Mini Thesis

Fault Detection in Semiconductor Manufacturing Process Using Data Mining Techniques (SECOM Case Study)

From

Binsy Babu (s0572046)

Hima Bindu Sariputi (s0572704)

Lahiru Pathiranage (s0572031)

Sharvari Rajendra Sawant (s0572041)

Wenzhang Xu (s0572037)

Date: 17.07.2020

Project Management and Data Science (MPMD)

Supervisor: Prof. Dr. Tilo Wendler

Hochschule für Technik
und Wirtschaft Berlin

University of Applied Sciences

Index

List of figures

List of tables

List of abbreviations

CRISP-DM: Cross-Industry Standard Process for Data Mining

PCA: Principle Component Analysis

NA: Not Applicable

FP: False Positive

FN: False Negative

TP: Tue Positive

TN: True Negative

FPR: False Positive Rate

FNR: False Negative Rate

KNN: K Nearest Neighbour

SVM: Support Vector Machine

RF: Random Forest

NB: Naïve Bayes

ROSE: Random Over-Sampling Examples

ADASYN: Adaptive Synthetic

SMOTE: Synthetic Minority Oversampling Technique

AUC: Area Under Curve

1 Introduction

Manufacturing of semiconductor is highly technology-intensive, it involves hundreds of steps which fall into 4 main categories: production of silicon wafers, fabrication of integrated circuit onto a silicon wafer, assembling integrated circuit into packaging, and testing of final products (May and Spanos, 2006, p. 1-24).

As the semiconductor industry demands absolute precision in production, the existing complex manufacturing processes are accompanied by numerous real-time sensors, aiming to monitor the production processes and maximise the efficiency in production control as well as equipment state (Munirathinam and Ramadoss, 2016, p. 273). However, the wafer fabrication alone involves over 500 steps, the sheer volume of data collected from real-time monitoring is vast, and this renders the traditional data analytical methods impractical. Therefore, machine learning algorithms are introduced into the semiconductor industry, it enables the computer to adaptively learn and build a predictive model based on empirical data to detect faulty products (Munirathinam and Ramadoss, 2016, p. 273). The SECOM dataset is donated by McCann and Johnston for researchers to experiment with different classification models (archive.ics.uci.edu, 15-Jul-20). One of the research papers used 'mean' and 'in-painting KNN' to impute missing data, and classified with a set of different models (Salem, Taheri, and Yuan, 2018, p. 30); some focused on different features selection and case balancing techniques(Kerdprasop and Kerdprasop, 2011, p. 399); another group of researchers proposed a feature selection technique by combining SME knowledge, correlation analysis, and PCA, they also experimented with several different classification models (Munirathinam and Ramadoss, 2016, p. 276). Overall, researchers and people of interest have been manoeuvring the SECOM dataset to find the best combination of techniques for an accurate and precise predictive model.

The primary aim of this study is to build a parsimonious model that can accurately predict the quality classes of the semiconductors. It is expected that our model should be able to handle missing values, imbalanced data, and other real-world issues.

To achieve this objective, the following research questions are proposed:

- Which data mining approaches and modeling techniques works best in CRISP-DM methodology to build a highly reliable model for fault detection in semiconductor manufacturing
- Which machine learning techniques, feature selection, and case balancing techniques forms an apt combination to generate a comprehensible model to minimize the error and maximize the accuracy of fault detection during SECOM semiconductor manufacturing?
- What are the best alternatives in dealing with imbalanced classes?

- Which steps are necessary to reduce false predictions and how to improve the model quality?
- What are the criteria to assess the model and how are they prioritized to accomplish the business objective?

To address these research questions CRISP-DM methodology is used which provides a standard approach for data mining. This paper provides theoretical explanation of major steps in CRISP-DM process. Implementation of these process are demonstrated in a detailed manner including the results of multiple models and their combination with feature selection and case balancing. Selection criteria are defined to obtain the best model, eliminating the insignificant ones. Application of bootstrap approach to find a suitable model based on cost approach is also exhibited. Finally, a conclusion is drawn with regard to the summary of this study, the answers of the research questions, and the recommended best practices.

2 CRISP DM Model

CRISP-DM model is a machine learning process model, it describes commonly used approaches that machine learning experts use to tackle problems (Figure 1). The main advantage of CRISP-DM is that it can be implemented in any data science project not with-standing its domain or destination (IBM, no date, p. 1). When starting a project, teams often suffer from the lack of domain knowledge or ineffective models of data evaluation they have. Thus, a project can become successful only if a team manages to reconfigure its strategy and can improve the technical processes it applies. CRISP-DM model offers a well-defined structure, it describes both the typical phases of a data science project as well as the data mining lifecycle (IBM, no date, p. 1).

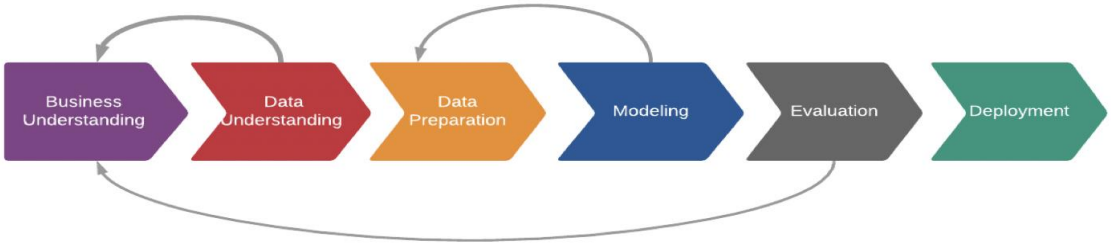

Figure 1: Visualisation of CRISP-DM process

2.1 Business Understanding and Data Understanding

The goal of the initial phase is to understand and gather information regarding the project objective from a business aspect, which is then converted into the definition of data mining problems. The main objective of this project is to build an accurate model to predict product quality. Modern semiconductor industries collect sensor data directly, such vast data is beyond the capability of traditional quality control methods, hence machine learning is a necessary procedure (Kerdprasop and Kerdprasop, 2011, p. 399). However, the predictions are not 100% accurate, it usually carries two types of errors: failed products predicted to be pass, pass products predicted to be failed, the two errors incur different cost. Therefore, the business goal here is to minimise the monetary cost of incorrect predictions, which can be translated to the data mining goal of this study: to maximise the accuracy of predictions and lower the overall cost by balancing the trade-off between two types of errors.

Understanding the characteristics of the dataset is one step to successfully build a model for data mining. SECOM dataset consists of 590 features, timestamp, primary ID, and the quality class. There are 1567 cases in total, 95 of them are failed cases which are encoded as 1 and the remaining pass cases are encoded as 0.

2.2 Data Preparation

Data preparation is a lengthy process, but it is an essential prerequisite to produce meaningful insights and eliminate bias resulting from poor data quality. Following data preparation steps are used to process the raw data from the dataset to obtain the most essential features which can be used in model building.

Step1: Data Partitioning:

Partitioning data into training and test data helps in developing a more reliable model. The training set is used for discovering data pattern, while test data is used to demonstrate how well the model performs on unseen data (Kerdprasop, 2003, p. 114).

Step2: Feature Removal:

Features primarily having values as NA (missing values) would not contribute much towards building a good statistical model and hence can be removed (Garson, 2015, p. 10). Also, variance tells the measure of how far is the set of numbers spread from their mean. Having zero or near zero means the feature column values are constant or almost similar. These features would not contribute to the statistical model as it lacks much valuable information. Features having zero or near-zero variance can also be removed.

Step3: Outlier detection and removal:

Outliers are extreme values that deviate significantly from other observations. Machine learning algorithms are sensitive to outliers. Outliers can skew and mislead the training process making the models less accurate and giving poor results (machinelearningmastery.com, 2013). Eliminating the outliers might result in losing the data, which probably could have an impact on the data model. Hence the outliers can be replaced with NAs or 3s values, they are identified using the 3s rule where anything above mean+3s and anything below mean-3s is considered as an outlier.

Step4: Imputation of missing values:

Many real-world datasets may contain missing values for various reasons. They are often encoded as NAs, blanks, or any other placeholders. Dataset that has many missing values can impact the machine learning model's quality. One way to handle this problem is to get rid of the observations that have missing data (Thanamani, 2012, p. 5). However, this may result in loss of valuable information for the modeling process. A better strategy would be to impute the missing values. One of the methods of imputation is to replace missing values with mean/median values, while another method is using the K Nearest Neighbour (KNN). KNN algorithm uses 'feature similarity' to predict the values of any new data points. This means that the new point is assigned a value based on how closely it resembles the points in the training set (towardsdatascience.com,2019). Taking a low k-value may increase the influence of noise, and the results are going to be less

generalized. On the other hand, taking a high k will tend to blur local effects (Thanamani, 2012, p. 6).

Step5: Feature Reduction: PCA:

Principal component analysis (PCA) is a multivariate technique that analyses dataset which consists of inter-correlated dependent variables. The main goal of PCA is to extract important variables and represent them with newly defined variables called principal components (Abdi and Williams, 2010, p. 433). In addition, the Kaiser-Meyer-Olkin (KMO) test is a measure of sample adequacy and it is performed on a dataset to assess the eligibility for PCA. Also, Kaiser and Rice mentioned that when sample adequacy is below 0.5, it means that the data set is not suitable for factor analysis (Kaiser and Rice, 1974, p. 112). Finally, Kaiser Guttman's rule can be used to select the important features where Eigen values greater than 1 are considered as "above average" components (Kaiser, 1960, p. 145).

Step6: Feature selection:

The feature selection is an important technique to extract the most important features that can be used for the predictive modeling process. This technique disregards features or combinations of features which create noise or impact the model performance negatively (medium.com, 14-Jul-20). There are two methods for performing feature selection, supervised and unsupervised. The supervised method includes class information, i.e. it evaluates the relationship between each variable (features) and target variable (class) and selects those which have the strongest relationship with the class variable (Haar et al., no date, p. 2). Unsupervised methods use features and their characteristics to select important features. Common supervised methods include Boruta, Gain Ratio, and Chi-Square etc.

Chi-square feature selection is based on the measure of association between class and features, this is indicated by Cramer's V coefficient (Gervasi et al., 2016, p. 625). Chi-square test is applied between each feature and the target variable to obtain Cramer's V coefficient. Features with Cramer’s V greater than 0.5 are selected as it suggests a strong relation between feature and class variable (acastat.com, 16-Aug-15). The other method which has proven beneficial is the Gain ratio which is an enhancement of Information-Gain technique, it takes account of the number and the size of the branches of a decision tree when choosing an attribute, it gives the weight of the features based on the entropy of the distribution of instances into branches of decision trees (R., M.L. and S., 2011, p. 203). Boruta is a commonly used wrapper based feature selection technique built around Random Forest (RF) (Kursa and Rudnicki, 2010, p. 2). It trains the RF classifier and applies a feature importance measure to assess the importance of each feature. It selects features which have higher importance than its shadow features and discard those which are unimportant (analyticsvidhya.com, 2016).

2.3 Data Modeling

The essential features which are obtained from the data preparation steps are used in the model building phase. This phase includes case balancing, scaling, and normalizing of the trained data which serves as preparatory steps for model building. Machine learning algorithms are used to train the model and test set is used to assess the performance of the model.

Case Balancing

A dataset is considered imbalanced if there is more than 10% disparity in the frequencies of the observed classes, and this can cause a significant negative impact on model fitting which leads to a biased model (Chawla, 2002, p. 322). Case balancing such as ROSE, SMOTE, ADASYN is performed to mitigate this issue.

ROSE generates synthetic data through a smoothed bootstrap-based technique. The idea of ROSE is to refine the issue of overfitting and improve the capability to generalisation, by creating cases that have not been observed in the data (Lunardon, 2014, p. 79). SMOTE uses an over-sampling approach in which the minority class is over-sampled by generating "synthetic" examples instead of over-sampling with replacement (Chawla, 2002, p. 326). ADASYN is an extension of SMOTE, it weighs distribution of minority class according to the level of difficulty in learning, i.e. more synthetic data is generated for minority classes that is harder to learn (imbalanced-learn.readthedocs.io, 2019).

Scaling and Normalizing

In the pre-processing stage, scaling and normalizing techniques are applied to improve forecasting. There are several techniques available such as Min-Max scaling, Z-Score, Box-Cox, and Log (Patro and sahu, 2015, p. 20–22). Scaling and normalization is important for certain machine learning algorithms such as KNN (Kotsiantis, Kanellopoulos and Pintelas, 2007, p. 4106).

Model Building/ Machine Learning Algorithms

Machine Learning algorithms can be categorised into three types, i.e. supervised learning, unsupervised learning, and reinforcement learning (kdnuggets.com, 2020). Some of the basic machine learning algorithms which belong to supervised learning are Random Forest (RF), Naïve Bayes (NB), KNN, and Support Vector Machine (SVM). RF produces, a best result most of the time even without tuning of hyperparameters (Donges, 2020). It is one of the most commonly used algorithms for both classification and regression tasks. RF build multiple number of decision trees and merges them together in order to get more accurate and stable predictions (Breiman, 2001, p. 1). Because it is a combination of multiple decision trees, it also adds additional randomness while growing the trees and searches for best feature amongst random subset of features, instead of searching among the most important feature while splitting the node (Breiman, 2001, p. 1).

Another machine learning algorithm is NB, which is based on Bayes theorem of probability to predict the class of unknown data sets. The NB classifier greatly simplifies learning by assuming that features are independent given class (Langley, 2013, p. 399). The algorithm converts the data set into a frequency table, then it calculates different probabilities, lastly, it uses Naïve Bayesian equation to calculate the probabilities of each class, the highest probability would be the outcome (Langley, 2013, p. 399).

SVM performs classification by finding the hyper-plane that differentiates the two classes very well and solves both classification and regression problems (analyticsvidhya.com, 2017). In classification, hyperlanes can easily separates the classes(like positive & negative). Initially draw the random hyperlane then select the datapoints from each class close to this. These data points to the hyperlane are defined as Support Vectors. The distance between the support vectors and the hyperlane is defined as Margin. Hence, SVM is used to classify the data based on the maximum distance between hyperlane and the support vectors is high (analyticsvidhya.com, 2017). Another supervised algorithm is KNN. It takes account of similarity (distance, proximity, or closeness) to perform an operation. KNN considers the Euclidean distance, and it computes the distance between each data point and test data. Then, it finds the probability of these points being similar to the test data and classifies it based on which points share the highest probabilities (Bzdok, 2018, p. 1).

Hyperparameter Tuning:

Tuning the parameters is a critical step in training models. It is a trial-and-error procedure by which hyperparameters are modified and applied to identify the best model. These parameters can be optimised for better model accuracy. The three commonly used techniques are random search, grid search, and manual search. Manual search requires less technical overhead while offering researchers insights about the hyperparameters of a model; on the other hand, grid search is easier to implement and usually offers better results than manual search in the same amount of time, grid search is also more reliable in lower-dimensional spaces; in comparison, random search sacrifices a small portion of accuracy but it requires much less computational expenses, hence it is a particularly efficient option for high dimensional spaces (Bergstra and Bengio, 2012, p. 283).

The set of hyperparameters giving the best results are used to build the model on the training set and the goodness of the model is tested on the test set. The model results are assessed to select the best models. The final model is evaluated by aligning the result with business objectives. It is then deployed if the results effectively and efficiently satisfy business needs.

3 Implementation

The CRISP-DM methodology is implemented on the SECOM dataset using R in version 3.6.3. The below figure exhibits each step performed. The subsequent chapters provides a detailed explanation of the implementation process and the results

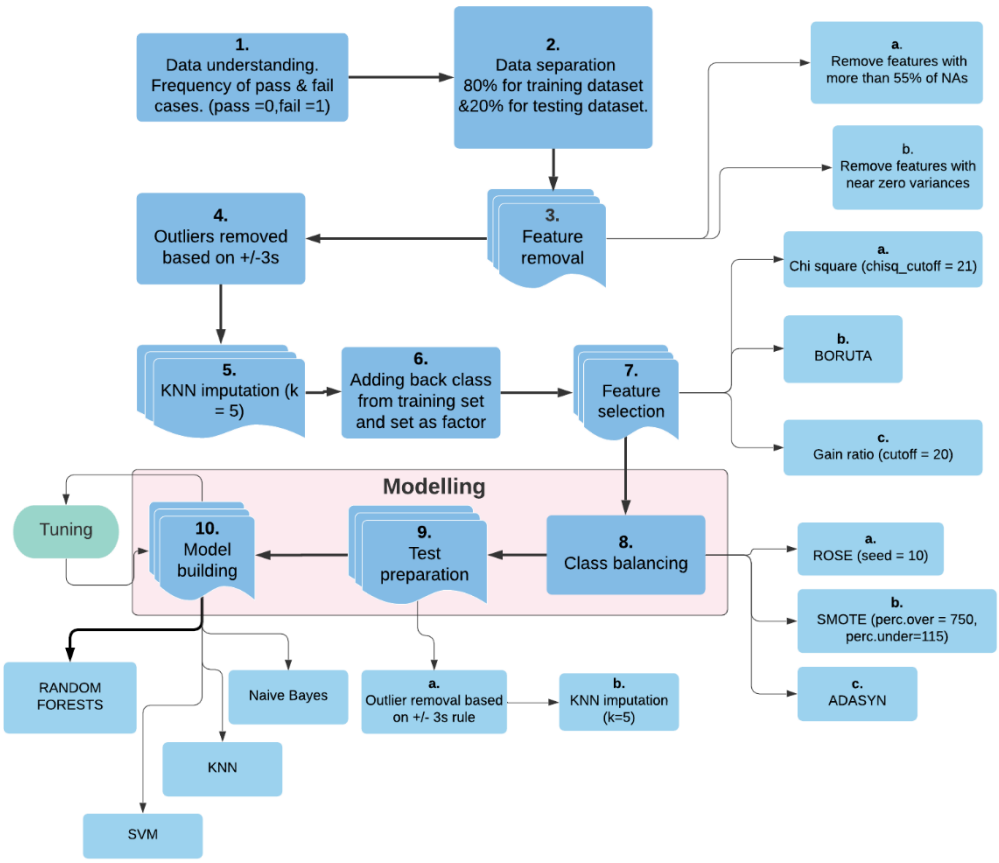

Figure 2: Process steps performed to implement CRISP-DM in R programming

3.1 Data preparation

The original data is divided into training and test dataset in a proportion of 80% and 20% respectively. The ratio of fail to total cases remain the same to ensure the representativeness of the subsets, which are 5.9% and 6.7% for the training and testing set respectively. After partitioning, features having more than 55% missing values are removed following which resulted in 566 features. Based on near- zero variance, 128 features are removed, the final dataset now has 438 features. Furthermore, outliers are detected and replaced with NA's, and the missing values are imputed using KNN with k-value equals to 5.

Feature Reduction: The PCA approach on SECOM dataset resulted in 438 principal components with very low percentages of variances. In the SECOM dataset, to explain at least 80% variance,

99 principal components are required. According to Kaiser Guttman's rule, SECOM dataset results in 119 variables. Based on above criteria, PCA results are incapable of building a parsimonious model. Hence, PCA approach is not considered as a suitable feature reduction method for the SECOM dataset.

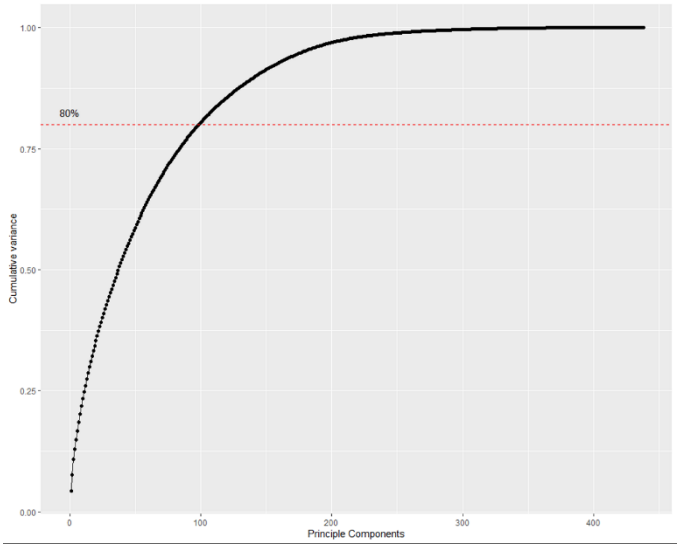

Figure 3: Cumulative variances of principle components

Feature selection: Boruta is performed on the imputed data using function 'Boruta()' available in R package 'Boruta'. 'maxrun' function in R is set to 300, this is decided as a compromise between efficiency and accuracy through multiple trial runs. 'getSelectedAttributes' function is used to get the names of the final selected features. This process results in 17 features out of 438 features. This is saved as a separate dataset for further processing. The importance graphs below is plotted showing the selection process performed by Boruta. Green lines correspond to important features, red to rejected ones and blue to minimal, average and maximal shadow feature importance respectively (Kursa and Rudnicki, 2010, p. 9)

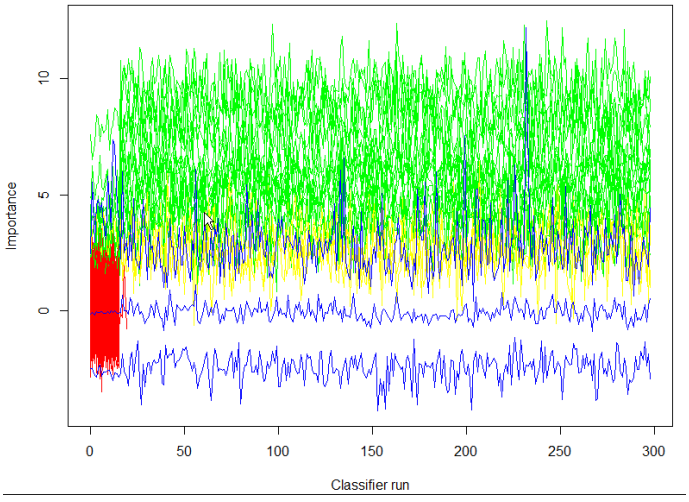

Figure 4: Boruta run importance plot

'chi.squared' function of 'Fselector' library is used to bin the continuous variables and weigh them in terms of Cramer's V. 21 features are selected with Cramer's V greater than 0.4 suggesting strong association with class variable. 'gain.ratio' function is used to select features based on gain ratio. Top 20 features are selected with comparatively higher gain ratio and saved as a separate dataset. Consequently, feature selection method Boruta, Chi-square and gain ratio give 17, 21, and 20 features respectively which are further used for modeling.

3.2 Model building

Based on Feature Selection outputs, data modeling is implemented on key features resulted from Boruta, Chi-square and Gain ratio datasets. A model built on these datasets provides biased results because of unequal proportion of classes. Hence, case balancing is performed.

Case Balancing:

For SECOM dataset, the number of pass cases (Majority) for training dataset is 1180 and fail cases (Minority) is 74. Here minority class proportion is less than 10%, hence balancing is performed. There are various methods to balance the dataset based on class such as under sampling, over sampling, ROSE, SMOTE, ADASYN. Due to various disadvantages such as overfitting (oversampling) and underfitting (undersampling), balancing techniques such as ROSE, SMOTE and ADASYN is performed.

- By implementing ROSE on SECOM training dataset, Majority class is resampled to remove modules (rows) to a ratio of nearly 50% (Under-sampling) and Minority class is resampled to repeat modules to a ratio of nearly 50% (Over-sampling).
- By performing SMOTE and ADASYN on training dataset, additional minority class representatives are synthetically generated and that leads to a balanced dataset (Journal of Artificial Intelligence Research, 2020) with equal proportion of pass cases and fail cases.

Preparation of Test dataset: After case balancing, it is important to prepare the test data set for testing where the raw test data set consists of untreated missing values and outliers. In the first step of test data preparation, outliers are identified using 3s rule and replaced them with NAs or upper/lower boundary values. In the second step, all the NA values are imputed using the KNN approach with the same k value used for the training dataset.

3.2.1 Implementation results

Confusion Matrix: A confusion matrix is a summary of the prediction results on a classification problem (Wang, 2005, p. 52).

Confusion Matrix			Actual class		
			Pass	*Fail*	
			0	1	
Predicted Class	*Pass*	0	True Positive (TP)	False Positive (FP)	Predicted Positive
	Fail	1	False Negative (FN)	True Negative (TN)	Predicted Negative
			Actual Positive	Actual Negative	

Table 1: Confusion Matrix configured based on Secom dataset

The target class in SECOM dataset has two categories, the majority of the samples are positive values (Pass), whereas the minority of them are negative class (Fail). In Secom dataset, there are two predicted classes: Pass (0) & Fail (1). Here, Prediction of the product to be not faulty is a " Pass" and prediction of the product to be faulty is a "Fail ". Confusion matrix comprises of:

- TP cases which are correctly predicted as 'Pass'

- TN cases which are correctly predicted as 'Fail'

- FP cases which are incorrectly predicted as 'Pass'

- FN cases which are incorrectly predicted as 'Fail'

Random Forest: RF modeling is applied on balanced datasets (ROSE, SMOTE and ADASYN) and predict on test dataset to generate the confusion matrix results and other important metrics such as sensitivity, specificity, F1 score and AUC.

Confusion Matrix			Actual class		
			True	*False*	
			0	1	Total
Predicted class	*True*	0	275	13	288
	False	1	17	8	25
		Total	292	21	

Table 2: Confusion matrix of RF-Boruta-ROSE model

In the above given sample confusion matrix, the model (RF-Boruta-ROSE) correctly predicted pass class 275 times and correctly predicted fail cases 8 times. Similarly, the model incorrectly predicted fail class 13 times and incorrectly predicted pass cases 17 times. Hyper parameter tuning is performed on RF in order to find the best parameters which increase the accuracy of the model. Applying these tuned parameters into the RF function changed the AUC from 0.53 to 0.716.

Based on confusion matrices of all the models, the following graph is built based on number of FP's and FN's. FP's are known as Type 1 error and FN's are known as Type 2 error. A model is considered good when it has a smaller number of FP's and FN's. Following this, the below graph shows, ROSE-Boruta, SMOTE-Boruta and Chi-square-ADASYN are considerable models under RF with less FP's and FN's.

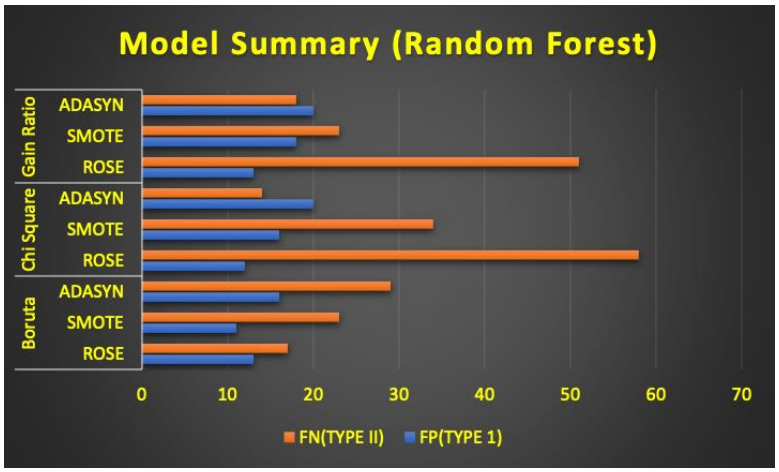

Figure 5: Summarization of all models based on FP's and FN's

Naïve Bayes (NB): While implementing NB, 10-Fold cross validation is used on training data set to obtain the best model. The training data set is transformed using Box Cox transformation and then centred and scaled before giving input to this model. NB produces the best result for Boruta/ROSE combination with AUC value of 0.718. The model correctly predicts 276 pass cases and 8 fail cases. Similarly, the model incorrectly predicts 16 times fail class and 16 times pass cases.

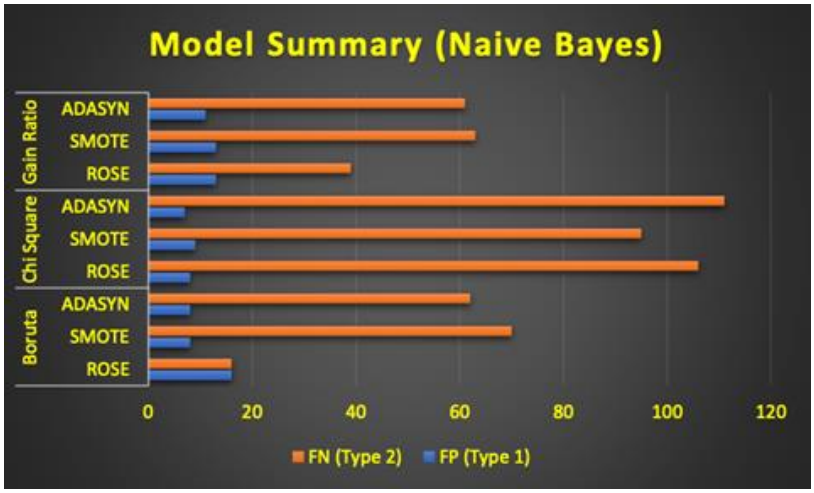

Figure 6: Summarization of all models based on FP's and FN's for Naïve Bayes

The above graph shows, ROSE-Boruta, SMOTE-Boruta and Chi-square-ADASYN are considerable models under NB function with less FP's and FN's. It can be observed that NB produces best result for Boruta /ROSE combination.

SVM and KNN: SVM and KNN are supervised machine learning models. For SVM and KNN, training dataset is centred and scaled. Linear SVM classifier is built using SVM Linear Kernel and to build non-linear classifier, polynomial and radial kernel is used. Total of 9 combinations are generated using the SVM classifier, Boruta and balancing techniques. Out of all the combinations Boruta/ADASYN with SVM linear gives better results with AUC value of 0.649. Similarly, for KNN, better results are produced from Boruta-ROSE with AUC 0.57.

3.2.2 Comparison of machine learning models

To choose the best model out of many, classification performance measures are used. The selection criteria are defined for important measures defining the success of the model as given below:

a) *AUC*: It tells how well the model can distinguish between classes 'Pass' and 'Fail'. Higher the AUC, better the model is at predicting pass as 'Pass' and fail as 'Fail' (analyticsvidhya.com, 2020). Applied model results have AUC ranging from 0.5 to 0.7. A cut off value of 0.65 is used to select classifiers with comparatively better ability to distinguish between classes.

b) *Type 1 error*: Type I error is of primary importance in this case study. Categorizing faulty product as good costs a lot to a firm in terms of money and reputation. Moreover, it can cause further damages to equipment using such product or semiconductors. After applying AUC cut off value of 0.65, Type 1 error ranges from 8 to 20. A cut of value 16 is considered to eliminate those with high Type 1 error.

c) *Type 2 error*: Semiconductors and its manufacturing is expensive. The losses incurred here is manufacturing losses which is lower compared to those caused by Type 1 error. After applying Type 1 error cut off value as 16, Type 2 error ranges from 31 to 111. A cut of value 30 is considered to eliminate those with high Type 2 error.

The above criteria results in four best model as given in the Table3. On comparing these models, RF-Boruta-ADASYN combination is discarded due to relatively high Type II errors. NB-Boruta-ROSE model combination has high False Positives compared to the remaining models. Both NB-Boruta-ROSE and RF-Boruta-ADASYN has high FPR and comparable sensitivity as shown in Figure 7. Therefore, these models are eliminated to arrive at the final best two models: RF algorithm applied on Boruta selected features with ROSE and SMOTE case balancing.

Model	Feature Selection	Case Balancing	AUC	FP (TYPE I)	FN (TYPE II)	FPR	FNR	Sensitivity
Random Forest	Boruta	ROSE	0.7164	13	17	0.6190	0.0582	0.9418
		SMOTE	0.7689	11	23	0.6111	0.0787	0.9212
		ADASYN	0.7275	16	29	0.7619	0.0993	0.9007
Naïve Bayes	Boruta	ROSE	0.7183	16	16	0.7619	0.0547	0.9452

Table 3: Model result metrics of best models

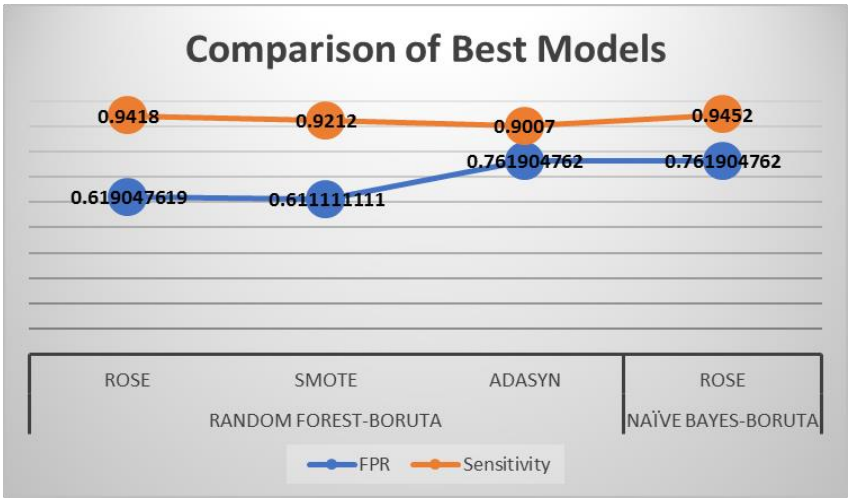

Figure 7: False Positive Rate and Sensitivity comparison of best models

It is clearly evident that both the RF model with ROSE and SMOTE combinations are comparable with respect to Type I errors, AUC, FPR and sensitivity. Hence it is difficult to arrive at one best model. Therefore, we performed bootstrap cost approach to obtain the total loss which is incurred on selecting either of these models.

Bootstrap

In order to validate the top model candidates, the bootstrapping technique applied to assess the cost variations. Each time the R algorithm runs it has unique training and test data set and results in a different cost based on the combination. Cost is a measurement for false predictions and the cost associated with it. Below equation used to calculate the cost. Moreover, a random estimate of 320 € for the damage for the company from FPs and 200€ estimate for FNs.

$$Cost = FP \times 320€ + FN \times 200€$$

If the model produces lower mean and lower standard deviation which implies that the model produces accurate and steady results compared to others.

Random Forest	Mean	SD	Median	Min	Max	Range	Skew	Kurtosis
Boruta/SMOTE	9352	2024	9100	5880	12960	7080	0.21	-0.82
Boruta/ROSE	7918	2004	8020	4440	10760	6320	-0.3	-1.16
Boruta/ADASYN	8320	1159	8100	6440	10400	3960	0.18	-1.39

Table 4: Bootstrap results of top candidates

According to the cost approach results, Boruta/ROSE/RF combination produces lower mean and Boruta/ADASYN/RF combination produce lower standard deviation. To sum up, RF/Boruta/ROSE combination is considered as the best model amongst all the given model due to minimized cost and relatively low standard deviation. The below flow chart summarizes the strategy used to get the best model using various model result metrics.

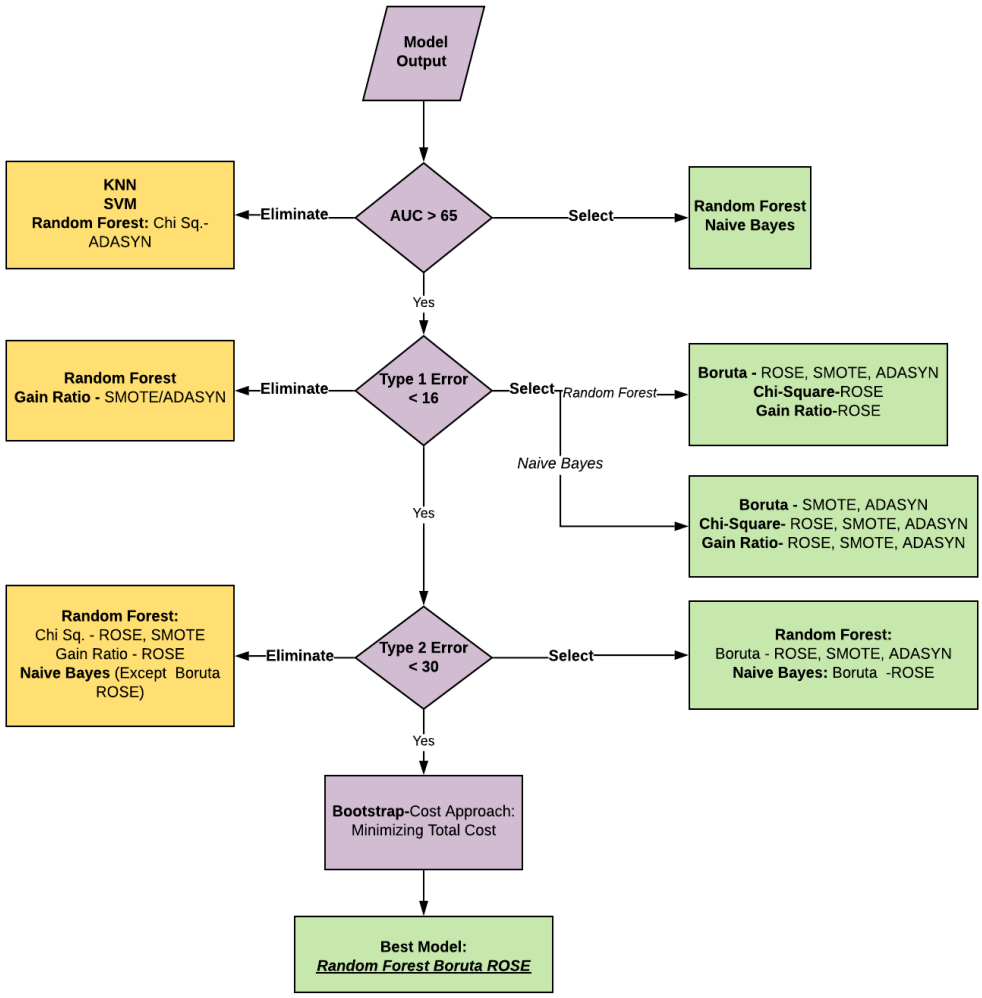

Figure 8: Model selection strategy to attain the best classification prediction model

4 Conclusion

Semiconductor manufacturing process is capital intensive and involves highly complex data which necessitates effective fault detection using data mining techniques. CRISP-DM methodology and optimal data mining techniques are used to attain the best classifier models, which also provides solutions to the proposed research question. This study confirms that each technique from data preparation to case balancing can have an impact on the results of the final model. Methods such as PCA, chi square and gain ratio fail to perform well, however Boruta is the crucial feature selection method which delivers significant results. Different combination of feature selection with case balancing technique is used to train the model using multiple machine learning algorithms which is then tested on test set to give model result metrics. K-fold validation, normalization and fine tuning of parameters are implemented to reduce false predictions. Approximately equal sensitivity and specificity reveals that the models are comparable in terms of correctly classifying the class variables, and hence cannot be a significant measure in identifying the best model. Therefore, models are evaluated based on AUC, Type I errors, false positive rate and Type II errors as they are of primary concern in semiconductor industry. This selection strategy concluded two best model of RF algorithm with ROSE, SMOTE combination. To find a trade-off between these two models, bootstrap cost estimation is used aiming to minimize the monetary loss incurred. This approach gave the best model as RF Boruta ROSE.

To attain this model, CRISP-DM methodology is used in this study: It is recommended to partition the original data into training and test datasets where the training set is used for discovering data pattern, test data is used to demonstrate how well the model performs on an unseen data. In training dataset preparation, It is preferred to remove features with a majority of missing values, as these features don't contribute any added value to the model. However, Features with minimal number of missing values can be handled by imputing them using KNN imputation method. It is recommended to implement feature selection right after imputation in order to retrieve the minimal and most important features from vast dataset, which affects the model. Based on experimental results in feature selection, Boruta is considered as the best technique with key features. Before proceeding with model building, the dataset should consists of equal proportion of pass and fail cases. Hence, it is suggested to perform case balancing. To improve model performance, parameter tuning can be considered as one of the best practices.

Reference list

May, G. S. and Spanos, C. J. (2006) Fundamentals of Semiconductor Manufacturing and Process Control. John Wiley & Sons. doi: 9780471790273.

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

archive.ics.uci.edu, https://archive.ics.uci.edu/ml/datasets/SECOM (15-Jul-20), 15-Jul-20 (Accessed: 15-Jul-20).

Kerdprasop, K., and Kerdprasop, N. (2011). Feature Selection and Boosting Techniques to Improve Fault Detection Accuracy in the Semiconductor Manufacturing Process, [online]. IMECS. Available at: http://www.iaeng.org/publication/IMECS2011/IMECS2011_pp398-403.pdf [Accessed 25 Jul. 2018]

Salem, M., Taheri, S. and Yuan, J.-S. (no date) 'An Experimental Evaluation of Fault Diagnosis from Imbalanced and Incomplete Data for Smart Semiconductor Manufacturing', BDCC, 2(4), p. 30. doi: 10.3390/bdcc2040030.

IBM 'IBM SPSS Modeler CRISP-DM Guide'. Available at: https://www.google.com/search?q=ibm+spss+modeler+crisp-dm+guide&oq=IBM+SPSS+Modeler+CRISP-DM+Guide&aqs=chrome.0.0l2.391j0j7&sourceid=chrome&ie=UTF-8 (Accessed: 16-Jul-20).

Kerdprasop, N. (2003) Data Partitioning for Incremental Data Mining, 10 July. Available at: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.101.7730&rep=rep1&type=pdf, pp. 114-118 (Accessed: 7 October 2020).

Garson, D. (2015) *MISSING VALUES ANALYSIS & DATA IMPUTATION,* 18 December. Available at: http://www.statisticalassociates.com/missingvaluesanalysis_p.pdf , pp. 1-26. (Accessed: 14 July 2020).

machinelearningmastery.com, https://machinelearningmastery.com/how-to-identify-outliers-in-your-data/ (2013), 30-Jun-2020 (Accessed: 14 July 2020)

Thanamani, A. (2012) 'K-Nearest Neighbor in Missing Data Imputation'. Available at: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.375.925&rep=rep1&type=pdf, pp. 1-3 (Accessed: 15 July 2020).

towardsdatascience.com, https://towardsdatascience.com/6-different-ways-to-compensate-for-missing-values-data-imputation-with-examples-6022d9ca0779(2019), 16-Apr-2020 (Accessed: 16 July 2020).

Abdi, H. and Williams, L.J. (2010) ‘Principal component analysis’, Wiley Interdisciplinary Reviews: Computational Statistics, 2(4), pp. 433–459. doi: 10.1002/wics.101

Kaiser, H.F. and Rice, J. (1974) ‘Little Jiffy, Mark Iv’, Educational and Psychological Measurement, 34(1), pp. 111–117. doi: 10.1177/001316447403400115

Kaiser, H.F. (1960) ‘The Application of Electronic Computers to Factor Analysis’, Educational and Psychological Measurement, 20(1), pp. 141–151. doi: 10.1177/001316446002000116

Haar, L. et al. ‘Comparison between Supervised and Unsupervised Feature Selection Methods’, p. 2. Available at: http://www.insticc.org/Primoris/Resources/PaperPdf.ashx?idPaper=73853.

Gervasi, O. et al. (eds.) (2016) Computational science and its applications -- ICCSA 2016. Part V: 16th International Conference, Beijing, China, July 4-7, 2016, Proceedings / Osvaldo Gervasi, Beniamino Murgante, Sanjay Misra, Ana Maria A.C. Rocha, Carmelo M. Torre, David Taniar, Bernady O. Apduhan, Elena Stankova, Shangguang Wang (eds.). Switzerland: Springer (LNCS sublibrary. SL 1, Theoretical computer science and general issues, 9790).

acastat.com, http://www.acastat.com/statbook/chisqassoc.htm (16-Aug-15), 16-Aug-15 (Accessed: 14-Jul-20).

medium.com, https://medium.com/fiverr-engineering/feature-selection-beyond-feature-importance-9b97e5a842f (14-Jul-20), 14-Jul-20 (Accessed: 14-Jul-20).

R., P.P., M.L., V. and S., S. (2011) ‘GAIN RATIO BASED FEATURE SELECTION METHOD FOR PRIVACY PRESERVATION’, ICTACT Journal on Soft Computing, 01(04), pp. 201–205. doi: 10.21917/ijsc.2011.0031

Kursa, M.B. and Rudnicki, W.R. (2010) ‘Feature Selection with the Boruta Package’, Journal of Statistical Software, 36(11) (14pp). doi: 10.18637/jss.v036.i11

analyticsvidhya.com, https://www.analyticsvidhya.com/blog/2016/12/introduction-to-feature-selection-methods-with-an-example-or-how-to-select-the-right-variables/ (2016), 05-Jul-20 (Accessed: 14-Jul-20).

Lunardon, N. (2014) ‘ROSE: A Package for Binary Imbalanced Learning’. Available at: https://pdfs.semanticscholar.org/42b6/c8c45702a32b4e68395a068cf1be72899713.pdf?_ga=2.30398143.141682011.1594711343-1416264642.1594711343, pp. 79-89 (Accessed: 15 July 2020).

Chawla, N. (2002) ‘SMOTE: Synthetic Minority Over-sampling Technique’. Available at: https://arxiv.org/pdf/1106.1813.pdf, pp. 321- 255 (Accessed: 15 July 2020).

imbalanced-learn.readthedocs.io, https://imbalanced-learn.readthedocs.io/en/stable/over_sampling.html (2019), 11 October (Accessed: 15 July 2020).

Patro, S.G.K. and sahu, K.K. (2015) 'Normalization: A Preprocessing Stage', IARJSET, pp. 20–22. doi: 10.17148/IARJSET.2015.2305

Kotsiantis, S.B., Kanellopoulos, D. and Pintelas, P.E. (2007) 'Data Preprocessing For Supervised Leaning', International Journal of Computer and Information Engineering, 1(12), pp. 4104–4109. doi: 10.5281/ZENODO.1082415

kdnuggets.com (2020) The 10 Algorithms Machine Learning Engineers Need to Know - KDnuggets, 16 July. Available at: https://www.kdnuggets.com/2016/08/10-algorithms-machine-learning-engineers.html (Accessed: 16 July 2020).

Donges, N. (2020) *The Random Forest Algorithm: A Complete Guide | Built In*, 15 July. Available at: https://builtin.com/data-science/random-forest-algorithm (Accessed: 15 July 2020).

Breiman, L. (2001) 'RANDOM FORESTS'. Available at: https://www.stat.berkeley.edu/~breiman/randomforest2001.pdf, pp. 1-32. (Accessed: 15 July 2020).

Langley, P. (2013) Induction of Selective Bayesian Classifiers, 28 February. Available at: https://arxiv.org/ftp/arxiv/papers/1302/1302.6828.pdf, pp. 399-406. (Accessed: 15 July 2020).

analyticsvidhya.com ,https://www.analyticsvidhya.com/blog/2017/09/understaing-support-vector-machine-example-code/ (2017), 15 April (Accessed: 15 July 2020).

Bzdok, D. (2018) *Machine learning;Supervised methods, SVM and kNN*, 5 July. Available at: https://hal.archives-ouvertes.fr/hal-01657491/document, pp. 1-6. (Accessed: 15 July 2020).

Bergstra, J. and Bengio, Y., 2012. Random Search for Hyper-Parameter Optimization. Journal of Machine Learning Research, 13, pp.281-305.

Wang, L. (2005) 'Support Vector Machines: Theory and Applications' pp. 344-360 (Accessed: 15 July 2020)

analyticsvidhya.com, https://www.analyticsvidhya.com/blog/2020/06/auc-roc-curve-machine-learning/ (2020), 16-Jun-20 (Accessed: 14-Jul-20).