during data analysis and processing, which reduces efficiency (Das, Nayak and Pani, 2019, p. 548). In order to deal with missing values, there are two main options: missing value imputation or missing value removal. Removing data may not be the best option. Removing features with missing values may cause information loss and affect our modelling process (Roye, Hilton and Smart, 2021, p. 15). In the SECOM dataset, in order to impute missing values, several methods are implemented. Mean imputation is a method where each variable's mean of the observed values is calculated, and this mean is used to fill in the missing values for that variable (Jamshidian and Mata, 2007, pp. 26-27). K-Nearest Neighbours (KNN) imputation method is used to fill the missing values in the datasets with the mean value from the parameter 'n_neighbors' nearest neighbours based on Euclidean distance (Kuhn and Johnson, 2016, p.159). Another method to apply is MICE, which stands for Multiple Imputation by Chained Equations. It works by training predictive models with the features that have missing values as the target and all the remaining features as predictors. It cycles through the models and uses predictive mean matching to find the imputed values from the output of the trained predictive models (Azur *et al.*, 2011, pp. 41-42).

2.3.5 Feature selection

The semiconductor manufacturing process produces a diverse assortment of characteristics with a high amount of noise. The process of selecting a subset of features from a set of high-dimensional characteristics that have the highest output performance based on specified evaluation criteria is known as feature selection.

It shrinks the feature space dimension by determining which characteristics to retain and which to discard. After filling the dataset with missing values via data imputation, the work of feature selection may be completed. This work is required to remove unnecessary and distracting elements while retaining just the relevant data for the following stages. The SECOM data collection contains a lot of redundant information. As a result, the unnecessary data must be removed.

Univariate Feature Selection: This approach selects the features with the highest scores after running a univariate statistical test on the input data. The Chi-square Test is used as one of the feature selection methods on this dataset. The Chi-square between each feature and the objective was calculated using this method, and the features with the greatest Chi-square scores were chosen.

Choosing from a Model: Using this feature significance, the highest-ranking features are chosen after fitting a Random Forest (RF) model to the dataset. Random forests are rated depending on how well they eliminate impurity (Gini impurity). The nodes with the least impurity are at the

beginning of the trees, while the nodes with the highest impurity are at the end. Thus, a subset of the most relevant characteristics may be constructed by pruning trees below a certain node.

Wrapper Method: It employs a greedy search strategy, assessing all potential feature combinations against the evaluation criterion. Forward Feature Selection is used, which is an iterative process that starts with the variable that performs the best against the target. Then, it chooses another variable that, when combined with the first, yields good results.

Embedded Techniques: These techniques combine the benefits of wrapper and filter approaches. Embedded approaches handle each iteration of the model training process and extract the features that contribute the most to training of that iteration. Methods used include Lasso Regression, Boruta, BorutaSHAP, and other techniques.

During the course of this study, 12 unique feature selection algorithms are tried.

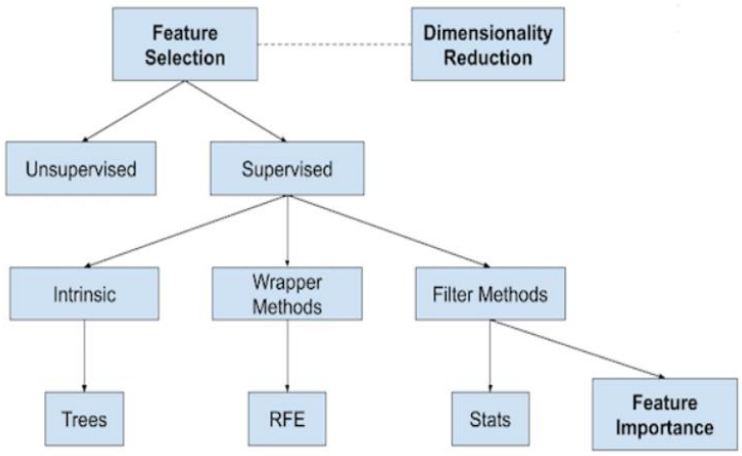

Figure 2: Feature Selection Methods

Data preparation is the most critical and time-consuming element of developing a predictive model. As a consequence, required steps are taken to remove unneeded or incorrect data and prepare clean data for the model construction phase, which employs machine learning techniques. These steps include dividing the dataset in an 80/20 ratio into training and test datasets; removing features with a high proportion of missing values that exceed a certain threshold; identifying outliers using the three-sigma rule and replacing them with values that are three standard deviations from the mean; and imputing missing values and selecting important features from the remaining dataset.

There are various benefits to choosing a subset of all relevant attributes; for example, it aids with data visualization and understanding. It reduces the requirements for measurement and storage, as well as training and utilisation durations. The accuracy of the model is improved by using a subset of the dataset. To obtain reliable results in cases involving hundreds of characteristics, a

subset of the data must be utilized. As a result, the previous methods were implemented to reduce the dataset to a subset that may produce more accurate results.

2.4 Modelling

2.4.1 Balancing

To address the issue of data imbalance, the relative number of minority instances must become equal to or comparable to the relative number of majority instances. To achieve this balance, either the minority class (in SECOM's case, the failures) can be oversampled or the majority distribution can be undersampled. There are several methods for balancing the dataset based on class, including undersampling, oversampling, and a hybrid of oversampling and undersampling (Chawla, Japkowicz, and Kotcz, 2004, p. 1). Techniques such as Random Over-Sampling Examples (ROSE), Synthetic Minority Oversampling Technique (SMOTE), and Adaptive Synthetic (ADASYN) are used to balance the data, as oversampling performs better than undersampling for various classifiers and achieves higher scores in various evaluation metrics (Mohammed, Rawashdeh, and Abdullah, 2020, p. 247). ROSE is a bootstrap-based method that facilitates binary classification in the presence of minority classes (Lunardon *et al.*, 2021, p. 2). SMOTE, an oversampling method, generates synthetic minority class data points to balance the dataset. The goal is to combine the oversampling and undersampling strategies to provide another sampling method for handling imbalanced class data (Chawla *et al.*, 2002, p. 331). As an extension of SMOTE, ADASYN implements a technique to automatically decide how many composite samples each minority sample needs to create in order to accomplish the aim of data balance (Chen, Zhou and Yu, 2021, p. 4).

2.4.2 Scaling and normalisation

In the model building process, before fitting the final model and evaluating its quality, there is a need to consider scaling the features because they have different dimensions. There are several techniques that can be used for scaling, such as Standardization, Min-Max scaling, Z-Score, Box-Cox and logarithmic (Dago *et al.*, 2021, p. 54).

2.4.3 Model building with machine learning algorithms

This paper focuses on addressing the classification problem in the semiconductor industry with the objective of developing a model that can reliably classify a given set of inputs into pass or fail categories. There are two types of machine learning (ML) algorithms: supervised learning and unsupervised learning (Alafandy *et al.*, 2022, p. 86). Supervised learning is the process of training the machine learning algorithm with previously classified inputs and outputs and then testing the

algorithm using newly unlabelled data to check if the new data is properly classified. If not, the process is repeated to improve the performance.

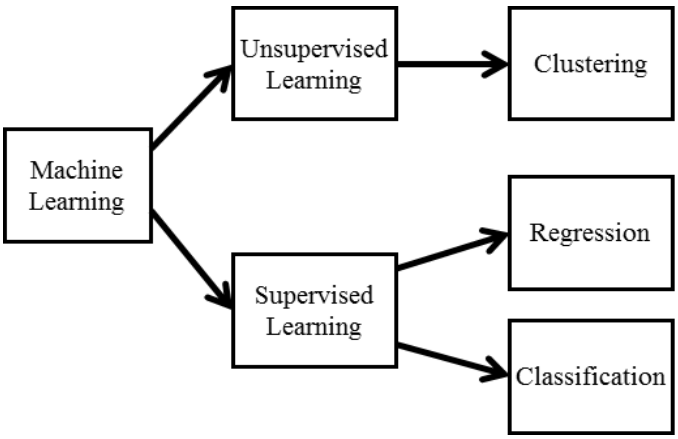

Figure 3: Types of machine learning algorithms

Supervised learning is further divided into two types: regression and classification. Classification is the process of identifying, understanding, and grouping data into predetermined categories. Using pre-categorized training datasets, machine learning programs classify future datasets using a number of techniques. In a classification model, the predicted output is discrete, meaning that the input variables are mapped into discrete groups or classes (Sen, Hajra, and Ghosh, 2019, p. 100).

The choice of ML algorithm depends on the type of machine learning that is best suited for your dataset. For classification problems, the choice is either to use classifying algorithms, which are useful for sorting items into categories, or a decision tree algorithm, which can be used to tackle both classification and regression issues but is most commonly used for classification problems. Some of the most common algorithms are the Naive Bayes (NB), the K-Nearest Neighbours (KNN), the Support Vector Machine (SVM), the Decision Tree (DT) Classifier, and the Random Forest (RF) Classifier.

NB calculates the possibility of whether or not a data point belongs within a certain category based on the concept of the Bayes theorem. Multinomial, Bernoulli, and Gaussian are the three main types of NB classifier (Alafandy *et al.*, 2022, p. 97). The advantages of this method are fast processing with fairly trustworthy results; working well with big data, having a short training time; and giving better classification performance by removing irrelevant features. The disadvantages are that it requires large amounts of data to give a satisfactory result and performs worse than other classifiers depending on the type of problem (Çolakoğlu and Akkaya, 2019, pp. 23-24).

KNN is one of the most basic machine learning algorithms used in supervised learning. It assumes that new and prior data are comparable. The process measures the distances between a query and

all data points, then identifies the K sample(s) closest to the query and places the query in the category that is most similar to the pre-existing groups (Zhang *et al.*, 2017, p. 2; Lubis, Lubis and Khowarizmi, 2020, p. 2). The K value is determined by iterations and tests. However, the K value usually is usually larger when there are more neighbours and lower when there are fewer neighbours.

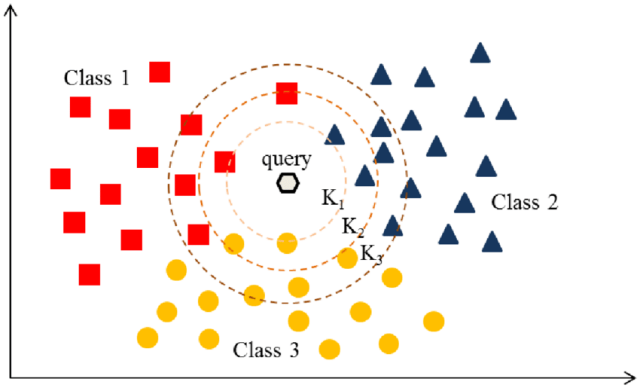

Figure 4: KNN machine learning algorithm

The positives of KNN are the ease of implementation and comprehension due to lack of assumption requirement, the ability to respond promptly to input changes during real-time use, and the easy application to multi-class classification issues. The negatives, on the other hand, are slower processing time for larger dataset and more difficult in obtaining the output with increasing variables. Besides, it is incapable of dealing with missing values, is influenced by outliers and requires scaling features to the same scale to function properly (Çolakoğlu and Akkaya, 2019, p. 24).

SVM identifies the best line or decision boundary (hyperplane) separating n-dimensional space into classes. Margin is the distance from the hyperplane to the nearest data point; and support vectors are the data points closest to the decision boundary. To develop the most efficient SVM classifier is to maximise the margins between classes, particularly the nearest classes, on both sides of the hyperplane (Alafandy *et al.*, 2022, p. 100).

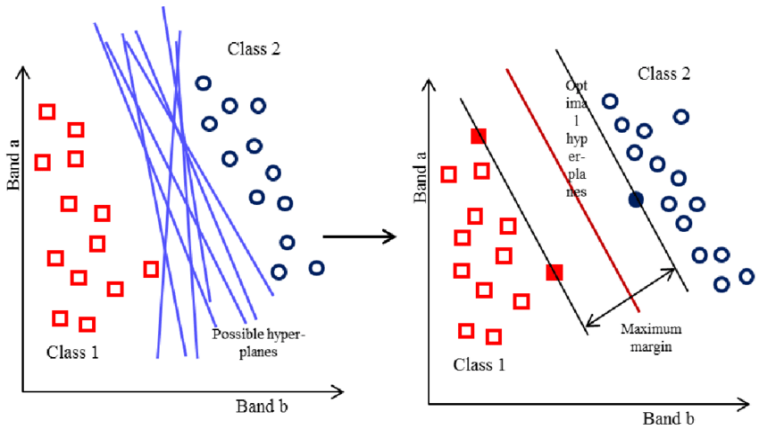

Figure 5: SVM machine learning algorithm

The strength of SVM is in generating decent results even when dealing with incomplete or unstructured data and solving any complicated issue with a suitable kernel function. Also, it is more effective with high dimensional data. However, long training time for large datasets, poor performance when classes are overlapping, and difficulty in selecting the suitable kernel function are some of SVM weaknesses (Çolakoğlu and Akkaya, 2019, p. 23; Auria and Moro, 2008, pp. 7-8).

DT Classifier is a type of tree-structured classification technique. Starting from the root node, the tree is recursively split into internal nodes based on a certain test condition until it is left with pure leaf nodes containing data with only one type of class. Different test conditions will result in different sets of internal nodes or leaf nodes (Alafandy *et al.*, 2022, p. 99).

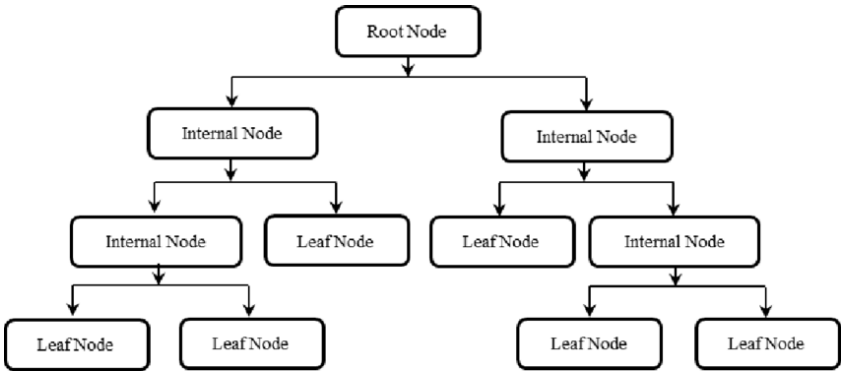

Figure 6: DT Classifier machine learning algorithm

This algorithm has several advantages, including the fact that it does not require data scaling, or normalisation and has high computational efficiency. It is also unaffected by missing values and is highly intuitive and easy to implement. However, the accuracy is largely determined by the tree design and feature selection (Alafandy *et al.*, 2022, p. 100). Other drawbacks include complex calculation, expensive and time-consuming training, and being prone to overfitting.

RF Classifier is a prominent tree-structured classification algorithm that incorporates decision trees and ensemble learning. The algorithm combines many decision trees, trains each of them on a distinct sample from the provided dataset, and generates the final outcome based on the majority vote of the trees.

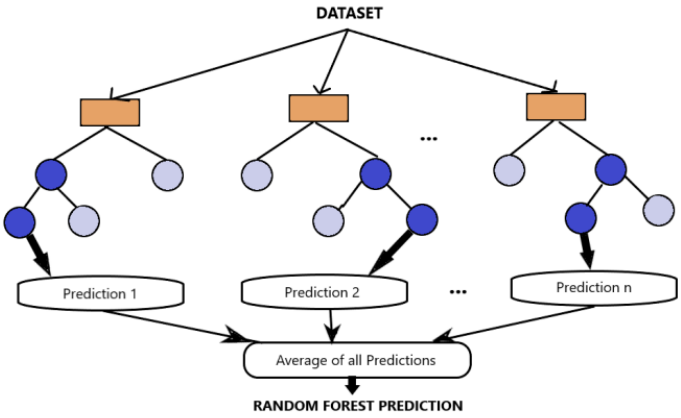

Figure 7: RF Classifier machine learning algorithm

The advantages of RF are numerous. By averaging the outputs of multiple trees, RF can overcome the usual overfitting issue of a single decision tree, resulting in a more accurate prediction. Besides, RF does not require feature scaling and can handle missing values automatically. The algorithm is quite stable since any small change may affect one decision tree but not all of them; hence, the overall algorithm is not greatly affected. However, this method is not without its shortcomings. It is a highly complex algorithm that is difficult to visualise and comprehend. RF also requires a substantial amount of computation and is far more difficult and time-consuming to train than a decision tree (Çolakoğlu and Akkaya, 2019, pp. 25).

A learning algorithm does not always work flawlessly. The two most common problems in supervised learning are overfitting and underfitting. Overfitting occurs when a model can predict the results of the training dataset with 100% accuracy after training but fails to do the same with another dataset. In the event of overfitting, there are two solutions: reducing features in the feature selection stage or increasing the quantity of training data using scaling approaches. Underfitting, on the other hand, occurs when the model fails to predict the output of the training dataset after the training phase has been completed. Increasing the number of features or training iterations might help to solve this problem (Alafandy *et al.*, 2022, p. 94). It is advised to do an analysis following the model building to avoid these problems.

2.4.4 Hyperparameter tuning

Data is used to train model parameters, and hyperparameters are adjusted to get the best fit. Finding the right hyperparameter takes time, and the choice of hyperparameters impacts the effectiveness of the training. The gradient descent learning rate defines how successful and exact the optimization process is at predicting parameters.

Manual hyperparameter tuning requires experimenting with various sets of hyperparameters, i.e., each trial will use a new set of hyperparameters. Because there are multiple trials to keep track of, manual tuning is a time-consuming and costly process.

Random Search CV algorithm provides a grid of possible hyperparameter values. Each iteration tries a different set of hyperparameters from this grid, logs the results, and finally returns the optimal set of hyperparameters.

Grid Search CV provides a grid of hyperparameter values. Each iteration tries a new combination of hyperparameters in order. It fits the model to every feasible hyperparameter combination and reports on the model's performance (Feurer and Hutter, 2019, p. 7). The best model with the best hyperparameters is then returned. A grid search, for example, would build a model for each feasible combination of nestimators and maxdepth.

Early Stopping is a type of learning curve estimation used in predictive termination, in which a learning curve model projects an observed learning curve with different combinations, and the training is terminated if the configuration performs worse than the best model generated, thus it is far from the optimization (Feurer and Hutter, 2019, p. 15). Early termination reduces overfitting. As a result, performance on data outside of the training set improves to some extent. However, strengthening the fit might increase the generalisation error. Early stopping criteria describe the number of iterations that can be conducted before the learning dataset becomes over-fit. Thus, an overfitting analysis is used to predict the number of iterations required to reduce the likelihood of overfitting.

2.5 Evaluation

Multiple assessment measures may be used to evaluate the success of the classification algorithms. In this study, the confusion matrix, precision, recall, F1-score, AUC, and ROC curve will be utilised to assess the performance of each class prediction. A confusion matrix is a two-by-two matrix used to evaluate the performance of a binary classifier. Each cell indicates the number of correct or incorrect predictions made by the model for each class. These measures are called True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN). TP is the number of occurrences where the model correctly identifies positive classification; TN is the number of occurrences where the model correctly identifies negative classification; FP is the number of occurrences where the model mislabels negative classification as positive; and FN is the number of occurrences where the model mislabels positive classification as negative.

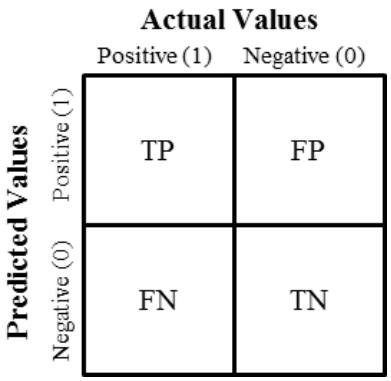

Figure 8: Confusion matrix for binary classifiers

Precision, recall, and F1-score may be determined as a consequence of this confusion matrix (Alafandy *et al.*, 2022, p. 106). Precision is defined as the percentage of correct positive classification out of all actual samples in a positive class. Recall is defined as the percentage of correct positive classifications out of the total number of actual samples in that class. The F1-score incorporates recall and precision into one metric. The following are the mathematical formulas:

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

$$F1 - score = 2 \times \frac{Precision \times Recall}{Precision + Recall} = \frac{2 \times TP}{\left(2 \times TP\right) + FP + FN}$$

When checking or visualising the performance of a multi-class classification issue, the Area Under the Curve (AUC) and Receiver Operating Characteristics (ROC) curves can be used. The degree of separability and the probability curve, often known as the AUC and ROC curves, are performance indicators for a model's ability to distinguish between classes. The greater the AUC is, the more accurate the model is in classifying data (Cortes and Mohri, 2003, p. 313).

2.6 Summary

To conclude, there are several phases and different approaches to develop a predictive model for fault detection. The decision of which method to use at each phase is often subjective to business issues and project objectives. Following is the summary of the most important theoretical research findings.

1. The initial stages in any project are to understand the business and its data. This helps in defining project objectives from a business perspective, identifying business issues, reviewing data quality to gain initial insights, and creating a plan to achieve project goals.
2. Data preparation is the most crucial and time-consuming process for the success of constructing a predictive model, therefore, necessary steps are taken to eliminate irrelevant or incorrect data and ensure that clean data is prepared for the model building phase using machine learning algorithms. These steps include dividing the dataset into training and test datasets with the ratio of 80% and 20% respectively; removing features having a high proportion of missing values that exceed a certain threshold; identifying outliers by the three-sigma rule and replacing them with 3s boundaries; imputing missing values and selecting important features from the remaining data.
3. There are three methods for missing value imputation: mean, KNN or MICE imputation. Each approach has its own set of benefits and drawbacks. However, MICE and KNN imputation are superior alternatives among the three approaches and should be considered in the SECOM re-experiment.
4. Statistical-based feature selection approaches entail applying statistics to evaluate the relationship between each input variable and the target variable and selecting the input variables having the strongest link with the target variable. Although the choice of statistical measures is dependent on the data type of both the input and output variables, these approaches can be quick and successful. Chi-square Test, Random Forest, Forward Feature Selection, Lasso Regression, Boruta were the various feature selection methodologies that were best suited for our classification data set.
5. Undersampling, oversampling, and a combination of both strategies are available methods for balancing a dataset. Oversampling frequently outperforms undersampling for many classifiers and obtains higher scores in numerous assessment criteria; therefore, in order to mitigate the disadvantages of undersampling, it is recommended to use oversampling techniques such as ROSE or a combination of both techniques such a SMOTE and ADASYN.
6. Machine learning algorithms are divided into two categories: supervised and unsupervised learning. Supervised learning is further divided into regression and classification types. Some of the most common algorithms for classification problems are Naïve Bayes, K-Nearest Neighbours, Support Vector Machine, Decision Tree Classifier and Random Forest Classifier. Each method comes with its advantages and disadvantages as seen in Table 1.

ML Algorithm	**Advantages**	**Disadvantages**
Naïve Bayes	– Fast process with fairly trustworthy results – Work well with big data – Short training time	– Need big data to produce a good result

ML Algorithm	Advantages	Disadvantages
	– Give better classification performance by removing irrelevant features	– Perform worse than the other classifiers depending on type of problem
K-Nearest neighbours	– Ease of implementation and comprehension due to lack of assumption requirement – Respond promptly to input changes during real-time use – Easy application to multi-class classification issues	– Slower processing time for larger dataset. Also, harder to obtain the output with increasing variables – Incapable of dealing with missing values and is influenced by outliers – Require scaling features to the same scale to work properly
Support Vector Machine	– Generate decent results even when dealing with incomplete or unstructured data – Solve any complicated issue with a suitable kernel function – More effective with high dimensional data	– Long training time for large datasets – Poor performance when classes are overlapping – Difficulty in selecting the suitable kernel function
Decision Tree Classifier	– No need for feature scaling or normalisation – Not affected by missing values – High computational efficiency – Implementation is easy, robust and simple	– Accuracy is largely determined by the tree design and feature selection – Complex calculation – Expensive and time-consuming training – Prone to overfitting
Random Forest Classifier	– Overcome overfitting issue of decision tree by averaging the outputs of multiple trees – Extremely powerful and highly accurate – Not require feature scaling	– Difficult to visualise and comprehend – Require a substantial amount of computation – More difficult and time-consuming to train than a decision tree

ML Algorithm	Advantages	Disadvantages
	– Handle missing values automatically – Stable algorithm	

Table 1: Comparison of machine learning algorithms for classification problems

7. A learning algorithm does not always work flawlessly. Overfitting and underfitting are the two most common issues. Therefore, it is advised to do an analysis following the model building phase to avoid these issues.
8. The metrics used to measure and assess the performance of different models include confusion matrix, type 1 and type 2 errors, sensitivity, specificity, F1-score, AUC, and ROC curve. However, there is always a trade-off between sensitivity and specificity; therefore, a cost function and business objectives should be taken into consideration for different model comparison and decision making.

3 Experimental result

The CRISP-DM methodology is used to approach the SECOM case. The dataset goes through various stages as detailed in Figure 9, which are thereafter executed using Python (version 3.10) as the programming language.

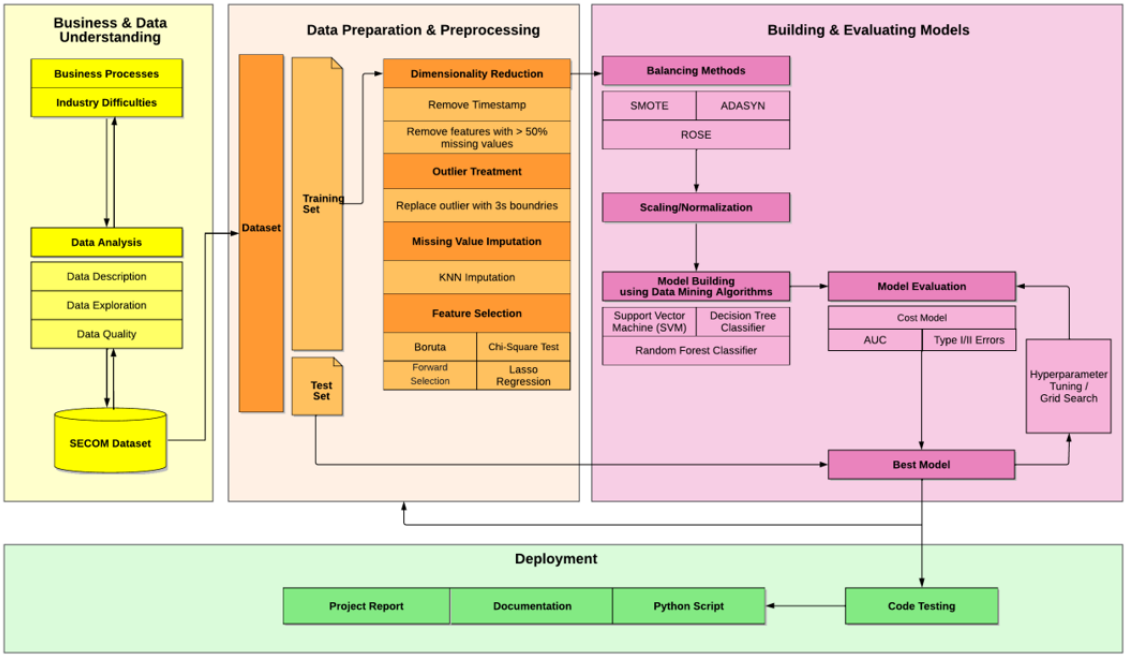

Figure 9: Steps performed to implement CRISP-DM

The original SECOM dataset consists of 2 files: Secom.data and Secom_labels.data. The first file has 1567 observations from a wafer manufacturing production line, with 590 sensor readings per observation. The second file provides the timestamp along with the classification status consisting of 1463 pass and 104 fail cases. Both the files are merged before being split into training and test datasets in an 80/20 ratio. To ensure that the subsets are representative of the original dataset, the ratio of failed cases to total cases remains constant at 6.6% and 6.7% for the training and test datasets, respectively.

Training dataset:

After splitting the data, 32 features, which have more than 50% of their data points missing, are removed. Then, the values that lie outside the 3s boundaries i.e. the outliers, are flagged. These values are then replaced with values corresponding to a mean ± 3 x standard deviation. Finally, as KNN imputation produces better overall results than MICE imputation, missing data is imputed using KNN imputation with n_neighbors = 5.

Test dataset:

The same processes are also applied to the test dataset.

3.1 Feature Selection

The following four methods of feature selection are applied.

Name	Tools	Implementation	Remaining Features
Boruta	BorutaPy Package	RF is used as the underlying model for feature selection. The number of iterations is set to 455. A function is composed to rank the features, and only features with rank 1 are considered; tentative features are discarded.	20
Chi-Square Test	chi2 module from the Sklearn library	Features having Cramer's V greater than 0.4, indicating a strong association with the class variable are selected.	29
Lasso Regression	lasso module from the Sklearn library	Linear Regression is used as the underlying model for feature selection. The alpha is set to 0.2.	29
Forward Feature Selection	SequentialFeatureSelector module from the mlxtend library	Linear Regression is used as the underlying model for feature selection. The forward argument is set to true.	25

Table 2: Outcomes of feature selection methods

Each method generates different subsets having different important remaining features. However, there is no optimum feature selection technique at this stage. When trying to determine which method works best by applying the RF model to these four subsets, the results are skewed due to the class imbalance. Instead of deciding which feature selection method to utilize, it is best to apply case balancing and feature scaling to all four subsets before applying any machine learning models and then evaluate the outcomes.

3.2 Balancing

A balanced training dataset, as opposed to an imbalanced dataset, provides the models with a greater number of instances to train on. As a result, models trained on balanced data can classify faults more accurately and perform better. The accuracy of the RF model changes, after being

trained on different subsets of selected features, using different balancing techniques. See Figure 10.

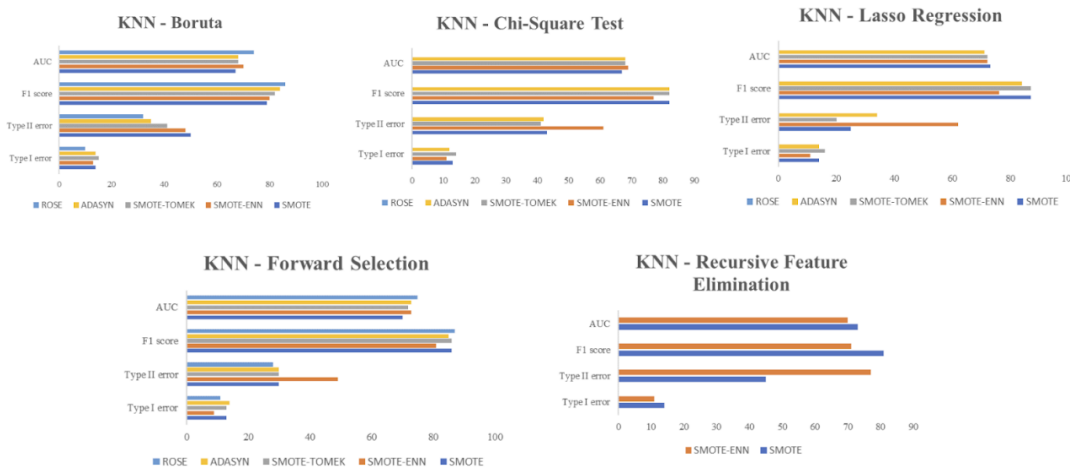

Figure 10: Accuracy of feature selection with balancing techniques

The SMOTE, ADASYN and RandomOverSampler modules from imblearn Library are used for implementing the following balancing methods.

3.3 Model building using machine learning algorithms

Before feeding the datasets into different machine learning algorithms, the data is standardized using the Standard Scaler module from the Sklear library. Then, the following machine learning algorithms are used: Decision Tree, Random Forest, and SVM. The models are trained on the training dataset and evaluated using the test dataset.

Different evaluation metrics are used on the models to predict the accuracy with the least error. The AUC is considered to be a good metric to use when classifying an imbalanced dataset. The total numbers of Type 1 and Type 2 errors, which are FP, and FN respectively, are also important from a business standpoint and taken into consideration. Therefore, models with higher AUC, lower Type 1 errors, and lower Type 2 errors are considered better for SECOM case.

Random Forest:

RF model is used on balanced datasets (ROSE, SMOTE, and ADASYN) to predict confusion matrix results and other important metrics such as sensitivity, recall, precision, F1-score, and AUC.

Confusion Matrix		Actual Class		
		Pass	Fail	Total
Predicted Class	Pass	269	10	279
	Fail	24	11	35
	Total	293	21	

Table 3: Confusion matrix of Boruta–ROSE–RF model

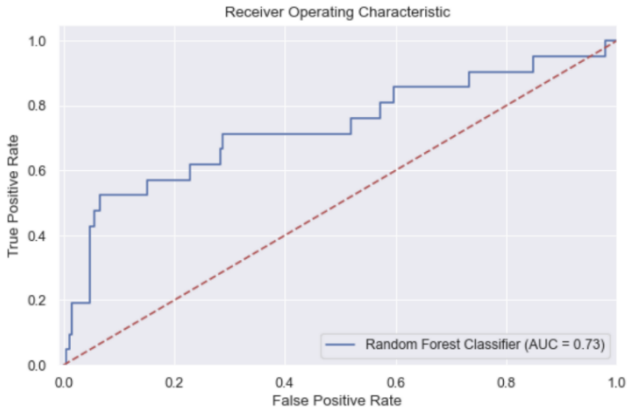

Figure 11: ROC curve and AUC of Boruta–ROSE–RF model

The confusion matrix of the Boruta–ROSE–RF model is depicted in Table 3. The model has F1–score of 0.89 and its AUC is 0.73. The type 2 errors were reduced from 30 to 24 after hyperparameter tuning the model's parameters using Grid Search CV.

While implementing RF, the bootstrap method is used on the training data set to obtain the best results.

Resulting from the confusion matrices of all the models, the following graph is constructed based on the number of Type 1 errors (FPs) and Type 2 errors (FNs). The AUCs of all the models are graphed as well. When a model has a higher AUC and fewer FPs and FNs, it is considered good. Following this, the graph below (Figure 12) shows that the three combinations (Boruta–ROSE–RF, Chi-square Test–ADASYN–RF, and Lasso Regression–ADASYN–RF) give the best results.

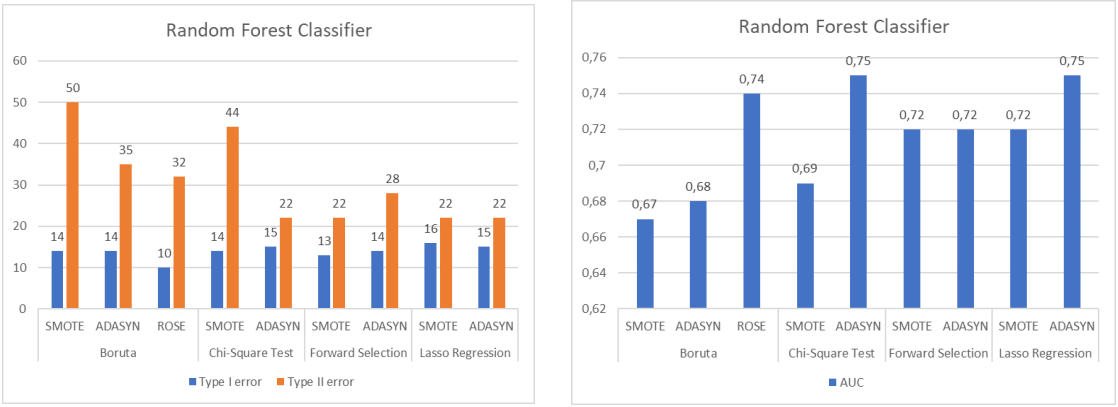

Figure 12: Summary results of RF models based on Type 1, Type 2 errors and AUC

SVM:

With the SVM model, the combination of Chi-square Test–ADASYN–SVM has the lowest number of errors. However, the Type 1 errors are too high in this case. Other combinations of Boruta–

SMOTE–SVM and Boruta–ADASYN–SVM have the highest AUC and lowest Type 1 errors, but the Type 2 errors are higher in comparison to other models.

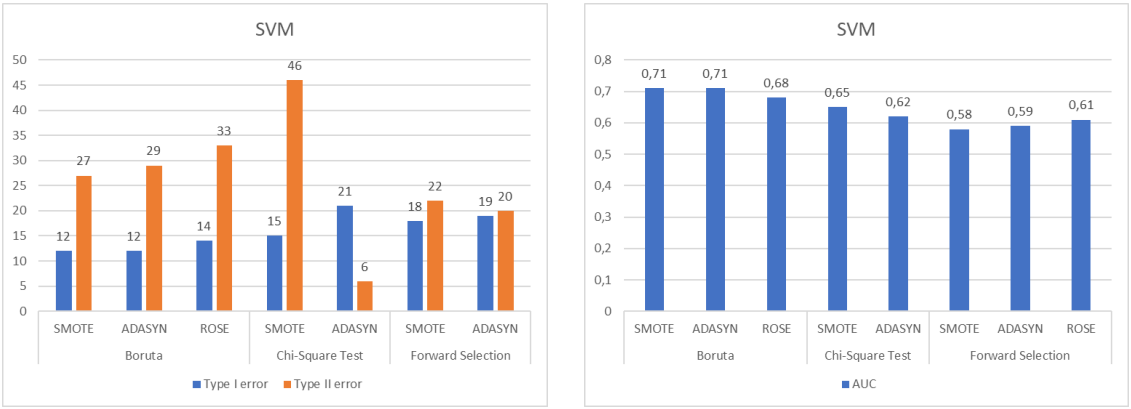

Figure 13: Summary results of SVM models based on Type 1, Type 2 errors and AUC

Decision Tree:

With the DT model, the combination of Boruta–ROSE–DT gives the best result with the lowest number of errors and the highest AUC.

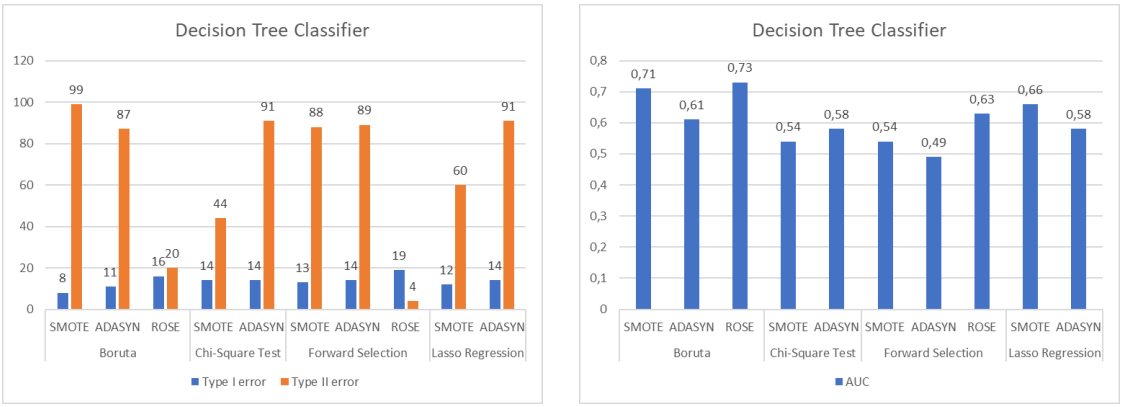

Figure 14: Summary results of DT models based on Type 1, Type 2 errors and AUC

KNN and Naïve Bayes:

The NB model is trained on 16 combinations of balanced data using the datasets generated from the feature selection methods. The KNN model is also trained on 16 combinations of balanced datasets where n_neighbors = 5. However, none of the results are better than the ones generated using RF, SVM, and DT.

3.4 Results comparison

Performance measures are used to select the best model from many candidates. The following are the selection criteria for important measures defining the model's success:

a) **AUC**: To measure the performance of classification models, AUC can be used. It tells how well the model can distinguish between classes 'Pass' and 'Fail'. When evaluating and comparing classification algorithms, AUC is a better measure than accuracy (Ling, Huang and Zhang, 2003, p. 524). The AUC of the applied model results ranges from 0.5 to 0.75. To select classifiers with a relatively better ability to distinguish between classes, a cut-off value of 0.7 is used.

   AUC will not provide us with a complete picture of misclassification costs (Adams and Hand, 1999, p. 1146). Hence, Type 1 errors, Type 2 errors, and overall costs are considered in order to measure model performance.

b) **Type 2 error**: Semiconductors are expensive to manufacture. The losses incurred here are manufacturing losses, which are less severe than those caused by Type 1 errors. The Type 2 error of applied models ranges from 6 to 99. A cut of 32 is considered to eliminate those models with a high Type 2 error.

c) **Type 1 error**: In this case study, Type 1 errors are crucial. Categorizing faulty products as good products costs a company significantly in terms of money and reputation. The Type 1 error of applied models ranges from 8 to 22. A cut of value 12 is considered to eliminate those models with high Type 1 error.

d) **Overall cost**: Since the weightage assigned to Type 1 and Type 2 errors is not the same, a cost model is used to evaluate the best model, keeping business objectives in mind. A cost of 225€ for the damage to the company from FPs and 100€ for the cost of FNs is estimated. The following equation is used to estimate the cost of misclassification:

$$\text{Cost} = \text{FN} \times 100€ + \text{FP} \times 225€$$

   The above thresholds are used to eliminate models that are outside the cut-off rates. See Figure 15.

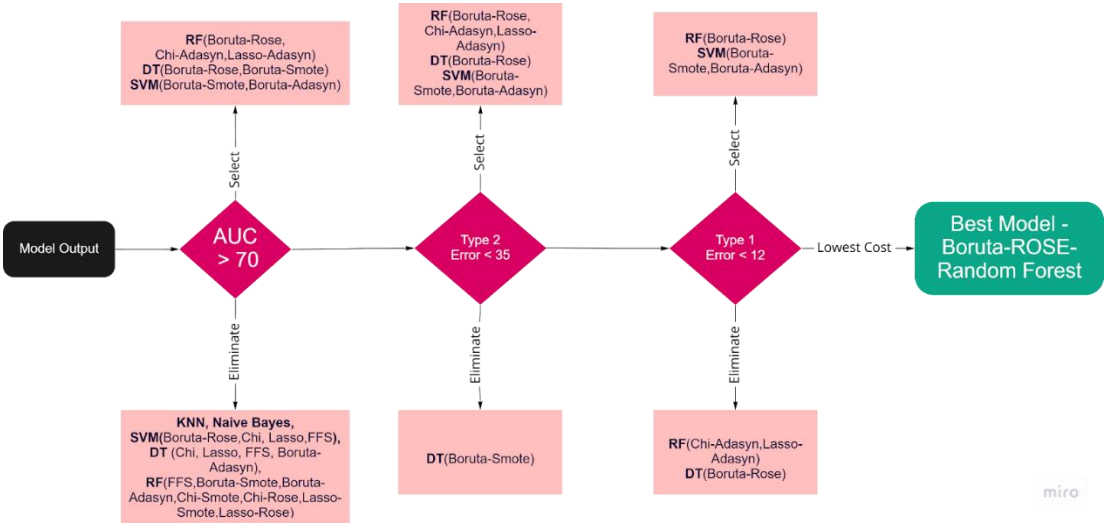

Figure 15: Best model selection using elimination

3.5 Overfitting analysis

Due to its low cost and high AUC, the Boruta–ROSE–RF combination is regarded as the best model among those tested. Following that, an overfitting analysis is run to determine at what depth the model begins overfitting when applying the RF classifier. As seen in Figure 16, the final model starts overfitting after reaching a max depth of 8. The same is taken into consideration and the parameters of the final model are changed accordingly.

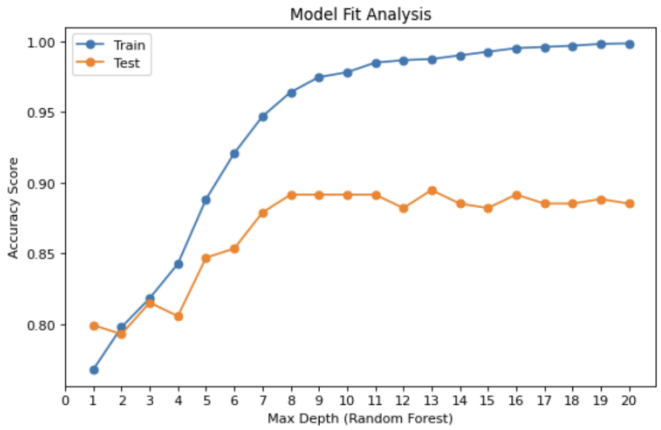

Figure 16: Model Fit Analysis

3.6 Feature importance

Once the models have been trained, it is possible to determine how each feature affects the classification outcome for each model. The Boruta–ROSE–RF is the best model. The significance of each feature is revealed by examining this model as depicted in Figure 17.

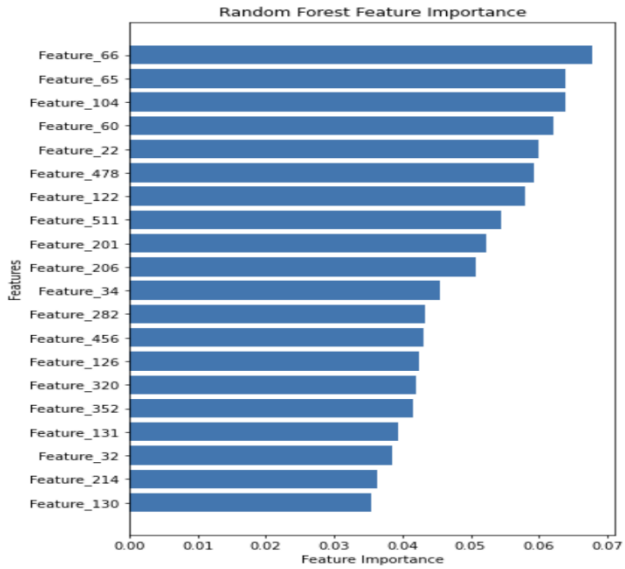

Figure 17: Feature importance in Boruta–ROSE–RF model

3.7 Summary

To summarise the pre-procession, the imported data is merged, preliminarily cleaned, and the timestamps are discarded. Following that, the data is split into a ratio of 80/ 20 for the training and test datasets, respectively. From both the training dataset and the test dataset, features with more than 50% of the data missing are removed, and outliers are replaced with the 3s boundary. Also, missing values are imputed using KNN with n_neighbor = 5. Thereafter, feature selection methods such as Boruta, Chi-square Test, Lasso Regression, and Forward Feature Selection are used to create new subsets. ROSE, SMOTE, and ADASYN are applied to the training dataset to balance the data. After the training data is standardized, the machine learning models Random Forest Classifier, SVM, and Decision Tree Classifier are trained on the training dataset. To identify the best combination, the models are evaluated on the test dataset using confusion matrix parameters and ROC curve/AUC. The hyperparameters are then tuned and an overfitting analysis is run to determine the best model.

The Boruta–ROSE–RF combination is regarded as the best model among those tested due to its low cost and high AUC. The flowchart in Figure 15 summarises the strategy used to obtain the best model by utilizing various model evaluation metrics.

4 Conclusion and recommendations

Fault detection is becoming increasingly crucial in the semiconductor industry. Traditional fault detection methods struggle to detect all problem characteristics. In the semiconductor manufacturing business, a good classification prediction model can be invaluable for defect identification. This paper investigated the SECOM dataset, which comprises data from a real-world semiconductor manufacturing factory.

During model fitting, many techniques with multiple steps for data imputation, data imbalance, feature selection, and classification were explored. A novel method for detecting if the model overfits the data was also introduced.

Finally, feature importance was assessed from the perspective of the best strategy, shedding light into failure reasons and identify the most crucial phases of the production line.

Recommendations

1. When it comes to data modelling, there are several ways that may be used. It is easy to become overwhelmed by the alternatives and lose sight of the processes. Using CRISP-DM enables self-orientation at all stages of the process and gives the ability to quickly restart steps if they fail. It also assures the capacity to progressively train the model and make a smart deal when it comes to various score criteria, as well as providing a solution to the presented research topic.
2. Before the data is cleaned or any other procedures are performed, it must be analysed to determine its quality.
3. The quality of the dataset influences a model's capacity to access and make smart judgments based on the input variables and features. Most models require a complete dataset, and missing data must be imputed. A crude feature selection strategy based on the fraction of missing values should be employed before outlier treatment and imputation. This guarantees that computationally expensive actions are not performed on characteristics that will not give any insights. Furthermore, following outlier treatment and imputation, more advanced feature selection algorithms cannot be utilized again. This guarantees that the model only learns from relevant data and not from noise.
4. Approaches on both outlier treatment and missing value imputation have a substantial impact on the outcome of feature selection. The key characteristics are not always (or even almost never) the same, therefore, it is critical to carefully select the best strategy for both of these operations.
5. It is critical to balance the data before training the model to ensure that the machine does not learn just from the majority class, which biases the predictions.

6. Scaling the data is required to get it into the same dimension. Large distances between data points, particularly in the case of SECOM, may have an influence on some models. Before training most models, it is critical to scale the data.
7. It is recommended to tune the hyperparamenters of the selected model to get the best result while also carrying out overfitting analysis to ensure the model does not overfit the training dataset.
8. Understanding the business is essential for evaluating the final model. Clarity about the model's purpose might help with selecting the correct measures to evaluate the model. It's easy to become lost in all of the sophisticated stats accessible. This may lead to experimenting with strategies that boost numerous vanity metrics while decreasing the KPIs that are most relevant to the firm. Choosing a metric based on business requirements and assessing the model on the basis of such metrics is critical. It is important to balance Type 1 and Type 2 errors, which may be accomplished with the use of a cost model.

List of literature

Adams, N.M. and Hand, D.J. (1999). Comparing classifiers when the misallocation costs are uncertain. *Pattern Recognition*, 32(7), pp.1139–1147. doi:10.1016/s0031-3203(98)00154-x.

Alafandy, K.A., Omara, H., Lazaar, M. and Al Achhab, M. (2022). Machine Learning. *Advances in Medical Technologies and Clinical Practice*, pp.83–113. doi:10.4018/978-1-7998-9831-3.ch005.

Auria, L. and Moro, R. A. (2008). Support Vector Machines (SVM) as a technique for solvency analysis. *SSRN Electronic Journal*, pp. 1–16. doi:10.2139/ssrn.1424949.

Azur, M.J., Stuart, E.A., Frangakis, C. and Leaf, P.J. (2011). Multiple imputation by chained equations: What is it and how does it work? *International Journal of Methods in Psychiatric Research*, 20(1), pp.40–49. doi:10.1002/mpr.329.

Benatti, N. (2019). *A machine learning approach to outlier detection and imputation of missing data*. [online] ideas.repec.org. Available at: https://ideas.repec.org/h/bis/bisifc/49-48.html.

Chawla, N.V., Bowyer, K.W., Hall, L.O. and Kegelmeyer, W.P. (2002). SMOTE: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, 16(16), pp.321–357. doi:10.1613/jair.953.

Chawla, N.V., Japkowicz, N. and Kotcz, A. (2004). Editorial: special issue on learning from imbalanced data sets. *ACM SIGKDD Explorations Newsletter*, 6(1), pp. 1-6. doi:10.1145/1007730.

Chen, Z., Zhou, L. and Yu, W. (2021). ADASYN−Random Forest Based Intrusion Detection Model. *2021 4th International Conference on Signal Processing and Machine Learning*, pp. 152-159. doi:10.1145/3483207.3483232.

Çolakoğlu, N. and Akkaya, B. (2019). Comparison of Multi-class Classification Algorithms on Early Diagnosis of Heart Diseases. *y-BIS Conference 2019: Recent Advances in Data Science and Business Analytics*. Mimar Sinan Fine Arts University Publications: 884, pp. 21-31. Available at: http://www.mi.imati.cnr.it/ettore/attached/y-BIS2019.pdf.

Cortes, C. and Mohri, M. (2003). AUC optimization vs. error rate minimization. *NIPS'03: Proceedings of the 16th International Conference on Neural Information Processing Systems*, pp. 313-320.

Dago, D.N., Kablan, G.A.J., Alui, K.A., Lallié, H.D., Dagnogo, D., Diarrassouba, N. and Giovanni, M. (2021). Normality Assessment of Several Quantitative Data Transformation Procedures. *Biostat Biom Open Access Journal*, 10(3), pp. 53-67. doi:10.19080/BBOAJ.2021.10.555786.

Das, D., Nayak, M. and Pani, S.K. (2019). Missing Value Imputation-A Review. *International Journal of Computer Sciences and Engineering*, 7(4), pp.548–558. doi:10.26438/ijcse/v7i4.548558.

Feurer, M. and Hutter, F. (2019). Hyperparameter Optimization. *Automated Machine Learning*, pp.3–33. doi:10.1007/978-3-030-05318-5_1.

Jamshidian, M. and Mata, M. (2007). Advances in Analysis of Mean and Covariance Structure when Data are Incomplete. *Handbook of Latent Variable and Related Models*, [online] pp.21–44. doi:10.1016/B978-044452044-9/50005-7.

Kang, H. (2013). The prevention and handling of the missing data. *Korean Journal of Anesthesiology*, [online] 64(5), p.402-406. doi:10.4097/kjae.2013.64.5.402.

Kerdprasop, N. and Kerdprasop, K., (2003). Data partitioning for incremental data mining. In *The 1st International Forum on Information and Computer Science*, pp. 114-118. Available at: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.101.7730&rep=rep1&type=pdf (Accessed: 03 August 2022).

Kuhn, M. & Johnson, K. (2016). *Applied predictive modeling*. New York: Springer, pp. 159–161.

Lubis, A.R., Lubis, M. and Khowarizmi, A. (2020). Optimization of distance formula in K-Nearest Neighbor method. *Bulletin of Electrical Engineering and Informatics*, 9(1), pp.326–338. doi:10.11591/eei.v9i1.1464.

Lunardon, N., Menardi, G., Torelli, N., Lunardon, M.N. and Suggests, M.A.S.S. (2021). Package 'ROSE' Type Package Title Random Over-Sampling Examples. [online] Available at: https://cran.hafro.is/web/packages/ROSE/ROSE.pdf.

Mohammed, R., Rawashdeh, J. and Abdullah, M., (2020). Machine learning with oversampling and undersampling techniques: overview study and experimental results. *2020 11th international conference on information and communication systems (ICICS)*, pp. 243-248. IEEE Xplore. https://doi.org/10.1109/ICICS49469.2020.239556.

Munirathinam, S. and Ramadoss, B. (2016). Predictive Models for Equipment Fault Detection in the Semiconductor Manufacturing Process. *International Journal of Engineering and Technology*, 8(4), pp.273–285. doi:10.7763/ijet.2016.v8.898.

Roye, K., Hilton, D. and Smart, C. (2021). Dealing with Missing Data – The Art and Science of Imputation. *ICEAA 2021 Online Workshop*, pp. 1-22.

Sen, P.C., Hajra, M. and Ghosh, M. (2019). Supervised Classification Algorithms in Machine Learning: A Survey and Review. *Advances in Intelligent Systems and Computing*, pp. 99–111. doi:10.1007/978-981-13-7403-6_11.

Shearer, C. (2000). The CRISP-DM Model: the new blueprint for data mining. *Journal of data warehousing*, 5(4), pp. 13-22.

Wendler, T. and Gröttrup, S. (2021). *Data Mining with SPSS Modeler*. Cham Springer International Publishing, pp. 1199.

Wirth, R. and Hipp, J. (2000). CRISP-DM: Towards a standard process model for data mining. [online] *Proceedings of the 4th international conference on the practical applications of knowledge discovery and data mining*, pp. 29-39. Available at: http://www.cs.unibo.it/~danilo.montesi/CBD/Beatriz/10.1.1.198.5133.pdf (Accessed: 01 August 2022).

Ling, C.X., Huang, J. and Zhang, H. (2003). AUC: a Statistically Consistent and more Discriminating Measure than Accuracy. *Proceedings of the 18th international joint conference on Artificial intelligence*, pp.519–524.

Zhang, S., Li, X., Zong, M., Zhu, X. and Cheng, D. (2017). Learning k for kNN Classification. *ACM Transactions on Intelligent Systems and Technology*, 8(3), pp.1–19. doi:10.1145/2990508.