A METHOD FOR PRODUCT DEFECTIVENESS PREDICTION WITH PROCESS ENACTMENT DATA IN A SMALL SOFTWARE ORGANIZATION

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

BY

DAMLA SİVRİOĞLU

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

JUNE 2012

A METHOD FOR PRODUCT DEFECTIVENESS PREDICTION WITH PROCESS ENACTMENT DATA IN A SMALL SOFTWARE ORGANIZATION

Submitted by **DAMLA SİVRİOĞLU** in partial fulfillment of the requirements for the degree of **Master of Science in Information Systems**, **Middle East Technical University** by,

Prof. Dr. Nazife Baykal ____________________
Director, **Informatics Institute**

Prof. Dr. Yasemin Yardımcı Çetin ____________________
Head of Department, **Information Systems**

Prof. Dr. Onur DEMİRÖRS ____________________
Supervisor, **Information Systems,METU**

Dr. Ayça TARHAN ____________________
Co-Supervisor, **Computer Engineering, Hacettepe University**

Examining Committee Members:

Prof. Dr. Semih BİLGEN ____________________
Electrical and Electronics Engineering, METU

Prof. Dr. Onur DEMİRÖRS ____________________
Information Systems, METU

Dr. Ayça TARHAN ____________________
Computer Engineering, Hacettepe University

Assist. Prof. Dr. Aysu BETİN CAN ____________________
Information Systems, METU

Assoc. Prof. Dr. Altan KOÇYİĞİT ____________________
Information Systems, METU

Date: 28.06.2012

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 : Damla Sivrioğlu

Signature :

ABSTRACT

A METHOD FOR PRODUCT DEFECTIVENESS PREDICTION BY USING PROCESS ENACTMENTT DATA IN A SMALL SOFTWARE ORGANIZATION

Sivrioğlu, Damla
M.Sc., Department of Information Systems
Supervisor: Prof. Dr. Onur Demirörs
Co-Supervisor: Dr. Ayça Tarhan

June 2012, 170 pages

As a part of the quality management, product defectiveness prediction is vital for small software organizations as for instutional ones. Although for defect prediction there have been conducted a lot of studies, process enactment data cannot be used because of the difficulty of collection. Additionally, there is no proposed approach known in general for the analysis of process enactment data in software engineering.

In this study, we developed a method to show the applicability of process enactment data for defect prediction and answered "Is process enactment data beneficial for defect prediction?", "How can we use process enactment data?" and "Which approaches and analysis methods can our method support?" questions. We used multiple case study design and conducted case studies including with and without process enactment data in a small software development company. We preferred machine learning approaches rather than statistical ones, in order to cluster the data which includes process enactment informationsince we believed that they are convenient with the pattern oriented nature of the data.

By the case studies performed, we obtained promising results. We evaluated performance values of prediction models to demonstrate the advantage of using process enactment data for the prediction of defect open duration value. When we have enough data points to apply machine learning methods and the data can be clusteredhomogeneously, we observed approximately 3% (ranging from -10% to %17) more accurate results from analyses including with process enactment data than the without ones.

Keywords: software defect prediction, machine learning, software measurement, defectiveness, software process enactment.

ÖZ

KÜÇÜK BİR KURUMDA ÜRÜN HATALILIK TAHMİNİ İÇİN SÜREÇ İŞLETME VERİSİNİN KULLANILDIĞI BİR METOT

Sivrioğlu, Damla
Yüksek Lisans, Bilişim Sistemleri Bölümü
Tez Yöneticisi: Prof. Dr. Onur Demirörs
Ortak Tez Yöneticisi: Dr. Ayça Tarhan

Haziran 2012, 170 sayfa

Kalite yönetiminin bir parçası olarak ürün hatalılığı tahmini kurumsal şirketlerde olduğu kadar küçük yazılım kurumları için de hayati önem taşır. Hata tahmini ile ilgili pek çok çalışma yürütülmüş olmasına rağmen süreç işletme verisi, toplama zorluğu nedeniyle kullanılamamaktadır. Buna ek olarak süreç işletme verisinin yazılım mühendisliğinde analizi için önerilen ve genel olarak bilinen herhangi bir yaklaşım yoktur.

Biz bu çalışmada, süreç işletme verisinin hata tahmini için uygulanabilirliğini gösteren bir metot geliştirdik ve "Süreç işletme verisinin kullanımı hata tahmini için yararlı mıdır?", "Süreç işletme verisini nasıl kullanabiliriz?" ve "Bizim geliştirdiğimiz metot hangi analiz metotlarını destekleyebilir?" sorularını cevapladık. Çoklu durum çalışması tasarımını kullandık ve küçük bir yazılım şirketinde süreç işletme verisinin kullanıldığı ve kullanılmadığı durumlar dahil olmak üzere dört durum çalışması için analizler gerçekleştirdik. Süreç işletme bilgisini içeren verinin gruplaması için istatistiksel yaklaşımlar yerine makine öğrenmesi yaklaşımlarını tercih ettik. Çünkü örüntü tanıma amaçlı olan makine öğrenmesi yöntemlerinin, örüntüye yönelik doğası gereği süreç işletme verisi için elverişli olduğunu değerlendirdik.

Yaptığımız durum çalışmaları ile ümit verici sonuçlar elde ettik. Hata açık kalma süresi değerinin tahmini için süreç işletme verisinin kullanımının avantajını göstermek için tahmin modellerinin performanslarını değerlendirdik. Makine öğrenmesi metotlarını uygulamak için yeterli veri noktamız olduğunda ve veri homojen olarak gruplanabildiğinde, süreç işletme verisinin dahil edildiği analiz sonuçlarının, dahil edilmemiş olanlara göre yaklaşık 3% (-10% ile 17% aralığında) daha doğru olduğunu gözlemledik.

Anahtar Kelimeler: yazılım hata tahmini, makine öğrenmesi, yazılım ölçümü, hatalılık, yazılım süreç işletme.

To my family, my friends and my life coach…

ACKNOWLEDGEMENTS

Special thanks should be given to Dr. Ayça Tarhan, my co-supervisor for her professional guidance, dedication and valuable support and to Prof. Onur Demirörs for his useful and constructive recommendations on this study.

I would like to express my very great appreciation to Assoc. Prof. Veysi İşler for his support in providing environment for scientific analyses. I wish to thank various people for their contribution to this study; Şafak Burak Çevikbaş, Dr. Aydın Okutanoğlu and Ahmet Birdal for their valuable technical support on this study; Gökhan Yılmaz, Emine Akman, Caner Kara, Behlül Kuran, Nurbanu Kından and Ayşegül Ünal, staff of Simsoft company, for their help in collecting the raw data. Their all assistance encouraged me to complete this study.

I send my deep thanks to my family and Muhammet Eren Aslan for their trust, patience and support.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

ACC: Accuracy

AUC: Area Under the ROC Curve

BBN: Bayesian Belief Networks

CA: Correlation Model

CMMI: Capability Maturity Model Integrated

DA: Discriminant Analysis

FPR: False Positive Rate

GM: Gokhale and Mullen

GQM: Goal-Question-Metric

ISO/IEC:International Organisation for Standardization and International Electrotechnical Commission

kNN: k Nearest Neighbor

KLOC: Kilo Lines of Code

MAE: Mean Absolute Error

MLP: Multilayer Perceptron

MMRE: Mean Magnitude of Relative Error

MUA: Metric Usability Attribute

MUF: Metric Usability Factor

MUQ: Metric Usability Questionnaire

ODC: Orthogonal Defect Classification

PAP: Process Attribute Pattern

PCA: Principal Component Analysis

PER: Process Execution Record

PREC: Precision

PSM: Process Similarity Matrix

RMSE: Root Mean Square Error

ROC: Receiver Operating Characteristic

RRSE: Root Relative Square Error

S: Schneidewind

SCU: Software Configuration Unit

SLOC: Source Lines of Code

SVM: Support Vector Machines

TPR: True Positive Rate

CHAPTER 1

INTRODUCTION

As stated in Weinberg's definition "Software quality is conformance to customer requirements.". Neverthless, software quality is a very crucial feature of a product to gain acceptance from the customer. In this viewpoint, software quality needs continuous monitoring and controlling through the software project. The defectiveness of software is an important quality measure to interpret the status of the product quality. Therefore, software defectiveness should be focus point of researches and quality models. For example, process reference models such as CMMI [1] proposes defect metrics for measurement and analysis activities to achieve multiple process areas.

In this context, we first performed a case study for searching for analysis techniques to understand product defectiveness and affecting factors in a small organization [2]. We applied various statistical and machine learning analysis methods to our product data. By doing this, we collected defect related and product related metrics in different data sets. At the end, we presented our inferences in three categories based on their confidence. According to our evaluation findings, the statistical analysis used for product data results could be considered as confident if supported by new studies. In addition, Apriori machine learning analysis used for defect data results could also be considered as confident, since we observed 90% "correctly classified instances" value in Weka tool. In contrary to this, C4.5 decision tree and logistic regression machine learning analyses used for defect data results had approximately 50% "incorrectly classified instances" value.

We have argued as one of the reasons of this low accurateness rate, process enactment information had not been used for analysis. The CMMI mentioned above suggest after second maturity level the mapping between the product and process data and also suggest to take into account this mapping for process improvement.

Since obtaining process traces and combining them with defect data are not easy, the analyses together with defect and process enactment data are not applied. Machine learning techniques are commonly used for prediction purposes, whereas process enactment data is slightly used. Our proposal is that machine learning approach can interpret more accurate performance results when the process enactment data is used together with product data.

To validate this proposal, we used a method for defect prediction by using machine learning classification [3]. The method clusters the data by using defect data with the context of defect management process before building the prediction model. The data of a small software company, Simsoft, was used for validation. This thesis explains the method in detail and provides its results from four case studies in two different projects.

1.1 Importance of Defect Data and Process Enactment Information Analysis

In all software projects correcting of detected software errors in an attentive and timely manner is vital. If defect correction cannot be completed on time and as it should be, it causes some risks such as giving poor quality products to the customer and / or exceeding the project budget due to error correction labor costs called as rework effort in literature. To minimize these risks, analysis of defect data is required. Besides defect data investigation provides quality improvement and prevents injection of new defects by application of preventive actions to the quality [4]. CMMI's Causal Analysis and Resolution support process area at maturity level 5 suggests selecting defect data for cause analysis [1, 5]. Percentage of defects removed, defect escape rates and number, and density of defects are suggested to be used as process-performance attributes in CMMI's Organizational Process Performance process area at maturity level 4. Historical defect data is suggested to be used for estimation of project planning parameters in CMMI's Integrated Project Management process area at maturity level 3. And finally, defect density derived measure is suggested to be used to address quality measurement objectives in CMMI's Measurement and Analysis process area at maturity level 2. On the other hand, percentage of defects is suggested to express process performance objectives in ISO/IEC 15504's performance management attribute [6]. Using defect density is suggested as process measurement attribute in ISO/IEC 15504's process measurement attribute.

Since software is different from other engineering disciplines, the information about executed software process during development constitutes importance for the quality and defectiveness of output product. What is the difference from other disciplines? Software production processes are not in a regular and static format as in a fabric production. For software development there are many ways for the production of process artifacts. And the results of applied processes show differences in different environment circumstances. Because of these reasons, evaluation of process knowledge with defect data might be so beneficial. In other words, without knowledge about the processes executed during developing the product, analyzing only defect data may not be sufficient to make decision and take preventive action. Process reference models like CMMI and assessment models like ISO 15504 address this issue over the concept of organizational maturity and process capability, and recommend applying prediction models at higher maturity/capability levels. But, we believe this should not be the only way to use such models.

1.2 Difficulty of Collecting Defect Data With Process Enactment

In recent years software defect data analysis has been a common research area [7, 8, 9]. But analysis and interpretation of software development process data are hard since software engineering is an area which is affected from multiple factors. For example, in some prediction studies [7, 10], authors suffer from the difficulty of collecting process-related data and taking into account all relevant evidences to generate a prediction model.

In order to understand the context of the product development traces, the traces throughout process practices must be recorded and the analysis of these tracks is required. However, since the nature of software process is abstract and dynamic, and there are too many variables which affect software process directly or indirectly, the measurement of software process is not easy especially in emergent contexts. This difficulty has supported the assessment insight by measuring the performance of software process using the characteristics of the developed product [11].

Since the process related data (e.g. the activities performed, the roles taken, the experience of the process performers) is not stored in the same tools with defect data, the accessibility to the product data and the mapping of process enactment are difficult. The collection of data from a tool's database is categorized as a third degree data collection technique since collection by extracting data from database is independent of real development time [12]. Since this situation causes some issues in mapping product data into process data which will be analyzed to understand the software development process, the most of the organizations can not use these data for prediction models. The organizations which use models or not, need guiding and methods about defectiveness evaluation and prediction.

1.3 Aims of This Study

The data of some process factors such as test type and project phase are stored in the defect tracking tool databases and analyzed by companies [2]. But the data of process enactment can not be provided in most of the cases. We aim to analyze product data with process enactment and show the benefits, if any, of this way in our study.

To do that we investigated the difference in machine learning prediction results with process enactment data and without enacment data. We chose machine learning analysis because of its pattern oriented nature. We believe that the patterns between software processes executed during development and related defect data from a product can be recognized with machine learning techniques.

1.4 Approaches Used in This Study

In this study, we intended to answer the questions; “Is process enactment data beneficial for defect prediction?”, “How can we use process enactment data?” and “Which approaches and analysis methods can our method support?”.

We used defect open duration metric as dependent variable in our analyses since defect open duration metric could easily be calculated according to the created date and closed date information of the defect obtained from the issue tracking tool. That is to say, we set open duration attribute as class attribute in Weka Tool [13] during machine learning classification operation.

To answer the questions, we first decided which indicators and metrics would be useful for this study. Therefore, the Goal-Question-Metric (GQM) [14] method was applied. The GQM goal was set as follows: to understand the effect of process enactment on software product defectiveness.

We used data of two completed projects in an emergent organization. We grouped defect data used in three categories.

1. Defect data detected during test activities: This data set was obtained from issue tracking tool database.
2. Product version and product size data: This data was obtained from configuration management tool and combined with the defect data. After combination, we had one data set that shows which defect is detected in which product version and how much size the product version has.
3. Process enactment data of defect management process: This data shows the features of each execution of the defect management process. In other words, inputs, outputs, performed process steps (activities) from the start to the end of the process, personnel roles which work for the process, and tools

andtechniquesinformation for each detected defect in software during tests and created in issue trackin tool is process enactment data of defect management process. This data set was manually obtained by using Process Execution Record (PER) and Process Similarity Matrix (PSM) assets.

PER (Process Execution Record) forms [16] was filled by interviewing with process experts. PSM (Process Similarity Matrix) was filled by manuallyreviewing issue tracking tool.

WEKA tool's [13] clustering facility (on cluster tab) was used to cluster the whole dataset obtained by combination of two categorized datasets, and classification facility (on classify tab) was used to conduct machine learning prediction.

We evaluated and compared the accuracy of the analysis results from the data sets with process enactment and without process enactment in the case study A and case study B separately.

1.5 Roadmap

The remainder of this thesis is organized as follows. Section two provides an overview of studies about the techniques used for software defect analysis and prediction, and explains the most known analysis methods. Section three gives the organization of the case studies and their results. Section four discusses the effect of process enactment in defect management process by comparing the performance results of the case studies with process enactment data and without process enactment data. Section five provides overall conclusions and future work.

CHAPTER 2

BACKGROUND

Defect prediction models do not only predict how many latent defects the software contains, but also in which parts of the software they are. In addition to that, they give clues on how to improve the quality of software development processes such as design and implementation. In other words, they aim to show project attributes that are related to better quality or reliability.

According to process reference models defect prediction can be used as an indicator of cause prevention. Therefore, the detection of cause and its place are visualized for process stakeholders.

2.1 Defect Prediction Basics

A "mistake" or "fault" can be committed to the software at any stage during development [15]. When it cannot be detected, it causes unintended work of the software product.

Defect is a stage of the “mistake” cycle. In most cases defects cause fault and failures but this is not a must.

Defects are crucial for the quality of the product since it shows the nonconformance to the customer requirements [17, 18]. Less defective software is more reliable and reliability is an attribute of quality.

Defect detection, correction and verification have cost in the project, because some effort is spent to find, resolve and verify detected defects. These activities are required for quality management. The cost of defect correction and re-testing has positive relation with the latency of the detection [19]. In other words, how much late the defect is detected, that much more defect correction and re-testing cost is. Therefore, defect prevention and the analysis of remaining defects are two important terms for software quality management.

For the defects, open duration metric is important because it gives information about the cost of the defect and makes us understand the trend in process with respect to time. Defect prevention is important to take actions before a flaw does not occur. That not only decreases rework effort, but also establishes an improved quality management system.

One of the defect prevention methods is defect prediction [5]. Defect prediction provides estimating number, type of the defects and their place in the software. In software development projects, planning of quality assurance and test activities, personnel allocating and training, process improvement can be done according to defect prediction results.

In this study we chose to answer our questions in a way that we try to predict defect open duration by using it as a class attribute in machine learning classification techniques for defect prevention.

The meanings of the terms mentioned in this study are below;

Case study: A research strategy, an empirical investigation technique that investigates a phenomenon within its real-life context [20]. This research technique is commonly used in software related studies.

Class attribute: Dependent variable in statistics that is used for classification, you have to select one of your attributes manually before executing classification analysis. Your data is classified according to your dependent variable and the tool gives you a model to be used for the prediction purpose with its performance evaluation values. Class attribute is called as classifier in some studies (i.e. [21]).

Defect: Software bug that causes an incorrect or unexpected result, or causes product to behave in unintended ways.

Defect open duration: The period that elapses from the detection and recording of the defect to the closure of it. It is in number of day unit.

Defect prediction: The analysis to forecast the behaviour of the defects in software product in future by various quantitative methods.

Defect prevention: The approach that avoids the defects from injection to the software. Defect prediction is only one of the activities that provide defect prevention [5].

Failure: The inability of software that does not perform its required functions within specified performance limits [18, 22].

Fault: An incorrect step, process, or data definition in a computer program which causes the program to perform in an unintended or unanticipated manner [23].

Machine learning:A scientific data mining discipline that concerns with the design and development of algorithms that allow computers to evolve behaviors based on empirical data, such as from sensor data or databases [24]. Machine learning aims to recognize patterns and learn. Then, make intelligent decisions based on data after learning. For this purpose some part of whole data is separated as training data and remaining data is kept for test.

Machine learning classification: The techniques that are called as supervised. A classifier is identified for classification.

Machine learning clustering: The techniques that are called as unsupervised. A classifier is not identified for clustering.

McCabe cyclomatic complexity:A software product complexity size measure computed with the number of decision nodes in software product.

Metric: The quantitative indicator of the measurement. In software engineering we can categorize the metrics in three classes: product metrics, process metrics and project metrics [17]. Product metrics are directly measured from software such as size, complexity and defect density. Process metrics are measure of performance of software processes such as testing time, and reviewing time. Project metrics give information about project characteristics such as earned value, and number of skilled project personnel.

Nonconformance: Lack of meeting specified requirements.

Performance evaluation values: The values evaluated in order to determine the accuracy and reliability of a technique.

Process: The series of activities to transform inputs to outputs. In software engineering, processes constitute software development life-cycle.

Process Enactment Data: The workflow of activities that are performed during process execution. The elements of the workflow are inputs, outputs, activities, roles, and tools and techniques.

Source lines of code (SLOC):A measure that shows the lenght of the software product which is computed by counting of the code lines.

Software configuration unit (SCU): The each part of the software product identified in order to to provide management easiness.

Software reliability: The probability of successful operation ofa computer program for a specified time in a specified environment.

Quality assurance: Systematic activities that are performed to determine whether product meets customer requirements.

Test: The software quality assurance activity that evaluates by running the code whether product meets customer requirements. This activity provides dynamic verification and validation of the software product.

2.2 Quantitative Analysis Methods Utilized for Defect Prediction

Both statistical and machine learning methods are used for the purpose of defect analysis and prediction. In addition to these studies there are reviews that assess the features and the technical characteristics of defect related measurement studies in literature. Before giving information about these previously performed studies, the analysis method commonly used in these researches are given in this section.

2.2.1 Statistical Methods

Before the discovery of data mining techniques, statistical methods are commonly used in software measurement and analysis like every other science. However, it is thought that statistical methods are insufficient to resolve complex patterns in high number of datasets. Common statistical methods used for defect analysis and prediction are given in the subsections below.

2.2.1.1 Reliability Models

Software reliability is a commonly used attribute of software quality for defect prediction. Software reliability models are based on defect data and the time between defect detected and resolved. They might be categorized in two types. One is called Rayleigh model which depicts the software development process beginning from project initiation to the end of maintenance phase. Second is called software reliability growth models and given with Jelinski-Moranda, Littlewood, Goel-Okumoto, Musa-Okumoto and S models in literature [17]. These second type models are based on exponential distribution approach.

Reliability models deal with several assumptions given below;

1. There are N unknown software faults at the start of testing.
2. Failures occur randomly (times between failures are independent).
3. All faults contribute equally to cause a failure.
4. Fix time is negligible.
5. Fix is perfect for each failure; there are no new faults introduced during collection.
6. Testing intervals are independent of each other.
7. Testing during intervals is reasonably homogeneous.
8. Numbers of defects detected during nonoverlapping intervals are independent of each other.
9. Test process is effective.

The accuracy of method is assessed according to the good-of-fit test results[25].

After data collection, below steps are performed.

Step 1: A model is selected.

Step 2: The parameters of the model are estimated.

Step 3: Fitted model is obtained by substituting the estimates of the parameters into the chosen model.

Step 4: A goodness-of-fit test is performed.

2.2.1.2 Hypothesis tests

The statistical method compares distribution characteristics such as mean and variance of two samples. Besides, whether there is the impact of an attribute on another attribute are searched with this analysis. According to the characteristics of our data set, t-test, Z-test, Chi-square, ANOVA tests are some of the applied statistical techniques [26].

During analysis, below steps are performed [26];
Step 1: Null hypothesis and alternative hypotheses are stated.
Step 2: Significance level is set.
Step 3: The probability value are obtained by using a statistical package program.
Step 4: The probability value is compared with significance level. If probability value is higher that significance level, null hypothesis is accepted.

2.2.1.3 Univariate analysis

With this analysis technique, defect classification and defect count understanding is easy. By analyzing representations, defect progress in future can be predicted, decision making are performed, and defect prevention is achieved [27].

Univariate analysis is carried out with the description of a single variable and its attributes of the applicable unit of analysis. If the variable defect data was the subject of the analysis, the researcher would look at how many subjects fall into a given defect data attribute categories. This analysis provides understanding with examined attribute of an object. Therefore, it is used for descriptive purposes. Variables could be either categorical or numerical.

A basic way of presenting univariate data is to create a frequency table which involves presenting the number of attributes of the variable studied for each case observed in the sample. Furthermore, graphical representation can be used to visualize data. Some of the mostly used graph types for defect data are Pareto Diagram, Histogram, Scatter Diagram and Control Chart.

Moreover, some quantitative measures called central tendency (mean, mode, median and dispersion) range, variance, max, min, quartiles, and standard deviation give information about the distribution of the attribute.

2.2.1.4 Bivariate Analysis

Bivariate analysis involves the analysis of two variables in order to determine the empirical relationship between them [27].

Bivariate analysis can be helpful in testing simple hypotheses of association and causality (checking to what extent it becomes easier to know and predict a value for the dependent variable if we know a case's value on the independent variable).Whereas the purpose of univariate analysis is describing, the purpose of bivariate analysis is explaining. It looks for the correlations, comparisons, relationships and causes between two variables.

During bivariate analysis, the steps given below are applied [28];

Step 1: The nature of the relationship whether the values of the independent variables relate to the values of the dependent variable or not is defined.

Step 2: The type and direction, if applicable, of the relationship are identified.

Step 3: It is determined if the relationship is statistically significant and generalizable to the population.

Step 4: The strength of the relationship is identified, i.e. the degree to which the values of the independent variable explain the variation in the dependent variable.

According to the measurement scales of our variables, statistical techniques that should be used are given below to understand the relationships between pairs of variables in a data set. When we called two variables as X and Y;

- If measurement scales of X and Y are interval and interval, and they are independently distributed, Pearson’s correlation is used.
- If measurement scales of X and Y are ordinal and ordinal, and they are independently distributed, Kendall’s Tau Spearman’s Rho Wilcoxon Signed Test or Mann-Whittney Test are performed.

•If measurement scales of X and Y are nominal and nominal, and they are independently distributed, Chi- square Lambda Test is performed.

•If measurement scales of X and Y are interval and interval, and one of them is dependent, simple linear regression is used.

•If measurement scales of X and Y are nominal and interval, and Y is independent, regression with dummy variables and one way analysis of variance are used.

2.2.1.5 Multivariate Analysis: Regression Models, PCA, DA, CA

Multivariate analysis involves observation and analysis of more than two statistical variables at a time.

Several mostly used multivariate analysis approaches are given below.

Linear Regression Analysis

In multivariate linear regression, several independent variables are used to predict one dependent variable. The relationship between dependent variable and independent variables are investigated [29].

Principal Component Analysis (PCA)

PCA decomposes a data table with correlated measurements into a new set of uncorrelated variables [30]. The importance of each component is expressed by the variance (i.e., eigenvalue) of its projections or by the proportion of the variance explained.

Discriminant Analysis (DA)

DA is used to predicting a nominal variable. The prediction of dependent variable is performed by looking for the relationships with the independent variables [29].

Correlation Analysis (CA)

Correlation analysis combines dependent variables to find pairs of new variables which have the highest correlation. However, new variables, even when highly correlated, do not necessarily explain a large portion of the variance of the original tables. This makes the interpretation of the new variable sometimes difficult [29].

2.2.2 Machine Learning Methods

Commonly used machine learning methods for defect prediction are given below.

2.2.2.1 K Nearest Neighbor (kNN)

There is no explicit training phase. K nearest neighbor algorithm searches for minimum distance from the query instance to the training samples to determine the K-nearest neighbors [31].

There is no assumption with data distribution [32]. kNN assumes that the data is in a feature space and the data points are in a metric space. The data can be scalars or possibly even multidimensional vectors. Since the points are in feature space, they have a notion of distance. This need not necessarily be Euclidean distance although it is the one commonly used.

During analysis, the steps given below are applied [33];

Step 1: Euclidean or Mahalanobis distance from target plot to those that were sampled is computed.

Step 2: Samples taking for account calculated distances are ordered.

Step 3: Optimal k-nearest neighbor according to performance value done by cross validation technique is heuristically chosen.

Step 4: An inverse distance weighted average with the *k*-nearest multivariate neighbors is calculated.

Its advantages arerobustness to noisy training data and effectiveness if the training data is large.

Its disadvantages areneed to determine value of parameter k (number of nearest neighbors), distance based learning is not clear which type of distance to use and which attribute to use to produce the best results, computation cost is quite high because of the need to compute distance of each query instance to all training samples.

2.2.2.2 C4.5 Decision Tree

Given a set S of cases, C4.5 first grows an initial tree using the divide-and-conquer algorithm as follows [34]:

• If all the cases in S belong to the same class or S is small, the tree is a leaf labeled withthe most frequent class in S.

• Otherwise, choose a test based on a single attribute with two or more outcomes. Makethis test the root of the tree with one branch for each outcome of the test, partition S intocorresponding subsets S1, S2, . . . according to the outcome for each case, and apply thesame procedure recursively to each subset.

1. Check for base cases for each attribute a,

2. Find the normalized information gain (difference in entropy) from splitting on a,

3. Let a_best be the attribute with the highest normalized information gain,

4. Create a decision node that splits on a_best,

5. Recurse on the sublists obtained by splitting on a_best, and add those nodes as children of node.

Its advantages are creating decision trees need no tuning parameters [35], no assumptions about distribution of attribute values or independence of attributes, no need for transformation of variables (any monotonic transformation of the variable will result in the same trees), the method automatically finds a subset of the features that are relevant to the classification, decision trees are robust to outliers as the choices of a split depends on the ordering of feature values and not on the absolute magnitudes of these values, and it can easily be extended to handle samples with missing values.

Its disadvantages are the need to construct a good classifier is proportional to the number of regions, complex view, and not a solution for all problems.

2.2.2.3 Multilayer Perceptron (MLP)

A learning rule is applied in order to improve the value of the MLP weights over a training set T according to a given criterion function [36].

This network has aninput layer(on the left) with three neurons, onehidden layer(in the middle) with three neurons and anoutput layer(on the right) with three neurons. There is one neuron in the input layer for each predictor variable. In the case of categorical variables,N-1 neurons are used to represent the N categories of the variable.

Step 1: The number of hidden layers to use in the network is selected.
Step 2: The number of neurons to use in each hidden layer is decided.
Step 3: A globally optimal solution that avoids local minima is found.
Step 4: It is converged to an optimal solution in a reasonable period of time.
Step 5: The neural network is validated to test for overfitting.

Its advantages are generalization and fault tolerance.

Its disadvantages are being computationally expensive learning process, giving no guaranteed solution, not scaling up well from small research systems to larger real systems.

2.2.2.4 Bayesian Belief Networks

A Bayesian belief network is a model that represents the possible states of a given domain. A Bayesian belief network also contains probabilistic relationships among some of the states of the domain [37].

Its steps are;

1. Gather information regarding the way in which the topic under discussion is influenced by conducting interviews
2. Identify the factors (i.e. nodes) that influence the topic, by analyzing and coding the interviews
3. Define the variables by identifying the different possible states (state-space) of the variables through coding and direct conversation with experts
4. Characterize the relationships between the different nodes using the idioms through analysis and coding of the interviews
5. Control the number of conditional probabilities that has to be elicited using the definitional/synthesis idiom [38]
6. Evaluate the Bayesian belief network, possibly leading to a repetition of (a number of) the first 5 steps
7. Identify and define the conditional probability tables that define the relationships in the Bayesian belief network
8. Fill in the conditional probability tables, in order to define the relationships in the Bayesian belief network
9. Evaluate the Bayesian belief network, possibly leading to a repetition of (a number of) earlier steps

Its advantages are providing knowledge in the form of causal structures [39], understandable and extensible network, used easily with missing data.

Its disadvantages are fixed sized hypothesis space [40], underfit or overfit of the data that may not contain any good classifiers if prior knowledge is wrong.

2.2.2.5 Apriori

Apriori mines for associations among items in a large database [41].

Its steps are;
Step 1: It mines a set of execution traces where each has a support value greater than the minimum support threshold [42].
Step 2: It extracts the traces which are a superset of all generator traces.
Step 3: It filters the non-generator traces away, leaving behind a set of generator traces.

Its advantages are usage of large itemset property, easily parallelization, easiness of implementation.

Its disadvantages are assuming transaction database is memory resident, requiring many database scans.

2.3 Defect Prediction Studies

We categorized studies in five categories as using process enactment data or not, using statistical methods or machine learning ones, using assets to collect process enactment data.

2.3.1 Prediction Models without Process Data by Statistical Analysis Methods

Koru and Tian [43] have validated the relationship between complexity and defect count metrics by using statistical hypothesis tests. They have investigated in their study how high complexity affects defect count.

Salman [44] has presented a measurement framework for component oriented software systems as his PhD thesis. He has generated statistical regression models to predict size and effort metrics. The independent variables of his models are component oriented metrics such as number of components, number of connectors, and number of interfaces.

Sivrioğlu and Tarhan [2] have prepared a case study by analyzing same dataset with both statistical and machine learning techniques but dataset has not included process enactment data. The dataset is the data of a completed software project. At the end of the study they have suggested to use contextual data for more accurate results.

Manzoor [45] has tried code metric to estimate defect fix time. But the estimation results have not been found promising. Manzoor has explained the reasons of this inaccurate estimation. He has given 14 factors which affect badly parametric estimation methods performed by using size metrics such as SLOC and FP (function points). His factors are pointed out to the dependence of analysis results to development environment and applied processes.

Ohlsson et al [46] have built prediction models by using Principal Component Analysis (PCA) and Discriminant Analysis (DA) methods. They have used product design metrics for prediction. And they have divided software modules into two categories called as fault-prone and not-fault-prone.

This type studies ignore process related data while analyzing software defect and product data, and their generated models have no process knowledge scraps. Because development environment has high impact on these models,they are specific to the examined project.

2.3.2 Prediction Models without Process Data by Machine Learning Methods

Boetticher [47] has suggested nearest neighbor machine learning method to group data. He has used product related metric data to predict the class in terms of its defectiveness status in the software.

Sivrioğlu and Tarhan [2] have analyzed defect data with both statistical and machine learning methods. They have mentioned that the results of machine learning techniques are more accurate than the ones of statistical techniques, because machine learning gives better results when number of data is high than statistical hypothesis tests when sufficient data is supplied.

Sandhu et al [48] have recommended genetic algorithm technique to predict fault proneness of software modules. He has used requirements and code metrics called as product related metrics for his research.

Çatal and Diri [49] have reviewed software defect prediction studies in a systematical way. They have separated the studies to categories before review. The review states that the studies with using class-level, process-level and component-level measures are not sufficient. Besides, machine learning methods are suggested because they give better results than statistical analysis and expert view methods.

Ahsan et al [50] have conducted a study to estimate bug fix effort. R (Pearson correlation coefficient), MAE (Mean Absolute Error), RMSE (Root Mean Square Error), MMRE (Mean Magnitude of Relative Error) and RRSE (Root Relative Square Error) performance values of five machine learning methods are compared at the end of the study. Because the defect fix effort data are not available, defect fix days metric is used as independent variable for prediction. Product metrics such as number of functions, number of changed operators, SLOC and complexity are included in analysis as input attributes.

When number of data is high, machine learning techniques can give promising results for prediction. But without process enactment obtained, models can not be used for other projects or other development teams of same project.

2.3.3 Prediction Models with Process Data by Statistical Analysis Methods

It is slightly possible to find studies by using process data in literature. Jalote et al [51] have explained a defect prediction approach by performing quantitative quality management and statistical process charts.

Wahyudin et al [9] have presented a defect prediction model by using statistical hypothesis with a combination of product and process measures.

Dhiauddin [8] has generated a prediction model for testing phase in his master thesis. With this model he discovers the strong factors that contribute to the number of testing defects by using statistical methods such as regression analysis.

Gokhale and Mullen [52] have hypothesized a Laplace Transform of the Lognormal distribution model with defect repair times data in day unit. At the same time, they give several factors which are considered affecting defect repair time and causing a lognormal distribution in repair rates because of the factors' multiplicativeness.

Schneidewind [53] has explained the delay between fault detection and fault correction times with exponential distribution. To obtain this statistical empirical result, MSE (Mean Square Error) values of three operational increments have observed in a project. Failure rate, test time parameters are used as input attributes in model.

As mentioned in introduction section, process measures can not be used in most cases because of the collection difficulty. However, the studies which includes process related metrics and analyzed product metrics together with process metrics gave more reliable results for software projects.

If we use process enactment by taking a step forward of process related metrics, the models are going to give more reliable results and predictions can be used for similar projects or development teams with similar environment.

2.3.4 Prediction Models with Process Data by Machine Learning Methods

Fenton and Neil [54] have evaluated defect oriented software metrics and statistical models. They have specified that reliability can not be computed by using defect density because the defects which cause not working of software (its fault) can not be parsed and user oriented defects cannot be chosen. They have stated some inconsistent results that while there is positive correlation between number of defects and other metrics such as software size, in some studies there are negative regression. Regression models provide information only about the past and it does not indicate a prediction model for new data. To analyze average values in data does not explain raw data; therefore it does not give realistic results. The relationship between size and defect is so complex that simple models are insufficient to present these complicated relations. They suggest probabilistic methods such as Bayesian Belief Network (BBN) to present complicated relations between defect and the factors which affect it.

Leey et al [7] have developed a prediction model with micro interaction metrics which are supposed as process-related metrics. In this study, they have made comparisons between the accuracy results of the model of code metrics, the model of history metrics, and the combination of them. They use machine learning classification and regression techniques.

Fenton et al [55] have suggested Bayesian Belief Networks machine learning technique as prediction model. Process data is given for this model, again.

He et al [56] have generated models with J48 (C4.5), Naïve Bayes and SVM (Support Vector Machines) by using same metrics with previously mentioned two studies. The performance of the models has been evaluated by MAE, MMRE and comparison between minimum MAE and median values of data groups.

Song et al [57] have suggested association rule mining for defect correction effort prediction. Apriori accuracy values such as mean, median and standard deviation have compared with the ones of PART, C4.5 and Naïve Bayes approaches. Defect type metric has been used as input data. Also, false negative rate, false positive rate performance values have been reviewed for evaluation.

Zeng and Rine [58] have estimated defect fix effort by using dissimilarity matrix and Self Organizing Maps (Kohonen Networks) which is a type of Neural Networks method. With this data mining technique the data have been clustered for prediction. Model performance has been evaluated by magnitude of relative error (MRE) values of 6 grouped data sets. The input attributes of the model are defect fix time in hour unit, defect severity, the activity during which the defect is detected, system mode, defect category and SLOC (source lines of code) changed. Defect severity, detection activity, system mode and defect category attributes can be considered as contextual metrics.

Thaw et al [59] have performed a similar study with Zeng and Rine. They have concluded their study that prediction model gives accurate results for the projects which have same software development processes like product line projects.

Menzies et al [60] have presented a case study that compares defect analysis results between machine learning and manual analysis used human expertise. ODC (Orthogonal Defect Classification) technique has been used. They have found that manual domain expertise gives more accurate results than treatment learning. But manual analysis is insufficient when we have a complex and large dataset. They have specified that the application of both manual and machine learning analysis gives the most accurate results.

Weiss et al [61] have used the defects life-time phases gone through issue tracking tool as the attributes for defect fix effort prediction. They compared two types of Nearest Neighbor approaches called as with (α-kNN) and without thresholds (kNN). They used text mining for grouping the data before kNN analysis.

Hassouna and Tahvildari [62] have improved Weiss' study by adding 1. data enrichment to infuse additional issue information into the similarity-scoring procedure, 2. majority voting to exploit many of the similar historical issues repeating effort values, 3. adaptive threshold to automatically adjust the similarity threshold to ensure that they obtain only the most similar matches and 4. binary clustering to form clusters when the similarity scores are very low phases.

Hewett and Kijsanayothin [63] have penned down a comprehensive study regarding defect repair time prediction. Firstly, they have applied five different empirical machine learning approaches to two individual data sets with and without attribute selection. AUC (Area Under the ROC Curve), TPR (True Positive Rate, Recall, Sensitivity, Hit Rate), PREC (Precision), FPR (False Positive Rate, False Alarm Rate), ACC (Accuracy) and RMSE (Root Mean Square Error) values have been evaluated for performance. Secondly, they have applied three analytical models: S (Schneidewind) model [53], GM (Gokhale and Mullen) [52] model, their own proposed model and compared the results. Defect detected testing phase, defect severity, defect state and defect state update dates have been used as input attributes for prediction models.

Menzies et al [64] have pointed the importance of the models of similar regions than global ones in empirical studies. Two tools called WHERE to cluster algorithm that divides the data and WHICH learner to find treatments in clusters used to compare the treatments learned from global or local contexts.

It is seen that researchers' insight has been changing as clustering data before modeling. Therefore, we can obtain more local (specialized) results and accurate models for prediction. We will provide this clustering by using process enactment data in our study before applying machine learning techniques. The performance results of clustered dataset and not clustered will be compared.

2.4 Methods to Collect Process Enactment Data

Tarhan and Demirörs [65, 66] have emphasized the importance of process differences in software projects. They have defined and applied some assets such as Metric Usability Questionaire (MUQ), Process Execution Record (PER), and Process Similarity Matrix (PSM) for data collection.They used MUQ for the decision of usable metrics, PER and PSM for collection and verification of process enactment data.

It is seen that researchers claim the benefits of process measures, machine learning methods, some data collection and grouping methods for defect prediction models one by one. However, none of them use several of these methods together for empirical studies. Combining defect data with process enactment and generating a model from combined data by using above quantitative measurement techniques, we believe, is a promising research topic.

2.5 Validation Methods in Machine Learning and Weka Tool

Machine learning validation methods provide assessing the performance of the models by estimating their accuracies. In other words, it can be evaluated how well the mining models perform against real data.

The descriptions of commonly used validation methods are given below.

Training and Testing Data Sets

In this method, the data set are separated into two sets for training and test. Mostly, training data set is bigger than the portion of the test set. After a model has been processed by using the training set, the model is tested by making predictions against the test set. Since, the data in the testing set already contains known values for the attribute that you want to predict, it is easy to determine whether the model's guesses are correct [72]. The splitting 66% of the data set for training set and remaining for test is a commonly used technique.

Cross Validation

The original data set is randomly partitioned into k sets. Of the k sets, a single set is retained as the validation data for testing the model, and the remaining $k - 1$ sets are used as training data. The cross-validation process is then repeated k times (the folds), with each of the k sets used exactly once as the validation data. The k results from the folds then can be averaged (or otherwise combined) to produce a single estimation. The advantage of this method over repeated random sub-sampling is that all observations are used for both training and validation, and each observation is used for validation exactly once [72]. 10-fold cross-validation is commonly used type of cross validation.

In Weka tool, the models are validated by selecting one of the three options given below;

- Using all data set file classified as training set,
- Using another supplied data set file from classified data set as test data,
- k-fold cross validation,
- 66% percentage split.

At the end of classification and clustering executions some performance values are given as output in Weka. These performance values are "correctly classified instances", "incorrectly classified instances", kappa statistic, mean absolute error, root mean squared error, relative absolute error, root relative squared error, TPR, FPR, recall, precision, F-measure and ROC area.

CHAPTER3

DESIGN OF CASE STUDIES

As seen from literature search, previous studies generally do not include process metrics. Even though number of them is low, process metrics which measure test performance, defect resolution timeliness and reliability are analyzed in several studies. But, in this study we do not focus on process metrics directly. Instead of this, we assume that process enactment data give detailed information about process tracks. Therefore, we can investigate the advantages of process related data usage for analysis and prediction. Our motivations to choose process enactment to understand and predict defect data are detailed below;

1. Since the nature of the metric is subsequent, process metrics can be collected only after application of the process. They are performance values. In other words you cannot collect test effectiveness metric, before running any test. This situation causes late feedback in most cases. It means that we are late to prevention; we can only apply corrective action items. However, enactment data can be collected before process execution according to our planning, by taking into consideration previous similar project process applications or company process assets.

2. Process performance metric results are specific to product and project, because they are affected from many factors. These factors can be skills of the project staff, customer experience in domain area, programming language, number of personnel, suitable tool usage etc. On the other hand, process enactment data is more usable to generalize the analysis results. We can use the analysis results of one project for the prediction of other projects that apply same process attribute

patterns.

3. Process metrics cannot be collected and recorded automatically by tools. We need manual calculations after process implementations even though we gather data from databases. But enactment data that is used by this study had been recorded in real time while process was being implemented.

Our base questions waiting to be answered in our study are “Is process enactment data beneficial for defect prediction?”, “How can we use process enactment data?” and “Which approaches and analysis methods can our method support?”.

We applied case study method from empirical investigation techniques. There are four types of case studies according to objective aspect [12]: exploratory, descriptive, explanatory, or improving. Other categorization related with case study attributes are: 1) Single-case vs. multiple-case, and 2) Holistic vs. embedded.

In these case studies we have four cases and we do not have multiple units within a case since we can say that our case study design is compatible with multiple-case and holistic one. The purpose is descriptive in Case Study 1A and Case Study 2A since we give machine learning analysis results with the only defect data metrics’ analysis results as is. On the other hand, the purposes of Case Study 1B and Case Study 2B are “exploratory” since we investigate what will happen when we use process enactmentmetrics together with defect metric. All four case studies are performed for an improving purpose. We intend to improve machine learning defect prediction aspect.

According to data collection aspect there are three categories of methods [12]: Direct (e.g. interviews), indirect (e.g. tool instrumentation) and independent (e.g. documentation analysis).

We used all of the three data collection approaches. Fully structured interviews were performed with process experts by filling Process Executions Records

(PER). Issue tracking tool and configuration management tool were used as third degree archival data. The data had already stored in tools while the process was being executed. The quality of the data has improved by the support from expert opinions.

We analyzed data quantitatively with machine learning classification techniques. We interpreted results on comparative basis. We compared the validity results of the project data with process enactmentwith the one without process enactment. Also, the performance values which show classification model prediction accuracy in Weka output were evaluated for validation.

The variations between four case studies are listed below;
Case Study 1A: Project-1 data was collected based on defined metrics. We ignored process enactmentdata in this case study concept.

Case Study 1B: Project-1 data wascollected based on defined metrics. We took into account process enactmentdata in this case study concept and we included it in the analysis.

Case Study 2A: Project-2 data wascollected based on defined metrics. We ignored process enactmentdata in this case study.

Case Study 2B: Project-2 data wascollected based on defined metrics. We took into account process enactmentdata in this case study and we included it inthe analysis.

Our proposed method consists of the sequential steps below(Figure 3.1);

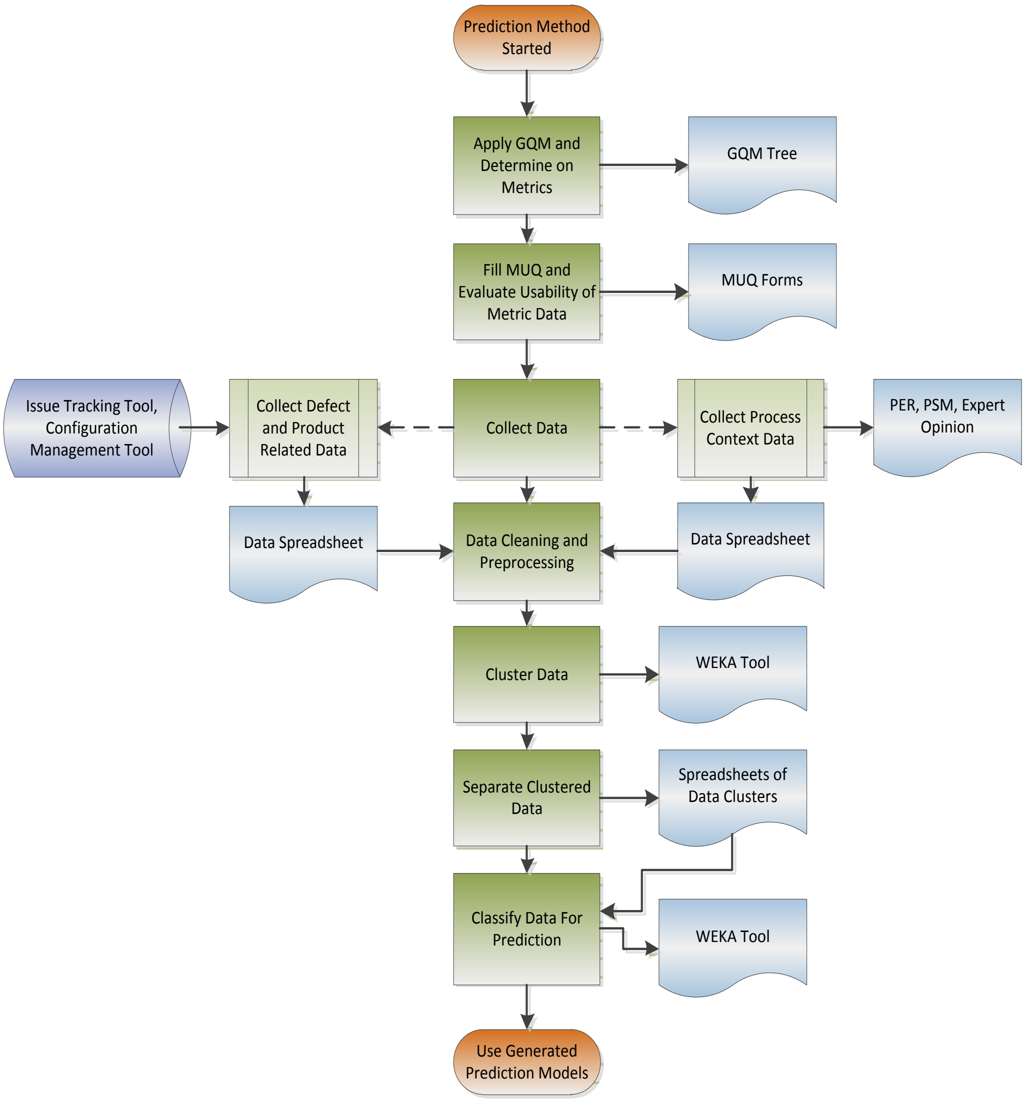

Figure 3.1 Proposed Method

3.1 Goal-Question-Metric (GQM) Tree Approach

GQM [14] approach proposes a top-down measurement definition. The approach states that a goal-based measurement way provides opportunity to the organizations for specifying themselves and their project's goals, tracing the goals to the questions that ask what they should wonder for that goal and finally specifying the interpretation of metrics collected for those questions.

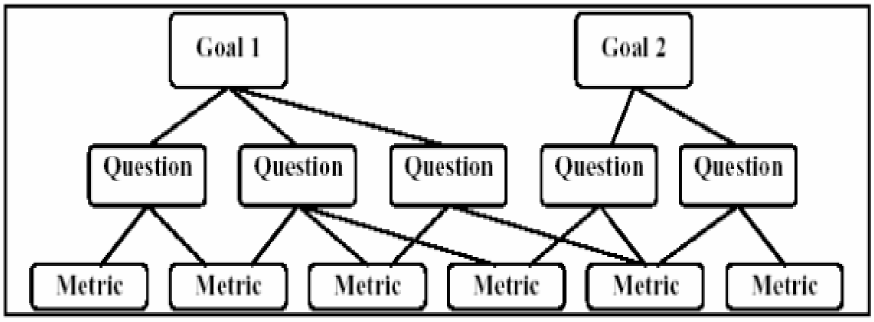

Figure 3.2 The Goal-Question-Metric Hierarchy [14]

In this study before analysis phase, to make analyses in terms of our goals, Goal-Question-Metric (GQM) method was applied. Firstly, our aims wereset; secondly the questionsweredefined for each goal; thirdly to answer the question, related metrics and analysis methods werespecified (Table 3.1).

Table 3.1 GQM for This Study

GOAL	QUESTION NO	QUESTION	ANALYSIS METHOD	DERIVED METRIC	BASIC METRIC NO	BASIC METRIC	CASE STUDY NO
To understand if there is effect of process enactment on software product defectiveness.	**3.1**	How much impact has process enactment on defect open duration prediction?	Bayesnet, Logistic, C4.5 Tree, Multilayer Perceptron Machine Learning Techniques	Defect Data: open duration (closed date-created date)	**3.1.1**	**Defect and Product Data:** detected module name, closed date, created date, detected test type, product version, product SLOC, product complexity, reproducibility, detected project phase	Case Study 1A (Project-1), Case Study 2A (Project-2)
				Defect Data: open duration (closed date-created date)	**3.1.2**	**Defectand Product Data:** detected module name, closed date, created date, detected test type, product version, product SLOC, product complexity, reproducibility, detected project phase **Process Enactment Data:** defect management process attributes	Case Study 1B (Project-1), Case Study 2B (Project-2)

3.2 Metric Usability Questionnaire (MUQ)

MUQ is a form filled according to metric usability attributes [16]. Each form is filled for one metric. The questions and ratings are different for basic metrics (Figure 3.3) and derived metrics (Figure 3.4). Rating is quantitatively calculated according to metric usability factors (MUF) by dividing “Yes” answers to the all number of questions. Obtained percentage value is qualitatively categorized according to the rules below.

- If the percentage value of factor is between %86-100, MUF is qualitatively categorized as fully statisfied (F).
- If the percentage value of factor is between %51-85, MUF is qualitatively categorized as largely statisfied (L).
- If the percentage value of factor is between %16-50, MUF is qualitatively categorized as partially statisfied (P).
- If the percentage value of factor is between %16-50, MUF is qualitatively categorized as not statisfied (N).

In rating phase, metric usability attributes (MUA) are ordered sequential to their criticality: 1) data metric identity, 2) data existence, 3) data verifiability, and 4) data dependability. If the regarding values of MUA-1 and MUA-2 are F and F; and MUA-3 and MUA-4 are F or L, the basic metric is “usable”.

		Please rate each attribute in four scales, based on answers to questions as indicators:
Metric Name:		F: Indicators of the attribute are fully satisfied (%86-100)
Conceptual Definition:		L: Indicators of the attribute are largely satisfied (%51-85)
Assessed On:		P: Indicators of the attribute are partially satisfied (%16-50)
Assessed By:		N: Indicators of the attribute are not satisfied (%0-15)

Attributes			**Answers**	**Rating**	**Expected Answers**
	Indicators				
Measure Identity			**MUF-1**	**F**	
	Q1	Which entity does the measure measure?			
	Q2	Which attribute of the entity does the measure measure?			
	Q3	What is the scale of the measurement data? (nominal, ordinal, interval, ratio, absolute)			Ratio, Absolute
	Q4	What is the unit of the measurement data?			
	Q5	What is the type of the measurement data? (integer, real, etc.)			
	Q6	What is the range of the measurement data?			
Data Existence			**MUF-2**	**F**	
	Q7	Is measurement data existent?			Available > 20
	Q8	What is the amount of overall observations?			
	Q9	What is the amount of missing data points?			
	Q10	Are data points missing in periods? (If yes, please state observation numbers for missing periods)			
	Q11	Is measurement data time sequenced? (If no, please state how measurement data is sequenced)			
Data Verifiability			**MUF-3**	**F**	
	Q12	When is measurement data recorded in the process? (at start, middle, end, later, etc.)			
	Q13	Is all measurement data recorded at the same place in the process? (at start, middle, end, later, etc.)		☐	Yes
	Q14	Who is responsible for recording measurement data?			
	Q15	Is all measurement data recorded by the responsible body?		☐	Yes
	Q16	How is measurement data recorded? (on a form, report, tool, etc.)			
	Q17	Is all measurement data recorded the same way? (on a form, report, tool, etc.)		☐	Yes
	Q18	Where is measurement data stored? (in a file, database, etc.)			
	Q19	Is all measurement data stored in the same place? (in a file, database, etc.)		☐	Yes
Data Dependability			**MUF-4**	**F**	
	Q20	What is the frequency of generating measurement data? (asynchronously, daily, weekly, monthly, etc.)			
	Q21	What is the frequency of recording measurement data? (asynchronously, daily, weekly, monthly, etc.)			
	Q22	What is the frequency of storing measurement data? (asynchronously, daily, weekly, monthly, etc.)			
	Q23	Are the frequencies for data generation, recording, and storing different?		☐	No
	Q24	Is measurement data recorded precisely?		☐	Yes
	Q25	Is measurement data collected for a specific purpose?		☐	Yes
	Q26	Is the purpose of measurement data collection known by process performers?		☐	Yes
	Q27	Is measurement data analyzed and reported?			Yes
	Q28	Is measurement data analysis results communicated to process performers?			Yes
	Q29	Is measurement data analysis results communicated to management?			Yes
	Q30	Is measurement data analysis results used as a basis for decision making?			Yes
Data Normalizability					
	Q31	Can measurement data be normalized by parameters or measures? (If yes, please specify them)			
Data Integrability					
	Q32	Is measurement data integrable at project level?			
	Q33	Is measurement data integrable at organization level?			

(a) Metric Usability Questionnaire

Metric Name:		
Conceptual Definition:		
Assessed On:		
Assessed By:		

Metric Usability Attributes	**Rating**	**Expected Rating**
Metric Identity (MUA-1)	F	F
Data Existence (MUA-2)	F	F
Data Verifiability (MUA-3)	F	L or F
Data Dependability (MUA-4)	F	L or F
Metric Usability Result	**F**	**L or F (Usable) -- Not Usable otherwise**

(b) Metric Usability Rating

Figure 3.3 Metric Usability Questionnaire and Rating for Basic Metrics

The difference of the derived metric rating from basic metric one is that MUF 3&4 values of the basic metrics should be F or L for a derived metric to be an “usable” derived metric.

		Please rate each attribute in four scales, based on answers to questions as indicators:
Metric Name:		F: Indicators of the attribute are fully satisfied (%86-100)
Conceptual Definition:		L: Indicators of the attribute are largely satisfied (%51-85)
Assessed On:		P: Indicators of the attribute are partially satisfied (%16-50)
Assessed By:		N: Indicators of the attribute are not satisfied (%0-15)

Attributes			**Answers**	**Rating**	**Expected Answers**
	Indicators				
Measure Identity			**MUF-1**	**F**	
	Q1	Which entity does the measure measure?			
	Q2	Which attribute of the entity does the measure measure?			
	Q3	What is the scale of the measurement data? (nominal, ordinal, interval, ratio, absolute)			Ratio, Absolute
	Q4	What is the unit of the measurement data?			
	Q5	What is the type of the measurement data? (integer, real, etc.)			
	Q6	What is the range of the measurement data?			
Data Existence			**MUF-2**	**F**	
	Q7	Is measurement data existent?			Available > 20
	Q8	What is the amount of overall observations?			
	Q9	What is the amount of missing data points?			
	Q10	Are data points missing in periods? (If yes, please state observation numbers for missing periods)			
	Q11	Is measurement data time sequenced? (If no, please state how measurement data is sequenced)			
Data Verifiability			**MUF-3**	**F**	
	Q12	When is measurement data recorded in the process? (at start, middle, end, later, etc.)			
	Q13	Is all measurement data recorded at the same place in the process? (at start, middle, end, later, etc.)		□	Yes
	Q14	Who is responsible for recording measurement data?			
	Q15	Is all measurement data recorded by the responsible body?		□	Yes
	Q16	How is measurement data recorded? (on a form, report, tool, etc.)			
	Q17	Is all measurement data recorded the same way? (on a form, report, tool, etc.)		□	Yes
	Q18	Where is measurement data stored? (in a file, database, etc.)			
	Q19	Is all measurement data stored in the same place? (in a file, database, etc.)		□	Yes
Data Dependability			**MUF-4**	**F**	
	Q20	What is the frequency of generating measurement data? (asynchronously, daily, weekly, monthly, etc.)			
	Q21	What is the frequency of recording measurement data? (asynchronously, daily, weekly, monthly, etc.)			
	Q22	What is the frequency of storing measurement data? (asynchronously, daily, weekly, monthly, etc.)			
	Q23	Are the frequencies for data generation, recording, and storing different?		□	No
	Q24	Is measurement data recorded precisely?		□	Yes
	Q25	Is measurement data collected for a specific purpose?		□	Yes
	Q26	Is the purpose of measurement data collection known by process performers?		□	Yes
	Q27	Is measurement data analyzed and reported?			Yes
	Q28	Is measurement data analysis results communicated to process performers?			Yes
	Q29	Is measurement data analysis results communicated to management?			Yes
	Q30	Is measurement data analysis results used as a basis for decision making?			Yes
Data Normalizability					
	Q31	Can measurement data be normalized by parameters or measures? (If yes, please specify them)			
Data Integrability					
	Q32	Is measurement data integrable at project level?			
	Q33	Is measurement data integrable at organization level?			

(a) Metric Usability Questionnaire

Metric Usability Attributes	Rating	Expected Rating
Metric Identity (MUA-1)	F	F
Data Existence (MUA-2)	F	F
Data Verifiability (MUA-3)	F	L or F
Data Dependability (MUA-4)	F	L or F
MUF-3&4 for basic metric-1	F	L or F
MUF-3&4 for basic metric-2	F	L or F
MUF-3&4 for basic metric-n	F	L or F
Metric Usability Result	**F**	**L or F (Usable) -- Not Usable otherwise**

(b) Metric Usability Rating

Figure 3.4 Metric Usability Questionnaire and Rating for Derived Metrics

In this study after defining the metrics, metric usability analysis for each basic metric has been performed to determine if the metric is applicable and available for our study. MUQ form was filled for each basic metric and the derived metric "defect open duration". During the examination of filled MUQ forms it was determined not to use number of requirements based on product version metric. Because "number of requirements" metric was collected on monthly basis instead of product version basis, this period was not applicable for our analysis goal.

3.3 Data Collection

In this study the two projects' data of Simsoft company is used. Simsoft Computer Technologies Co., Ltd. is a software development company established in 2006. It is especially experienced in simulation systems. Simsoft is conducting business as a university - industry Cooperation Company in Technology Development Center at Middle East Technical University Technopolis in METU Campus. It has 30 personnel, including Software Engineers, Modeling and Graphics Designers, and Quality Assurance Supporters. The company has developed software projects for a large number of institutes especially for defense industry by now. The organization has already ISO 9001 [67] certificate and executes documented process assets in compatible with CMMI Level 3. The company has a specific measurement process, in this concept obeying policies for analyzing the monthly data and reporting the results to high level management.

The projects whose defect data is used are listed below;

Project-1: The software product developed in the project has 2 Software Configuration Units (SCU) with 4 module types, and 6 personnel worked for 7 months project duration. At the end of the development, C++ source lines of code are 23 KLOC, number of requirements is 955, and the number of defects detected during tests is 296. This project's development phase was completed in January 2012.

Project-2: The product has 14 Software Configuration Units (SCU), and 15 personnel worked for 8 months project duration. At the end of the development C# source lines of code is 188 KLOC, number of requirements is 1492, and the number of defects detected during tests is 425. This project's development phase was completed in June 2011.

3.3.1 Defect and Product Size Data Collection

Since software testing is a must and a part of development, resolution of detected defects is a necessity. With this aspect for the tracking of defects in software, a tracking tool is used by lots of institutions contemporarily.With these tools a detected defect during any quality activity can be recorded and assigned to related personnel for resolution. After assignment; monitoring, verifying and closing activities are tracked over these tools. In addition to the tracking of defect status, the detailed information regarding the defect such as software module, product version where the defect is detected, test type and source project phase during which the defect is detected can be accessed at any time since defect information is stored with its history in the database. These tools store descriptions of the defects detected on software, detection dates and resolution status of defects.

While using issue tracking tool for the monitoring of the status of the defects detected in software product, in order to perform the updates on product in a controlled manner, organizations need configuration management tool. Configuration management tools provide a common environment to the developers to track the modifications in product. These tools do not allow multiple personnel to modify the product at the same time. The personnel can access whole update information beginning from first creation of the product in the tool. With the aid of configuration management tool, the important information about software product can be obtained historically since it stores all product versions in a historical manner and anyone can access versioned product at any time.

The defect related basic metrics' data;detected software configuration unit (SCU) name, created date, closed date, test type, product version and reproducibility were extractedfrom issue tracking tool database. Besides, the defect related derived metric called as"defect open duration" was manually calculated as the difference between the closed and the created dates.

The project phase process metric data is manually collected by filling "Project Phase" column in Excel sheet while directly interviewing with the process expert.

The product size basic metrics' data; product version size (logical source lines of code) and complexity (McCabe cyclomatic complexity), however, are obtained indirectly from the tool. We say "indirectly" because these metrics are calculated with LOCMetrics tool [71] by using the product version where the defect is detected from the information recorded in the tool. In other words, to collect SLOC and McCabe cyclomaticcomplexity, configuration management tool was used together with the product version information in issue tracking tool, and the total SLOC was counted by LOCMetrics and recorded manually. Metric descriptions are given in Table 3.2.

Table 3.2 Defect and Product Related Metric Descriptions

Metrics	Metric Description	Measurement Scale
Remaining Open Duration	The time starting with the creation of the defect and finishing with the closure of the defect. Calculated by the difference of defect closed date and defect created date. Unit is number of days.	Absolute
Detected SCU Name	The name of the software configuration unit (SCU) where the defect is detected. Entered by developer to the issue tracking tool.	Nominal
Created Date	The date when the defect is detected. Filled by the issue tracking tool automatically when the tester record the defect.	Interval
Closed Date	The date when the defect is closed. Filled by the issue tracking tool automatically when the project manager change the status of the defect as "Closed".	Interval
Test Type	The name of test type during which the defect is detected. Entered by tester to the issue tracking tool.	Nominal
Product Version	The version of the software product which the defect is detected. Entered by tester to the issue tracking tool.	Ordinal
SLOC (Source Lines of Code)	The size of the product version where the defect is detected. Collected from configuration tool by using Locmetrics tool.	Absolute
Complexity	The McCabe complexity of the product version where the defcet is detected. Collected from configuration tool by using Locmetrics tool.	Absolute
Reproducibility	The repetability of the defect detected. Entered by tester to the issue tracking tool.	Nominal
Project Phase	The project phase where the defect detected. Collected manually by domain expert.	Nominal

The raw data of regarding metrics are gathered in an Excel sheet.

3.3.2 Process EnactmentData Collection

Process Execution Record (PER) and Process Similarity Matrix (PSM) assets are utilized to gather process traces [16]. First, you decide on which process data is needed for your analysis. Then, PER is filled out for your regarding process and process attributes such as inputs, outputs, activities and tools. This knowledge isthen entered to PSM Excel sheet for each process executions. For example, process execution might be each product version release for a configuration management process.

3.3.2.1 Process Execution Record (PER) Asset

PER is a form in Word file format (Figure 3.5) used to define all actual process values in process attributes basis. Inputs, outputs, roles, tools and techniques all are process attributes and with the help of PER form, all alternative values of them for process executions are recorded. Prepared list in PER are used to fill PSM.

Process Execution Record
(Internal Attributes)

Process Name:		Recorded On:	
Process Execution No:		Recorded By:	

1. **Inputs:** Please list the inputs to the process execution.

No	Name	Description
1		

2. **Outputs:** Please list the outputs from the process execution.

No	Name	Description
1		
2		

3. **Activities:** Please list in sequence the activities that were performed while executing the process.

No	Name	Description
1		
2		
3		
4		

4. **Roles:** Please list the roles that were allocated responsibilities in process execution.

No	Name	Description
1		
2		

5. **Tools and Techniques:** Please list the tools and techniques that are used to support process execution.

No	Name	Description

Figure 3.5 Process Execution Record (PER)

In this study the collection of defect management process enactment data was aimed in order to capture the traces of defect management process and combine it with defect related process data and product data for prediction analysis. PER forms were collected with expert opinions by interviewing.

3.3.2.2 Process Similarity Matrix (PSM) Asset

PSM is a spreadsheet in Excel file format (Figure 3.6) used to gather process attribute values for all process executions. Horizontally there are process attributes specified in PER before, vertically there are numbered process executions. The cells in matrix is filled by entering a circle sign if the process attribute is applicable for regarding process execution. After PSM is completed, the differences in columns are examined and clustering is manually performed.

		Process Executions																						
Process Attributes		PE 1	PE 2	PE 3	PE 4	PE 5	PE 6	PE 7	PE 8	PE 9	PE 10	PE 11	PE 12	PE 13	PE 14	PE 15	PE 16	PE 17	PE 18	PE 19	PE 20	PE 21	PE 22	… ....
1	1.1 <Input 1>	O	O	… …																				
	1.2 <Input 2>	O	O	… …																				
2	2.1 <Output 1>	O	O	… …																				
	2.2 <Output 2>		O	… …																				
3	3.1 <Activity 1>	O	O	… …																				
	3.2 <Activity 2>	O	O	… …																				
	3.3 <Activity 3>	O	O	… …																				
	3.4 <Activity 4>		O	… …																				
4	4.1 <Role 1>	O	O	… …																				
	4.2 <Role 2>	O	O	… …																				
5	5.1 <Tools and Techniques 1>	O	O	… …																				
	5.2 <Tools and Techniques 2>		O	… …																				

Figure 3.6 Process Similarity Matrix (PSM) In Literature

In this study we used PSM a little bit different from the utilization in literature (Figure 3.7). We transposed the matrix vectors. The process executions were horizontally collected since this structure was more convenient to combine with collected defect and product data. In other words, this way provided straightforwardness since also in the spreadsheet that consisted of the defect and product data, the metric attributes were in column vector against which regarding process executions exist. Besides, we entered “1” or “0” instead of “o” or “ ”. Thus, “1” and “0” scaling could be identified in numeric measurement scale by Weka tool. Process enactment data identified as numeric could be clustered by machine learning clustering technique. PSM sheets were collected from issue tracking tool by extracting historical defect management process data such as defect status updates and the roles of the personnel who had updated the defect status.

	Process Attributes											
	1 Inputs		2 Outputs		3 Activities				4 Roles		5 Tools and Techniques	
Process Executions	1.1 <Input 1>	1.2 <Input 2>	2.1 <Output 1>	2.2 <Output 2>	3.1 <Activity 1>	3.2 <Activity 2>	3.3 <Activity 3>	3.4 <Activity 4>	4.1 <Role 1>	4.2 <Role 2>	5.1 <Tools and Techniques 1>	5.2 <Tools and Techniques 2>
PE1	1	1	1	0	1	1	1	0	1	1	1	0
PE2	1	1	1	1	1	1	1	1	1	1	1	1
PE3	……	……	……	……	……	……	……	……	……	……	……	……
PE4												
PE5												
PE6												
PE7												
PE8												
PE9												
PE10												
PE11												
PE12												
PE13												
PE14												
PE15												
PE16												
PE17												
PE18												
PE19												
PE20												
PE21												
PE22												
…….												

Figure 3.7 Process Similarity Matrix (PSM) In This Study

3.4 Data Cleaning and Preprocessing

In this phase the redundant data and attributes whether there are in data set are removed from data set to avoid from overfitting and multicollinearity during machine learning analysis techniques. The redundant data might be the rows that have missing values or attribute columns that give same information. Removing of redundant attributes is called as data reduction. Some approaches such as Principal Component Analysis (PCA) can be used for data reduction too [68]. By using PCA, redundant attributes are composed and attribute number decreases by providing new attributes, andat the end more meaningful and explanatory attributes can be obtained. Otherwise overfitting [69] problem is common in machine learning techniques.

In numeric scale, attribute data should be discretized before analysis to obtain more meaningful analysis results. Some machine learning classification approaches such as C4.5 decision tree does not accept a numeric scaled attribute as class attribute for classification analysis. There are several techniques used for discretization such as equal-width or equal-frequency [68] in Weka. Applying clustering before discretization is another way to determine discretized bin number. Because of these reasons data cleaning and preprocessing phase is important for machine learning techniques.

Since in this study we needed only data in “Defect” category for prediction model, the issues recorded as “Change” were removed from data set.

All defects detected during test activities are recorded to issue tracking tool although all defects detected during review activities are not stored in tool. Therefore, the detected defects except during test activities were removed from data set. Only defect data detected during tests was taken into account after data cleaning.

We had to discretize defect open duration attribute to set as class attribute in machine learning classification. We discretized this attribute by using equal-width method before (i.e. 0-5, 5-10, 10-15). Before discretization operation, we clustered open duration data by using K-Means with Euclidean distance technique to display how many clusters would be better to contain. Screenshots of analysis views are provided in Appendix-C and Appendix-E.

3.5 Clustering According to Process EnactmentData Approach

In machine learning if the user has no idea about data set, s/he should use unsupervised methods for grouping of data. Since s/he does not know which attribute can be considered as independent variable to set as class attribute. One example of unsupervised methods is clustering. In clustering method, the user do not have to set an attribute as class attribute.

In this study the process enactment data was examined in Weka and by clustering, similar process attribute columns were removed. With Weka tool the row data regarding process executions that had same process attributes was separated in different clusters.SimpleKMeans approach was used and the difference between process executions was obtained. According to cluster number automatically given by Weka, the separate Excel sheets were manually prepared for each cluster. Clustering according to process enactment approach was applied only in case studies 1B and 2B since they were the only case studies that contain process enactment data for analysis. Screenshots of analysis views are provided in Appendix-C and Appendix-E.

3.6 Analysis

When evaluated with the presence of high volume data stored in software engineering tools, it has been observed that data mining applications over the software data are being increased especially in recent years [12].

Machine learning classification approaches are utilized for the purposes of generating prediction models. Mostly used techniques are Bayesian Belief Networks (BBN), Multilayer Perceptron, Logistics Regression and Decision Trees. Despite the fact that there are a lot number of divergent studies related with using machine learning techniques for building prediction models, there is not any model technique defined as the best prediction approach or any way to apply in sequential manner described as the best method. Therefore, the studies in literature can be successful only by comparing their selected techniques among themselves and assuming the one that has the most accurate results as the best model.

Weka gives performance evaluation values for model validation. In addition, there are other validation methods such as using cross-validation or separating the data into training and test data sets [68].

In this study we chose defect open duration metric as dependent variable for classification analysis since this metric was directly related with defect management process and product quality status.

Bayesnet, Multilayer Perceptron, Logistic and C4.5 Tree machine learning analysis approaches [68] were performed by keeping defect open duration metric as class attribute (dependent variable). By selecting these approaches for analysis, we paid attention to apply machine learning techniques from different categories.

CHAPTER4

CASE STUDIES

4.1 Case Study 1 (Project-1 Data)

Case Study 1A was conducted with the data of Project-1 (for the characteristics of Project-1 please refer to Section 3.3). In this case study firstly, only defect and product data were used for analysis. After case study 1A had been completed, we performed case study 1B with applying same analysis approaches but this time we used both defect and product data, and process enactmentdata of Project-1.

4.1.1 Case Study 1A (Project-1)

GQM Tree was prepared as shown in Table 4.1 after the data fields which the basic metrics were tracing to our goal in issue tracking tool database had been examined. The metric descriptions are provided in Table 4.2.

Table 4.1 GQM for Case Study 1A

GOAL	QUESTION NO	QUESTION	ANALYSIS METHOD	DERIVED METRIC	BASIC METRIC NO	BASIC METRIC
To understand if there is effect of enactment context on software product defectiveness.	**4.1**	What is software product defectiveness prediction accuracy without using process enactment data?	Bayesnet, Logistic, C4.5 Tree, Multilayer Perceptron Machine Learning Techniques	Defect Data: open duration (closed date-created date)	**4.1.1**	**Defect and Product Data:** source component, closed date, created date, detected test type, product version, product SLOC, product complexity, reproducibility, project phase

Table 4.2 Defectand Product Related Metric Descriptions for Case Study 1A

Metrics	Metric Description	Measurement Scale
Remaining Open Duration	The time starting with the creation of the defect and finishing with the closure of the defect. Calculated by the difference of defect closed date and defect created date. Unit is number of days.	Absolute
Source Component	The component name of the defect detected. Component name can be component-A, component-B, component-C, component-D or component-E. Filled by the issue tracking tool automatically when the tester record the defect.	Nominal
Created Date	The date when the defect is detected. Filled by the issue tracking tool automatically when the tester record the defect.	Interval
Closed Date	The date when the defect is closed. Filled by the issue tracking tool automatically when the project manager change the status of the defect as "Closed".	Interval
Test Type	The name of test type during which the defect is detected. Entered by tester to the issue tracking tool.	Nominal
Product Version	The version of the software product which the defect is detected. Entered by tester to the issue tracking tool.	Ordinal
SLOC (Source Lines of Code)	The size of the product version where the defect is detected. Collected from configuration tool by using Locmetrics tool.	Absolute
Complexity	The McCabe complexity of the product version where the defect is detected. Collected from configuration tool by using Locmetrics tool.	Absolute
Reproducibility	The repeatability of the defect detected. Entered by tester to the issue tracking tool.	Nominal
Project Phase	The project phase where the defect detected. Collected manually by domain expert.	Nominal

We filled MUQ shown in Figure 3.3 and 3.4 for basic and derived metrics (filled questionnaires are provided in Appendix-B). Afterrating results, we had idea about the usability of the metric. According to MUQ results, all basic metrics and derived metrics of Project-1 were classified as “partially usable”. Since MUA-1 is N, MUA-2 and MUA-3 are F, and MUA-4 is P.

Detected project phase data manually collected by using project’s archival data such as project meeting minutes, and expert opinions. Source component, closed date, created date, test type, product version and reproducibility metrics’ data had already been stored in issue tracking tool. These data directly extracted from tool database. Source lines of code (SLOC) and complexity metrics’ data are calculated by LocMetrics and manually entered into spreadsheet that includes defect data. Open duration metric data was calculated in the one column of the spreadsheet. All defect and product data were recorded in an Excel file (Appendix-B).

Data Excel file converted to .csv file format to be analyzed in Weka.

Open duration attribute had to be discretized, in other words the continuous scale of this attribute had to be transformed to discrete scale to identify as class attribute (classifier). Before discretization operation, we clustered open duration data with K-Means technique to display how many clusters it contains. After trials with 3, 4, 5 and 6 number of clusters, we observed that the 5-clustered data set denotes the most frequency equivalent within clusters than others. Therefore, we discretized open duration data to five equal-width clusters as “0-27”, “27-54”, “54-81”, “81-108”, and “108-135” days. Screen views of the operation implemented in Weka are provided in Appendix-B.

After we transformed class attribute to nominal scale by discretization, we used Weka classification techniques by choosing defect open duration attribute as dependent attribute (class attribute). We applied Multilayer Perceptron, Bayesian Belief Networks, Logistic Regression and C4.5 Decision Tree (J48) machine learning techniques. We used 10-folds validation technique because of its high accurateness rate. Screen views of the operation implemented in Weka are provided in Appendix-B.

Findings from the study:

We observed that 296 data points are sufficient to obtain confident prediction results. Since Project-1 is newly completed and all personnel who had developed the project software still exist in company, expert opinions increased the reliability of the data and results.

Correctly classification performance values of the generated models are given below. The other performance values of the models are provided in Appendix-B. Multilayer perceptron gave the best performance values compared with other machine learning approaches.

- Multilayer perceptron machine learning technique validated with 10-folds gives 95% correctly classified instances value.
- Bayesian networks machine learning technique validated with 10-folds gives 85% correctly classified instances value.
- Logistic machine learning technique validated with 10-folds gives 82% correctly classified instances value.
- J48 decision tree machine learning technique validated with 10-folds gives 92% correctly classified instances value.

To complete this case study, we spent 5 person-days. The effort includes applying the approach, performing the analyses, and interpreting the results. If the product size and complexity metrics had previously been collected in the same Excel sheet with defect data and project phase metric had been recorded in real time during creating defect in issue tracking tool, spent effort for this case study could have been lower than now. The complete set of Weka outputs are provided in Appendix-B.

4.1.2 Case Study 1B (Project-1)

GQM Tree was prepared shown in Table 4.3.

Table 4.3 GQM for Case Study 1B

GOAL	QUESTION NO	QUESTION	ANALYSIS METHOD	DERIVED METRIC	BASIC METRIC NO	BASIC METRIC
To understand if there is effect of process enactment on software product defectiveness.	**4.1**	What is software product defectiveness prediction accuracy with using process enactment data?	Bayesnet, Logistic, C4.5 Tree, Multilayer Perceptron Machine Learning Techniques	Defect Data: open duration (closed date-created date)	**4.1.2**	**Defect and Product Data:** source component, closed date, created date, detected test type, product version, product SLOC, product complexity, reproducibility, project phase **Process Enactment Data:** defect management process attributes

We filled out PER to identify all alternative process attributes of the process executions (shown in Figure 3.5). PER form was filled by interviewing with Configuration Management Responsible personnel of the company. Defect management process is performed by issue tracking tool and with the monitoring and control of Configuration Management Responsible personnel in company.

Process Execution Record
(Internal Attributes)

Process Name:	Issue Management	Recorded On:	26.03.2012
Process Execution No:	N/A	Recorded By:	Damla Sivrioğlu

1. **Inputs:** Please list the inputs to the process execution.

No	Name	Description
1	Defects	
2	Change requests	

2. **Outputs:** Please list the outputs from the process execution.

No	Name	Description
1	Updated product version	

3. **Activities:** Please list in sequence the activities that were performed while executing the process.

No	Name	Description
1	Assign defect	
2	Defect resolution (Fix issues)	
3	Not verified for second time	
4	Defect verification	
5	Close defect	

4. **Roles:** Please list the roles that were allocated responsibilities in process execution.

No	Name	Description
1	Project Manager	Track issues, Fix issues
2	Configuration Manager	Track issues
3	Developer	Fix issues
4	Modelling and Graphics Designer	Fix issues
5	Tester	Open issues

5. **Tools and Techniques:** Please list the tools and techniques that are used to support process execution.

No	Name	Description
1	Redmine	Issue tracking tool
2	Excel	Version Description List is in Excel format.
3	SVN	Configuration management tool

Figure 4.1 PER for Case Study 1B

After completing PER form, same process attributes were entered into PSM columns and process execution values were filled in PSM shown in Figure 3.7 for each defect. Process attributes were given with abbreviations starting with "dm" phrase, which means "defect management", in PSM in order to ease reading of data file when opened in Weka. Because of place constraint, only 21 of the 296 data points could be shown in Figure 4.2.

		Process Attributes															
		1 Inputs		2 Outputs	3 Activities					4 Roles					5 Tools and Techniques		
		1.1 <Input 1>	1.2 <Input 2>	2.1 <Output 1>	3.1 <Activity 1>	3.2 <Activity 2>	3.3 <Activity 3>	3.4 <Activity 4>	3.5 <Activity 5>	4.1 <Role 1>	4.2 <Role 2>	4.3 <Role 3>	4.4 <Role 4>	4.5 <Role 5>	5.1 <Tools and Techniques 1>	5.2 <Tools and Techniques 2>	5.3 <Tools and Techniques 3>
Process Executions	Defect No	dmI1	dmI2	dmO1	dmA1	dmA2	dmA3	dmA4	dmA5	dmR1	dmR2	dmR3	dmR4	dmR5	dmT1	dmT2	dmT3
PE1	1	1	0	1	1	1	0	1	1	1	1	1	0	1	1	1	1
PE2	2	1	0	0	1	1	0	0	1	1	0	1	0	1	1	1	1
PE3	3	1	0	1	1	1	0	1	1	1	1	1	0	1	1	1	1
PE4	4	1	0	1	1	1	0	1	1	1	1	1	0	1	1	1	1
PE5	5	1	0	1	1	1	1	1	1	1	1	1	0	1	1	1	1
PE6	6	1	0	1	1	1	0	1	1	1	1	1	0	1	1	1	1
PE7	7	1	0	1	1	1	0	1	1	1	1	1	0	1	1	1	1
PE8	8	1	0	1	1	1	0	1	1	1	1	1	0	1	1	1	1
PE9	9	1	0	1	1	1	1	1	1	1	1	1	0	1	1	1	1
PE10	10	1	0	1	1	1	0	1	1	1	1	1	0	1	1	1	1
PE11	11	1	0	1	1	1	0	1	1	1	1	1	0	1	1	1	1
PE12	12	1	0	0	1	0	0	0	1	1	0	1	0	1	1	1	1
PE13	13	1	0	1	1	1	0	1	1	1	1	1	0	1	1	1	1
PE14	14	1	0	1	1	1	0	1	1	1	1	1	0	1	1	1	1
PE15	15	1	0	1	1	1	0	1	1	1	1	1	0	1	1	1	1
PE16	16	1	0	0	1	0	0	0	1	1	0	1	0	1	1	1	1
PE17	17	1	0	1	1	1	0	1	1	1	1	1	0	1	1	1	1
PE18	18	1	0	1	1	1	0	1	1	1	1	1	0	1	1	1	1
PE19	19	1	0	1	1	1	0	1	1	1	1	1	0	1	1	1	1
PE20	20	1	0	1	1	1	0	1	1	1	1	1	0	1	1	1	1
PE21	21	1	0	1	1	1	0	1	1	1	1	1	0	1	1	1	1
PE22	22	1	0	1	1	1	0	1	1	1	1	1	0	1	1	1	1
....	23	1	0	0	1	0	0	0	1	1	0	1	0	1	1	1	1

Figure 4.2 PSM for Case Study 1B

To prevent multicollinearity during the analysis in Weka, we should remove redundant process attributes, if exists, from spreadsheet. When we examined PSM, we observed that dmA1, dmA5, dmR1, dmR3 and dmR5 had displayed same behaviors. In other words, assigning personnel and closing defect activities had been implemented for all 296 process executions, and project manager, developer and tester personnel had performed their roles in all 296 process executions. Since dmR3 had alone fulfilled the characteristics (differences among executions) of these process attributes, we kept only dmR3 from these sixattributes for the analysis. Additionally, dmI1 and dmI2 do not give any information for analysis. Since, they do not differ in values among executions. In other words, since process input called “defect” was only input category that we had taken into account for our study, we had ignored the process input execution data categorized as “change” request. Therefore, we do not include dmI2 for our analyses. After data cleaning, we had an Excel file that consists of dmO1, dmA2, dmA3, dmA4, dmR2, dmR3 and dmR4 process attributes described in Figure 4.1.

Table 4.4 Process Enactment Metric Descriptions for Case Study 1B

Metrics	Metric Description	Measurement Scale
dmO1	New software version is the output of defect management process.	Nominal
dmA2	Developer response is one of the activities of defect management process. It means that developer has resolved the defect.	Nominal
dmA3	Not verified is one of the activities of defect management process. It means that tester has tested resolved defect but can not verified for second time.	Nominal
dmA4	Defect verification is one of the activities of defect management process. It means that tester has tested resolved defect and verified.	Nominal
dmR2	Configuration manager personnel is one of the roles of defect management process. This personnel is responsible of configuration control of software product versions.	Nominal
dmR3	Developer personnel is one of the roles of defect management process. This personnel is responsible of develop software product and fix the defects.	Nominal
dmR4	Graphic designer is one of the roles of defect management process. This personnel is responsible of developing graphics of software product and fix the defects.	Nominal

We combined collected defect, product and process enactment data in an Excel file spreadsheet.

We used K-Means and Euclidean Distance clustering technique and clustered the data. We obtained seven clusters which were called as c0, c1, c2, c3, c4, c5 and c6 in the rest of the case study. The differences of clusters are provided in Table 4.5. Implemented clustering steps are provided in Appendix-C.

Table 4.5 Process Attributes Patterns for Case Study 1B Clusters

Cluster Name	c0						
	2 Outputs	3 Activities				4 Roles	
Process Attributes Pattern (PAP)	2.1 <Output 1> dmO1	3.2 <Activity 2> dmA2	3.3 <Activity 3> dmA3	3.4 <Activity 4> dmA4	4.2 <Role 2> dmR2	4.3 <Role 3> dmR3	4.4 <Role 4> dmR4
PAP1	1	1	0	1	1	1	0
PAP2	1	1	1	1	1	1	0
Cluster Name	c1						
	2 Outputs	3 Activities			4 Roles		
Process Attributes Pattern (PAP)	2.1 <Output 1> dmO1	3.2 <Activity 2> dmA2	3.3 <Activity 3> dmA3	3.4 <Activity 4> dmA4	4.2 <Role 2> dmR2	4.3 <Role 3> dmR3	4.4 <Role 4> dmR4
PAP1	1	1	0	1	1	0	1
PAP2	1	1	1	1	1	0	1
PAP3	1	1	1	1	0	0	1
PAP4	0	0	1	1	1	0	1
Cluster Name	c2						
	2 Outputs	3 Activities			4 Roles		
Process Attributes Pattern (PAP)	2.1 <Output 1> dmO1	3.2 <Activity 2> dmA2	3.3 <Activity 3> dmA3	3.4 <Activity 4> dmA4	4.2 <Role 2> dmR2	4.3 <Role 3> dmR3	4.4 <Role 4> dmR4
PAP1	1	1	0	1	0	1	0
Cluster Name	c3						
	2 Outputs	3 Activities			4 Roles		
Process Attributes Pattern (PAP)	2.1 <Output 1> dmO1	3.2 <Activity 2> dmA2	3.3 <Activity 3> dmA3	3.4 <Activity 4> dmA4	4.2 <Role 2> dmR2	4.3 <Role 3> dmR3	4.4 <Role 4> dmR4
PAP1	1	1	0	1	1	0	0
Cluster Name	c4						
	2 Outputs	3 Activities			4 Roles		
Process Attributes Pattern (PAP)	2.1 <Output 1> dmO1	3.2 <Activity 2> dmA2	3.3 <Activity 3> dmA3	3.4 <Activity 4> dmA4	4.2 <Role 2> dmR2	4.3 <Role 3> dmR3	4.4 <Role 4> dmR4
PAP1	1	1	1	1	1	0	0
Cluster Name	c5						
	2 Outputs	3 Activities			4 Roles		
Process Attributes Pattern (PAP)	2.1 <Output 1> dmO1	3.2 <Activity 2> dmA2	3.3 <Activity 3> dmA3	3.4 <Activity 4> dmA4	4.2 <Role 2> dmR2	4.3 <Role 3> dmR3	4.4 <Role 4> dmR4
PAP1	0	1	0	0	0	1	0
Cluster Name	c6						
	2 Outputs	3 Activities			4 Roles		
Process Attributes Pattern (PAP)	2.1 <Output 1> dmO1	3.2 <Activity 2> dmA2	3.3 <Activity 3> dmA3	3.4 <Activity 4> dmA4	4.2 <Role 2> dmR2	4.3 <Role 3> dmR3	4.4 <Role 4> dmR4
PAP1	0	0	0	0	0	1	0
PAP2	0	0	1	0	0	1	0

We separated data Excel sheet to clusters and prepared a separate .csv file for each cluster. At the end we obtained the files shown in Figure 4.3. Each of these files included defect, product and process enactment metrics of the related defects.

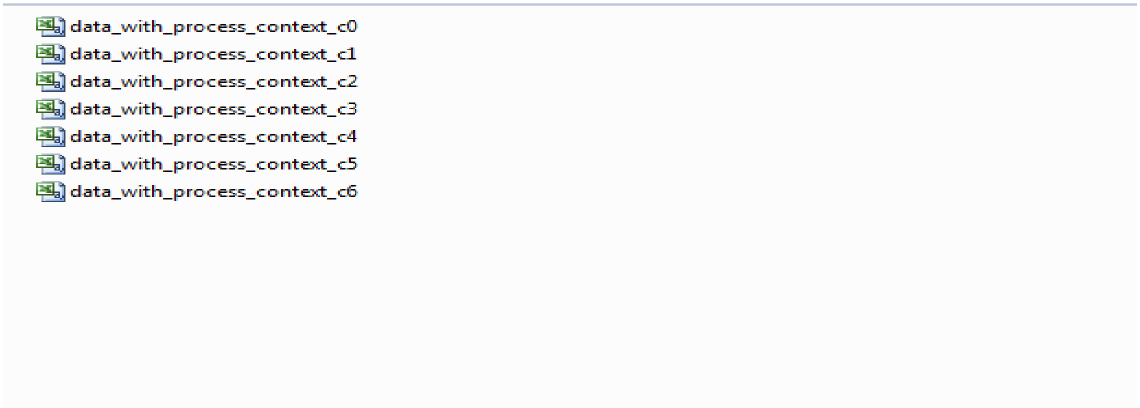

Figure 4.3 Clustered Metric Files for Case Study 1B

After clustering, we applied Multilayer Perceptron, Bayesian Belief Networks, Logistic Regression and C4.5 Decision Tree (J48) machine learning techniques for each clusterseparately. During these analyses, we identified open duration as class attribute. Screen views of the operation implemented in Weka are provided in Appendix-C.

Findings from the study:

We observed that the history data stored by issue tracking tool is beneficial to collect process enactment data. We collected process enactment data by firstly filling PER to identify process attributes. These process attributes can be identified easier by reviewing history data in tool database since all process activity alternatives are stored with their dates and the personnel who perform the activity. For example, when any personnel updates the defect status as “verified”, the tool constitutes a record that “Defect status was updated by <personnel name> on <date>.” in database. This process history data is used to fill out PSM for each defect record, in other words for each process execution.

Correctly classification performance values of the generated models for cluster-0 are given below. The other performance values of the models and the clusters are provided in Appendix-C. Bayesian networks gave the best performance values compared with other machine learning approaches.

- Multilayer perceptron machine learning technique validated with 10-folds gives 96% correctly classified instances value for cluster 0.
- Bayesian networks machine learning technique validated with 10-folds gives 97% correctly classified instances value for cluster 0.
- Logistic machine learning technique validated with 10-folds gives 95% correctly classified instances value for cluster 0.
- J48 decision tree machine learning technique validated with 10-folds gives 96% correctly classified instances value for cluster 0.

Since clusters 3, 4 and 5 include low number of data, we could not apply machine learning techniques to them. If the cluster number is decreased or we have more data points, this issue can be solved.

To complete this case study, we spent 10 person-days. The effort includes applying the approach, performing the analyses, and interpreting the results. If the process enactmentdata had previously been collected or the process history data could automatically be extracted by a query from issue tracking tool, spent effort for this case study could have been lower than now. In other words, the most important reason of high spent effort is that we have collected process enactment data by entering each of 296 defects in tool and recording the history data to Excel sheet. The complete set of Weka outputs are provided in Appendix-C.

4.1.3 Results Comparison for Case Study 1 (Project-1)

According to Table 4.5, the characteristics of clusters can be described as follows in terms of process attribute patterns;

- Cluster 0 includes the metrics of process executions through which an updated product version is obtained as output, defect resolution and defect verification activities are implemented, and configuration manager and developer perform their roles. But, modeling and graphics designer does not perform his role.

- Cluster 1 includes the metrics of process executions through which defect verification activity is implemented, and modeling and graphics designer perform his role. But, developer does not perform his role.
- Cluster 2 includes the metrics of process executions through which an updated product version is obtained as output, defect resolution and defect verification activities are implemented, and developer performs hisrole. But, configuration manager and modeling and graphics designer do not perform their roles.
- Cluster 3 includes the metrics of process executions through which an updated product version is obtained as output, defect resolution and defect verification activities are implemented, and configuration manager performs his role. But, developer and modeling and graphics designer do not perform their roles.
- Cluster 4 includes the metrics of process executions through which an updated product version is obtained as output, defect resolution, not verified for second time and defect verification activities are implemented, and configuration manager performs his role. But, developer and modeling and graphics designer do not perform their roles.
- Cluster 5 includes the metrics of process executions through which defect resolution activity is implemented, and developer performs his role. But, configuration manager and modeling and graphics designer do not perform their roles.
- Cluster 6 includes the metrics of process executions through which no activities documented in PER are implemented, and only developer performs his role. In only one of the 425 executions not verified for second time activity is implemented. It means that in one defect management process execution the defect in resolved status could not be verified during second test repetition by test specialist.

We observed that generally the analysis results ofclustered data sets with process enactmentare more accurate than data set without process enactmentas shown in Table 4.6.

Table 4.6 Results Comparison for Case Study 1

Number of instances (data points)	Data set	Method	Correctly Classified Instances	Incorrectly Classified Instances	Kappa statistic	Mean absolute error	Root mean squared error	Relative absolute error
112	Cluster 0 Data (With Process Enactment)	Multilayer Perceptron	96,43%	3,57%	94,86%	1,70%	10,37%	6,06%
		Bayesnet	97,32%	2,68%	96,16%	1,40%	10,45%	4,98%
		Logistic	94,64%	5,36%	92,28%	2,14%	14,64%	7,63%
		J48	95,54%	4,46%	93,55%	2,15%	11,47%	7,64%
71	Cluster 1 Data (With Process Enactment)	Multilayer Perceptron	84,51%	15,49%	79,06%	7,35%	24,39%	61,41%
		Bayesnet	80,28%	19,72%	73,61%	8,31%	27,57%	69,87%
		Logistic	81,69%	18,31%	75,58%	7,19%	26,46%	23,84%
		J48	85,92%	14,08%	80,95%	7,41%	21,94%	24,57%
70	Cluster 2 Data (With Process Enactment)	Multilayer Perceptron	95,71%	4,29%	92,13%	3,61%	14,75%	9,76%
		Bayesnet	91,43%	8,57%	83,48%	5,53%	21,94%	14,96%
		Logistic	90,00%	10,00%	81,04%	6,54%	25,37%	17,70%
		J48	82,86%	17,14%	64,87%	17,21%	31,64%	46,55%
26	Cluster 3 Data (With Process Enactment)	Multilayer Perceptron	N/A (all 26 are between 81-108)					
		Bayesnet	N/A (all 26 are between 81-108)					
		Logistic	N/A (all 26 are between 81-108)					
		J48	N/A (all 26 are between 81-108)					
5	Cluster 4 Data (With Process Enactment)	Multilayer Perceptron	N/A (only 5 data points)					
		Bayesnet	N/A (only 5 data points)					
		Logistic	N/A (only 5 data points)					
		J48	N/A (only 5 data points)					
1	Cluster 5 Data (With Process Enactment)	Multilayer Perceptron	N/A (only 1 data point 81-108)					
		Bayesnet	N/A (only 1 data point 81-108)					
		Logistic	N/A (only 1 data point 81-108)					
		J48	N/A (only 1 data point 81-108)					
11	Cluster 6 Data (With Process Enactment)	Multilayer Perceptron	100,00%	0,00%	100,00%	4,02%	7,04%	9,16%
		Bayesnet	100,00%	0,00%	100,00%	0,03%	0,06%	0,07%
		Logistic	100,00%	0,00%	100,00%	0,03%	0,10%	6,58%
		J48	100,00%	0,00%	100,00%	0,00%	0,00%	0,00%
296	Data Without Process Enactment	Multilayer Perceptron	94,93%	5,07%	93,38%	2,40%	13,14%	7,80%
		Bayesnet	85,14%	14,86%	80,54%	5,79%	20,81%	18,86%
		Logistic	82,43%	17,57%	76,90%	7,00%	26,16%	22,78%
		J48	91,55%	8,45%	88,87%	5,63%	17,03%	18,35%

The average of correctly classified intances values of the methods applied to cluster 0 data is 95,98%. On the other hand the average of correctly classified intances values of the methods applied to data without process enactment is 88,51%. The correctly classified rate is 7,47% higher in cluster 0 than the result of the data set that do not include process enactment. The average of root mean squared error values of the methods applied to cluster 0 data is 11,73%. On the other hand the average of root mean squared error values of the methods applied to data without process enactment is 19,29%. The root mean squared error is 7,55% lower in cluster 0 than the result of the data set that do not include process enactment.

The average of correctly classified intances values of the methods applied to cluster 1 data is 83,10%. The correctly classified rate is 5,41% lower in cluster 1 than the result of the data set that do not include process enactment. The average of root mean squared error values of the methods applied to cluster 1 data is 25,09%. The root mean squared error is 5,81% higher in cluster 1 than the result of the data set that do not include process enactment. We could not obtain promising results from this cluster, the reason of this is the noise in cluster patterns that is seen in Table 4.5. To avoid this noise and achieve more accurate prediciton for cluster 1, one more clustering operation can be performed within cluster 1 data.

The average of correctly classified intances values of the methods applied to cluster 2 data is 90,00%. The correctly classified rate is 1,49% higher in cluster 2 than the result of the data set that do not include process enactment. The average of root mean squared error values of the methods applied to cluster 2 data is 23,43%. The root mean squared error is 4,14% higher in cluster 2 than the result of the data set that do not include process enactment.Although, average correctly classified instances is high, we obtained a high average error value. The reason of this is the low error rate in J48 (C4.5) decision tree method, since this machine learning method needs more data point for a more accurate prediction than the other machine learning methods.

The average of correctly classified intances values of the methods applied to cluster 6 data is 100,00%. The correctly classified rate is 11,49% higher in cluster 6 than the result of the data set that do not include process enactment. The average of root mean squared error values of the methods applied to cluster 6 data is

1,80%. The root mean squared error is 17,49% lower in cluster 6 than the result of the data set that do not include process enactment.

4.2 Case Study 2 (Project-2 Data)

Case Study 2A conducted with the data of Project-2 (for the characteristics of Project-2 please refer to Section 3.3). In this case study firstly only defect and product data used for analysis. After case study 2A had been completed, we performed case study 2B with applying same analysis approaches but this time we used both defect and product data, and process enactmentdata of Project-2.

4.2.1 Case Study 2A (Project-2)

GQM Tree shown in Table 4.7 was prepared after the data fields that were basic metrics tracing to our goal in issue tracking tool database had been examined. The metric descriptions are provided in Table 4.8. As different from Case Study 1, we identified detected software configuration unit (SCU) metric for Case Study-2. Since Project-2 includes several SCUs in its developed software product, this data might give important information for the patterns in data set. Second difference from Case Study 1 is that we selected Decision Table technique rather than Multilayer Perceptron. And we included Simple Logistic Regression rather that Logistic Regression, since we again wanted to validate our proposed method for various machine learning techniques.

Table 4.7 GQM for Case Study 2A

GOAL	QUESTION NO	QUESTION	ANALYSIS METHOD	DERIVED METRIC	BASIC METRIC NO	BASIC METRIC
To understand if there is effect of process enactment on software product defectiveness.	**4.2**	What is software product defectiveness prediction accuracy without using process enactment data?	Bayesnet, SimpleLogistic, C4.5 Tree, Decision Table, Multilayer Perceptron Machine Learning Techniques	Defect Data: open duration (closed date-created date)	**4.2.1**	**Defect and Product Data:** detected SCU name, detected module name, closed date, created date, detected test type, product version, product SLOC, product complexity, reproducibility, project phase, source component

Table 4.8 Defect and Product Related Metric Descriptions for Case Study 2A

Metrics	Metric Description	Measurement Scale
Remaining Open Duration	The time starting with the creation of the defect and finishing with the closure of the defect. Calculated by the difference of defect closed date and defect created date. Unit is number of days.	Absolute
Detected SCU Name	The name of the software configuration unit (SCU) where the defect is detected. Entered by developer to the issue tracking tool.	Nominal
Created Date	The date when the defect is detected. Filled by the issue tracking tool automatically when the tester record the defect.	Interval
Closed Date	The date when the defect is closed. Filled by the issue tracking tool automatically when the project manager change the status of the defect as "Closed".	Interval
Test Type	The name of test type during which the defect is detected. Entered by tester to the issue tracking tool.	Nominal
Product Version	The version of the software product which the defect is detected. Entered by tester to the issue tracking tool.	Ordinal
SLOC (Source Lines of Code)	The size of the product version where the defect is detected. Collected from configuration tool by using Locmetrics tool.	Absolute
Complexity	The McCabe complexity of the product version where the defect is detected. Collected from configuration tool by using Locmetrics tool.	Absolute
Reproducibility	The repetability of the defect detected. Entered by tester to the issue tracking tool.	Nominal
Project Phase	The project phase where the defect detected. Collected manually by domain expert.	Nominal
Source Component	The component name of the defect detected. Component name can be BusinessManager, Form, GMManager, Report, DBManager, Table and Menu-Template. Manually collected by domain expert.	Nominal

We filled MUQ shown in Figure 3.3 and Figure 3.4 for basic and derived metrics (filled questionnaires are provided in Appendix-D). After obtained rating results, we had idea about the usability of the metric. According to MUQ results, all basic metrics and derived metric of Project-2 were classified as “partially usable”. Since MUA-1 is N, MUA-2 and MUA-3 are F, and MUA-4 is P.

Detected project phase and source component data manually collected by using project’s archival data such as project meeting minutes, and expert opinions. Detected module name, closed date, created date, test type, product version and reproducibility metrics’ data had already been stored in issue tracking tool. These data directly extracted from tool database. Source lines of code (SLOC) and complexity metrics’ data are calculated by LocMetrics and manually entered into spreadsheet that includes defect data. Open duration metric data was calculated in the one column of the spreadsheet. All defect and product data were recorded in an Excel file (Appendix-D).

Data Excel file was converted to .csv file format to be analyzed in Weka.

We discretized open duration data to seven equal-width clusters as "0-20", "20-40", "40-60", "60-80", "80-100", "100-120" and "120-140" days. Since, there were not any open duration value in "80-100" range, this cluster had no data.

We applied Decision Table, Bayesian Belief Networks, Simple Logistic Regression and C4.5 Decision Tree (J48) machine learning techniques by selecting open duration as class attribute. Screen views of the operation implemented in Weka are provided in Appendix-D.

Findings from the study:

We observed that 425 data points are sufficient to obtain confident prediction results. Since Project-2 had been completed a long time ago and several personnel who had developed the project software do not work for the company anymore, we believe that the reliability of the data collected by interviews might be lower than Case Study 1.

Correctly classification performance values of the generated models are given below. The other performance values of the models are provided in Appendix-D. Decision Table, Bayesian Networks, Simple Logistic and J48 Decision Tree gave the best performance values compared with other machine learning approaches.

- Decision Table machine learning technique validated with 10-folds gives 88% correctly classified instances value.
- Bayesian networks machine learning technique validated with 10-folds gives 89% correctly classified instances value.
- Simple Logistic machine learning technique validated with 10-folds gives 86% correctly classified instances value.

- J48 decision tree machine learning technique validated with 10-folds gives 89% correctly classified instances value.

To complete this case study, we spent 10 person-days. The effort includes applying the approach, performing the analyses, and interpreting the results. If the source component metric had previously been collected in the same Excel sheet with defect data and project phase metric had been recorded in real time during creating defect in issue tracking tool, spent effort for this case study could have been lower than now. The complete set of Weka outputs are provided in Appendix-D.

4.2.2 Case Study 2B (Project-2)

GQM Tree was prepared shown in Table 4.9.

Table 4.9 GQM for Case Study 2B

GOAL	QUESTION NO	QUESTION	ANALYSIS METHOD	DERIVED METRIC	BASIC METRIC NO	BASIC METRIC
To understand if there is effect of process enactment on software product defectiveness.	**4.2**	What is software product defectiveness prediction accuracy with using process enactment data?	Bayesnet, SimpleLogistic, C4.5 Tree, Decision Table, Multilayer Perceptron Machine Learning Techniques	Defect Data: open duration (closed date-created date)	**4.2.2**	**Defect and Product Data:** detected SCU name, closed date, created date, detected test type, product version, product SLOC, product complexity, reproducibility, project phase, source component **Process Enactment Data:** defect management process attributes

We filled out PER to identify all alternative process attributes of the process executions (shown in Figure 3.5). PER form was filled by interviewing with Project Manager personnel of the project (Figure 4.4).

Process Execution Record
(Internal Attributes)

Process Name:	Issue Management	Recorded On:	27.03.2012
Process Execution No:	N/A	Recorded By:	Damla Sivrioğlu

1. **Inputs:** Please list the inputs to the process execution.

No	Name	Description
1	Defects	
2	Change requests	Improvements

2. **Outputs:** Please list the outputs from the process execution.

No	Name	Description
1	Modified software	Target version

3. **Activities:** Please list in sequence the activities that were performed while executing the process.

No	Name	Description
1	Adding explanation to defect	
2	Requesting more feedback	From test specialist by developer
3	Defect resolution (Issue implementing)	
4	Defect rejection	Defect is rejected and is not resolved.
5	Defect have not been tried again	
6	Status changed as "Resolved" by test specialist	
7	"Subject" field of defect is changed	
8	Adding additional picture for explanation	

4. **Roles:** Please list the roles that were allocated responsibilities in process execution.

No	Name	Description
1	Project Manager	Tracks issues
2	Configuration Responsible	Tracks issues
3	Developer	Implements issues
4	Test specialist	Opens issues

5. **Tools and Techniques:** Please list the tools and techniques that are used to support process execution.

No	Name	Description
1	Redmine	Issue tracking tool
2	SVN	Configuration management tool
3	Visual Studio	Development environment

Figure 4.4 PER for Case Study 2B

After completing PER form, same process attributes were entered into PSM columns and process execution values were filled in PSM shown in Figure 3.7 for each defect. Process attributes were given with abbreviations starting with "dm" (defect management) phrase in PSM in order to ease reading of data file when opened in Weka. Because of place constraint, only 21 of the 425 data points could be shown in Figure 4.5.

		Process Attributes																	
		1 Inputs		2 Outputs	3 Activities								4 Roles				5 Tools and Techniques		
		1.1 <Input 1>	1.2 <Input 2>	2.1 <Output 1>	3.1 <Activity 1>	3.2 <Activity 2>	3.3 <Activity 3>	3.4 <Activity 4>	3.5 <Activity 5>	3.6 <Activity 6>	3.7 <Activity 7>	3.8 <Activity 8>	4.1 <Role 1>	4.2 <Role 2>	4.3 <Role 3>	4.4 <Role 4>	5.1 <Tools and Techniques 1>	5.2 <Tools and Techniques 2>	5.3 <Tools and Techniques 3>
Process Executions	Defect No	dmI1	dmI2	dmO1	dmA1	dmA2	dmA3	dmA4	dmA5	dmA6	dmA7	dmA8	dmR1	dmR2	dmR3	dmR4	dmT1	dmT2	dmT3
PE1	1	1	0	0	0	0	0	0	1	1	0	0	1	0	1	0	1	1	1
PE2	2	1	0	1	0	1	0	0	0	1	0	0	1	1	1	1	1	1	1
PE3	3	1	0	1	0	1	0	0	0	1	0	0	1	1	1	1	1	1	1
PE4	4	1	0	1	0	1	0	0	0	1	0	0	1	1	1	1	1	1	1
PE5	5	1	0	1	0	1	0	0	0	1	0	0	1	1	1	1	1	1	1
PE6	6	1	0	1	0	1	0	0	0	1	0	0	1	1	1	1	1	1	1
PE7	7	1	0	1	0	1	0	0	0	1	0	0	1	1	1	1	1	1	1
PE8	8	1	0	1	0	0	0	0	0	1	0	0	1	1	1	1	1	1	1
PE9	9	1	0	1	0	1	0	0	0	1	0	0	1	1	1	1	1	1	1
PE10	10	1	0	1	0	1	0	0	0	1	0	0	1	1	1	1	1	1	1
PE11	11	1	0	1	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1
PE12	12	1	0	1	0	0	0	0	0	1	0	0	1	1	0	1	1	1	1
PE13	13	1	0	1	0	0	0	0	0	1	0	0	1	1	0	1	1	1	1
PE14	14	1	0	1	0	0	0	0	0	1	0	0	1	1	0	1	1	1	1
PE15	15	1	0	1	0	0	0	0	0	1	1	0	1	1	0	1	1	1	1
PE16	16	1	0	1	0	0	0	0	0	1	0	0	1	1	0	1	1	1	1
PE17	17	1	0	1	0	0	0	0	0	1	0	0	1	1	0	1	1	1	1
PE18	18	1	0	1	0	0	0	0	0	1	0	0	1	1	0	1	1	1	1
PE19	19	1	0	1	0	0	0	0	0	1	0	0	1	1	0	1	1	1	1
PE20	20	1	0	1	0	0	1	0	0	0	0	0	1	1	0	1	1	1	1
PE21	21	1	0	1	0	0	0	0	0	0	1	0	1	1	0	1	1	1	1
PE22	22	1	0	1	0	1	0	0	0	1	0	0	1	1	1	1	1	1	1
…….	23	1	0	1	0	1	0	0	0	1	0	0	1	1	1	1	1	1	1

Figure 4.5 PSM for Case Study 2B

To prevent multicollinearity during the analysis in Weka, we should remove redundant process attributes, if exists, from spreadsheet. When we examined PSM, we observed that I1 and I2 are same in all 425 executions; we have removed them from analysis.Since O1, R2 and R4 showed same behavior for each defect and there was little difference when reviewed all rows, we removed them from analysis. Since A7 and A8 showed same behavior for each defect and these activities had low impact on defect management process, we removed them from analysis. We specify these activities as having low effect on independent variable, open duration metric, because changing “subject” field (a field to fill in issue tracking tool) in defect record or adding additional picture to defect record have no technical context on the quality of final product and they are executions that are rarely seen during defect management process of whole project. Since all row data were same in R1, in other words project manager has role in for all 296 process executions, it is redundant to include it in analysis. Therefore, we removed R1 process attribute from analysis. Since T1, T2, T3 were used in all defect management process executions, they did not give additional information about process change through defect management. Therefore, T1, T2 and T3 were removed from analysis because of being redundant. After data cleaning, we had an Excel file that consisted of dmA1, dmA2, dmA3, dmA4, dmA5, dmA6 and dmR3 process attributes described in Figure 4.10.

Table 4.10 Process Enactment Metric Descriptions for Case Study 2B

Metrics	Metric Description	Measurement Scale
dmA1	Adding explanation to defect is one of the activities of defect management process. It means that developer and/or tester fills "additional explanation" field to give more detailed information about the defect.	Nominal
dmA2	Requesting more feedback is one of the activities of defect management process. It means that developer needs more information about the defect before resolving it.	Nominal
dmA3	Defect resolution is one of the activities of defect management process. It means that developer has resolved the defect.	Nominal
dmA4	Defect rejection is one of the activities of defect management process. It means that the defect record is examined and decided that it is not a defect actually.	Nominal
dmA5	Defect have not been tried again is one of the activities of defect management process. It means that the defect recorded can not be repeated.	Nominal
dmA6	Status changed as “Resolved” by test specialist is one of the activities of defect management process. It means that test specialist has verified the resolution of the defect.	Nominal
dmR3	Developer personnel is one of the roles of defect management process. This personnel is responsible of develop software product and fix the defects.	Nominal

We combined collected defect, product and process enactmentdata in an Excel file spreadsheet (Appendix-E).

We used K-Means and Euclidean Distance clustering technique and separated the data into five clusters which were called as c0, c1, c2, c3 and c4 in the rest of the case study. The differences of clusters are provided in Table 4.11. Implemented clustering steps are provided in Appendix-E.

Table 4.11 Process Attributes Patterns for Case Study 2B Clusters

Cluster Name	c0						
	3 Activities						4 Roles
Process Attributes Pattern (PAP)	3.1 <Activity 1> dmA1	3.2 <Activity 2> dmA2	3.3 <Activity 3> dmA3	3.4 <Activity 4> dmA4	3.5 <Activity 5> dmA5	3.6 <Activity 6> dmA6	4.3 <Role 3> dmR3
PAP1	0	1	0	0	0	1	1
PAP2	0	1	0	0	0	0	1
PAP3	1	1	0	0	0	1	1
PAP4	1	1	0	0	0	0	1
PAP5	1	1	0	1	0	1	1
PAP6	1	1	1	0	0	0	1
PAP7	1	1	0	0	1	1	1
PAP8	1	1	0	0	1	0	1
PAP9	1	0	1	0	0	0	1
PAP10	0	0	0	0	0	1	1
PAP11	0	1	1	0	0	0	1
PAP12	0	1	0	0	1	1	1
PAP13	0	0	1	0	0	0	0
PAP14	0	0	0	0	0	1	0
PAP15	0	0	0	0	0	0	1

Cluster Name	c2						
	3 Activities						4 Roles
Process Attributes Pattern (PAP)	3.1 <Activity 1> dmA1	3.2 <Activity 2> dmA2	3.3 <Activity 3> dmA3	3.4 <Activity 4> dmA4	3.5 <Activity 5> dmA5	3.6 <Activity 6> dmA6	4.3 <Role 3> dmR3
PAP1	0	1	0	0	0	0	1
PAP2	0	1	0	0	0	1	1
PAP3	0	1	0	0	1	1	1
PAP4	0	1	1	0	0	1	1
PAP5	0	0	0	0	0	0	0
PAP6	0	1	0	0	0	1	0
PAP7	1	1	0	0	0	0	1
PAP8	1	1	0	0	0	1	1
PAP9	1	0	0	0	0	0	1
PAP10	1	1	1	0	0	1	1
PAP11	1	1	0	1	0	1	1
PAP12	1	0	0	0	0	0	0

Cluster Name	c1						
	3 Activities						4 Roles
Process Attributes Pattern (PAP)	3.1 <Activity 1> dmA1	3.2 <Activity 2> dmA2	3.3 <Activity 3> dmA3	3.4 <Activity 4> dmA4	3.5 <Activity 5> dmA5	3.6 <Activity 6> dmA6	4.3 <Role 3> dmR3
PAP1	0	0	0	0	0	0	0
PAP2	0	0	0	0	0	0	1
PAP3	0	0	0	1	0	0	1
PAP4	0	0	0	0	0	1	0
PAP5	0	0	0	1	0	1	1
PAP6	1	0	0	1	0	0	1
PAP7	1	1	0	1	0	1	1
PAP8	1	0	0	0	0	0	1

Cluster Name	c3						
	3 Activities						4 Roles
Process Attributes Pattern (PAP)	3.1 <Activity 1> dmA1	3.2 <Activity 2> dmA2	3.3 <Activity 3> dmA3	3.4 <Activity 4> dmA4	3.5 <Activity 5> dmA5	3.6 <Activity 6> dmA6	4.3 <Role 3> dmR3
PAP1	0	0	1	0	0	0	1
PAP2	0	0	0	1	0	0	1
PAP3	1	0	0	1	0	0	1
PAP4	0	1	1	0	0	0	1
PAP5	0	1	0	0	1	1	1
PAP6	1	0	1	0	0	0	1
PAP7	0	0	1	0	0	0	0
PAP8	0	0	0	1	0	1	1
PAP9	1	0	1	1	0	0	1

Cluster Name	c4						
	3 Activities						4 Roles
Process Attributes Pattern (PAP)	3.1 <Activity 1> dmA1	3.2 <Activity 2> dmA2	3.3 <Activity 3> dmA3	3.4 <Activity 4> dmA4	3.5 <Activity 5> dmA5	3.6 <Activity 6> dmA6	4.3 <Role 3> dmR3
PAP1	1	1	0	0	0	1	1
PAP2	0	1	0	0	0	1	1
PAP3	0	1	0	0	0	1	0
PAP4	1	1	0	0	0	0	1
PAP5	0	0	0	0	1	1	1
PAP6	0	1	1	0	0	1	1
PAP7	0	1	0	0	1	1	1
PAP8	0	0	0	0	0	0	0
PAP9	0	0	0	0	0	1	0
PAP10	0	1	0	0	0	0	1
PAP11	0	1	0	1	0	0	1
PAP12	1	1	0	1	0	1	1
PAP13	0	1	1	0	0	0	1

After preparing five Excel files for clustered data sets, we applied Decision Table, Bayesian Belief Networks, Simple Logistic Regressionand C4.5 Decision Tree (J48) machine learning techniques. Screen views of the operation implemented in Weka are provided in Appendix-E.

Findings from the study:

We mentioned that Project-2 had been completed more previously than Project-1, and therefore collecting process enactment data was harder than the first project. Additionally, we could interview with lower number of personnel who developed project's software product. Besides, this project is an old project, and executed processes are so changeable. This is observed with the variety of the process attribute patterns provided in Table 4.11. It is seen that the clusters are more noisy than the ones of project one.

Correctly classification performance values of the generated models for cluster-0 are given below. The other performance values of the models and the clusters are provided in Appendix-E. Decision Table, Bayesian Networks, Simple Logistic and J48 Decision Tree were applied and Bayesian Networks gave the best performance values compared with other machine learning approaches. 10-folds technique was used for validation.

- Decision Table machine learning technique validated with 10-folds gives 95% correctly classifies instances value for cluster 0.
- Bayesian Networks machine learning technique validated with 10-folds gives 97% correctly classifies instances value for cluster 0.

- Simple Logistic machine learning technique validated with 72% of all data points allocated for training data set gives 96% correctly classifies instances value for cluster 0.
- J48 Decision Tree machine learning technique validated with 10-folds gives 94% correctly classifies instances value for cluster 0.

To complete this case study, we spent 10 person-days. The effort includes applying the approach, performing the analyses, and interpreting the results. If the process enactmentdata had previously been collected or the process history data could automatically be extracted by a query from issue tracking tool, spent effort for this case study could have been lower than now. In other words, the most important reason of high spent effort is that we have collected process enactmentdata by entering each of 425 defects in tool and recording the history data to Excel sheet. The complete set of Weka outputs are provided in Appendix-E.

4.2.3 Results Comparison for Case Study 2 (Project-2)

According to Table 4.11, the characteristics of clusters can be described as follow in terms of process attribute patterns;

- Cluster 0 predominantly includes the metrics of process executions through which status changed as "Resolved" by test specialist activity is implemented, and developer performs hisrole. But, adding explanation to defect, defect rejection and not tried again activities are not implemented.Requesting more feedback activity is seen in the 13% some of the executions.
- Cluster 1 predominantly includes the metrics of process executions through which defect rejection is implemented, and developer performs his role. But, adding explanation to defect, requesting more feedback, defect resolution, not tried again activities, status changed as "Resolved" by test specialist are predominantly not implemented.
- Cluster 2 predominantly includes the metrics of process executions through which requesting more feedback and status changed as "Resolved" by test specialist activities are implemented, and developer performs his role. But,

adding explanation to defect, defect rejection and not tried again activities are not implemented.

- Cluster 3 predominantly includes the metrics of process executions through which defect resolution activity are implemented, and developer performs his role except 25% executions.
- Cluster 4 predominantly includes the metrics of process executions through which requesting more feedback and status changed as “Resolved” by test specialist activities are implemented, and developer performs his role. But, adding explanation to defect activity is implemented in 18% of executions.

We observed that generally the analysis results of clustered data sets with process enactmentare more accurate than data set without process enactmentshown in Table 4.12. However, we can not say the same thing for cluster 4. Although it has the highest number of data points, its performance values are lower than the analysis without process enactment data.

Table 4.12 Results Comparison for Case Study 2

Number of instances (data points)	Data set	Method	Correctly Classified Instances	Incorrectly Classified Instances	Kappa statistic	Mean absolute error	Root mean squared error	Relative absolute error
121	Cluster 0 Data (With Process Enactment)	Decision Table	95,04%	4,96%	85,03%	8,92%	15,71%	69,02%
		Bayesnet	96,69%	3,31%	90,35%	1,18%	10,37%	9,10%
		Simple Logistic	95,87%	4,13%	87,71%	5,90%	12,46%	45,62%
		J48	94,21%	5,79%	82,01%	2,96%	13,04%	22,88%
26	Cluster 1 Data (With Process Enactment)	Decision Table	88,46%	11,54%	0,00%	11,35%	20,26%	99,55%
		Bayesnet	96,15%	3,85%	78,33%	1,93%	11,53%	16,95%
		Simple Logistic	96,15%	3,85%	78,33%	5,67%	13,07%	49,77%
		J48	88,46%	11,54%	0,00%	5,20%	18,98%	45,58%
109	Cluster 2 Data (With Process Enactment)	Decision Table	92,66%	7,34%	80,78%	11,99%	19,54%	91,13%
		Bayesnet	91,74%	8,26%	79,28%	2,98%	15,78%	22,61%
		Simple Logistic	90,83%	9,17%	74,16%	3,59%	14,54%	27,27%
		J48	93,58%	6,42%	82,82%	3,29%	13,42%	25,01%

32	**Cluster 3 Data (With Process Enactment)**	**Decision Table**	87,50%	12,50%	66,84%	13,07%	21,61%	81,67%
		Bayesnet	93,75%	6,25%	83,42%	3,24%	15,70%	20,27%
		SimpleLogistic	81,25%	18,75%	53,62%	6,62%	23,53%	41,39%
		J48	78,13%	21,88%	47,66%	8,30%	24,48%	51,86%
137	**Cluster 4 Data (With Process Enactment)**	**Decision Table**	75,18%	24,82%	52,97%	14,22%	24,42%	77,89%
		Bayesnet	72,26%	27,74%	48,45%	10,13%	27,61%	55,47%
		SimpleLogistic	70,80%	29,20%	43,91%	12,42%	27,70%	68,03%
		J48	70,80%	29,20%	41,36%	14,13%	26,98%	77,38%
425	**Data Without Process Enactment**	**Decision Table**	88,47%	11,53%	83,26%	10,52%	18,92%	46,34%
		Bayesnet	88,94%	11,06%	84,00%	4,02%	18,36%	17,71%
		SimpleLogistic	88,24%	11,76%	82,70%	4,64%	17,95%	20,42%
		J48	88,94%	11,06%	83,03%	4,80%	16,66%	21,12%

The average of correctly classified intances values of the methods applied to cluster 0 data is 95,45%. On the other hand the average of correctly classified intances values of the methods applied to data without process enactment is 88,65%. The correctly classified rate is 6,08% higher in cluster 0 than the result of the data set that do not include process enactment. The average of root mean squared error values of the methods applied to cluster 0 data is 12,90%. On the other hand the average of root mean squared error values of the methods applied to data without process enactment is 17,97%. The correctly classified rate is 5,08% lower in cluster 0 than the result of the data set that do not include process enactment.

The average of correctly classified intances values of the methods applied to cluster 1 data is 92,31%. The correctly classified rate is 3,66% higher in cluster 1 than the result of the data set that do not include process enactment. The average of root mean squared error values of the methods applied to cluster 1 data is 15,96%. The correctly classified rate is 2,01% lower in cluster 1 than the result of the data set that do not include process enactment.

The average of correctly classified intances values of the methods applied to cluster 2 data is 92,20%. The correctly classified rate is 3,56% higher in cluster

2than the result of the data set that do not include process enactment. The average of root mean squared error values of the methods applied to cluster 2 data is 15,82%. The correctly classified rate is 2,15% lower in cluster 2 than the result of the data set that do not include process enactment.

The average of correctly classified intances values of the methods applied to cluster 3 data is 85,16%. The correctly classified rate is 3,49% lower in cluster 3 than the result of the data set that do not include process enactment. The average of root mean squared error values of the methods applied to cluster 3 data is 21,33%. The correctly classified rate is 3,36% higher in cluster 3 than the result of the data set that do not include process enactment. We could not obtain promising results from this cluster. To investigate the reason of this we reviewed data and observed that the cluster noise based on between process attibute patterns and defect open duration metric are high in cluster 3, although the number of data points is low.

The average of correctly classified intances values of the methods applied to cluster 4 data is 72,26%. The correctly classified rate is 16,39% lower in cluster 4 than the result of the data set that do not include process enactment. The average of root mean squared error values of the methods applied to cluster 4 data is 26,68%. The correctly classified rate is 8,71% higher in cluster 4 than the result of the data set that do not include process enactment. We could not obtain promising results from this cluster. To investigate the reason of this we reviewed data and observed that the cluster noise based on between process attibute patterns and defect open duration metric are high in cluster 4.

CHAPTER 5

CONCLUSION AND FUTURE WORK

Defect data gives information related to the software quality. The accessibility to defect data is easy in most cases, since a detailed view of detected defect is recorded to issue tracking tools and all thedata is stored from the initiation of the project to the end of maintenance phase. When the defect data is analyzed by researchers, the understanding of the product environment and process execution is provided.

Quality models such as CMMI enforces in Level 5 that defect prevention is vital for mature process and product. When the cost effectiveness is considered, achieving defect prevention for emergent enterprises is as beneficial as for the institutional ones. One of the activities used for defect prevention is defect data analysis or defect prediction. In order to point out the usable techniques for the understanding of product defectiveness and the factors that have impact on it, we applied various statistical and machine learning analysis methods to our data in our first study. By doing this, we collected defect related and product related metrics in different data sets. At the end, we presented our inferences in three categories based on their confidence [2].

We aimed to understand the effect of process enactment on product defectiveness prediction. After literature search, we decided to use machine learning algorithms for prediction, since these algorithms are suitable for recognizing the patterns in process enactment data. In this context, we performed case studies by using two different software projects. Before conducting case studies, we needed a method in order to systematically plan and analyze case studies.Therefore, we developed a method shown in Figure 3.1. By this method, we achieved the collection of process enactment data, data preprocessing and machine learning analysis.

The method applied in this thesisuses GQM, MUQ, PER, PSM, clustering and classification approaches. Goal-Question-Metric was used to determine the metrics that should be collected. Metric Usability Questionnaires were used to determine usable metrics data. Process Execution Record and Process Similarity Matrix were used to capture process traces and collect process enactment data. Attribute discretization and data reduction were performed in data cleaning and preprocessing phase of the case studies.

To validate the method, we performed four case studies which are conducted on the data of completed two software projects in a small company. In the first case study (case study-1A), product size metrics and defect related metrics data of Project-1 was classified with machine learning approaches. In second case study (case study-1B), same metrics in Project-1 were combined with defect management process enactment attributes and machine learning approaches were repeated. After case-study-1, we observed that the performance values of prediction models with process enactment data are better than the ones without process enactment data. The implementation of case study-1 was repeated with the data of Project-2 in case study-2. We observed similar results in case study 2 with case study 1 except a roughness. The roughness is that two of the clustered data sets with process enactment data gave lower performance values than the analysis results of the data set without process enactment data.

Defect open duration metric was the classifier for all case studies. In other words, it was identified as dependent variable for prediction models. Clustering was applied only in case study 1B and 2B. Clustered defect data was split to separate data sets.

In case study 1 (when compared case studies 1A and 1B) we observed that the data clustered according to process enactmentpatterns gives approximately3% more accurate results when the cluster has a low number of noisy process patterns (low number of pattern difference) and has sufficient data points to apply machine learning methods. The correctly classified instance values that are the performance evaluation value in machine learning approachesare ranging from -10 to 17%.

In case study 2 (when compared case studies 2A and 2B) we observed that the data clustered according to process enactment patterns gives 3% more accurateresults in terms of defect open duration metric (ranging from -7% to 8%) when the cluster has a low number of noisy process patterns.The cause of this high noise is implementation of different activities during process execution. Since the project-2 data is so old that the development processes applied might have not been stablein organization in these days. To decrease the noise several more clustering operations can be performed.

Another reason of the inconsistent result in case study-2 is that the data used for case study-2 might be retrospective, although the project in case study 1 is a newly completed one. This circumstance causes to collect unreliable data especially for process enactment in case study 2. The MUQs were filled via interviews with current data providers. However,since the providers of data have changed for Project-2 (most of the staffproviding data for the project does not work in the company anymore), theanswers to the questions in MUQ might not have reflected the realsituation for already stored data. Therefore a new part questioning thecharacteristics of the providers of data under evaluation might be good toadd to the MUQ.

While conducting case studies, we paid attention to take help from process experts by interviewing. But since several personnel of Project-2 were not working for the company anymore, we had to fill PER with the experts who knew only the second half of the development phase.On the other hand, for Project-1 we could easily collect data by using suggested assets.

We can say that GQM provides a systematic way to determine the data that will be collected and the analysis methods. MUQ provides to obtain more accurate results by using more accurate data. PER and PSM provides to collect process enactment attributes. Especially the newly proposed usage ways for these assets provides more practical solutions to collect process enactment data.Aside from interviewing approach to fill PER, the usage of the historical process data in issue tracking tool was advantageous to fill PSM during or after process executions. Multilayer Perceptron and Bayesian Networks methods gave more accurate results than the other applied machine learning techniques in this study.

In conclusion, multiple case study implementations showed us that our method can be used if we access reliable PER data in emergent organizations. Our first question was“Is process enactmentdata beneficial for defect prediction?”. To answer this, we assessed case study 1 and case study 2 results and the answer is yes. Thesecond question was “How can we use process enactmentdata?”. For this question we applied several assets called PER, PSM [16] and clustering in Weka. The third question was “Which approaches or analysis methods can our method support?”,and we explained the approaches applied in Section 3 in detail.

When we think of cost of quality [70], performing defect prediction approach costs 10 person-days for a project that shows similar features with the project of the case study 1B that has 296 defects detected. After applying the generated prediction models in new projects, we can calculate the decrease in defect management costs. Therefore, our proposed methodcan support and might be beneficial for the quality system of the organization.

We suggest using process enactmentpatterns for defect prediction operation and also we recommend methods to extract process enactmentdata. In other words, regardless of the analysis method applied, defect and product data must be tracked and assessed with its context to understand the product quality and process performance in turn.Since machine learning is a pattern oriented domain area, process enactmentdata is very convenient for pattern recognition.However, more studies should be performed for more evidence as a future work. Besides, we suggest coding a script to automatically extract historical process data from issue tracking tool, since manual collection of process enactment data for each defect management execution is costly.

The prediction model of defect open duration generated with the proposed way provides a basis for the estimation of the open period of a defectthat has been detected in software. If the distribution of the defects isdisplayed, the trend of open duration for detected defects can beestimated within a project. However, this assumption is not verified inthis thesis and might be subject to future work.

The factors that have impact on software product defectiveness can be considered in two categories:Environmental factors and internal process execution. The process enactmentdata which we have gathered for this study contains only inner processes. However, there are some outer factors, such as environmental impacts like personnel skills that affect the results. These outer process factors can be investigated and different collection methods might be discovered for the data. Additionally one more idea for future work is using of classification results to improve processes in organization. By observing the process patterns which give lower open duration values in PSM, organizational processes can be updated according to realized process attributes patterns that show better performance.

REFERENCES

[1] CMMI Product Team, “CMMI for Development, Version 1.3, Technical Report, SEI, 2010.

[2] Sivrioğlu, D. and Tarhan A., “Yazılım Modül Özelliklerine Göre Hatalılık Analizi: Bir Durum Çalışması”, UYMS Bildiri Kitabı, February 2012.

[3] Sivrioğlu, D., Tarhan A. and Demirörs O., “An Analysis of Product Defectiveness and Affecting Factors In A Small Software Organization”, under review by Euromicro Conference on Software Engineering and Advanced Applications (SEAA) Committee, April 2012.

[4] Jalote, P. and Agrawal, N., “Using Defect Analysis Feedback For Improving Quality and Productivity in Iterative Software Development”, Information and Communications Technology, Enabling Technologies for the New Knowledge Society: ITI 3rd International Conference, p. 703 – 713, 2005.

[5] Chrissis, M.B., Konrad, M. and Shrum, S., “CMMI Second Edition Guidelines for Process Integration and Product Improvement”, CMMI for Development, Version 1.2, SEI Series in Software Engineering, 2006.

[6] International Organisation for Standardization and International Electrotechnical Commission, “ISO/IEC 15504 Software Process Improvement and Capability Determination Model (SPICE)”, 2004.

[7] Leey, T., Namx, J., Hanx D., Kimx S. and Iny H. P., “Micro Interaction Metrics for Defect Prediction”, ESEC/FSE '11 Proceedings of the 19th ACM SIGSOFT Symposium and the 13th European Conference on Foundations of Software Engineering, 2011.

[8] Dhiauddin, M., “Defect Prediction Model For Testing Phase”, Master Thesis, Universiti Teknologi Malaysia, Faculty of Computer Science and Information System, May 2009.

[9] Wahyudin, D., Schatten, A., Winkler, D., Tjoa, A. M. And Biffl S., “Defect Prediction using Combined Product and Project Metrics A Case Study from the Open Source “Apache” MyFaces Project Family”, Software Engineering and Advanced Applications, 2008. SEAA '08. 34th Euromicro Conference, p.207-215, 3-5 Sept. 2008.

[10] Koru, A. G. and Liu, H., “Building Effective Defect-Prediction Models in Practice”, IEEE Software, Vol. 22, No. 6, November/December 2005.

[11] Florac, A.W., Park, R.E., and Carleton A.D., Practical Software Measurement: Measuring for Process Management and Improvement. Guidebook: CMU/SEI-97-HB-003, 1997.

[12] Runeson, P. and Höst, M., “Guidelines for conducting and reporting case study research in software Engineering”, Empir Software Eng (2009) 14:131–164.

[13] http://www.cs.waikato.ac.nz/~ml/weka/, last access date: 11th April 2012.

[14] Basili, V. R., Caldiera, G. ve Rombach, H. D., “Goal Question Metric Paradigm”, Encyclopedia of Software Engineering – 2 Volume Set, ISBN#1-54004-8, 1994.

[15] http://en.wikipedia.org/wiki/Software_bug, last access date: 16th May 2012.

[16] Tarhan, A., “An Assessment Model For The Applicability of Statistical Process Control For Software Processes”, PhD Thesis, Informatics Institute, Middle East Technical University, Turkey, 2006.

[17] Kan, S. H., “Metrics and Models in Software Quality Engineering”, 2nd Edition, Pearson Education Inc, 2003.

[18] Department of Defense, “Military Standard Quality Assurance Terms and Definitions”, MIL-STD-109C, 1994.

[19] Boehm, B., “Software Engineering Economics”, Prentice Hall, 1981.

[20] Schell, C., "The Value of the Case Study as a Research Strategy", Manchester Business School, 1992.

[21] Challagulla, V. U. B., Bastani, F. B., Yen, I. and Paul, R. A., “Empirical Assessment of Machine Learning based Software Defect Prediction Techniques”, WORDS '05 Proceedings of the 10th IEEE International Workshop on Object-Oriented Real-Time Dependable Systems, USA, 2005.

[22] International Organisation for Standardization and International Electrotechnical Commission, “ISO/IEC JTC 1/SC 7 - Software and Systems Engineering", 2009.

[23] International Organisation for Standardization and International Electrotechnical Commission, “ISO/IEC 24765 - Systems and software engineering - Vocabulary", 2010.

[24] http://en.wikipedia.org/wiki/Machine_learning#cite_ref-0, last access date: 17th May 2012.

[25] http://en.wikipedia.org/wiki/Goodness_of_fit, last access date: 17th May 2012.

[26] Freund, J., E., “Modern Elementary Statistics”, 11th Edition, Pearson, 2004.

[27] Babbie, E., R., The Practice of Social Research", 12th edition, Wadsworth Publishing, 2009.

[28] http://en.wikipedia.org/wiki/Bivariate_analysis, last access date: 5th June 2012.

[29] Abdi, H., “Multivariate Analysis”, The University of Texas at Dallas, 2003.

[30] http://support.sas.com/publishing/pubcat/chaps/56903.pdf, last access date: 17th May 2012.

[31] http://people.revoledu.com/kardi/tutorial/KNN/HowTo_KNN.html, last access date: 17th May 2012.

[32] http://saravananthirumuruganathan.wordpress.com/2010/05/17/a-detailed-introduction-to-k-nearest-neighbor-knn-algorithm/, last access date: 17th May 2012.

[33] http://en.wikipedia.org/wiki/K-nearest_neighbor_algorithm, last access date: 5th June 2012.

[34] Malik, J., S., Goyal, P., Sharma, A., K., "Data Mining Problem Solving Algorithms & Their Comparative Study", Pioneer Journal, 12th National Conference,2011.

[35] http://cis.poly.edu/~mleung/FRE7851/f07/decisionTrees.pdf, last access date: 17th May 2012.

[36] http://www.uow.edu.au/~markus/teaching/CSCI323/Lecture_MLP.pdf, last access date: 17th May 2012.

[37] https://www.cra.com/pdf/bnetbuilderbackground.pdf, last access date: 17th May 2012.

[38] Neil, M., Fenton, N., Nielson, L, "Building large-scale Bayesian Networks", The Knowledge Engineering Review, Vol. 15, No. 3, pp. 257-284, 2000.

[39] http://www.aiai.ed.ac.uk/links/bn.html, last access date: 17th May 2012.

[40] http://www-scf.usc.edu/~csci567/11-12-Bayesian-Learning.pdf], last access date: 17th May 2012.

[41] http://css.engineering.uiowa.edu/~comp/Public/Apriori.pdf, last access date: 17th May 2012.

[42] http://ijrte.academypublisher.com/vol02/no03/ijrte02037377.pdf, last access date: 17th May 2012.

[43] Koru, A. G. and Tian, J., "An Empirical Comparison and Characterization of High Defect and High Complexity Modules, The Journal of Systems and Software, 67 (2003) 153–163, 2003.

[44] Salman, N., "A Measurement Framework for Component Oriented Software Systems", PhD Thesis, Computer Engineering, Middle East Technical University, Turkey, 2006.

[45] Manzoor, K., "A Practical Approach to Estimating Defect-Fix Time", http://homepages.com.pk/kashman/defectsEstimation.htm, last access date: 19th May 2012.

[46] Ohlsson, N., Zhao, M. and Helander, M., "Application of multivariate analysis for software fault prediction, Software Quality Journal 7,(1998) 51–66, 1996.

[47] Boetticher, G., D., "Nearest Neighbor Sampling for Better Defect Prediction", PROMISE '05 Proceedings of the 2005 workshop on Predictor models in software engineering, p. 1-6, USA, 2005.

[48] Sandhu, P. S., Khullar, S., Singh, S., Bains S. K., Kaur, M. and Singh, G., "A Study on Early Prediction of Fault Proneness in Software Modules using Genetic Algorithm", World Academy of Science, Engineering and Technology 72, 2010.

[49] Çatal, Ç. ve Diri, B., "A Systematic Review of Software Fault Prediction Studies", Expert Systems with Applications, 36 (2009) 7346-7354, 2009.

[50] Ahsan, S. N., Ferzund, J. and Wotawa, F., "Program File Bug Fix Effort Estimation UsingMachine Learning Methods for Open SourceSoftware Projects", IST-TR-2009-04-24, IST Technical Report, Institut für Softwaretechnologie, Australia, 2009.

[51] Jalote, P., Dinesh, K., Raghavan, S., Bhashyam R. and Ramakrishnan M., "Quantitative Quality Management through Defect Prediction and Statistical Process Control", http://www.iiitd.edu.in/~jalote/papers/2WCSQPaper.pdf, last access date: 19th May 2012.

[52] Gokhale, S. S., Mullen, R. E., "Software Defect Repair Times: A Multiplicative Model", PROMISE '08 Proceedings of the 4th international workshop on Predictor models in software engineering, p. 93-100, USA, 2008.

[53] Schneidewind, N. F., “Modelling the Fault Correction Process”, ISSRE '01 Proceedings of the 12th International Symposium on Software Reliability Engineering, p. 185, USA, 2001.

[54] Fenton, N. and Neil, M., “A Critique of Software Defect Prediction Models, IEEE Transactions on Software Engineering, Vol. 25, No 3, p. 675-689, 1999.

[55] Fenton, N., Krause, M., Neil P., “A Probabilistic Model for Software Defect Prediction”, For submission to IEEE Transactions in Software Engineering.

[56] He, Z., Shu F., Yang Y., Zhang, W. and Wang, Q., “Data Unpredictability in Software Defect-Fixing Effort Prediction”, 2010 10th International Conference on Quality Software, 2010.

[57] Song, Q., Shepperd, M., Cartwright, M. and Mair, C., “Software Defect Association Mining and Defect Correction Effort Prediction”, IEEE Transactions on Software Engineering, Vol. 32, NO. 2, 2006.

[58] Zeng, H. and Rine D., “Estimation of Software Defects Fix Effort Using Neural Networks”, COMPSAC '04 Proceedings of the 28th Annual International Computer Software and Applications Conference - Workshops and Fast Abstracts – Vol. 02, p. 20-21, USA, 2004.

[59] Thaw, T., Aung, M. P., Wah, N. L., Nyein, S. S., Phyo, Z. L., K. Z., and Htun, K. Z., “Comparison for the Accuracy of Defect Fix Effort Estimation” http://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=05485813, last access date: 20th May 2012.

[60] Menzies, T., Lutz, R., Mikulski, C., “Better Analysis of Defect Data at NASA”, Proceedings of the 15th International Conference on Software Engineering and Knowledge Engineering (SEKE’03), 2003.

[61] Weiss, C., Premraj, R., Zimmermann, T. and Zeller, A.,“ How Long will it Take to Fix This Bug?”, MSR '07 Proceedings of the Fourth International Workshop on Mining Software Repositories, p. 1, USA, 2007.

[62] Hassouna, A. and Tahvildari, T., “An Effort Prediction Framework for Software Defect Correction “, Information and Software Technology 52 (2010) 197–209, 2010.

[63] Hewett, R. and Kijsanayothin, P., “On Modeling Software Defect Repair Time”, Empir Software Eng (2009) 14:165–186, 2008.

[64] Menzies, T., Butcher, A., Marcus, A., Zimmermann, T. and Cok, C.,“Local vs. Global Models for Effort Estimation and Defect Prediction”, ASE '11 Proceedings of the 2011 26th IEEE/ACM International Conference on Automated Software Engineering, p.
343-351, USA, 2011.

[65] Tarhan, A. and Demirörs, O., “Investigating the Effect of Variations in Test Development Process: A Case from a Safety-Critical System”, Software Quality Journal, DOI: 10.1007/s11219-011-9129-8.

[66] Tarhan, A. and Demirörs, O., “Assessment of Software Process and Metrics to Support Quantitative Understanding: Experience from an Undefined Task Management Process”, “Apply Quantitative Management Now”, IEEE Software, DOI: 10.1109/MS.2011.91, 2011.

[67] ISO, ISO 9001: 2000: Quality Management Systems – Requirements.

[68] Witten, I. H. and Frank E., “Data Mining Practical Machine Learning Tools and Techniques”, 2nd Edition, Elsevier, 2005.

[69] Langford, J., “Clever Methods of Overfitting“, Machine Learning (Theory), http://hunch.net/?p=22, last access date: 20th May 2012 , 2005.

[70] Demirörs, O., Yıldız, Ö. and Güceğlioğlu, A. S., “Using Cost of Software Quality for a Process Improvement Initiative”, METU, 26th EUROMICRO Conference (EUROMICRO'00)-Volume 2, 2007.

[71] http://www.locmetrics.com/, last access date: 11th April 2012.

[72] http://msdn.microsoft.com/en-us/library/ms174493, last access date: 15th July2012.

APPENDICES

A. CASE STUDY PLAN

Table A.1 Tasks of Study

Design of Study	Purpose
Preparation of GQM tree	Define goals, metrics and statistical analysis methods.
Data collection from tools	Defect density and other factors data will be available to start analysis.
Data verification	Verify data before using for analysis and to decide on applicability for our analysis, Metric Usability Questionnaire Forms are filled for each basic and derived metrics. (A Sample Form given in Attachment-1)
Process data collection	Obtain Process Data. PER (Process Execution Report), PSM (Process Similarity Matrix) will be used.
Conduct interviews with domain experts	Obtain Process Data. PER and PSM will be used.
Data Analysis	Statistical and Machine Learning data analysis methods will be applied to data after data cleaning.
Presentation Preparation, Reporting	Analysis results will be documented. Observed interesting patterns will be shared, suggestions will be discussed.

Table A.2 Study Calendar

Tasks	Start Date	Finish Date	Duration
Preparation of GQM tree	14.03.2012	31.03.2012	17 days
Data collection from tools	14.03.2012	29.03.2012	17 days
Data verification	30.03.2012	31.03.2012	2 days
Process data collection	01.04.2012	15.04.2012	14 days
Conduct interviews with domain experts	01.04.2012	15.04.2012	14 days
Data Analysis	16.04.2012	30.04.2012	14 days
Presentation Preparation, Make Corrections according to Review Items, Reporting	01.05.2012	28.05.2012	27 days

B. DETAILS OF CASE STUDY 1A

Metric Name: Source component				
Attributes		**Answers**	**Rating**	**Expected Answers**
	Indicators			
Measure Identity			N	
Q1	Which entity does the measure measure?	Product		
Q2	Which attribute of the entity does the measure measure?	Defective components of product		
Q3	What is the scale of the measurement data? (nominal, ordinal, interval, ratio, absolute)	Nominal		Ratio, Absolute
Q4	What is the unit of the measurement data?	Not applicable		
Q5	What is the type of the measurement data? (integer, real, etc.)	Text		
Q6	What is the range of the measurement data?	Five component types		
Data Existence			F	
Q7	Is measurement data existent?	Yes		
Q8	What is the amount of overall observations?	296	√	Available > 20
Q9	What is the amount of missing data points?	0		
Q10	Are data points missing in periods? (If yes, please state observation numbers for missing periods)	0		
Q11	Is measurement data time sequenced? (If no, please state how measurement data is sequenced)	Yes		
Data Verifiability			F	
Q12	When is measurement data recorded in the process? (at start, middle, end, later, etc.)	At start		
Q13	Is all measurement data recorded at the same place in the process? (at start, middle, end, later, etc.)	Yes	√	Yes
Q14	Who is responsible for recording measurement data?	Test Specialist		
Q15	Is all measurement data recorded by the responsible body?	Yes	√	Yes
Q16	How is measurement data recorded? (on a form, report, tool, etc.)	Tool		
Q17	Is all measurement data recorded the same way? (on a form, report, tool, etc.)	Yes	√	Yes
Q18	Where is measurement data stored? (in a file, database, etc.)	The tool's database		
Q19	Is all measurement data stored in the same place? (in a file, database, etc.)	Yes	√	Yes
Data Dependability			P	
Q20	What is the frequency of generating measurement data? (asynchronously, daily, weekly, monthly, etc.)	Asynchronously		
Q21	What is the frequency of recording measurement data? (asynchronously, daily, weekly, monthly, etc.)	Synchronously		
Q22	What is the frequency of storing measurement data? (asynchronously, daily, weekly, monthly, etc.)	Synchronously		
Q23	Are the frequencies for data generation, recording, and storing different?	No	√	No
Q24	Is measurement data recorded precisely?	Yes	√	Yes
Q25	Is measurement data collected for a specific purpose?	No	√	Yes
Q26	Is the purpose of measurement data collection known by process performers?	No	√	Yes
Q27	Is measurement data analyzed and reported?	No		Yes
Q28	Is measurement data analysis results communicated to process performers?	No		Yes
Q29	Is measurement data analysis results communicated to management?	No		Yes
Q30	Is measurement data analysis results used as a basis for decision making?	No		Yes
Data Normalizability				
Q31	Can measurement data be normalized by parameters or measures? (If yes, please specify them)	No		
Data Integrability				
Q32	Is measurement data integrable at project level?	No		
Q33	Is measurement data integrable at organization level?	No		

Figure B.1 MUQ for “Source component” Basic Metric of Project-1

Metric Name: Created date				
Attributes		Answers	Rating	Expected Answers
	Indicators			
Measure Identity			N	
Q1	Which entity does the measure measure?	Process		
Q2	Which attribute of the entity does the measure measure?	The date of the defect record		
Q3	What is the scale of the measurement data? (nominal, ordinal, interval, ratio, absolute)	Nominal		Ratio, Absolute
Q4	What is the unit of the measurement data?	Time (dd.mm.yy hh:mm)		
Q5	What is the type of the measurement data? (integer, real, etc.)	Date		
Q6	What is the range of the measurement data?	00.00.0000 00:00		
Data Existence			F	
Q7	Is measurement data existent?	Yes		
Q8	What is the amount of overall observations?	296	√	Available > 20
Q9	What is the amount of missing data points?	0		
Q10	Are data points missing in periods? (If yes, please state observation numbers for missing periods)	0		
Q11	Is measurement data time sequenced? (If no, please state how measurement data is sequenced)	Yes		
Data Verifiability			F	
Q12	When is measurement data recorded in the process? (at start, middle, end, later, etc.)	At start		
Q13	Is all measurement data recorded at the same place in the process? (at start, middle, end, later, etc.)	Yes	√	Yes
Q14	Who is responsible for recording measurement data?	Test Specialist		
Q15	Is all measurement data recorded by the responsible body?	Yes	√	Yes
Q16	How is measurement data recorded? (on a form, report, tool, etc.)	Tool		
Q17	Is all measurement data recorded the same way? (on a form, report, tool, etc.)	Yes	√	Yes
Q18	Where is measurement data stored? (in a file, database, etc.)	The tool's database		
Q19	Is all measurement data stored in the same place? (in a file, database, etc.)	Yes	√	Yes
Data Dependability			P	
Q20	What is the frequency of generating measurement data? (asynchronously, daily, weekly, monthly, etc.)	Asynchronously		
Q21	What is the frequency of recording measurement data? (asynchronously, daily, weekly, monthly, etc.)	Synchronously		
Q22	What is the frequency of storing measurement data? (asynchronously, daily, weekly, monthly, etc.)	Synchronously		
Q23	Are the frequencies for data generation, recording, and storing different?	No	√	No
Q24	Is measurement data recorded precisely?	Yes	√	Yes
Q25	Is measurement data collected for a specific purpose?	No	√	Yes
Q26	Is the purpose of measurement data collection known by process performers?	No	√	Yes
Q27	Is measurement data analyzed and reported?	No		Yes
Q28	Is measurement data analysis results communicated to process performers?	No		Yes
Q29	Is measurement data analysis results communicated to management?	No		Yes
Q30	Is measurement data analysis results used as a basis for decision making?	No		Yes
Data Normalizability				
Q31	Can measurement data be normalized by parameters or measures? (If yes, please specify them)	No		
Data Integrability				
Q32	Is measurement data integrable at project level?	No		
Q33	Is measurement data integrable at organization level?	No		

Figure B.2 MUQ for “Created Date” Basic Metric of Project-1

Metric Name: Closed date				
Attributes		**Answers**	**Rating**	**Expected Answers**
	Indicators			
Measure Identity			N	
Q1	Which entity does the measure measure?	Process		
Q2	Which attribute of the entity does the measure measure?	Closed date of the defect record		
Q3	What is the scale of the measurement data? (nominal, ordinal, interval, ratio, absolute)	Nominal		Ratio, Absolute
Q4	What is the unit of the measurement data?	Time (dd.mm.yy hh:mm)		
Q5	What is the type of the measurement data? (integer, real, etc.)	Date		
Q6	What is the range of the measurement data?	00.00.0000 00:00		
Data Existence			F	
Q7	Is measurement data existent?	Yes		
Q8	What is the amount of overall observations?	296	√	Available > 20
Q9	What is the amount of missing data points?	0		
Q10	Are data points missing in periods? (If yes, please state observation numbers for missing periods)	0		
Q11	Is measurement data time sequenced? (If no, please state how measurement data is sequenced)	Yes		
Data Verifiability			F	
Q12	When is measurement data recorded in the process? (at start, middle, end, later, etc.)	Later		
Q13	Is all measurement data recorded at the same place in the process? (at start, middle, end, later, etc.)	Yes	√	Yes
Q14	Who is responsible for recording measurement data?	Project Manager		
Q15	Is all measurement data recorded by the responsible body?	Yes	√	Yes
Q16	How is measurement data recorded? (on a form, report, tool, etc.)	Tool		
Q17	Is all measurement data recorded the same way? (on a form, report, tool, etc.)	Yes	√	Yes
Q18	Where is measurement data stored? (in a file, database, etc.)	The tool's database		
Q19	Is all measurement data stored in the same place? (in a file, database, etc.)	Yes	√	Yes
Data Dependability			P	
Q20	What is the frequency of generating measurement data? (asynchronously, daily, weekly, monthly, etc.)	Asynchronously		
Q21	What is the frequency of recording measurement data? (asynchronously, daily, weekly, monthly, etc.)	Asynchronously		
Q22	What is the frequency of storing measurement data? (asynchronously, daily, weekly, monthly, etc.)	Synchronously		
Q23	Are the frequencies for data generation, recording, and storing different?	No	√	No
Q24	Is measurement data recorded precisely?	Yes	√	Yes
Q25	Is measurement data collected for a specific purpose?	No	√	Yes
Q26	Is the purpose of measurement data collection known by process performers?	No	√	Yes
Q27	Is measurement data analyzed and reported?	No		Yes
Q28	Is measurement data analysis results communicated to process performers?	No		Yes
Q29	Is measurement data analysis results communicated to management?	No		Yes
Q30	Is measurement data analysis results used as a basis for decision making?	No		Yes
Data Normalizability				
Q31	Can measurement data be normalized by parameters or measures? (If yes, please specify them)	No		
Data Integrability				
Q32	Is measurement data integrable at project level?	No		
Q33	Is measurement data integrable at organization level?	No		

Figure B.3 MUQ for "Closed Date" Basic Metric of Project-1

Metric Name: Test type				
Attributes		**Answers**	**Rating**	**Expected Answers**
	Indicators			
Measure Identity			N	
Q1	Which entity does the measure measure?	Process		
Q2	Which attribute of the entity does the measure measure?	Defect detection rate of different test activities		
Q3	What is the scale of the measurement data? (nominal, ordinal, interval, ratio, absolute)	Nominal		Ratio, Absolute
Q4	What is the unit of the measurement data?	Not applicable		
Q5	What is the type of the measurement data? (integer, real, etc.)	Text		
Q6	What is the range of the measurement data?	Two test activity types		
Data Existence			F	
Q7	Is measurement data existent?	Yes		
Q8	What is the amount of overall observations?	296	√	Available > 20
Q9	What is the amount of missing data points?	0		
Q10	Are data points missing in periods? (If yes, please state observation numbers for missing periods)	0		
Q11	Is measurement data time sequenced? (If no, please state how measurement data is sequenced)	Yes		
Data Verifiability			F	
Q12	When is measurement data recorded in the process? (at start, middle, end, later, etc.)	At start		
Q13	Is all measurement data recorded at the same place in the process? (at start, middle, end, later, etc.)	Yes	√	Yes
Q14	Who is responsible for recording measurement data?	Test Specialist		
Q15	Is all measurement data recorded by the responsible body?	Yes	√	Yes
Q16	How is measurement data recorded? (on a form, report, tool, etc.)	Tool		
Q17	Is all measurement data recorded the same way? (on a form, report, tool, etc.)	Yes	√	Yes
Q18	Where is measurement data stored? (in a file, database, etc.)	The tool's database		
Q19	Is all measurement data stored in the same place? (in a file, database, etc.)	Yes	√	Yes
Data Dependability			P	
Q20	What is the frequency of generating measurement data? (asynchronously, daily, weekly, monthly,	Asynchronously		
Q21	What is the frequency of recording measurement data? (asynchronously, daily, weekly, monthly,	Synchronously		
Q22	What is the frequency of storing measurement data? (asynchronously, daily, weekly, monthly, etc.)	Synchronously		
Q23	Are the frequencies for data generation, recording, and storing different?	No	√	No
Q24	Is measurement data recorded precisely?	Yes	√	Yes
Q25	Is measurement data collected for a specific purpose?	No	√	Yes
Q26	Is the purpose of measurement data collection known by process performers?	No	√	Yes
Q27	Is measurement data analyzed and reported?	No		Yes
Q28	Is measurement data analysis results communicated to process performers?	No		Yes
Q29	Is measurement data analysis results communicated to management?	No		Yes
Q30	Is measurement data analysis results used as a basis for decision making?	No		Yes
Data Normalizability				
Q31	Can measurement data be normalized by parameters or measures? (If yes, please specify them)	No		
Data Integrability				
Q32	Is measurement data integrable at project level?	Yes		
Q33	Is measurement data integrable at organization level?	Yes		

Figure B.4 MUQ for "Test Type" Basic Metric of Project-1

Metric Name: Product version					
Attributes			Answers	Rating	Expected Answers
	Indicators				
Measure Identity				N	
	Q1	What is the measure formula? (please refer to related basic metrics)	Process		
	Q2	Which attribute of the entity does the measure measure?	Frequency of product development updates		
	Q3	What is the scale of the measurement data? (nominal, ordinal, interval, ratio, absolute)	Nominal		Ratio, Absolute
	Q4	What is the unit of the measurement data?	Not applicable		
	Q5	What is the type of the measurement data? (integer, real, etc.)	Text (x.y.z)		
	Q6	What is the range of the measurement data?	0.0.1-2.0.8		
Data Existence				F	
	Q7	Is measurement data existent?	Yes		
	Q8	What is the amount of overall observations?	296	√	Available > 20
	Q9	What is the amount of missing data points?	0		
	Q10	Are data points missing in periods? (If yes, please state observation numbers for missing periods)	0		
	Q11	Is measurement data time sequenced? (If no, please state how measurement data is sequenced)	Yes		
Data Verifiability				F	
	Q12	When is measurement data recorded in the process? (at start, middle, end, later, etc.)	At start		
	Q13	Is all measurement data recorded at the same place in the process? (at start, middle, end, later, etc.)	Yes	√	Yes
	Q14	Who is responsible for recording measurement data?	Test Specialist		
	Q15	Is all measurement data recorded by the responsible body?	Yes	√	Yes
	Q16	How is measurement data recorded? (on a form, report, tool, etc.)	Tool		
	Q17	Is all measurement data recorded the same way? (on a form, report, tool, etc.)	Yes	√	Yes
	Q18	Where is measurement data stored? (in a file, database, etc.)	The tool's database		
	Q19	Is all measurement data stored in the same place? (in a file, database, etc.)	Yes	√	Yes
Data Dependability				P	
	Q20	What is the frequency of generating measurement data? (asynchronously, daily, weekly, monthly, etc.)	Asynchronously		
	Q21	What is the frequency of recording measurement data? (asynchronously, daily, weekly, monthly, etc.)	Synchronously		
	Q22	What is the frequency of storing measurement data? (asynchronously, daily, weekly, monthly, etc.)	Synchronously		
	Q23	Are the frequencies for data generation, recording, and storing different?	No	√	No
	Q24	Is measurement data recorded precisely?	Yes	√	Yes
	Q25	Is measurement data collected for a specific purpose?	Yes	√	Yes
	Q26	Is the purpose of measurement data collection known by process performers?	Yes	√	Yes
	Q27	Is measurement data analyzed and reported?	No		Yes
	Q28	Is measurement data analysis results communicated to process performers?	No		Yes
	Q29	Is measurement data analysis results communicated to management?	No		Yes
	Q30	Is measurement data analysis results used as a basis for decision making?	No		Yes
Data Normalizability					
	Q31	Can measurement data be normalized by parameters or measures? (If yes, please specify them)	No		
Data Integrability					
	Q32	Is measurement data integrable at project level?	No		
	Q33	Is measurement data integrable at organization level?	No		

Figure B.5 MUQ for “Product Version” Basic Metric of Project-1

Metric Name: Product SLOC				
Attributes		Answers	Rating	Expected Answers
	Indicators			
Measure Identity			N	
Q1	Which entity does the measure measure?	Product		
Q2	Which attribute of the entity does the measure measure?	Size of the product version		
Q3	What is the scale of the measurement data? (nominal, ordinal, interval, ratio, absolute)	Absolute		Ratio, Absolute
Q4	What is the unit of the measurement data?	LOC, KLOC		
Q5	What is the type of the measurement data? (integer, real, etc.)	Integer		
Q6	What is the range of the measurement data?	[8469,23425]		
Data Existence			F	
Q7	Is measurement data existent?	No (collected manually)		
Q8	What is the amount of overall observations?	11	√	Available > 20
Q9	What is the amount of missing data points?	0		
Q10	Are data points missing in periods? (If yes, please state observation numbers for missing periods)	0		
Q11	Is measurement data time sequenced? (If no, please state how measurement data is sequenced)	Yes		
Data Verifiability			F	
Q12	When is measurement data recorded in the process? (at start, middle, end, later, etc.)	Monthly		
Q13	Is all measurement data recorded at the same place in the process? (at start, middle, end, later, etc.)	Yes	√	Yes
Q14	Who is responsible for recording measurement data?	Project Manager		
Q15	Is all measurement data recorded by the responsible body?	Yes	√	Yes
Q16	How is measurement data recorded? (on a form, report, tool, etc.)	Report		
Q17	Is all measurement data recorded the same way? (on a form, report, tool, etc.)	Yes	√	Yes
Q18	Where is measurement data stored? (in a file, database, etc.)	tool		
Q19	Is all measurement data stored in the same place? (in a file, database, etc.)	Yes	√	Yes
Data Dependability			P	
Q20	What is the frequency of generating measurement data? (asynchronously, daily, weekly, monthly, etc.)	Asynchronously		
Q21	What is the frequency of recording measurement data? (asynchronously, daily, weekly, monthly, etc.)	Monthly		
Q22	What is the frequency of storing measurement data? (asynchronously, daily, weekly, monthly, etc.)	Synchronously		
Q23	Are the frequencies for data generation, recording, and storing different?	No	√	No
Q24	Is measurement data recorded precisely?	Yes	√	Yes
Q25	Is measurement data collected for a specific purpose?	Yes	√	Yes
Q26	Is the purpose of measurement data collection known by process performers?	Yes	√	Yes
Q27	Is measurement data analyzed and reported?	Yes		Yes
Q28	Is measurement data analysis results communicated to process performers?	Yes		Yes
Q29	Is measurement data analysis results communicated to management?	Yes		Yes
Q30	Is measurement data analysis results used as a basis for decision making?	Yes		Yes
Data Normalizability				
Q31	Can measurement data be normalized by parameters or measures? (If yes, please specify them)	Yes (KLOC)		
Data Integrability				
Q32	Is measurement data integrable at project level?	Yes		
Q33	Is measurement data integrable at organization level?	Yes		

Figure B.6 MUQ for “Product SLOC” Basic Metric of Project-1

Metric Name: Product complexity				
Attributes		**Answers**	**Rating**	**Expected Answers**
	Indicators			
Measure Identity			N	
Q1	Which entity does the measure measure?	Product		
Q2	Which attribute of the entity does the measure measure?	Complexity of the product version		
Q3	What is the scale of the measurement data? (nominal, ordinal, interval, ratio, absolute)	Absolute		Ratio, Absolute
Q4	What is the unit of the measurement data?	Number of decision nodes in software		
Q5	What is the type of the measurement data? (integer, real, etc.)	Integer		
Q6	What is the range of the measurement data?	[0...]		
Data Existence			F	
Q7	Is measurement data existent?	No (collected manually)		
Q8	What is the amount of overall observations?	11	√	Available > 20
Q9	What is the amount of missing data points?	0		
Q10	Are data points missing in periods? (If yes, please state observation numbers for missing periods)	0		
Q11	Is measurement data time sequenced? (If no, please state how measurement data is sequenced)	Yes		
Data Verifiability			F	
Q12	When is measurement data recorded in the process? (at start, middle, end, later, etc.)	Never		
Q13	Is all measurement data recorded at the same place in the process? (at start, middle, end, later, etc.)	No	√	Yes
Q14	Who is responsible for recording measurement data?	No one		
Q15	Is all measurement data recorded by the responsible body?	No	√	Yes
Q16	How is measurement data recorded? (on a form, report, tool, etc.)	No		
Q17	Is all measurement data recorded the same way? (on a form, report, tool, etc.)	No	√	Yes
Q18	Where is measurement data stored? (in a file, database, etc.)	No		
Q19	Is all measurement data stored in the same place? (in a file, database, etc.)	No	√	Yes
Data Dependability			P	
Q20	What is the frequency of generating measurement data? (asynchronously, daily, weekly, monthly, etc.)	Asynchronously		
Q21	What is the frequency of recording measurement data? (asynchronously, daily, weekly, monthly, etc.)	Synchronously		
Q22	What is the frequency of storing measurement data? (asynchronously, daily, weekly, monthly, etc.)	Synchronously		
Q23	Are the frequencies for data generation, recording, and storing different?	Yes	√	No
Q24	Is measurement data recorded precisely?	No	√	Yes
Q25	Is measurement data collected for a specific purpose?	No	√	Yes
Q26	Is the purpose of measurement data collection known by process performers?	No	√	Yes
Q27	Is measurement data analyzed and reported?	No		Yes
Q28	Is measurement data analysis results communicated to process performers?	No		Yes
Q29	Is measurement data analysis results communicated to management?	No		Yes
Q30	Is measurement data analysis results used as a basis for decision making?	No		Yes
Data Normalizability				
Q31	Can measurement data be normalized by parameters or measures? (If yes, please specify them)	No		
Data Integrability				
Q32	Is measurement data integrable at project level?	Yes		
Q33	Is measurement data integrable at organization level?	Yes		

Figure B.7 MUQ for "Product Complexity" Basic Metric of Project-1

Metric Name: Reproducibility				
Attributes		**Answers**	**Rating**	**Expected Answers**
	Indicators			
Measure Identity			N	
Q1	Which entity does the measure measure?	Process		
Q2	Which attribute of the entity does the measure measure?	Repeatability status of detected defects		
Q3	What is the scale of the measurement data? (nominal, ordinal, interval, ratio, absolute)	Nominal		Ratio, Absolute
Q4	What is the unit of the measurement data?	Not applicable		
Q5	What is the type of the measurement data? (integer, real, etc.)	Text		
Q6	What is the range of the measurement data?	Three status types		
Data Existence			F	
Q7	Is measurement data existent?	Yes		
Q8	What is the amount of overall observations?	296	√	Available > 20
Q9	What is the amount of missing data points?	0		
Q10	Are data points missing in periods? (If yes, please state observation numbers for missing periods)	0		
Q11	Is measurement data time sequenced? (If no, please state how measurement data is sequenced)	Yes		
Data Verifiability			F	
Q12	When is measurement data recorded in the process? (at start, middle, end, later, etc.)	At start		
Q13	Is all measurement data recorded at the same place in the process? (at start, middle, end, later, etc.)	Yes	√	Yes
Q14	Who is responsible for recording measurement data?	Test Specialist		
Q15	Is all measurement data recorded by the responsible body?	Yes	√	Yes
Q16	How is measurement data recorded? (on a form, report, tool, etc.)	Tool		
Q17	Is all measurement data recorded the same way? (on a form, report, tool, etc.)	Yes	√	Yes
Q18	Where is measurement data stored? (in a file, database, etc.)	The tool's database		
Q19	Is all measurement data stored in the same place? (in a file, database, etc.)	Yes	√	Yes
Data Dependability			P	
Q20	What is the frequency of generating measurement data? (asynchronously, daily, weekly, monthly,	Asynchronously		
Q21	What is the frequency of recording measurement data? (asynchronously, daily, weekly, monthly,	Synchronously		
Q22	What is the frequency of storing measurement data? (asynchronously, daily, weekly, monthly, etc.)	Synchronously		
Q23	Are the frequencies for data generation, recording, and storing different?	No	√	No
Q24	Is measurement data recorded precisely?	Yes	√	Yes
Q25	Is measurement data collected for a specific purpose?	No	√	Yes
Q26	Is the purpose of measurement data collection known by process performers?	No	√	Yes
Q27	Is measurement data analyzed and reported?	No		Yes
Q28	Is measurement data analysis results communicated to process performers?	No		Yes
Q29	Is measurement data analysis results communicated to management?	No		Yes
Q30	Is measurement data analysis results used as a basis for decision making?	No		Yes
Data Normalizability				
Q31	Can measurement data be normalized by parameters or measures? (If yes, please specify them)	No		
Data Integrability				
Q32	Is measurement data integrable at project level?	Yes		
Q33	Is measurement data integrable at organization level?	Yes		

Figure B.8 MUQ for “Reproducibility” Basic Metric of Project-1

Metric Name: Project phase					
Attributes			**Answers**	**Rating**	**Expected Answers**
	Indicators				
Measure Identity				N	
	Q1	Which entity does the measure measure?	Process		
	Q2	Which attribute of the entity does the measure measure?	Project phase of detected defects		
	Q3	What is the scale of the measurement data? (nominal, ordinal, interval, ratio, absolute)	Nominal		Ratio, Absolute
	Q4	What is the unit of the measurement data?	Not applicable		
	Q5	What is the type of the measurement data? (integer, real, etc.)	Text		
	Q6	What is the range of the measurement data?	Two phase types		
Data Existence				F	
	Q7	Is measurement data existent?	No		
	Q8	What is the amount of overall observations?	Not applicable	√	Available > 20
	Q9	What is the amount of missing data points?	296		
	Q10	Are data points missing in periods? (If yes, please state observation numbers for missing periods)	296		
	Q11	Is measurement data time sequenced? (If no, please state how measurement data is sequenced)	Not applicable		
Data Verifiability (After manual collection)				F	
	Q12	When is measurement data recorded in the process? (at start, middle, end, later, etc.)	Later		
	Q13	Is all measurement data recorded at the same place in the process? (at start, middle, end, later, etc.)	Later	√	Yes
	Q14	Who is responsible for recording measurement data?	Project Manager		
	Q15	Is all measurement data recorded by the responsible body?	Yes	√	Yes
	Q16	How is measurement data recorded? (on a form, report, tool, etc.)	On a form		
	Q17	Is all measurement data recorded the same way? (on a form, report, tool, etc.)	Yes	√	Yes
	Q18	Where is measurement data stored? (in a file, database, etc.)	In a file		
	Q19	Is all measurement data stored in the same place? (in a file, database, etc.)	Yes	√	Yes
Data Dependability				P	
	Q20	What is the frequency of generating measurement data? (asynchronously, daily, weekly, monthly,	Asynchronously		
	Q21	What is the frequency of recording measurement data? (asynchronously, daily, weekly, monthly,	Asynchronously		
	Q22	What is the frequency of storing measurement data? (asynchronously, daily, weekly, monthly, etc.)	Synchronously		
	Q23	Are the frequencies for data generation, recording, and storing different?	Yes	√	No
	Q24	Is measurement data recorded precisely?	No	√	Yes
	Q25	Is measurement data collected for a specific purpose?	No	√	Yes
	Q26	Is the purpose of measurement data collection known by process performers?	No	√	Yes
	Q27	Is measurement data analyzed and reported?	No		Yes
	Q28	Is measurement data analysis results communicated to process performers?	No		Yes
	Q29	Is measurement data analysis results communicated to management?	No		Yes
	Q30	Is measurement data analysis results used as a basis for decision making?	No		Yes
Data Normalizability					
	Q31	Can measurement data be normalized by parameters or measures? (If yes, please specify them)	No		
Data Integrability					
	Q32	Is measurement data integrable at project level?	Yes		
	Q33	Is measurement data integrable at organization level?	Yes		

Figure B.9 MUQ for “Project Phase” Basic Metric of Project-1

Metric Name: Remaining open duration					
Attributes			Answers	Rating	Expected Answers
	Indicators				
Measure Identity				N	
	Q1	What is the measure formula? (please refer to related basic metrics)	Created date, closed date		
	Q2	What is the scale of the measurement data? (nominal, ordinal, interval, ratio, absolute)	Absolute		Ratio, Absolute
	Q3	What is the unit of the measurement data?	day		
	Q4	What is the type of the measurement data? (integer, real, etc.)	integer		
	Q5	What is the range of the measurement data?	[0-108]		
Data Existence				F	
	Q6	Is measurement data existent?	Yes		
	Q7	What is the amount of overall observations?	296	√	Available > 20
	Q8	What is the amount of missing data points?	0		
	Q9	Are data points missing in periods? (if yes, please state observation numbers for missing periods)	0		
	Q10	Is measurement data time sequenced? (if no, please state how measurement data is sequenced)	Yes		
Data Verifiability				F	
	Q11	How is the measure data calculated? (by a tool, manually, etc.)	By a tool		
	Q12	Is all measurement data calculated with the same way? (by a tool, manually, etc.)	Yes	√	Yes
	Q13	Is all measurement data calculated according to measure formula?	Yes		
	Q14	Where is measurement data stored? (In a file, database, etc.)	In a file	√	Yes
	Q15	Is all measurement data stored in the same place? (In a file, database, etc.)	Yes		
Data Dependability				P	
	Q16	Is measurement data stored precisely?	Yes	√	Yes
	Q17	Is measurement data stored for a specific purpose?	Yes	√	Yes
	Q18	Is the purpose of measurement data collection known by process performers?	Yes	√	Yes
	Q19	Is measurement data analyzed and reported?	No		Yes
	Q20	Is measurement data analysis results communicated to process performers?	No		Yes
	Q21	Is measurement data analysis results communicated to management?	No		Yes
	Q22	Is measurement data analysis results used as a basis for decision making?	No		Yes
Data Normalizability					
	Q23	Can measurement data be normalized by parameters or measures? (if yes, please specify them)	No		
Data Integrability					
	Q24	Is measurement data integrable at project level?	Yes		
	Q25	Is measurement data integrable at organization level?	Yes		

Figure B.10 MUQ for "Defect Open Duration" Derived Metric of Project-1

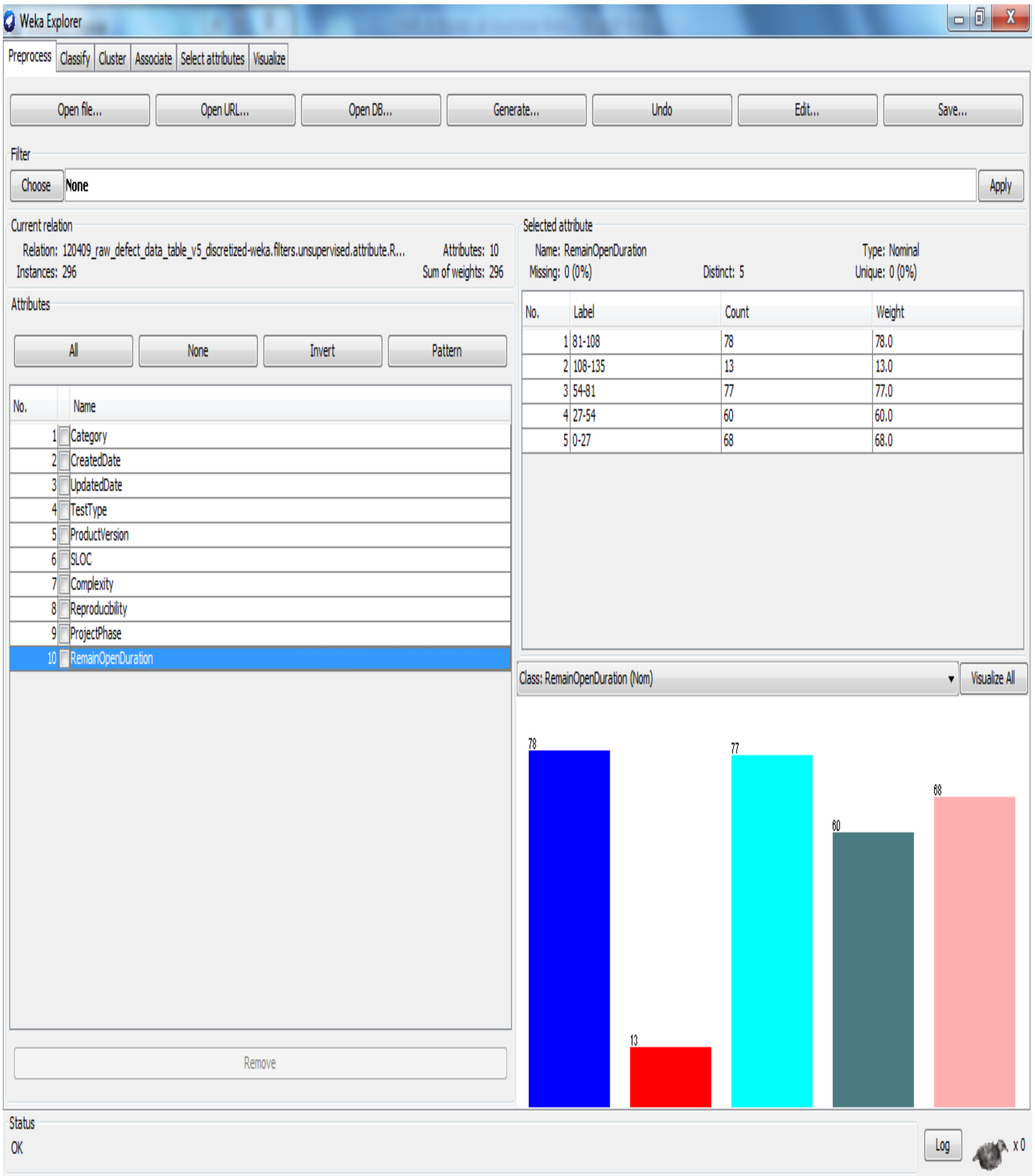

Figure B.11 Weka View of Case Study 1A

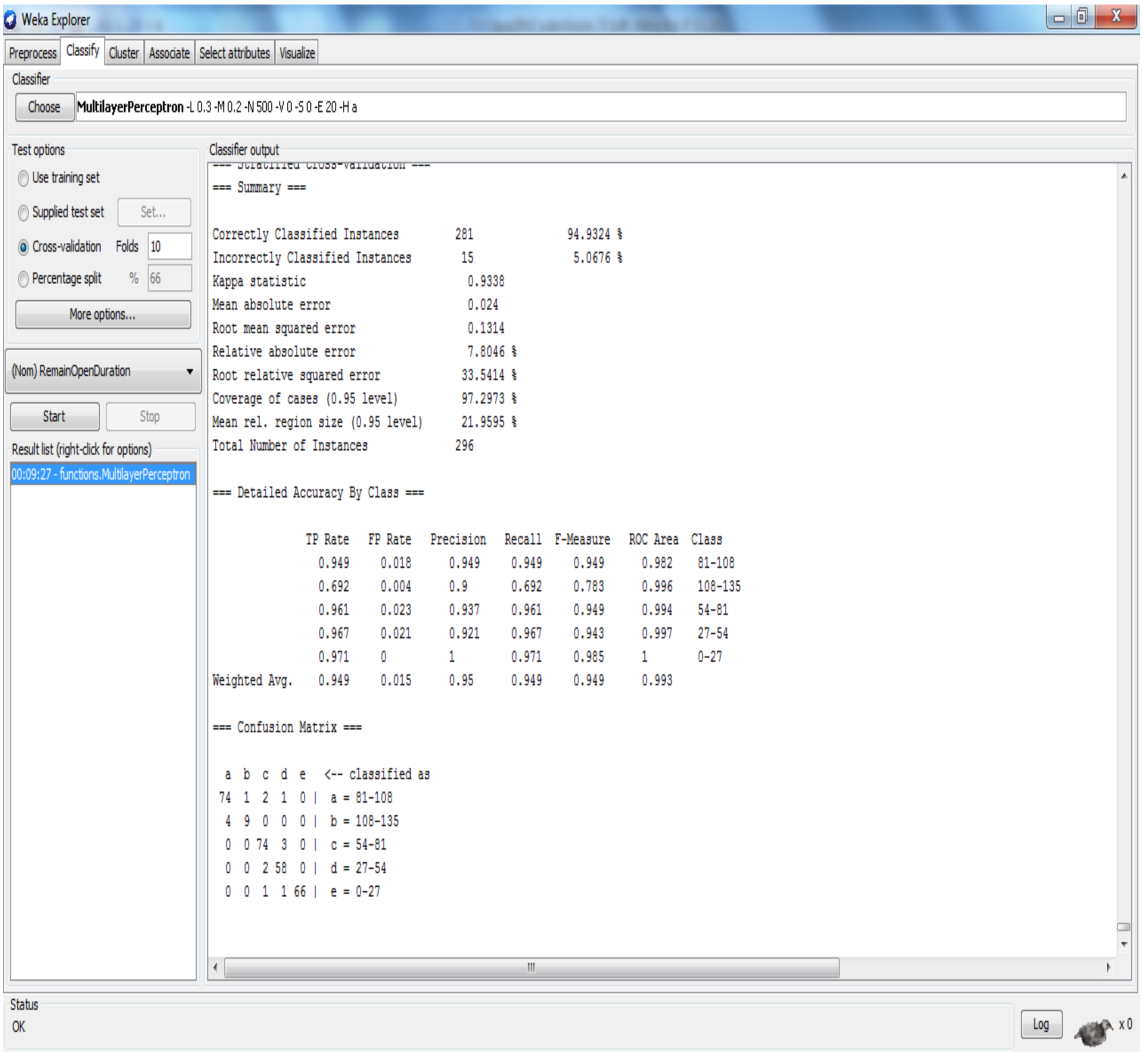

Figure B.12 Multilayer Perceptron Results of Case Study 1A

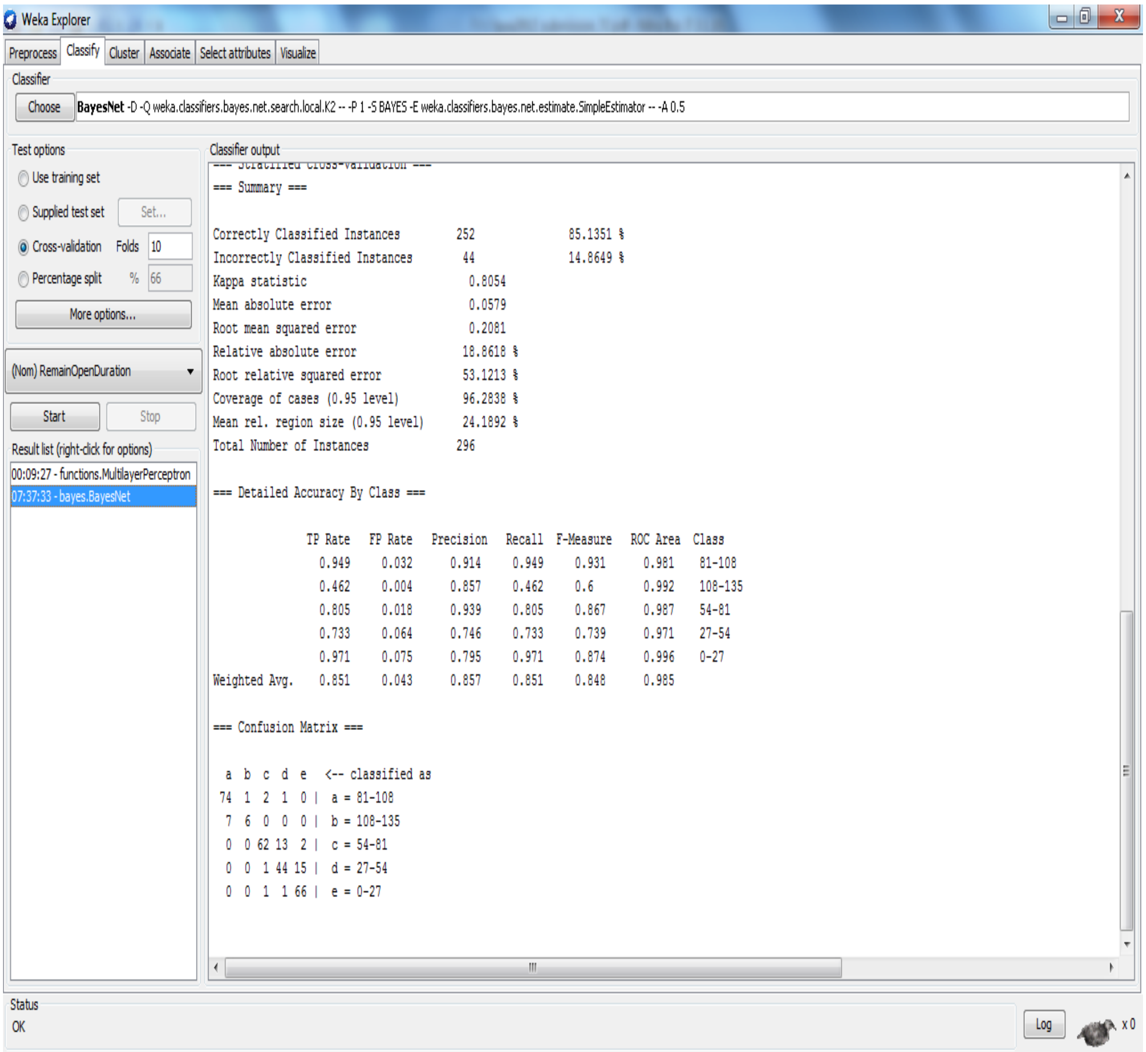

Figure B.13 BayesNet Results of Case Study 1A

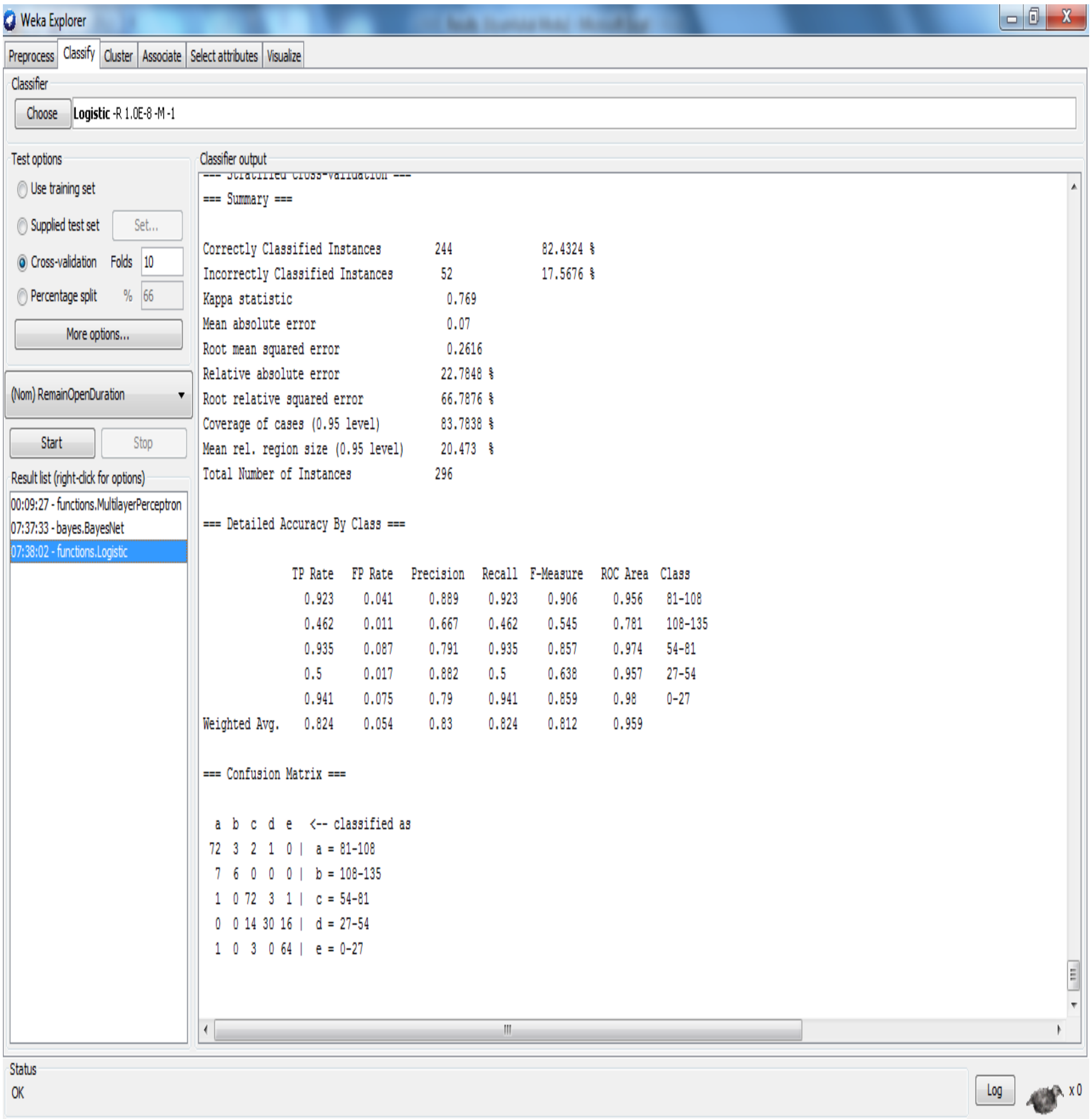

Figure B.14 Logistic Results of Case Study 1A

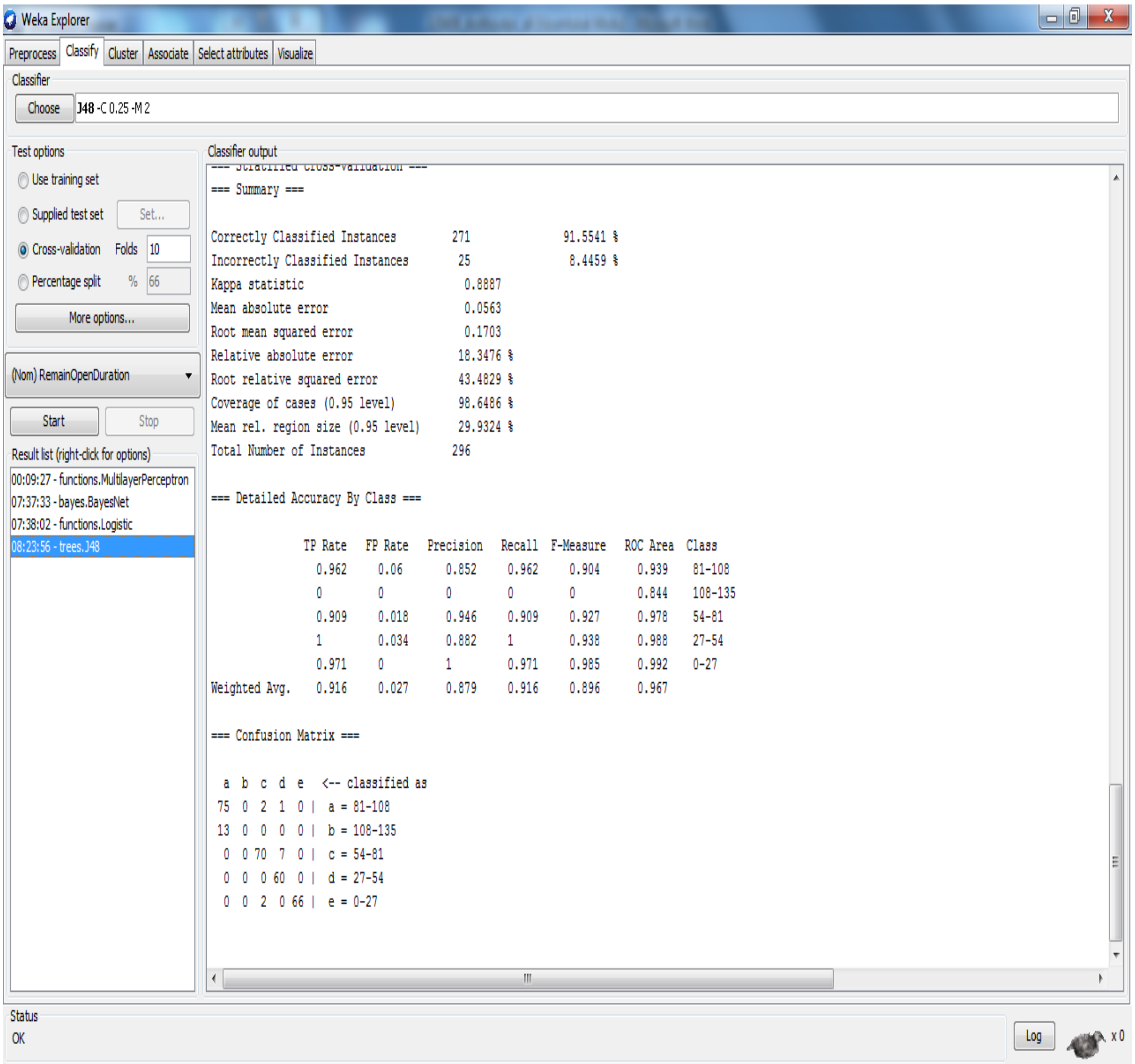

Figure B.15 J48 Results of Case Study 1A

C. DETAILS OF CASE STUDY 1B

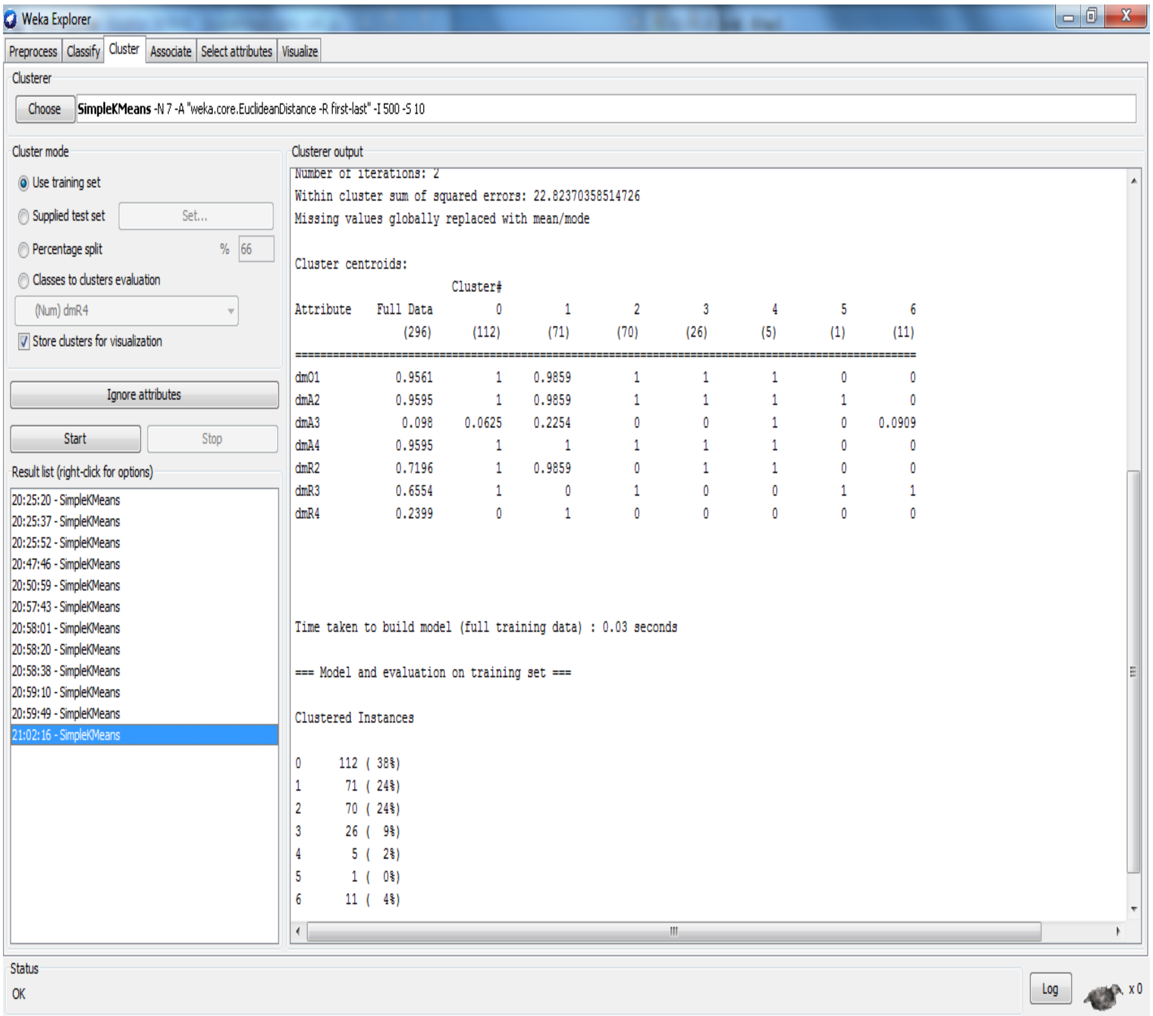

Figure C.1SimpleKMeans Clustering of Case Study 1B

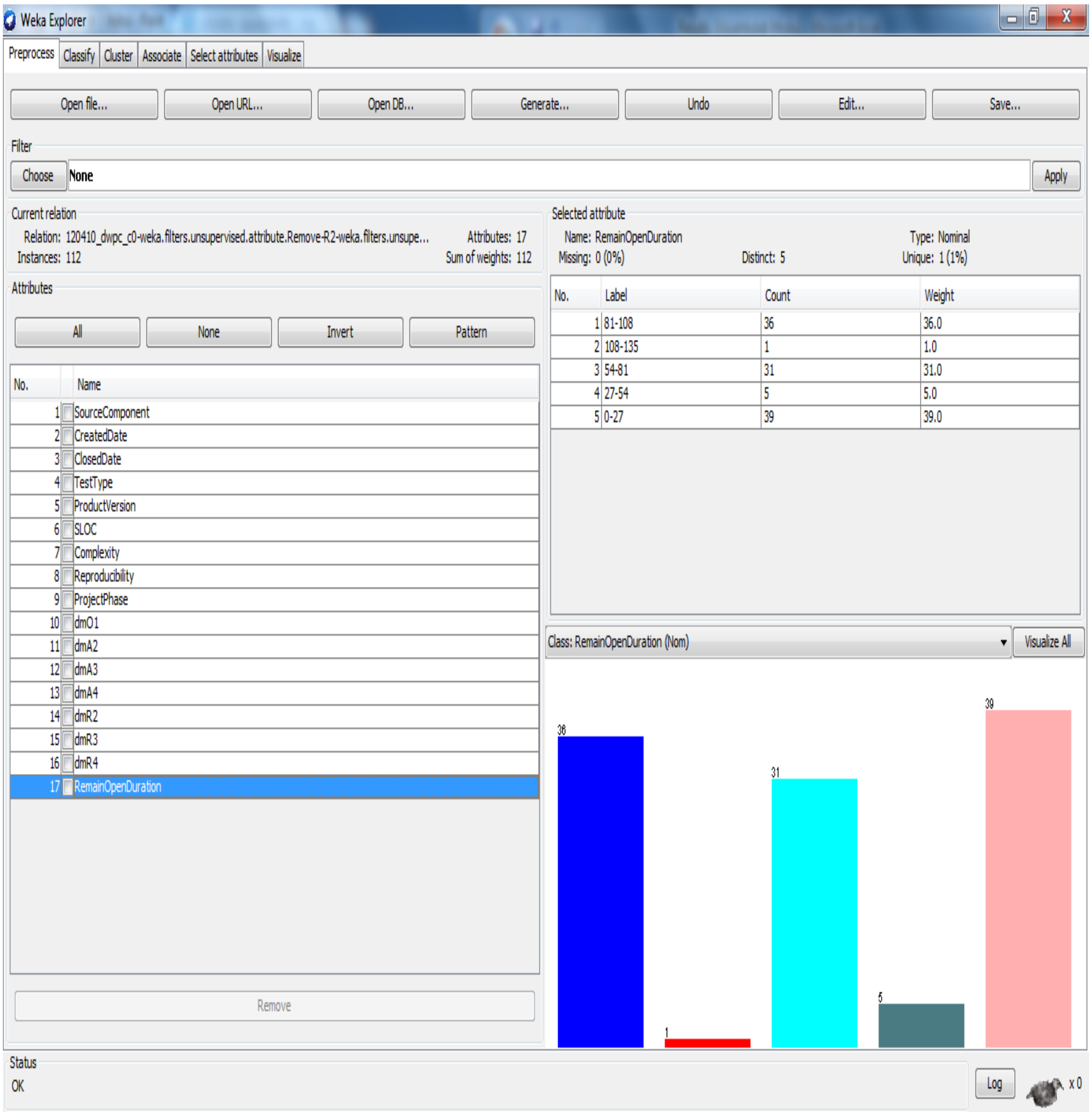

Figure C.2 Weka View of Case Study 1BCluster 0

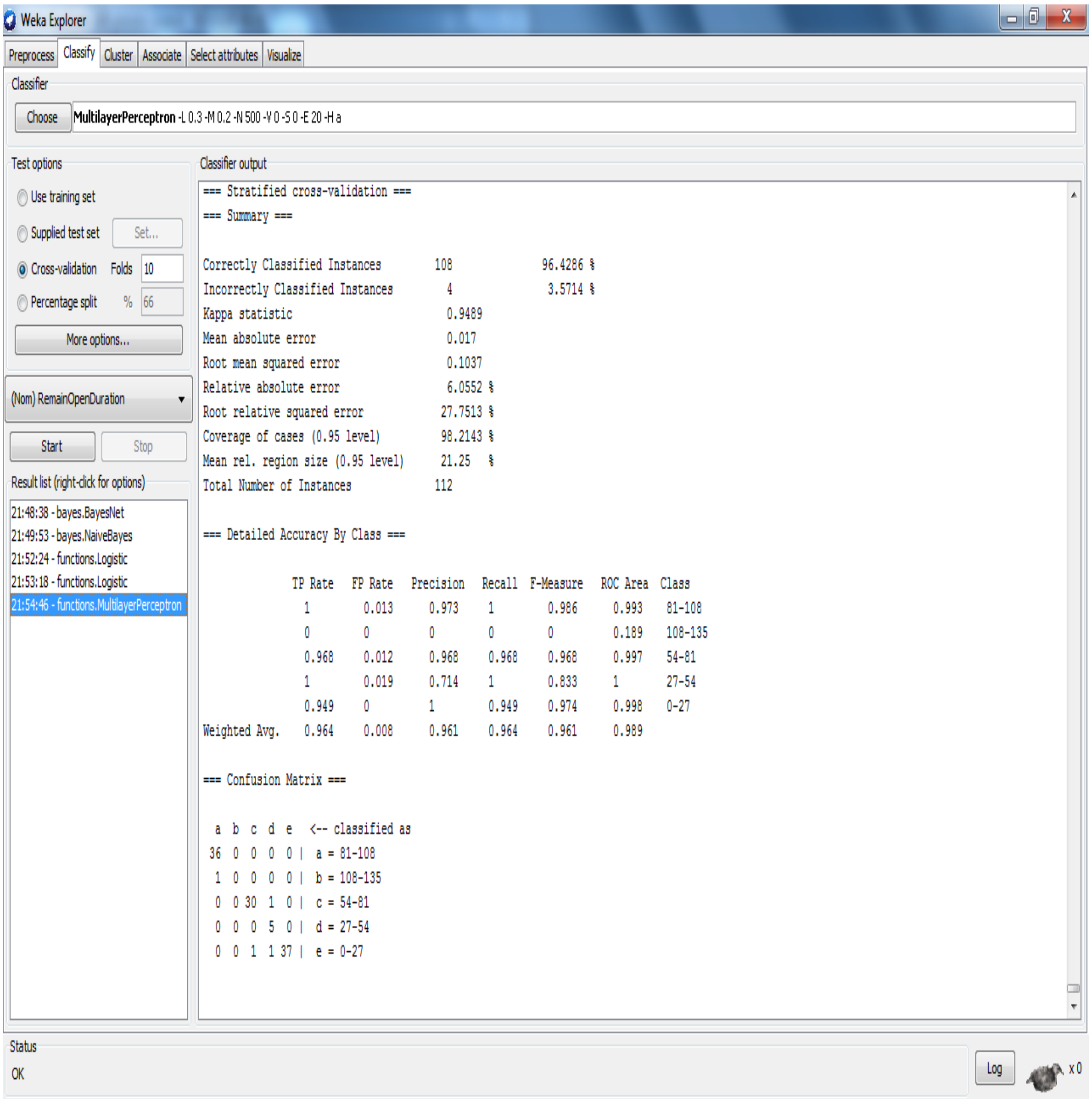

Figure C.3 Multilayer Perceptron Results of Case Study 1BCluster 0

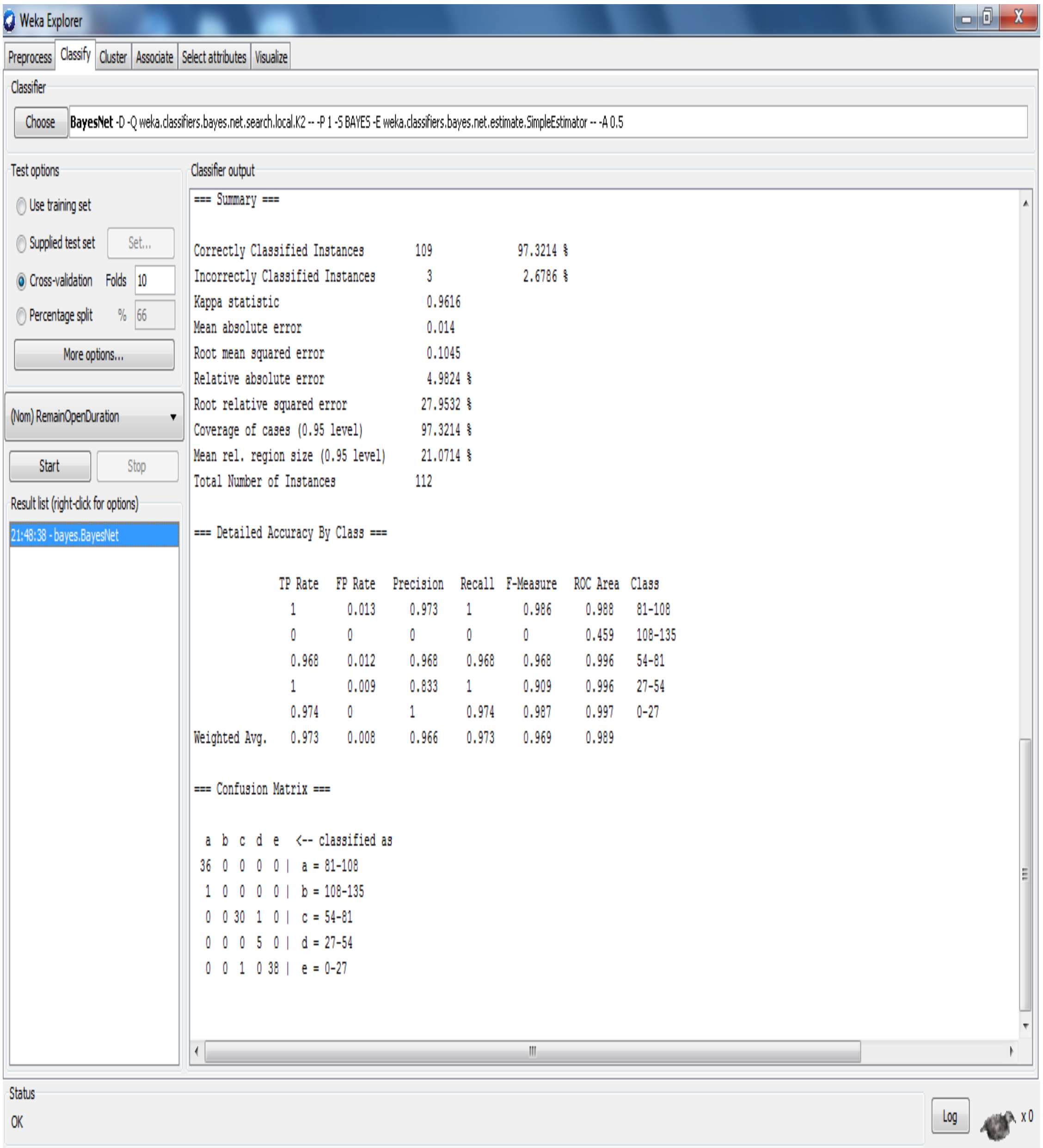

Figure C.4 BayesNet Results of Case Study 1BCluster 0

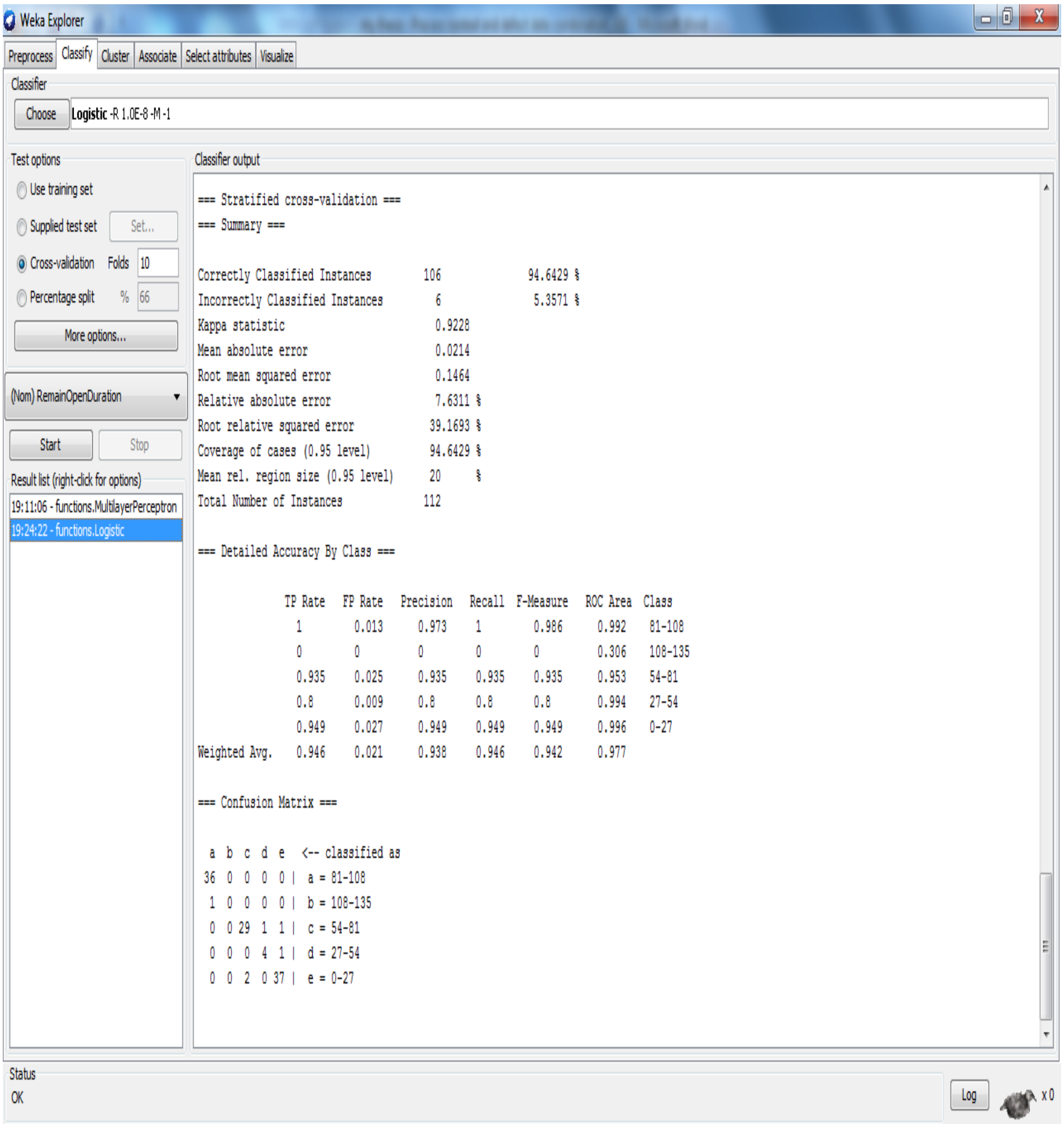

Figure C.5 Logistic Results of Case Study 1BCluster 0

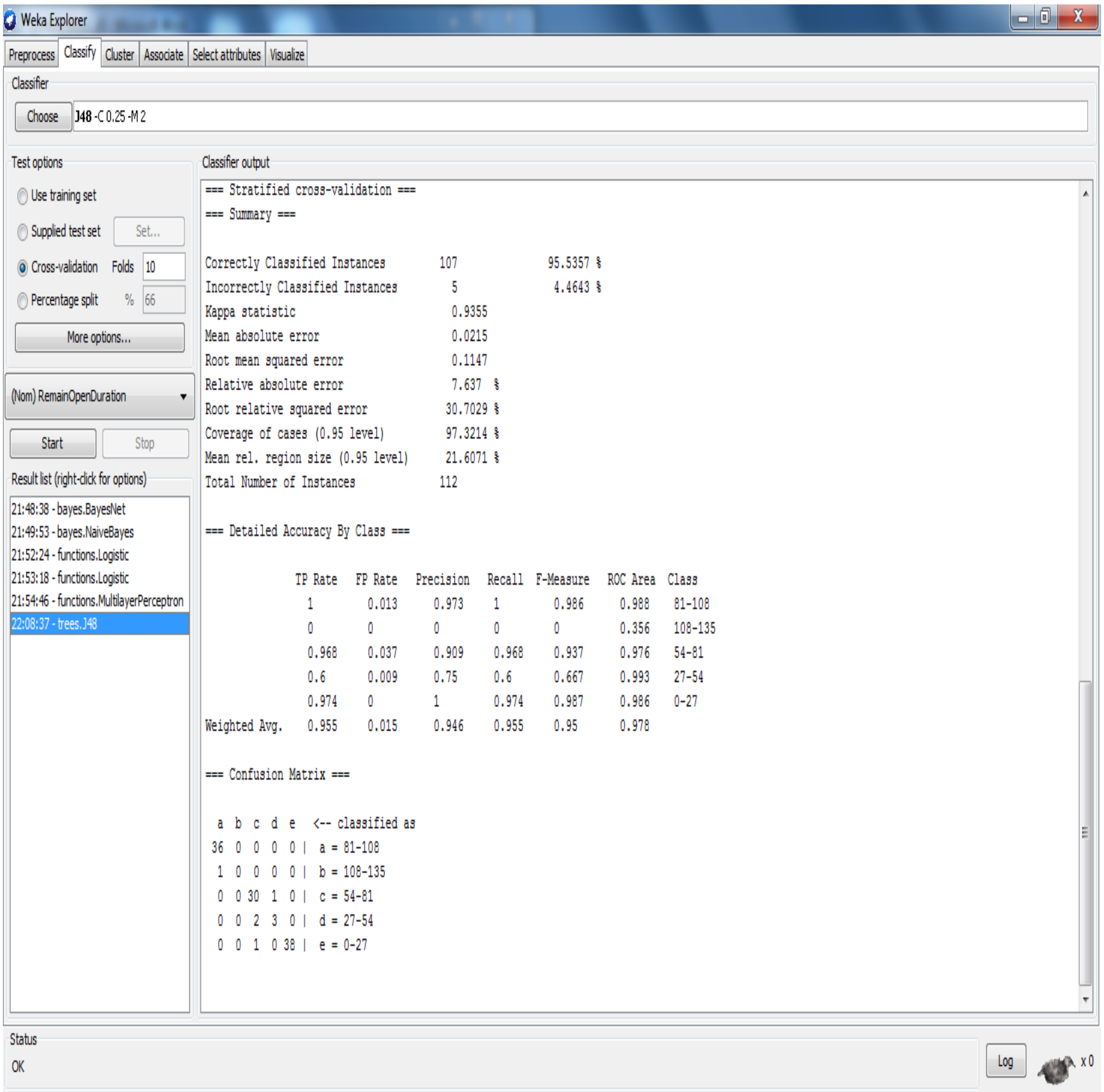

Figure C.6 J48 Results of Case Study 1BCluster 0

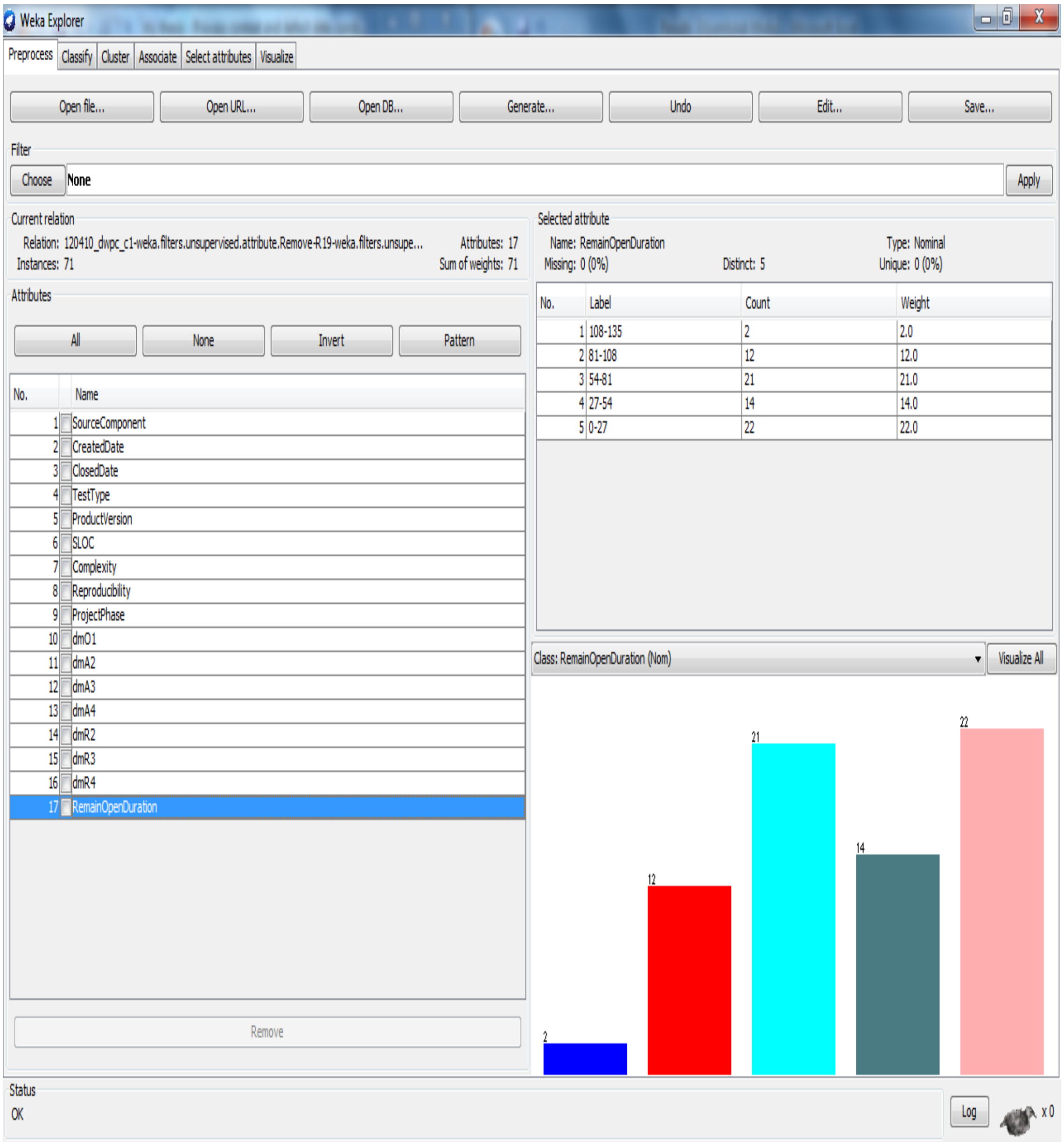

Figure C.7 Weka View Results of Case Study 1BCluster 1

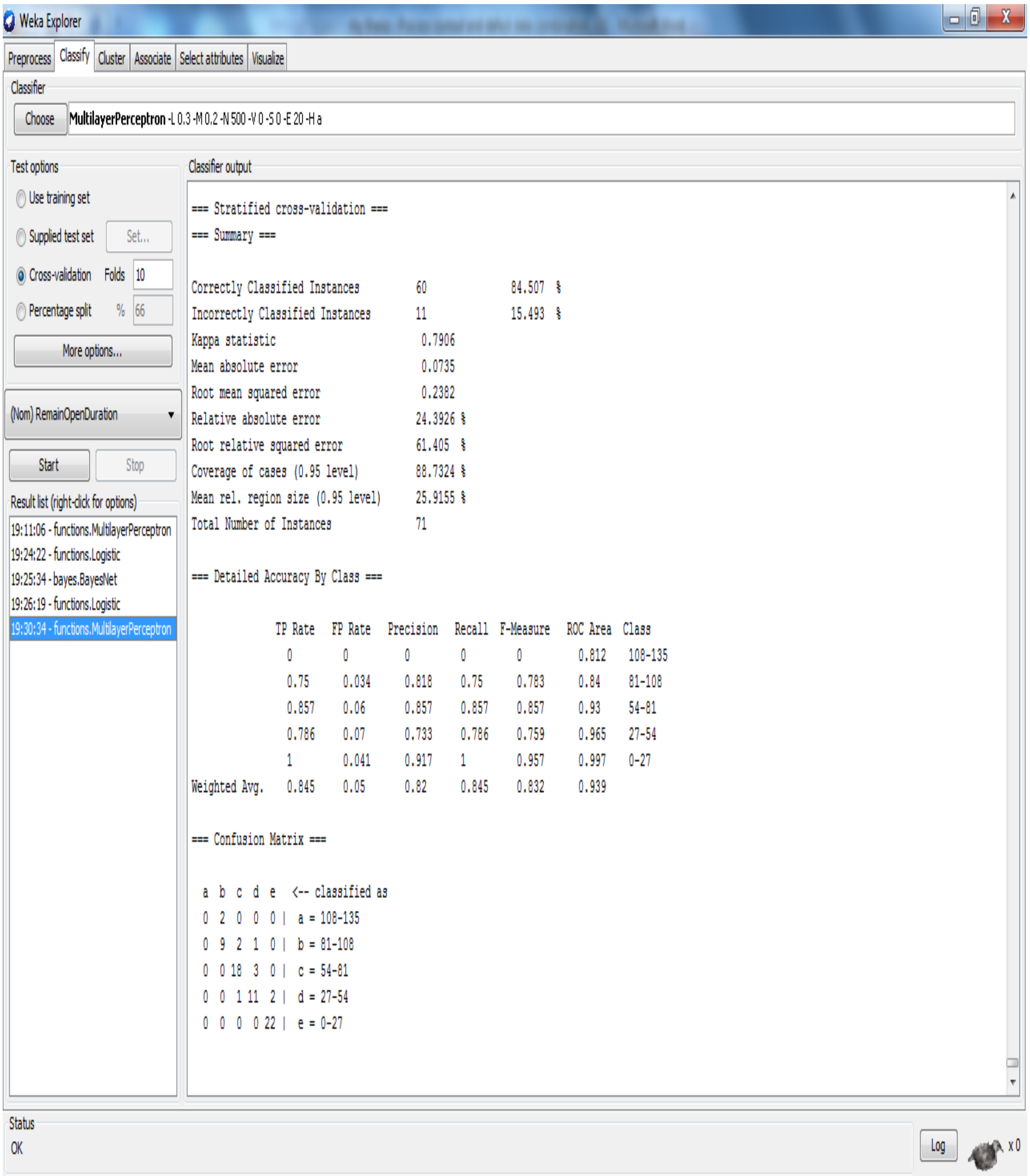

Figure C.8 Multilayer Perceptron Results of Case Study 1BCluster 1

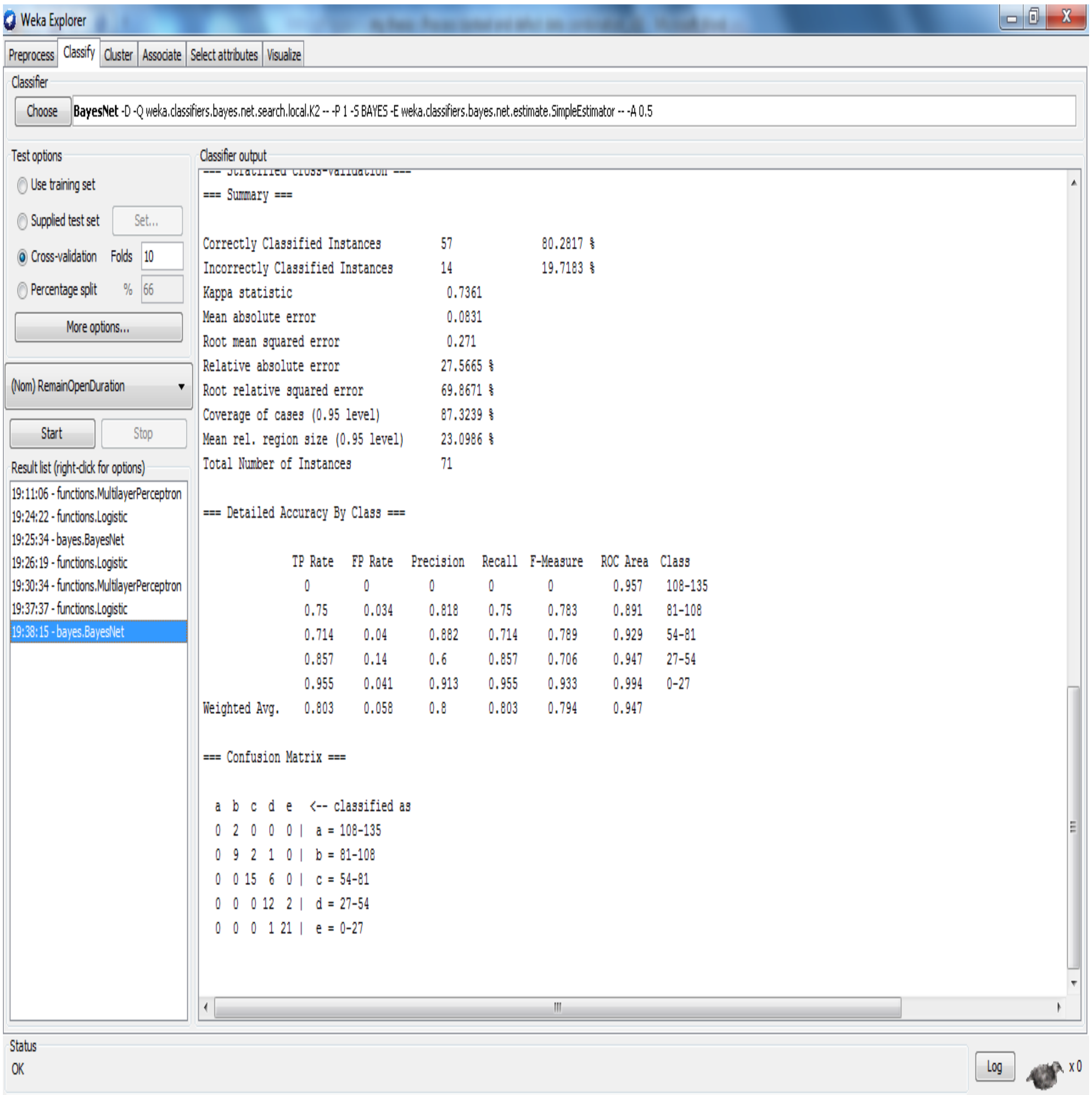

Figure C.9 BayesNet Results of Case Study 1BCluster 1

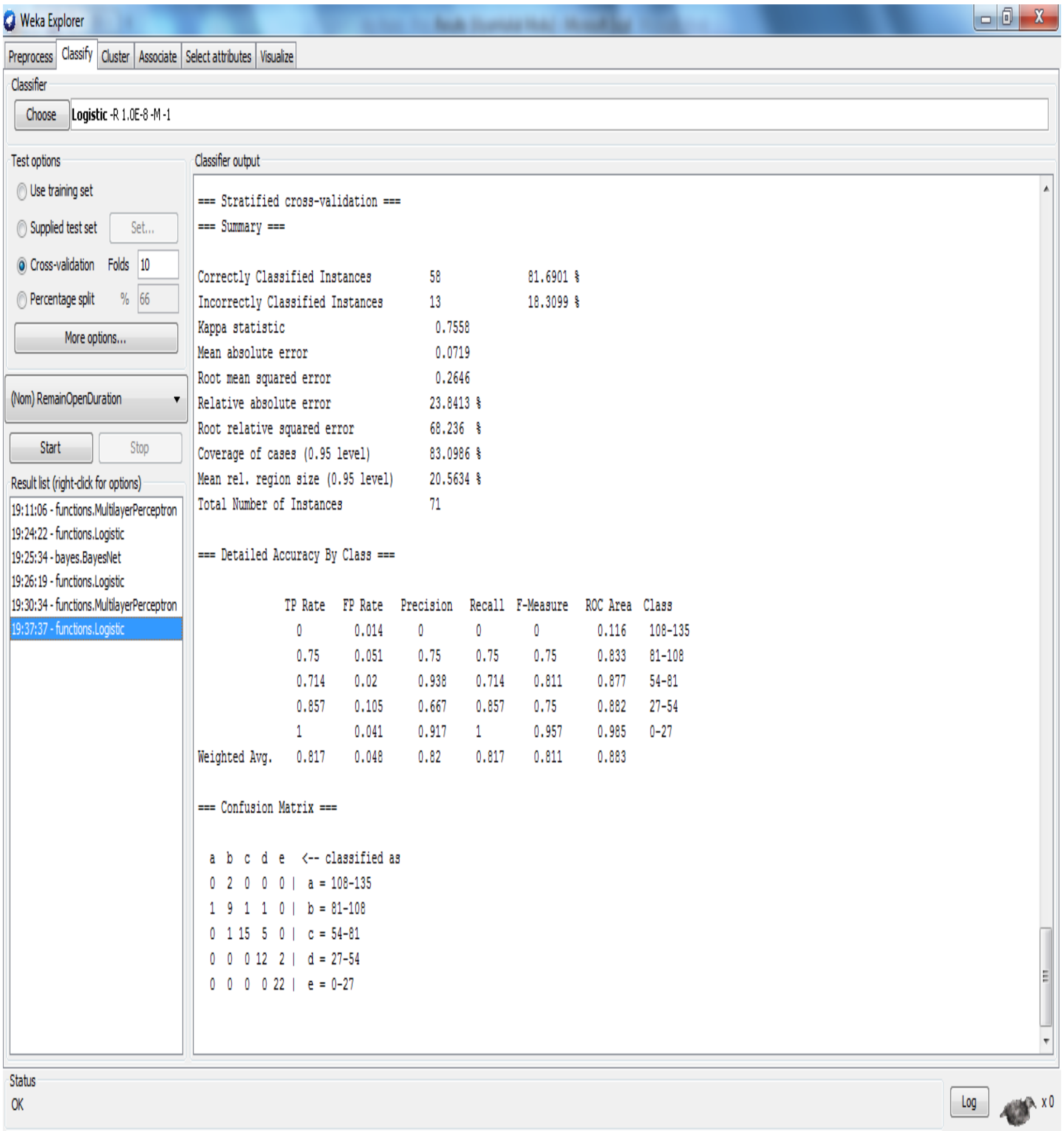

Figure C.10 Logistic Results of Case Study 1BCluster 1

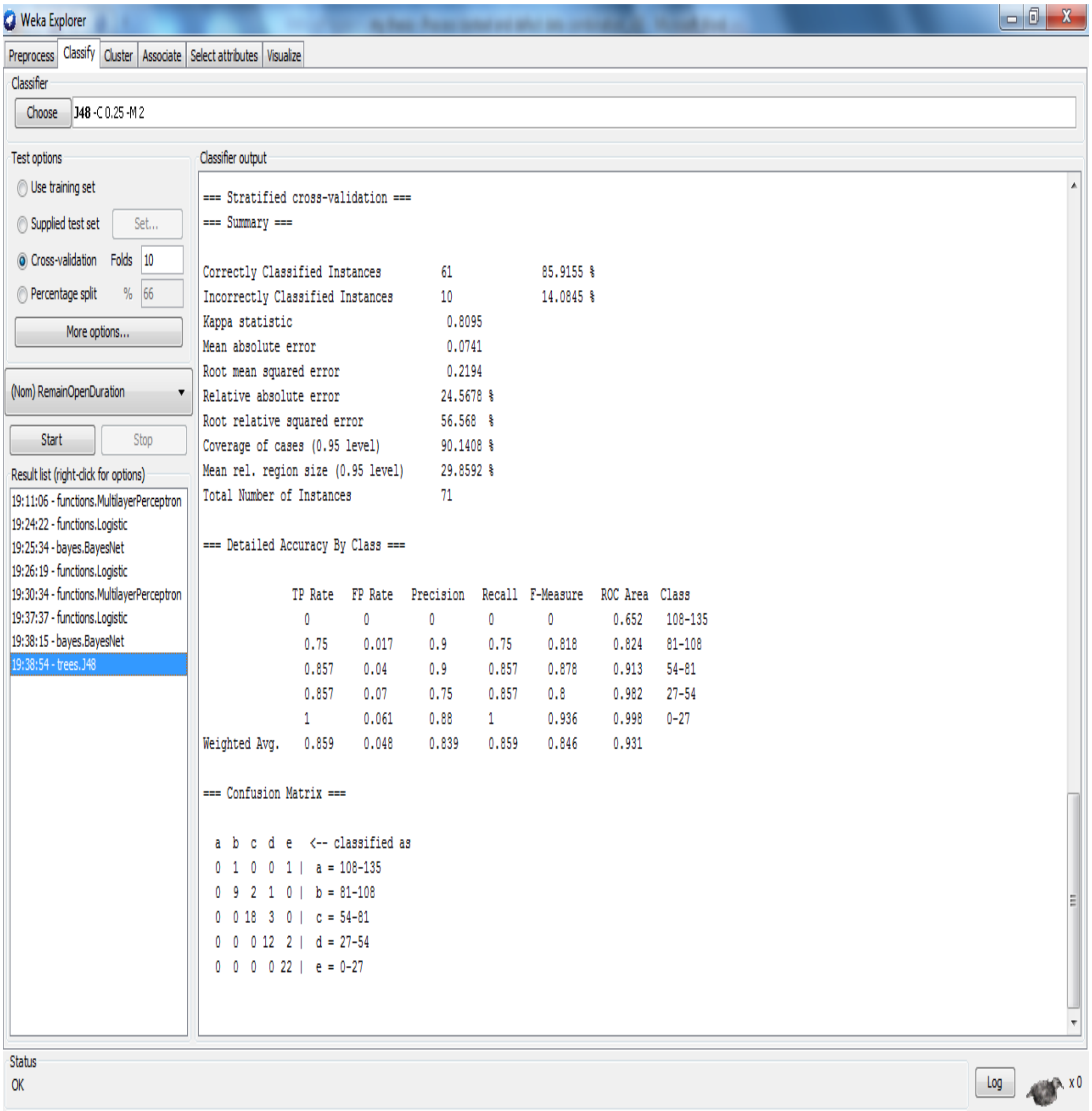

Figure C.11 J48 Results of Case Study 1BCluster 1

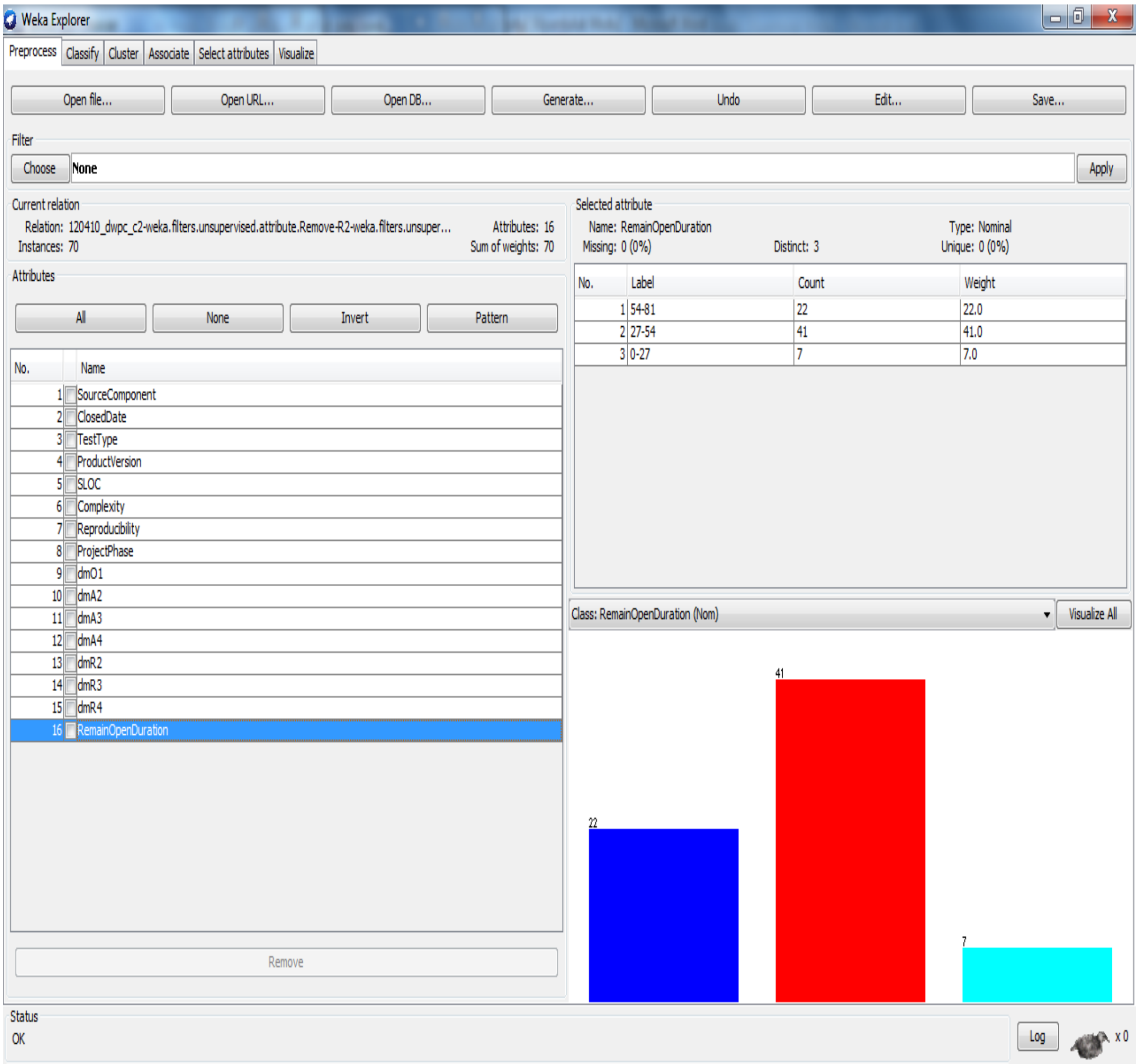

Figure C.12 Weka View of Case Study 1BCluster 2

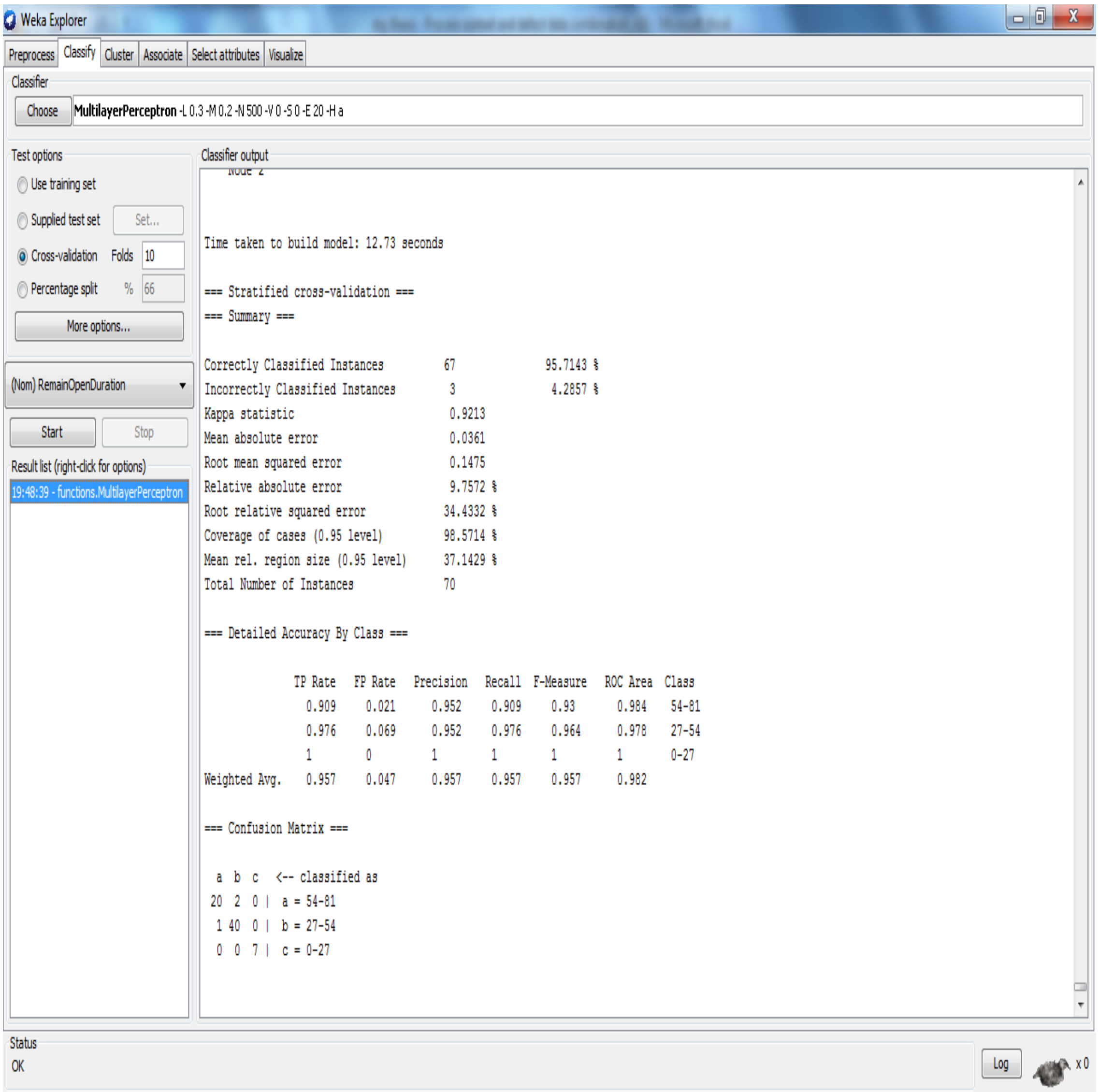

Figure C.13 Multilayer Perceptron Results of Case Study 1BCluster 2

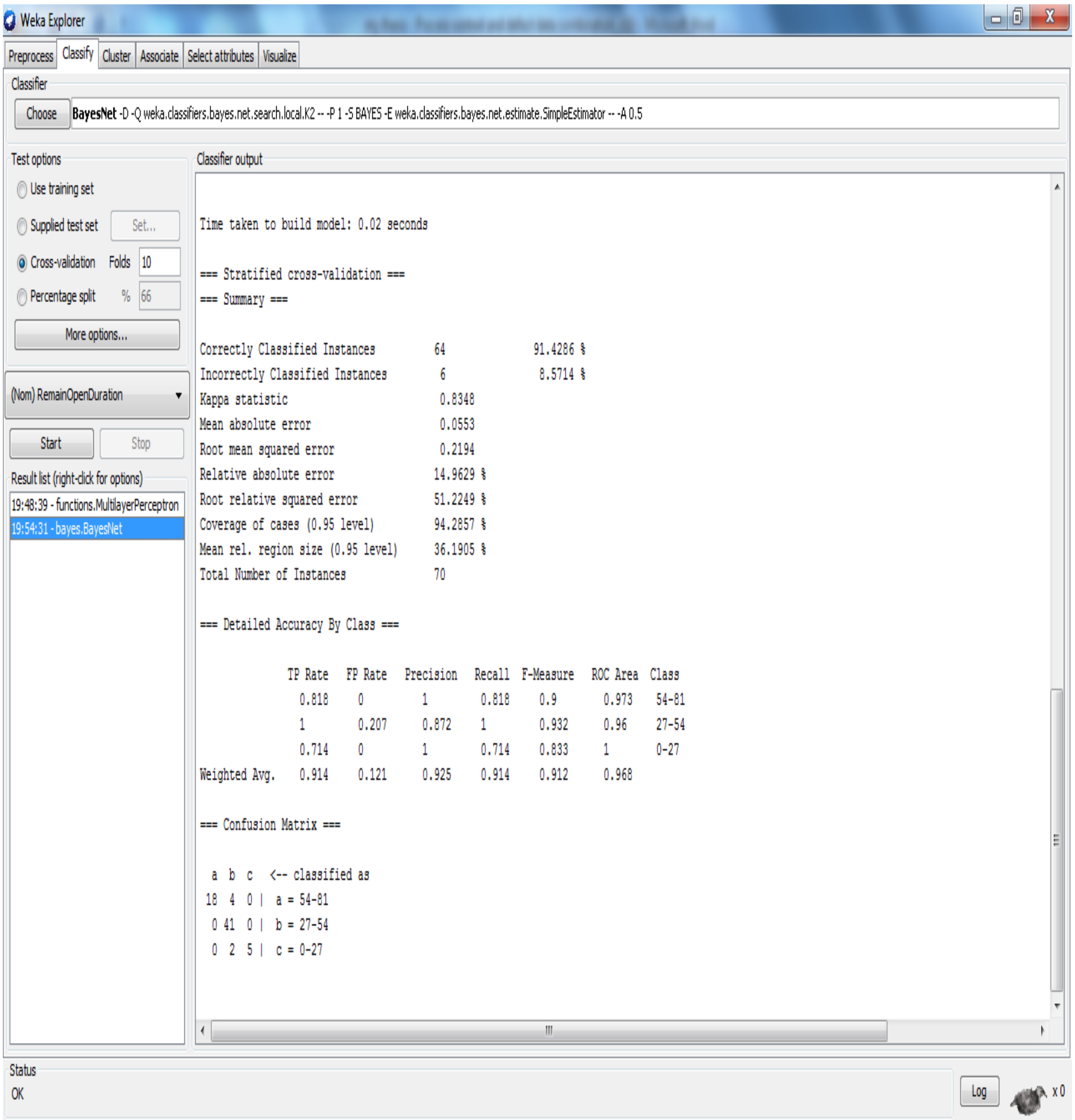

Figure C.14 BayesNet Results of Case Study 1BCluster 2

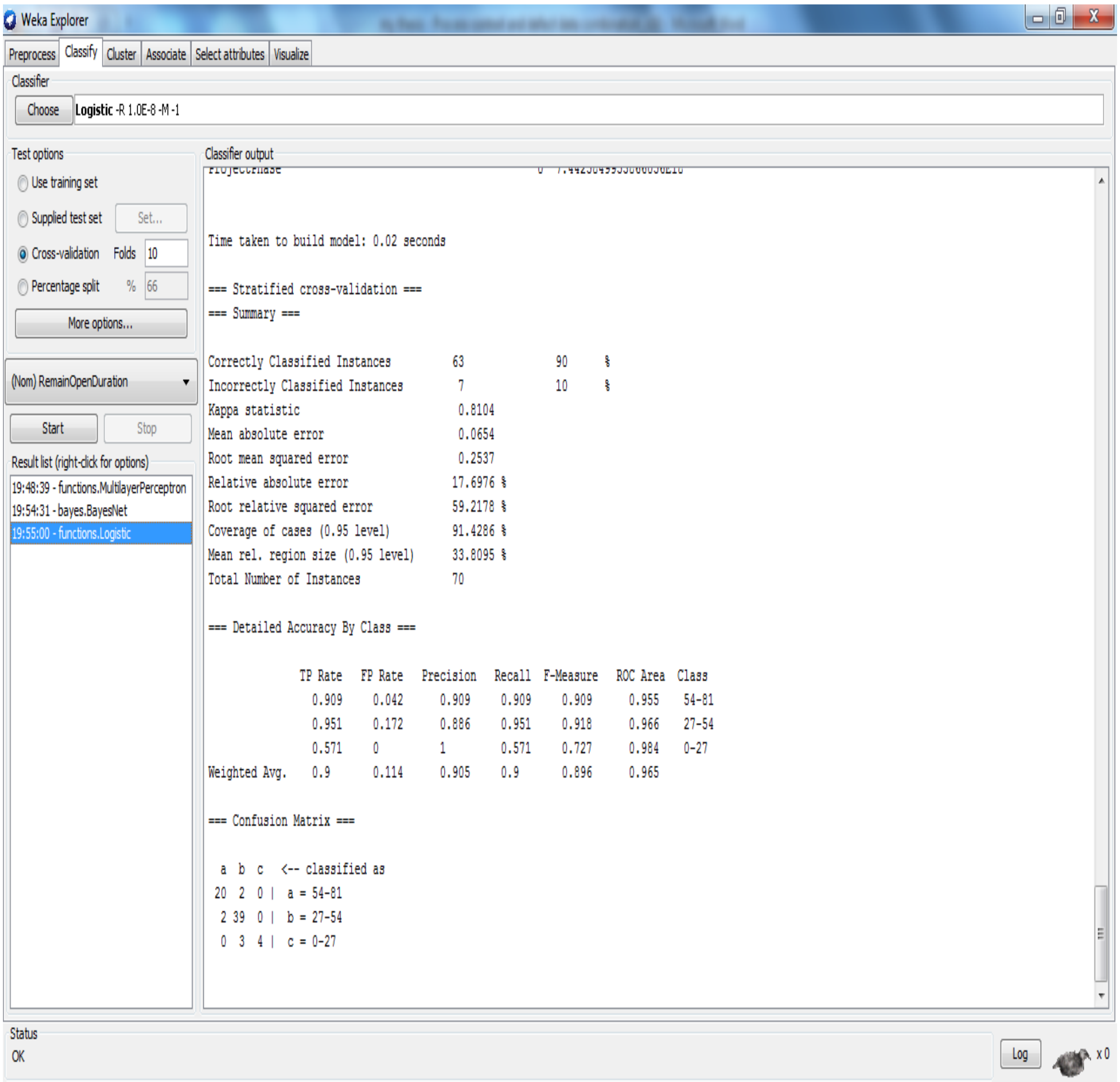

Figure C.15 Logistic Results of Case Study 1BCluster 2

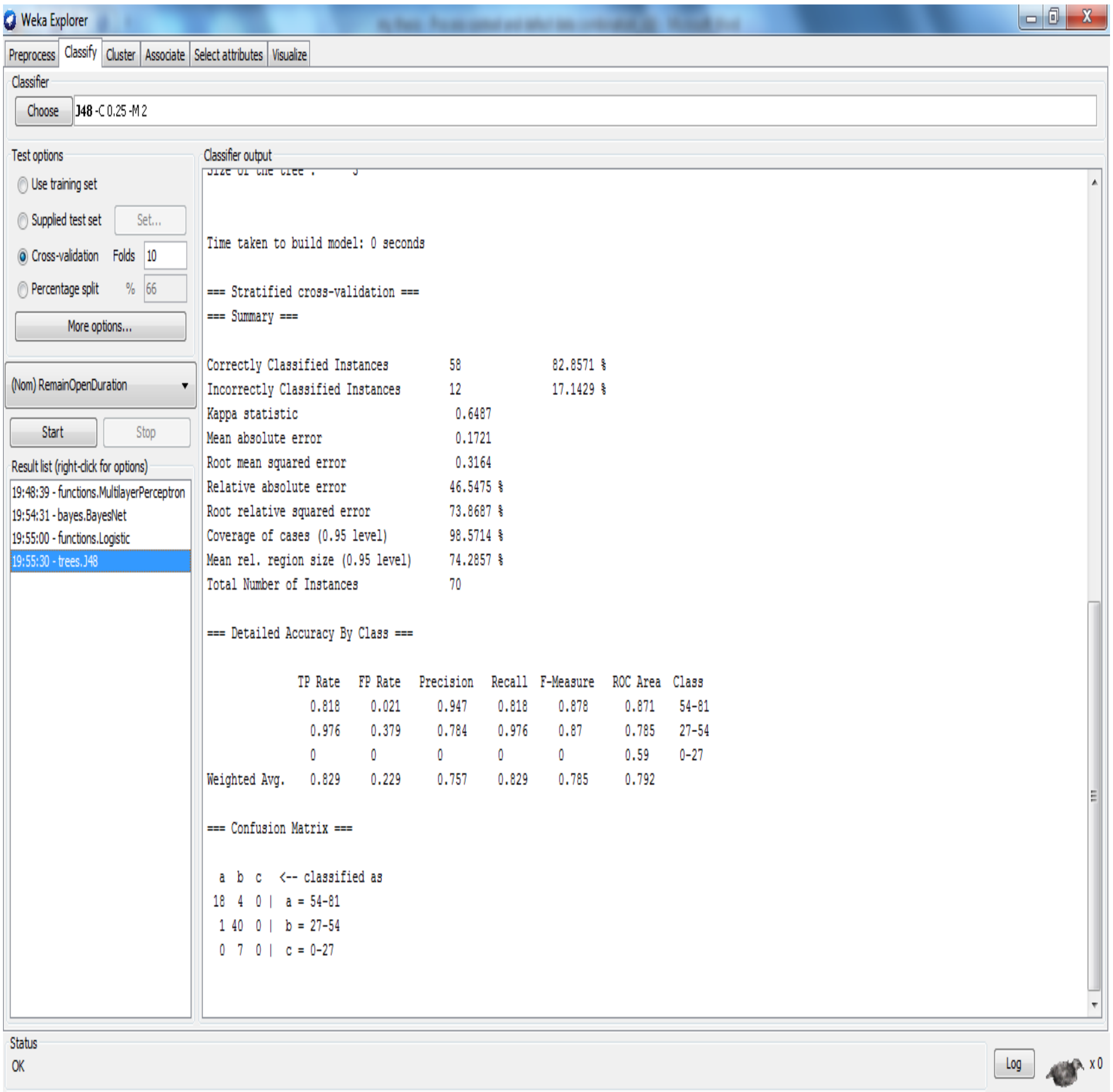

Figure C.16 J48 Results of Case Study 1BCluster 2

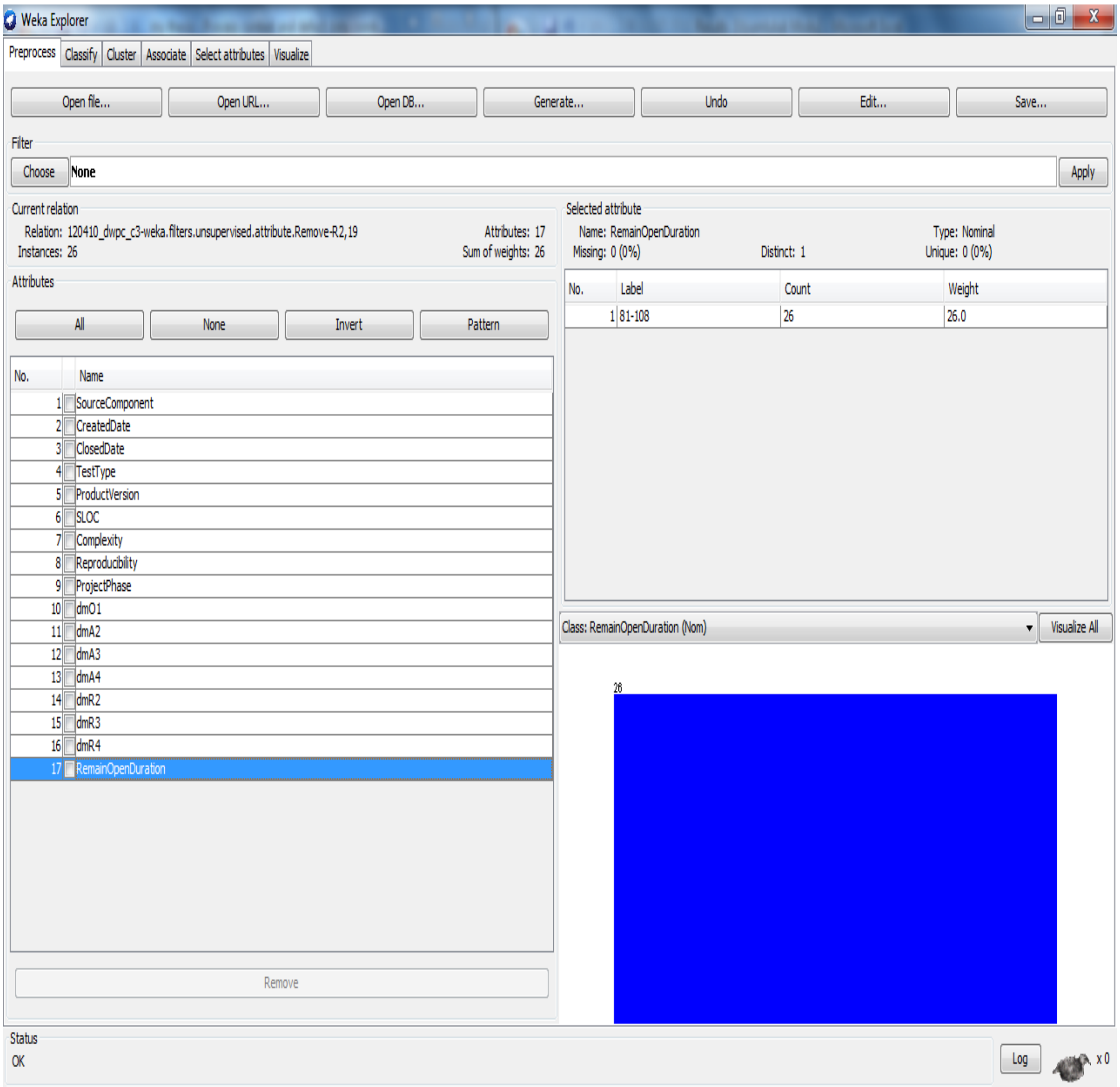

Figure C.17 Weka View of Case Study 1BCluster 3

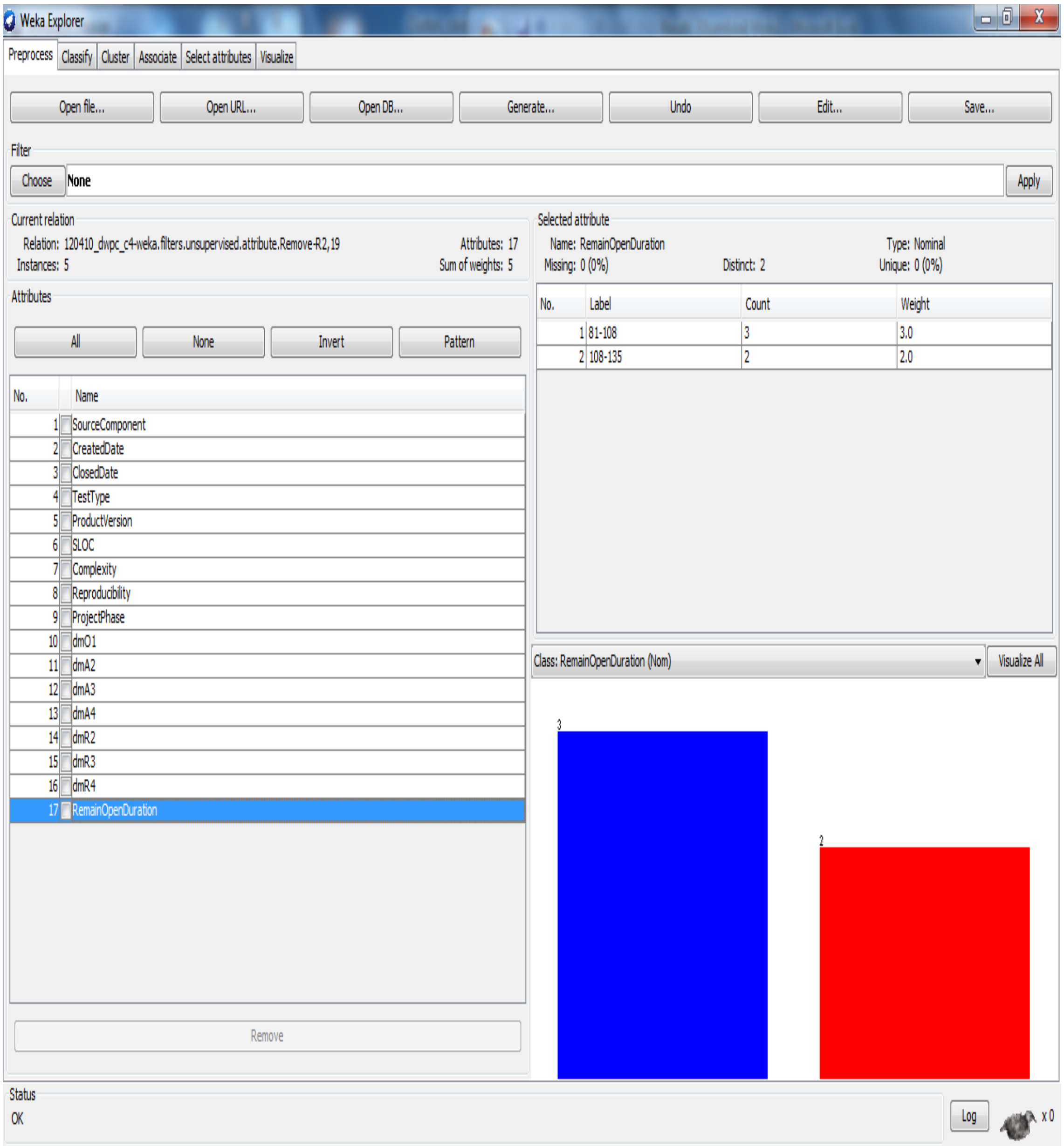

Figure C.18 Weka View of Case Study 1BCluster 4

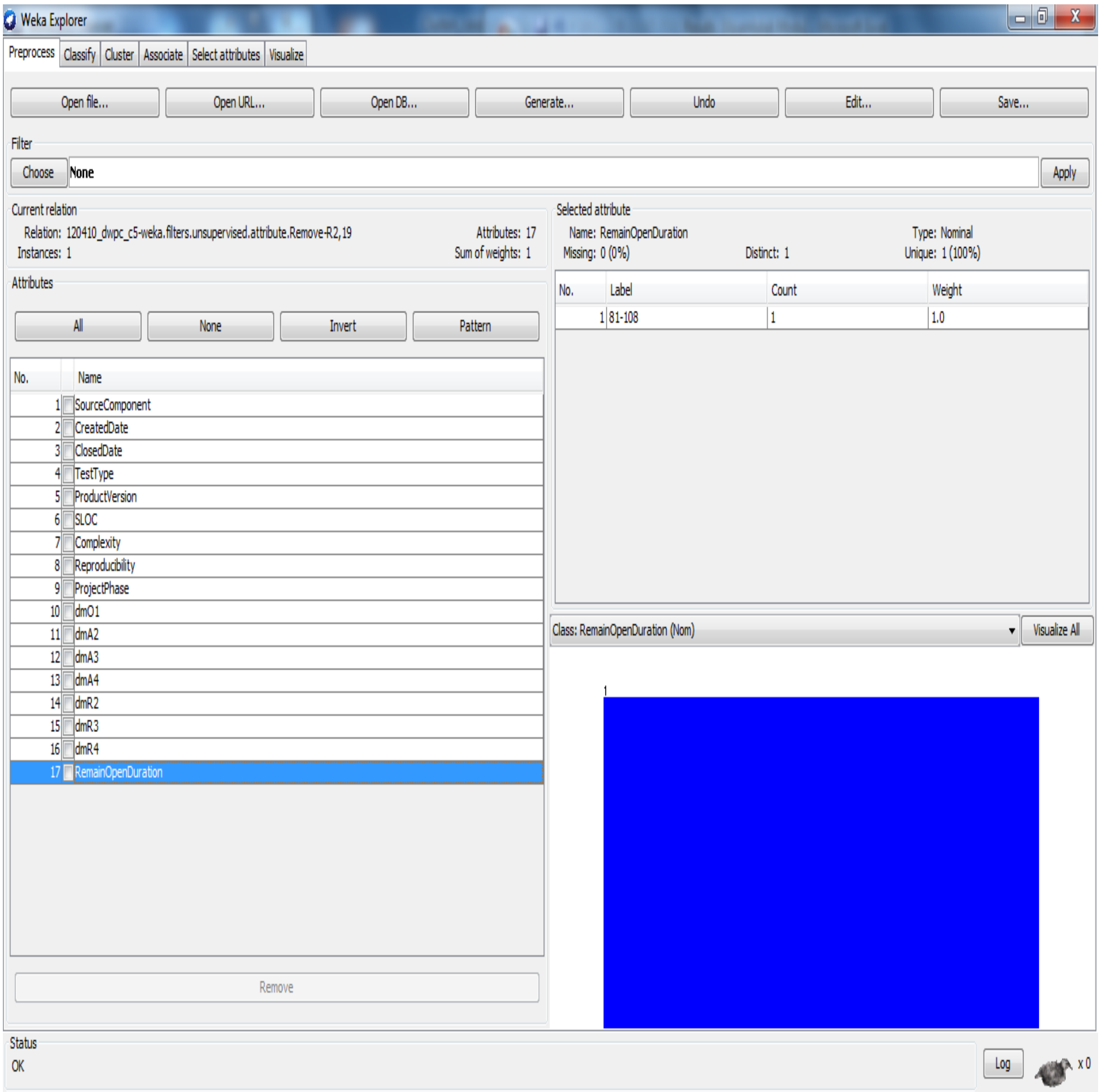

Figure C.19 Weka View of Case Study 1BCluster 5

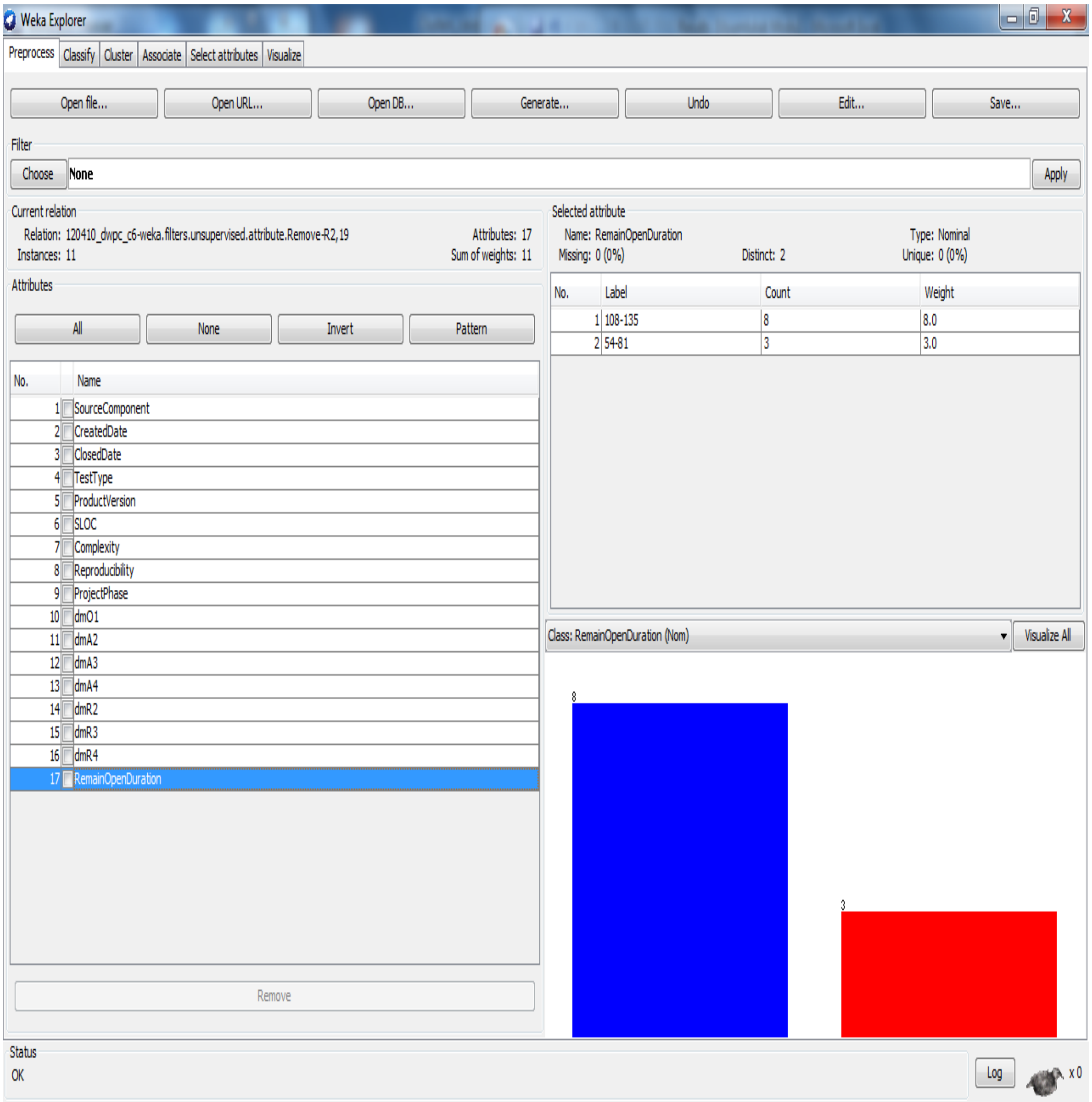

Figure C.20 Weka View of Case Study 1BCluster 6

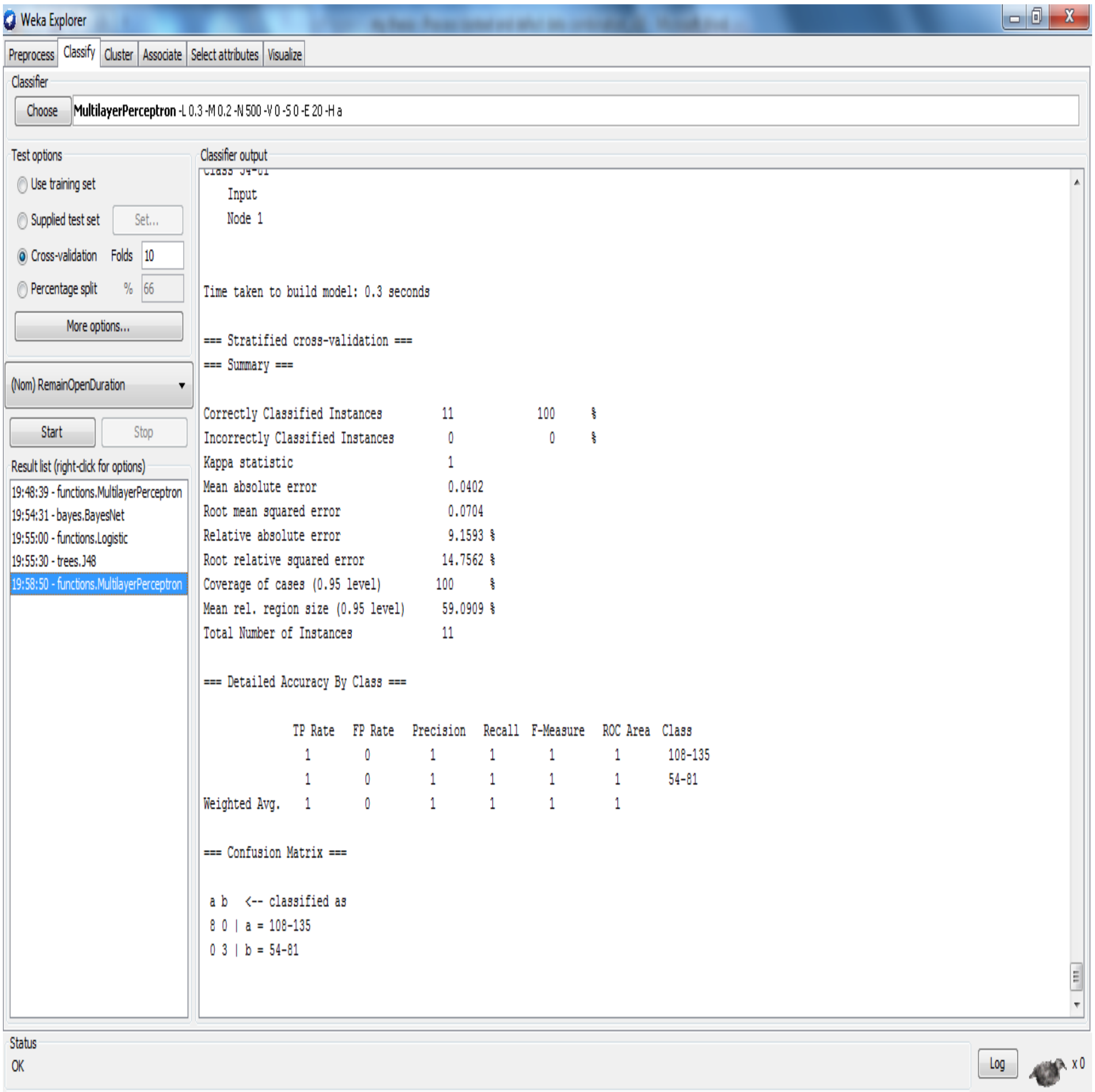

Figure C.21 Multilayer Perceptron Results of Case Study 1BCluster 6

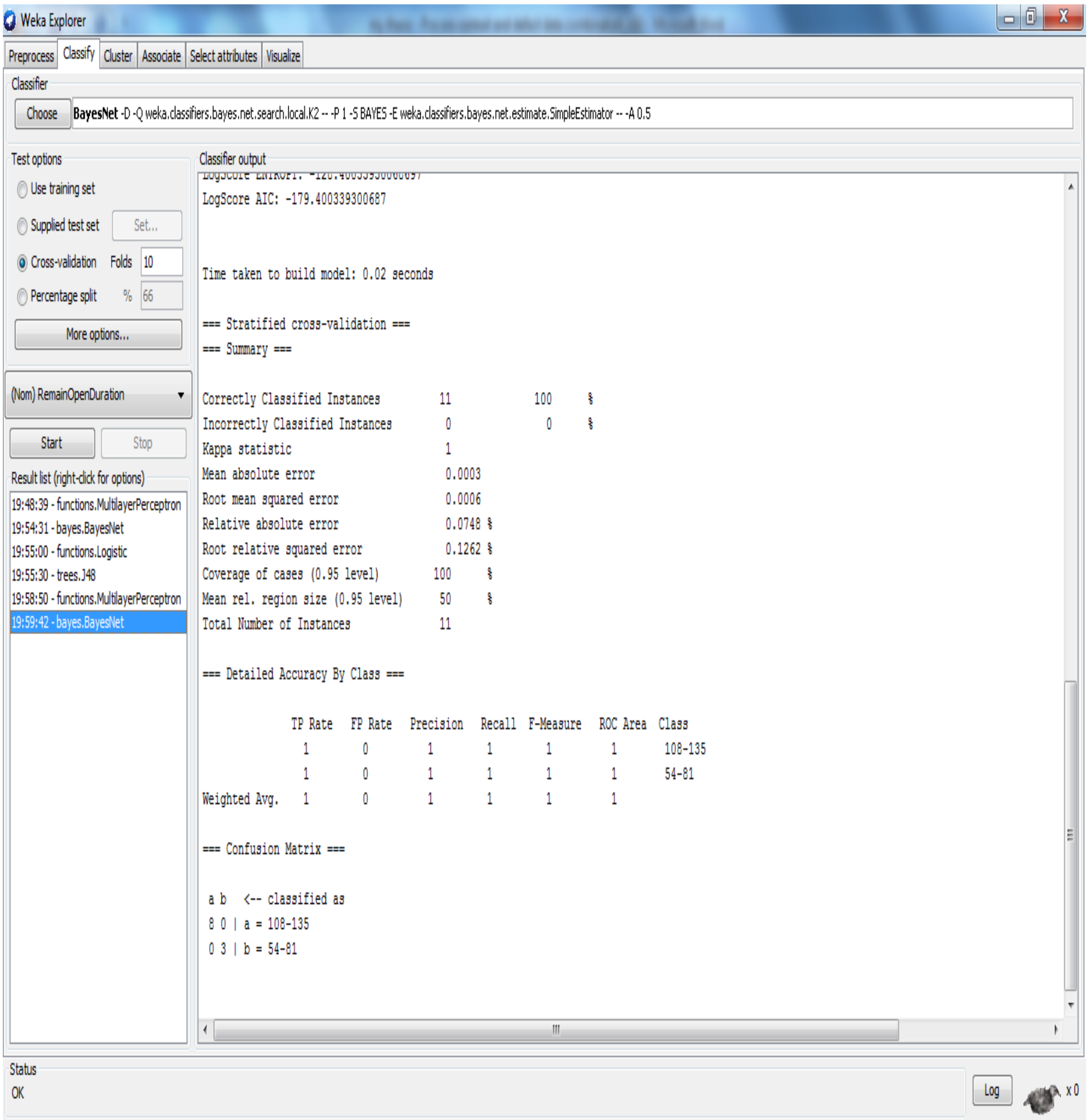

Figure C.22 BayesNet Results of Case Study 1BCluster 6

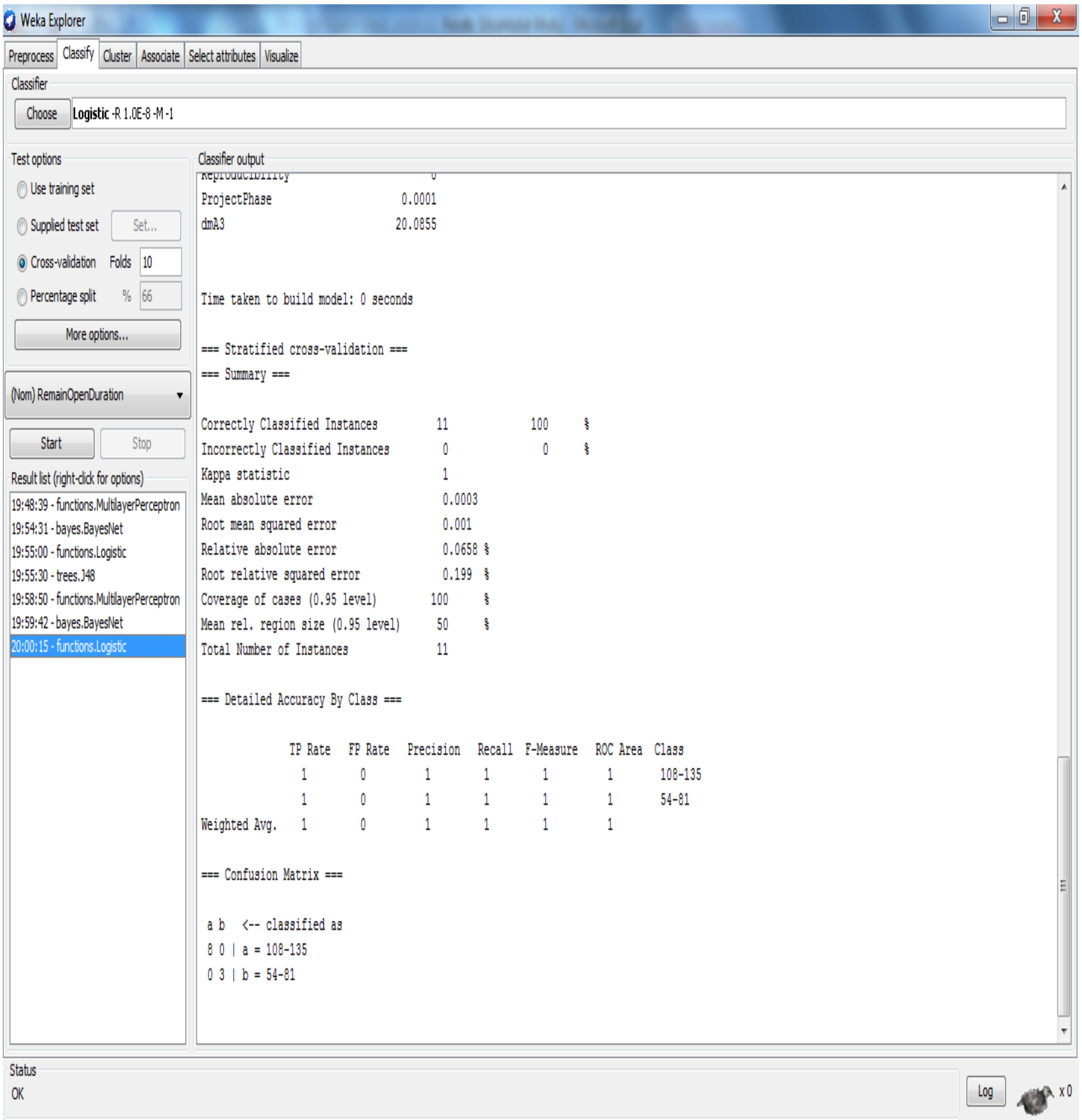

Figure C.23 Logistic Results of Case Study 1BCluster 6

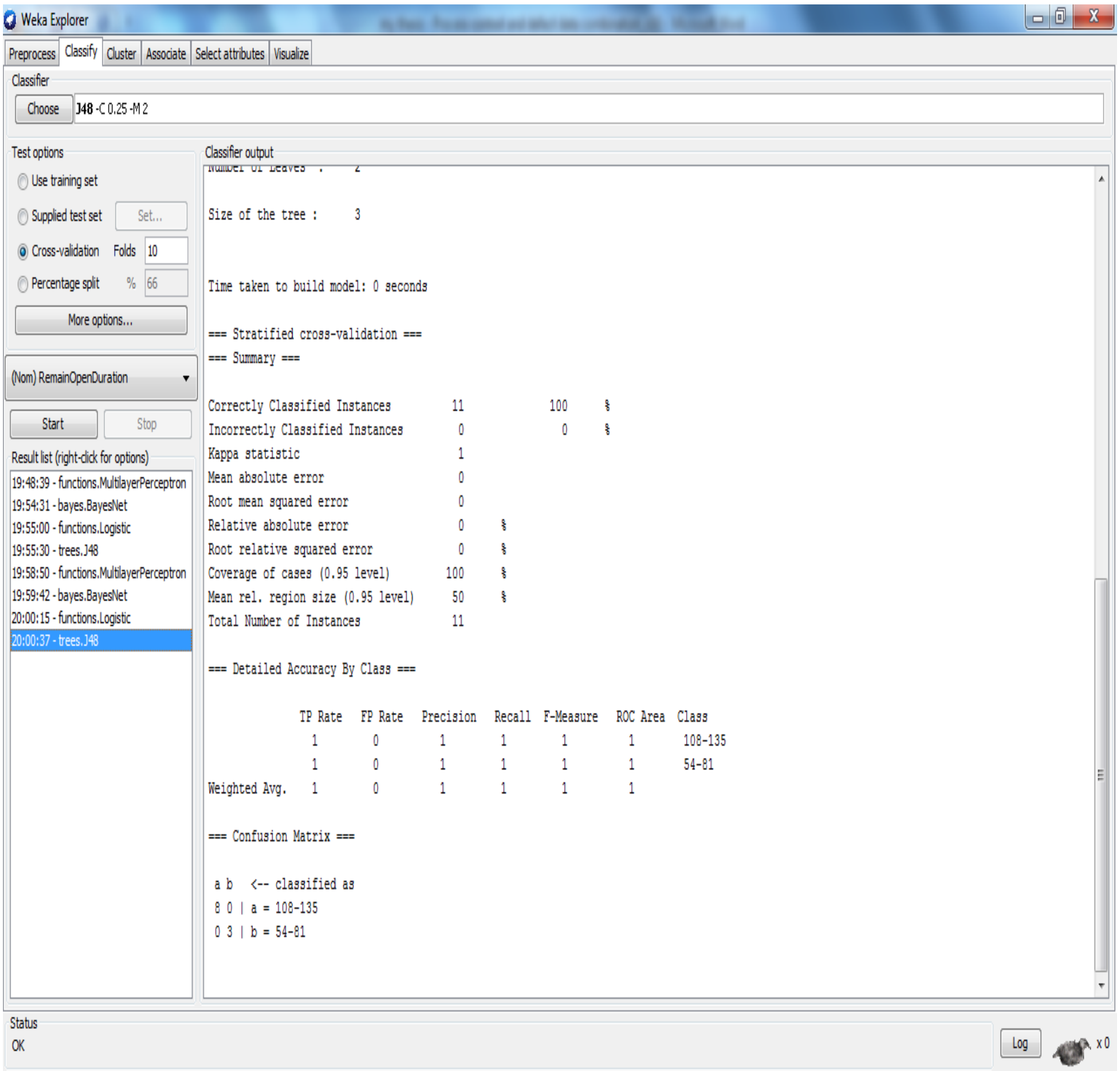

Figure C.24 J48 Results of Case Study 1BCluster 6

D. DETAILS OF CASE STUDY 2A

Metric Name: Detected SCU name					
Attributes			Answers	Rating	Expected Answers
	Indicators				
Measure Identity				N	
	Q1	Which entity does the measure measure?	Product		
	Q2	Which attribute of the entity does the measure measure?	Defective unis of product		
	Q3	What is the scale of the measurement data? (nominal, ordinal, interval, ratio, absolute)	Nominal		Ratio, Absolute
	Q4	What is the unit of the measurement data?	Not applicable		
	Q5	What is the type of the measurement data? (integer, real, etc.)	Text		
	Q6	What is the range of the measurement data?	Fourteen configuration units of		
Data Existence				F	
	Q7	Is measurement data existent?	Yes		
	Q8	What is the amount of overall observations?	425	√	Available > 20
	Q9	What is the amount of missing data points?	0		
	Q10	Are data points missing in periods? (If yes, please state observation numbers for missing periods)	0		
	Q11	Is measurement data time sequenced? (If no, please state how measurement data is sequenced)	Yes		
Data Verifiability				F	
	Q12	When is measurement data recorded in the process? (at start, middle, end, later, etc.)	At start		
	Q13	Is all measurement data recorded at the same place in the process? (at start, middle, end, later, etc.)	Yes	√	Yes
	Q14	Who is responsible for recording measurement data?	Test Specialist		
	Q15	Is all measurement data recorded by the responsible body?	Yes	√	Yes
	Q16	How is measurement data recorded? (on a form, report, tool, etc.)	Tool		
	Q17	Is all measurement data recorded the same way? (on a form, report, tool, etc.)	Yes	√	Yes
	Q18	Where is measurement data stored? (in a file, database, etc.)	The tool's database		
	Q19	Is all measurement data stored in the same place? (in a file, database, etc.)	Yes	√	Yes
Data Dependability				P	
	Q20	What is the frequency of generating measurement data? (asynchronously, daily, weekly, monthly, etc.)	Asynchronously		
	Q21	What is the frequency of recording measurement data? (asynchronously, daily, weekly, monthly, etc.)	Synchronously		
	Q22	What is the frequency of storing measurement data? (asynchronously, daily, weekly, monthly, etc.)	Synchronously		
	Q23	Are the frequencies for data generation, recording, and storing different?	No	√	No
	Q24	Is measurement data recorded precisely?	Yes	√	Yes
	Q25	Is measurement data collected for a specific purpose?	No	√	Yes
	Q26	Is the purpose of measurement data collection known by process performers?	No	√	Yes
	Q27	Is measurement data analyzed and reported?	No		Yes
	Q28	Is measurement data analysis results communicated to process performers?	No		Yes
	Q29	Is measurement data analysis results communicated to management?	No		Yes
	Q30	Is measurement data analysis results used as a basis for decision making?	No		Yes
Data Normalizability					
	Q31	Can measurement data be normalized by parameters or measures? (If yes, please specify them)	No		
Data Integrability					
	Q32	Is measurement data integrable at project level?	No		
	Q33	Is measurement data integrable at organization level?	No		

Figure D.1 MUQ for “Detected SCU Name” Basic Metric of Project-2

Metric Name: Source component				
Attributes		**Answers**	**Rating**	**Expected Answers**
	Indicators			
Measure Identity			N	
Q1	Which entity does the measure measure?	Product		
Q2	Which attribute of the entity does the measure measure?	Defective components of product		
Q3	What is the scale of the measurement data? (nominal, ordinal, interval, ratio, absolute)	Nominal		Ratio, Absolute
Q4	What is the unit of the measurement data?	Not applicable		
Q5	What is the type of the measurement data? (integer, real, etc.)	Text		
Q6	What is the range of the measurement data?	Seven component types		
Data Existence			F	
Q7	Is measurement data existent?	Yes		
Q8	What is the amount of overall observations?	425	√	Available > 20
Q9	What is the amount of missing data points?	0		
Q10	Are data points missing in periods? (If yes, please state observation numbers for missing periods)	0		
Q11	Is measurement data time sequenced? (If no, please state how measurement data is sequenced)	Yes		
Data Verifiability			F	
Q12	When is measurement data recorded in the process? (at start, middle, end, later, etc.)	At start		
Q13	Is all measurement data recorded at the same place in the process? (at start, middle, end, later, etc.)	Yes	√	Yes
Q14	Who is responsible for recording measurement data?	Test Specialist		
Q15	Is all measurement data recorded by the responsible body?	Yes	√	Yes
Q16	How is measurement data recorded? (on a form, report, tool, etc.)	Tool		
Q17	Is all measurement data recorded the same way? (on a form, report, tool, etc.)	Yes	√	Yes
Q18	Where is measurement data stored? (in a file, database, etc.)	The tool's database		
Q19	Is all measurement data stored in the same place? (in a file, database, etc.)	Yes	√	Yes
Data Dependability			P	
Q20	What is the frequency of generating measurement data? (asynchronously, daily, weekly, monthly, etc.)	Asynchronously		
Q21	What is the frequency of recording measurement data? (asynchronously, daily, weekly, monthly, etc.)	Synchronously		
Q22	What is the frequency of storing measurement data? (asynchronously, daily, weekly, monthly, etc.)	Synchronously		
Q23	Are the frequencies for data generation, recording, and storing different?	No	√	No
Q24	Is measurement data recorded precisely?	Yes	√	Yes
Q25	Is measurement data collected for a specific purpose?	No	√	Yes
Q26	Is the purpose of measurement data collection known by process performers?	No	√	Yes
Q27	Is measurement data analyzed and reported?	No		Yes
Q28	Is measurement data analysis results communicated to process performers?	No		Yes
Q29	Is measurement data analysis results communicated to management?	No		Yes
Q30	Is measurement data analysis results used as a basis for decision making?	No		Yes
Data Normalizability				
Q31	Can measurement data be normalized by parameters or measures? (If yes, please specify them)	No		
Data Integrability				
Q32	Is measurement data integrable at project level?	No		
Q33	Is measurement data integrable at organization level?	No		

Figure D.2 MUQ for “Source Component” Basic Metric of Project-2

Metric Name: Created date				
Attributes		**Answers**	**Rating**	**Expected Answers**
	Indicators			
Measure Identity			N	
Q1	Which entity does the measure measure?	Process		
Q2	Which attribute of the entity does the measure measure?	The date of the defect record		
Q3	What is the scale of the measurement data? (nominal, ordinal, interval, ratio, absolute)	Nominal		Ratio, Absolute
Q4	What is the unit of the measurement data?	Time (dd.mm.yy hh:mm)		
Q5	What is the type of the measurement data? (integer, real, etc.)	Date		
Q6	What is the range of the measurement data?	00.00.0000 00:00		
Data Existence			F	
Q7	Is measurement data existent?	Yes		
Q8	What is the amount of overall observations?	425	√	Available > 20
Q9	What is the amount of missing data points?	0		
Q10	Are data points missing in periods? (If yes, please state observation numbers for missing periods)	0		
Q11	Is measurement data time sequenced? (If no, please state how measurement data is sequenced)	Yes		
Data Verifiability			F	
Q12	When is measurement data recorded in the process? (at start, middle, end, later, etc.)	At start		
Q13	Is all measurement data recorded at the same place in the process? (at start, middle, end, later, etc.)	Yes	√	Yes
Q14	Who is responsible for recording measurement data?	Test Specialist		
Q15	Is all measurement data recorded by the responsible body?	Yes	√	Yes
Q16	How is measurement data recorded? (on a form, report, tool, etc.)	Tool		
Q17	Is all measurement data recorded the same way? (on a form, report, tool, etc.)	Yes	√	Yes
Q18	Where is measurement data stored? (in a file, database, etc.)	The tool's database		
Q19	Is all measurement data stored in the same place? (in a file, database, etc.)	Yes	√	Yes
Data Dependability			P	
Q20	What is the frequency of generating measurement data? (asynchronously, daily, weekly, monthly, etc.)	Asynchronously		
Q21	What is the frequency of recording measurement data? (asynchronously, daily, weekly, monthly, etc.)	Synchronously		
Q22	What is the frequency of storing measurement data? (asynchronously, daily, weekly, monthly, etc.)	Synchronously		
Q23	Are the frequencies for data generation, recording, and storing different?	No	√	No
Q24	Is measurement data recorded precisely?	Yes	√	Yes
Q25	Is measurement data collected for a specific purpose?	No	√	Yes
Q26	Is the purpose of measurement data collection known by process performers?	No	√	Yes
Q27	Is measurement data analyzed and reported?	No		Yes
Q28	Is measurement data analysis results communicated to process performers?	No		Yes
Q29	Is measurement data analysis results communicated to management?	No		Yes
Q30	Is measurement data analysis results used as a basis for decision making?	No		Yes
Data Normalizability				
Q31	Can measurement data be normalized by parameters or measures? (If yes, please specify them)	No		
Data Integrability				
Q32	Is measurement data integrable at project level?	No		
Q33	Is measurement data integrable at organization level?	No		

Figure D.3 MUQ for “Created Date” Basic Metric of Project-2

Metric Name: Closed date				
Attributes		**Answers**	**Rating**	**Expected Answers**
	Indicators			
Measure Identity			N	
Q1	Which entity does the measure measure?	Process		
Q2	Which attribute of the entity does the measure measure?	Closed date of the defect record		
Q3	What is the scale of the measurement data? (nominal, ordinal, interval, ratio, absolute)	Nominal		Ratio, Absolute
Q4	What is the unit of the measurement data?	Time (dd.mm.yy hh:mm)		
Q5	What is the type of the measurement data? (integer, real, etc.)	Date		
Q6	What is the range of the measurement data?	00.00.0000 00:00		
Data Existence			F	
Q7	Is measurement data existent?	Yes		
Q8	What is the amount of overall observations?	298	√	Available > 20
Q9	What is the amount of missing data points?	0		
Q10	Are data points missing in periods? (If yes, please state observation numbers for missing periods)	0		
Q11	Is measurement data time sequenced? (If no, please state how measurement data is sequenced)	Yes		
Data Verifiability			F	
Q12	When is measurement data recorded in the process? (at start, middle, end, later, etc.)	Later		
Q13	Is all measurement data recorded at the same place in the process? (at start, middle, end, later, etc.)	Yes	√	Yes
Q14	Who is responsible for recording measurement data?	Project Manager		
Q15	Is all measurement data recorded by the responsible body?	Yes	√	Yes
Q16	How is measurement data recorded? (on a form, report, tool, etc.)	Tool		
Q17	Is all measurement data recorded the same way? (on a form, report, tool, etc.)	Yes	√	Yes
Q18	Where is measurement data stored? (in a file, database, etc.)	The tool's database		
Q19	Is all measurement data stored in the same place? (in a file, database, etc.)	Yes	√	Yes
Data Dependability			P	
Q20	What is the frequency of generating measurement data? (asynchronously, daily, weekly, monthly, etc.)	Asynchronously		
Q21	What is the frequency of recording measurement data? (asynchronously, daily, weekly, monthly, etc.)	Asynchronously		
Q22	What is the frequency of storing measurement data? (asynchronously, daily, weekly, monthly, etc.)	Synchronously		
Q23	Are the frequencies for data generation, recording, and storing different?	No	√	No
Q24	Is measurement data recorded precisely?	Yes	√	Yes
Q25	Is measurement data collected for a specific purpose?	No	√	Yes
Q26	Is the purpose of measurement data collection known by process performers?	No	√	Yes
Q27	Is measurement data analyzed and reported?	No		Yes
Q28	Is measurement data analysis results communicated to process performers?	No		Yes
Q29	Is measurement data analysis results communicated to management?	No		Yes
Q30	Is measurement data analysis results used as a basis for decision making?	No		Yes
Data Normalizability				
Q31	Can measurement data be normalized by parameters or measures? (If yes, please specify them)	No		
Data Integrability				
Q32	Is measurement data integrable at project level?	No		
Q33	Is measurement data integrable at organization level?	No		

Figure D.4 MUQ for “Closed Date” Basic Metric of Project-2

Metric Name: Test type				
Attributes		**Answers**	**Rating**	**Expected Answers**
	Indicators			
Measure Identity			N	
Q1	Which entity does the measure measure?	Process		
Q2	Which attribute of the entity does the measure measure?	Defect detection rate of different test activities		
Q3	What is the scale of the measurement data? (nominal, ordinal, interval, ratio, absolute)	Nominal		Ratio, Absolute
Q4	What is the unit of the measurement data?	Not applicable		
Q5	What is the type of the measurement data? (integer, real, etc.)	Text		
Q6	What is the range of the measurement data?	Two test activity types		
Data Existence			F	
Q7	Is measurement data existent?	Yes		
Q8	What is the amount of overall observations?	296	√	Available > 20
Q9	What is the amount of missing data points?	0		
Q10	Are data points missing in periods? (If yes, please state observation numbers for missing periods)	0		
Q11	Is measurement data time sequenced? (If no, please state how measurement data is sequenced)	Yes		
Data Verifiability			F	
Q12	When is measurement data recorded in the process? (at start, middle, end, later, etc.)	At start		
Q13	Is all measurement data recorded at the same place in the process? (at start, middle, end, later, etc.)	Yes	√	Yes
Q14	Who is responsible for recording measurement data?	Test Specialist		
Q15	Is all measurement data recorded by the responsible body?	Yes	√	Yes
Q16	How is measurement data recorded? (on a form, report, tool, etc.)	Tool		
Q17	Is all measurement data recorded the same way? (on a form, report, tool, etc.)	Yes	√	Yes
Q18	Where is measurement data stored? (in a file, database, etc.)	The tool's database		
Q19	Is all measurement data stored in the same place? (in a file, database, etc.)	Yes	√	Yes
Data Dependability			P	
Q20	What is the frequency of generating measurement data? (asynchronously, daily, weekly, monthly,	Asynchronously		
Q21	What is the frequency of recording measurement data? (asynchronously, daily, weekly, monthly,	Synchronously		
Q22	What is the frequency of storing measurement data? (asynchronously, daily, weekly, monthly, etc.)	Synchronously		
Q23	Are the frequencies for data generation, recording, and storing different?	No	√	No
Q24	Is measurement data recorded precisely?	Yes	√	Yes
Q25	Is measurement data collected for a specific purpose?	No	√	Yes
Q26	Is the purpose of measurement data collection known by process performers?	No	√	Yes
Q27	Is measurement data analyzed and reported?	No		Yes
Q28	Is measurement data analysis results communicated to process performers?	No		Yes
Q29	Is measurement data analysis results communicated to management?	No		Yes
Q30	Is measurement data analysis results used as a basis for decision making?	No		Yes
Data Normalizability				
Q31	Can measurement data be normalized by parameters or measures? (If yes, please specify them)	No		
Data Integrability				
Q32	Is measurement data integrable at project level?	Yes		
Q33	Is measurement data integrable at organization level?	Yes		

Figure D.5 MUQ for “Test Type” Basic Metric of Project-2

Metric Name: Product version				
Attributes		**Answers**	**Rating**	**Expected Answers**
	Indicators			
Measure Identity			N	
Q1	What is the measure formula? (please refer to related basic metrics)	Process		
Q2	Which attribute of the entity does the measure measure?	Frequency of product development updates		
Q3	What is the scale of the measurement data? (nominal, ordinal, interval, ratio, absolute)	Nominal		Ratio, Absolute
Q4	What is the unit of the measurement data?	Not applicable		
Q5	What is the type of the measurement data? (integer, real, etc.)	Text (x.y.z)		
Q6	What is the range of the measurement data?	0.0.1-2.0.8		
Data Existence			F	
Q7	Is measurement data existent?	Yes		
Q8	What is the amount of overall observations?	296	√	Available > 20
Q9	What is the amount of missing data points?	0		
Q10	Are data points missing in periods? (If yes, please state observation numbers for missing periods)	0		
Q11	Is measurement data time sequenced? (If no, please state how measurement data is sequenced)	Yes		
Data Verifiability			F	
Q12	When is measurement data recorded in the process? (at start, middle, end, later, etc.)	At start		
Q13	Is all measurement data recorded at the same place in the process? (at start, middle, end, later, etc.)	Yes	√	Yes
Q14	Who is responsible for recording measurement data?	Test Specialist		
Q15	Is all measurement data recorded by the responsible body?	Yes	√	Yes
Q16	How is measurement data recorded? (on a form, report, tool, etc.)	Tool		
Q17	Is all measurement data recorded the same way? (on a form, report, tool, etc.)	Yes	√	Yes
Q18	Where is measurement data stored? (in a file, database, etc.)	The tool's database		
Q19	Is all measurement data stored in the same place? (in a file, database, etc.)	Yes	√	Yes
Data Dependability			P	
Q20	What is the frequency of generating measurement data? (asynchronously, daily, weekly, monthly, etc.)	Asynchronously		
Q21	What is the frequency of recording measurement data? (asynchronously, daily, weekly, monthly, etc.)	Synchronously		
Q22	What is the frequency of storing measurement data? (asynchronously, daily, weekly, monthly, etc.)	Synchronously		
Q23	Are the frequencies for data generation, recording, and storing different?	No	√	No
Q24	Is measurement data recorded precisely?	Yes	√	Yes
Q25	Is measurement data collected for a specific purpose?	Yes	√	Yes
Q26	Is the purpose of measurement data collection known by process performers?	Yes	√	Yes
Q27	Is measurement data analyzed and reported?	No		Yes
Q28	Is measurement data analysis results communicated to process performers?	No		Yes
Q29	Is measurement data analysis results communicated to management?	No		Yes
Q30	Is measurement data analysis results used as a basis for decision making?	No		Yes
Data Normalizability				
Q31	Can measurement data be normalized by parameters or measures? (If yes, please specify them)	No		
Data Integrability				
Q32	Is measurement data integrable at project level?	No		
Q33	Is measurement data integrable at organization level?	No		

Figure D.6 MUQ for "Product Version" Basic Metric of Project-2

Metric Name: Product SLOC				
Attributes		**Answers**	**Rating**	**Expected Answers**
Indicators				
Measure Identity			N	
Q1	Which entity does the measure measure?	Product		
Q2	Which attribute of the entity does the measure measure?	Size of the product version		
Q3	What is the scale of the measurement data? (nominal, ordinal, interval, ratio, absolute)	Absolute		Ratio, Absolute
Q4	What is the unit of the measurement data?	LOC, KLOC		
Q5	What is the type of the measurement data? (integer, real, etc.)	Integer		
Q6	What is the range of the measurement data?	[283,51533]		
Data Existence			F	
Q7	Is measurement data existent?	No (collected manually)		
Q8	What is the amount of overall observations?	18	√	Available > 20
Q9	What is the amount of missing data points?	0		
Q10	Are data points missing in periods? (If yes, please state observation numbers for missing periods)	0		
Q11	Is measurement data time sequenced? (If no, please state how measurement data is sequenced)	Yes		
Data Verifiability			F	
Q12	When is measurement data recorded in the process? (at start, middle, end, later, etc.)	Monthly		
Q13	Is all measurement data recorded at the same place in the process? (at start, middle, end, later, etc.)	Yes	√	Yes
Q14	Who is responsible for recording measurement data?	Project Manager		
Q15	Is all measurement data recorded by the responsible body?	Yes	√	Yes
Q16	How is measurement data recorded? (on a form, report, tool, etc.)	Report		
Q17	Is all measurement data recorded the same way? (on a form, report, tool, etc.)	Yes	√	Yes
Q18	Where is measurement data stored? (in a file, database, etc.)	tool		
Q19	Is all measurement data stored in the same place? (in a file, database, etc.)	Yes	√	Yes
Data Dependability			P	
Q20	What is the frequency of generating measurement data? (asynchronously, daily, weekly, monthly, etc.)	Asynchronously		
Q21	What is the frequency of recording measurement data? (asynchronously, daily, weekly, monthly, etc.)	Monthly		
Q22	What is the frequency of storing measurement data? (asynchronously, daily, weekly, monthly, etc.)	Synchronously		
Q23	Are the frequencies for data generation, recording, and storing different?	No	√	No
Q24	Is measurement data recorded precisely?	Yes	√	Yes
Q25	Is measurement data collected for a specific purpose?	Yes	√	Yes
Q26	Is the purpose of measurement data collection known by process performers?	Yes	√	Yes
Q27	Is measurement data analyzed and reported?	Yes		Yes
Q28	Is measurement data analysis results communicated to process performers?	Yes		Yes
Q29	Is measurement data analysis results communicated to management?	Yes		Yes
Q30	Is measurement data analysis results used as a basis for decision making?	Yes		Yes
Data Normalizability				
Q31	Can measurement data be normalized by parameters or measures? (If yes, please specify them)	Yes (KLOC)		
Data Integrability				
Q32	Is measurement data integrable at project level?	Yes		
Q33	Is measurement data integrable at organization level?	Yes		

Figure D.7 MUQ for “Product SLOC” Basic Metric of Project-2

Metric Name: Product complexity				
Attributes		Answers	Rating	Expected Answers
	Indicators			
Measure Identity			N	
Q1	Which entity does the measure measure?	Product		
Q2	Which attribute of the entity does the measure measure?	Complexity of the product version		
Q3	What is the scale of the measurement data? (nominal, ordinal, interval, ratio, absolute)	Absolute		Ratio, Absolute
Q4	What is the unit of the measurement data?	Number of decision nodes in software		
Q5	What is the type of the measurement data? (integer, real, etc.)	Integer		
Q6	What is the range of the measurement data?	[33,9180]		
Data Existence			F	
Q7	Is measurement data existent?	No (collected manually)		
Q8	What is the amount of overall observations?	18	√	Available > 20
Q9	What is the amount of missing data points?	0		
Q10	Are data points missing in periods? (If yes, please state observation numbers for missing periods)	0		
Q11	Is measurement data time sequenced? (If no, please state how measurement data is sequenced)	Yes		
Data Verifiability			F	
Q12	When is measurement data recorded in the process? (at start, middle, end, later, etc.)	Never		
Q13	Is all measurement data recorded at the same place in the process? (at start, middle, end, later, etc.)	No	√	Yes
Q14	Who is responsible for recording measurement data?	No one		
Q15	Is all measurement data recorded by the responsible body?	No	√	Yes
Q16	How is measurement data recorded? (on a form, report, tool, etc.)	No		
Q17	Is all measurement data recorded the same way? (on a form, report, tool, etc.)	No	√	Yes
Q18	Where is measurement data stored? (in a file, database, etc.)	No		
Q19	Is all measurement data stored in the same place? (in a file, database, etc.)	No	√	Yes
Data Dependability			P	
Q20	What is the frequency of generating measurement data? (asynchronously, daily, weekly, monthly, etc.)	Asynchronously		
Q21	What is the frequency of recording measurement data? (asynchronously, daily, weekly, monthly, etc.)	Synchronously		
Q22	What is the frequency of storing measurement data? (asynchronously, daily, weekly, monthly, etc.)	Synchronously		
Q23	Are the frequencies for data generation, recording, and storing different?	Yes	√	No
Q24	Is measurement data recorded precisely?	No	√	Yes
Q25	Is measurement data collected for a specific purpose?	No	√	Yes
Q26	Is the purpose of measurement data collection known by process performers?	No	√	Yes
Q27	Is measurement data analyzed and reported?	No		Yes
Q28	Is measurement data analysis results communicated to process performers?	No		Yes
Q29	Is measurement data analysis results communicated to management?	No		Yes
Q30	Is measurement data analysis results used as a basis for decision making?	No		Yes
Data Normalizability				
Q31	Can measurement data be normalized by parameters or measures? (If yes, please specify them)	No		
Data Integrability				
Q32	Is measurement data integrable at project level?	Yes		
Q33	Is measurement data integrable at organization level?	Yes		

Figure D.8 MUQ for “Product Complexity” Basic Metric of Project-2

Metric Name: Reproducibility				
Attributes		**Answers**	**Rating**	**Expected Answers**
	Indicators			
Measure Identity			N	
Q1	Which entity does the measure measure?	Process		
Q2	Which attribute of the entity does the measure measure?	Repeatability status of detected defects		
Q3	What is the scale of the measurement data? (nominal, ordinal, interval, ratio, absolute)	Nominal		Ratio, Absolute
Q4	What is the unit of the measurement data?	Not applicable		
Q5	What is the type of the measurement data? (integer, real, etc.)	Text		
Q6	What is the range of the measurement data?	Four status types		
Data Existence			F	
Q7	Is measurement data existent?	Yes		
Q8	What is the amount of overall observations?	425	√	Available > 20
Q9	What is the amount of missing data points?	0		
Q10	Are data points missing in periods? (If yes, please state observation numbers for missing periods)	0		
Q11	Is measurement data time sequenced? (If no, please state how measurement data is sequenced)	Yes		
Data Verifiability			F	
Q12	When is measurement data recorded in the process? (at start, middle, end, later, etc.)	At start		
Q13	Is all measurement data recorded at the same place in the process? (at start, middle, end, later, etc.)	Yes	√	Yes
Q14	Who is responsible for recording measurement data?	Test Specialist		
Q15	Is all measurement data recorded by the responsible body?	Yes	√	Yes
Q16	How is measurement data recorded? (on a form, report, tool, etc.)	Tool		
Q17	Is all measurement data recorded the same way? (on a form, report, tool, etc.)	Yes	√	Yes
Q18	Where is measurement data stored? (in a file, database, etc.)	The tool's database		
Q19	Is all measurement data stored in the same place? (in a file, database, etc.)	Yes	√	Yes
Data Dependability			P	
Q20	What is the frequency of generating measurement data? (asynchronously, daily, weekly, monthly,	Asynchronously		
Q21	What is the frequency of recording measurement data? (asynchronously, daily, weekly, monthly,	Synchronously		
Q22	What is the frequency of storing measurement data? (asynchronously, daily, weekly, monthly, etc.)	Synchronously		
Q23	Are the frequencies for data generation, recording, and storing different?	No	√	No
Q24	Is measurement data recorded precisely?	Yes	√	Yes
Q25	Is measurement data collected for a specific purpose?	No	√	Yes
Q26	Is the purpose of measurement data collection known by process performers?	No	√	Yes
Q27	Is measurement data analyzed and reported?	No		Yes
Q28	Is measurement data analysis results communicated to process performers?	No		Yes
Q29	Is measurement data analysis results communicated to management?	No		Yes
Q30	Is measurement data analysis results used as a basis for decision making?	No		Yes
Data Normalizability				
Q31	Can measurement data be normalized by parameters or measures? (If yes, please specify them)	No		
Data Integrability				
Q32	Is measurement data integrable at project level?	Yes		
Q33	Is measurement data integrable at organization level?	Yes		

Figure D.9 MUQ for "Reproducibility" Basic Metric of Project-2

Metric Name: Project phase				
Attributes		Answers	Rating	Expected Answers
	Indicators			
Measure Identity			N	
Q1	Which entity does the measure measure?	Process		
Q2	Which attribute of the entity does the measure measure?	Project phase of detected defects		
Q3	What is the scale of the measurement data? (nominal, ordinal, interval, ratio, absolute)	Nominal		Ratio, Absolute
Q4	What is the unit of the measurement data?	Not applicable		
Q5	What is the type of the measurement data? (integer, real, etc.)	Text		
Q6	What is the range of the measurement data?	Four phase types		
Data Existence			F	
Q7	Is measurement data existent?	No		
Q8	What is the amount of overall observations?	Not applicable	√	Available > 20
Q9	What is the amount of missing data points?	425		
Q10	Are data points missing in periods? (If yes, please state observation numbers for missing periods)	425		
Q11	Is measurement data time sequenced? (If no, please state how measurement data is sequenced)	Not applicable		
Data Verifiability (After manual collection)			F	
Q12	When is measurement data recorded in the process? (at start, middle, end, later, etc.)	Later		
Q13	Is all measurement data recorded at the same place in the process? (at start, middle, end, later, etc.)	Later	√	Yes
Q14	Who is responsible for recording measurement data?	Project Manager		
Q15	Is all measurement data recorded by the responsible body?	Yes	√	Yes
Q16	How is measurement data recorded? (on a form, report, tool, etc.)	On a form		
Q17	Is all measurement data recorded the same way? (on a form, report, tool, etc.)	Yes	√	Yes
Q18	Where is measurement data stored? (in a file, database, etc.)	In a file		
Q19	Is all measurement data stored in the same place? (in a file, database, etc.)	Yes	√	Yes
Data Dependability			P	
Q20	What is the frequency of generating measurement data? (asynchronously, daily, weekly, monthly, etc.)	Asynchronously		
Q21	What is the frequency of recording measurement data? (asynchronously, daily, weekly, monthly, etc.)	Asynchronously		
Q22	What is the frequency of storing measurement data? (asynchronously, daily, weekly, monthly, etc.)	Synchronously		
Q23	Are the frequencies for data generation, recording, and storing different?	Yes	√	No
Q24	Is measurement data recorded precisely?	No	√	Yes
Q25	Is measurement data collected for a specific purpose?	No	√	Yes
Q26	Is the purpose of measurement data collection known by process performers?	No	√	Yes
Q27	Is measurement data analyzed and reported?	No		Yes
Q28	Is measurement data analysis results communicated to process performers?	No		Yes
Q29	Is measurement data analysis results communicated to management?	No		Yes
Q30	Is measurement data analysis results used as a basis for decision making?	No		Yes
Data Normalizability				
Q31	Can measurement data be normalized by parameters or measures? (If yes, please specify them)	No		
Data Integrability				
Q32	Is measurement data integrable at project level?	Yes		
Q33	Is measurement data integrable at organization level?	Yes		

Figure D.10 MUQ for “Project Phase” Basic Metric of Project-2

Metric Name: Remaining open duration					
Attributes			Answers	Rating	Expected Answers
	Indicators				
Measure Identity				N	
	Q1	What is the measure formula? (please refer to related basic metrics)	Created date, closed date		
	Q2	What is the scale of the measurement data? (nominal, ordinal, interval, ratio, absolute)	Absolute		Ratio, Absolute
	Q3	What is the unit of the measurement data?	day		
	Q4	What is the type of the measurement data? (integer, real, etc.)	Integer		
	Q5	What is the range of the measurement data?	[0-140]		
Data Existence				F	
	Q6	Is measurement data existent?	Yes		
	Q7	What is the amount of overall observations?	425	√	Available > 20
	Q8	What is the amount of missing data points?	0		
	Q9	Are data points missing in periods? (If yes, please state observation numbers for	0		
	Q10	Is measurement data time sequenced? (If no, please state how measurement data is	Yes		
Data Verifiability				F	
	Q11	How is the measure data calculated? (by a tool, manually, etc.)	By a tool		
	Q12	Is all measurement data calculated with the same way? (by a tool, manually, etc.)	Yes	√	Yes
	Q13	Is all measurement data calculated according to measure formula?	Yes		
	Q14	Where is measurement data stored? (in a file, database, etc.)	In a file	√	Yes
	Q15	Is all measurement data stored in the same place? (in a file, database, etc.)	Yes		
Data Dependability				P	
	Q16	Is measurement data stored precisely?	Yes	√	Yes
	Q17	Is measurement data stored for a specific purpose?	Yes	√	Yes
	Q18	Is the purpose of measurement data collection known by process performers?	Yes	√	Yes
	Q19	Is measurement data analyzed and reported?	No		Yes
	Q20	Is measurement data analysis results communicated to process performers?	No		Yes
	Q21	Is measurement data analysis results communicated to management?	No		Yes
	Q22	Is measurement data analysis results used as a basis for decision making?	No		Yes
Data Normalizability					
	Q23	Can measurement data be normalized by parameters or measures? (If yes, please speci	No		
Data Integrability					
	Q24	Is measurement data integrable at project level?	Yes		
	Q25	Is measurement data integrable at organization level?	Yes		

Figure D.11 MUQ for “Defect Open Duration” Derived Metric of Project-2

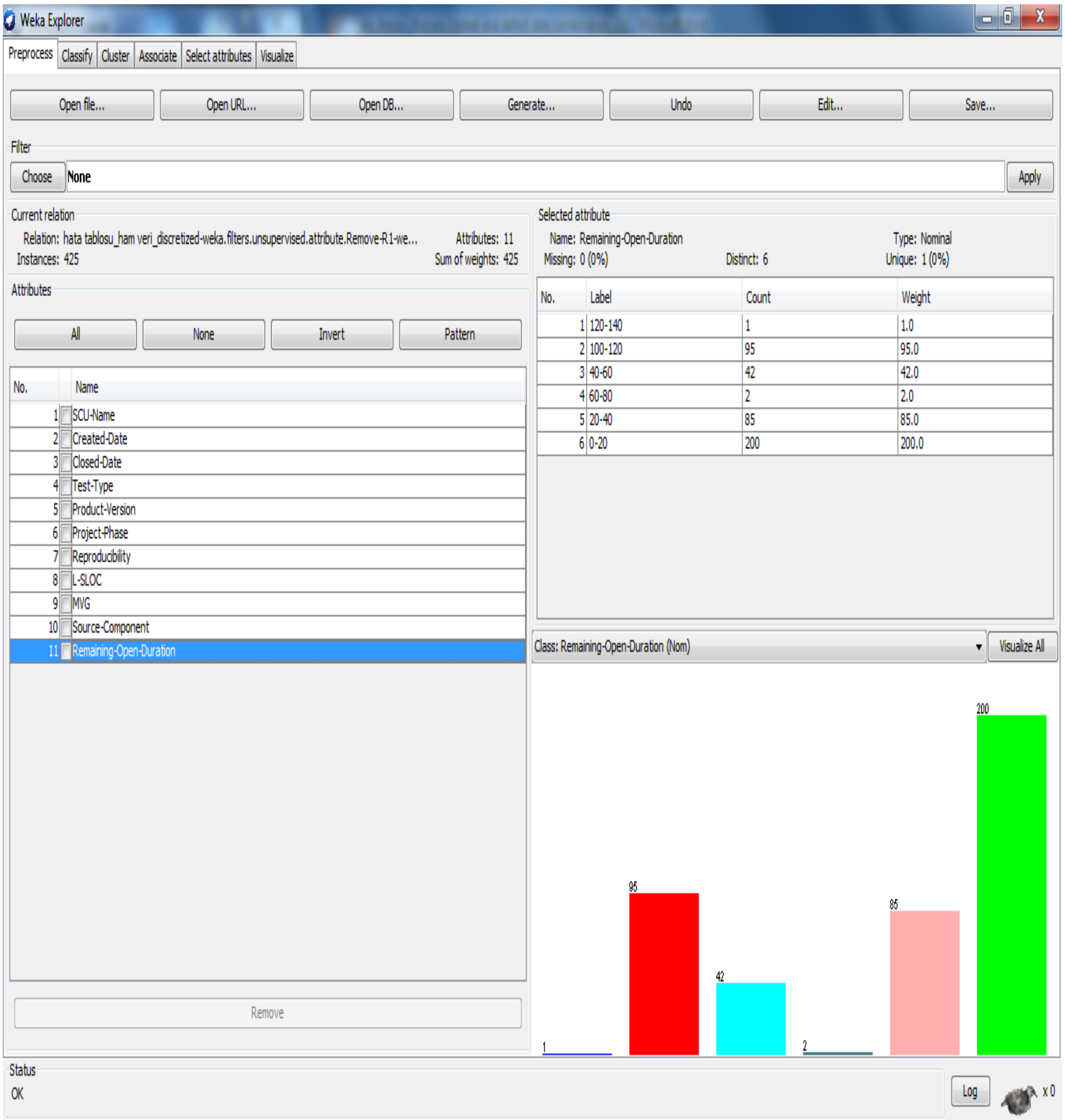

Figure D.12 Weka View of Case Study 2A

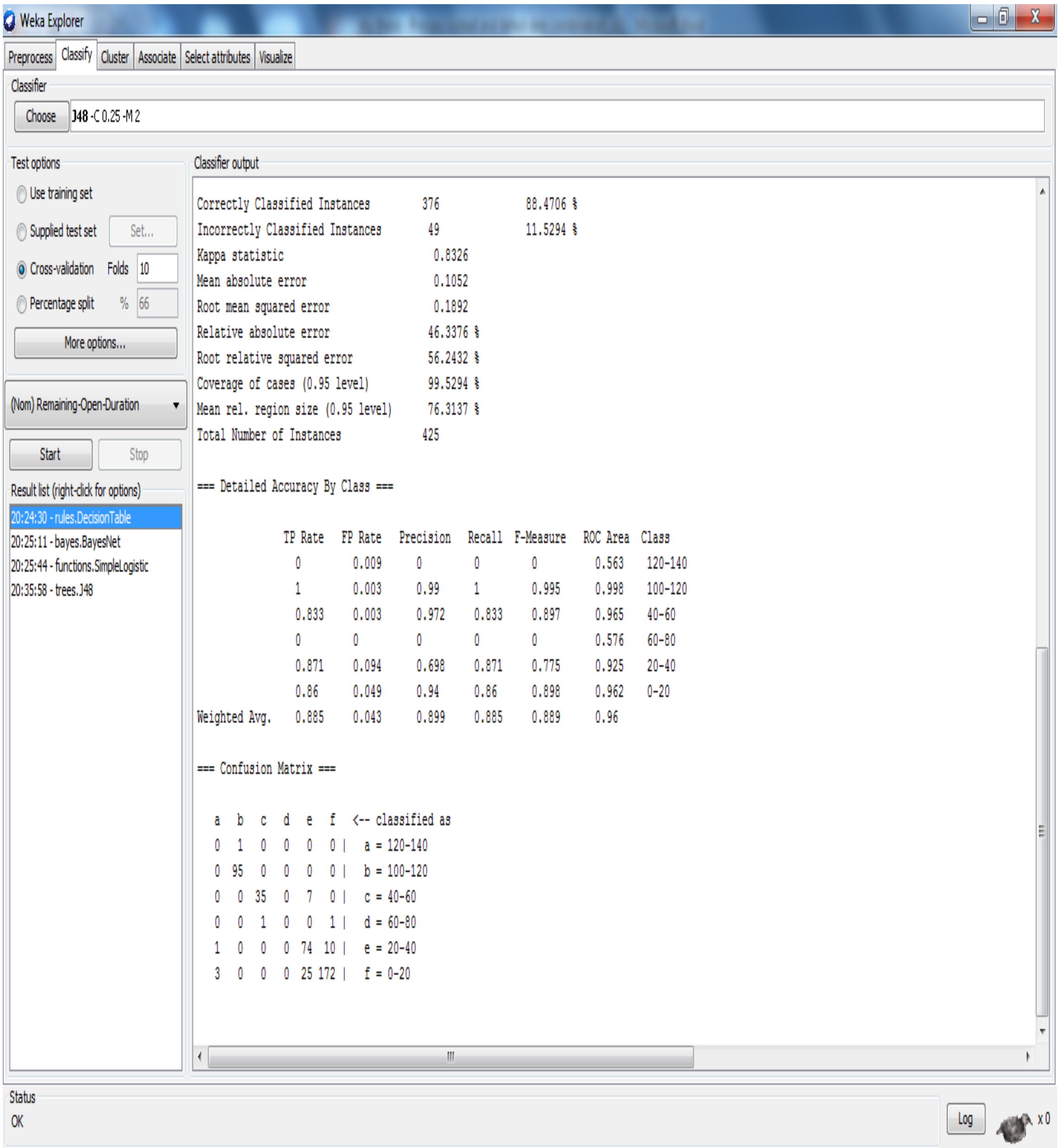

Figure D.13 Decision Table Results of Case Study 2A

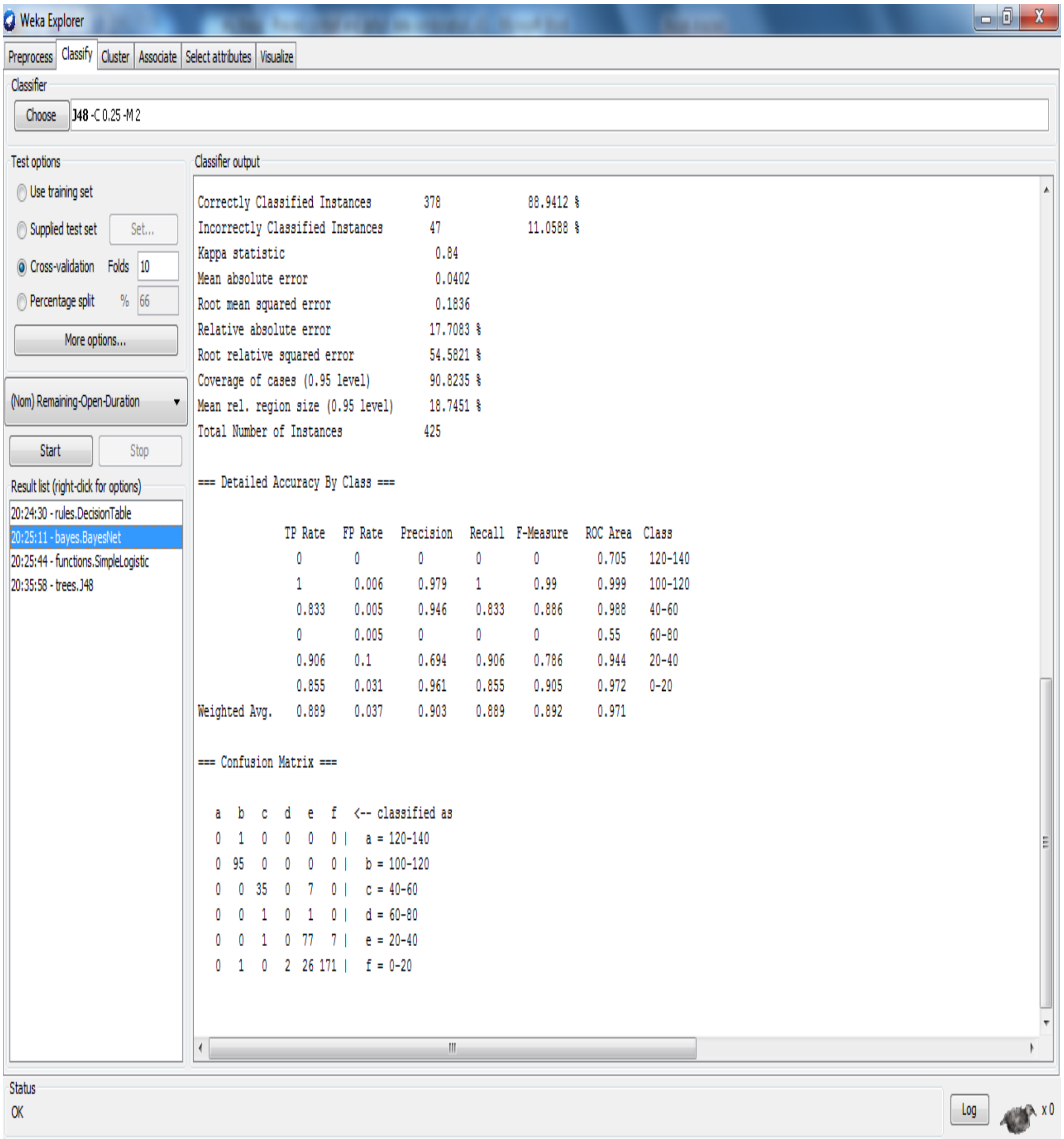

Figure D.14 BayesNet Table Results of Case Study 2A

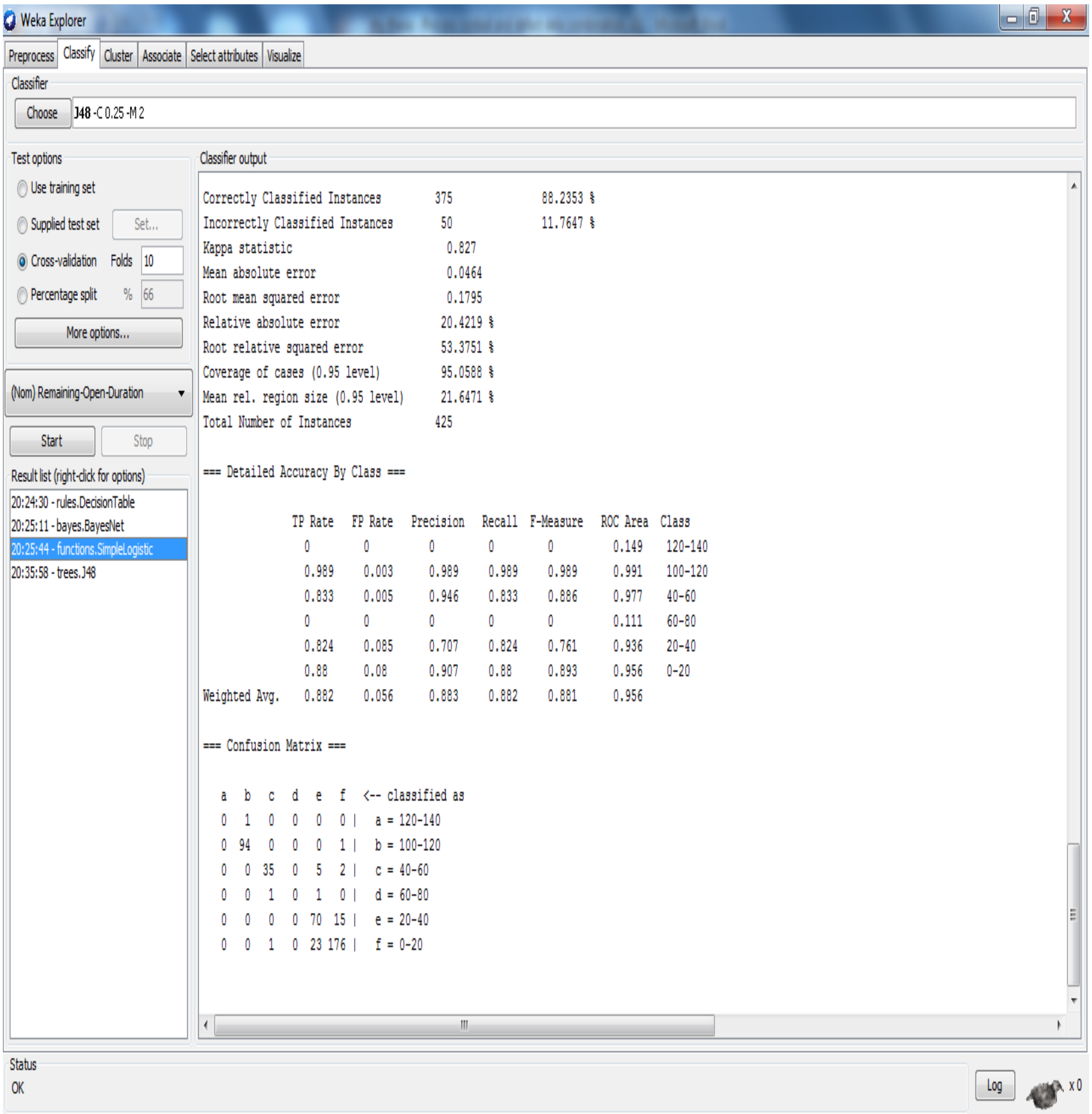

Figure D.15 SimpleLogistic Table Results of Case Study 2A

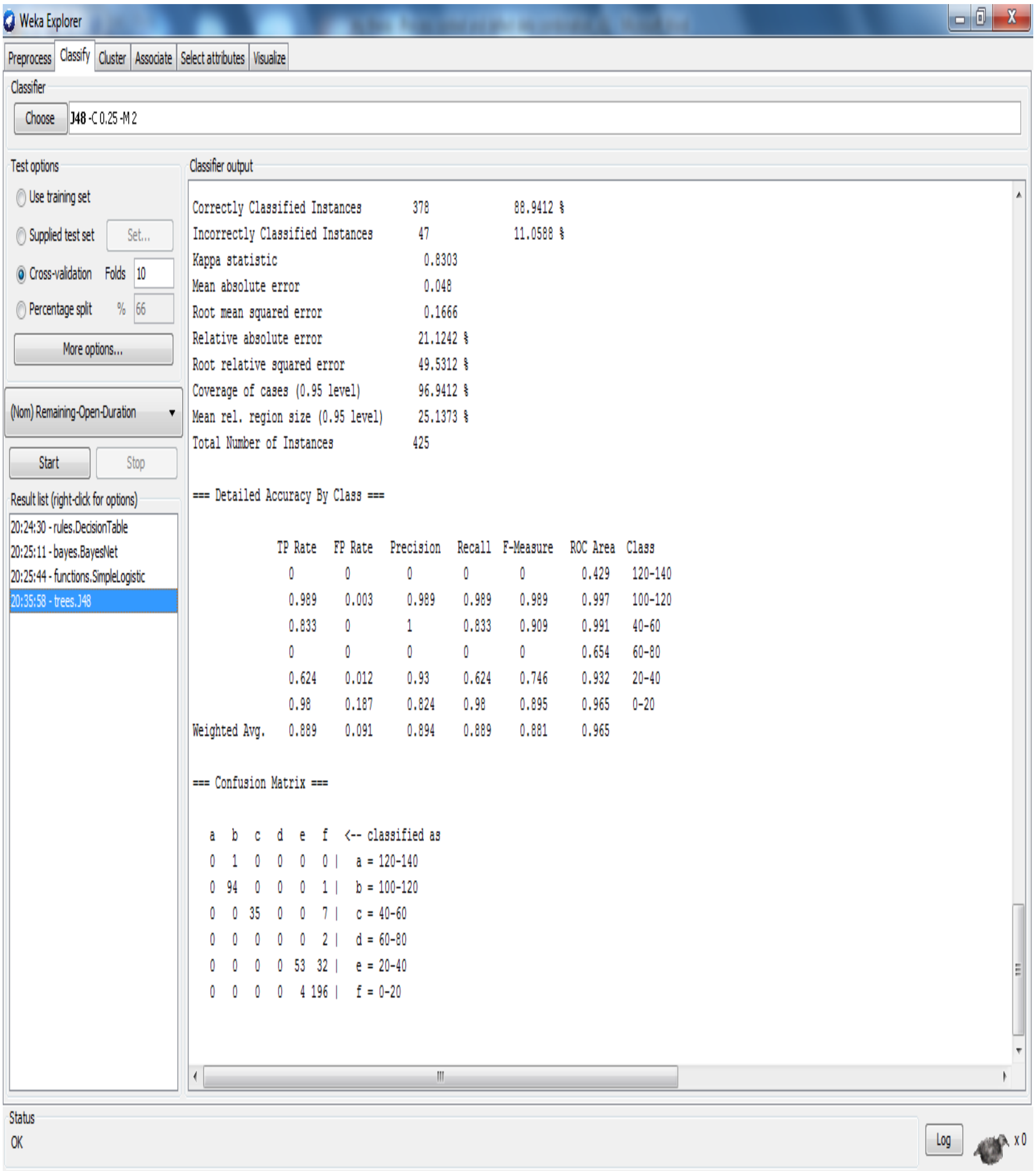

Figure D.16 J48Results of Case Study 2A

E. DETAILS OF CASE STUDY 2B

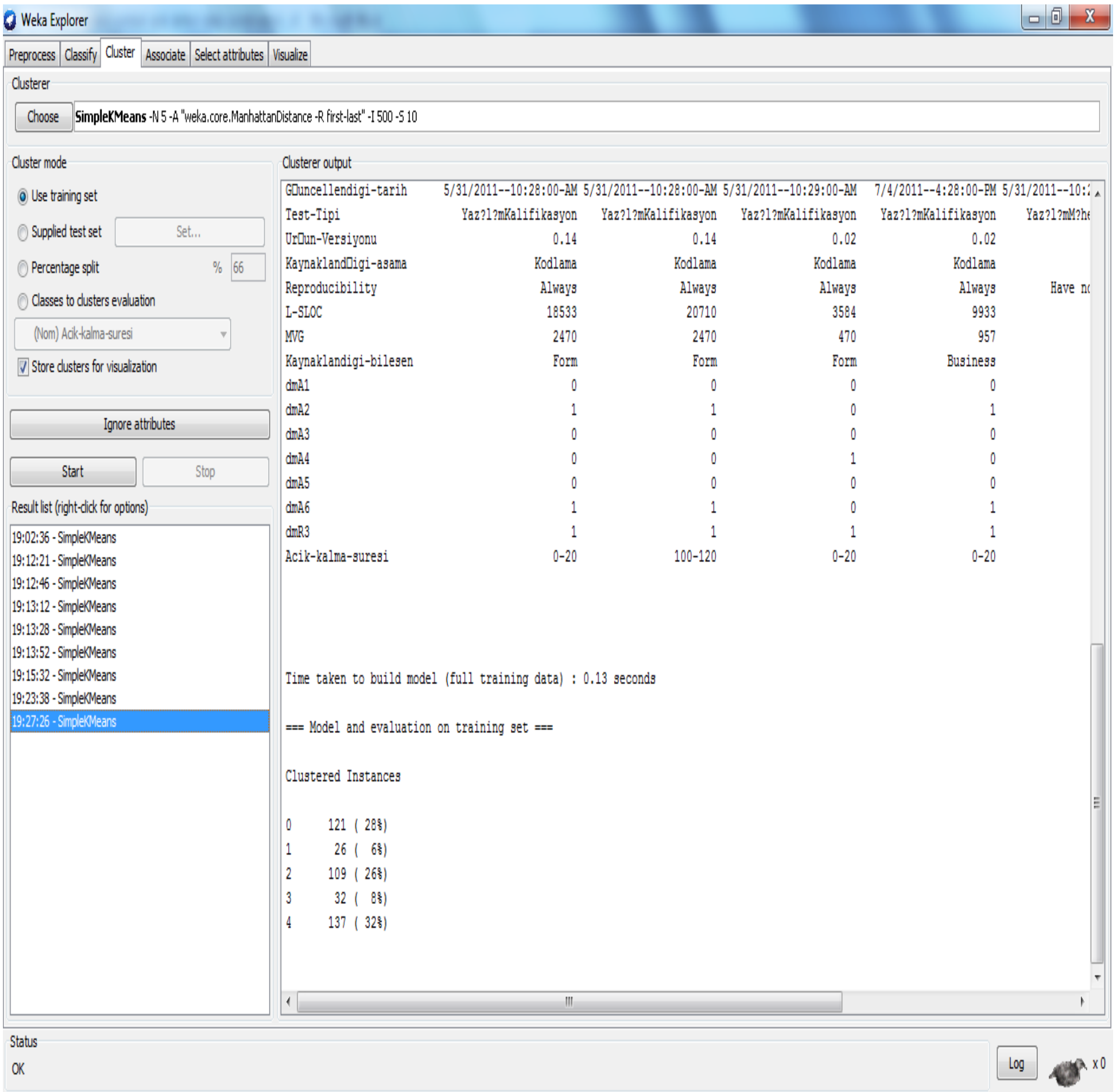

Figure E.1 SimpleKMeans Clustering of Case Study 2B

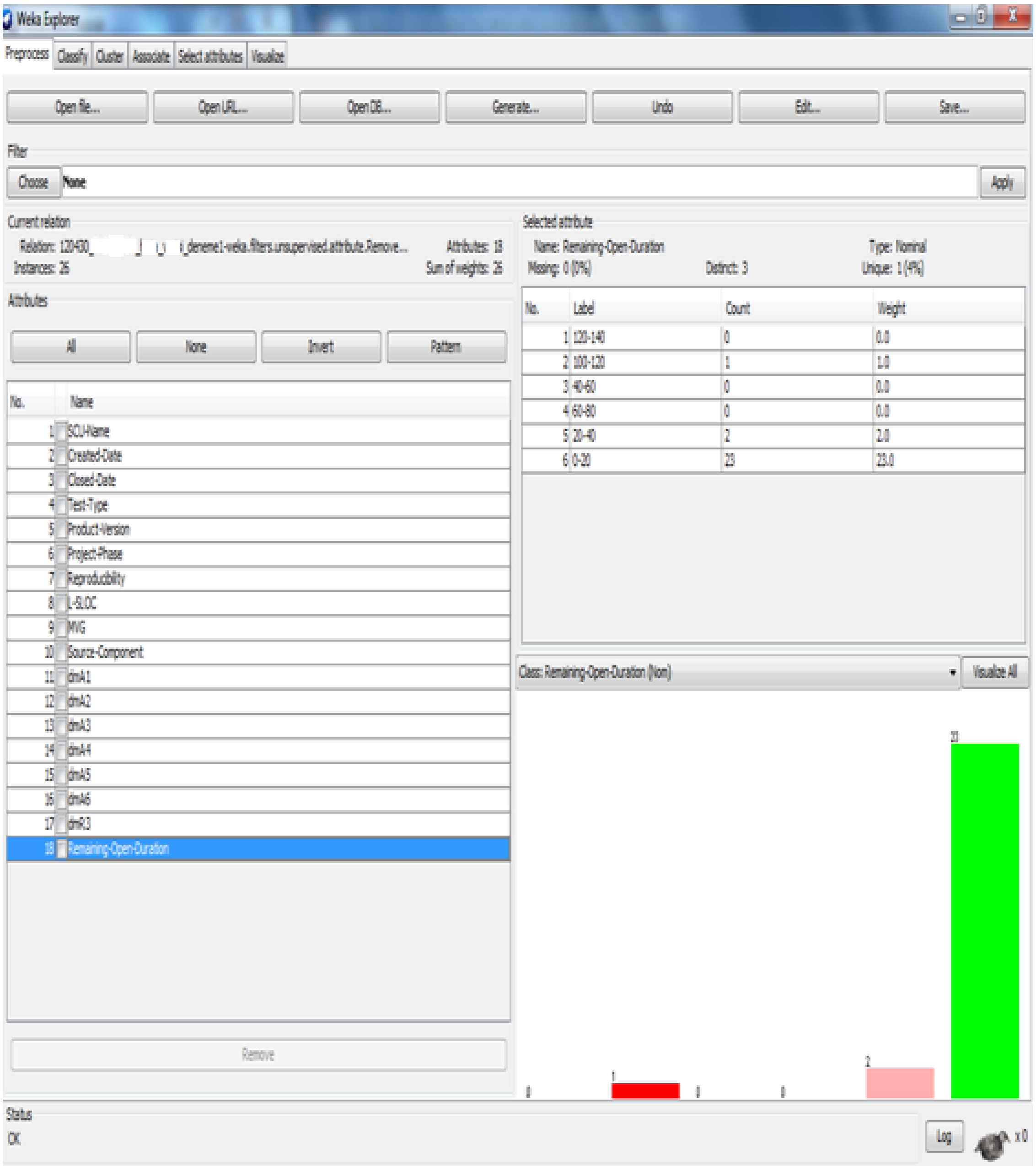

Figure E.2 Weka View of Case Study 2B Cluster 0

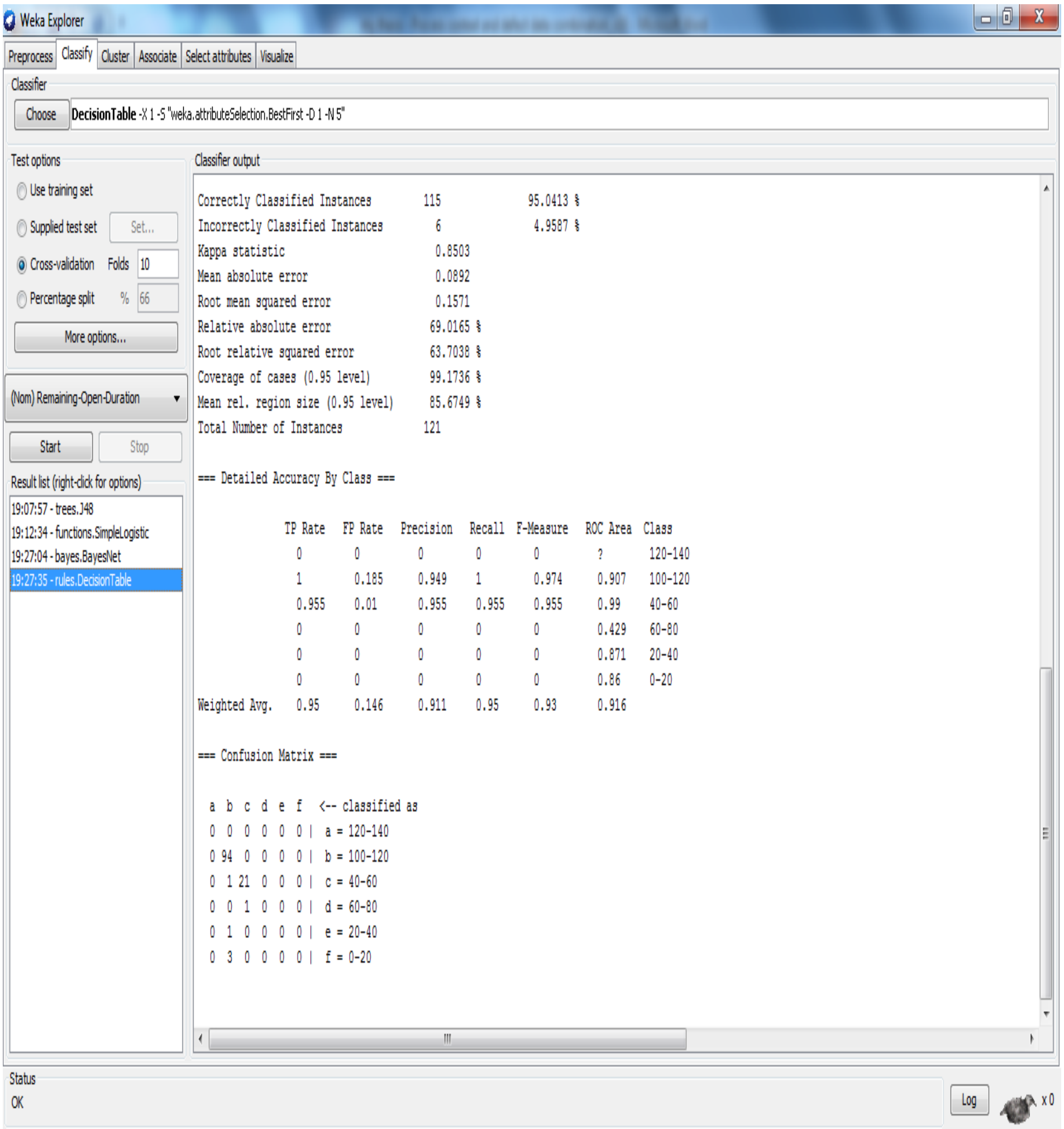

Figure E.3 DecisionTable Results of Case Study 2B Cluster 0

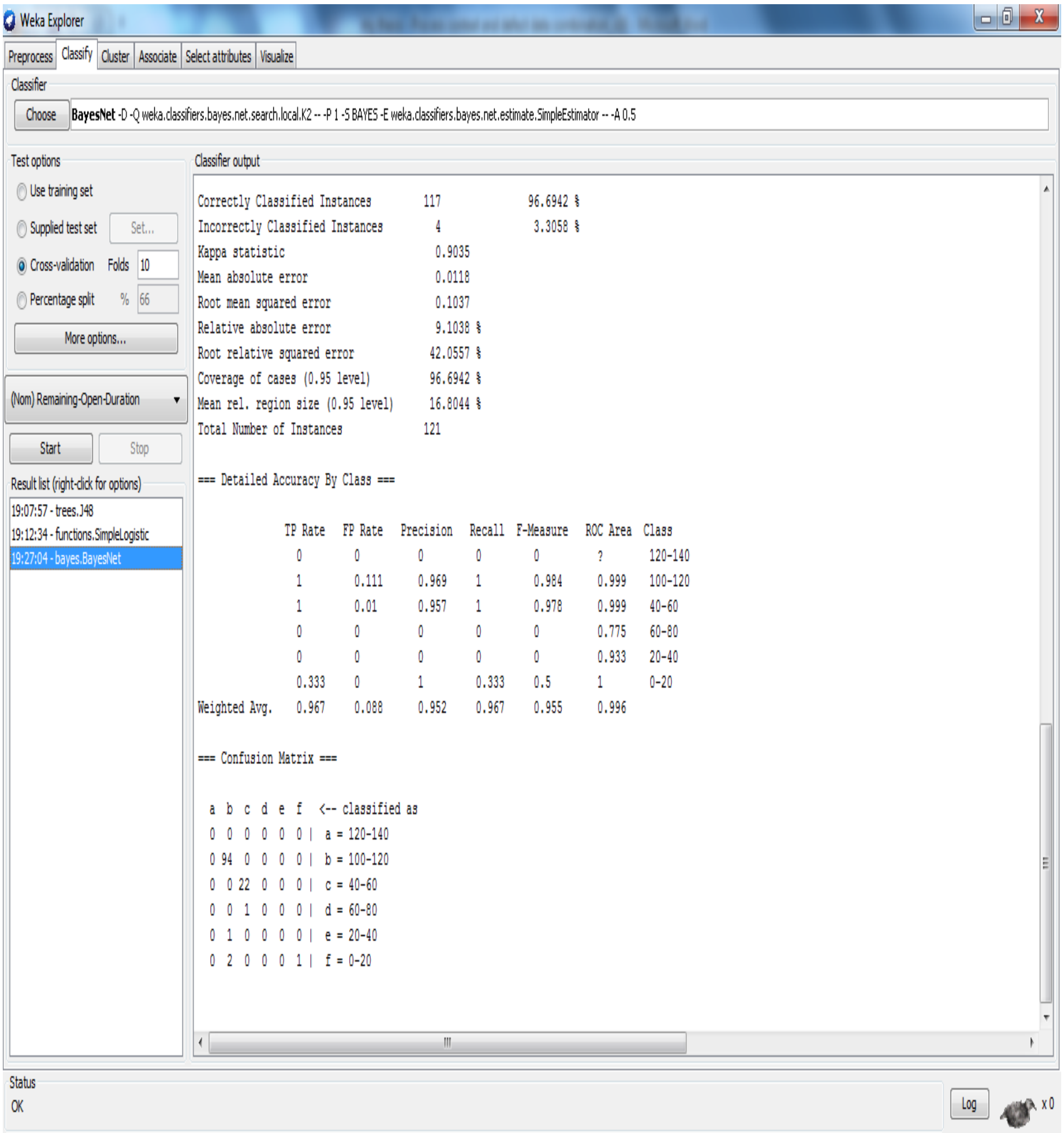

Figure E.4 BayesNet Results of Case Study 2B Cluster 0

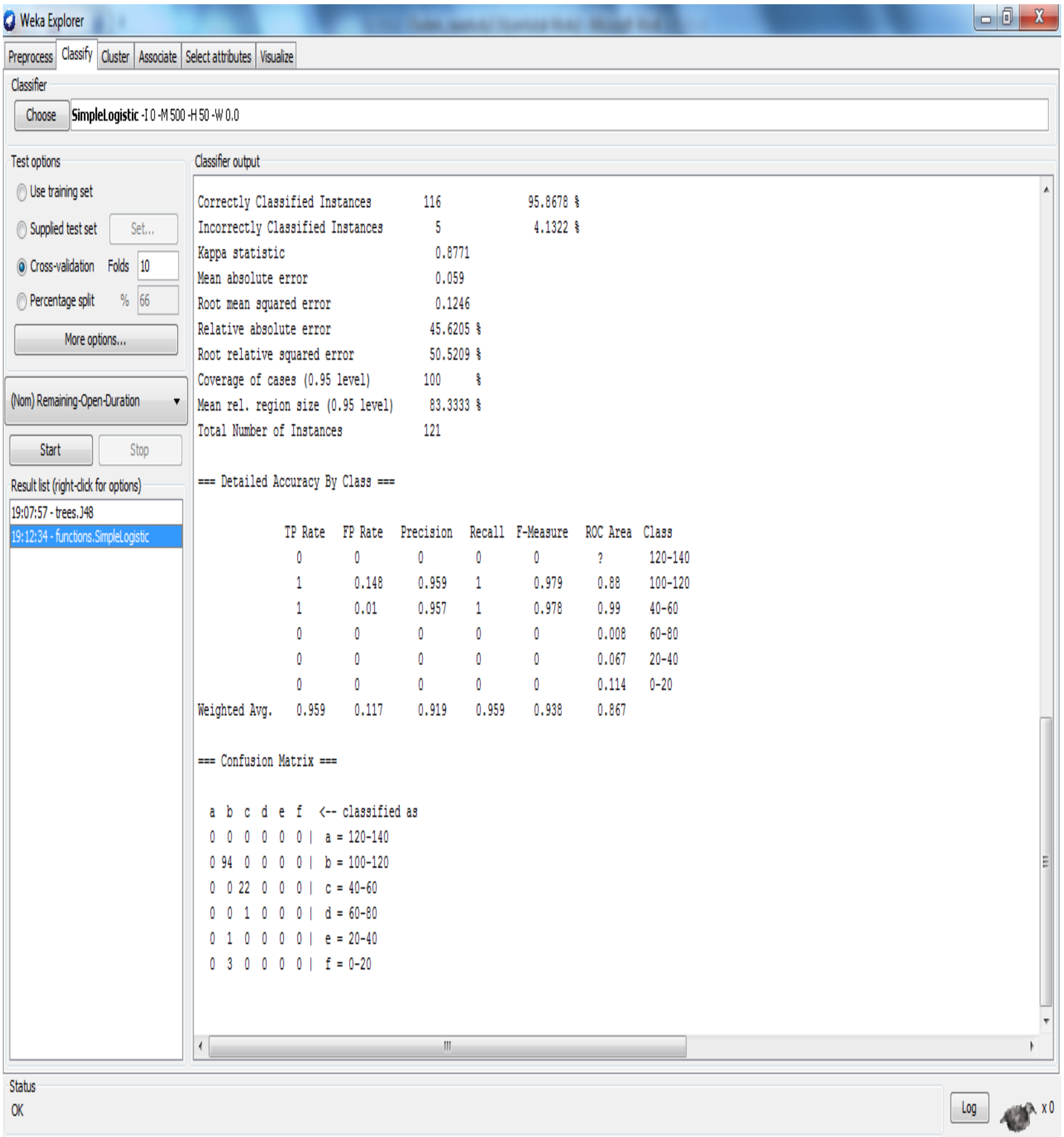

Figure E.5 SimpleLogistic Results of Case Study 2B Cluster 0

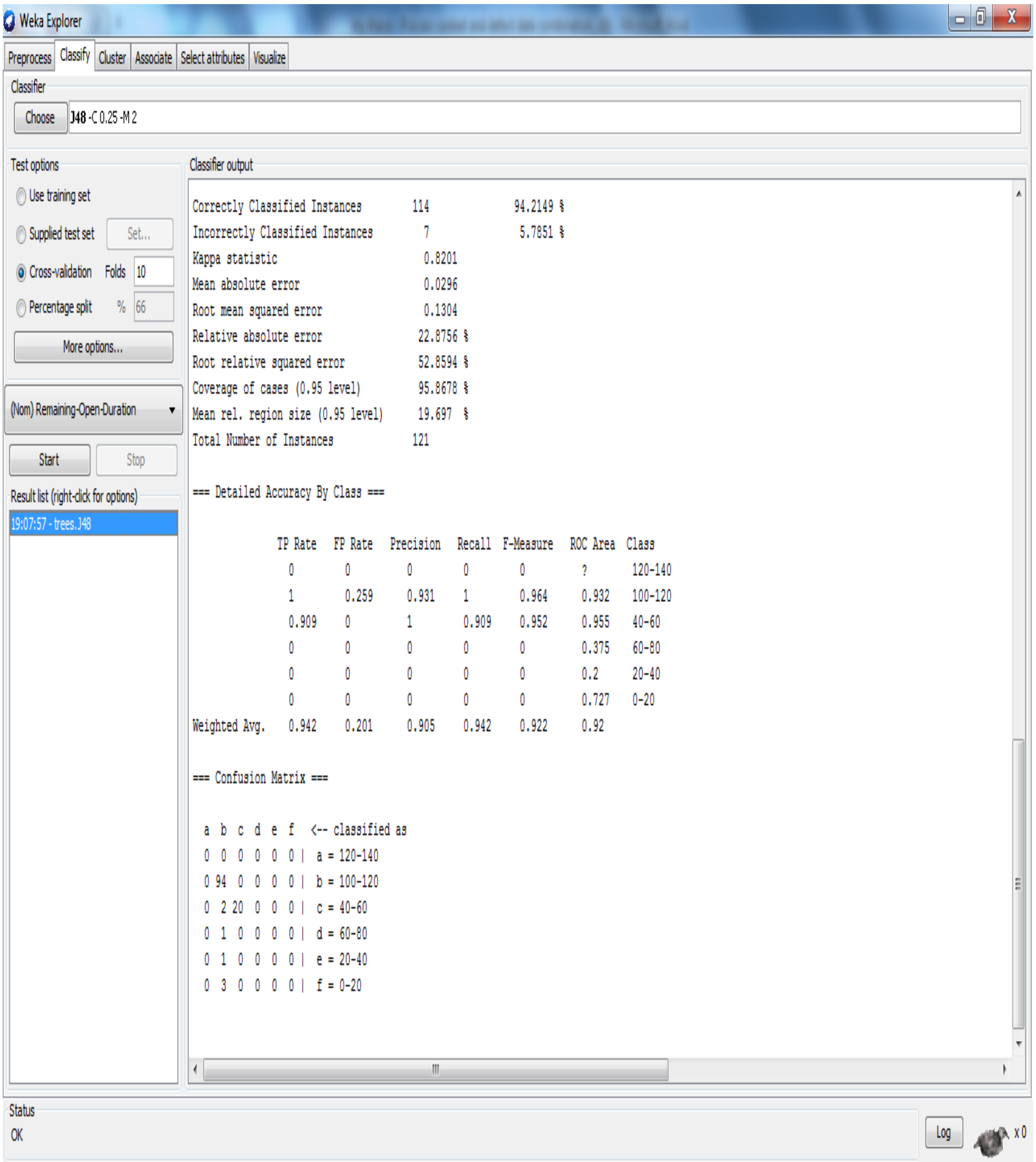

Figure E.6 J48 Results of Case Study 2B Cluster 0

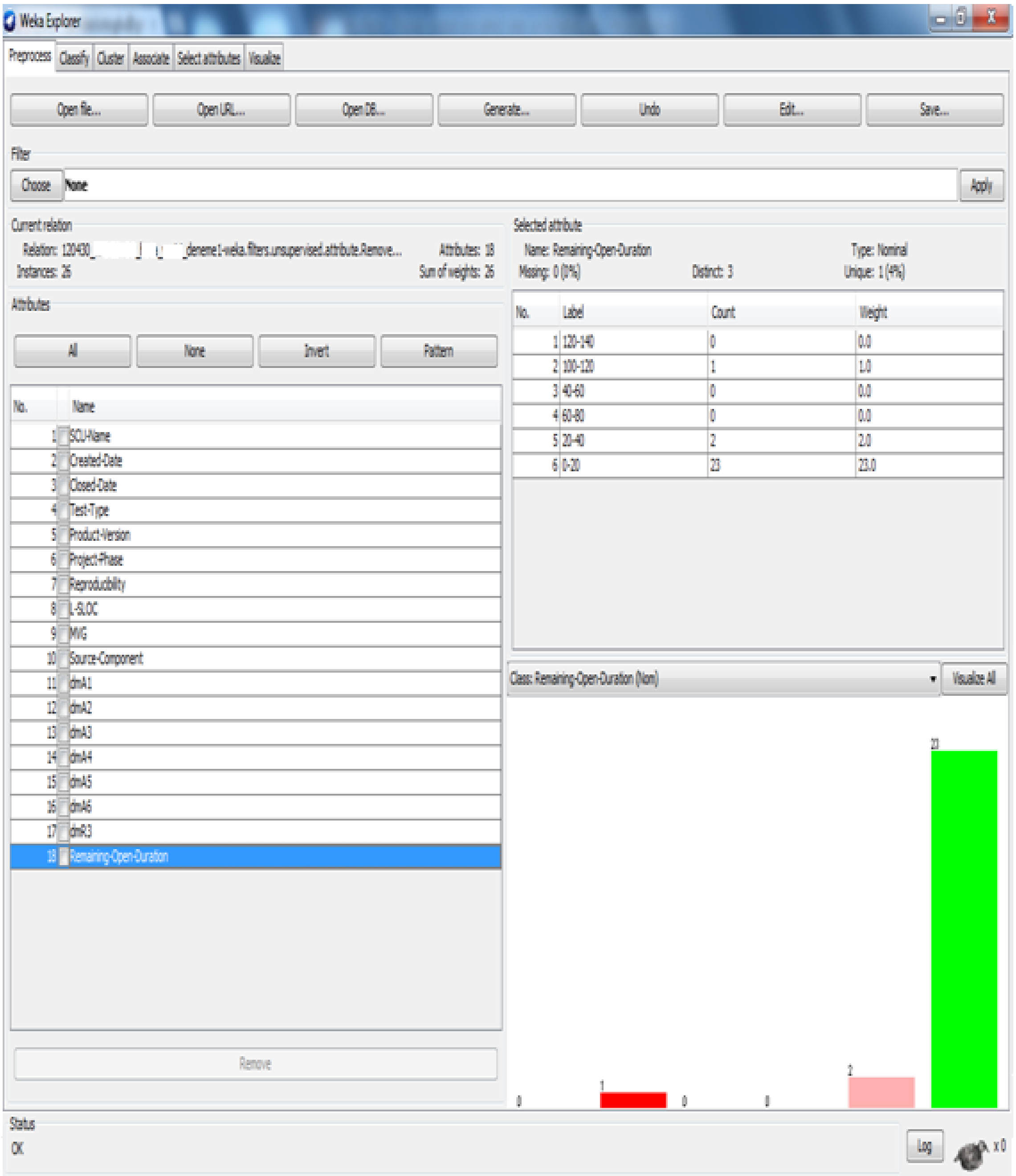

Figure E.7 Weka View of Case Study 2B Cluster 1

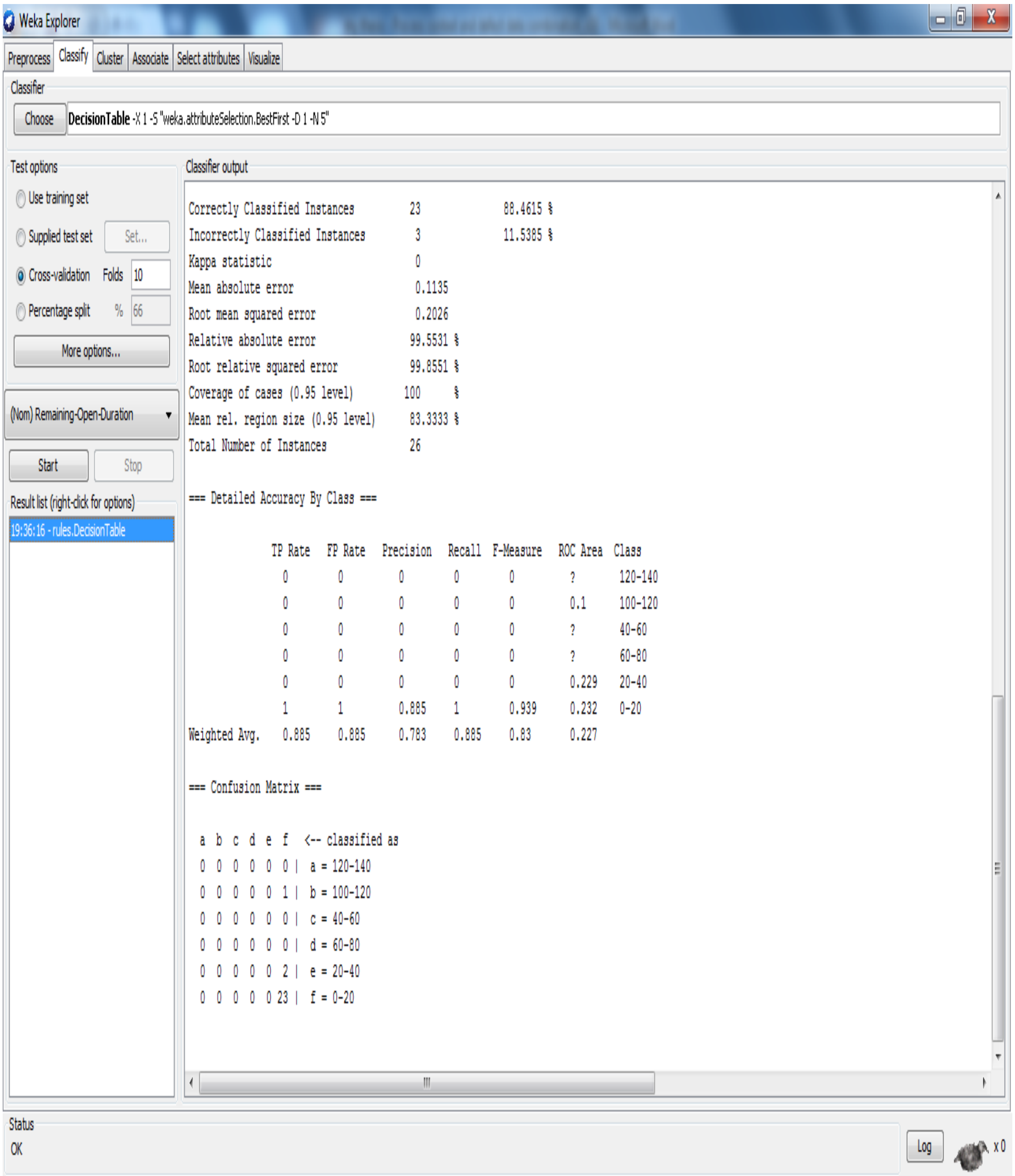

Figure E.8 DecisionTable Results of Case Study 2B Cluster 1

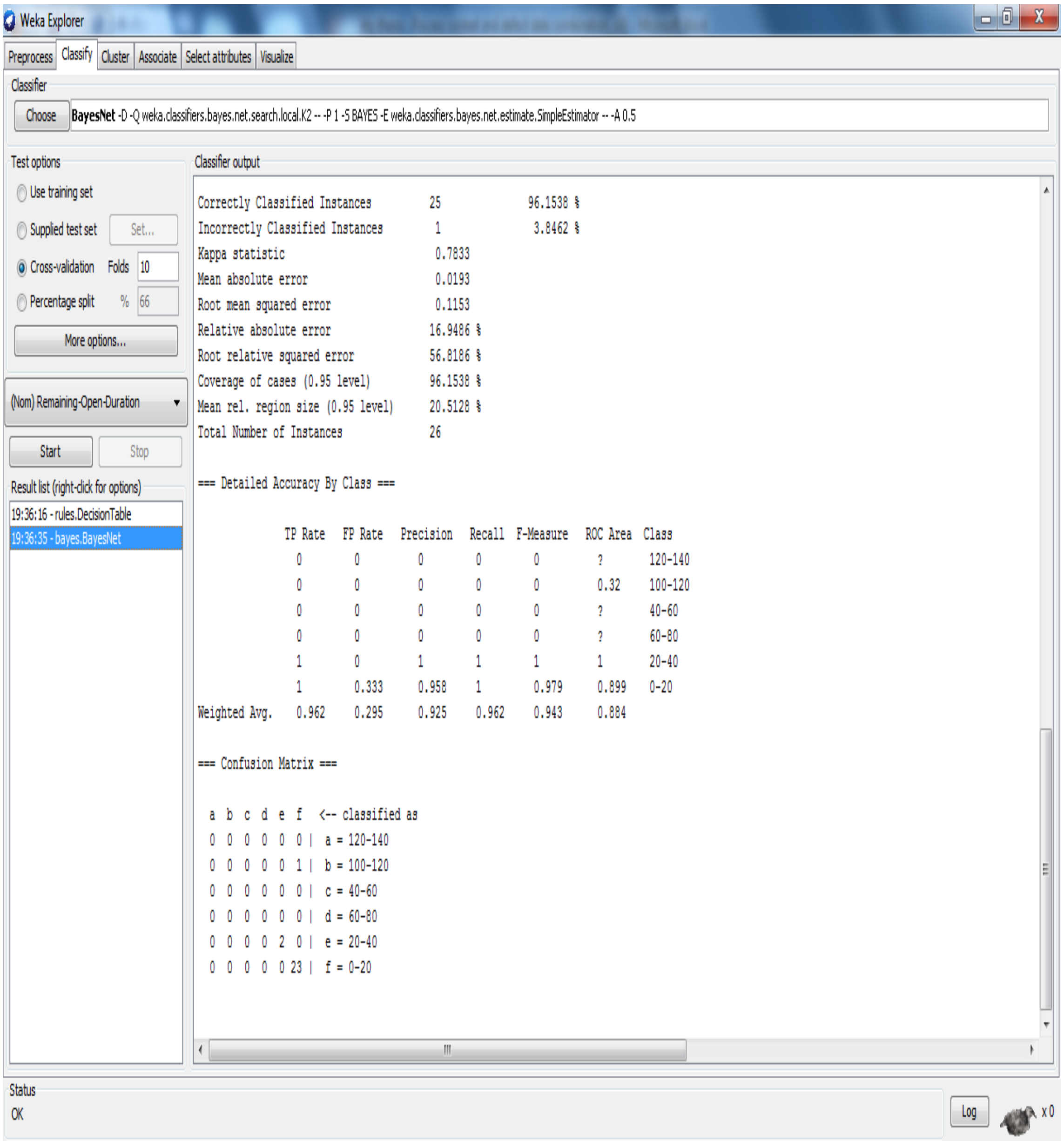

Figure E.9 BayesNet Results of Case Study 2B Cluster 1

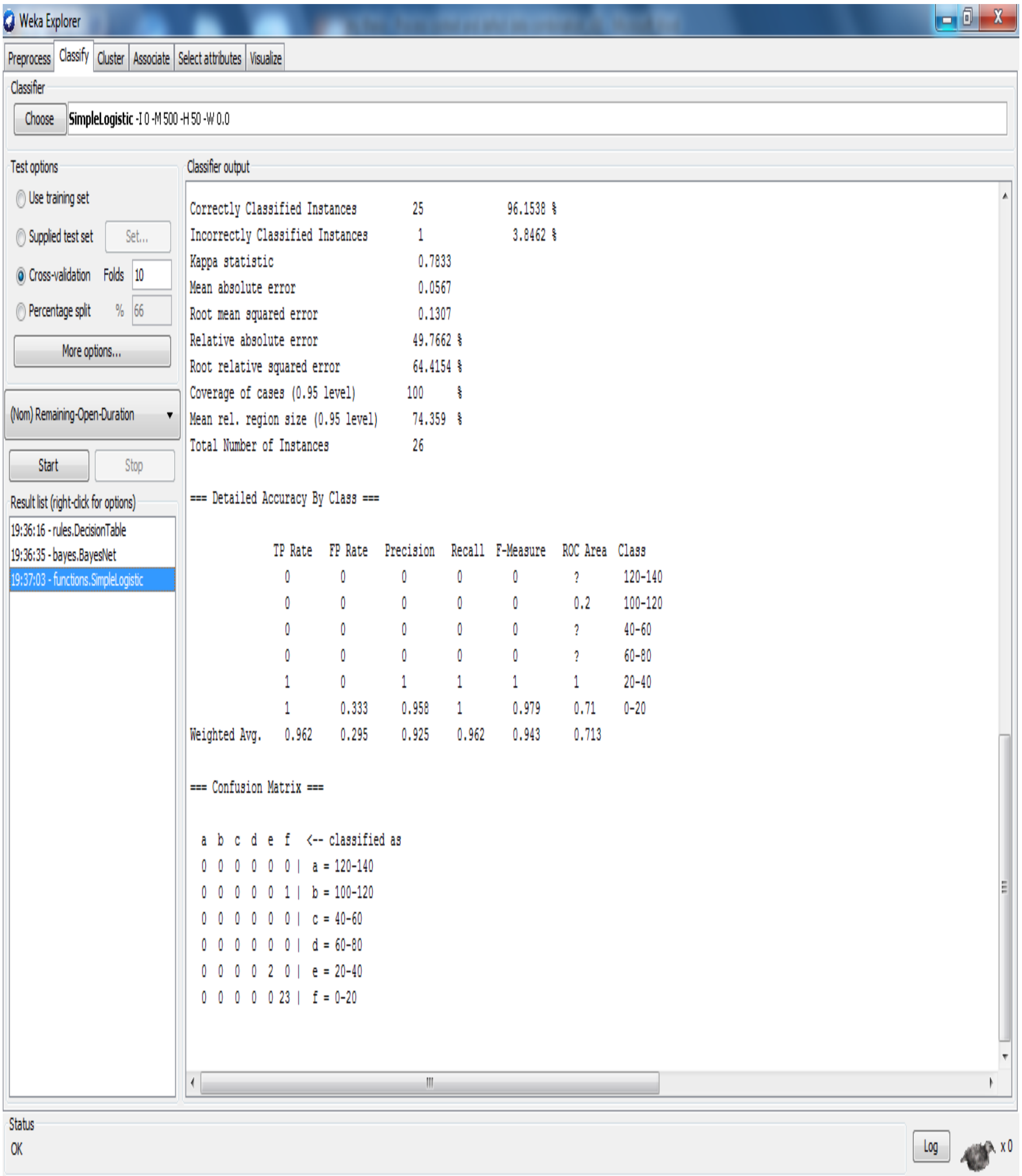

Figure E.10 Simple Logistic Results of Case Study 2B Cluster 1

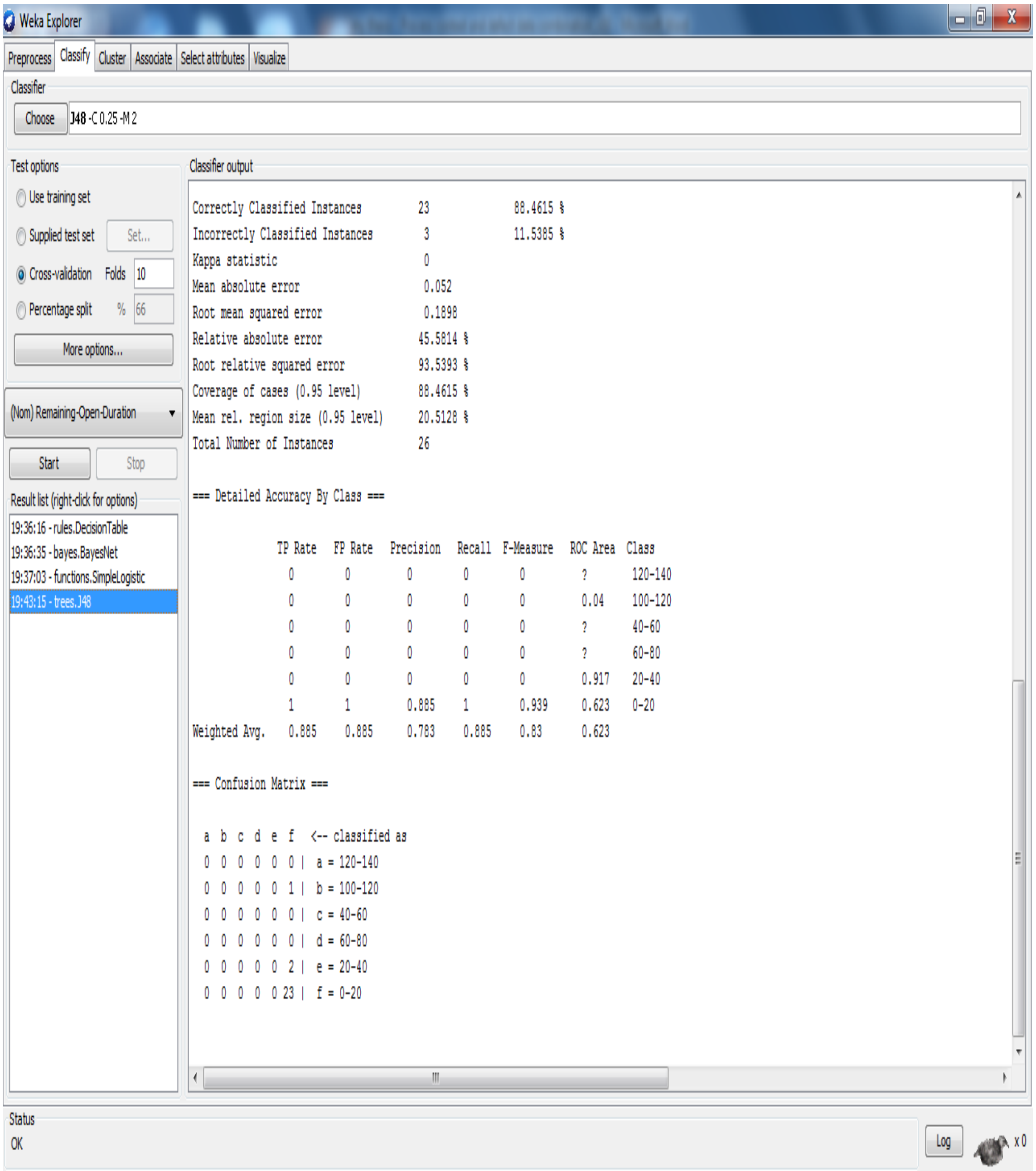

Figure E.11 J48 Results of Case Study 2B Cluster 1

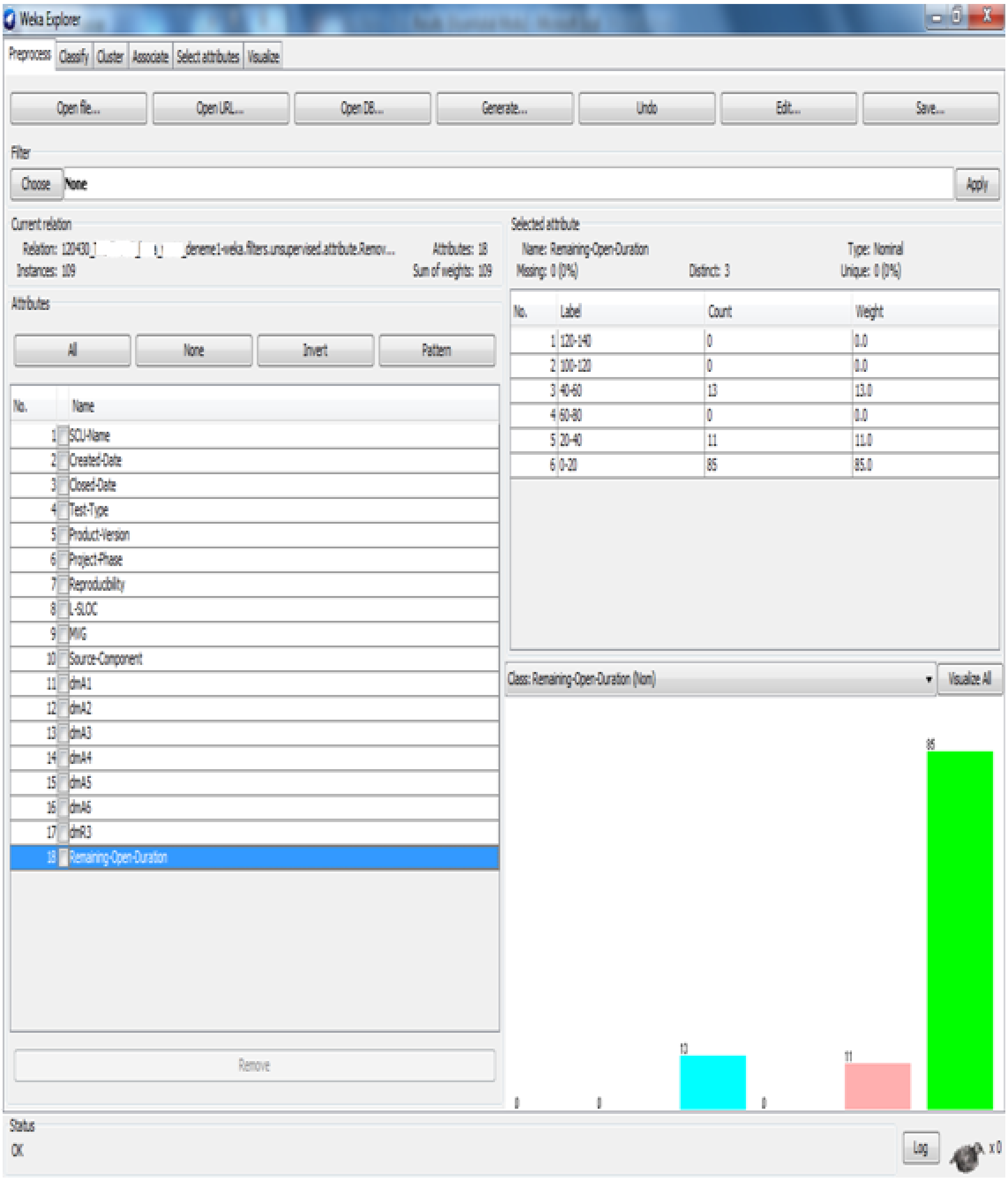

Figure E.12 Weka View of Case Study 2B Cluster 2

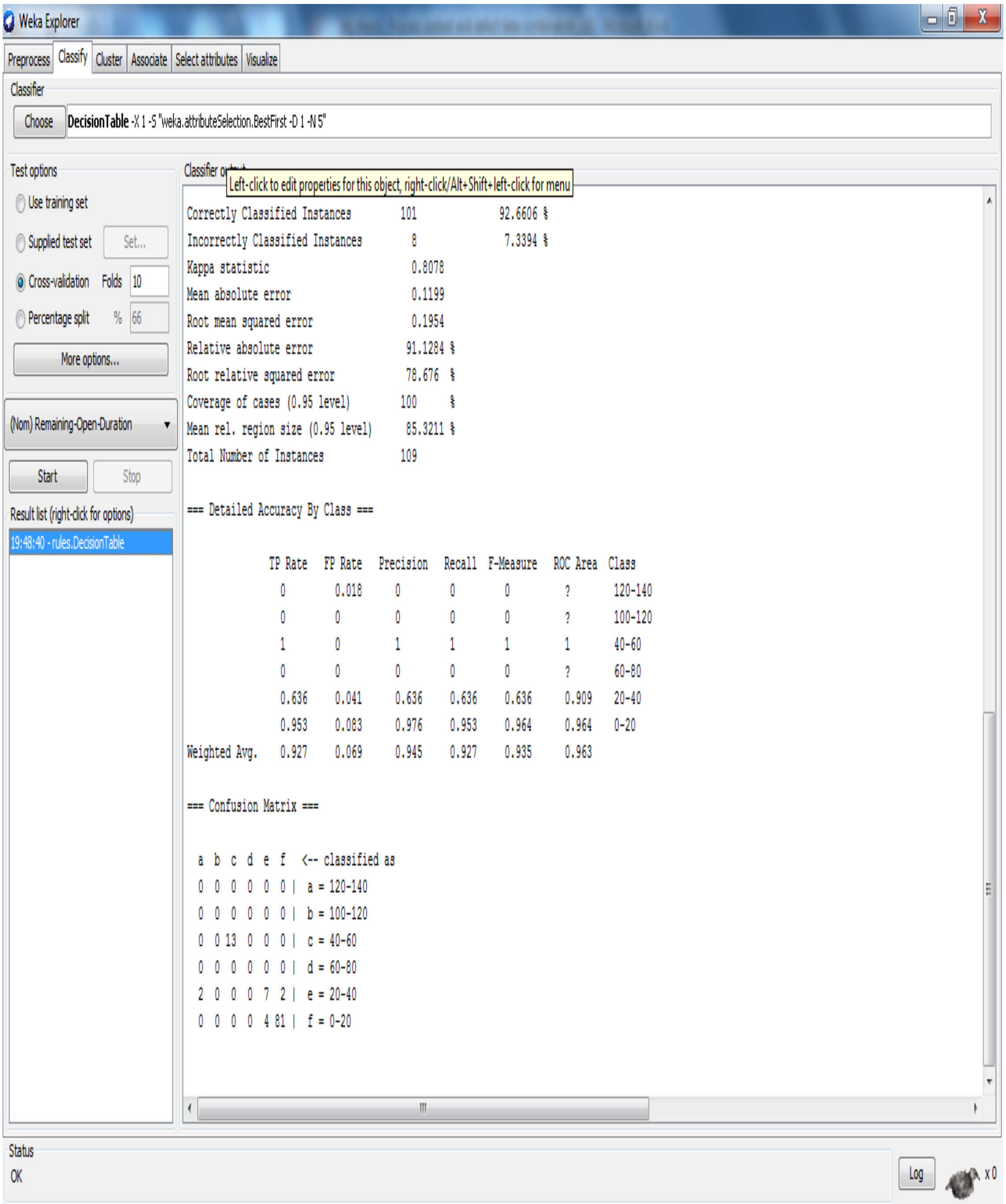

Figure E.13 DecisionTable Results of Case Study 2B Cluster 2

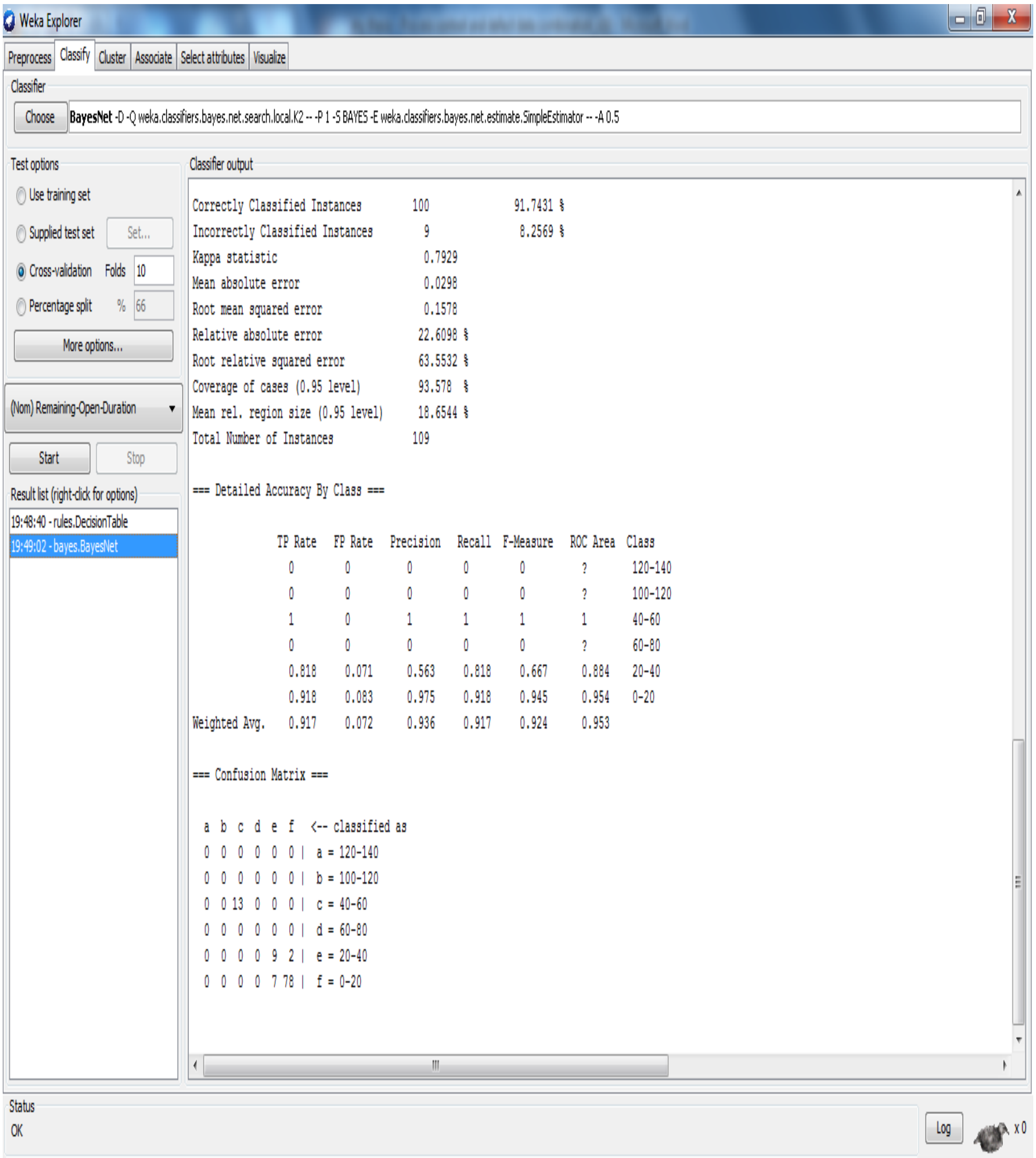

Figure E.14 BayesNet Results of Case Study 2B Cluster 2

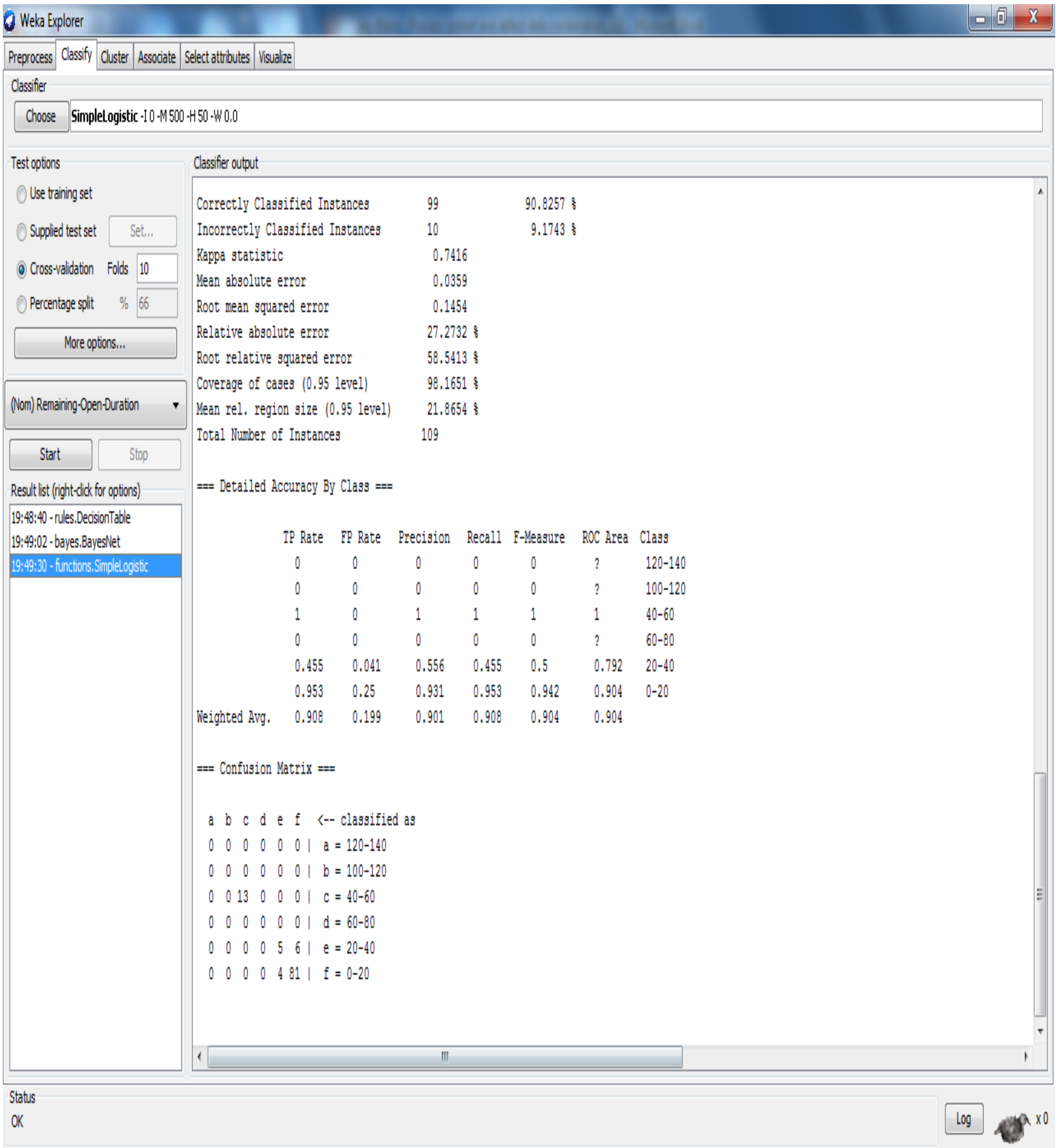

Figure E.15 SimpleLogistic Results of Case Study 2B Cluster 2

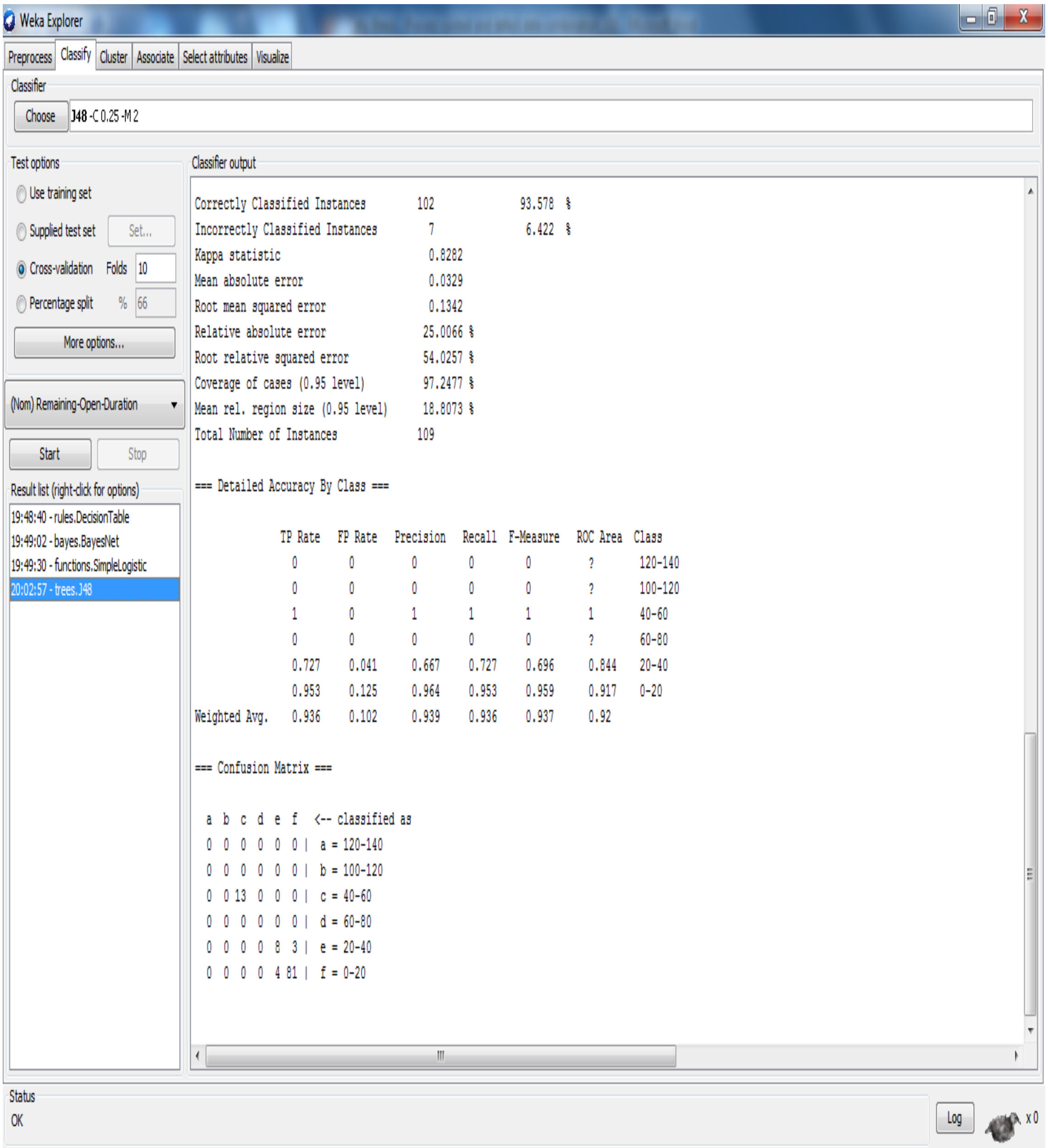

Figure E.16 J48 Results of Case Study 2B Cluster 2

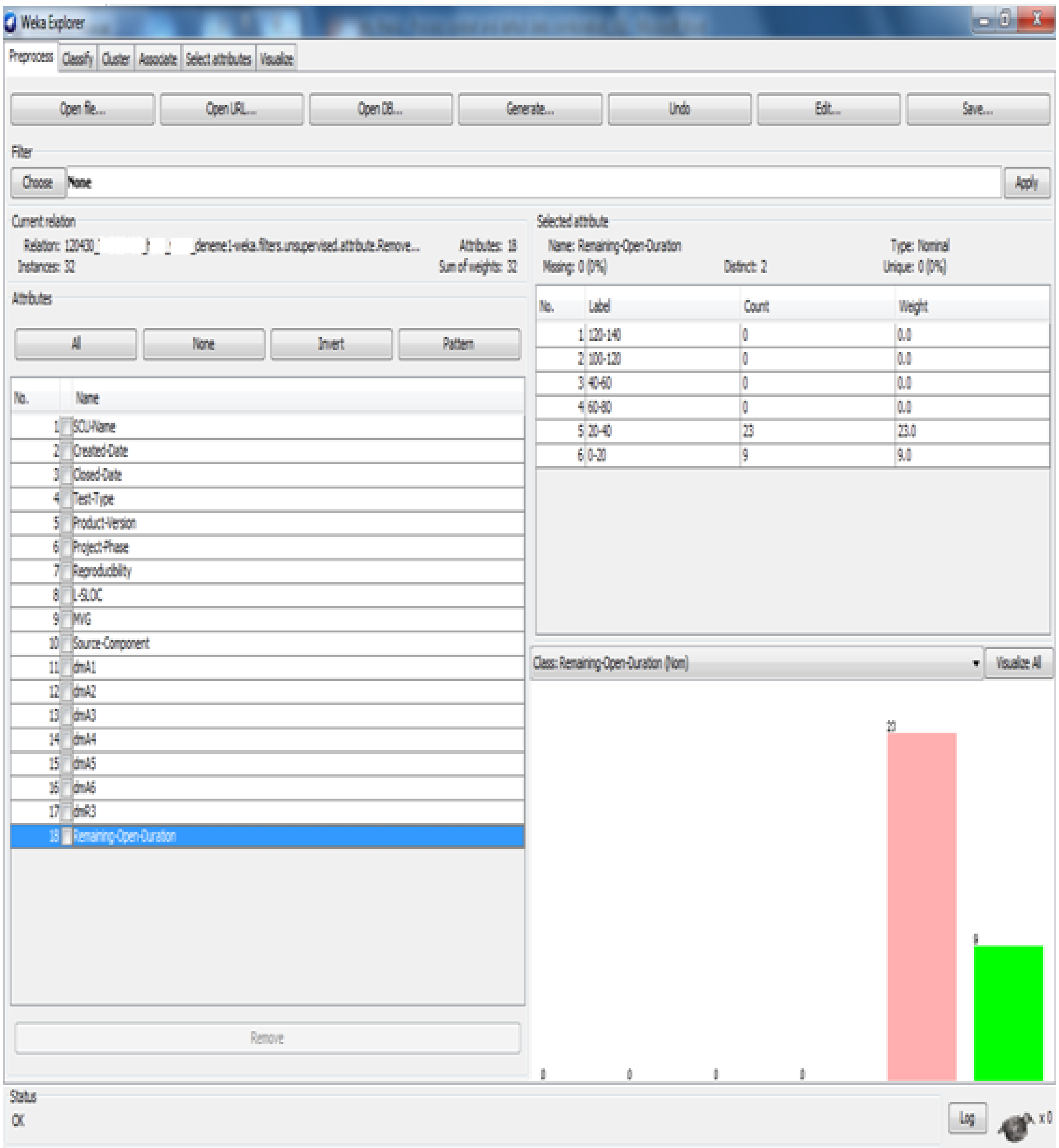

Figure E.17 Weka View of Case Study 2B Cluster 3

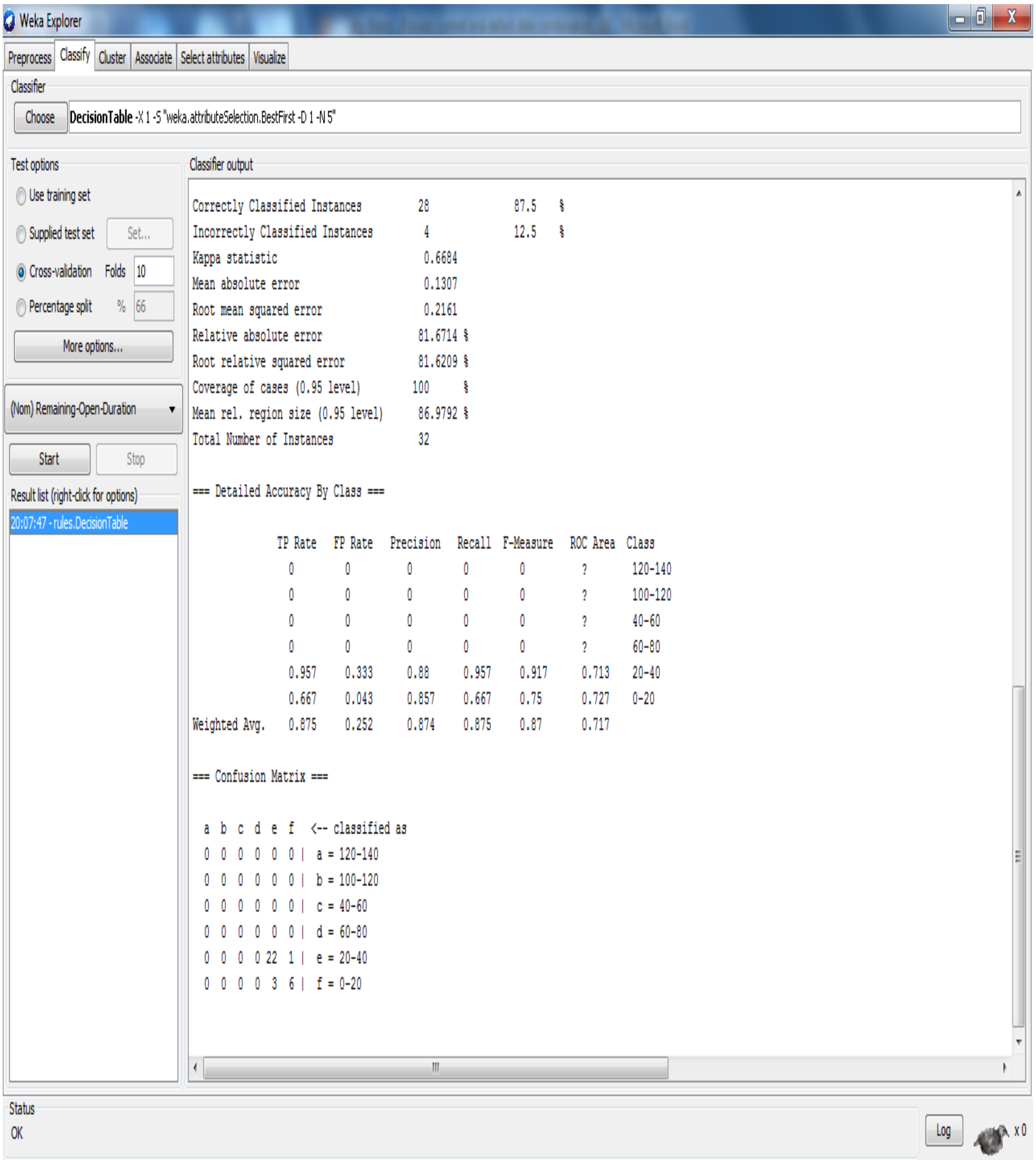

Figure E.18 DecisionTable Results of Case Study 2B Cluster 3

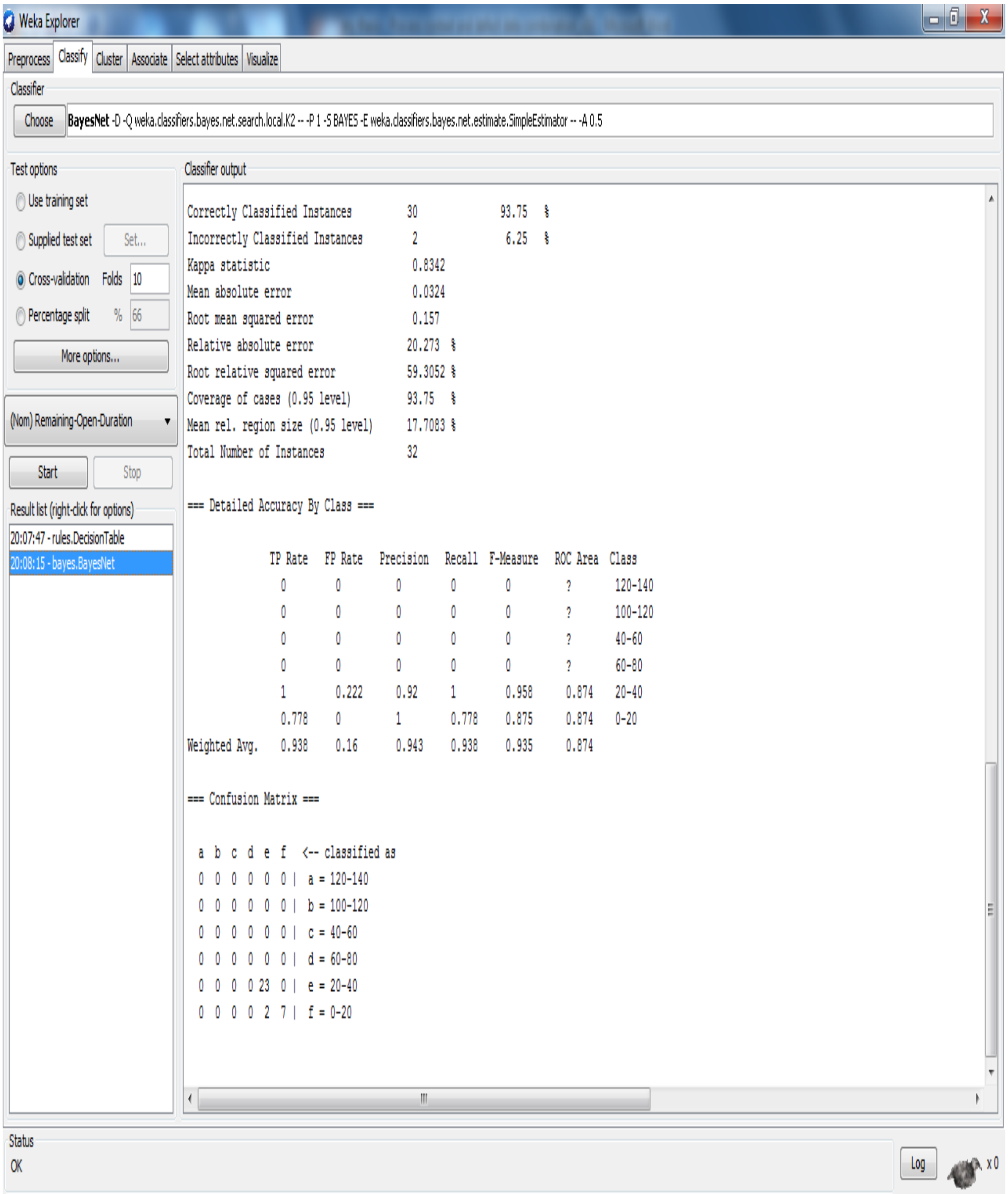

Figure E.19 BayesNet Results of Case Study 2B Cluster 3

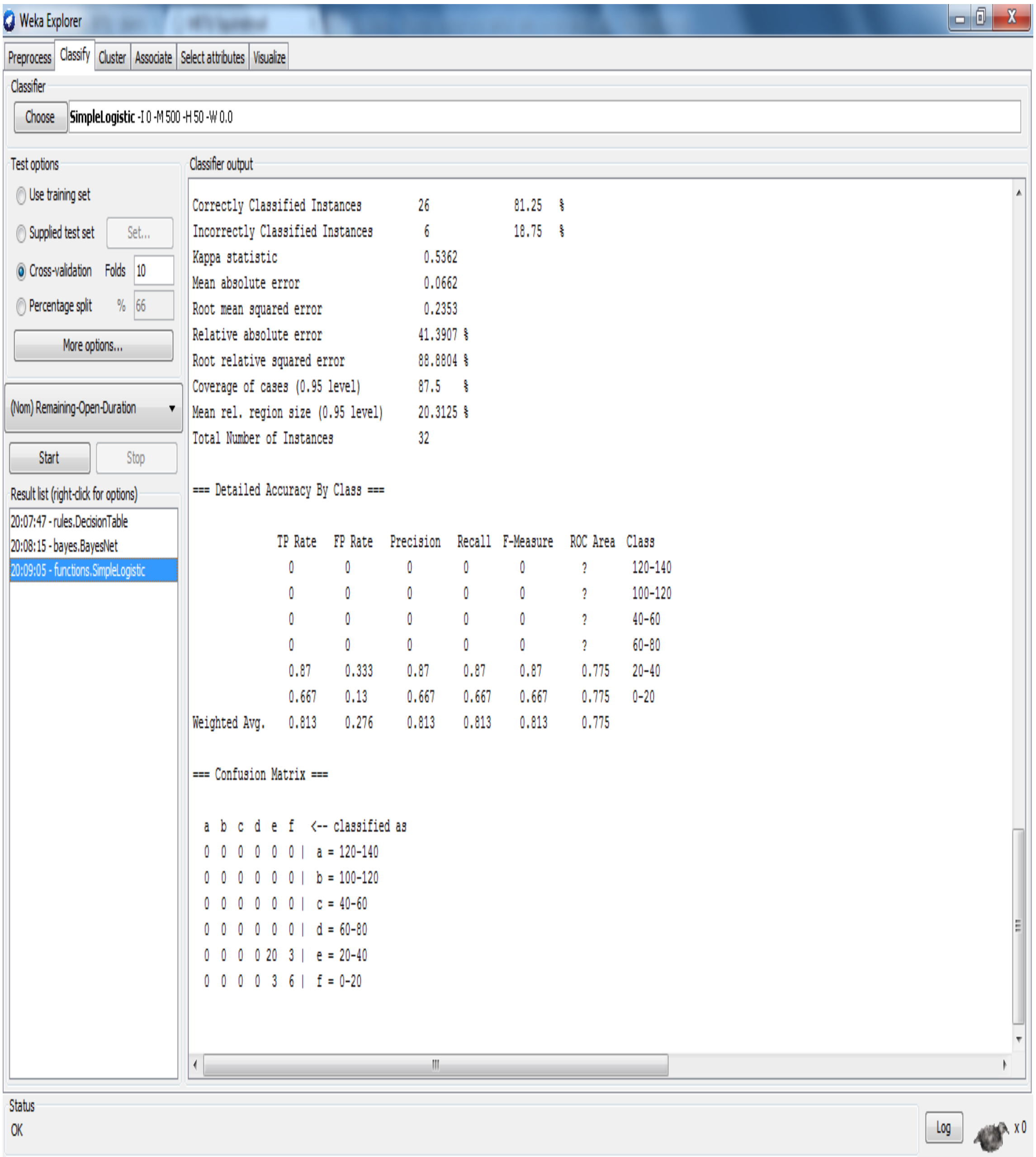

Figure E.20 SimpleLogistic Results of Case Study 2B Cluster 3

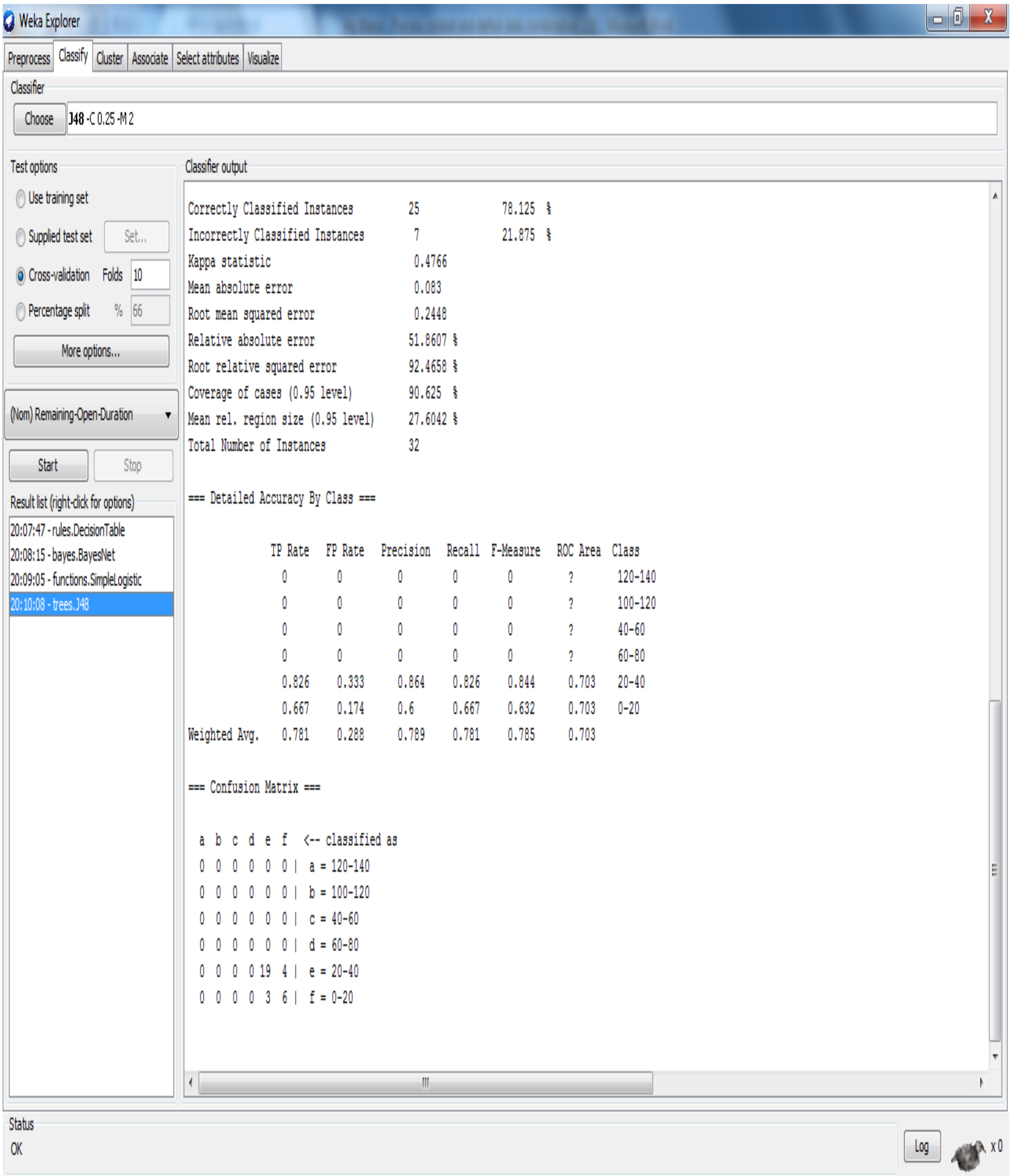

Figure E.21 J48 Results of Case Study 2B Cluster 3

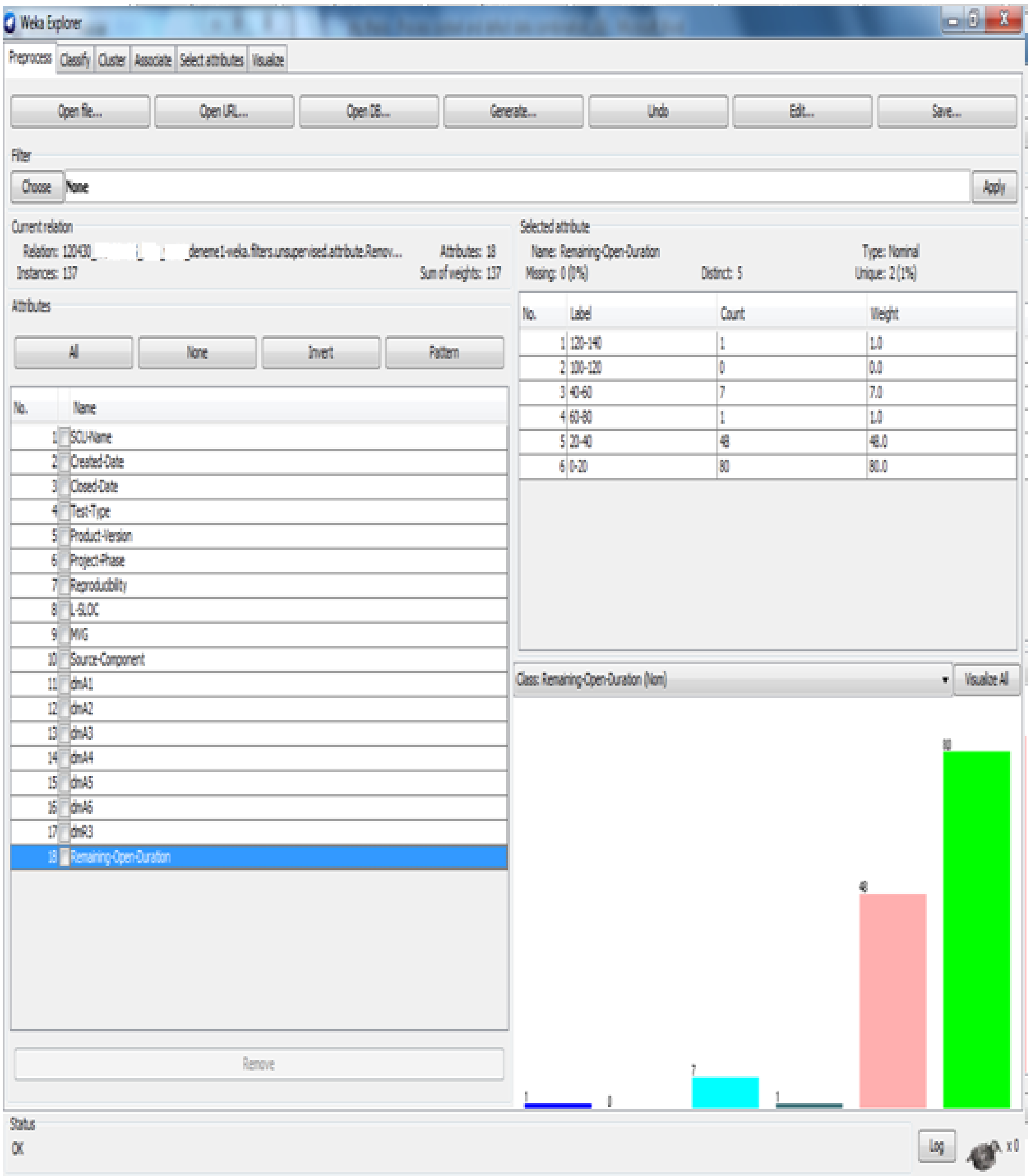

Figure E.22 Weka View of Case Study 2B Cluster 4

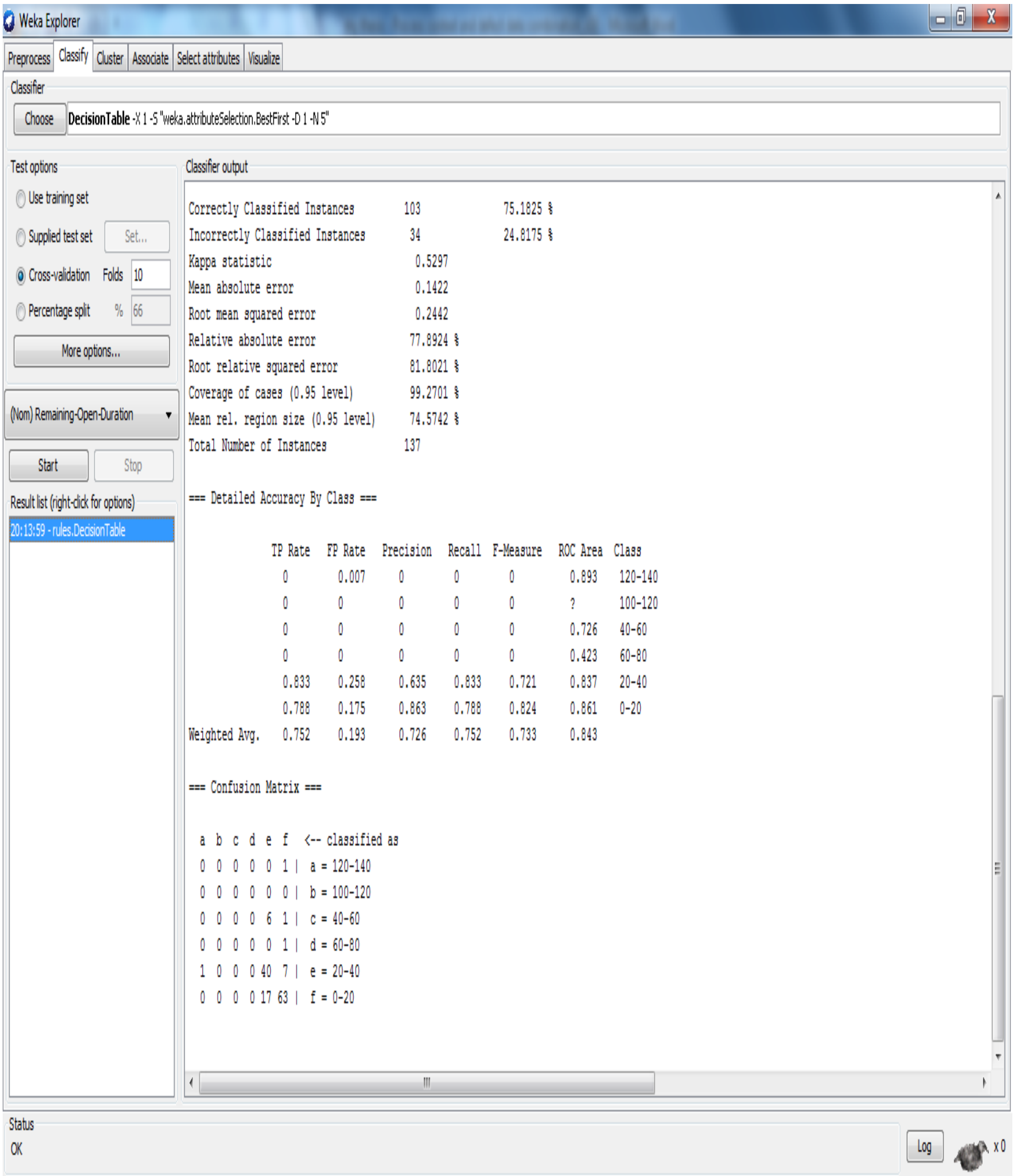

Figure E.23 DecisionTable Results of Case Study 2B Cluster 4

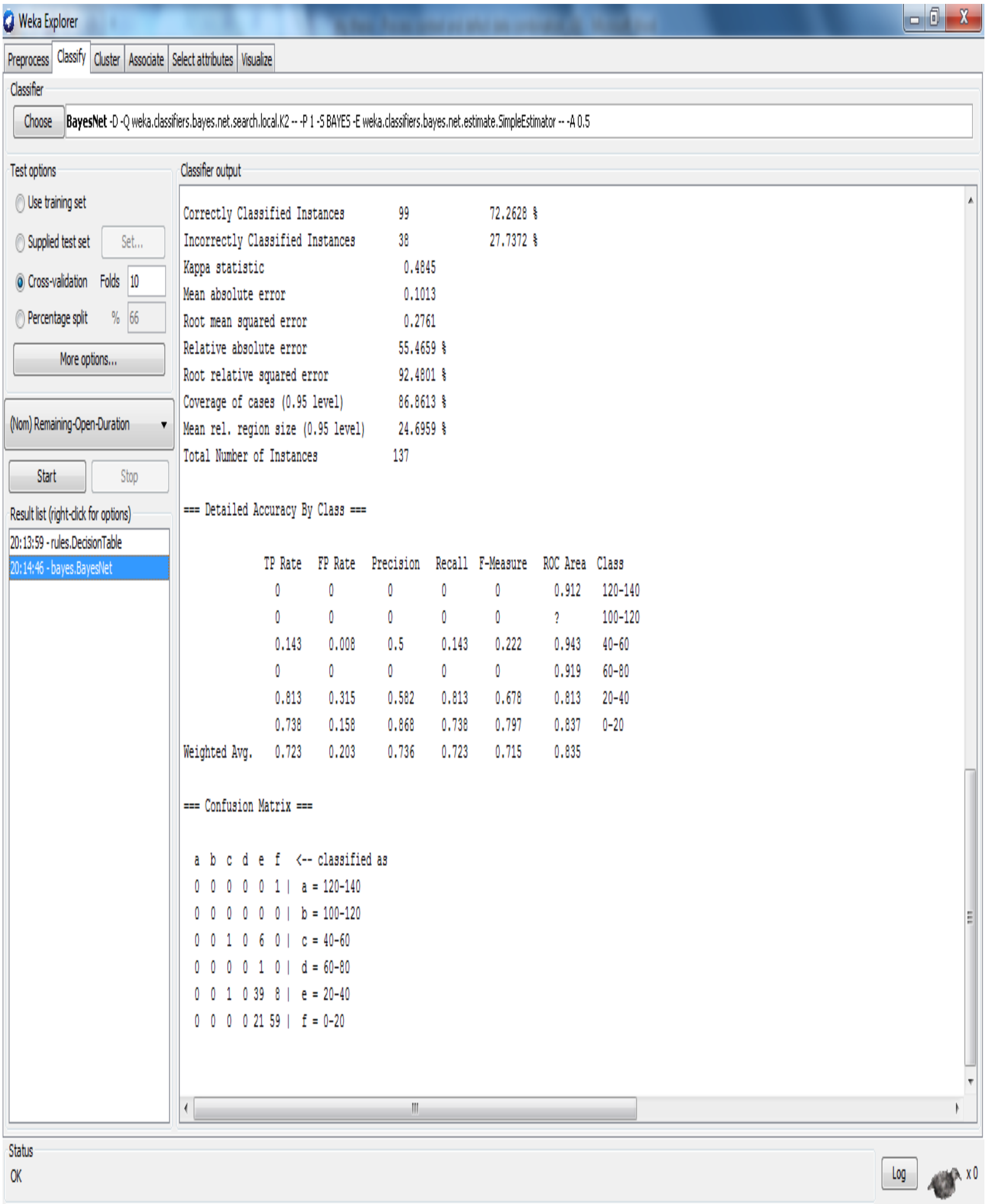

Figure E.24 BayesNet Results of Case Study 2B Cluster 4

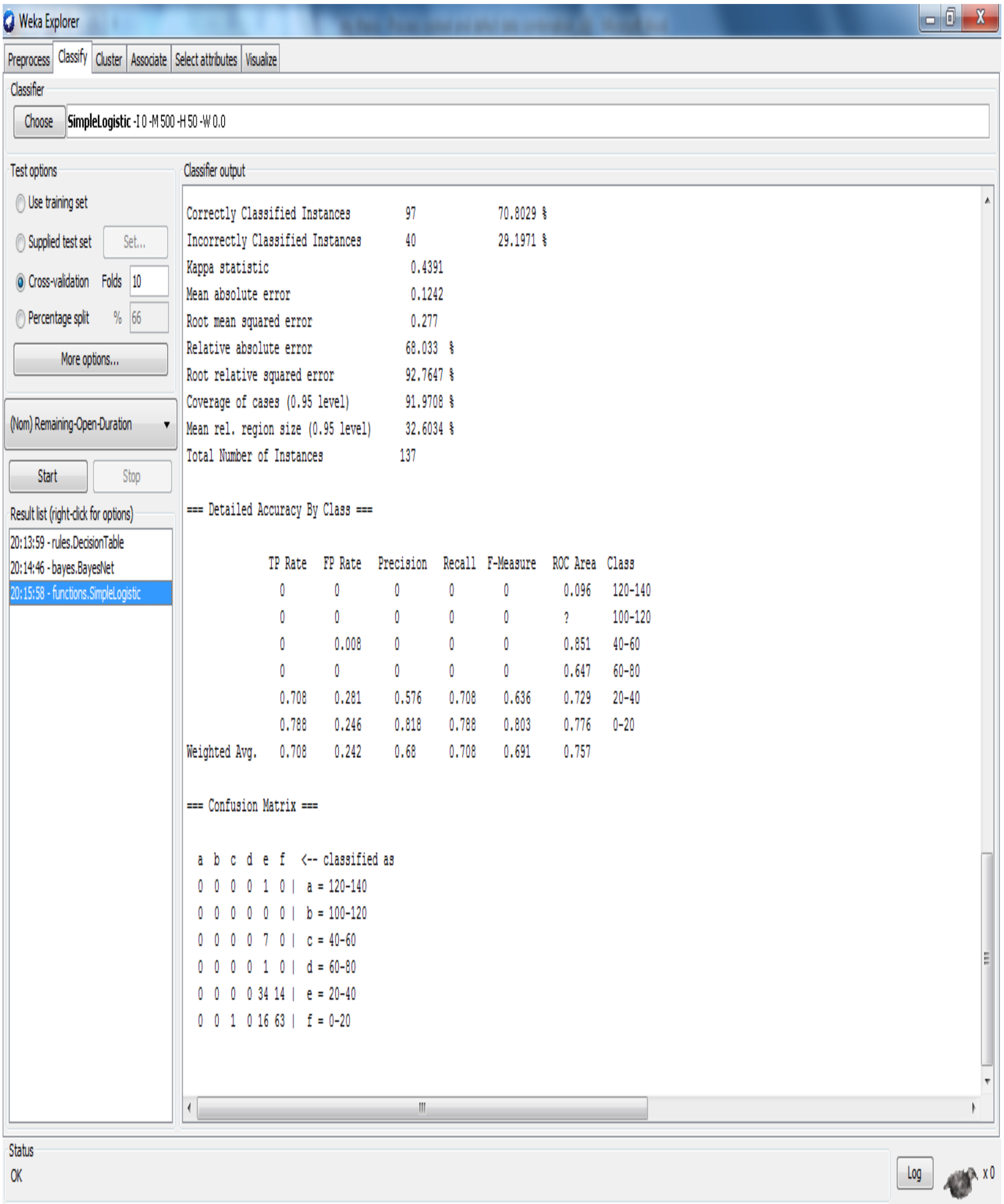

Figure E.25 SimpleLogistic Results of Case Study 2B Cluster 4

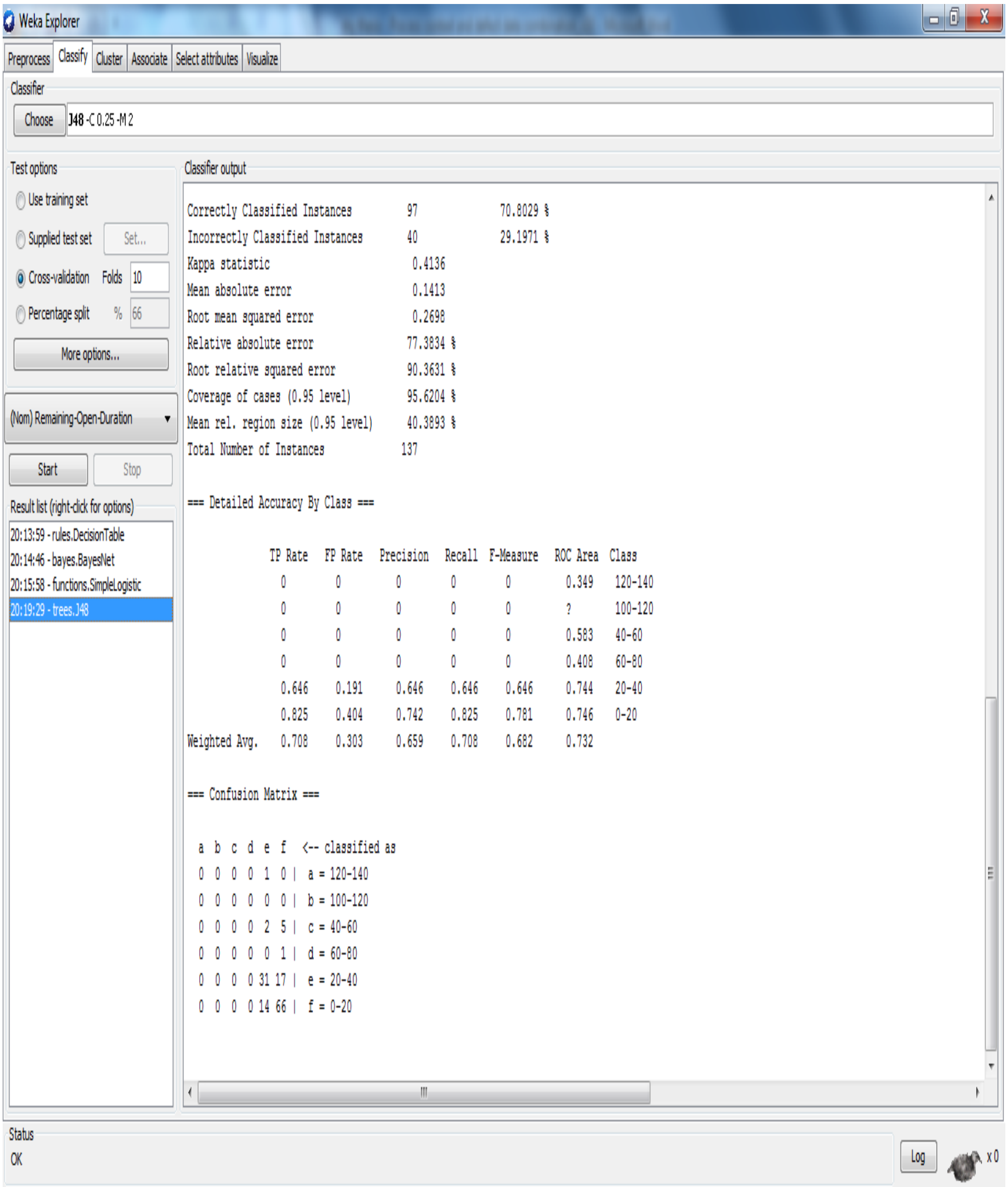

Figure E.26 J48 Results of Case Study 2B Cluster 4

METU LIBRARY

TEZ FOTOKOPİ İZİN FORMU

ENSTİTÜ

Fen Bilimleri Enstitüsü ☐

Sosyal Bilimler Enstitüsü ☐

Uygulamalı Matematik Enstitüsü ☐

Enformatik Enstitüsü ☐

Deniz Bilimleri Enstitüsü ☐

YAZARIN

Soyadı : ....
Adı : ....
Bölümü : ....

TEZİN ADI (İngilizce) : ....
....
....
....
....

TEZİN TÜRÜ : Yüksek Lisans ☐ Doktora ☐

1. Tezimin tamamı dünya çapında erişime açılsın ve kaynak gösterilmek şartıyla tezimin bir kısmı veya tamamının fotokopisi alınsın. ☐

2. Tezimin tamamı yalnızca Orta Doğu Teknik Üniversitesi kullancılarının erişimine açılsın. (Bu seçenekle tezinizin fotokopisi ya da elektronik kopyası Kütüphane aracılığı ile ODTÜ dışına dağıtılmayacaktır.) ☐

3. Tezim bir (1) yıl süreyle erişime kapalı olsun. (Bu seçenekle tezinizin fotokopisi ya da elektronik kopyası Kütüphane aracılığı ile ODTÜ dışına dağıtılmayacaktır.) ☐

Yazarın imzası .... Tarih ....