DEFECT CAUSE MODELING WITH DECISION TREE AND REGRESSION ANALYSIS: A CASE STUDY IN CASTING INDUSTRY

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF INFORMATICS
OF
MIDDLE EAST TECHNICAL UNIVERSITY

BY

BERNA BAKIR

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE
IN
INFORMATION SYSTEMS

MAY 2007

Approval of the Graduate School of Informatics

Assoc. Prof. Dr. Nazife Baykal
Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Science.

Assoc. Prof. Dr. Yasemin Yardımcı
Head of Department

This is to certify that we have read this thesis and that in our opinion it is fully adequate, in scope and quality, as a thesis for the degree of Master of Science.

Assoc. Prof. Dr. İnci Batmaz
Co-Supervisor

Assoc. Prof. Dr. Nazife Baykal
Supervisor

Examining Committee Members

Prof. Dr. Gülser Köksal (METU, IE)

Assoc. Prof. Dr. Nazife Baykal (METU, II)

Assoc. Prof. Dr. İnci Batmaz (METU, STAT)

Dr. Tuğba Taşkaya Temizel (METU, II)

Assoc. Prof. Dr. Yasemin Yardımcı (METU, II)

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Name, Last name: Berna Bakır

Signature :

ABSTRACT

DEFECT CAUSE MODELING WITH DECISION TREE AND REGRESSION ANALYSIS: A CASE STUDY IN CASTING INDUSTRY

Bakır, Berna
M.Sc., Department of Information Systems
Supervisor: Assoc. Prof. Dr. Nazife Baykal
Co-Supervisor: Assoc. Prof. Dr. İnci Batmaz

May 2007, 108 pages

In this thesis, we study improvement of product quality in manufacturing industry by identifying and optimizing influential process variables that cause defects on the items produced. Real data provided by a manufacturing company from the metal casting industry were studied. Two well-known approaches, logistic regression and decision trees, were used to model the relationship between process variables and defect types. The approaches used were compared.

Keywords: Decision trees, logistic regression, quality improvement, manufacturing, casting industry

ÖZ

KARAR AĞACI VE REGRESYON ANALİZİ İLE HATA NEDENİ MODELLEME: DÖKÜM ENDÜSTRİSİNDEN ÖRNEK BİR ÇALIŞMA

Bakır, Berna
Yüksek Lisans, Bilişim Sistemleri Bölümü
Tez Yöneticisi: Doç. Dr. Nazife Baykal
Ortak Tez Yöneticisi: Doç. Dr. İnci Batmaz

Mayıs 2007, 108 sayfa

Bu tezde, üretim endüstrisinde, üretilen ürünlerin kusurlu olmasında etkili süreç değişkenlerini ve bu değişkenlerin en iyi değerlerini saptayarak ürün kalitesini artırmayı amaçladık. Metal döküm endüstrisinden bir üretim firmasının sağladığı gerçek veri üzerinde çalışıldı. Süreç değişkenleri ve kusur türleri arasındaki ilişkileri modellemek amacı ile, yaygın olarak bilinen iki yaklaşım, lojistik regresyon ve karar ağaçları kullanıldı. Kullanılan iki yaklaşım ve sonuçları karşılaştırıldı.

Anahtar Kelimeler: Karar ağaçları, lojistik regresyon, kalite iyileştirme, üretim, döküm endüstrisi

To my family

ACKNOWLEDGMENTS

I wish to express my deepest gratitude to my supervisor Assoc. Prof. Dr. Nazife Baykal and co-supervisor Assoc. Prof. Dr. İnci Batmaz for their guidance, support, encouragements and insight throughout the research.

I would also like to thank Prof. Dr. Gülser Köksal for letting me work within an excellent research group and for her guidance, patience, advice and criticism.

I would like to present my thanks to my friends Fatma Güntürkün, İlker İpekçi and Başak Öztürk for their invaluable support and help throughout the study.

I would also like to thank Prof. Dr. Sinan Kayalıgil, Prof. Dr. Nur Evin Özdemirel, Assoc. Prof. Dr. Murat Caner Testik and Prof. Dr. Gerhard Wilhelm Weber for their suggestions and comments.

I would also like to present my thanks to Erhan İşkol from the company for his helpful attitude and support throughout the study.

I would like to thank all my friends in Informatics Institute for providing me with friendly and nice place to work.

Special thanks to Sibel Gülnar for her help and support whenever I need.

I would like to thank Hasan Sertkaya for his endless love.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

Knowledge is the most precious entity for enterprises. In the competitive environment of the business, well understood and efficiently managed systems enable the companies to put themselves one step further from the competitors. In order to understand and manage their business, companies have embedded some data analysis methodologies such as statistical quality control, total quality management, decision support systems, etc. into their business for years.

Manufacturing, the fundamental part of the economy, is one of the most complex industries. Manufacturing systems consist of several subparts processed parallel or sequential manner and they are influenced by many factors. Each of these parts produces data during the processing. Developments in database technology and computer science (faster computers, more memory, storage capability, automatic data collection tools) enable the companies to collect and store significant amount of data easily. Nevertheless, most of these data are not fully used. There are several reasons of this. Firstly, because of the complex structure of manufacturing systems, traditional data analysis techniques have some limitations. They have computational limitations in terms of number of dimensions, number of observations, etc. Interactions among the process variables are not easily modeled by the traditional techniques. Generally, these techniques focus on specific and limited part of the system. However, usually complex relationships exist between subparts and these interactions cannot be extracted using traditional techniques since they fail to model the process as a whole. Another reason is that the companies may not have technical staff expert on advanced data analysis techniques.

Data mining is a current technology found to be useful in many complicated fields such as CRM (Customer Relationship Management), finance and retail. It can be defined as the process of extracting interesting patterns from databases. For this purpose, many fields have extensively used data mining techniques for decades. In manufacturing, it recently became very popular. These techniques can provide considerable competitive advantages to the manufacturing field.

Data mining studies in the field of manufacturing generally focus on process control, quality improvement, fault detection, process design, and maintenance. In Harding, Shahbaz, Srinivas & Kusiak (2006), a comprehensive review of data mining applications in manufacturing can be found.

1.1 Scope of the Study

This study deals with quality improvement in manufacturing by identifying the most influential process variables that cause defects on the items produced and suggesting optimum regions for those variables using data mining approach. A casting company located in Turkey contributed to the study. To this end, one of the items produced by the company with high defective rates is selected. The name of the process variables and the name of the defect types related to the product were not declared throughout the text due to protect confidential commercial information. Two approaches, the logistic regression analysis and decision trees, are used to model the relationship between process parameters and defect types.

Logistic regression is a traditional technique widely used in the manufacturing field to identify the best production regions in the factor space. This technique is usually applied to data collected by design experiments and the data includes a few numbers of factors selected by the domain experts. Therefore, possible important factors may not be included in the models developed. In this study, observational data were used to develop logistic regression models and all factors, data of which are available, were considered.

Decision tree is one of the most popular data mining techniques. It is extensively used for several reasons (Ye, 2003). First, the tree models are simple and can be easily converted to readable rules that are easy to interpret. The next reason, decision tree models are as successful as the other data mining techniques and usually faster than the other methods. Another reason is that tree models are nonparametric and suitable for exploratory data analysis. Because of these advantages, decision tree technique was preferred in this study among the other data mining techniques.

1.2 Organization of the Work

This study is presented in six chapters, which are organized as follows:

In Chapter II, background knowledge about data mining is given. Major data mining tasks and techniques that can be used to solve business problems are introduced.

Chapter III presents data mining studies in the literature performed to find answers to quality problems.

Material and methods used in this study are presented in Chapter IV. Logistic regression, two decision tree algorithms and the software used to implement the algorithms are explained briefly.

In Chapter V, casting industry is introduced, the data provided by the casting industry are described and results of the models developed are given.

Summary and conclusion of the study, and possible future work are presented in Chapter VI.

CHAPTER 2

BACKGROUND ON DATA MINING

2.1 What is Data Mining?

Data mining, one of the steps of Knowledge Discovery in Databases (KDD) process, is an exploratory data analysis that discovers interesting knowledge, such as associations, patterns, structures and anomalies from large amount of data stored in databases or other information repositories. Data mining is often used interchangeably with the term KDD since it is the core of the whole process. It is an interdisciplinary field being confluence of several sciences such as statistics, computer science, machine learning, artificial intelligence, database technology, and fuzzy systems. According to the perspective of researchers many definitions exist. The most well known definition in words of Fayyad is "Data mining is the nontrivial process of identifying valid, novel, potentially useful, and ultimately understandable patterns in data" (Fayyad, Piatetsky-Shapiro, Smyth & Uthurusamy, 1996).

2.2 Steps in KDD Process

Data mining process consists of several steps. Before carrying out these steps, first, the problem and business goal should be clearly defined. Once the problem is understood and target data is selected, the longest and challenging step of the process, preprocessing, is performed. It accounts for about 60% of the efforts in the whole process. Preprocessing is very important because results strongly depend on this step. Real-world data are usually noisy, incomplete and inconsistent. This step includes data cleaning, data integration, if data are obtained from different resources, data transformation, and data reduction. Next, appropriate data mining methods and techniques are applied to extract information from the data. The last step is evaluation of the mined patterns. The knowledge obtained is used for decision-making. Figure 1 shows the steps of knowledge discovery process.

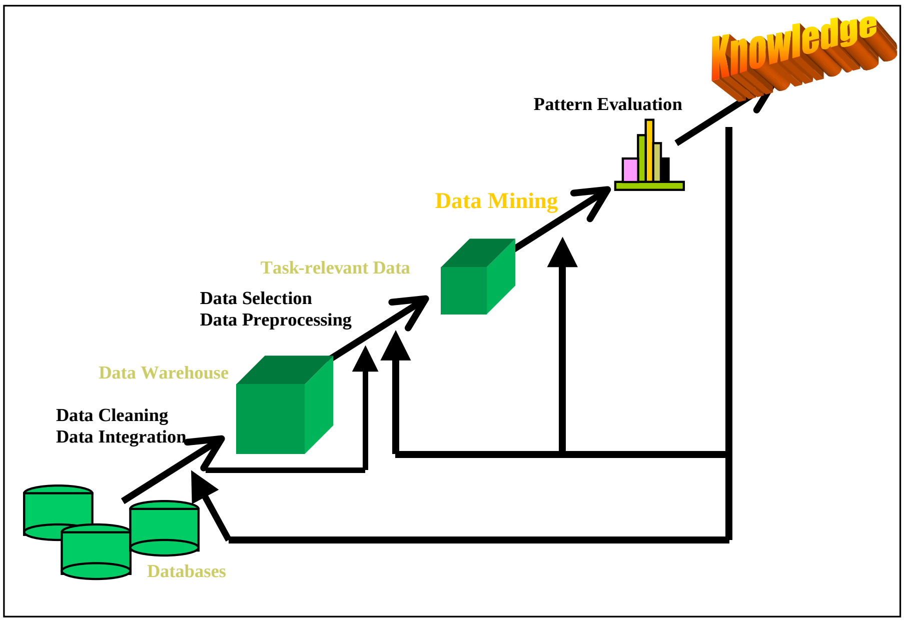


Figure 1: Steps of knowledge discovery process

2.3 Application Areas

There is no theoretical restriction about the data type. Relational databases, data warehouses, transactional databases, object-oriented databases, spatial databases, text databases, multimedia databases, and internet are examples of data type that data mining can be applied (Han, & Kanber, 2001). That is why data mining attracts many application areas. It is being used extensively in direct marketing, retail, finance, banking, communication, and insurance. These application domains frequently focus on customers. The following are some questions from different domains that can be answered by data mining techniques[1]:

1. Retail/Marketing:
   - What are the buying patterns of our customers?
   - Is there an association among customer demographic characteristics?
   - Which customers do respond to mailing campaigns?
   - Which product groups are frequently bought together?
2. Banking:
   - What are the characteristics of fraudulent credit card use?
   - What are the characteristics of `loyal' customers?
   - Which customers are likely to change their credit card affiliation?
   - What are the differences of credit card spending according to customer groups?
   - What are the correlations among different financial indicators?
3. Insurance and Health Care:
   - Which medical procedures are claimed together?

[1] Retrieved September 29, 2006, from http://www.estard.com/data_mining/using_data_mining.asp

- Which customers will buy new policies?
- What is the behavior of risky customers?
- What are the characteristics of fraudulent behavior?

4. Transportation:
   - What are distribution schedules among outlets?
5. Medicine:
   - What is the characteristic of patients that visit office frequently?
   - Which medical therapies are successful for different illnesses?

2.4 Data Mining Tasks

According to their final goal, data mining tasks can be classified as descriptive or predictive. Descriptive data mining aims to summarize data and extract their characteristics. Predictive data mining, on the other hand, try to find models to forecast future behaviors. The most common tasks can be performed by data mining are classification, prediction, clustering, dependency analysis, and outlier analysis. (Han, & Kanber, 2001)

1) Classification

Classification, an example of supervised learning, maps data items to one of the several predefined groups. It is the most popular data mining method and suitable for many applications such as credit scoring, image and pattern recognition and medical diagnosis (Fayyad et al., 1996). This approach uses a training set to learn where all objects are already associated with known class labels.

The model built using training set is used to classify new objects. There are several algorithms to perform classification task. They can be classified as statistical-based algorithms, distance-based algorithms, decision tree-based algorithms, neural network-based algorithms and rule-based algorithms.

2) Prediction

Prediction can be used to forecast some unavailable data values or pending trends. The major idea is to use a large number of past values to consider probable future values. Prediction may be considered as a type of classification, however, the difference between prediction and classification is that the prediction aims to forecast a continuous value whereas classification forecast a discrete value, such as class labels.

3) Clustering

Clustering, also known as unsupervised classification, aims to segment data into meaningful subparts. Unlike classification, class labels are unknown and algorithms discover acceptable classes. There are many clustering approaches, all of which are based on the maximization of the similarity between cases in a same class (intra-class similarity) and minimization of the similarity between cases of different classes (inter-class similarity). Clustering algorithms can be categorized as hierarchical, partitional, density-based, grid-based and Self Organizing Maps (SOM).

4) Dependency Analysis

Dependency analysis determines significant relationships (trends and patterns) between items appearing in transactions of such items. Two types of dependency analysis are considered; association rule discovery and sequential rule discovery. Association rule discovery studies the frequency of items occurring together in transactional databases, and based on a threshold called support, identifies the frequent item sets. Another threshold, confidence, which is the conditional probability of an item appears in a transaction when another item appears, is used to pinpoint association rules. As opposed to association rules, sequential rules take the order into account. That is, these patterns are based on a time sequence of actions.

Retail stores to assist in marketing, inventory control and advertising use association analysis commonly. Some other examples that used association analysis are detecting the mutual interactions of two or more different drugs, detecting the correlations between patients' responses to a drug.

5) Outlier Analysis

Outliers are data elements that cannot be grouped in a given class or cluster. They are often very important to identify. While outliers can be considered as noise and discarded in some applications, they can reveal important knowledge in other domains, and thus can be very significant and valuable. For instance, this type of analysis is suitable for fraud detection. Some other examples are finding unusual responses to a medical treatment, identifying spending behavior of customers with extremely low or high income. Methods for outlier analysis can be classified as statistical-based, distance-based, and deviation-based algorithms.

2.5 Data Mining Techniques

In this section, well-known data mining algorithms for three major tasks, classification, clustering, and association, are introduced.

2.5.1 Classification Techniques

Decision Trees

Decision trees are simple and successful predictive learning algorithms. They represent all possible paths associated with every outcome by the flow-diagram-like tree structure (Russell & Norvig, 2003). Decision trees can be used in both classification and prediction tasks. Learning using decision trees consist of two steps. In the first step, a tree is constructed using the training data. Then, for each record, the tree is traversed to determine the class to which the record belongs. Figure 2 illustrates a simple decision tree. Each internal node in the tree indicates a test on an attribute, each branch represents an outcome of the test, and each leaf node points a class.

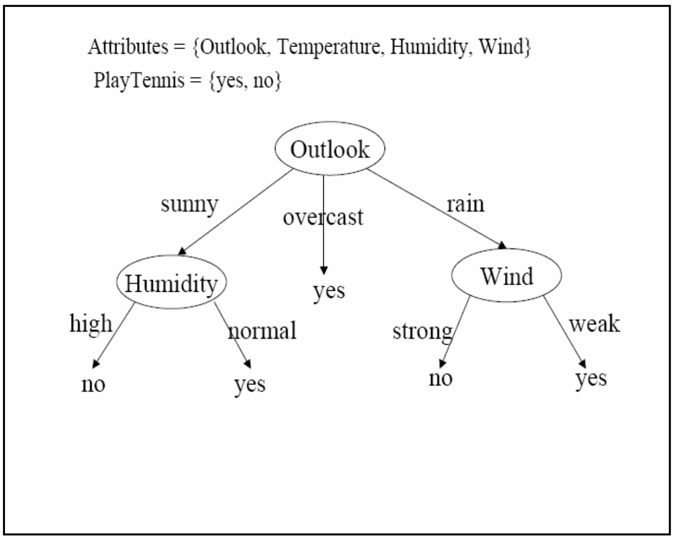


Figure 2: A simple decision tree

Several algorithms exist that can be work with either continuous and categorical dependent variable or one of the type. Predictor variables on the other hand, can be of type both continuous and categorical.

Because the technique has several advantages, it is very popular among the data mining techniques (Ye, 2003; Weis & Indurkhya, 1998; Breiman, Friedman, Olshen, & Stone, 1998). One of the advantages of decision trees is their simplicity and efficiency. Tree algorithms are usually faster than the other methods. Size of the tree is independent of the database size. In other words, decision trees are also efficient for large database and high dimensional data. Complexity of a tree model can be expressed by the number of leaf nodes and dept of the tree. Pruning can be used to obtain simpler models. Decision tree results are easy to understand and interpret since readable rules can be easily derived from the models (IF…THEN…). In addition, classification or prediction accuracy of decision tree models are as successful as the other data mining techniques. This technique is suitable for exploratory data analysis since the tree models are nonparametric.

There are some issues faced by most of the decision tree algorithms. Tree is affected by anomalies such as, noise, missing data and outliers and these anomalies may result overfitting. Tree pruning, the process of removing the least reliable branches from the tree using some statistical measures, may solve this problem. In addition, correlations among the attributes are not taken into consideration; they are ignored. Choosing the splitting attributes is another important issue. Some attributes may not have valuable information or ability to split input space efficiently. The order of the splitting attributes is also important. In order to select informative attributes correctly, impurity-based measures can be used. For instance, entropy (Quinlan, 1986) and gini-index (Breiman et al., 1998) are this type of split selection measures. By this way, the selected attributes minimize the impurity function and the number of tests needed to classify the data. Number of splits is another difficult issue to be determined. The number of splits is obvious when the attribute is categorical with small domain or if the algorithm produces a binary tree. However, if the attribute is continuous, it is not an easy task to determine this number. Stopping criteria is also important issue to be considered due to the accuracy of the classification and performance of the tree. Tree construction depends on the training dataset. Different datasets may result different classifications. A too small datasets may not represent classification correctly. Conversely, if the dataset is too large, then overfitting may occur. Cross-validation technique can be used to reduce overfitting (Russell et al., 2003).

Several decision tree algorithms were proposed. ID3 is one of the most well-known algorithms. The basic idea of this algorithm is to ask questions whose answers give the most information. The idea is based on the information theory and aims to minimize the expected number of comparisons. Another algorithm is C4.5, which improves ID3 in terms of missing values, continuous data, pruning, and the number of splits (Quinlan, 1986). CART (Classification and Regression Trees) is another decision tree algorithm proposed by Breiman et al., (1984) that generates a binary decision tree. It uses entropy as a measure to select splitting attribute like ID3 does.

K Nearest Neighbors

K Nearest Neighbors is a distance-based algorithm for non-parametric pattern classification. Given the knowledge of N prototype objects and their correct classification into several classes, the training examples are mapped into multidimensional feature space, and the space is partitioned into regions by class labels of the training samples. A point in the space is assigned to the class *c* if it is the most frequent class label among the *k* nearest training samples. Usually, Euclidian distance is used. Algorithm makes no assumption on the probabilistic distribution of the sample points and of their classification. The algorithm is easy to implement, but it is computationally intensive, especially when the size of the training set grows.

Neural Networks

Artificial neural networks (ANN) are modeling techniques derived from human brain (Haykin, 1994). They can be used to model complex non-linear relationships between inputs and outputs or to find patterns in data. An ANN consists of interconnected processing elements, nodes, where every connection has a weight. As decision trees, ANN approach requires a graphical structure to be built before applying it to the data. General topology of an ANN is shown in Figure 3:

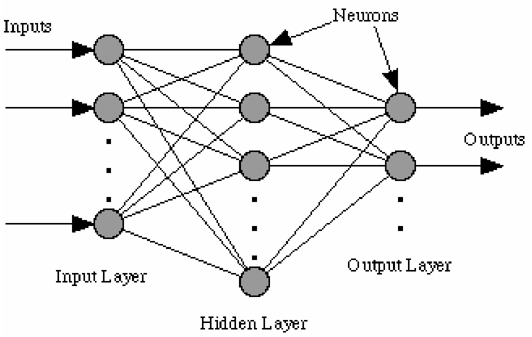


Figure 3: Topology of a multilayer ANN

Nodes in a network have three common elements: a set of connection links characterized by weights, base function, and activation function.

During the learning phase, the correct class for each record is known (supervised learning). Records are presented to the network one at a time, and the weights are adjusted each time by a learning algorithm according to errors of the previous feeding to improve accuracy of the prediction. After all cases are presented, the process often starts over again. The most popular neural network algorithm, proposed by Werbos (1974), is back-propagation algorithm.

Advantages of neural networks include their high tolerance to noisy data and robustness to tolerate error or failure. However, there are some issues to be considered (Dunham, 2003). One of them is the number of hidden layers. Another difficulty is the determination of the number of hidden nodes per hidden layer. It depends on problem, structure of the network, learning algorithm and types of activation function. One other question is interconnections. There are many activation functions that can be used. The proper function should be selected. Learning algorithm is another issue to be examined. Although the most common learning algorithms are some form of backpropagation, there are many others that can be used such as Boltzmann learning, ARTMap, etc. Number of output nodes and stopping criteria should also be determined.

2.5.2 Clustering Techniques

Hierarchical Algorithms

Hierarchical clustering algorithms work via grouping or splitting observations into tree of clusters. Two approaches, which are top-down and bottom-up, exist (Han, & Kanber, 2001). Followings are examples for these approaches

Agglomerative Clustering: In agglomerative clustering, each object is initially placed into its own group. That is, if we have N objects to cluster, we start with N groups and each of these groups contains only one object known as a singleton. According to a distance measure, all clusters are compared and the pair closest to each other is merged. This step continues iteratively until all items belong to one cluster or stopped when a certain number of clusters are reached. Distance between the clusters can be determined via single, complete and average link techniques. These techniques are based on graph theory concepts.

Divisive Clustering: This algorithm is opposite of agglomerative clustering. Here, all items are initially placed into one cluster, and at each step, clusters are spitted into two parts until all clusters have only one item. To split a cluster, first, farest object pairs in the cluster are determined. These objects are treated as seed for forming two clusters. Then, remaining objects are assigned to the group having the closest seed point.

Partitional Algorithms

K-Means Clustering: K-means is one of the simplest unsupervised learning algorithms that solve clustering problem. The main idea is to define k centroids, one for each cluster. The next step is to take each point belonging to a given data set and associate it to the nearest centroid. Then, centroid of each cluster is re-calculated. After we have these k new centroids, all data are reassigned to the clusters. This loop continues until no more changes are done. This algorithm aims at minimizing an objective function, squared error function.

The k-means algorithm has some weaknesses (Han, & Kanber, 2001). It is significantly sensitive to the initial randomly selected cluster centers. The algorithm can be run multiple times to reduce this effect. Another problem is the fact that the results depend on the metric used to measure the distance. Besides, the results depend on the value of k and we often do not know how many clusters exist.

PAM (Partitioning Around Medoids): PAM algorithm is also called as K-medoids algorithm. It represents a cluster by a medoid. PAM first computes k representative objects, called medoids. A medoid can be defined as that object of a cluster whose average dissimilarity to all the objects in the cluster is minimal. After finding the set of medoids, each object of the data set is assigned to the nearest medoid. The k representative objects should minimize the objective function, which is the sum of the dissimilarities of all objects to their nearest medoid. At each step, the algorithm looks at all pairs of medoid, non-medoid objects if there is an item that should be replaced one of the existing medoids that improves overall quality of clustering (Dunham, 2003).

Using medoids handles outliers well. It is more robust, because it minimizes a sum of dissimilarities instead of a sum of squared Euclidean distances.

Self Organizing Feature Maps (SOFM)

The self-organizing map (SOM), which is a subtype of ANN, was proposed by Kohonen (1982). A SOM networks is trained using competitive unsupervised learning to produce low-dimensional representation of the training samples while preserving the properties of the input space. Mathematical foundation of the SOM algorithm can be found in Kohonen (2001). Learning is based on the concept that behavior of a node should impact only itself and its neighbors. Network is organized into clusters based on the similarity between them. Basic elements of a self-organized network are as described in Haykin (1994):

1. One or two-dimensional lattice of neurons that calculates discriminant functions of inputs.

2. A mechanism to select the winner node having the largest discriminant function value.

3. An interactive network that activates winner node and its neighbors.

4. A process that increases discriminant function values of activated neurons in relation to the input.

2.5.3 Association Techniques

Association is a technique that is used to find relationships between item sets. These relationships are represented by association rules consisting of two parts, which are an antecedent and a consequent. The simplest type of association rules is the Boolean, single-dimensional, and single-level association rule. Technique searches for all the rules that satisfy a certain level of support and confidence. Most of the association rule algorithms use frequent item set strategy and the most well known is the Apriori algorithm proposed by Agrawal & Srikant (1994).

APRIORI Algorithm: Apriori is the basic association rule algorithm used to find frequent itemsets in databases. Several variations of Apriori algorithm exist. These variations aim improving efficiency and scalability (Ye, 2003). The Apriori algorithm is a breadth-first type of search algorithm. The basic property of the algorithm is the fact that if an itemset is frequent, all non-empty subset of it must also be frequent. The algorithm first generates frequent itemsets of size *k* then combines them to generate candidate frequent itemsets of size *k+1*. After finding frequent itemsets, strong association rules are extracted from these itemsets. Strong rules are described by the support and confidence statistics (Agrawal, Imielinski & Swami, 1993). Confidence value is the conditional probability of consequent given conditions. It can be calculated with the following equation.

$$confidence\ (A \Rightarrow B) = P(B \setminus \mathrm{A}) = \mathrm{support(A} \cup \mathrm{B)\ /\ support(A)}$$

where support (A) is the number of transactions containing itemset A and support (AUB) is the number of transactions containing itemsets (AUB).

CHAPTER 3

DATA MINING APPLICATIONS IN MANUFACTURING

Data mining is recently being used in manufacturing. General focuses in this field are process control, quality improvement, fault detection, process design, and maintenance. From the quality improvement perspective, problems can be classified as explaining poor-high quality, predicting quality, classification of quality, identifying factors that are critical for quality and optimization of the critical factors.

3.1 Literature Review

Semiconductor industry is the area where lots of data mining studies in literature were applied. This is because large amounts of data are available in this industry. In addition, chemical industry, iron and steel industry, metal processing industry, PCB (Printed Circuit Board) manufacturing, integrated circuit manufacturing are the application areas where data mining techniques have been used.

In order to find answers to the quality problems, mostly classification (see, for example, Abajo & Diez, 2004; Braha & Shmilovici, 2002; Hou, Liu & Lin, 2003; Huang & Wu, 2005; Jemwa & Aldrich, 2005; Krimpenis, Benardos, Vosniakos & Koukouvitaki, 2006 and Mieno, Sato, Shibuya, Odagiri, Tsuda & Take, 1999), prediction (see, for example, Wang, Wang, L., Zhao & Liu, 2006; Zhou, Xiong, Zhang & Xu, 2006; Skinner, et al., 2002; Shi & Tsung, 2003; Feng & Wang, 2003; Fan, Guo, Chen, Hsu & Wei, 2001; Deng & Liu, 2002; Li, Feng, Sethi, Luciow & Wagner, 2003 and Shi, Schillings & Boyd, 2004), rule induction (see, for example, Ho, Lau, Lee, Ip & Pun, 2006; Hou & Huang, 2004 and Kusiak & Kurasek, 2001) and clustering (see, for example, Lian, Lai, Lin & Yao, 2002; Skinner, et al., 2002; Chien, Wang & Cheng, 2007 and Cser, Gulyas, Szücs, Horvath, Arvai & Baross, 2001) methods were used in the literature. For instance, Brinksmeier, Tönshoff, Czenkusch & Heinzel (1998) used genetic algorithm for modeling and optimizing grinding processes. Gardner & Bieker (2000) applied clustering and rule induction techniques to explain poor-high quality in semiconductor wafer manufacturing. Here, the critical process variables that cause the poor yield were identified from observational wafer manufacturing data.

Classification and prediction studies in the literature usually were done by using regression, decision tree and neural network techniques. Some studies, which use classification and prediction tools, are presented briefly in Table 1 and Table 2, respectively. Except for the ordinary least square regression and generalized linear models used in Skinner, et al. (2002), data mining tools were reported to be successful to extract desired knowledge from the manufacturing data. For example, Braha & Shmilovici (2002) identified the factors that are significant in the cleaning process in semiconductor industry by using decision tree and neural network. With the given conditions, how much a new item cleaned was determined by the study. Feng & Wang (2003) developed an empirical model for surface roughness by using non-linear regression and neural network techniques to predict quality in metal casting industry. Another successful study performed in semiconductor industry by Fan, et al. (2001) is the identification of individual and combinational effects of equipments on the quality of products produced. Both regression and decision tree approaches used in the study produced successful results to find out critical factors for quality.

Table 1: Selected classification studies in the literature

Aim of the Research	Quality Task	Tools Used	Motivation of the researchers for the selected tools
A data mining approach was used by Mieno, et al. (1999) to specify the failure cause and improve yield in semiconductor manufacturing.	Explaining poor/high quality	Decision tree	Decision trees are fast and easy to interpret without depending on experience or skill.
Lian, et al. (2002) used data mining for improving the quality of vehicles by controlling the dimensional deviation of body-in-white in assembly of sheet metal products by identifying variation causes.	Finding critical factors of quality	Decision tree	Decision trees are fast.
Braha & Shmilovici (2002) aimed to improve the wafer cleaning processes by identifying the significant factors in the cleaning process. With the given conditions, how much a new item cleaned is determined. The application area is semiconductor manufacturing.	Classification of quality	Decision tree ANN	These techniques have ability to handle complex interactions and non-linearities.
Data mining was used in conveyor belts manufacturing by Hou, et al. (2003) to monitor and diagnose processes.	Explaining poor/high quality	ANN Rough Set	NNs have classification capability with high accuracy. Rough set has capability to extract practical rules.
In packaging manufacturing, a data mining study was performed by Abajo & Diez (2004) to develop new tinplate quality diagnostic models that meets ISO 9000 standards	Classification of quality	Multilayer-layer perceptron	Success of the NN models to classify output with high accuracy is the motivation.
In order to develop quality improvement strategies in ultra-precision manufacturing industry, product quality was analyzed using data mining by Huang & Wu (2005). In the paper, important factors influencing the percentage of defective products were identified.	Explaining poor/high quality	Decision tree (CHAID)	Decision trees are fast enough to provide feedback effectively

Table 2: Selected prediction studies in the literature

Aim of the Research	Quality Task	Tools Used	Motivation of the researchers for the selected tools
In semiconductor manufacturing individual and combinational equipment effects was identified by Fan, et al. (2001).	Finding critical factors of quality	Regression/ANOVA Decision tree	Decision trees do not need a pre-defined model
Skinner, et al. (2002) determined the yield of the wafers and the cause of low yield wafers in semiconductor manufacturing	Explaining poor/high quality	Ordinary Least Square Regression both on the original variables and principle components Generalized linear models on principle components CART	CART is distribution-free. There are no assumptions about the distributional properties of the data.
In metal casting an empirical model for surface roughness prediction was developed by Feng & Wang (2003)	Predicting quality	Nonlinear regression ANN	NNs has universal function approximation capability, resistance to noisy or missing data, good generalization capability.
In glass manufacturing, improving the glass coating process by adjusting machine settings that resulting in shorter setup time and better glass coating quality was studied by Li, et al. (2003).	Explaining poor/high quality Finding CTQ	CART ANN	Both approaches are powerful and nonparametric.
Shi, et al. (2004) studied in both chemical manufacturing and PCB manufacturing to understand process behavior and improve the process quality.	Predicting quality	ANN	NNs are successful to model, predict, control and optimize non-linear systems.

3.2 Major Issues in Manufacturing Data

Several issues exist in batch manufacturing data (Forrest, 2003). The most important ones include processing several products at once with the same process settings, missing data due to sampling and integration of data, high dimensionality relative to cases, aggregation and duplication.

In batch manufacturing environment, several lots are produced under the same process settings. Because of the natural variation of the process parameters, the system can produce products with good and bad quality. To understand the exact causes lead to poor product quality, values of factors during the manufacturing of those products are needed to be known. However, in batch manufacturing, items are considered as a group and data of the system are collected by sampling to monitor batches as a whole instead of concentrating on individual items. In addition, it is impossible to know which product is produced at a certain time of the manufacturing process since no labeling is not made. Consequently, a matching between a product and values of parameters measured by sampling at a certain time can not be done. This fact forces us to consider process values measured during the production of a batch as representative for all of the items produced in the associated batch.

As stated above, in batch manufacturing, measurements of the process variables are made by sampling. For different variables, samplings are performed with different frequencies and some of them are measured with very low frequency, e.g., once a week. This is because measurements of values for those variables are expensive or difficult. Differences between sampling rates of the parameters result in missing values in the datasets. If these differences increase, number of tuples having missing values increases too. Furthermore, the amount of missing data may be increased by integration of the datasets from different processes of the entire system. That is, if two data sets have some tuples with missing values, most probably, integrated set will have more tuples with missing values.

Traditionally, parts of the manufacturing systems are examined separately because of the limitation of techniques used to analyze data. One of the reasons is high dimensionality of entire process. This is one of the main characteristics of manufacturing systems. Studying sub-processes individually results in missing the interactions between sub-processes. However, when the process is considered as a whole by including all parameters from different sub-processes, number of cases will be small relative to process parameters due to the batch-by-batch representation.

Finally, granularity of the data collected from different subsystems may be different. During the integration of the data, aggregation or duplication may be needed to have the same granularity.

CHAPTER 4

MATERIAL AND METHODS USED IN THIS STUDY

In this study, two popular techniques were used to develop models that relate input factors to defect types. Firstly, logistic regression modeling, which is traditional statistical technique commonly used in manufacturing industry, was studied. Secondly, the popular data mining techniques, decision trees, was used to develop models. Decision tree approach was selected among the other data mining techniques because it is simple, fast, easy to interpret, and powerful. In addition, the technique is nonparametric and distribution-free. Algorithms were implemented by using SPSS Clementine 10.1.

4.1 SPSS Clementine

In this study, Clementine 10.1, (Clementine® 10.1 User's Guide, 2006) which is data mining component of the statistical software SPSS, was used to implement the algorithms. Clementine supports CRISP-DM methodology, hierarchical process model that enable users to guide data mining projects to achieve business objectives. Performing data mining via Clementine is the process of running data through a series of nodes, referred to as a stream. This series of nodes represents operations to be performed on the data, while links between the nodes indicate the direction of data flow. Typically, tasks of a stream of nodes are reading data into Clementine, running it through a series of manipulations and sending it to a destination to display results. Figure 4 shows an example of a stream created in Clementine. To illustrate, (1) export data from the excel worksheet named as “lots1.xls”. (2) is the node where the types of the variables are defined. In this node, input and output variables are also stated. (3) is the partition node which is used to divide dataset into training testing and validation subsets. (4) is the model node where model parameters and settings are defined. By processing stream from (1) to (4), the model, settings of which are defined in node (4), is generated. Node (5), which is generated model node, presents results of the model such as decision rules, important variables, and statistics. (6) is an output node called “analysis” which gives the performance statistics of the generated model such as classification accuracy, mean error, etc. Results of the output node can be viewed by processing the nodes (1)-(2)-(3)-(5)-(6) in sequence. You can look at Figure 5 for Clementine User Interface.

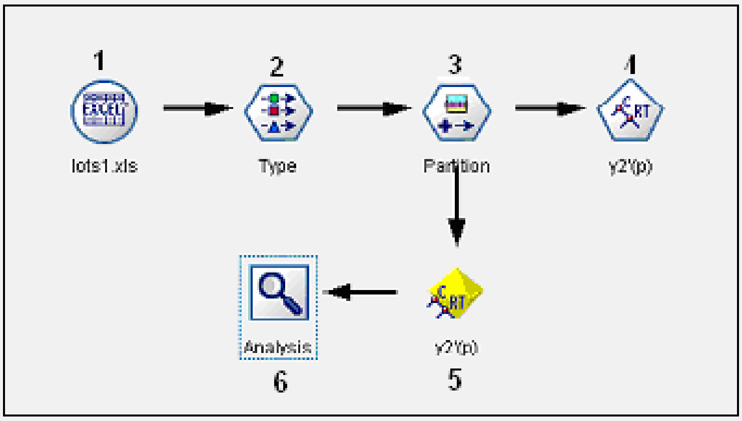


Figure 4: A stream in SPSS Clementine

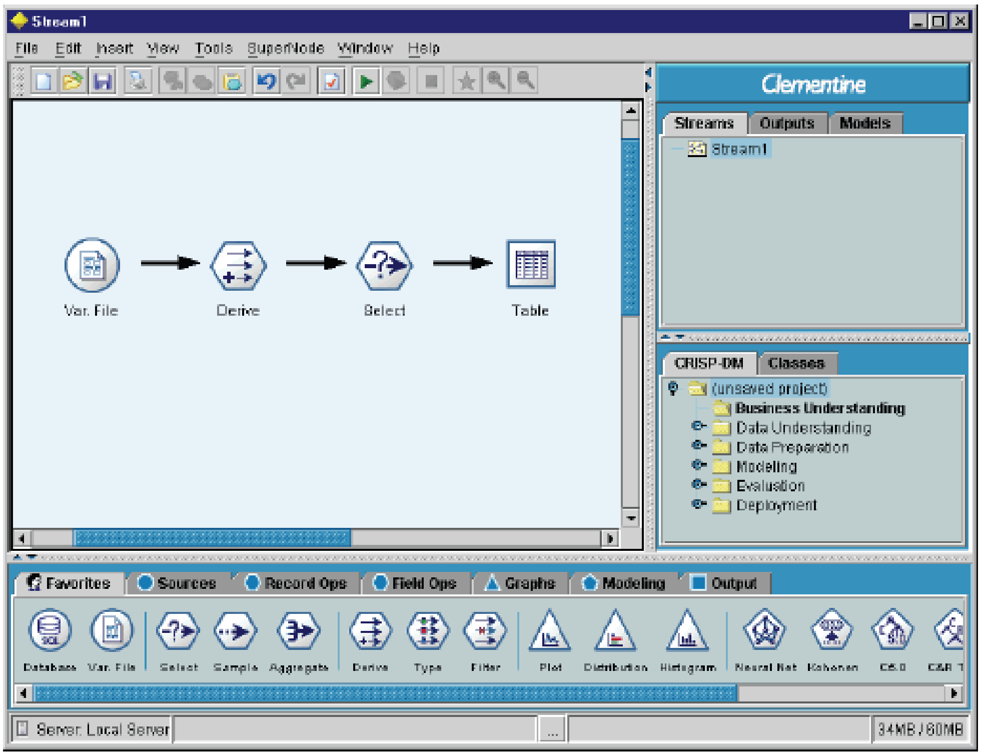

Stream1
File Edit Insert View Tools SuperNode Window Help
Var. File
Derive
Select
Table
Clementine
Streams Outputs Models
Stream1
CRISP-DM Classes
(unsaved project)
Business Understanding
Data Understanding
Data Preparation
Modeling
Evaluation
Deployment
Favorites Sources Record Ops Field Ops Graphs Modeling Output
Database Var. File Select Sample Aggregate Derive Type Filter Plot Distribution Histogram Neural Net Kohonen C5.0 C&R T
Server: Local Server
34MB / 60MB

Figure 5: SPSS Clementine 10.1 User Interface

4.2 Logistic Regression

Regression methods find relationships between a dependent response variable and one or more covariates. The most well known model of regression analysis is linear model where the response variable is of type continuous. However, in many field such as social science, the variable of interest is binary or nominal. For instance, the variable of interest may show presence or absence of some properties such as patient has disease or not, married or unmarried. Logistic regression is a statistical method which relates categorical dependent variables to input factors (Hosmer & Lemeshow, 2000). Dependent variables can be of type binary, ordinal or nominal. It is extensively used in medical science, social science and manufacturing industry.

There are some differences between linear and logistic models. A set of assumptions underlies the linear model (Montgomery & Peck, 1982). First, the response variable is linear function of covariates. The relationship between the predictor and response variables is not linear in logistic regression. The other assumptions of linear regression model are about the distribution of error term ε. These can be listed as follows:

- E (ε_i) = 0 (ε and covariates are uncorrelated)
- Var (ε_i) = σ^2 (variance of error term is constant, homoscedasticity)
- Cov ($\varepsilon_{i,}$ ε_j) = 0 (errors are uncorrelated)
- ε_i ~ N(0, σ^2) (errors are normally distributed)

When we consider binary logistic regression, distribution of errors is described by the binomial distribution instead of normal so that the error variance is equal to θ(1-θ).

Binary dependent variable in logistic regression takes the value 1 with a probability of success, θ, or the value 0 with probability of failure 1- θ. If the dependent variable is multinomial, it takes more than two values. Logistic regression equation can be written as:

$$\boxed{\theta = \frac{\exp(\beta_0 + \beta_1 x_1 + \ldots + \beta_k x_k)}{1 + \exp(\beta_0 + \beta_1 x_1 + \ldots + \beta_k x_k)}}$$

or

$$\boxed{\operatorname{logit}\left[\theta(x)\right] = \log\left[\frac{\theta(x)}{1 - \theta(x)}\right] = \beta_0 + \beta_1 x_1 + \ldots + \beta_k x_k}$$

where θ(x) is called a link function, k is the number of input factors and β_i is the parameter associated with the i^{th} input factor.

***Odds-Ratios*:**

Logistic regression calculates the probability of success over the probability of failure and the results are shown in the form of odds-ratio (Hosmer & Lemeshow, 2000). The odds of an event is defined to be the chance of success, (y = 1), relative to the chance of failure, (y = 0). It is defined as the ratio of the probability that an event occurs over the probability that it fails to occur. Thus,

Odds(y=1) = Pr(y=1) / [1 - Pr(y=1)]

or

Odds(y=1) = Pr(y=1) / Pr(y=0)

The odds-ratio is defined as the ratio of the odds for success to the odds for failure. It takes value between 0 and plus infinity. The relationship between odds-ratio and a regression coefficient can be expressed as follows:

Odds-ratio = $e^{\beta i}$

where β_i is the parameter associated with the i^{th} input factor. Odds-ratio, which is a measure of association, approximates how much more likely or unlikely that an event occurs than not to occur.

4.3 Classification and Regression Tree (CART)

The CART is a tree-based classification and prediction algorithm that is suggested by Breiman et al. (1984). It uses recursive partitioning to split the training records into segments with similar output field values. Construction of CART tree starts by examining the input fields to find the best split. Best split is measured by the reduction in an impurity index that results from the split. Each split defines two subgroups; these subgroups are subsequently split into two more subgroups. This procedure continues until a stopping criterion is achieved (Breiman et al., 1998). The CART algorithm constructs a binary tree. It works with both continuous and categorical responses and input values.

CART algorithm uses three different impurity measures to split data according to type of the response variable (Clementine® 10.1 Algorithms Guide, 2006). If the response value is continuous, least squared deviation (LSD) is used as an impurity measure. LSD is the weighted within-node variance for node *t*. It can be written as;

$$R(t) = \frac{1}{N_W(t)} \sum_{i \in t} w_i f_i (y_i - \overline{y}(t))^2$$

where $N_W(t)$ is the weighted number of records in node *t*, w_i is the value of the weighting field for record *i,* f_i is the value of the frequency field; y_i is the value of the target field, and $y(t)$ is the (weighted) mean for node *t*. The split is chosen to maximize the LSD criterion function below:

$$\Phi(s,t) = R(t) - p_L R(t_L) - p_R R(t_R)$$

where t_L and t_R are the nodes created by the split, *p*L is the proportion of records in *t* sent to the left child node, and *p*R is the proportion of records in *t* sent to the right child node.

For categorical response, three measures can be used. These are gini, twoing and ordered twoing measures. Gini measure shows inequality of the distribution and takes values between 0 and 1. It is based on probabilities of category membership for the branch. Gini measure at node *t* is defined as follows:

$$g(t) = 1 - \sum_{j} p^2(j|t)$$

where *j* shows the categories and

$$p(j|t) = \frac{p(j,t)}{p(t)}$$

Here,

$$p(j,t) = \frac{\pi(j)N_j(t)}{N_j}$$

$$p(t) = \sum_{j} p(j,t)$$

where π(j) is the prior probability value for category *j*, *N*j(*t*) is the number of records in category *j* of node *t*, and *N*j is the number of records of category *j* in the root node.

Twoing measure punctuates the binary split that results approximately equal-sized branches. The split point is decided to satisfy this criterion. Ordered twoing measure is used when the response variable is of type ordinal. It adds a constraint that only abutting categories can be grouped together.

CART models are quite robust in the presence of missing data and large numbers of data fields.

4.4 C5.0 Algorithm

C5.0 algorithm is the commercial version of C4.5 which is proposed by Quinlan (1986). C5.0 algorithm provides two kinds of model: decision tree and rule set. In the decision tree model, each case belongs to exactly one terminal node. That is, only one prediction is possible for each case. However, rule set model works quite different. It tries to make predictions for individual records. Rule sets are simplified versions of decision tree. The most important difference between decision tree and rule set is the fact that for rule set more than one or no rule may be applied to a particular record. In the study, the decision tree model was used to model.

A C5.0 tree algorithm splits data based on the field that provides the maximum information gain. The algorithm process until the sub-samples cannot be split any further. After construction, the lowest-level splits, which do not contribute significantly to the value of the model, are pruned. Pruning can be done by defining value of pruning severity parameter. Higher value results smaller tree. Fixing minimum number of records in leaf nodes also can control size of the tree.

Rules extracted via C5.0 models can be evaluated using two performance measures: support and confidence. These values are measured for each individual rule. Support and confidence value of a rule expressed as A→B can be defined as follows:

Support- Number of records for which the antecedent is true divided by the total number of records

Confidence- Ratio of number of records for which the entire rule is true over the number of records for which the antecedent is true

Confidence takes values between 0 and 1. We expect to obtain high confidence values for reliable rules. However confidence value is not sufficient alone. It should be evaluated with another measure, called support, which is important to generalize a rule. A rule can be generalized if number of records to which the rules applied is high enough. In the literature, minimum 5% of all records are acceptable.

C5.0 provides boosting method to increase accuracy of classification. The method builds multiple models in a sequence according to user-defined number of trials. The first model is built in the usual way. Each of next models is built by focusing on the records that were misclassified by the previous model. Final classification is made using a weighted voting procedure. Developing model using boosting method requires longer training time (Clementine® 10.1 Node Reference, 2006).

Similar with CART algorithm, in the presence of missing data and large numbers of input fields, C5.0 are quite robust.

CHAPTER 5

APPLICATION: A CASE STUDY IN CASTING INDUSTRY

5.1 Introducing the Casting Industry

Casting is the process of making product having complex shapes by pouring molten material into a mold and breaking out the solidified material from the mold. The main reason of using this procedure to make products is the difficulty of other methods such as cutting from the metal. Other methods can also be not economical. Metal casting has three main subsequent processes: core making, molding, and melting. Figure 6 shows all steps in sequence in metal casting process.

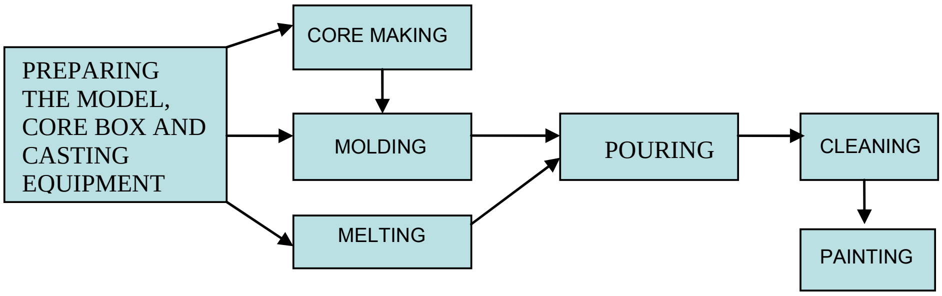


Figure 6: Casting process

Model Making: The model has the physical shape of the product to be manufactured. It is used to make molds. Material used to make mold is packed around the model so that the material takes its shape.

Molding: Molds are prepared using a mixture of sand. This process includes the following steps:

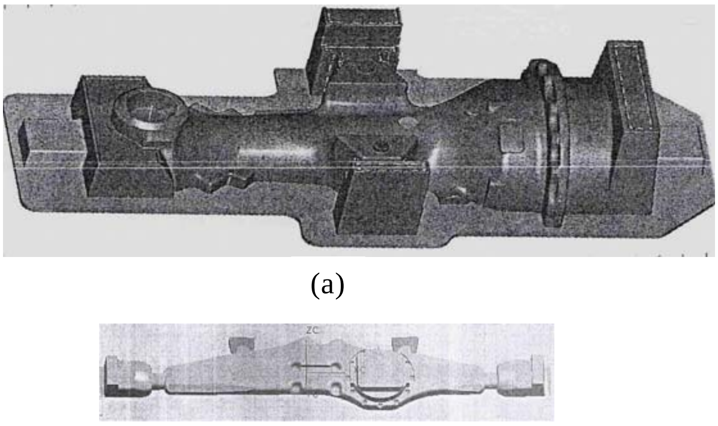
1) Packing the sand around the model with a frame and compressing it to form shape of the model.

2) Withdrawing the model from sand

3) Positioning core in the mold

4) Finally closing the mold.

Core Making: Cores are used to form interior surface of castings such as holes inside the product. Core figures are also made up of sand and placed in mold cavity. Figure 7 shows an example model for a mold and a core figure.



(a)

(b)

Figure 7: (a) A Model for a mold (b) Core figure

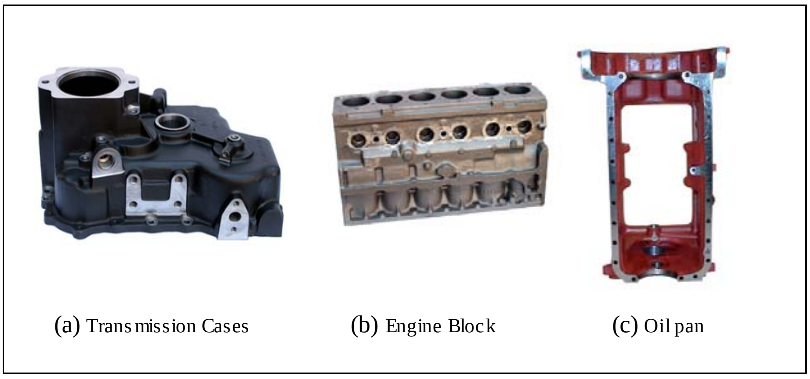
Melting and Pouring: In melting, mixture of metals is melt in an oven and molten material is transferred to the area where closed molds are placed. Here, pouring is performed and molten metals covered by molds are left to be solidified.

Cleaning: Solidified castings are removed from the sand and sent to the cleaning line where surface of the products are smoothed.

After operations in sequence are performed in all production lines, general quality control steps and inspection of products for defects are performed.

5.2 Introducing the Company

The company contributed to this study has two factories located in Ankara and produces intermediate products for agricultural tractors, automotive, and motor industry. It is an ISO/TS 16949 and ISO 9001 registered company and applies six sigma methodologies for improving its processes. Their commitment is to meet and surpass all requirements in manufacturing. They strive to improve efficiency and effectiveness of processes through the continual improvement of the quality. Figure 8 illustrates some products produced by the company.



(a) Transmission Cases

(b) Engine Block

(c) Oil pan

Figure 8: Some products of the company

One of the quality objectives of the company is to reduce the percentage of defective items by identifying and optimizing the most important process parameters. This is typically achieved by analyzing data collected by designed experiments. However, before such experimentation, it is necessary to determine the most significant factors involved in the process. To answer this question, quality team of the company provided us with the data for a particular product, a cylinder head, which is shown in Figure 9, collected during the first five-month production period of 2006.

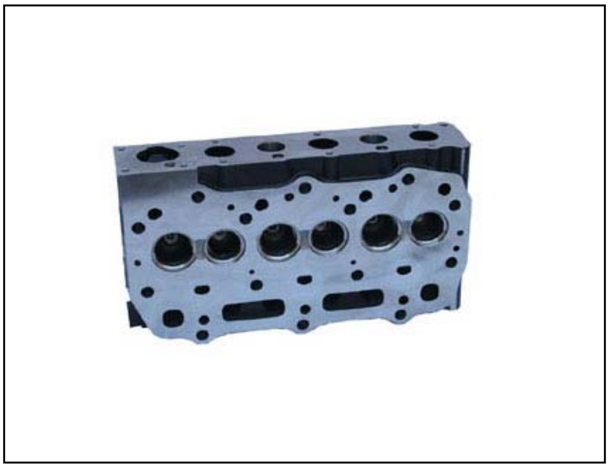


Figure 9: A cylinder head

The cylinder head sits top of the cylinders and limits size of the cylinder from the top in an internal combustion engine. It contains position of the spark plugs and valves. The cylinder head is an important part since it affects performance of the internal combustion engine[2].

5.3 Data Description and Preprocessing

The company performs batch production. A batch is a group of product produced in a certain day under the same process settings. Number of products produced in the batches is different. The product

[2] *Cylinder Head*. (n.d.). Retrieved February 15, 2007, from http://en.wikipedia.org/wiki/Cylinder_head

studied has high percentage of defectives. The dataset consists of observational values of 46 process variables particularly collected from three subsequent processes, which are core, molding and melting without conducting any specific data analysis. There are 11 quality variables representing the number of defective items in a batch for each of 10 defect types and the total number of defective items.

The company collects data to monitor batches and there is no way to know that under which exact process values the individual items were produced. For that reason, process values of a batch represents all items belong to that batch. All of the process values are measured by sampling from product produced. Frequency of sampling varies among the variables because of the economical reasons or difficulties. Most of the measurements are taken during the production so that every batch has its separate value. However, some of them are taken once a week and considered as the values of batches performed during the following week. At the end of each batch production, all of the products are inspected for certain defect types and number of defective items is recorded according to the types of defect. Another problem arising here is the fact that, if any defect type is observed on a product enough to reject it, no further analysis is performed to see the existence of other defect types. Only main cause to reject a product is recorded even more than one defect types are observed on the same product. Consequently, possible correlations between defect types are not provided by the data. Since defect type assigned to a product may not be the only defect that the product has, the quality variables will be considered as “quality reason to reject the product” instead of “a certain defect type exists on the product” or “a certain defect type does not exist on the product”.

Differences between frequencies of sampling resulting in lots of missing data and uncertain values of individual items forced us to aggregate data to batch level. For that reason, the dataset is arranged as described below:

1- An Excel worksheet was used to store dataset. 58 columns are labeled, 46 of them for process variables, one for total number of products, one for total defective products, and ten for defect types.

2- Each row is considered to represent batches. For each batch, the total number of production, the total number of defective products, the numbers of defective products for each defect type were entered into related columns.

3- Average values of the process variables measured more than once during a batch were calculated. For each batch calculated average values are also entered into the associated columns. If there is only one value measured, it is directly used to represent the batch to which it belongs.

4- Values measured once a week were repeated for the batches produced during the following week.

All matching were made using the date and time of the production and sampling made. Eleven of process variables having over than 50% of missing values were discarded from the dataset. Here, the reason for missing data is different from the missing data problem described above. These values do not exist because no samplings were done for that variable during the production of associated batches. Three of the batches were detected as outlier and these rows were also discarded. Outlier analysis was performed using anomaly detection algorithm of the SPSS Clementine. This algorithm extracts unusual observations based on the deviation from the norm of their cluster groups (You can find brief information about Anomaly Detecting Algorithm in Appendix C). Additionally, quality team of the company stated that occurrences of the four of the defect types are not related with process variables. Instead, the possible reasons of those defects are design fault, nonconformance of the components, etc. For that reason, defect categories were decreased to six and number of total defective items was reduced accordingly. This is the basic dataset having 36 process variables and 7 quality variables for 92 cases. For each batch proportion of total defective products and proportion of defective products for each of defect types were calculated using the total number of production. These values are the response values for the batch analysis. The basic dataset prepared can be considered as batch-based dataset. Table 3 gives basic statistics for the dataset.

Table 3: Descriptive statistics of the basic dataset

	Min	Max	Average	St. Dev.
xm	0.2150	0.5900	0.2604	0.0818
x2	20.0000	28.0000	23.5217	2.5530
x3	30.0000	40.0000	34.3382	2.8683
x4	12.1712	13.6781	12.6444	0.3124
x5	12.2700	13.6600	13.1115	0.3045
x6	7.5853	8.2500	7.9866	0.1280
x7	119.5000	145.0000	129.9310	5.0495
x8	35.0000	42.0000	39.7930	0.9618
x9	2.9800	3.3870	3.1932	0.0653
x10	470.0000	550.0000	510.5195	17.7223
x11	19.8000	22.9000	21.0655	0.5306
x12	290.0000	360.0000	322.3368	12.6114
x13	66.6000	76.6000	71.1210	1.8460
x14	4.7000	5.2000	4.9237	0.1192
x15	20.0000	49.3630	39.0428	4.9784
x16	13.2000	30.0000	20.6901	3.5998
x17	15.9000	31.5000	20.5120	3.5440
x18	16.3000	27.1000	20.1379	3.2409
x19	14.1000	24.9000	17.8428	2.7461
x20	38.9923	42.8500	41.2081	1.0072
x21	48.6800	52.7100	50.7796	0.8881
x22	10.8500	33.4281	17.2226	5.9205
x23	1410.2883	1480.0000	1464.1969	10.9125
x24	2.8406	3.9000	3.3736	0.1594
x25	4.3500	6.9000	5.8261	0.4439
x26	1380.8341	1428.2353	1421.9117	4.9088
x27	4.1018	4.9500	4.3135	0.1200

Table 3 (cont.): Descriptive statistics of the basic dataset

x28	11.7000	16.9000	14.3777	1.1453
x29	3.2080	3.4100	3.3159	0.0356
x30	1.8239	2.0000	1.9307	0.0366
x31	0.6884	0.8300	0.7379	0.0253
x32	0.1710	0.2830	0.2233	0.0214
x33	0.0767	0.5520	0.2187	0.1207
x34	0.0040	0.0570	0.0209	0.0146
x35	0.0762	0.1123	0.0902	0.0073
x36	0.0000	0.0270	0.0018	0.0062

5.3.1 Dataset I

Dataset I response variables of which are of type binary and nominal is another representation of the basic dataset. Preparation of dataset I is described below with an example:

Suppose that the dataset has two records and x1 is the number of total production; x2, x3, and x4 are the process variables; y1 is the total number of defective products and y2, y3, and y4 are the number of defective items having certain defect types. A few sample records for this dataset are shown in Table 4. To illustrate, in the first record, among 10 products produced under the process variables x2=0.25, x3=1.80 and x4=6.70, there are three defective items (y1) two of which have one type of defect (y2) whereas one of which has another type (y4).

Table 4: Sample records from batch-based dataset

x1	**x2**	**x3**	**x4**	**y1**	**y2**	**y3**	**y4**
10	0.25	1.80	6.0	3	2	0	1
15	0.34	1.74	5.93	6	1	2	3

While preparing the dataset, firstly, values of process variables of each batch were repeated the number of total production times so that each row represent one product. The number of rows is equal to the total number of production of two batches, which is 25. In this representation, quality variables represent the existence of defects instead of the frequencies. Because only one defect type, which is the main cause to reject, is recorded for each product by the company, only one defect type can take the value “1” in a single row. The others set to “0”. The dataset expanded as describe above is given in Table 5.

Table 5: Product-based data with binary quality variable

	x1	x2	x3	x4	y1	y2	y3	y4
1	10	0.25	1.80	6.70	1	1	0	0
2		0.25	1.80	6.70	1	1	0	0
3		0.25	1.80	6.70	1	0	0	1
4		0.25	1.80	6.70	0	0	0	0
5		0.25	1.80	6.70	0	0	0	0
6		0.25	1.80	6.70	0	0	0	0
7		0.25	1.80	6.70	0	0	0	0
8		0.25	1.80	6.70	0	0	0	0
9		0.25	1.80	6.70	0	0	0	0
10		0.25	1.80	6.70	0	0	0	0
11	15	0.34	1.74	5.93	1	1	0	0
12		0.34	1.74	5.93	1	0	1	0
13		0.34	1.74	5.93	1	0	1	0
14		0.34	1.74	5.93	1	0	0	1
15		0.34	1.74	5.93	1	0	0	1
16		0.34	1.74	5.93	1	0	0	1
17		0.34	1.74	5.93	0	0	0	0
18		0.34	1.74	5.93	0	0	0	0
19		0.34	1.74	5.93	0	0	0	0
20		0.34	1.74	5.93	0	0	0	0
21		0.34	1.74	5.93	0	0	0	0
22		0.34	1.74	5.93	0	0	0	0
23		0.34	1.74	5.93	0	0	0	0
24		0.34	1.74	5.93	0	0	0	0
25		0.34	1.74	5.93	0	0	0	0

To crate a nominal variable for representing all defect categories, the following coding schema was used.

Table 6: Coding schema for categorical output

y2	y3	y4	y
1	0	0	1
0	1	0	2
0	0	1	3
0	0	0	0

In Table 6, the values of the variable y can be expressed as follows:

0: product is non-defective

1: the main reason for rejecting the product is of defect type “y2”

2: the main reason for rejecting the product is of defect type “y3”

3: the main reason for rejecting the product is of defect type “y4”

Using the coding described above, the product-based process data with nominal response variable can be written as follows (Table 7):

Table 7: Product-based data with nominal quality variable

	x1	x2	x3	x4	y
1	10	0.25	1.80	6.70	1
2		0.25	1.80	6.70	1
3		0.25	1.80	6.70	3
4		0.25	1.80	6.70	0
5		0.25	1.80	6.70	0
6		0.25	1.80	6.70	0
7		0.25	1.80	6.70	0
8		0.25	1.80	6.70	0
9		0.25	1.80	6.70	0
10		0.25	1.80	6.70	0
11	15	0.34	1.74	5.93	1
12		0.34	1.74	5.93	2
13		0.34	1.74	5.93	2
14		0.34	1.74	5.93	3
15		0.34	1.74	5.93	3
16		0.34	1.74	5.93	3
17		0.34	1.74	5.93	0
18		0.34	1.74	5.93	0
19		0.34	1.74	5.93	0
20		0.34	1.74	5.93	0
21		0.34	1.74	5.93	0
22		0.34	1.74	5.93	0
23		0.34	1.74	5.93	0
24		0.34	1.74	5.93	0
25		0.34	1.74	5.93	0

The steps explained with an example above were applied to the basic dataset. As a result, number of rows was increased from 92 batches to 10997, which is the total number of production for all batches.

As a result of the data collection procedure of the company, in Dataset I, we had to represent both defective and non-defective items produced by the same values of process variables within a batch.

5.3.2 Dataset II

The organization of Dataset I is not useful for performing the classification task. For that reason, a sampling method was followed which allows us to classify products as non-defective or defective having one of the defect types which is the main cause to reject. Sampling was performed using Dataset I having binary type of defect categories (See Table 5 for an example view). The key point for the sampling is the fact that if a batch produces defects, it represents nothing but the defective products. Similarly, non-defective products were only represented by the batches having no defects at all. The sampling procedure used is described below with an example.

Suppose that a dataset has four records as the one given in Table 8. x1 is the total number of production; x2, x3, x4 are the process variables; y1 is the total number of defective products, and y2, y3 and y4 are the number of defectives of certain defect types.

Table 8: Sample records from batch-based dataset

x1	x2	x3	x4	y1	y2	y3	y4
10	0.3	2.3	0.7	8	3	2	3
15	0.2	2.6	0.6	5	0	3	2
20	0.4	2.1	0.5	8	6	0	2
13	0.7	2.9	0.8	0	0	0	0

As stated above, while the first three records describe the defective situations, the last one represents non-defective products. Each of first three batches was repeated y1 times and the reason to reject for those was shown as a binary variable representing the defect type. Illustration of defectives is shown in Table 9. Total number of rows sampled from first three records is equal to total defective items, which is 21.

Table 9: Product-based representation of defectives quality variables of which are binary

x1	x2	x3	x4	y1	y2	y3	y4
10	0.3	2.3	0.7	1	1	0	0
	0.3	2.3	0.7	1	1	0	0
	0.3	2.3	0.7	1	1	0	0
	0.3	2.3	0.7	1	0	1	0
	0.3	2.3	0.7	1	0	1	0
	0.3	2.3	0.7	1	0	0	1
	0.3	2.3	0.7	1	0	0	1
	0.3	2.3	0.7	1	0	0	1
15	0.2	2.6	0.6	1	0	1	0
	0.2	2.6	0.6	1	0	1	0
	0.2	2.6	0.6	1	0	1	0
	0.2	2.6	0.6	1	0	0	1
	0.2	2.6	0.6	1	0	0	1
20	0.4	2.1	0.5	1	1	0	0
	0.4	2.1	0.5	1	1	0	0
	0.4	2.1	0.5	1	1	0	0
	0.4	2.1	0.5	1	1	0	0
	0.4	2.1	0.5	1	1	0	0
	0.4	2.1	0.5	1	1	0	0
	0.4	2.1	0.5	1	0	0	1
	0.4	2.1	0.5	1	0	0	1

To illustrate, consider the first record in Table 9. It represents one defective item out of eight whose defect type is y2. However, if defect counts are directly used without considering the total number of production, a batch having high defective proportion may be treated the same as a batch having low defective proportion because of the equal number of defectives in each batch. For instance, for the first and third batches defect counts are the same, which is 8, and both are represented by equal number of rows. However, they are not equal. First batch has 80% defective items whereas the third one has 40%. To overcome this problem, the total number of productions was fixed to a constant value, which is 10, and defect counts were calculated using simple direct proportion. If the calculated defect count is a decimal number, it is rounded to the closest integer. After these calculations were done, defective items are totally represented by 15 rows (See Table 10).

Table 10: Product-based representation of defectives quality variables of which are binary (normalized)

x1	x2	x3	x4	y1	y2	y3	y4
10	0.3	2.3	0.7	1	1	0	0
	0.3	2.3	0.7	1	1	0	0
	0.3	2.3	0.7	1	1	0	0
	0.3	2.3	0.7	1	0	1	0
	0.3	2.3	0.7	1	0	1	0
	0.3	2.3	0.7	1	0	0	1
	0.3	2.3	0.7	1	0	0	1
	0.3	2.3	0.7	1	0	0	1
10	0.2	2.6	0.6	1	0	1	0
	0.2	2.6	0.6	1	0	1	0
	0.2	2.6	0.6	1	0	0	1
10	0.4	2.1	0.5	1	1	0	0
	0.4	2.1	0.5	1	1	0	0
	0.4	2.1	0.5	1	1	0	0
	0.4	2.1	0.5	1	0	0	1

The number of rows that will be included in the final dataset for the fourth record in Table 8, that is non-defective case, was calculated by equating the ratio of non-defective products over total number of production of the initial dataset to the corresponding ratio of the final dataset to be constructed. To illustrate, for the initial dataset (See Table 8) this ratio is $\frac{13}{58}$ and it is equal to $\frac{n}{15+n}$ for the final dataset where *n* is the total number of records for non-defective products in final dataset. When the equation $\frac{13}{58} = \frac{n}{15+n}$ is solved for *n*, *n* is found to be 4.33 (~ 4). After inserting non-defective cases into the dataset shown in Table 10, final dataset having binary quality variables representing defective items rejected for a certain defect type and non-defective items was constructed (See Table 11).

Table 11 : Final dataset with binary quality variables

x2	x3	x4	y1	y2	y3	y4
0.3	2.3	0.7	1	1	0	0
0.3	2.3	0.7	1	1	0	0
0.3	2.3	0.7	1	1	0	0
0.3	2.3	0.7	1	0	1	0
0.3	2.3	0.7	1	0	1	0
0.3	2.3	0.7	1	0	0	1
0.3	2.3	0.7	1	0	0	1
0.3	2.3	0.7	1	0	0	1
0.2	2.6	0.6	1	0	1	0
0.2	2.6	0.6	1	0	1	0
0.2	2.6	0.6	1	0	0	1
0.4	2.1	0.5	1	1	0	0
0.4	2.1	0.5	1	1	0	0
0.4	2.1	0.5	1	1	0	0
0.4	2.1	0.5	1	0	0	1
0.7	2.9	0.8	0	0	0	0
0.7	2.9	0.8	0	0	0	0
0.7	2.9	0.8	0	0	0	0
0.7	2.9	0.8	0	0	0	0

A nominal variable representing defect categories was created using the following coding schema in Table 12.

Table 12: Coding schema for categorical output

y2	y3	y4	y
1	0	0	**1**
0	1	0	**2**
0	0	1	**3**
0	0	0	**0**

In Table 12, the values of the variable y can be expressed as follows:

0: product is non-defective

1: the main reason for rejecting the product is of defect type “y2”

2: the main reason for rejecting the product is of defect type “y3”

3: the main reason for rejecting the product is of defect type “y4”

Final dataset having nominal quality variable coded as explained above is shown in Table 13. Quality variable of this dataset illustrates either defect categories that cause to reject the product or non-defective items.

Table 13: Final dataset with categorical quality variable

x2	x3	x4	y
0.3	2.3	0.7	1
0.3	2.3	0.7	1
0.3	2.3	0.7	1
0.3	2.3	0.7	2
0.3	2.3	0.7	2
0.3	2.3	0.7	3
0.3	2.3	0.7	3
0.3	2.3	0.7	3
0.2	2.6	0.6	2
0.2	2.6	0.6	2
0.2	2.6	0.6	3
0.4	2.1	0.5	1
0.4	2.1	0.5	1
0.4	2.1	0.5	1
0.4	2.1	0.5	3
0.7	2.9	0.8	0
0.7	2.9	0.8	0
0.7	2.9	0.8	0
0.7	2.9	0.8	0

To construct actual Dataset II by using Dataset I having binary type of defect categories, first, the number of total production of each batch was fixed to 120, which is the average of x1. Then, defect counts were calculated using simple direct proportion and they were rounded to the closest integer if necessary. The total number of records after sampling was 1389, 61 of which was for the non-defective items and the remaining was for defective items. The nominal variable created using the following coding schema in Table 14 has seven levels. The values of the variable y in Table 14 can be expressed as follows:

0: product is non-defective

1: the main reason for rejecting the product is of defect type “y2”

2: the main reason for rejecting the product is of defect type “y3”

3: the main reason for rejecting the product is of defect type “y6”

4: the main reason for rejecting the product is of defect type “y8”

5: the main reason for rejecting the product is of defect type “y9”

6: the main reason for rejecting the product is of defect type “y10”

Table 14: Coding Schema for actual dataset

y2	y3	y6	y8	y9	y10	y
0	0	0	0	0	0	**0**
1	0	0	0	0	0	**1**
0	1	0	0	0	0	**2**
0	0	1	0	0	0	**3**
0	0	0	1	0	0	**4**
0	0	0	0	1	0	**5**
0	0	0	0	0	1	**6**

5.3.3 Dataset III

According to the company, the first and second defect types labeled as y2 and y3, respectively are more important to represent quality than the others. In the term of the data collected these two defect types have an increase and this situation causes some troubles for the company. Second type of defect causes lots of material lost. To prevent or at least to decrease this defect type will allow the company to save money. Trouble with the first defect type is much more different. Although high defective percentage of this defect type costs much, the most important problem with this defect type is the fact that the company can not decide whether this defect type has occurred. It can only be determined by the customers while using the product. Determining and controlling influential factors which cause the first defect type, and also predicting this quality characteristic accurately before sending the product to customers for use will provide company with competitive advantage and loyalty of customers. Hence, the Dataset III was prepared to satisfy these requirements. The same approach used to prepare Dataset II was also used for preparing Dataset III. The main difference is that, unlike all defect types, only two of them, y2 and y3, were considered here. In other words, cases having none of these two defect types are categorized in the same class. As a result, there are three classes to be represented by the nominal quality variable y. The values of the y can be expressed as follows:

0: product is either non-defective or the main reason for rejecting the product is neither defect type of “y2” nor defect type of “y3”

1: the main reason for rejecting the product is of defect type “y2”

2: the main reason for rejecting the product is of defect type “y3”

During the preparation of the Dataset III, the number of total production of each batch was fixed to 100. After all steps of sampling were done, the Dataset III contains 36 process variables and only one response variable of type nominal with three levels for 809 cases.

5.4 Logistic Regression Modeling for Classification

One of the traditional techniques used to determine the most influential variables involved in a production process is the regression analysis. Therefore, in this study, first regression approach was used to develop a model which relates defect types to input factors. Name of the logistic models whose details are given in the following are coded for readability. Model names and their descriptions are shown in Table 15.

Table 15: Descriptions for logistic models

Model Name	Model type	Description	Data
Logit Model I	Main effects	• Developed to classify quality of products • Quality variable has 7 levels. Six of them represent quality reasons (type of defects) that cause rejecting product, one stands for representing non-defective products	Dataset II
Logit Model II	Main effects and two-way interactions.		
Logit Model III	Main effects	• Developed to classify quality of products • Quality variable has 3 levels. Two of them represent quality reasons (two important type of defects) that cause rejecting product, one stands for representing the products which either non-defective or rejected for other quality reasons	Dataset III
Logit Model IV	Main effects and two-way interactions.		

Logit Model I

The forward stepwise procedure was used to developed multinomial logistic model, which relates 36 input variables to a nominal response with 7 levels. During the development of the model, quasi-complete separation, a numerical problem that leads to either infinite or non-unique maximum likelihood parameter estimates, was faced. One of the possible reasons for this problem is sensitivity of the classification technique to the relative sizes of the response groups (Hosmer & Lemeshow, 2000), which is also the case for Dataset II having few records for category “0”. As a result, it may be concluded that the data does not fit the model adequately (McCullagh, 1980). However, as stated in (Allison, 1999), the variables with large coefficient can be discarded from the analysis to overcome this problem. In this sense, a series of models was developed by discarding predictor variables having large coefficient one by one. During the analysis, it was experienced that all models developed have large intercept values. In addition, standard errors of the intercepts are also very high. For that reason, above steps were repeated by using models without intercept. Numerical problems faced during the analysis are solved after discarding 21 of 36 process variables. 14 of the remaining variables were selected by stepwise procedure. Then variables found to be insignificant ($p > 0.05$) were removed from the model. As a result, final model has four variables, which are x7, x12, x22 and x23. Although all parameters involved in the model found to be significant individually and Pseudo R-Square statistics are high (Cox and Snell = 0.752, Nagelkerke = 0.768), goodness of fit of the overall model (Pearson = 0.00, Deviance = 0.00) shows that the model does not fit the data adequately. Classification accuracy of the model is found to be 55.4% (See Table 16 for classification details).

Logit Model I derived from the last iteration in stepwise procedure is as follows:

$\mathbf{Logit_1\ P(Y = 1)} = -\ 0.1498 * \mathbf{x12} - 0.142 * \mathbf{x22} + 0.06857 * \mathbf{x23} - 0.3577 * \mathbf{x7}$

$\mathbf{Logit_1\ P(Y = 2)} = -\ 0.132 * \mathbf{x12} - 0.4546 * \mathbf{x22} + 0.06685 * \mathbf{x23} - 0.3295 * \mathbf{x7}$

$\mathbf{Logit_1\ P(Y = 3)} = -\ 0.1563 * \mathbf{x12} - 0.457 * \mathbf{x22} + 0.06864 * \mathbf{x23} - 0.2996 * \mathbf{x7}$

$\mathbf{Logit_1\ P(Y = 4)} = -\ 0.146 * \mathbf{x12} - 0.413 * \mathbf{x22} + 0.06959 * \mathbf{x23} - 0.3416 * \mathbf{x7}$

$\mathbf{Logit_1\ P(Y = 5)} = -\ 0.1248 * \mathbf{x12} - 0.303 * \mathbf{x22} + 0.06028 * \mathbf{x23} - 0.3134 * \mathbf{x7}$

$\mathbf{Logit_1\ P(Y = 6)} = -\ 0.1289 * \mathbf{x12} - 0.4501 * \mathbf{x22} + 0.06521 * \mathbf{x23} - 0.3461 * \mathbf{x7}$

Y = 0 is the reference category.

Table 16: Classification table for Logit Model I

Observed	Predicted							
	1	**2**	**3**	**4**	**5**	**6**	**0**	**Percent Correct**
1	219	47	0	0	0	0	7	80.2%
2	135	501	0	0	0	0	0	78.8%
3	36	151	0	0	0	0	0	.0%
4	45	120	0	0	0	0	1	.0%
5	27	19	0	0	0	0	0	.0%
6	5	15	0	0	0	0	0	.0%
0	0	12	0	0	0	0	49	80.3%
Overall Percentage	33.6%	62.3%	.0%	.0%	.0%	.0%	4.1%	55.4%

Logit Model II

To improve the performance of Logit Model I, two-way interactions of the variables selected to be significant in the final model were taken into account. A new model was developed by including all two-way interactions of four main effects in the stepwise procedure. The final fitted model consists of two main effects and three interactions. It improves Logit Model I in terms of R-Square statistics (Cox and Snell = 0.770, Nagelkerke = 0.786). However, it fails to improve overall significance of the model (Pearson = 0.00, Deviance = 0.00) and classification accuracy (54.6%) (See Table 17 for classification details).

Logit Model II derived from the last iteration in stepwise procedure is as follows:

Logit$_2$ P(Y = 1) = 12.49 * **x22** - 1.126 * **x7** - 0.0286 * **x12** * **x22** + 0.003541 * **x12** * **x7** - 0.02472 * **x22** * **x7**

Logit$_2$ P(Y = 2) = 9.498 * **x22** - 0.6283 * **x7** - 0.01956 * **x12** * **x22** + 0.002174 * **x12** * **x7** - 0.02673 * **x22** * **x7**

Logit$_2$ P(Y = 3) = 8.612 * **x22** - 0.542 * **x7** - 0.01824 * **x12** * **x22** + 0.001874 * **x12** * **x7** - 0.02313 * **x22** * **x7**

Logit$_2$ P(Y = 4) = 8.902 * **x22** - 0.507 * **x7** - 0.01738 * **x12** * **x22** + 0.00175 * **x12** * **x7** - 0.02723 * **x22** * **x7**

Logit$_2$ P(Y = 5) = 9.725 * **x22** - 0.768 * **x7** - 0.02116 * **x12** * **x22** + 0.002485 * **x12** * **x7** - 0.02333 * **x22** * **x7**

Logit$_2$ P(Y = 6) = 11.6 * **x22** - 0.6166 * **x7** - 0.02214 * **x12** * **x22** + 0.002068 * **x12** * **x7** - 0.03695 * **x22** * **x7**

Y = 0 is reference category.

Table 17: Classification table for Logit Model II

	Predicted							
Observed	**1**	**2**	**3**	**4**	**5**	**6**	**0**	**Percent Correct**
1	187	73	0	0	0	0	13	68.5%
2	115	519	0	0	0	0	2	81.6%
3	34	152	0	0	0	0	1	.0%
4	43	121	0	0	0	0	2	.0%
5	24	20	0	0	0	0	2	.0%
6	4	16	0	0	0	0	0	.0%
0	0	12	0	0	0	0	49	80.3%
Overall Percentage	29.3%	65.7%	.0%	.0%	.0%	.0%	5.0%	54.4%

Logit Model III

The same as in the development of Logit Model I, during the development of Logit Model III, quasi-complete separation problem again led to infinite maximum likelihood estimates. Although removing variables having large coefficient from the analysis solved this problem, parameter estimates still reached very large values. Especially constant terms had extremely large values. The model developed without constant term produced better results relative to the intercept model but the values were still large. Differences between scales of the variables may cause the variables having high values to dominate the model. To prevent effect of differences between scales of the variables to the model, the variables were standardized. Quasi-complete separation problem was not observed with the model without constant term developed by standardized variables. Final model achieved by removing insignificant model parameters has four process variables.

Logit Model **III** derived from the last iteration of the stepwise procedure is as follows:

Logit$_3$ P(Y = 1) = - 0.3885 * **z11** + 0.7513 * **z22** + 0.3884 * **z27** - 0.4781 * **z29**

Logit$_3$ P(Y = 2) = - 0.2246 * **z11** - 1.406 * **z22** + 0.504 * **z27** -0.5654 * **z29**

Y=0 is reference category.

Although model parameters are significant individually, overall model is found to be insignificant (Pearson = 0.00, Deviance = 0.00). Pseudo R-Square statistics are low (Cox and Snell = 0. 455, Nagelkerke = 0. 511). Accuracy of classification seems good, (77.8%) but there is sufficient evidence to claim that the model does not fit the data adequately. Classification details are given in Table 18.

Table 18: Classification table for Logit Model III

Observed	Predicted			
	1	2	0	Percent Correct
1	225	0	8	96.6%
2	124	368	30	70.5%
0	12	6	36	66.7%
Overall Percentage	44.6%	46.2%	9.1%	77.8%

Logit Model IV

To investigate the effect of interactions, all possible two-way interactions of main effects included in Logit Model III were also added to the stepwise procedure. The model developed has three interactions besides the four main effects. The model using interactions are significantly better than the model including only main effects. Both Pseudo R-Square statistics and classification accuracy are improved (Cox and Snell = 0. 565, Nagelkerke = 0. 635, Accuracy = 80%). However there is no improvement on overall significance of the model (Pearson = 0.00, Deviance = 0.00). Thus, we can conclude that the model does not fit to the data. Classification results are shown in Table 19.

Table 19: Classification table for Logit Model IV

Observed	Predicted			
	1	2	0	Percent Correct
1	222	6	5	95.3%
2	112	382	28	73.2%
0	0	11	43	79.6%
Overall Percentage	41.3%	49.3%	9.4%	80.0%

The fitted logistic model including main effect and two-way interactions is as follows:

$$\textbf{Logit P(Y = 1)} = -0.8323 * \mathbf{z11} + 1.129 * \mathbf{z22} + 0.8182 * \mathbf{z27} - 0.4248 * \mathbf{z29} - 0.375 * \mathbf{z11} * \mathbf{z22} + 0.3914 * \mathbf{z11} * \mathbf{z27} + 0.4255 * \mathbf{z22} * \mathbf{z27}$$

$$\textbf{Logit P(Y = 2)} = -\ 0.9096 * \mathbf{z11} - 1.432 * \mathbf{z22} + 1.055 * \mathbf{z27} - 0.7153 * \mathbf{z29} - 1.189 * \mathbf{z11} * \mathbf{z22} + 0.4101 * \mathbf{z11} * \mathbf{z27} + 1.217 * \mathbf{z22} * \mathbf{z27}$$

Y=0 is reference category.

5.5 Decision Tree Modeling

Decision tree models were developed by using basic dataset and all dataset designs described in Section 5.3. Several models were constructed. For the readability of the work, each of the models is given a coded name. Table 20 describes decision tree models explained in the following.

Table 20: Description of Decision Tree Models

Model Name	Description	Data
CART Model 0	• Developed to predict quality • Quality variable is the proportion of total defective products (y1%) in the batches	Basic Dataset
CART Model I	• Developed to predict quality • Quality variable is the proportion of products rejected because of defect type I (y2%) in the batches	Basic Dataset
CART Model II	• Developed to predict quality • Quality variable is the proportion of products rejected because of defect type II (y3%) in the batches	Basic Dataset
CART Model III	• Developed to predict quality • Quality variable is the proportion of products rejected because of defect type III (y6%) in the batches	Basic Dataset
CART Model IV	• Developed to predict quality • Quality variable is the proportion of products rejected because of defect type IV (y8%) in the batches	Basic Dataset
CART Model V	• Developed to predict quality • Quality variable is the proportion of products rejected because of defect type V (y9%) in the batches	Basic Dataset

Table 20 (cont.): Description of Decision Tree Models

CART Model VI	• Developed to predict quality • Quality variable is the proportion of products rejected because of defect type VI (y10%) in the batches	Basic Dataset
C5.0 MODEL I	• Developed to classify quality of products • Quality variable has 7 levels. Six of them represent quality reasons (type of defects) that cause rejecting product, one stands for non-defective products	Dataset II
C5.0 MODEL II	• Developed to classify quality of products • Quality variable has 3 levels. Two of them represent quality reasons (two important types of defects) that cause rejecting product, one stands for the product which is either non-defective or rejected for other quality reasons	Dataset III

5.5.1 Prediction Using CART

CART models can be used for either classification or prediction purposes. In this section, results of CART analysis using basic dataset with 92 records are given. The models were developed to predict the proportion of defective products in the batches. Each of the defect type was modeled individually. The proportion of total defectives was also modeled. Since average values of the process variables were used during the preparation of the basic dataset, these models represent average conditions for the batches to observe particular defective proportions. Since the objective of the company is to reduce the proportion of the defective items, rules that predict minimum defective proportions were given here. Other rules can be extracted from the tree of each model if needed (see Appendix A for all possible predictions that the decision tree graph of each model provides). To evaluate these models, several criteria can be used. The following are the extensively used ones:

- *Minimum Error:* minimum difference between the observed and predicted values.

- *Maximum Error:* maximum difference between the observed and predicted values.

- *Mean Error:* the mean errors of all records; this indicates whether there is a systematic bias in the model.

- *Mean Absolute Error:* the mean of the absolute values of the errors of all records.

- *Standard Deviation:* the standard deviation of all errors.

- *Linear Correlation:* the linear correlation between the predicted and actual values; values close to +1 indicate a strong positive association; values close to 0 indicate a weak association, and values close to -1 indicate a strong negative association.

CART MODEL 0

CART Model 0 can guide the overall optimization of the process parameters. Process variables included in the model can be considered as influential on all defect types. It can be also used with individual models of each of the defect type. Following rules are taken from the model for proportion of total defectives and gives influential factors and their values to achieve minimum defective proportions in total. Because average of total defective proportion for all batches is about 12%, prediction values for the proportion of the defective items are quite high.

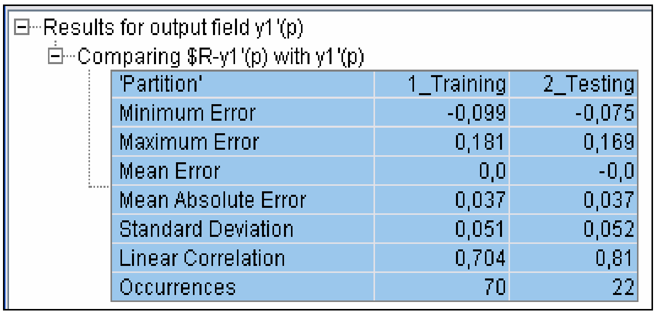
Rules Minimizing Proportion of Total Defectives

<u>Rule 1:</u> (number of cases = 36)

$$IF \quad x16 > 158.65 \quad AND \quad x29 > 3.325 \quad THEN \quad y1\% = 0.084$$

Rule 2: **(**number of cases = 5**)**

$$IF \quad x16 > 158.65 \quad AND \quad x29 \leq 3.325 \quad AND \quad x3 > 31 \quad AND$$
$$x4 > 12.295 \quad AND \quad x17 > 265.55 \quad THEN \quad y1\% = 0.042$$



Results for output field y1'(p)

Comparing $R-y1'(p) with y1'(p)

'Partition'	1_Training	2_Testing
Minimum Error	-0,099	-0,075
Maximum Error	0,181	0,169
Mean Error	0,0	-0,0
Mean Absolute Error	0,037	0,037
Standard Deviation	0,051	0,052
Linear Correlation	0,704	0,81
Occurrences	70	22

Figure 10: Performance Statistics of CART Model 0

As it is shown in Figure 10, the model developed is successful on the test data too. The predicted values are highly correlated with the actual values.

CART MODEL I

Extracted rules from this model give the best conditions to minimize the first type of defect. Rule 3 shows that by optimizing two of the process variables, product returns from the customers due to the first type of defect, which is one of the important quality problem of the company, can be prevented. This rule also has high support value, which is 51 of 92 cases. Predicted proportions of the first defect type provided by the remaining two rules are also low.

Rules Minimizing Proportion of Defectives

Rule 3: **(**number of cases = 51**)**

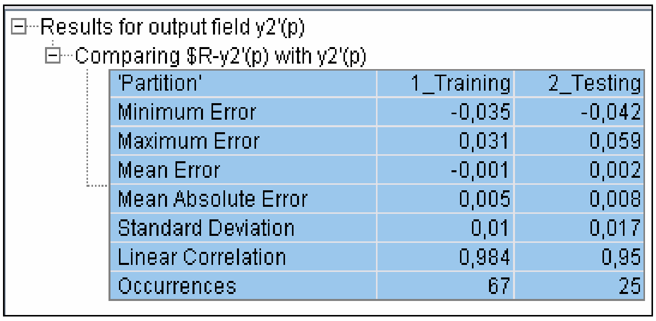
$$IF \quad x14 > 4.724 \quad AND \quad x22 \leq 17.2 \quad THEN \quad y2\% = 0.0$$

Rule 4: **(**number of cases = 6**)**

$$IF \quad x14 > 4.724 \quad AND \quad x22 > 17.2 \quad AND \quad x9 > 3.183$$
$$AND \quad x16 \leq 18.8798 \quad THEN \quad y2\% = 0.001$$

Rule 5: **(**number of cases = 8**)**

$$IF \quad x14 > 4.724 \quad AND \quad x22 > 17.2 \quad AND \quad x9 > 3.183$$
$$AND \quad x16 > 18.8798 \quad AND \quad x33 > 0.174 \quad AND \quad y2\% = 0.013$$



⊟ Results for output field y2'(p)
⊟ Comparing $R-y2'(p) with y2'(p)

'Partition'	1_Training	2_Testing
Minimum Error	-0,035	-0,042
Maximum Error	0,031	0,059
Mean Error	-0,001	0,002
Mean Absolute Error	0,005	0,008
Standard Deviation	0,01	0,017
Linear Correlation	0,984	0,95
Occurrences	67	25

Figure 11: Performance Statistics of CART Model I

In addition to achieving low defect proportion, performance statistics of the model is also desirable. Errors are very low and correlation between actual and predicted values is very close to 1.

CART MODEL II

Results of CART Model II are also important since it is related with the second type of defect, which causes lots amount of material loss. According to the sixth rule, this defect type can be decreased below the maximum acceptable defective proportion stated by the company. In addition, a simple rule indicates the lower limit for the process variable x22, and the rule says that the variable have to be greater than this lover limit to prevent an increase in the second type of defect.

Rules Minimizing Proportion of Defectives

<u>Rule 6:</u> (number of cases = 24**)**

$$IF\ \ x22 > 13.125 \quad AND \quad x29 > 3.304 \quad AND \quad x11 > 20.339$$
$$AND \quad x3 <= 37.5 \quad AND \quad x9 \leq 3.216 \ \ THEN \quad y3\% = 0.013$$

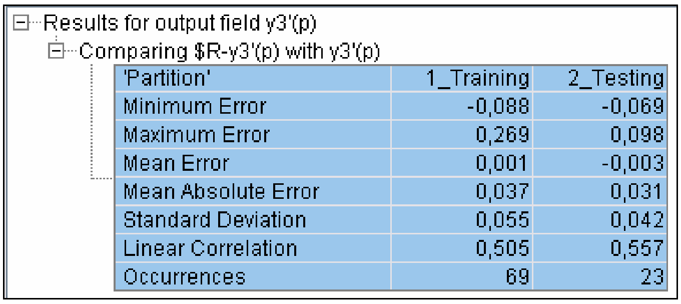
<u>Rule 7:</u> (number of cases = 6**)**

$$IF\ \ x22 > 13.125 \quad AND \quad x29 \leq 3.304$$
$$AND \quad x20 \leq 41.32 \ \ THEN \quad y3\% = 0.027$$

Rule Maximizing Proportion of Defect Type "y3"

(number of cases = 40**)**

$$IF\ \ x22 \leq 13.125 \quad THEN \quad y3\% = 0.088$$



Results for output field y3'(p)

Comparing $R-y3'(p) with y3'(p)

'Partition'	1_Training	2_Testing
Minimum Error	-0,088	-0,069
Maximum Error	0,269	0,098
Mean Error	0,001	-0,003
Mean Absolute Error	0,037	0,031
Standard Deviation	0,055	0,042
Linear Correlation	0,505	0,557
Occurrences	69	23

Figure 12: Performance Statistics of CART Model II

The performance of CART Model II is not as good as CART Model I. Mean absolute error is high and the linear correlation between actual and predicted values is lower than the previous model. However, the statistics are still at acceptable levels and the model can be used as a result.

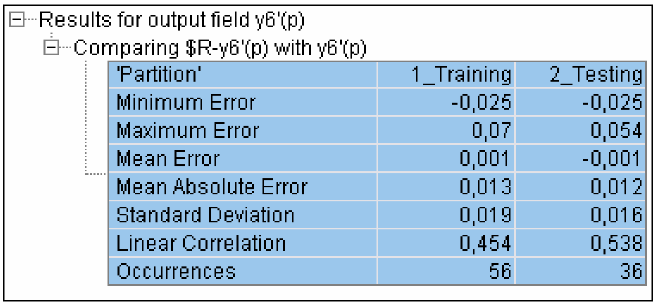
CART MODEL III

CART Model III was developed to predict the proportion of the third frequently observed defect type. Rule 8 gives the conditions to minimize this defect type. These conditions are also compatible with the rules derived from CART Model II.

Rule Minimizing Proportion of Defectives

<u>Rule 8:</u> (number of cases = 48**)**

$$IF \quad x22 > 13.275 \quad AND \quad x9 > 3.095 \quad THEN \quad y6\% = 0.006$$



Results for output field y6'(p)

Comparing $R-y6'(p) with y6'(p)

'Partition'	1_Training	2_Testing
Minimum Error	-0,025	-0,025
Maximum Error	0,07	0,054
Mean Error	0,001	-0,001
Mean Absolute Error	0,013	0,012
Standard Deviation	0,019	0,016
Linear Correlation	0,454	0,538
Occurrences	56	36

Figure 13: Performance Statistics of CART Model III

Although linear correlation of the model is low, any other performance statistics for CART Model III are acceptable.

CART MODEL IV

While focusing on frequently observed defects and preventing or reducing them, decreasing the proportion of infrequent defect types at the same time will provide manufacturing companies with well-tuned processes. CART Model IV predicts the proportion of one of the infrequently observed defect types. There are two rules that help to minimize this defect. Both rules contain the same process variables with the same suggested regions except for the last one, which is x5. Although predicted proportions for both rules are at acceptable levels, minimum of them can be preferred.

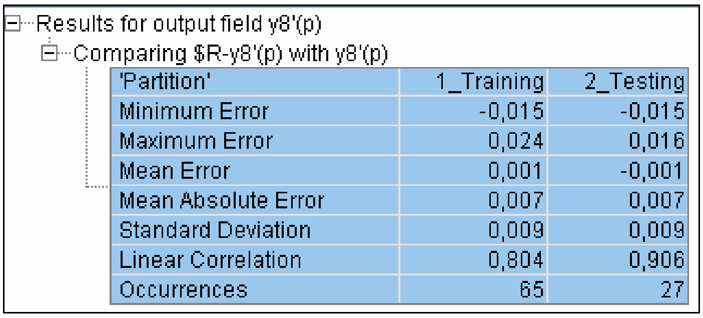
Rules Minimizing Proportion of Defectives

Rule 9: (number of cases = 27**)**

$$IF \quad x9 > 3.02 \quad AND \quad x26 \le 1425.982 \quad AND \quad x25 \le 6.533$$
$$AND \quad x6 > 7.917 \quad AND \quad x5 > 13.165 \quad THEN \quad y8\% = 0.005$$

Rule 10: (number of cases = 31**)**

$$IF \quad x9 > 3.02 \quad AND \quad x26 \le 1425.982 \quad AND \quad x25 \le 6.533$$
$$AND \quad x6 > 7.917 \quad AND \quad x5 \le 13.165 \quad THEN \quad y8\% = 0.01$$



Results for output field y8'(p)

Comparing $R-y8'(p) with y8'(p)

'Partition'	1_Training	2_Testing
Minimum Error	-0,015	-0,015
Maximum Error	0,024	0,016
Mean Error	0,001	-0,001
Mean Absolute Error	0,007	0,007
Standard Deviation	0,009	0,009
Linear Correlation	0,804	0,906
Occurrences	65	27

Figure 14: Performance Statistics of CART Model IV

Performance of the overall model shown in Figure 14 is good. Because, errors are small and the correlation is high. In addition, performance of the testing data is as good as the training dataset.

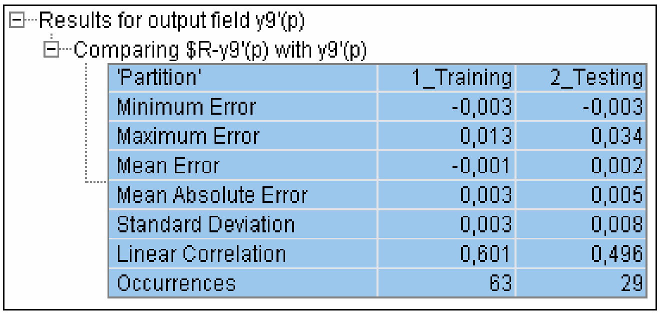
CART MODEL V

CART Model V was developed to predict proportion of another infrequent defect type, which is defect type V. Conditions shared by 84 of 92 cases, where in which this defect type is minimum, are given by Rule 11.

Rule Minimizing Proportion of Defectives

<u>Rule 11:</u> (number of cases = 84**)**

$$IF \quad x21 > 49.181 \quad AND \quad x26 \leq 1426.955 \quad THEN \quad y9\% = 0.003$$



Results for output field y9'(p)
Comparing $R-y9'(p) with y9'(p)

'Partition'	1_Training	2_Testing
Minimum Error	-0,003	-0,003
Maximum Error	0,013	0,034
Mean Error	-0,001	0,002
Mean Absolute Error	0,003	0,005
Standard Deviation	0,003	0,008
Linear Correlation	0,601	0,496
Occurrences	63	29

Figure 15: Performance Statistics of CART Model V

The errors produced by the model are desirably low. However, the correlation between actual and predicted values is better for the training data. This value decreases to near 0.5 for the test set.

CART MODEL VI

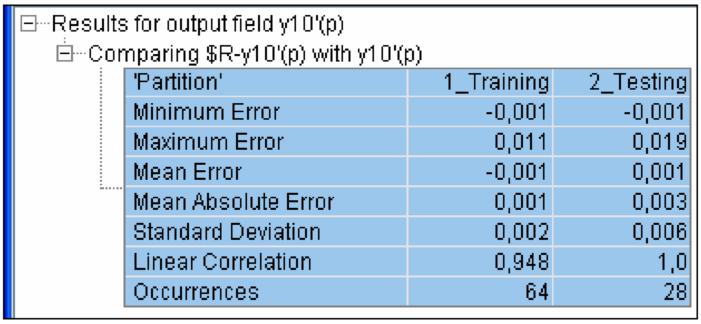
Observation of defect type VI is very rare relative to other defect types. For that reason, actual values for the proportion of this defect type are usually 0.

Rule Minimizing Proportion of Defectives

<u>Rule 12:</u> (number of cases = 91**)**

$$IF \quad x20 > 39.067 \quad THEN \quad y10\% = 0.001$$

Although performance statistics for CART Model VI seem to be successful, there is no acceptable decision tree model. Algorithm stopped after one record was separated from the dataset and no further splitting was performed. In addition, domain experts found rules of the model meaningless.



Results for output field y10'(p)

Comparing $R-y10'(p) with y10'(p)

'Partition'	1_Training	2_Testing
Minimum Error	-0,001	-0,001
Maximum Error	0,011	0,019
Mean Error	-0,001	0,001
Mean Absolute Error	0,001	0,003
Standard Deviation	0,002	0,006
Linear Correlation	0,948	1,0
Occurrences	64	28

Figure 16: Performance Statistics of CART Model VI

5.5.2 Classification Using C5.0

In previous section, batch-based analysis was performed to predict proportion of defective products in the bathes under given conditions. In this section, datasets I, II, and III described in Chapter 5 were used to perform product-based analysis. These sets were prepared under the assumption of being representative of batch values for all individual items in the associated batches.

Classification Using Dataset I

This analysis was carried out to classify products as (defective, non-defective) or (reason to reject is defect type$_1$,..., reason to reject is defect type$_n$, accepted) using Dataset I. However, representing both defective and non-defective products with the same process values and domination of number of non-defective products in the data cause the analysis to fail. For both cases no splitting were achieved and all of the product were classified as non-defective.

Classification of All Defect Types Using Dataset II

In Dataset II, the categorical response variable has seven levels; one for non-defective case, and each of the remaining levels stand for one of the six defect types. This categorization allows us to describe all cases within one model if a reliable model is achieved. The rules extracted from the tree model can be used to determine influential factors that cause defects on the products. Rules are also representative for process settings that produce non-defective products.

C5.0 MODEL I

The C5.0 algorithm selected 11 process variables to build model. These are, in the order of importance, x22, (x30, x32), (x2, x29, x12), x28, (x9, x19), x35 and x26. Variables in parenthesis are equally important. Variable X22 is found to be the most influential parameter and important for all categories. X30 is also found to be important and used at least in one rule of each category.

The model classifies five classes, which are non-defectives and the first four defect types. Records for rarely observed other two defect types that cause rejecting a product were mostly assigned to be the second defect type. Totally, 14 rules were extracted for five categories. According to categories, influential process variables are found to be as following (in the order of importance):

- **0** *(non-defective)*: x22, (x30, x32), (x2, x12)
- **1** *(the main reason for rejecting the product is of defect type “y2”)*: x22, x32, x12, x30, x19, x9, x26
- **2** *(the main reason for rejecting the product is of defect type “y3”)*: x22, (x30, x32), (x2, x29, x12), x28, (x9, x19), x26

- **3** *(the main reason for rejecting the product is of defect type "y6")*: x22, x30, x29, x28, x9, and x35

- **4** *(the main reason for rejecting the product is of defect type "y8")*: x22, x30, and x29.

Generated Rules

The values in parenthesis following the rule number stand for instances, the number of records to which rule is applied, and confidence value of the rule, respectively.

<u>Rule 1:</u> (8; 1.0)

$IF \quad \mathrm{x22} \le 14.350 \quad AND \quad \mathrm{x30} \le 1.880 \quad AND \quad \mathrm{x2} > 23$
$THEN \quad y = 0$

<u>Rule 2:</u> (17; 1.0)

$IF \quad \mathrm{x22} > 14.350 \quad \mathrm{A}ND \quad \mathrm{x32} \le 0.184 \quad THEN \quad y = 0$

<u>Rule 3:</u> (15; 1.0)

$IF \quad \mathrm{x22} > 14.350 \quad AND \quad \mathrm{x32} > 0.184$
$AND \quad x12 > 350 \quad THEN \quad y = 0$

<u>Rule 4:</u> (257; 0.681)

$IF \quad \mathrm{x22} > 14.350 \quad \mathrm{A}ND \quad \mathrm{x32} > 0.184 \quad AND \quad x12 \le 350$
$AND \quad x30 > 1.895 \quad AND \quad x19 > 159.5 \quad AND \quad x9 \le 3.206$
$THEN \quad y = 1$

Rule 5: **(12; 0.417)**

$IF \quad x22 > 14.350 \quad AND \quad x32 > 0.184 \quad AND \quad x12 \leq 350$
$AND \quad x30 > 1.895 \quad AND \quad x19 > 159.5 \quad AND \quad x9 > 3.206$
$AND \quad x26 \leq 1422.11 \quad THEN \quad y = 1$

Rule 6: **(42; 0.381)**

$IF \quad x22 > 14.350 \quad AND \quad x32 > 0.184 \quad AND \quad x12 \leq 350$
$AND \quad x30 > 1.895 \quad AND \quad x19 > 159.5 \quad AND \quad x9 > 3.206$
$AND \quad x26 > 1422.11 \quad THEN \quad y = 2$

Rule 7: **(39; 0.769)**

$IF \quad x22 > 14.350 \quad AND \quad x32 > 0.184 \quad AND \quad x12 \leq 350$
$AND \quad x30 \leq 1.895 \quad THEN \quad y = 2$

Rule 8: **(41; 0.634)**

$IF \quad x22 > 14.350 \quad AND \quad x32 > 0.184 \quad AND \quad x12 \leq 350$
$AND \quad x30 > 1.895 \quad AND \quad x19 \leq 159.5 \quad THEN \quad y = 2$

Rule 9: **(26; 0.577)**

$IF \quad x22 \leq 14.350 \quad AND \quad x30 \leq 1.88 \quad AND \quad x2 \leq 23 \quad THEN \quad y = 2$

Rule 10: **(407; 0.688)**

$IF \quad x22 \leq 14.350 \quad AND \quad x30 > 1.880 \quad AND \quad x29 \leq 3.355$
$AND \quad x28 \leq 15.797 \quad AND \quad x9 \leq 3.260 \quad AND \quad x35 > 0.080$
$THEN \quad y = 2$

Rule 11: **(40; 0.525)**

$IF \quad x22 \leq 14.350 \quad AND \quad x30 > 1.880 \quad AND \quad x29 \leq 3.355$
$AND \quad x28 \leq 15.797 \quad AND \quad x9 > 3.260 \quad THEN \quad y = 2$

Rule 12: **(20; 0.55)**

$IF \quad x22 \leq 14.350 \quad AND \quad x30 > 1.880 \quad AND \quad x29 \leq 3.355$
$AND \quad x28 \leq 15.797 \quad AND \quad x9 \leq 3.260 \quad AND \quad x35 \leq 0.080$
$THEN \quad y = 3$

Rule 13: **(25; 0.68)**

$IF \quad x22 \leq 14.350 \quad AND \quad x30 > 1.880 \quad AND \quad x29 \leq 3.355$
$AND \quad x28 > 15.797 \quad THEN \quad y = 3$

Rule 14: **(19; 0.579)**

$IF \quad x22 \leq 14.350 \quad AND \quad x30 > 1.880 \quad AND \quad x29 > 3.355$
$THEN \quad y = 4$

Overall classification accuracy of C5.0 Model I is 60.3%. The strongest rules provided by the model are the 4th and 10th rules and these rules were applied to 257 and 407 records, respectively. Confidence levels of both rules are about 70%. Fourth rule describes conditions for category “1” and 10th rule describes conditions for category “2”. Other rules for these categories have low support values (under 5% of all records). Support values for 6th, 7th, 8th, and 11th rules are between 4% and 5%. Impact of different variables included in these rules in the production process may be examined by conducting a controlled experiment including these variables. Gain charts of the model when the outcome of interest is category “1” and “2” are illustrated in Figure 17 and Figure 18 respectively. It can be said that gains provided by the model for category “1” and “2” are very close to the best line indicating perfect confidence, thus it can be said that performance of the model is good for these categories (See Appendix D for a brief tutorial on cumulative gain charts)

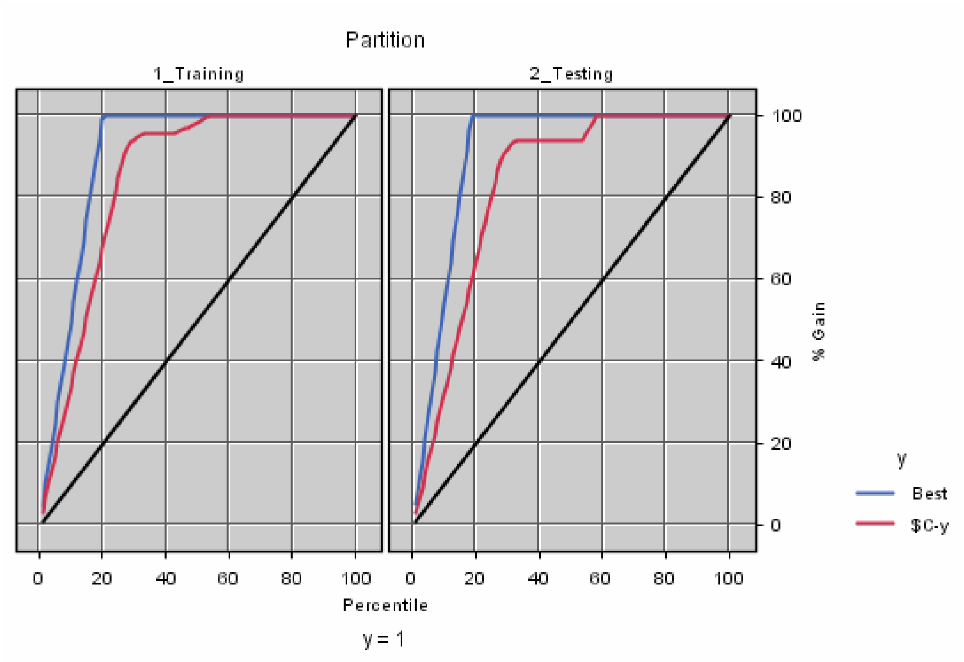


Figure 17: Gain chart of C5.0 Model I when the hit category is "1"

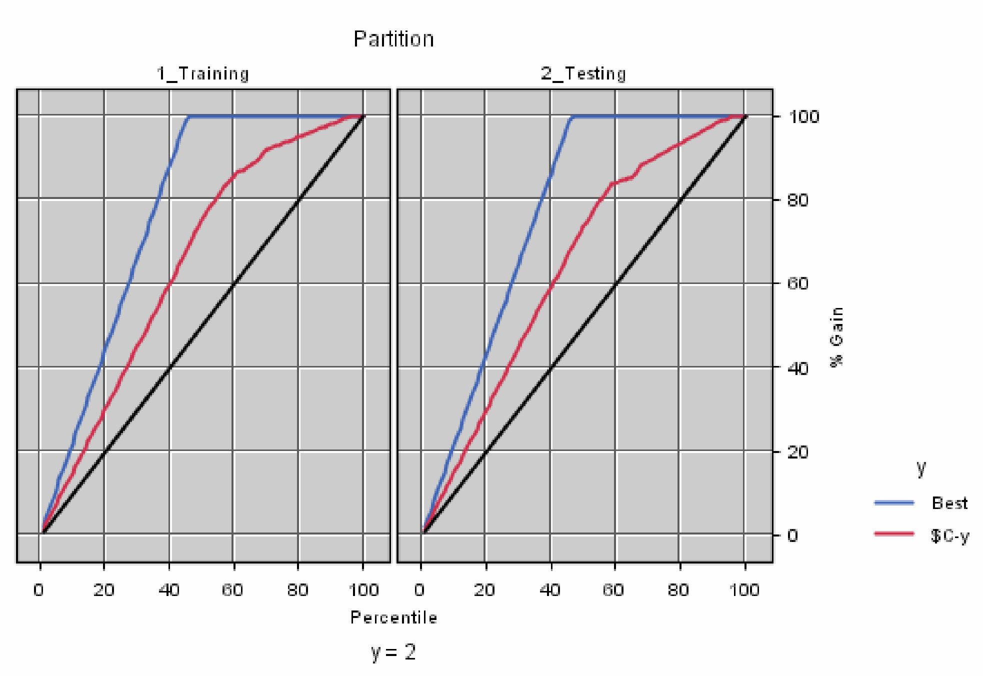


Figure 18: Gain chart of C5.0 Model I when the hit category is "2"

Conditions for non-defective cases whose category is “0” are provided in the first three rules. These rules have very low support values. However, when we examined the data, all of the records belong to this category are just 4.1% of all dataset. In addition to this, confidence levels of rules are 1.0. That means, model correctly classifies all non-defective cases. Gain chart for category “0” illustrated in Figure 19 also shows that performance of the model in classifying non-defective products is good. Thus, the rules can be generalized and the model can be used to classify non-defective products.

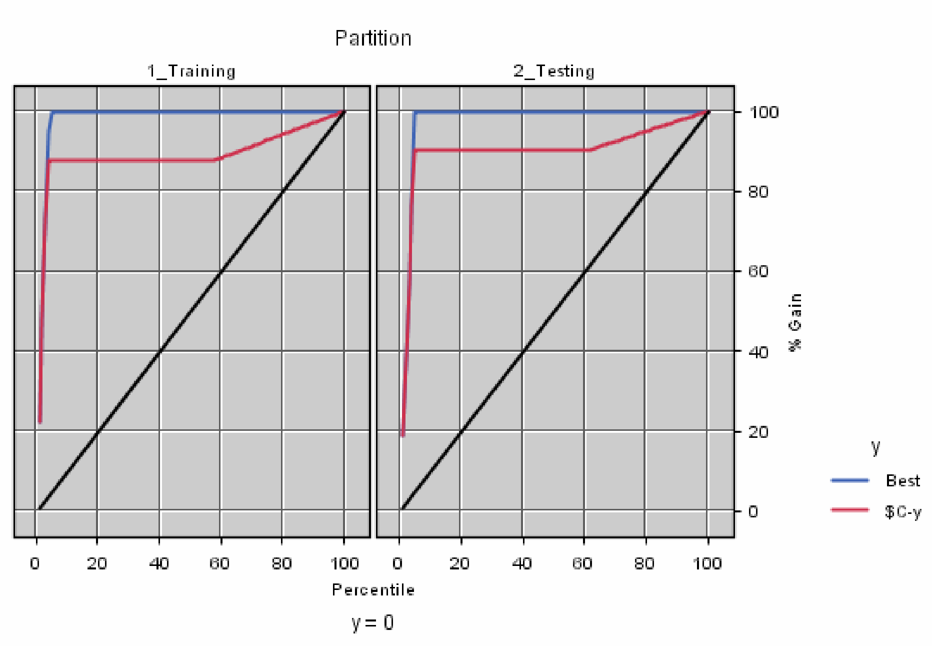


Figure 19: Gain chart of C5.0 Model I when the hit is category “0”

Although categories “3” and “4” are included in predicted values and three rules (12th, 13th and 14th) are extracted, most of the records belong to these categories were assigned to category “2”. In addition, support levels of the extracted rules are very low. Categories not exist in predicted values were also assigned to category “2”. Figure 20 and Figure 21 shows performance of the model on predicting categories “3” and “4”. As it shown in the figures, model line is very close to the base line indicating a perfectly random distribution of hits where confidence becomes irrelevant.

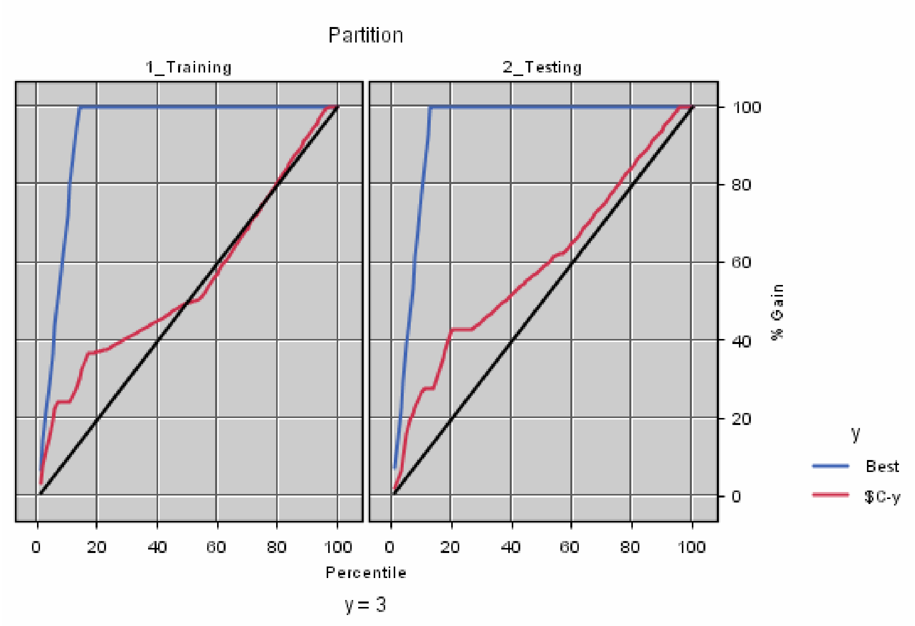


Figure 20: Gain chart of C5.0 Model I when the hit is category “3”

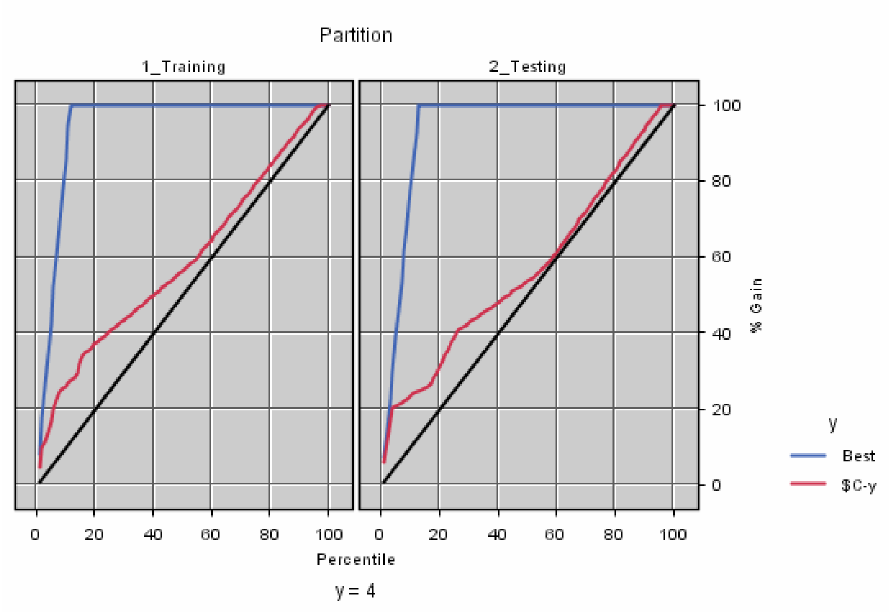


Figure 21: Gain chart of C5.0 Model I when the hit is category “4”

Coincidence matrix for predicted categories is shown in Table 21. It can be concluded that model is successful to predict categories “0”, “1”, “2”. Rules for categories “3” and “4” can be considered to determine influential parameters on those defect types instead of prediction. You can look at Figure 37 in Appendix B for decision tree graph of C5.0 Model I.

Table 21: Coincidence Matrix For C5.0 Model I

Training	**0**	**1**	**2**	**3**	**4**	**5**	**6**
0	**40**	0	0	0	0	0	0
1	0	**180**	15	0	0	0	0
2	0	36	**388**	15	2	0	0
3	0	20	80	**28**	5	0	0
4	0	21	78	2	**11**	0	0
5	0	9	23	0	0	**0**	0
6	0	3	11	0	1	0	**0**

Classification of the First and Second Type Defects Using Dataset III

To prevent the first and second type defects, which are observed most as a cause to reject a product, has a priority for the company. Model presented in this section was built using Dataset III prepared according to requirements of the company stated in Section 5.3.3.

C5.0 Model II

The C5.0 Model II finds nine process variables to be influential on the response and it also extracts ten rules associated with these significant input variables. The variables are, in the order of importance, x22, (x8, x27), (x30, x35, x9), (x2, x12) and x19. Note that variable in parenthesis have the same order of importance. As in the C5.0 Model I, variable x22 is the most informative variable for all categories. Variable x27 is also important and used at least in one rule of each category.

Totally, 10 rules were extracted for three categories. According to categories, influential process variables are as follows (in the order of importance):

- **0** *(non-defective)*: x22, (x8, x27), (x30, x35), x2
- **1** *(the main reason for rejecting the product is of defect type I)*: x22, x27, x9, x12, x19
- **2** (*the main reason for rejecting the product is of defect type II)*: x22, (x8, x27), (x30, x35, x9), (x2, x12) and x19

Generated Rules

<u>Rule 1:</u> (17; 1.0)

$$IF \quad x22 \leq 14.35 \quad AND \quad x8 \leq 35 \quad THEN \quad y = 0$$

<u>Rule 2:</u> (16; 1.0)

$$IF \quad x22 > 14.35 \quad AND \quad x27 \leq 4.2 \quad AND \quad x35 > 0.088$$
$$THEN \quad y = 0$$

<u>Rule 3:</u> (5; 1.0)

$$IF \quad x22 \leq 14.35 \quad AND \quad x8 > 35 \quad AND \quad x30 \leq 1.88$$
$$AND \quad x2 > 23 \quad THEN \quad y = 0$$

Rule 4: **(198; 0.828)**

$$IF \quad x22 > 14.35 \quad AND \quad x27 > 4.2 \quad AND \quad x9 \leq 3.216$$
$$AND \quad x12 > 305 \quad AND \quad x19 > 15.95 \quad THEN \quad y = 1$$

Rule 5: **(13; 1.0)**

$$IF \quad x22 \leq 14.35 \quad AND \quad x8 > 35 \quad AND \quad x30 \leq 1.88$$
$$AND \quad x2 \leq 23 \quad THEN \quad y = 2$$

Rule 6: **(268; 1.0)**

$$IF \quad x22 \leq 14.35 \quad AND \quad x8 > 35 \quad AND \quad x30 > 1.88 \quad THEN \quad y = 2$$

Rule 7: **(8; 0.875)**

$$IF \quad x22 > 14.35 \quad AND \quad x27 \leq 4.2 \quad AND \quad x35 \leq 0.088 \quad THEN \quad y = 2$$

Rule 8: **(34; 0.765)**

$$IF \quad x22 > 14.35 \quad AND \quad x27 > 4.2 \quad AND \quad x9 > 3.216 \quad THEN \quad y = 2$$

Rule 9: **(9; 0.889)**

$$IF \quad x22 > 14.35 \quad AND \quad x27 > 4.2$$
$$AND \quad x9 \leq 3.216 \quad AND \quad x12 \leq 305 \quad THEN \quad y = 2$$

Rule 10: **(18; 0.778)**

$$IF \quad x22 > 14.35 \quad AND \quad x27 > 4.2 \quad AND \quad x9 \leq 3.216$$
$$AND \quad x12 > 305 \quad AND \quad x19 \leq 15.95 \quad THEN \quad y = 2$$

Fourth rule gives the important variables and threshold values for the defect type that cannot be measured by the company. This is the second strongest rule provided by the model. It has 82.8% confidence level and 33.7% support. The strongest rule is the sixth rule with a 100% confidence level and 44.7% support. This rule can be used to control and decrease material lost caused by the second defect type. Prediction accuracy of the model is desirably high (see coincidence matrix in Table 22 and gain charts for categories “0”, “1” and “2” in Figure 22, Figure 23 and Figure 24, respectively). Overall classification accuracy is 92.15%. First, second and third rules for non-defective cases have low support values as it is the case in C5.0 Model I. They will be treated as in those models since all records in this category were predicted with 100% confidence, thus, these rules will be accepted. You can look at Figure 38 in Appendix B for decision tree graph of C5.0 Model II.

Table 22: Coincidence Matrix For C5.0 Model 2

Training	**0**	**1**	**2**
0	**38**	0	0
1	0	**164**	14
2	0	34	**336**

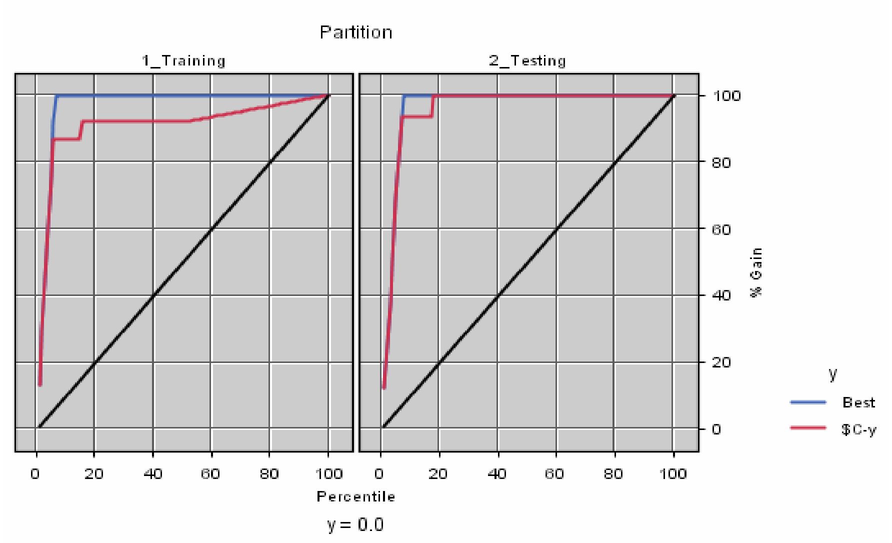


Figure 22: Gain chart of C5.0 Model II when the hit category is “0”

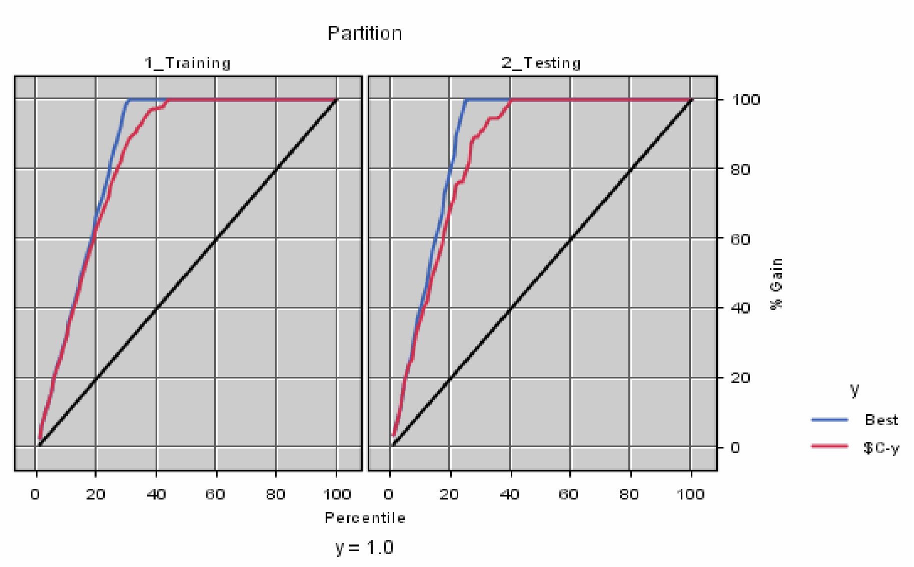


Figure 23: Gain chart of C5.0 Model II when the hit category is "1"

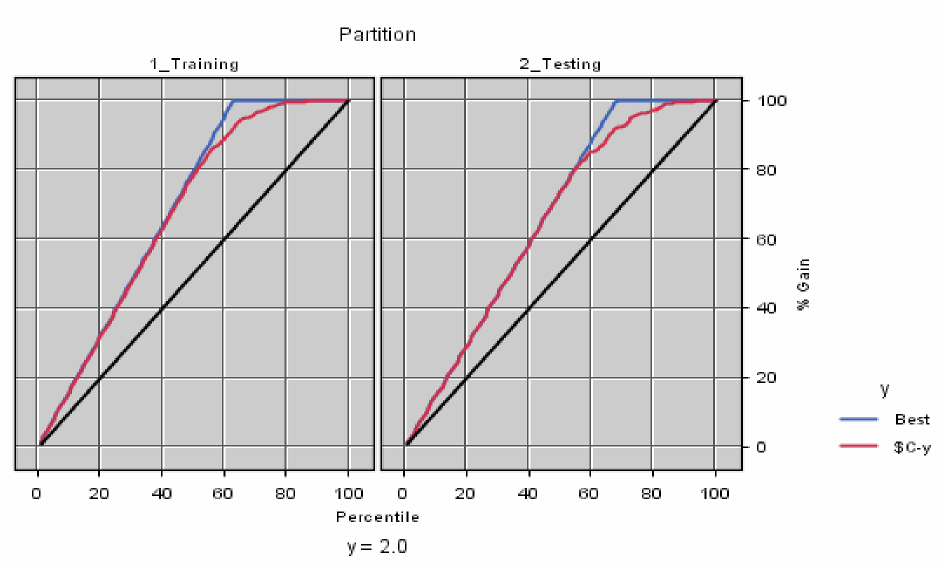


Figure 24: Gain chart of C5.0 Model II when the hit category is "2"

5.5.3 Combination of Tree Results

Tree models can be used either individually or together according to the requirements. For instance, C5.0 Model II can be used alone to optimize the manufacturing process if the aim is reducing "y2" and “y3” types of defects. In this section, all generated models (CART and C5.0 models) were combined to find optimal regions for all defect types together. In other words, the best regions in factor space were examined to optimize the entire process.

Firstly, generated models were summarized. To do this, the strongest rules and the rules found to be acceptable for defective and non-defective situations were selected. The best regions for the variables pointed out by the rules were determined. Summary of models are arranged in Table 23 and Table 24. After that, the most important variables selected by the models were explored. These were found to be the variables x9, x22, x29, x30 and x32. Then, the best intervals defined by the rules for these variables were examined. Figure 25, 26, 27, 28 and 29 show the intervals for the important variables, which are optimum for the defect types pointed by the rules. To illustrate, in Figure 25, according to Rule 6 extracted from CART models, [2.98, 3.216] is the best interval for x9 not to observe defect type II as a main reason for rejecting a product. In Figure 26, Rule 2 extracted from C5.0 Model I shows that the interval [14.35, 33.42] for x22 is the best interval for all defect types; in other words, product produced in this range is non-defective.

Generally, intervals for a variable pointed by a few rules do not contradict. Optimum values on which all related rules are agreed can be found out for such variables. On the contrary, variables included in many rules may be assigned to intervals that do not overlap. All disjoint regions can be thought as possible good regions and should be studied. Design of experiments which is the most commonly used statistical technique in the manufacturing industry to determine the impact of different levels of the parameters and interactions of them on the process can be used here to decide on the best production regions in factor space. Suggested experimental design is shown in

Table 25. In the table, shaded rows, the important variables proposed by the models, are the design variables. Levels of the design variables were selected from different disjoint parts of the suggested intervals (See Figure 25, 26, 27, 28 and 29). Other variables included in the rules were considered as constant in the intersections of the intervals suggested by the rules (factors with 1 level). If those variables are controllable, their settings need to be changed to shift natural variation of those to the suggested interval as much as possible.

Table 23: Summary of CART Models (suggested regions)

		x3	x4	x5	x6	x9	x11	x14	x16	x17	x20	x21	x22	x25	x26	x29	x33
CART MODEL 0	**R1 8.4%**								>15.865							>3.325	
	R2 4.2%	>31	>12.295						>15.865	>26.555						<3.325	
CART MODEL I	**R3 0%**							>4.724					<17.2				
	R4 0.1%					>3.183		>4.724	<18.88				>17.2				
	R5 0.13%					>3.183		>4.724	>18.88				>17.2				>0.174
CART MODEL II	**R6 1.3%**	<37.5				<3.216	>20.339						>13.125			>3.304	
	R7 2.7%										<41.32		>13.125			<3.304	
CART MODEL III	**R8 0.6%**					>3.095							>13.275				
CART MODEL IV	**R9 0.5%**			>13.165	>7.917	>3.02								<6.533	<1425.982		
	R10 0.1%			<13.165	>7.917	>3.02								<6.533	<1425.982		
CART MODEL V	**R11 0.3%**											>49.181			<1426.955		

Table 24: Summary of C5.0 Models (suggested regions)

		x2	x8	x9	x12	x19	x22	x27	x28	x29	x30	x32	x35	Support	Confidence
C5.0 Model I	**R1: for all**	>23					<14.35				<1.88			8/968	1
	R2: for all						>14.35					<0.184		17/968	1
	R3: for all				>350		>14.35					>0.184		15/968	1
	R4: for y2			<3.206	<350	>15.95	>14.35				>1.895	>0.184		257/968	0.681
	R10: for y3			<3.26			<14.35		<15.79	<3.355	>1.88		>0.08	407/968	0.688
	R13: for y6						<14.35		>15.79	<3.355	>1.88			25/968	0.68
	R14: for y8						<14.35			>3.35	>1.88			19/968	0.579
C5.0 Model II	**R1: for both**		<35				<14.35							17/586	1
	R2: for both						>14.35	<4.2					>0.088	16/586	1
	R3: for both	>23	>35				<14.35				<1.88			5/586	1
	R4: for y2			<3.216	<305	<15.95	>14.35	>4.2						198/586	0.828
	R6: for y3		>35				<14.35				<1.88			268/586	1
	R8: for y3			<3.216			>14.35	>4.2						34/586	0.765

Table 25: Factor Levels Determined via Extracted Rules

Var.	Cont.	Setting	Observed interval of means	Intersection of intervals suggested by rules	Related defects	Suggested levels for means
x2	N	[15, 30]	[20, 28]	[23, 28]	(y2),(y3),(y6),(y8)	if possible [23, 28]
x3	N	[15, 30]	[30, 40]	[31, 37.5]	y1,y3	if possible [31, 37.5]
x4	Y	[13, 15]	[12.171, 13.678]	[12.295, 13.678]	y1	constant [12.295, 13.678]
x5	Y	[14, 16]	[12.27, 13.66]	[12.27, 13.165]	y8	constant [12.27, 13.165]
x6	Y	[7.5, 9.5]	[7.585, 8.25]	[7.917, 8.25]	y8	constant [7.917, 8.25]
x8	Y	[35, 42]	[21.75, 42]	[21.75, 35]	y3, (y2)	constant [21.75, 35]
x9	Y	[3, 3.5]	[2.98, 3.387]	not exist	y2, y3, y6, y8	3 levels [3.183, 3.216], [3.216, 3.26], [3.26, 3.387]
x11	Y	[18, 23]	[19.8, 22.9]	[20.339, 22.9]	y3	constant [20.339, 22.9]
x12	Y	[250, 400]	[290, 360]	[350, 360]	y2	constant [350, 360], if it is too narrow, constant [305, 360]
x14	Y	[3.5, 5.5]	[4.7, 5.2]	[4.724, 5.2]	y2	constant [4.724, 5.2]
x16	N	[11, 23]	[13.2, 30]	[15.86, 30]	y1, (y2)	if possible [15.86, 30]
x17	N	[11, 23]	[15.9, 31.5]	[26.55, 31.5]	y1	if possible [26.55, 31.5]
x19	N	[11, 23]	[14.1, 24.9]	not exist	y2	leave as before
x20	Y	40	[38.992, 42.85]	[38.992, 41.32]	y3	constant [38.992, 41.32]
x21	Y	50	[48.68, 52.71]	[49.181, 52.71]	y9	constant [49.181, 52.71]
x22	Y	until 28th of March:12 after 28th of March: 22	until 28th of March: [10.85, 14,35], after 28th of March: [20.05, 33.428]	not exist	y1,y2,y3,y6	4 levels [10.85, 13.125], [12.275, 14.35], [14.35, 17.2], [17.2, 33.42]
x25	N	NA	[2.5, 6.9]	[2.5, 6.533]	y8	if possible [2.5, 6.533]
x26	Y	[1420, 1430]	[1367.59, 1428.23]	[1367.59, 1425.98]	y8, y9	constant [1367.59, 1425.98]
x27	N	NA	[2.259, 4.95]	[2.259, 4.2]	y2, (y3)	if possible [2.259, 4.2]
x28	N	NA	[11.7, 16.9]	not exist	y3, y6	leave as before
x29	Y	[3.2, 3.35]	[3.208, 3.41]	not exist	y1,y3,y6, y8	3 levels [3.208, 3.304], [3.304, 3.325], [3.355, 3.41]
x30	Y	[1.85, 2]	[1.823, 2]	not exist	y1,y2,y3	2 levels [1.823, 1.88], [1.88, 2]
x32	Y	[0.2, 0.3]	[0.171, 0.283]	not exist	y1,y2	2 levels [0.171, 0.184], [0.184, 0.283]
x33	Y	maximum 0.3	[0.0767, 0.552]	[0.174, 0.552]	y2	constant [0.174, 0.552]
x35	Y	[0.08, .12]	[0.0762, 0.1122]	[0.088, 0.1122]	y1	constant [0.088, 0.1122]

In the Table 25, the first column contains variables selected by the decision tree models, the second indicates whether the variable is controllable or not (Y: Yes, N: No), the third is the points or intervals set by the company, the fourth is the observed interval of the means and the fifth shows the defect types related with the variable (related defect types in parentheses are suggested by the weak rules).

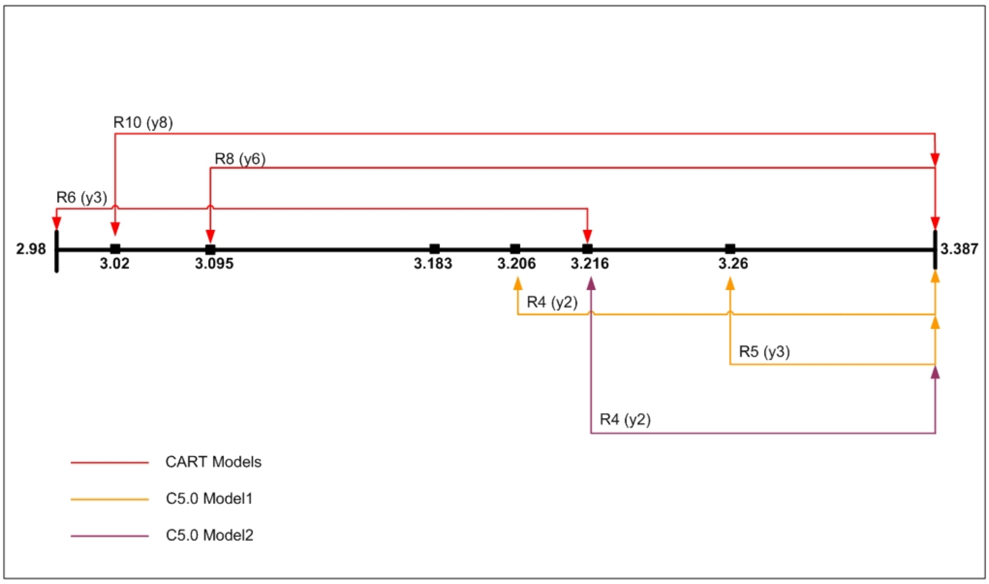


Figure 25: Regions suggested by the decision tree models for the design variable x9

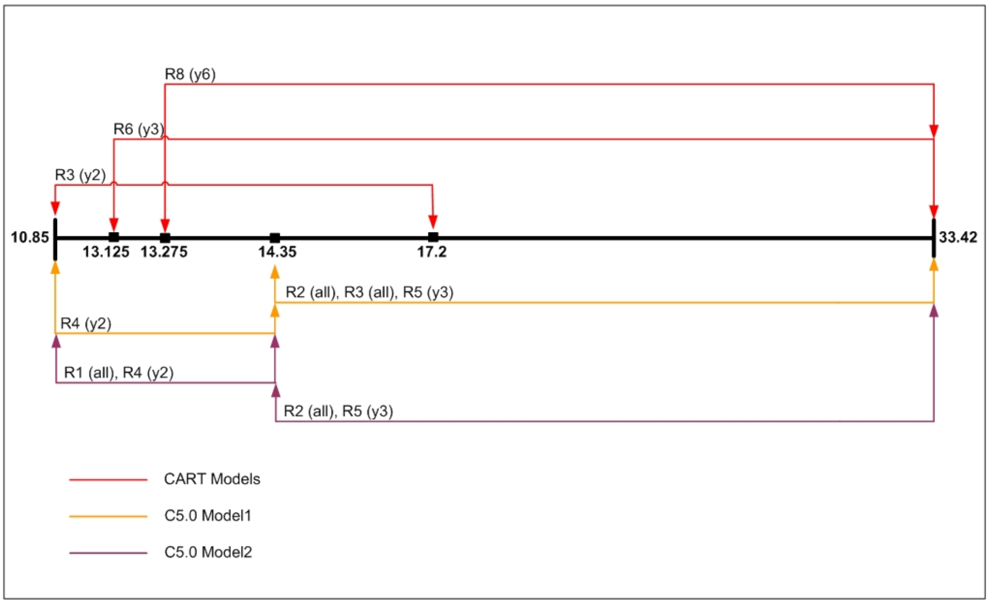


Figure 26: Regions suggested by the decision tree models for the design variable x22

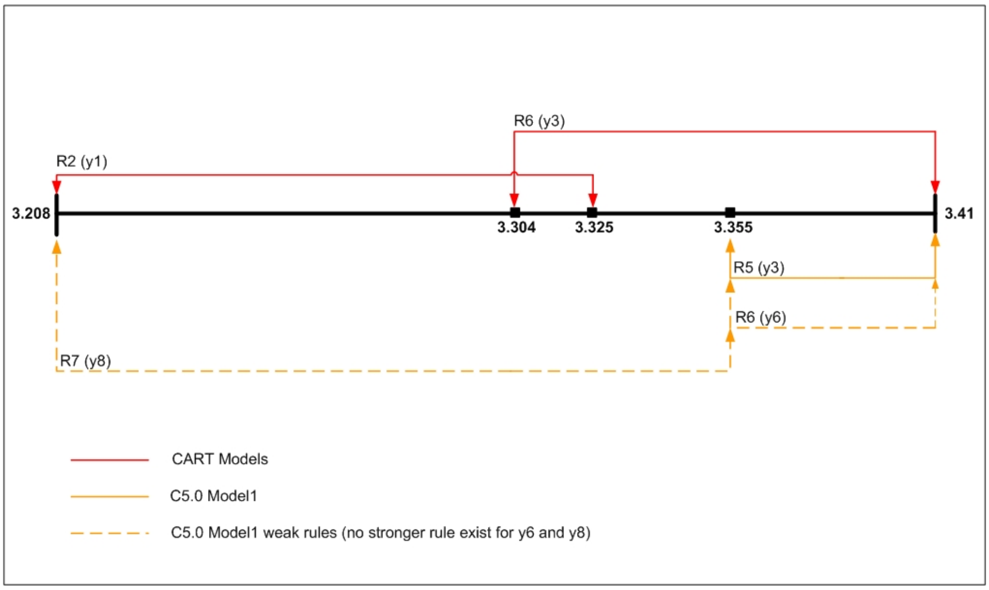


Figure 27: Regions suggested by the decision tree models for the design variable x29

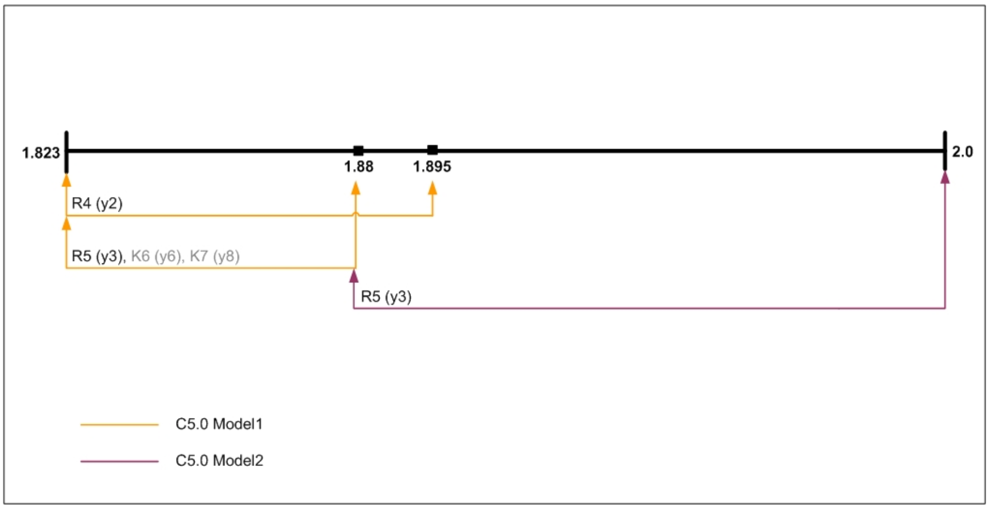


Figure 28: Regions suggested by the decision tree models for the design variable x30

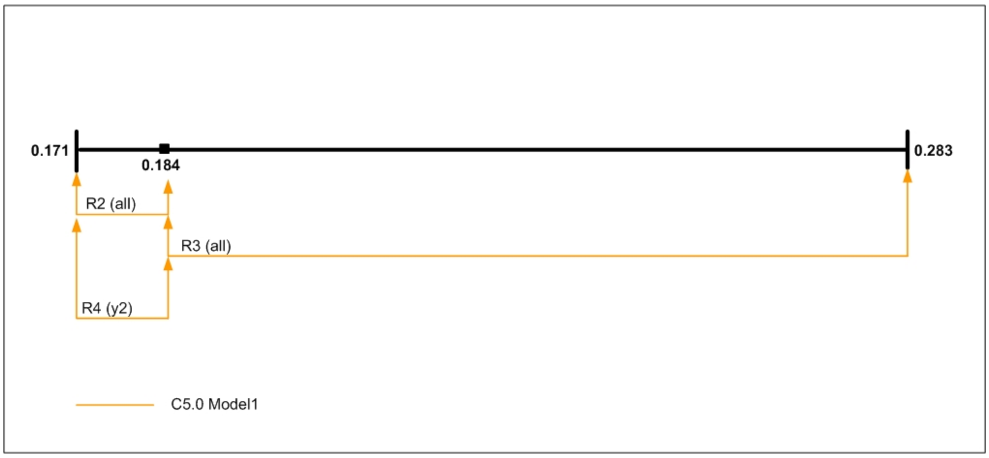


Figure 29: Regions suggested by the decision tree models for the design variable x32

CHAPTER 6

DISCUSSION AND CONCLUSION

Manufacturing processes in which lots of factors are involved have complex structures. A challenging issue in manufacturing environment is the identification of the influential process variables that cause defects or defective products. End product characteristics such as quality and process characteristics such as yield may be highly influenced by contributions due to the levels of controllable and uncontrollable variables at subsequent stages of manufacturing processes. Better understanding of these influential factors and development of models for the quality of these manufacturing processes are important issues for manufacturing decision-making.

In this study, logistic regression and decision tree approaches were used to model relationships between process and quality variables to identify important process variables and their respective values that cause defects on the products. Logistic regression is a common approach used in quality problems. Decision tree approach is recently used in manufacturing field. It is more popular among the other data mining techniques since it has some desirable properties such as simplicity, efficiency and interpretability that attract people in this field. Logistic regression and C5.0, one of the popular decision tree algorithms, were used for classification of the quality. Another popular decision tree algorithm, CART, was also used for predicting quality.

Four models were developed by logistic regression method. During the development of the models, numeric problems were encountered. Suggested solutions for convergence problems in (Allison, 1999) were applied to the models. Unfortunately, none of the final models was found to be significant although R-Square statistics and overall classification accuracies of Logit Model III and Logit Model IV are high and model parameters are found to be statistically significant. In addition, logistic regression model was biased towards the major classes when we examined classification accuracy for individual categories. It is especially the case in Logit Model I and Logit Model II (See Table 16, 17, 18 and 19).

On the contrary of Logistic Regression models, the decision tree approach has provided us with satisfactory results. This is likely to be caused by the partitioning facilitated by the tree construction. Classification accuracies of the C5.0 Model I and Model II are slightly better than logistic regression results (see Table 21 and 22). Estimated accuracy for the C5.0 Model I and C5.0 Model II were found to be 60.3% and 92.15%, respectively. C5.0 Model II, which is developed for the most important two defect categories, is successful in terms of both high classification accuracy and extracting meaningful rules for the categories. In addition, CART Models that predict defective proportions of the batches are also found to be successful. They are successful on the test data as well. Each of CART models can be used individually if only one defect type is of interest.

In addition to prediction power of the decision tree models, interpretation of the results derived from the tree models is very simple. Decision tree models generate simple decision rules to predict or categorize the response variable. Unlike decision tree models, the use and interpretation of logistic regression models is rather complicated. Logistic models are interpreted in terms of odds-ratio, which is the probability of occurrences of a category of interest relative to the other category or categories. If the number of levels that the response variable has is high, interpretation of the model is more complicated since more comparisons are needed. Another issue is the determination of the meaningful unit changes for the continuous predictors. From the industry perspective, ease of the use of models is an important feature and therefore preferable.

At the end of the study, the results of the decision tree models were presented to the quality team of the company. The approach was found to be beneficial and simple. Some of the parameters and their respective thresholds in the models were judged to be meaningful, e.g., x22, whereas some others were found to be unexpected (interesting), e.g., x29. The threshold values of the parameters provided by the model were considered to be useful in optimization of the casting process. An experiment whose design variables and their respective levels shown in Table 25 were suggested to the company to let them investigate the impact of variables selected by the models on casting process. In this sense, the decision tree analysis can be considered as a way of planning for statistical design of experiments for optimization purposes. Before such experimentation, decision tree approach can also be used for both feature selection and factor levels determination.

Possible future work can be the improvement of prediction and classification accuracies of the models using alternative data mining algorithms, such as neural networks. Then, the finding can be compared with the decision tree models obtained in this study. Although neural networks are black-box models and their results can not be used as easily as results of the decision tree models, neural networks can be used to determine important process variables by performing a sensitivity analysis of the input fields after the network has been trained. Another possible future work can be focus on the preparation and preprocessing of manufacturing data having several problems to be handled since the quality of data strongly affects the results.

REFERENCES

Abajo, N. & Diez, A. B. (2004). ANN Quality Diagnostic Models for Packaging Manufacturing: An Industrial Data Mining Case Study, USA: International Conference on Knowledge Discovery and Data Mining, (pp. 799-804).

Agrawal, R., Imielinski, T. & Swami, A. (1993). Mining associations between set of items in massive databases. New York: *Proceedings of the 1993 ACM-SIGMOD International Conference on Managemnt of Data* (pp. 207-216)

Agrawal, R. & Srikant, R. (1994). Fast algorithms for mining association rules. San Francisco: *Proceedings of the 20th International Conference on Very Large Databases,* (pp. 487-499).

Allison, P. D. (1999). *Logistic Regression Using the SAS System: Theory and Application*, NC, USA: SAS Institute Inc.

Braha, D. &Shmilovici, A. (2002). Data Mining for Improving a Cleaning Process in the Semiconductor Industry, *IEEE Transactıons on Semiconductor Manufacturing, 15(1)*, 91-101.

Breiman, L., Friedman, J. H., Olshen, R. A., Stone, C. J. (1984). *Classification and regression trees*. Boca Raton, Fla. : CRC Pres.

Breiman, L., Friedman, J. H., Olshen, R. A., Stone, C. J. (1998). *Classification and regression trees*. Boca Raton, Fla. : Chapman & Hall

Brinksmeier, E., Tönshoff, H. K., Czenkusch, C. & Heinzel, C. (1998). Modeling and Optimization of Grinding Processes, *Journal of Intelligent Manufacturing, 9*, 303-314.

Chien, C. F., Wang, W. C. & Cheng, J. C. (2007). Data mining for yield enhancement in semiconductor manufacturing and an empirical study , *Expert Systems with Applications, 33(1)*, 192-198.

Clementine® 10.1 Algorithms Guide. (2006). USA: Integral Solutions Limited. http://www.spss.com/clementine/

Clementine® 10.1 Node Reference. (2006). USA: Integral Solutions Limited. http://www.spss.com/clementine/

Clementine® 10.1 User's Guide. (2006). USA: Integral Solutions Limited. http://www.spss.com/clementine/

Cser, L., Gulyas, J., Szücs, L., Horvath, A., Arvai, L. & Baross, B. (2001). *Different Kinds of Neural Networks in Control and Monitoring of Hot Rolling Mill*, Budapest , HONGRIE: Proceedings of the Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems , (pp.791-796).

Deng, B. & Liu, X. (2002). *Data Mining in Quality Improvement*, Orlando, Florida: Proceedings of the Twenty-first Annual SAS Users Group International Conference, (pp.111-127).

Dunham, M. (2003). *Data mining introductory and advanced topics*, New Jersea: Pearson Education, Inc.

Fan, C.M., Guo, R. S., Chen, A., Hsu, K. C. & Wei, C. S. (2001). *Data Mining and Fault Diagnosis based on Wafer Acceptance Test Data and In-line Manufacturing Data*, San Jose, CA: International Symposium on Semiconductor Manufacturing, (pp. 171-174).

Fayyad, U., Piatetsky-Shapiro, G., Smyth, P., & Uthurusamy, R. (1996). *Advances in Knowledge Discovery and Data Mining,* Cambridge: MIT.

Feng, C. X. & Wang, X. F. (2003). Surface Roughness Predictive Modeling: Neural Networks versus Regression, *IIE Transactions on Design and Manufacturing 35,* 11-27.

Forrest, D. R., (2003). *High Dimensional Data Mining in Complex Manufacturing Processes*. Unpublished doctoral dissertation, University of Virginia.

Gardner, M. & Bieker, J. (2000). *Data Mining Solves Tough Semiconductor Manufacturing Problems*, Boston, MA USA: Proceedings of the Conference on Knowledge Discovery and Data Mining, (pp. 376-383).

Han, J., & Kanber, M. (2001). *Data mining: concepts and techniques,* San Francisco: Morgan Kaufmann.

Harding, J. A., Shahbaz, M., Srinivas & Kusiak, A. (2006). Data Mining in Manufacturing: A Review, *Journal of Manufacturing Science and Engineering*, *128*, 969-976.

Haykin, S. (1994) *Neural Networks: A Comprehensive Foundation*, New York: Macmillan.

Ho, G. T. S., Lau, H. C. W., lee, C. K. M., Ip, A. W. H. & Pun, K. F. (2006). An Intelligent Production Workflow Mining System for Continual Quality Enhancement, *Intelligent Journal of Advanced Manufacturing Technology, 28*, 792–809.

Hosmer, D. W. & Lemeshow, S. (2000). *Applied Logistic Regression*, New York: Wiley.

Hou, T., Liu, W. & Lin, L. (2003). Intelligent Remote Monitoring and Diagnosis of Manufacturing Processes Using an Integrated Approach of

Neural Networks and Rough Sets, *Journal of Intelligent Manufacturing, 14(2),* 239-253.

Hou, T. H. & Huang, C. C. (2004). Application of Fuzzy Logic and Variable Precision Rough Set Approach in a Remote Monitoring Manufacturing Process for Diagnosis Rule Induction, *Journal of Intelligent Manufacturing, 15(3)*, 395-408.

Huang, H. & Wu, W. (2005). *Product Quality Improvement Analysis Using Data Mining: A Case Study in Ultra-Precision Manufacturing Industry*, China: Conference on Fuzzy Systems and Knowledge Discovery (pp.577-580).

Jemwa, G. T. & Aldrich, C. (2005). Improving Process Operations Using Support Vector Machines and Decision Trees, *American Institute of Chemical Engineers, 51(2)*, 526–543.

Kohonen, T. (1982) Self-organized formation of topologically correct feature maps. *Biological Cybernetics*, 43, 59-69.

Kohonen, T. (2001) *Self-organizing maps*. Berlin ; New York : Springer.

Krimpenis, A., Benardos, P. G., Vosniakos, G. C. & Koukouvitaki, A. (2006). Simulation-Based Selection of Optimum Pressure Die-Casting Process Parameters Using Neural Nets and Genetic Algorithms, *Intelligent Journal of Advanced Manufacturing Technology, 27*, 509–517.

Kusiak, A. & Kurasek, C. (2001). Data Mining of Printed-Circuit Board Defects, *IEEE Transactions on Robotics and Automation, 17(2)*, 191-196.

Li, M., Feng, S., Sethi, I. K., Luciow, J. & Wagner, K. (2003). *Mining Production Data with Neural Network & CART*, Melbourne, Florida, USA: Proceedings of the Third IEEE International Conference on Data Mining (pp.731-734).

Lian, J., Lai, X. M., Lin, Z. Q. & Yao, F. S. (2002). Application of Data Mining and Process Knowledge Discovery in Sheet Metal Assembly Dimensional Variation Diagnosis, *Journal of Materials Processing Technology, 129(1)*, 315-320.

McCullagh, P. (1980). Regression models for ordinal data (with discussion), *Journal of the Royal Statistical Society, Series B, 42*, 109-127.

Mieno, F., Sato, T., Shibuya, Y., Odagiri, K., Tsuda, H. & Take, R. (1999). *Yield Improvement Using Data Mining System*, USA: Conference on Semiconductor Manufacturing, (pp.391-394).

Montgomery, D. C. & Peck, E. A. (1982). *Introduction to Linear Regression Analysis,* New York: Wiley.

Quinlan, J. R. (1986). Induction of Decision Trees. *Machine Learning,* 1, 81-106.

Russell, S. J., Norvig, P. (2003). *Artificial intelligence : a modern approach*. N.J. : Prentice Hall.

Shi, D., & Tsung, F. (2003). Modelling and diagnosis of feedback-controlled processes using dynamic PCA and neural networks, *International Journal of Production Research, 41(2)* 365–379.

Shi, X., Schillings, P. & Boyd, D. (2004). Applying artificial neural networks and virtual experimental design to quality improvement of two industrial processes, *International Journal of Production Research, 42(1),* 101–118.

Skinner, K. R., Montgomery, D. C., Runger, G. C., Fowler, J. W., McCarville, D. R., Rhoads, T. R., et al. (2002). Multivariate Statistical Methods for Modeling and Analysis of Wafer Probe Test Data, *IEEE Transactions on Semiconductor Manufacturıng, 15(4),* 523-530.

Wang, R. J., Wang, L., Zhao, L. & Liu, Z. (2006). Influence of Process Parameters on Part Shrinkage in SLS, *Intelligent Journal of Advanced Manufacturing Technology.*

Weiss, S. M., Indurkhya, N. (1998). *Predictive data mining : a practical guide.* San Francisco : Morgan Kaufmann Publishers.

Werbos, P. J. (1974). *Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences*. Unpublished doctoral dissertation, Harvard University.

Ye, N. (2003) *The Handbook of Data Mining.* Mahwah, N.J. : Lawrence Erlbaum Associates.

Zhou, Q., Xiong, Z., Zhang, J. & Xu, Y. (2006). Hierarchical Neural Network Based Product Quality Prediction of Industrial Ethylene Pyrolysis Process, *Lecture Notes in Computer Science, 3973*, 1132-1137.

APPENDICES

APPENDIX A - DECISION TREE GRAPHS OF CART MODELS

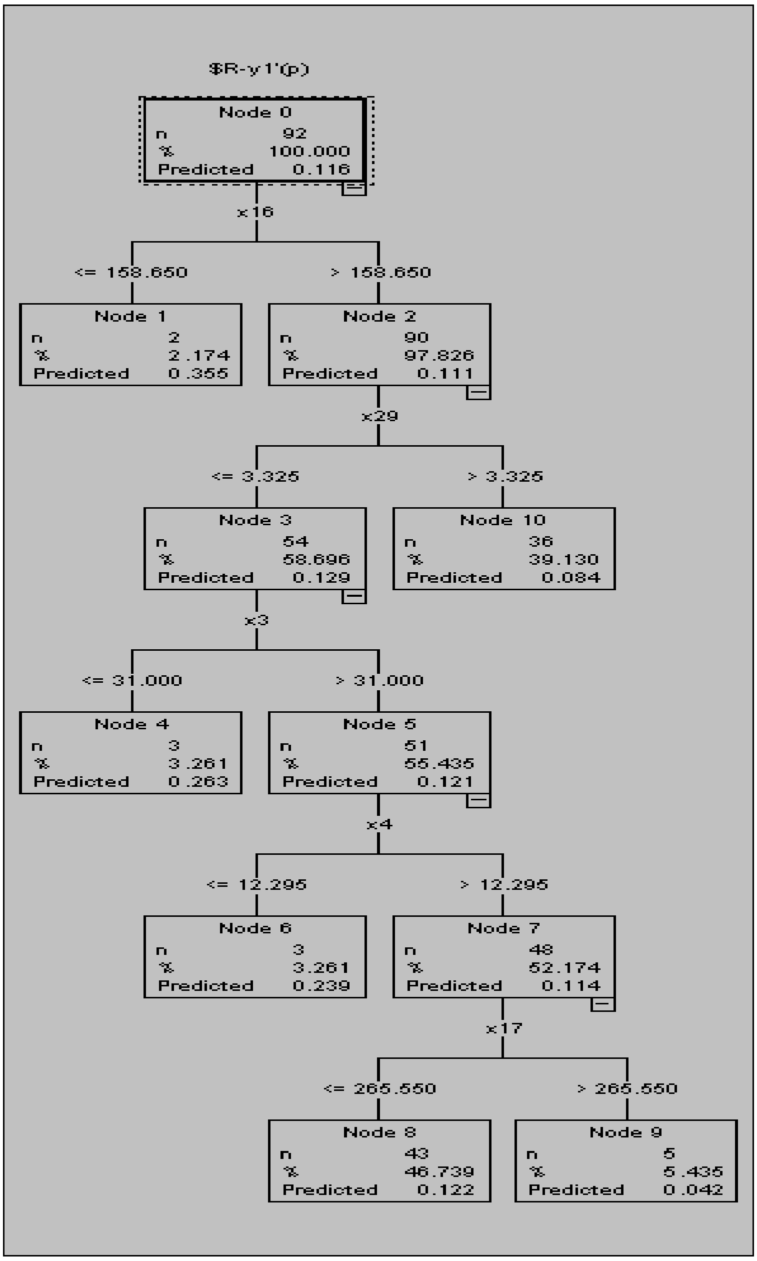


Figure 30: Tree of the CART Model 0

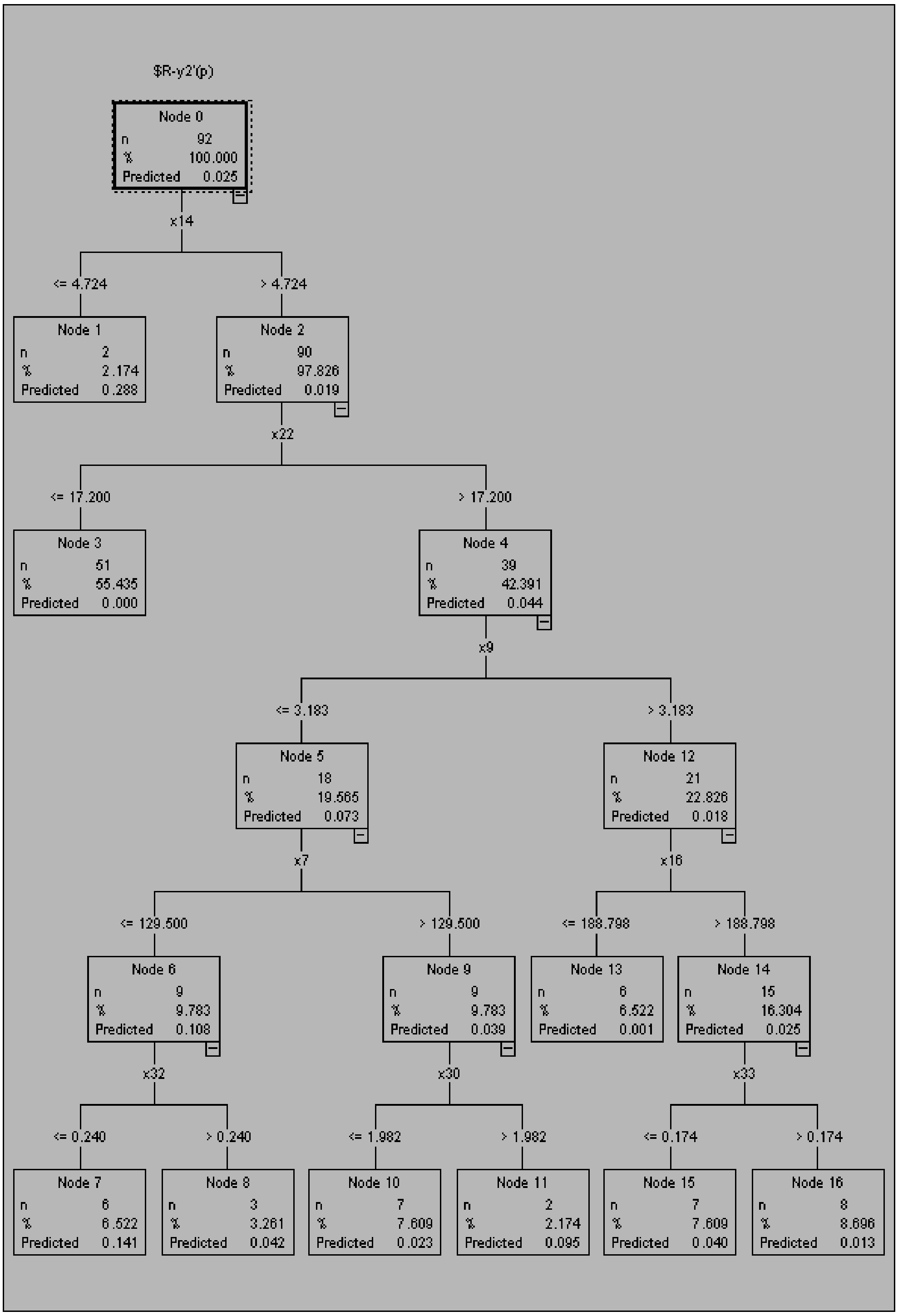


Figure 31: Tree of the CART Model I

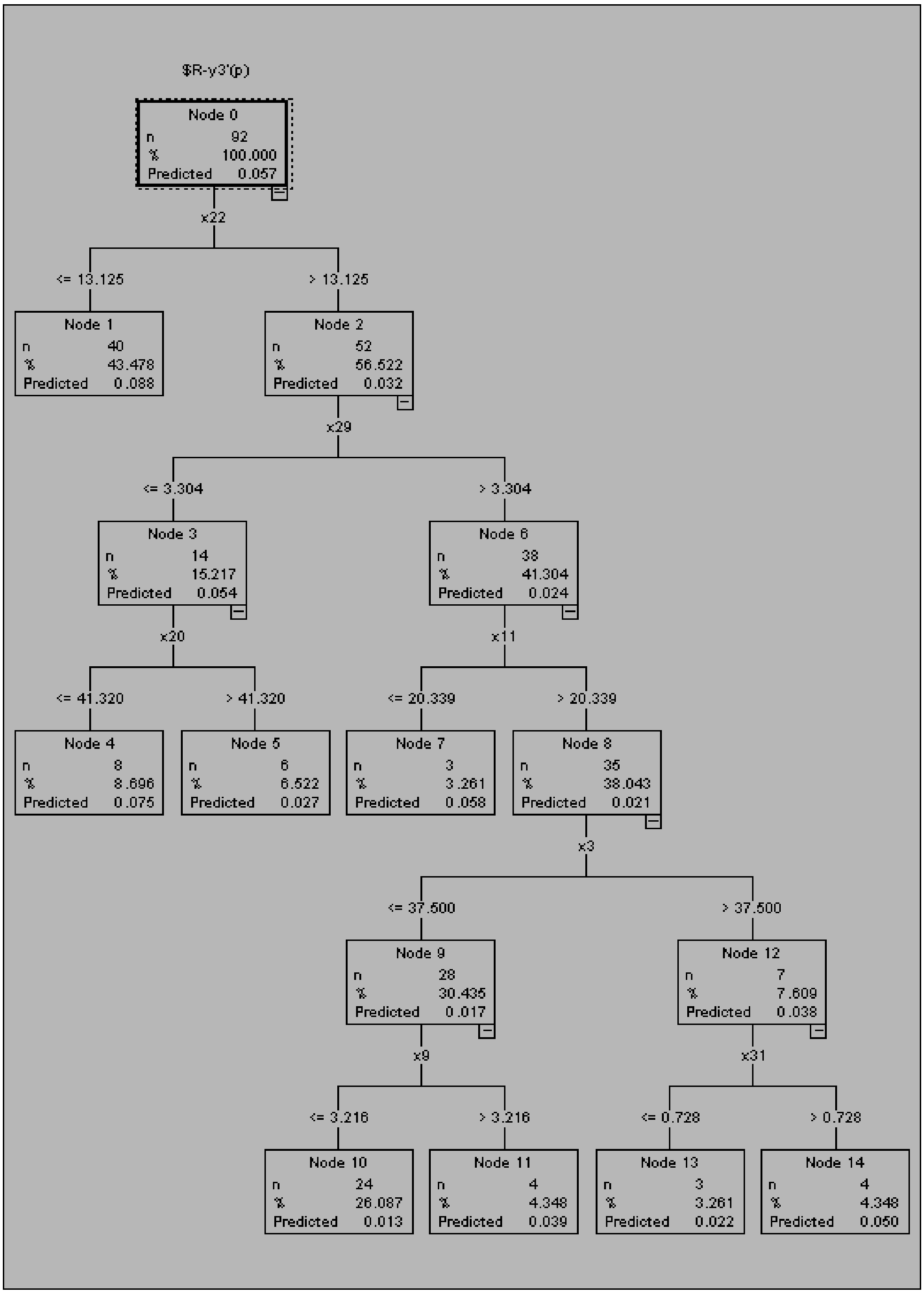


Figure 32: Tree of the CART Model II

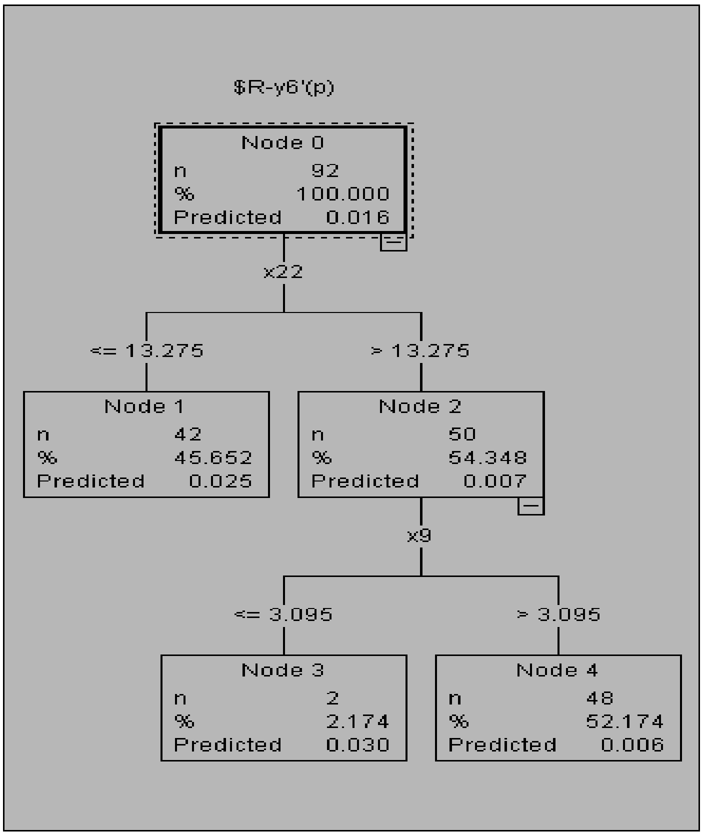


Figure 33: Tree of the CART Model III

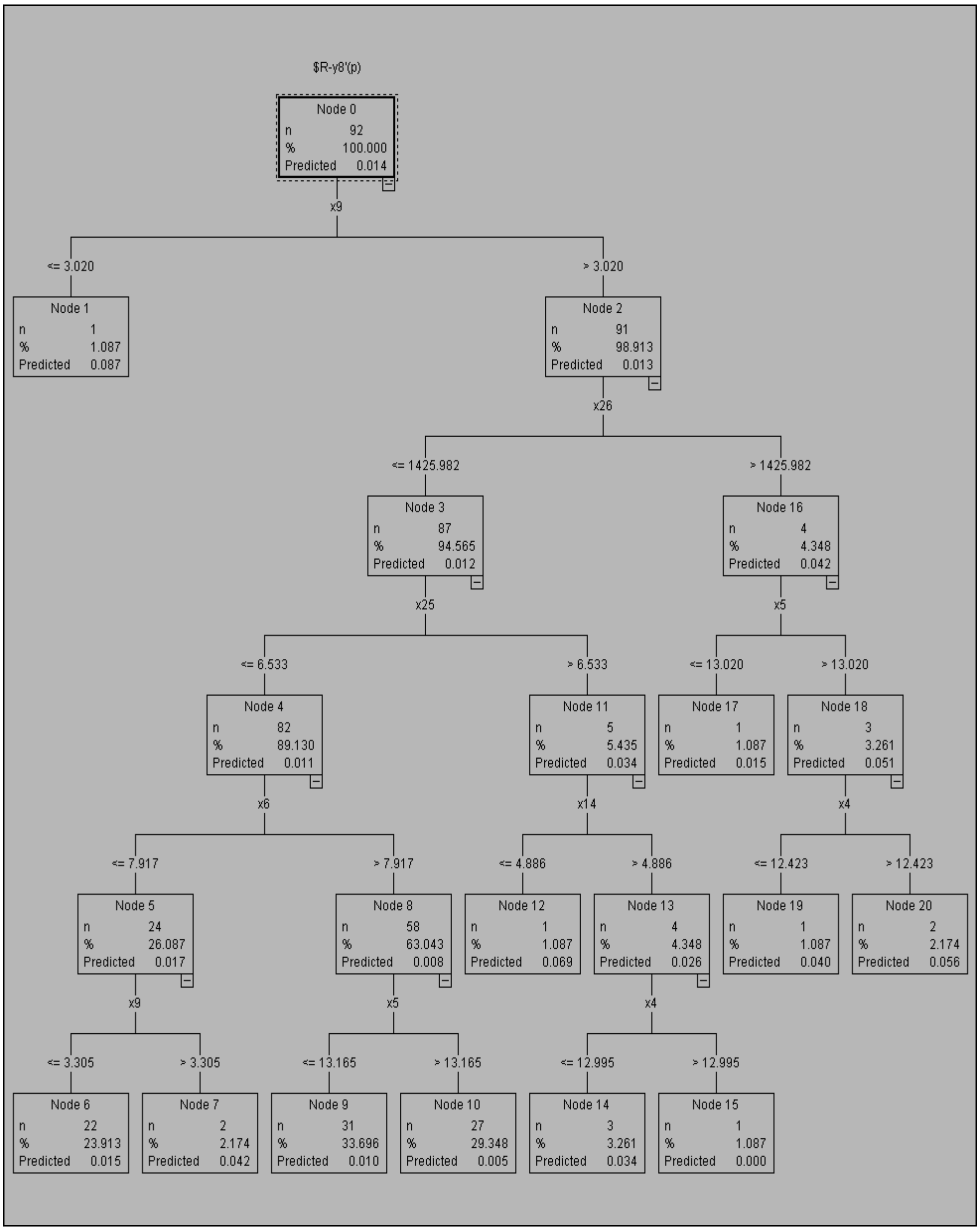


Figure 34: Tree of the CART Model IV

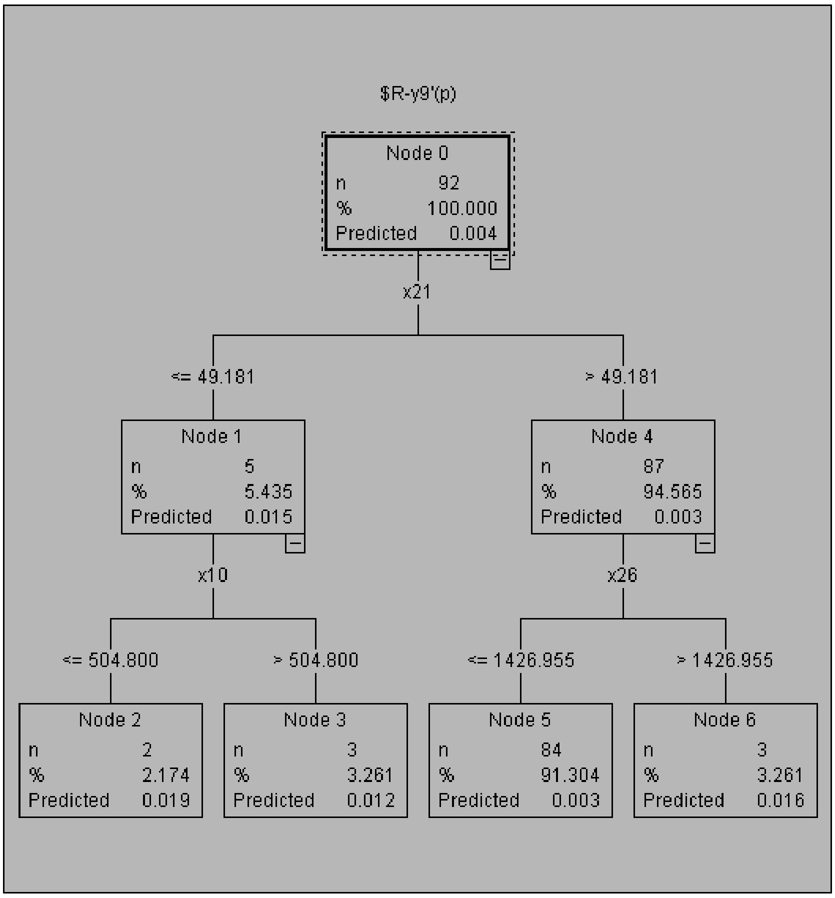


Figure 35: Tree of the CART Model V

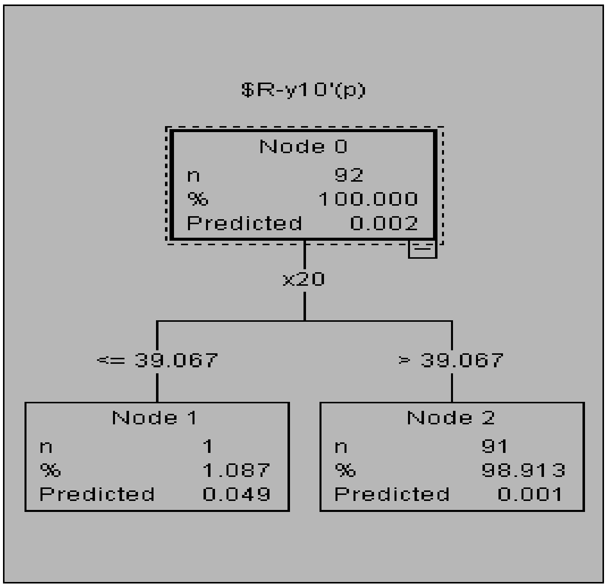


Figure 36: Tree of the CART Model VI

APPENDIX B- DECISION TREE GRAPHS OF C5.0 MODELS

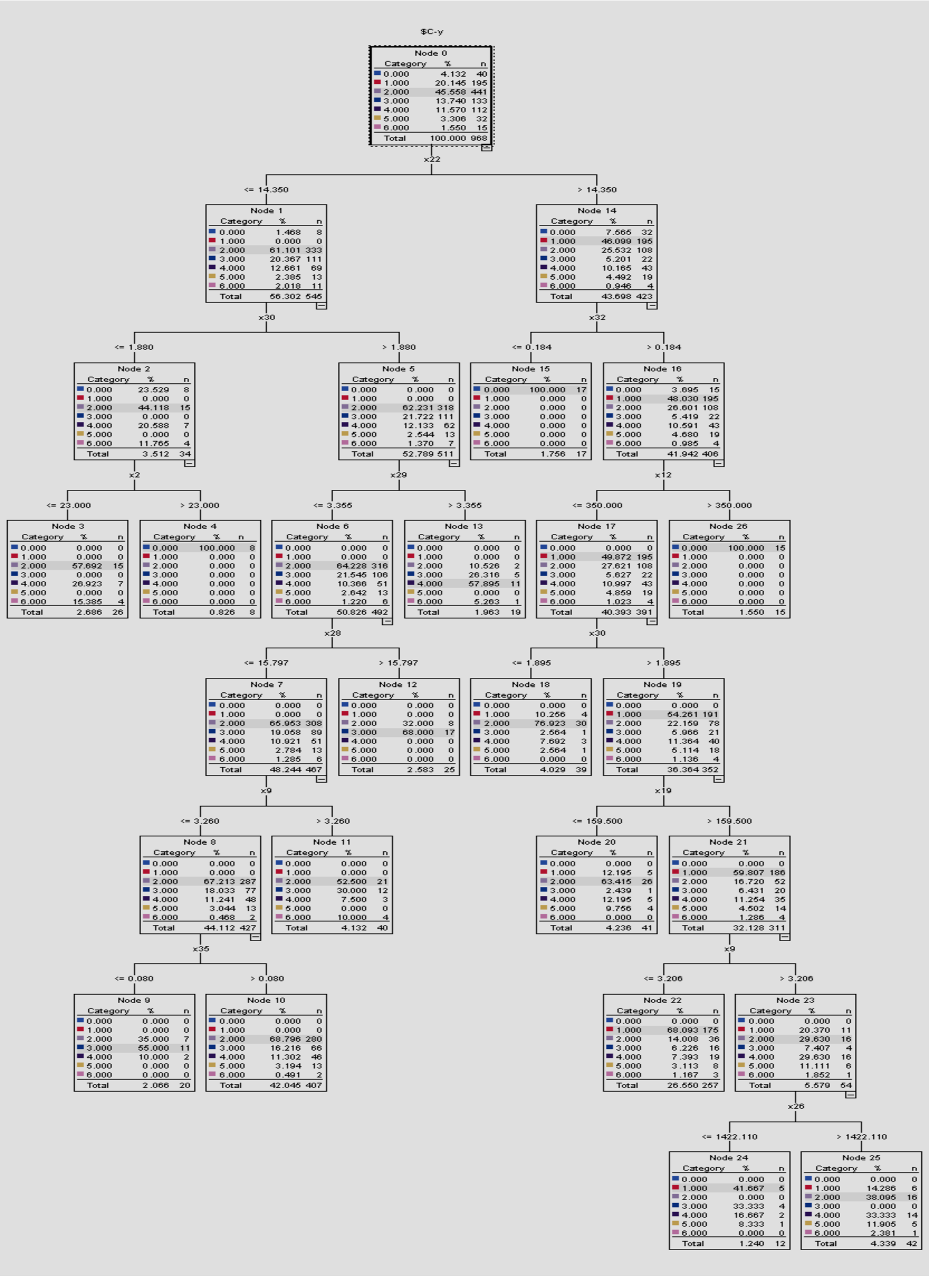


Figure 37: Tree of the C5.0 Model I

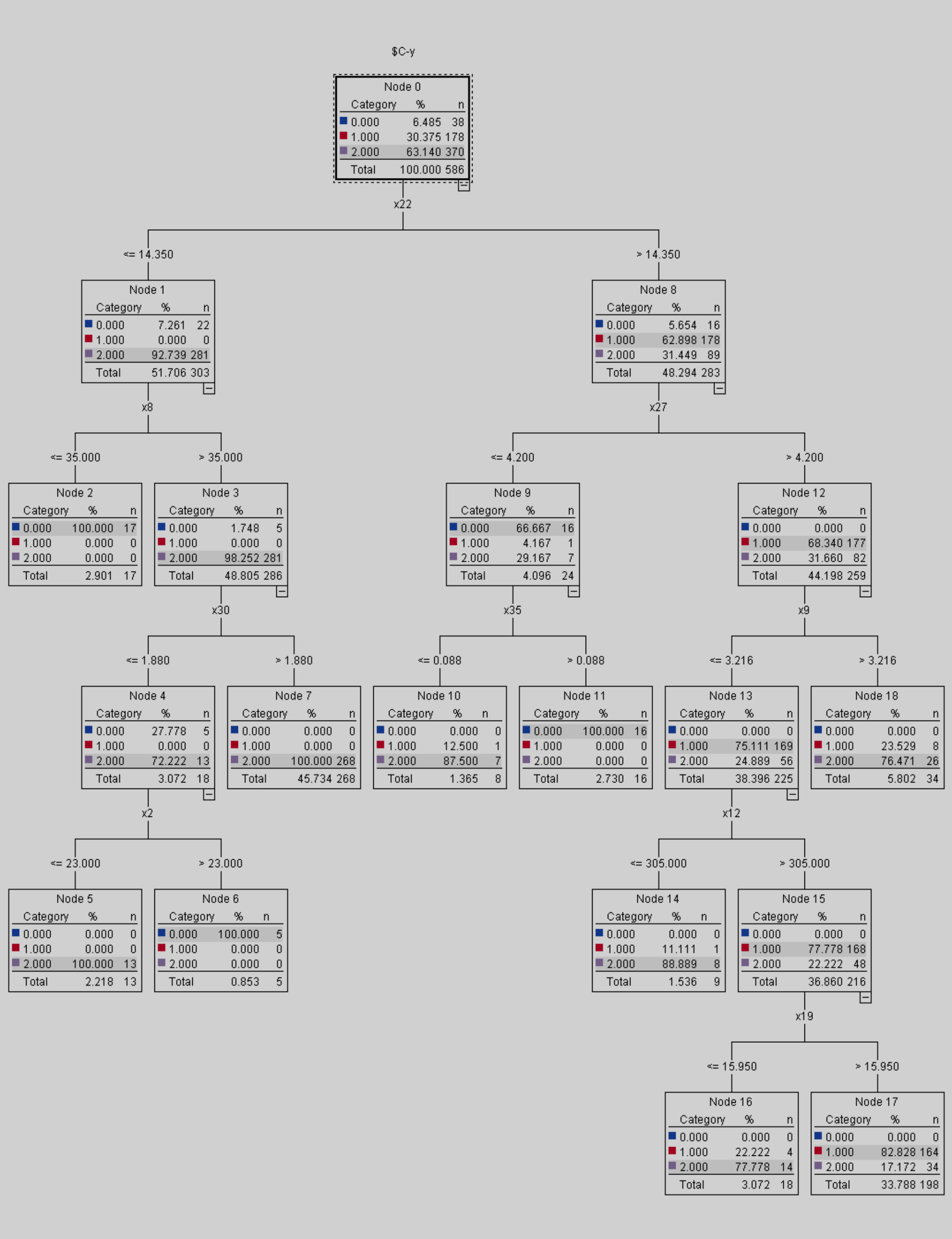


Figure 38: Tree of the C5.0 Model II

APPENDIX C - ANOMALY DETECTION ALGORITHM

Information in this section was gathered from Clementine® 10.1 Algorithm Guide (2006) and Clementine® 10.1 Node Reference (2006).

Anomaly detection algorithm is used to determine outliers. Extraction of these unusual cases is performed based on deviation from the norms of their clusters. It is an unsupervised method. Unlike the traditional methods used to detect outliers, anomaly detection algorithm can examine large number of variables together.

Algorithm has three steps, which are modeling, scoring and reasoning. In modeling step, the variables are used to form clusters. Clusters are determined via two-step clustering algorithm which is consists of a pre-clustering step that divides data into many sub-clusters and a cluster step that combines sub-clusters to decrease initial number of clusters to the desired number of clusters. The algorithm can select number of clusters automatically. In scoring step, cases are assigned to closest cluster. After that, variable deviation indices (VDI) defined as contribution of a variable to its log-likelihood distance, group deviation index (GDI) of the cases, which is sum of all the VDI, anomaly index of cases calculated as ratio of the case's GDI to the average GDI of the cluster to which the case belongs and variable contribution measures defined as ratio of the variable's VDI to the case's GDI. Finally, in reasoning step, most anomalous cases are identified by using anomaly index.

APPENDIX D - GAIN CHARTS

Information in this section was gathered from Clementine® 10.1 Node Reference (2006).

A gain chart is a visual evaluation tool which shows the performance of a specified model on predicting particular outcomes. Gains are defined as the proportion of total hits that occurs in each quantile. It can be computed as

Gain = (number of hits in quantile / total number of hits) × 100%.

Followings are the steps for how a gain chart works:

- Records are sorted based on the predicted value and confidence of the prediction
- Record are splitted into quantiles
- A business rule or hit, a specific value or range of values, is defined
- Value of business criterion for each quantile are plotted from highest to lowest

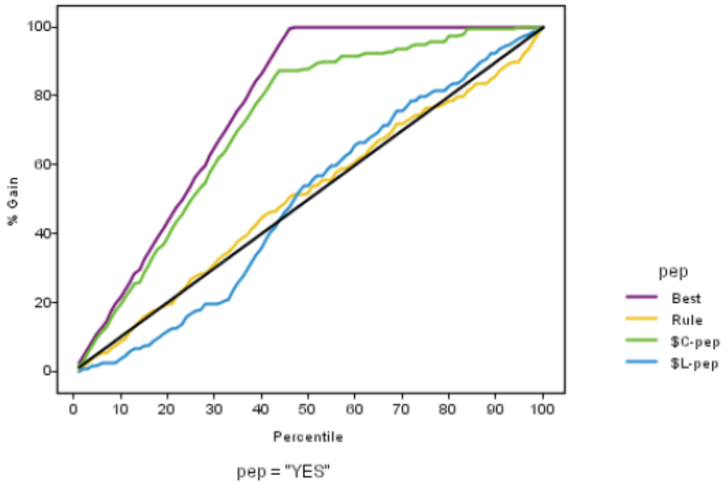


Figure 39: Gains chart (cumulative) with baseline, best line and business rule displayed

Base line, which is the diagonal line, in Figure 39 indicates a perfectly random distribution of hits where confidence becomes irrelevant. If a model provides no information it follows the diagonal. The best line on the other hand, denotes perfect confidence where hits are 100% of cases. A good model is expected to be close to the best line. It rises steeply toward 100% and then level off if the chart is cumulative.

Two or more models can be viewed in a single chart to compare their prediction accuracy. For cumulative charts, higher lines denote better models.