# DEVELOPMENT OF A DATA MINING SOFTWARE ON HIGHER EDUCATIONAL DATA

OSMAN YÜCEL

BOĞAZİÇİ UNIVERSITY

2012

DEVELOPMENT OF A DATA MINING SOFTWARE ON HIGHER EDUCATIONAL DATA

Thesis submitted to the

Institute for Graduate Studies in the Social Sciences

in fulfillment of the requirements for the degree of

Master of Arts

in

Management Information Systems

by

Osman Yücel

Boğaziçi University

2012

Thesis Abstract

Osman Yücel, "Development of A Data Mining Software of Higher Educational Data"

The purpose of this study is to develop a software which anlayzes the student data in Bogazici University, creates association rules and makes suggestions to the students about the course selection.

The software works on two datasets. The first dataset accepts the grades as they are, so the grade set consists of 8 grades (AA, BA, BB, CB, CC, DC, DD and F). The second dataset groups similar grades and processes them together, so the gradeset consists of 4 grades such as (HIGH(AA-BA),MID(BB-CB,CC),LOW(DC-DD), FAIL(F)).

The software consists of two main parts. The first part is the "Rule Creator Tool", which analyzes the past data in the Bogazici University database and creates the rules. In rule creating part, the software uses association rules mining algorithm.

The second part is the "User Tool", which analyzes one student's past data and finds the rules which are suitable for that student. After finding suitable rules, the software makes suggestions to the students about the course selection.

The result of the study shows that the prediction accuracy may be increased about 2 times better than the naïve approach and 1.5 times better than the majority approach. This shows the data is suitable to run association rules mining algorithm.

Keywords: Education, Association Rules Mining, Students, Advisors, Decision Support System (DSS), Grade Optimization, Course Selection

Tez Özeti

Osman Yücel, "Yüksek Eğitim Verisi Üzerinde Veri Madenciliği Yazılımı Geliştirilmesi"

Bu çalışmanın amacı Boğaziçi Üniversitesi'ndeki öğrenci verilerini inceleyip, birliktelik kuralları yaratıp öğrencilere ders seçimi konusunda tavsiyeler verecek bir yazılım geliştirilmesidir.

Geliştirilen yazılım iki farklı veri kümesi üzerinde çalışmaktadır. İlk veri kümesi, notları olduğu gibi kabul ederek çalışır, böylece not kümesi 8 nottan oluşur (AA, BA, BB, CB, CC, DC, DD and F). İkinci veri kümesi benzer notları gruplandırarak birlikte işler, böylece not kümesi 4 nottan oluşur (HIGH(AA-BA),MID(BB-CB,CC),LOW(DC-DD), FAIL(F)).

Yazılım iki ana kısımdan oluşmaktadır.İlk kısım, "Kural Yaratıcı Araç", Boğaziçi Üniversitesi'ndeki geçmiş veriyi inceleyerek kuralları yaratır.Bu kısımda yazılım birliktelik kuralları madenciliği algoritmasını kullanır.

İkinci kısım "Kullanıcı Aracı", bir öğrencinin geçmiş verilerini inceler ve o öğrenci için geçerli kuralları bulur.Kuralları bulduktan sonra öğrenciye ders seçimi ile ilgili tavsiyeler verir.

Çalışmanın sonuçları bu yazılımı kullanarak yapılan tahminlerin isabet oranlarının rastgele yaklaşıma göre 2 kat, çoğunluk yaklaşımına göre 1,5 kat daha yüksek olduğunu göstermiştir. Böylece öğrencilerin aldıkları dersler ve notların birbirleri ile ilgisi, veri madenciliği yöntemlerinin kullanmaya uygun olduğunu göstermiştir.

Anahtar Kelimeler: Eğitim, Birliktelik Kuralları Madenciliği, Öğrenciler, Danışmanlar, Karar Destek Sistemleri (DSS), Not Optimizasyonu, Ders Seçimi

ACKNOWLEDGEMENTS

First of all I would like to thank my supervisor, Prof. Dr. Birgül Kutlu Bayraktar, for all her guidance, mentorship and last but not least, for her patience.

I want to thank Prof. Dr. Meltem Özturan and Dr. Suzan Üsküdarlı for spending time to help me finish my thesis.

I want to thank Asst. Prof. Bertan Badur for sharing his ideas and questions with me and also for listening my ideas and questions.

I want to thank Aysun Bozanta, Can Aytekin and Gülşah Yılmaz my dear colleagues, for always being there for me whenever I needed or I felt down in spirits.

I want to thank to Gökçen Çırakoğlu, Gülsüm Demirci, Figen Bacıoğlu, Arzu Tuncalp, Burcu Kör and Hanife Bocuoğlu for creating a great environment to work in.

I want to thank Gizem Dilber, who gave me the motivation to study and for making it possible for me to attend a PhD program.

I want to thank Mustafa Ali Acar, Yakup Kale and Gürsoy Yazaroğlu for sharing their ideas, and also creating perfect distractions whenever I got tired.

And at last I would like to thank my family for everything, because they did every thing for me.

Finally, I would like to thank everybody who was important to the successful realization of this thesis, as well as expressing my apology that I could not mention personally one by one.

CONTENTS

TABLES

FIGURES

# CHAPTER 1 : INTRODUCTION

## Definition and Significance of the Problem

One of the main factors that affect a student's success in the university is the selection of the right courses for themselves. One of the reasons which causes the students to get low grades is that they select the courses which are not good for them. The students need a system to help them with course selection so the selection of the courses, which will lead the students to get low grades, may be prevented.

The main source for the students to select the courses is word of mouth. They mainly choose the courses which their friends suggest. But as their friends may have different perspective or different skills, their suggestion may not be the best solution for respective student. On the other hand most of the schools have the advisory system, where each student has an instructor assigned to them as an advisor. But as one advisor has many students these advisors have neither the time nor the energy to research and help the students to select the best courses for them.

The reasons stated above shows that there is a need for a system which will advise the best courses for each student to select, in order to be successful. Such a system would increase the chance of the students to get higher grades, without taking that much time of advisors.

By the increasing computing power and high data storage, data mining has become one of the subjects which is used by many companies in many sectors. It is used to make more accurate predictions about the future, using past data, in many areas. So it can be said that data mining procedures may be used in order to make suggestions to the students, to help their course selection process.

## Goal of the Study

This study aims to create a system which shows the best courses to take for a student when the past information is entered to the system as the input. This way the students may achieve better GPAs' in their educational life. Universities would also benefit from this tool as they will have a better average of student GPA among the university. Also as the students would learn what they will be better at, the market would have a better qualified range of employee candidates. The outputs of such a system will also help while scheduling the courses which will be proposed for a semester. If a course will be a good fit for many students, university may decide to offer that course for that term.

# CHAPTER 2 : LITERATURE REVIEW

## Knowledge Discovery in Databases

Knowledge Discovery in Databases (KDD) has started to gain importance with the improved usage of computers and database algorithms. KDD is defined as the process of the identification of valid, novel, potentially useful, and ultimately understandable patterns in data (Fayyad et al., 1996).

KDD is widely used in the fields which have large amounts of historical data in their databases. It has reached in many areas in the process too.

In banking, KDD process is used to detect frauds. All the accounts and transactions are worked through the KDD process. New transactions and accounts are analyzed and the suspicious transactions are marked for further inspection. KDD is also used in banking for risk and portfolio management (Dass, 2006).

In marketing KDD is used to predict buying behaviors and trends. The products which are bought together are discovered using KDD steps and new marketing strategies are created (Shaw et al., 2001).

In sports KDD is used to find the best combination of players. The successes of the line-ups are calculated to find out which line-up can give the best result. Also teams make the decisions about the transfers according to the results of KDD steps (Fayyad et al., 1996).

KDD procedure has 5 steps to convert the data to knowledge. They are listed as (Fayyad et al., 1996):

1. Selection
2. Preprocessing
3. Transformation

4. Data Mining
5. Interpretation / Evaluation

Selection part is basically selecting which data and which properties are to be used in the KDD processes. It consists of two kinds of selection: selecting rows and selecting columns. Selecting rows means which data is to be used in KDD, for example if the analysis is going to be done on a list of customers, some customers may be excluded from the analysis. On the other hand selecting the columns means which attributes are to be used in KDD, for example on the same analysis, the apartment number or the postal code of the customers may be ignored if it is not relevant to the case (Fayyad et al., 1996).

Preprocessing step is converting the data into the kind of variables, which data mining algorithms can work on. For example customer addresses cannot be processed as a long text variable so it should be converted to a few variables such as city, district, postal code and etc. If this is not done as formulas which calculates the similarity of customers cannot work on the data, so the customers will not be considered similar. Data preprocessing is used to achieve (Berka & Bruha, 1998; Han, Kamber & Pei, 2012):

- accuracy,
- completeness,
- consistency,
- timeliness,
- believability and
- interpretability.

Transformation step is where the variables are transformed into the formats which are suitable for selected data mining algorithm. For example if an algorithm based on a mathematical method is selected, the name of the district cannot be used directly in the algorithm. So if it is needed to use the district information in the analysis, we need to transform it into some number variable such as number of stores in the districts or the distance of this district to the closest store in kilometers (Fayyad et al., 1996).

Data mining step is the step where the calculations are made and algorithm is applied. In this step an algorithm is selected and then applied. The algorithms contain many mathematical operations and formulas which are applied on the data. This part will be explained in detail in the following section.

Interpretation/evaluation step is the last step of KDD where the results of the data mining step are evaluated and decisions are made. Outcomes, which are consistent with the definition of KDD explained in the previous sections, are valid, novel, potentially useful and ultimately understandable. The results are interpreted to some rules or explanations into a language that can be understood by the people who are in the decision making positions (Fayyad et al., 1996).

## Data Mining

The data mining part of KDD process is the part that is covered in the process of this work. Data mining practices can be examined in a number of categories (Fayyad et al., 1996):

- Classification
- Regression

- Clustering
- Summarization
- Dependency Modeling
- Link Analysis
- Sequence Analysis

Classification methods deal with how to separate the records into classes and aim to identify the formulas and combinations which plays role to define which class a record belongs to. When a new record, of unknown class, is presented to the system, the algorithm applies the formula to the new record and guesses which class it will belong to (Phyu, 2009).

Regression methods are used to predict real values for the target variables by defining the formulas and combinations, which determine the target variable. So new records' values are presented to the formulas the results will be calculated to predict a value for that new record (Larose, 2006).

Clustering methods do not actually work on a specific variable; instead it only divides the records into a number of sets and segments according to its variables. So it actually creates the segments and leaves the interpretation to the decision-making people. This way the strategies for every segment can be set (Berkhin,2001).

Summarization methods are mostly used for reporting purposes. The summarization methods include some function sets and visualization techniques to summarize the whole data into some reports or graphics which can easily be read and understood by the people (Fayyad et al., 1996).

Dependency modeling methods determine if the variables have any relationship between the variables such as if there is a relationship between the income and the district that the customer lives in. Dependency modeling methods are used to discover the relationships and the strength of the relations (Ensel, 2001).

Link analysis methods are used to determine the correlation between the entities, such as which products are bought together or which movies are liked by the same people. These methods are going to be explained in detail in the following sections.

Sequence analysis methods are used to make predictions for future using time-series data. Predicting the stock values and the prices of gold or petrol are good examples of sequence analysis (Fayyad et al., 1996).

## Association Rules

Association rules method is the method which is to be examined as one of the link analysis methods in this study. Association rules are frequently used for analyzing the event occurrences and finding the probabilities and frequencies of the sets of events which happen together (Agarwal et al. 1993).

In this method two parameters are set in the beginning. The first parameter is the support count value, which is the minimum number of times that an event or a set of events should occur in order to be considered as an interesting set. This may also be used as a support value which is the support count divided by total number of occurrences of all events. The second parameter is the confidence value, which is minimum probability of a rule produced by the algorithm in order to consider the rule interesting (Agarwal et al. 1993).

Table 1. Transactions

| Customer | Products Bought |
|---|---|
| Customer 1 | A B C |
| Customer 2 | A B C D |
| Customer 3 | A C E |
| Customer 4 | B C D |
| Customer 5 | B C D E |

## Apriori Algorithm

Apriori Algorithm is used in order to find which sets are frequent. In this algorithm the items are counted. In this step all the items form sets each having one element. The list of these sets are called $C_1$ (Candidates Size 1). After that step, the sets which does not satisfy the minimum support condition are eliminated. The list of remaining sets is called $L_1$(Large Sets Size 1). Then these steps are repeated, by increasing the size one by one, until the $L_k$ is an empty list. At each step $C_k$ is created by combining all sets which are in $L_{k-1}$ (Agarwal et al. 1994).

For example let the support count value be 3 and the confidence value be 0.70. In Table 1 the customers and the products they bought are shown.The events are counted first and the information in Table 2, about the number of transactions, is acquired.

Table 2. The Count of The Events

| Product | Count of Times Bought |
|---|---|
| A | 3 |
| B | 4 |
| C | 5 |
| D | 3 |
| E | 2 |

So product E is eliminated in this step of algorithm, because of occurring less times than predetermined support count value. After this step all the combinations of remaining events are created and counted. When counting the sets, the number of purchases of the products by the same customer is found.

Table 3. The Count of Event Sets Count 2

| Set | Count of Times Bought |
|---|---|
| AB | 2 |
| AC | 3 |
| AD | 1 |
| BC | 4 |
| BD | 3 |
| CD | 3 |

After this step, only 4 sets have enough number of times bought together, which are AC, BC, BD and CD. The iteration must continue until there are no sets satisfying the condition that the count must be at least equal to the predetermined support value.

So the remaining sets are combined again. In the combination process in order to combine two sets all the members of two sets except the last member must be same.

Table 4. The Count of Event Sets Count 3

| Set | Count of Times Bought |
|---|---|
| BCD | 3 |

After combining 2 member sets, C3 has only one set, which is BCD. That set satisfies the minimum support condition, so it is included in L3. The number of sets in L3 is 1 so there are no combinations to make. So creation of sets step is over.

Rule Creation

So the next step can start where the rules are created. In this part the sets are converted to rules. This conversion is made by choosing each subset as the left side of the rule (excluding the empty set and the whole set itself) and using the remaining elements as the right side. So we can create 14 rules from the 5 sets remaining (AC, BC, BD, CD and ABC).

Table 5. The Rules Created

| | |
|---|---|
| A->C | C->A |
| B->C | C->B |
| B->D | D->B |
| C->D | D->C |
| BC->D | D->BC |
| BD->C | C->BD |
| CD->B | B->CD |

After these rules are created, they are needed to be evaluated. For this the confidence value has to be calculated for each rule. The formula of the confidence is calculated by dividing the support value of the whole set by the support value of the support value of the left side set.

Confidence (C –> A) = SupportCount (A, C)/ SupportCount (C)

Confidence (C -> A) = 4 / 5 = 0, 80

All the rules' confidence values are calculated we have the results given in Table 6.

Table 6. The Rules And The Confidences

| Rule | SupportCount | SupportCountLeft | Confidence |
|---|---|---|---|
| A -> C | 3 | 3 | 1 |
| C -> A | 3 | 5 | 0.6 |
| B -> C | 4 | 4 | 1 |
| C -> B | 4 | 5 | 0.8 |
| B -> D | 3 | 4 | 0.75 |
| D -> B | 3 | 3 | 1 |
| C -> D | 3 | 5 | 0.6 |
| D -> C | 3 | 3 | 1 |
| BC -> D | 3 | 4 | 0.75 |
| D -> BC | 3 | 3 | 1 |
| BD -> C | 3 | 3 | 1 |
| C -> BD | 3 | 5 | 0.6 |
| B -> CD | 3 | 4 | 0.75 |
| CD -> B | 3 | 3 | 1 |

After the rules are created, the ones which do not satisfy the minimum confidence value are eliminated.

In Table 7 the rules are created as a result of the algorithm and their confidence and support values can be seen.

Table 7. The Rules Which Satisfy the Minimum Confidence

| Rule | SupportCount | SupportCountLeft | Confidence |
|---|---|---|---|
| A -> C | 3 | 3 | 1 |
| B -> C | 4 | 4 | 1 |
| C -> B | 4 | 5 | 0.8 |
| B -> D | 3 | 4 | 0.75 |
| D -> B | 3 | 3 | 1 |
| D -> C | 3 | 3 | 1 |
| BC -> D | 3 | 4 | 0.75 |
| D -> BC | 3 | 3 | 1 |
| BD -> C | 3 | 3 | 1 |
| B -> CD | 3 | 4 | 0.75 |

There are a few problems about this approach. First of all some rules may be unnecessary depending on the previous rules. For example there is a rule which says if a customer buys B he/she will also buy D with 0.75 probability. Also there is another rule which says if the customer buys B and C he/she will also buy D with 0.75 probability. So whenever a customer bought B it is known that buying D probability is 0.75 and knowing he/she also bought C actually adds nothing to this knowledge. So another measure called lift is calculated in order to decide which rules are interesting. The formula of lift for rule B->D is (Geng and Hamilton, 2006):

$$Lift (B\text{->}D) = Support (B, D) / Support (B) * Support (D)$$

If this lift value is less than 1 then this rule has negative lift. In that case this rule may not be interesting. Mainly the rules with lift greater than 1 are interesting. The rules with lift less than 1 show that the occurrence of one event decreases the possibility of the other. If the lift is equal to 1 we can say occurrence of one event does not change the possibility of the other event so the rule is completely uninteresting, so those rules are eliminated. In Table 8 the lift values for the rules can be seen.

Table 8. The Rules And The Lift Values

| Rule | SupportCount | SupportCountLeft | Confidence | Lift |
|---|---|---|---|---|
| A -> C | 3 | 3 | 1 | 1 |
| B -> C | 4 | 4 | 1 | 1 |
| C -> B | 4 | 5 | 0.8 | 1 |
| B -> D | 3 | 4 | 0.75 | 1.25 |
| D -> B | 3 | 3 | 1 | 1.25 |
| D -> C | 3 | 3 | 1 | 1 |
| BC -> D | 3 | 4 | 0.75 | 1.25 |
| D -> BC | 3 | 3 | 1 | 1.25 |
| BD -> C | 3 | 3 | 1 | 1.25 |
| B -> CD | 3 | 4 | 0.75 | 1.25 |

As it can be seen in Table 5 any time the right side consists of C, the confidence value is very high. This occurs when a product (in this example C) is bought very frequently. Such as if every customer of a store buys bread when they shop, some rules with very high confidence will occur such as the customers who buy chairs will buy bread or whomever buys a computer will also buy bread. In order to solve this problem, another value is calculated called importance. The formula of the importance value is calculated as (Seidman, 2001):

C (B & not D) = the number of times that B occurred but D did not occur

C (not D) = the number of times D did not occur

$$\text{Importance (D->B)} = \text{Log}_{10}\ (\text{Confidence (D->B)} / (\text{C (B \& not D)} / \text{C (not D)}))$$

When the importance values for the rules are calculated we get the results given in Table 9.

Table 9. The Rules and The Importance Values

| Rule | SupportCount | SupportCountLeft | Confidence | Lift | Importance |
|---|---|---|---|---|---|
| B -> D | 3 | 4 | 0.75 | 1.25 | ∞ |
| D -> B | 3 | 3 | 1 | 1.25 | 0.30103 |
| BC -> D | 3 | 4 | 0.75 | 1.25 | ∞ |
| D -> BC | 3 | 3 | 1 | 1.25 | 0.30103 |
| BD -> C | 3 | 3 | 1 | 1.25 | 0 |
| B -> CD | 3 | 4 | 0.75 | 1.25 | ∞ |

As it can be seen in Table 9 the rule which has C on the right side has 0 as importance so they can be ignored. So only 5 of the rules remain in the example and they are:

- B->D

- D->B
- BC->D
- D->BC
- B->CD

By looking at the remaining rules it can be said that whenever a customer buys product D, the customer will also buy product B with 1 probability. The customers who buy product B will also buy product D with 0.75 probability.

## Accuracy Measures

There are many accuracy measures used in data mining studies. In this study we will use 2 of them, majority rule and gini index, so those 2 will be explained in this part.

### Majority Rule

Majority rule accuracy measure is one of the simplest measures for checking the data complexity. This method basically predicts every record as a member of the class which has the majority in the sample. If it is assumed there is a set where 40% of the set belongs to class A, 35% belongs to class B and remaining 25% belongs to class C, this method predicts all of them as class A so will have a 40% accuracy. (Roiger et al., 1997).

### Gini Index

Gini Index is another method to measure the accuracy, which is based on calculating the complexity of the data. In this method every record will be assigned to a class randomly. But this randomness will be based on the distribution of the records in the classes. If it is assumed there is a set where 40% of the set belongs to class A, 35%

belongs to class B and remaining 25% belongs to class C, this method predicts 40% of the records as class A, 35% of the records as class B and 25% as class C. So the results will have a distribution as in Table 10.

As it can be seen in Table 10 , only the underlined cells, which are on the diagonal, are predicted correctly. So in this example the Gini Index is 16+12.25+6.25 = 34.5%. This shows how dispersed the data is. The formula for finding Gini Index the sum of squares of all the probabilities (Atkinson, 1970).

Table 10. The Predictions and The Real Results

| | Prediction | | | |
|---|---|---|---|---|
| Distribution | | A(40%) | B(35%) | C(25%) |
| | A(40%) | 16 | 14 | 10 |
| | B(35%) | 14 | 12.25 | 8.75 |
| | C(25%) | 10 | 8.75 | 6.25 |

## Data Mining In Education

Data mining approaches are expected to have a greater adoption in higher education researchers and administrators. In higher education data mining is expected to have even larger and wider applications. The reason for this prediction is three duties carried by higher education institutions (Luan, 2002):

- Scientific research that relates to the creation of knowledge,
- Teaching that concerns with the transmission of knowledge,
- Institutional research that uses the knowledge for decision making.

The flexibility of the university curriculums, and leaving the decision of the courses to be taken to the students, who lack the experience of making enrollment decisions may lead the students to make some wrong decisions about the course selection. In order to help the students to select the courses which are good for them a recommendation system using data mining techniques, should be built (Vialardi et al., 2009).

Leading universities should have a DSS system to analyze the data of the students to develop better student relationships. The success or failure of the students should be measured, and the trends should be determined. The determination of the weaknesses and strengths of the students is a very important step to make the plans for the upcoming years (Guvenc, 2001).

# CHAPTER 3 : METHODOLOGY

This study aims to build a system to help students to select the courses for a term, depending on their completed courses and grades. In order to achieve that goal, association rules mining method is used to determine which courses may lead the student to get high grades. To use this algorithm, a software is developed in this study.

## Selection of Tools

For the preparation of the software, first the database system needed to be selected. Because of the ease of use and transportability, the database is held in a Microsoft Access file. This helped the building, inspecting and transferring the data into another computer and working there easier.

After the database is selected the programming language which the software will be developed needed to be selected. Because of the problem's nature and ease of programming, object oriented languages are considered. Among those languages JAVA was the most widely used, most compatible and which had the most sources to use. So Java is selected to develop the software.

## Acquiring The Data

The data was taken from the registrar of the Bogazici University. The data was queried and the results were sent as a text document. A small program was written in order to process the text file and convert it to an Access file. After the data was converted to Access, the data was normalized in order to save the space.

## Analysis of the Data

First the data is analyzed in order to gain familiarity with the data and the properties of the students and the courses. Data starts from 1971 and the most recent data in the study is from year 2010. In the data the background data is also kept, but as they cannot be tracked by the universities, they are not used in this study.

The number of students whose data is used in the study is 59,829 and the number of courses used in the study is 5,312. In total 1,811,151 grades taken by the students are used in order to get the algorithm to be able to create the rules.

In the next step, 2 more tables are created. First one is the Rules table, which holds the general properties of the rules, such as confidence, importance and support. The other table is RuleParts table, which holds the grades which takes part in the rules and which side of the rule that grade takes place in. The final design of the database can be seen in Figure 1.

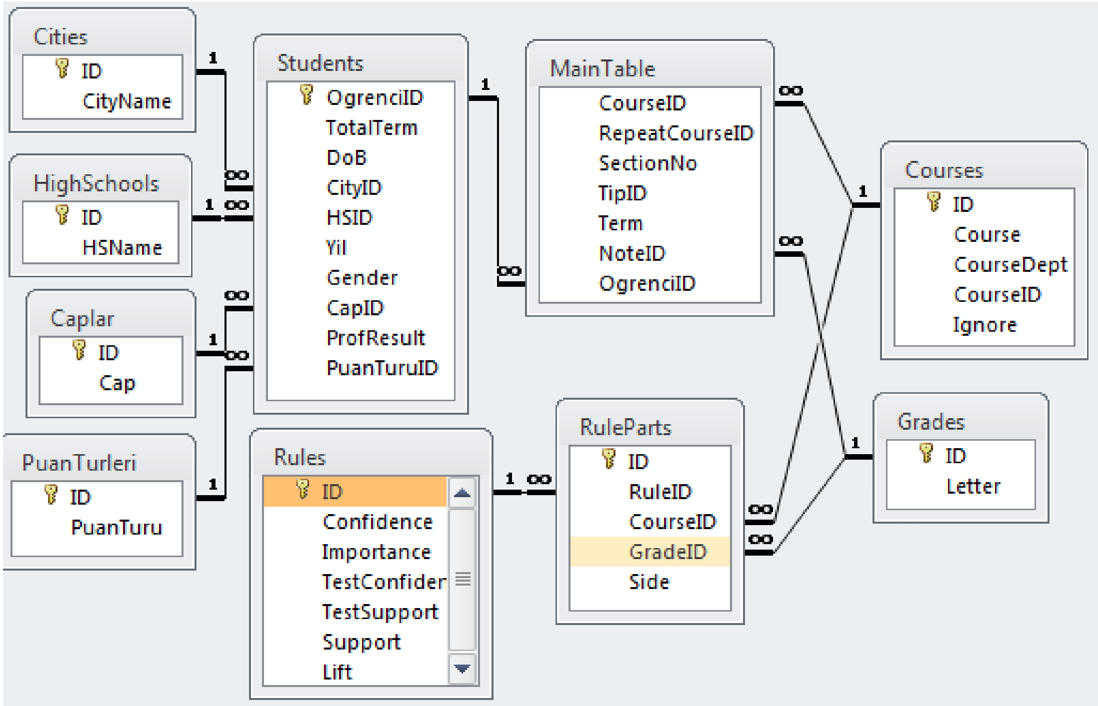

Figure 1 – The relationship diagram of the database

The grade data is used in 2 different ways in this software :

1. There are 8 grades in this approach (AA,BA,BB,CB,CC,DC,DD,F). Each grade is used as they are, such that students who got AA from a course are considered in the same group. In this way less rules are created and the confidence of the rules are lower. On the other hand by this way the rules created are more precise, such as it can suggest a student will get BA from a course.
2. The grade values are grouped into 4, such as HIGH(AA-BA), MID(BB-CB-CC), LOW(DC-DD) and FAIL(F). This way the tool can create more rules, and the rules can have higher confidence values. The drawback of this approach is that, in this way suggestions are selected from these groups, so instead of suggesting that the student will get an AA from the course, this way suggests the student will get a HIGH(AA-BA) grade from this course.

When the counts of grades are checked for the first way, the list is in Table 11.

Table 11. The Counts and The Ratios of Grades in Database

| Grade | Count | Ratio |
|---|---|---|
| AA | 299,701 | 0.187 |
| BA | 236,124 | 0.147 |
| BB | 244,291 | 0.152 |
| CB | 195,177 | 0.121 |
| CC | 183,387 | 0.114 |
| DC | 120,658 | 0.075 |
| DD | 109,402 | 0.068 |
| F | 218,037 | 0.136 |

For the second way the list appears as in Table 12.

Table 12. The Counts and The Ratios of Grade Groups in Database

| Grade | Count | Ratio |
|---|---|---|
| HIGH(AA-BA) | 535,825 | 0.333 |
| MID(BB-CB-CC) | 622,855 | 0.388 |
| LOW(DC-DD) | 230,060 | 0.143 |
| FAIL(F) | 218,037 | 0.136 |

## Data Preparation

Before the tool is developed the data is cleaned and transformed. The first thing done is that, the grades which won't be used in the study are deleted. After this step only the grades AA, BA, BB,CB,CC,DC,DD and F are remaining. The other grades such as NP,W, I and TP are deleted. As this step is performed, 1,606,777 records are left.

## Development of the Software

After the database is prepared the development of the software has started. The tool consists of 2 main parts. First part is called "Rule Creator" and it is actually not to be used by the students. This part will be run once, before each semester begins, and prepare the rules and write them into the database.  The second part of the tool, which is called “User Tool”, will be used by students or advisors in order to see the suggestions which are suitable for a student when the previous grades of the student is given.

### Rule Creator

In this part the first thing done is the minimum support count value is taken as parameter and they are set in related variables.

#### Creation of Sets

First all the grades which are taken by the students more than the predetermined support count value are found by using a query and the first level list is prepared. In this step only the data before 2008 is used. The data after 2008 is going to be used in the testing step. By using the database tools L1 is directly acquired and the step of acquiring C1 is skipped. After the list is acquired, each grade is combined with each other and C2 sets are created. When C2 sets are created, these sets are counted. In this step a problem has occurred. When the list of students are held in the object of sets, as so many sets are created, the lists occupy too much space that the computer runs out of space. So instead of holding the list of every set, the number of students who take the grades in the set is held. This way the problem about the memory is solved, but this time as the lists are acquired again and again each time a set is used its list is retrieved from the database repeatedly. So the creation of sets goes on until no more sets in a level satisfy the minimum support count.

The problem at this step is setting the support count value. If it is too high just a few sets, but really frequent ones, satisfy the condition. But in this case the grades which satisfy the condition are mostly the courses which are required by almost all of the departments. This makes the system ignore even the most important departmental courses. On the other hand if the support count value is set too low then there are too many sets accepted as interesting. This causes two problems:

1. For the courses which are required by almost all departments, almost all possible sets are considered as interesting sets.
2. It makes the program's run time very long.

This set creation part is the main for the slowness of software works. Because the software finds all the subsets, the software works with $O(2^n)$ complexity. As there are 37,541 distinct grades in the system, without the elimination of minimum support count method, it will take creating $2^{37541}$ sets which makes more than $2*10^{11262}$ sets. Even with the support value set as high as 300, more than 935 grades satisfy the condition so $2^{935}$ sets which is around $10^{280}$.

As this study tries to catch all the interesting rules, the support count value is set to 75 for the first data set, which is a fairly low value, and the elimination process is left to rule creation step. For the second dataset the minimum support value is set to 150, because the grades are combined so higher number grades will have higher support count values. At the end of this step all the sets which satisfy the minimum support condition are created.

The next step is defining the minimum confidence value. This value is different for 2 datasets. For the first dataset, with 8 grade values, if majority rule is used and all the grades are guessed as AAs, the program will have 18.7% accuracy. So in order to consider a rule interesting, the confidence of the rule must be higher than this value.

For the second dataset majority rule is used again. If all the rules are guessed as MID(BB-CB-CC), the accuracy will be 38.8%, so this value is set as the minimum confidence for the second dataset. The numbers of grades which satisfy the minimum support count per department can be seen in Table 13. 4G dataset means the dataset used by grouping the grades in to 4 groups, and the 8G dataset is without grouping.

Table 13. The Count of Grades Satisfying the Minimum Support by Department

| DEPT | 8G DataSet | 4G DataSet | DEPT | 8G DataSet | 4G DataSet | DEPT | 8G DataSet | 4G DataSet |
|---|---|---|---|---|---|---|---|---|
| AD | 310 | 200 | ENGG | 7 | 4 | MIS | 73 | 100 |
| AE | 39 | 27 | ENGL | 45 | 24 | MUS | 0 | 1 |
| AL | 8 | 5 | EQE | 1 | 5 | OTL | 9 | 42 |
| AR | 6 | 8 | ESC | 14 | 23 | PA | 9 | 12 |
| ARCM | 0 | 1 | ETM | 7 | 17 | PE | 10 | 8 |
| ATA | 2 | 5 | FA | 16 | 31 | PER | 4 | 6 |
| BIO | 82 | 81 | FE | 1 | 10 | PHIL | 97 | 74 |
| BM | 15 | 18 | FLED | 93 | 61 | PHYS | 100 | 95 |
| CE | 149 | 104 | FR | 43 | 29 | POLS | 232 | 166 |
| CET | 15 | 24 | GER | 52 | 31 | PRED | 11 | 38 |
| CHE | 93 | 88 | GR | 1 | 1 | PSY | 98 | 82 |
| CHEM | 164 | 127 | HIST | 107 | 103 | RUS | 8 | 8 |
| CIN | 0 | 6 | HTR | 44 | 24 | SCED | 33 | 44 |
| CL | 29 | 18 | HUM | 11 | 8 | SCO | 1 | 1 |
| CMPE | 139 | 113 | IE | 129 | 86 | SOC | 68 | 82 |
| COMP | 9 | 42 | INT | 9 | 6 | SP | 3 | 3 |
| DRA | 6 | 4 | INTT | 78 | 94 | SPA | 12 | 9 |
| EC | 318 | 203 | ITA | 2 | 3 | STAT | 3 | 6 |
| ED | 127 | 104 | JP | 7 | 8 | STS | 11 | 12 |
| EE | 114 | 120 | LING | 25 | 18 | TK | 56 | 38 |
| EF | 0 | 13 | LIT | 10 | 12 | TKL | 87 | 88 |
| EL | 62 | 34 | MAN | 20 | 47 | TOUR | 1 | 16 |
| ELT | 15 | 38 | MATH | 226 | 162 | TR | 88 | 66 |
| ENG | 7 | 4 | ME | 123 | 98 | TRM | 65 | 101 |

## Creation of Rules

Creation of rules step is run on each set seperately. In this step the first thing done is creating all possible rules by using a recursive algorithm. For example if the set contains three grades as MIS131 (AA), MIS132 (BB) and MIS233 (BA) all the rules below are created:

- MIS131(AA) MIS132(BB) -> MIS233(BA)
- MIS131(AA)MIS233(BA) -> MIS132(BB)
- MIS132(BB)MIS233(BA) -> MIS131(AA)
- MIS131(AA) -> MIS132(BB) MIS233(BA)
- MIS132(BB) -> MIS131(AA) MIS233(BA)
- MIS233(BA) -> MIS131(AA)MIS132(BB)

After all these rules are created, the calculation of the confidence, lift and importance variables' calculation starts for each rule. The rules are added to data base only if the confidence is higher than predefined value, importance is greater than 0 and lift is greater than 1.

## Calculation of Confidence

In this step an adjustment is made in the confidence formula in order to get the formula better fit for the software. By definition the confidence of rule MIS131 (AA) MIS132 (BB) -> MIS233 (BA) is calculated as:

X= the number of students who got AA from MIS131, BB from MIS132 and BA from MIS233.

Y = the number of students who got AA from MIS131 and BB from MIS132.

Confidence (MIS131 (AA) MIS132 (BB) -> MIS233 (BA)) = X / Y

This confidence gives the probability of getting the grade BA from MIS233 if the student got AA from MIS131 and BB from MIS132. But the actual confidence level this study aims to find is the probability of a student to get BA from MIS233 if the student has got AA from MIS132 and BB from MIS132 and this student takes the course MIS233 no matter what grade he takes. So the confidence formula used in this software is calculated as:

X= the number of students who got AA from MIS131, BB from MIS132 and BA from MIS233.

Y = the number of students who got AA from MIS131 and BB from MIS132 and taken the course MIS233.

Confidence (MIS131 (AA) MIS132 (BB) -> MIS233 (BA)) = X / Y

The function to calculate the confidence is implemented according to this formula.

<u>Calculation of Lift</u>

The next step of this study is to calculate the lift value of the rules. For this step a little adjustment is done on the lift formula too. The lift formula by definition for the rule MIS131 (AA) MIS132 (BB) -> MIS233 (BA) is:

X = Number of students who got AA from MIS131 / number of all students

Y = Number of students who got BB from MIS132 / number of all students

Z = Number of students who got BA from MIS233 / number of all students

K = Number of students who got AA from MIS131, BB from MIS132 and BA from MIS233 / number of all students

Lift = K/(X*Y*Z)

As the X value above does not truly represent the possibility of getting AA from MIS131, the formula is adjusted to fit the aim of the study. So the formula is changed as:

X = Number of students who got AA from MIS131 / number of students who have taken MIS131

Y = Number of students who got BB from MIS132 / number of students who have taken MIS132

Z = Number of students who got BA from MIS233 / number of students who have taken MIS233

K = Number of students who got AA from MIS131, BB from MIS132 and BA from MIS233 / number of students who have taken MIS131, MIS132 and MIS233

Lift = K/(X*Y*Z)

By making this adjustment the lift formula is made more explanatory for this study. Using this formula the function to calculate the lift is implemented.

Calculation of Importance

To calculate the importance value is the next step of this study. As it was in the previous steps, again some adjustments will be made on the importance formula. The definition of the importance suggests that the importance value for the rule MIS131 (AA) MIS132 (BB) -> MIS233 (BA) to be :

X = Number of students who have not taken AA from MIS131 or BB from MIS132.

Y = Number of students who have not taken AA from MIS131 or BB from MIS132 but taken BA from MIS233.

Importance(MIS131 (AA) MIS132 (BB) -> MIS233 (BA)) =

$\text{Log}_{10}$(Confidence(MIS131 (AA) MIS132 (BB) -> MIS233 (BA))/(Y/X))

As these values would not represent the true probabilities, the adjustment is done on this formula too. The adjusted formula is :

X = Number of students who have taken MIS131 and MIS132 courses but did not take AA from MIS131 or BB from MIS132 and taken MIS233 regardless of the grade.

Y = Number of students who have taken MIS131 and MIS132 courses but did not take AA from MIS131 or BB from MIS132 but taken BA from MIS233.

Importance(MIS131 (AA) MIS132 (BB) -> MIS233 (BA)) =

$\text{Log}_{10}$(Confidence(MIS131 (AA) MIS132 (BB) -> MIS233 (BA))/(Y/X))

So as the formula is adjusted the importance value can now be properly calculated.

Up to this point all the variables are calculated using only the data prior to 2008. For the calculation of test confidence step, the data from the years later than 2008 will also be used.

Calculation of Test Confidence

In this step another confidence value is calculated to test the rules. The aim of this test is to see if the rules created by the software are consistently reliable for the new students. So the test confidence variable for the rule MIS131 (AA) MIS132 (BB) -> MIS233 (BA) is calculated as explained below :

X = The number of the students who have taken AA from MIS131 and BB from MIS132 in any time period either before or after 2008 and taken MIS233 after 2008 regardless of the grade taken.

Y = The number of the students who have taken AA from MIS131 and BB from MIS132 in any time period either before or after 2008 and taken MIS233 after 2008 and got BA.

Test Confidence = Y/X

The result of this formula shows the validity of the rules created, using data prior to 2008, in the data after 2008.

So all the rules are created and written into the database with their variables.

User Tool

In this step of the study, an interface is built in order to make users to be able to enter their previous grades and let the software recommend the best courses for them. To achieve this the first thing done is to find the rules with most confidence, which the student's previous data fit for.

All the rules in the database are taken, ordered by their confidence, descending. The rules with greatest confidence are going to be processed first. After that, all the rules are checked one by one to see if all the grades on the left side of the rule are searched on student's previous grades. If all of them are found, the courses and the grades on the right side of the rule are added to the suggestions list for that student. If two rules suggest the same course to a student, only one of them (the one with higher confidence) is added to the suggestion list. In the beginning as the student's previous data is unknown, the most confident 20 rules in the database are suggested. The initial state of the program can be seen in Figure 2.

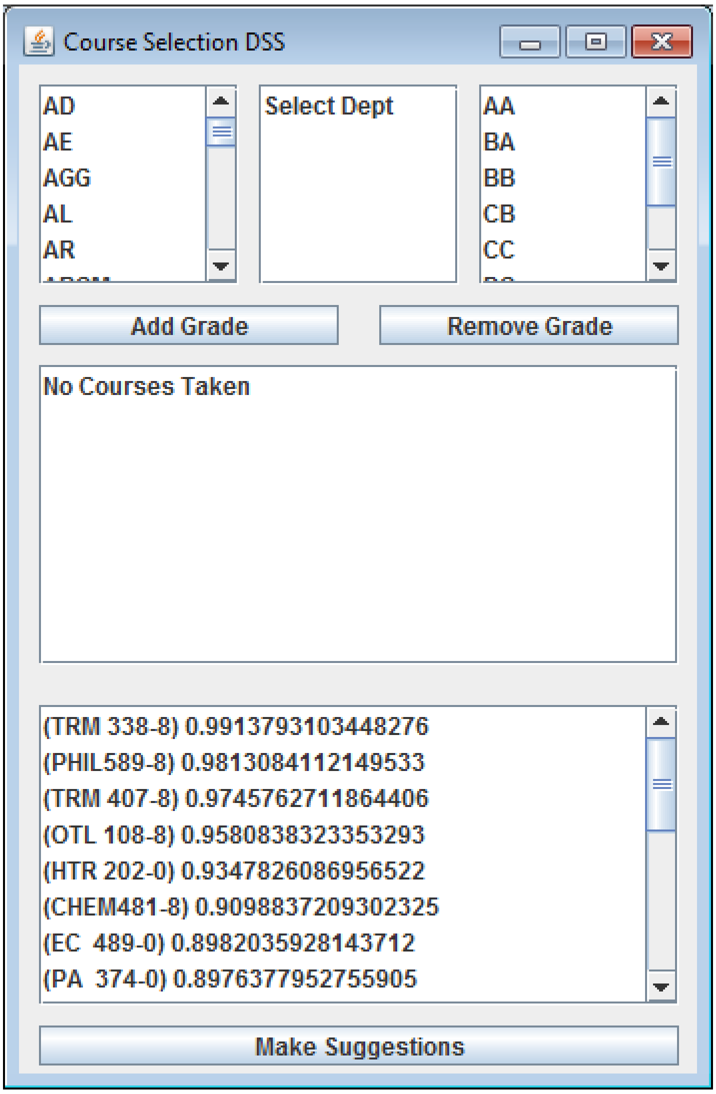


Figure 2 - The screenshot of the program in initial state

After that step the user adds the grade they have taken and then press the make suggestions button. Then the software finds the best suggestions and lists them on the screen. In the example which can be seen in Figure 3, the student added AA for MIS131 course and AA for MIS212. So the software suggests if this student takes MIS481 course, he will get AA from that course with 0.804687 probability. If the student takes MIS211 course then he will get AA with 0.75 probability.

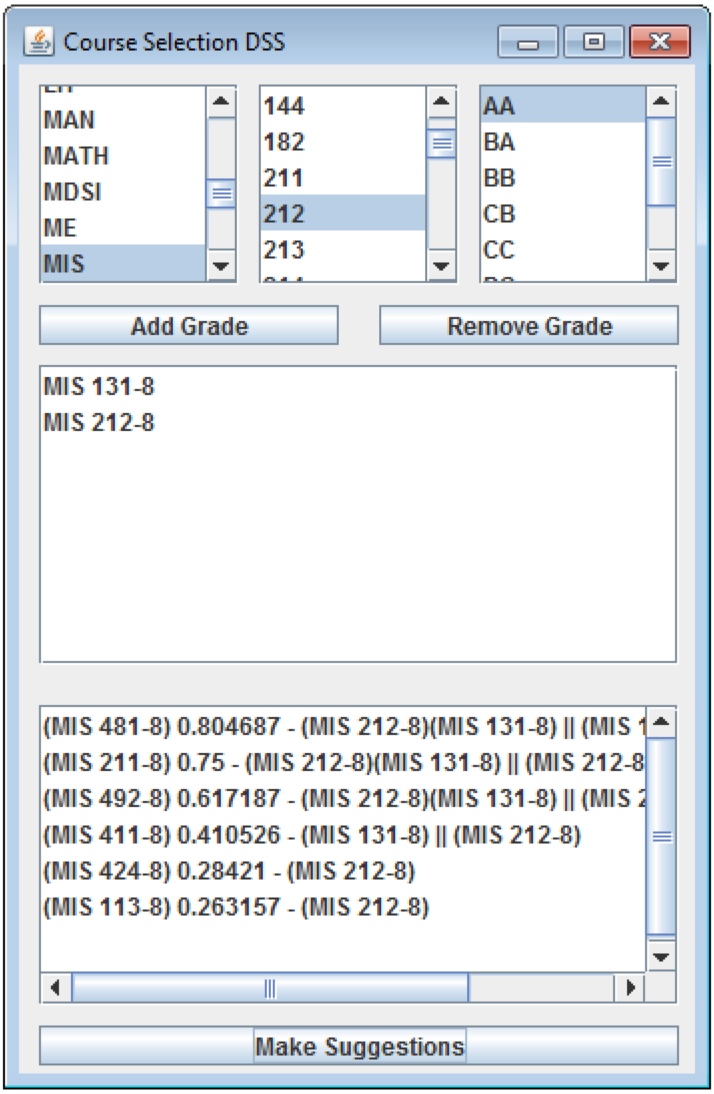


Figure 3 – The Screenshot of the Program After The Previous Grades Are Entered

# CHAPTER 4 : RESULTS

The rules in the database are checked after they are created.  The total accuracy values for the rules are calculated. The results are explained in 2 parts in this study. As the rules are created separately for 2 datasets, the results and rules created will be explained in two parts.

## Results For Dataset With 8 Grades

Using 8 grade sets and creating the rules according to these sets 4,052 rules are created. Before inspecting the rules some accuracy measures should be calculated. The distribution of the grades must be used in this calculation.

Table 14. The Count Of Grades to Calculate The Gini Index

| Grade | Count | Ratio |
|---|---|---|
| AA | 299,701 | 0.187 |
| BA | 236,124 | 0.147 |
| BB | 244,291 | 0.152 |
| CB | 195,177 | 0.121 |
| CC | 183,387 | 0.114 |
| DC | 120,658 | 0.075 |
| DD | 109,402 | 0.068 |
| F | 218,037 | 0.136 |

The distribution can be seen in Table 14. The distribution shows that if the majority rule is applied and all the grades are predicted as AA the accuracy would be 18.7%.

One other accuracy measure is Gini index. When the Gini index is calculated as:

Gini = 0.136064

So it can be seen that the Gini index shows an accuracy around 13.6%.

The accuracy of this study can be calculated in 2 ways. The first accuracy measure is the accuracy on the training set. For this step the minimum confidence value must be defined. The confidence value defined here may not be equal to the minimum confidence value defined at the creation of rules step. If it is set to 18.7% as it was done in the rule creation step, all the rules created, will be used in this step. After that all the rules' confidence values should be summed using their supports as the weight. So the calculations are made with the weights, the accuracy is calculated as around 30.7%. This result is almost 2 times better than the majority rule. If the minimum confidence value is updated to 50% then the accuracy is calculated as 62.6%. This accuracy can be increased by increasing the minimum confidence value, but this will result having fewer number of rules to be used, so some of the information found by the rule creator step will be lost.

When test confidence value are summed using their weights, it can be seen if the accuracy is close to the accuracy measure which was calculated above. When the accuracy on the test data is calculated the accuracy value is 26.47%. As the test accuracy value was expected to be lower than the training accuracy value, this result seems reasonable.

As the last measure which should be calculated is the average error. In this study the definition of error is the difference between confidence and test confidence values. The square of one error is taken and they were weighted by test support. After that the weighted average should be taken and then the square root of the result would show the average error on rules. This measure is calculated as 14.83% in this study.

## Results For Dataset With 4 Grades

As it was mentioned before, in the second dataset the grades are grouped into 4 sets. The groups are HIGH, MID, LOW and FAIL. Using 4 grade sets and creating the rules according to these sets 11403 rules are created. Before inspecting the rules some accuracy measures should be calculated. The distribution of the grades must be used in this calculation.

Table 15. The Count of Grade Groups to Calculate the Gini Index

| Grade | Count | Ratio |
|---|---|---|
| HIGH(AA-BA) | 535,825 | 0.333 |
| MID(BB-CB-CC) | 622,855 | 0.388 |
| LOW(DC-DD) | 230,060 | 0.143 |
| FAIL(F) | 218,037 | 0.136 |

The distribution shows that if the majority rule is applied and all the grades are predicted as MID(BB-CB-CC) the accuracy would be 38.8% (Table 15).

One other accuracy measure is Gini index. The Gini index is calculated as:

$$Gini = 0.333^2 + 0.388^2 + 0.143^2 + 0.136^2 = 0.300378$$

So it can be seen that the Gini index shows an accuracy around 30%.

The accuracy of this study can be calculated in 2 ways. The first accuracy measure is the accuracy on the training set. For this step the minimum confidence value must be defined. The confidence value defined here may not be equal to the minimum confidence value defined at the creation of rules step. If it is set to 38.8% as it was done in the rule creation step, all the rules created, will be used in this step.

After that all the rules' confidence values should be summed using their supports as the weight. So the calculations are made with the weights the accuracy is calculated as around 54.76%. This result is almost 1.5 times better than the majority rule. If the minimum confidence value is updated to 50% then the accuracy is calculated as 59.6%. This accuracy can be increased by increasing the minimum confidence value, but this will result having fewer number of rules to be used, so some of the information found by the rule creator step will be lost.

When test confidence values are summed using their weights, it can be seen if the accuracy is close to the accuracy measure which was calculated above. When the accuracy on the test data is calculated the accuracy value is 48.97%. As the test accuracy value was expected to be lower than the training accuracy value this seems a reasonable result.

As the last measure which should be calculated is the average error. In this study the definition of error is the difference between confidence and test confidence. The square of one error is taken and they were weighted by test support. After that the weighted average should be taken and then the square root of the result would show the average error on rules. This measure is calculated as 15% in this study.

# CHAPTER 5 : CONSTRAINTS

During the implementation of this study, one of the things which made it difficult is that the list of all the prerequisite list was not present in the database. This caused a problem with the rule creator part of the software. The problem is that the software prepares all the rules without considering the prerequisite rules, so a rule which suggests that taking an AA from course X will precede taking an AA from course Y, where Y is the prerequisite of X can be created, and all the variables will be calculated by the software. These unnecessary calculations makes the program's running time even more. As the program's running time is already too long, acquiring such a prerequisites list will shorten this time. On the other hand this problem does not affect the results of the program, because as the student's previous grades are entered to the user tool, even a student enters taking an AA from X, as that student has to enter a grade for Y course too, because of the rules of the university, so Y course will never be suggested to that student. So absence of prerequisite list only affects the speed of the rule creator part of the software.

As it was mentioned before there is the main problem in this study is that the rule creator part works slow because of the huge amount of the data. To overcome that 2 ways are tried. First way was keeping all the information of all the grades in the program, as objects. But the amount of data makes it impossible and when this way is tried a personal computer even with 8GB RAM is not enough to hold and process this kind of data at once. The second way, which is already being used in this study, does not hold the lists in the program but reaches them by running queries on the database each time they are required. This makes the program make the same queries again and again which makes the program to run very slow. When this approach is used, even running one grade with all the others takes days to process. So the program is run on all the departments separately. So all sets include courses from one department each. This causes the software to miss the inter-departmental rules. Which means this software will not be able to create rules such as students who got AA from MIS131 will get AA from TK101. In order to solve this, some super computers should be dedicated to run this software.

One of the main problems about this study is that, the data used is acquired from Bogazici University, where the program is not very flexible. As the program is not flexible, the most of the students' programs contain the same mandatory courses. So suggesting taking or not taking courses will not have an effect because students cannot change the courses they will take.

Many of the background data is missing in the database, for this reason in this study these variables cannot be used. For these variables to be used properly, more information on these variables must be collected.

One of the other problems is the frequent changes in the university entrance exam in Turkey. As the entrance system changes as frequent as once in 2-3 years, the university entrance exam points cannot be used in the study.

# CHAPTER 6 : FUTURE WORK

This study is open to make further improvements. Some of the improvements are basically for solving the limitations mentioned above. One of the first extensions which should be done on the study is that by adding the prerequisite condition to the creation of rules step. Addition of such a control would decrease the number of unnecessary steps, so it will increase the running speed of the software.

Another improvement should be addition of instructor data. As one of the factors which affect the student's success is the teaching style of the instructor, information about the instructors should also be held and used in the algorithm. The student's way of understanding courses should fit the instructor's way of teaching. Of course these cannot be held in the database as variables, but the relations between students and instructors could lead the software to be able to suggest using these similarities. This way the students will be able to get help from the software about selecting the sections of courses. As some courses must be taken and the only selection a student can make is the selection of section and the instructor. So in order to help the students to make this selection, instructor data should also be used in the future studies. As the data was unavailable in the database used for this study this part was not included in the scope of this study.

For some students the thing which affects the grades taken, is the order of the courses taken. In this study, the algorithm used is not capable of handling those sequential information. In order to find the order which the courses should be taken in, a sequential data mining algorithm should be applied on this data.

# CHAPTER 7 : DISCUSSION AND CONCLUSION

In this study a software is prepared which analyzes the data of students and the grades they got from the courses they took, and prepares rules to suggest the students the courses they should take, using association rules mining algorithm.

This software checks all the database for all the grades and creates the rules, another part of the software, which is used by the students or the advisors, checks the given student's previous grades and searches the rules to find which ones fits for this student. The fitting rules are used and the courses are suggested to the student.

When the results are inspected, it can be seen that the selection of the confidence value would affect the accuracy of the rules' suggestions. But even with the lowest confidence value, the accuracy is about 1.5 times better than the majority rule approach and almost twice better than Gini Index.

The main problem about preparing this software, is that too much data was in the database. However it is good for being able to work with more data for data mining purposes, having this much data caused the program to work too slow or even crash. To solve this the software's method is changed to work slower but stable.

# APPENDICES

## Appendix A – Rules For MIS Department

### 8 Grade Dataset

The list of the rules which are created using the data for MIS department, and having the confidence above 0.50 can be seen below. The rules are sorted by the confidence values. The students who got the grades on the left should be offered the courses on the right. The letters stand for:

- S : Support
- C : Confidence
- I : Importance
- TS : Test Support
- TC : Test Confidence
- L : Lift

| Left Side | Right Side | S | C | I | TS | TC | L |
|---|---|---|---|---|---|---|---|
| (MIS131-AA)(MIS211-AA)(MIS481-AA) | (MIS212-AA) | 78 | 0.96 | 0.53 | 3 | 0.33 | 6.28 |
| (MIS492-AA)(MIS411-AA) | (MIS481-AA) | 90 | 0.95 | 0.09 | 0 | -1.00 | 1.37 |
| (MIS131-AA)(MIS211-AA) | (MIS212-AA) | 96 | 0.91 | 0.57 | 4 | 0.50 | 5.46 |
| (MIS411-AA)(MIS212-AA) | (MIS481-AA) | 93 | 0.90 | 0.06 | 0 | -1.00 | 2.10 |
| (MIS463-BA) | (MIS481-AA) | 83 | 0.88 | 0.06 | 10 | 0.90 | 1.01 |
| (MIS131-AA)(MIS492-AA) | (MIS481-AA) | 83 | 0.87 | 0.24 | 5 | 0.80 | 1.54 |
| (MIS424-AA)(MIS492-AA) | (MIS481-AA) | 115 | 0.87 | 0.07 | 8 | 0.88 | 1.36 |
| (MIS212-AA)(MIS492-AA) | (MIS481-AA) | 127 | 0.87 | 0.17 | 12 | 0.92 | 1.27 |
| (MIS492-AA)(MIS211-AA) | (MIS481-AA) | 95 | 0.86 | 0.04 | 21 | 0.90 | 1.28 |
| (MIS211-AA)(MIS492-AA) | (MIS212-AA) | 95 | 0.86 | 0.48 | 0 | -1.00 | 2.56 |
| (MIS492-AA)(MIS212-AA)(MIS211-AA) | (MIS481-AA) | 82 | 0.86 | 0.04 | 10 | 0.90 | 2.80 |
| (MIS492-AA)(MIS481-AA)(MIS211-AA) | (MIS212-AA) | 82 | 0.86 | 0.40 | 0 | -1.00 | 2.80 |
| (MIS424-AA) | (MIS481-AA) | 149 | 0.86 | 0.08 | 13 | 0.85 | 1.02 |
| (MIS211-AA)(MIS481-AA) | (MIS212-AA) | 110 | 0.86 | 0.54 | 9 | 0.33 | 2.05 |
| (MIS131-AA)(MIS492-AA) | (MIS212-AA) | 79 | 0.83 | 0.41 | 0 | -1.00 | 2.97 |
| (MIS324-AA) | (MIS481-AA) | 91 | 0.83 | 0.09 | 0 | -1.00 | 1.10 |
| (MIS481-AA)(MIS131-AA) | (MIS212-AA) | 103 | 0.82 | 0.42 | 8 | 0.13 | 2.64 |
| (MIS211-AA)(MIS212-AA)(MIS131-AA) | (MIS481-AA) | 78 | 0.81 | 0.22 | 5 | 0.80 | 6.28 |
| (MIS212-AA)(MIS131-AA) | (MIS481-AA) | 103 | 0.80 | 0.24 | 6 | 0.67 | 2.64 |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| (MIS463-AA) | (MIS492-AA) | 80 | 0.78 | 0.14 | 16 | 0.69 | 1.18 |
| (MIS481-AA)(MIS424-AA) | (MIS492-AA) | 115 | 0.77 | 0.14 | 10 | 0.70 | 1.36 |
| (MIS211-AA)(MIS131-AA) | (MIS481-AA) | 81 | 0.76 | 0.19 | 13 | 0.77 | 2.57 |
| (MIS424-AA) | (MIS492-AA) | 132 | 0.76 | 0.16 | 12 | 0.67 | 1.23 |
| (MIS213-CC) | (MIS481-AA) | 79 | 0.76 | 0.09 | 18 | 0.89 | 1.19 |
| (MIS131-AA)(MIS481-AA)(MIS212-AA) | (MIS211-AA) | 78 | 0.76 | 0.39 | 1 | 1.00 | 6.28 |
| (MIS211-AA) | (MIS212-AA) | 167 | 0.76 | 0.62 | 16 | 0.44 | 2.17 |
| (MIS212-AA)(MIS131-AA) | (MIS211-AA) | 96 | 0.75 | 0.40 | 5 | 0.40 | 5.46 |
| (MIS211-AA)(MIS212-AA)(MIS481-AA) | (MIS492-AA) | 82 | 0.75 | 0.21 | 11 | 0.82 | 2.80 |
| (MIS211-AA)(MIS481-AA) | (MIS492-AA) | 95 | 0.74 | 0.22 | 26 | 0.73 | 1.28 |
| (MIS411-AA) | (MIS481-AA) | 141 | 0.74 | 0.30 | 0 | -1.00 | 1.26 |
| (MIS481-AA)(MIS212-AA) | (MIS492-AA) | 127 | 0.71 | 0.09 | 16 | 0.69 | 1.27 |
| (MIS212-AA)(MIS211-AA)(MIS481-AA) | (MIS131-AA) | 78 | 0.71 | 0.57 | 1 | 0.00 | 6.28 |
| (MIS131-AA) | (MIS212-AA) | 128 | 0.67 | 0.57 | 16 | 0.31 | 2.29 |
| (MIS481-AA)(MIS131-AA) | (MIS492-AA) | 83 | 0.66 | 0.01 | 7 | 0.57 | 1.54 |
| (MIS113-BB) | (MIS481-AA) | 75 | 0.66 | 0.11 | 19 | 0.63 | 1.11 |
| (MIS211-AA)(MIS212-AA) | (MIS481-AA) | 110 | 0.66 | 0.05 | 16 | 0.94 | 2.05 |
| (MIS131-AA) | (MIS481-AA) | 125 | 0.66 | 0.25 | 22 | 0.77 | 1.24 |
| (MIS212-AA)(MIS492-AA) | (MIS211-AA) | 95 | 0.65 | 0.43 | 0 | -1.00 | 2.56 |
| (MIS131-AA)(MIS481-AA) | (MIS211-AA) | 81 | 0.65 | 0.32 | 7 | 0.43 | 2.57 |
| (MIS481-AA)(MIS492-AA)(MIS212-AA) | (MIS211-AA) | 82 | 0.65 | 0.44 | 0 | -1.00 | 2.80 |
| (MIS113-BA) | (MIS481-AA) | 78 | 0.64 | 0.09 | 13 | 0.85 | 1.18 |
| (MIS411-AA)(MIS481-AA) | (MIS492-AA) | 90 | 0.64 | 0.14 | 0 | -1.00 | 1.37 |
| (MIS211-AA)(MIS481-AA) | (MIS131-AA) | 81 | 0.63 | 0.52 | 5 | 0.20 | 2.57 |
| (MIS212-AA)(MIS131-AA) | (MIS492-AA) | 79 | 0.62 | 0.16 | 3 | 0.67 | 2.97 |
| (MIS212-AA)(MIS481-AA) | (MIS211-AA) | 110 | 0.61 | 0.44 | 2 | 1.00 | 2.05 |
| (MIS212-AA) | (MIS211-AA) | 167 | 0.59 | 0.61 | 10 | 0.60 | 2.17 |
| (MIS212-AA)(MIS481-AA) | (MIS131-AA) | 103 | 0.58 | 0.60 | 1 | 0.00 | 2.64 |
| (MIS211-AA)(MIS212-AA) | (MIS131-AA) | 96 | 0.57 | 0.55 | 5 | 0.20 | 5.46 |
| (MIS211-AA)(MIS212-AA) | (MIS492-AA) | 95 | 0.57 | 0.09 | 12 | 0.83 | 2.56 |
| (MIS131-AA) | (MIS211-AA) | 106 | 0.56 | 0.36 | 29 | 0.31 | 2.07 |
| (MIS411-AA) | (MIS212-AA) | 103 | 0.54 | 0.09 | 0 | -1.00 | 1.83 |
| (MIS212-AA)(MIS492-AA) | (MIS131-AA) | 79 | 0.54 | 0.45 | 0 | -1.00 | 2.97 |
| (MIS481-AA)(MIS212-AA) | (MIS411-AA) | 93 | 0.52 | 0.34 | 0 | -1.00 | 2.10 |
| (MIS212-AA) | (MIS492-AA) | 146 | 0.51 | 0.11 | 17 | 0.71 | 1.16 |
| (MIS492-AA)(MIS481-AA) | (MIS212-AA) | 127 | 0.50 | 0.08 | 2 | 0.00 | 1.27 |
| (MIS492-AA) | (MIS212-AA) | 146 | 0.50 | 0.10 | 2 | 0.00 | 1.16 |
| (MIS131-AA) | (MIS492-AA) | 95 | 0.50 | 0.15 | 11 | 0.45 | 1.38 |
| (MIS411-AA) | (MIS492-AA) | 95 | 0.50 | 0.25 | 0 | -1.00 | 1.15 |
| (MIS211-AA) | (MIS492-AA) | 110 | 0.50 | 0.03 | 33 | 0.67 | 1.17 |

## <u>4 Grade Dataset</u>

| Left Side | Right Side | S | C | I | TS | TC | L |
|---|---|---|---|---|---|---|---|
| (MIS463-H)(MIS492-H) | (MIS481-H) | 155 | 0.95 | 0.04 | 24 | 0.96 | 1.05 |
| (MIS424-H)(MIS492-H) | (MIS481-H) | 240 | 0.95 | 0.09 | 49 | 0.96 | 1.00 |
| (MIS213-M)(MIS492-H) | (MIS481-H) | 166 | 0.94 | 0.16 | 36 | 0.89 | 1.22 |
| (MIS113-H)(MIS492-H) | (MIS481-H) | 160 | 0.94 | 0.16 | 33 | 0.97 | 1.19 |
| (MIS212-H)(MIS492-H) | (MIS481-H) | 200 | 0.93 | 0.18 | 29 | 0.97 | 1.07 |
| (MIS212-H)(MIS424-H) | (MIS481-H) | 159 | 0.93 | 0.16 | 23 | 0.96 | 1.08 |

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| (MIS233-M) | (MIS481-H) | 159 | 0.92 | 0.01 | 10 | 1.00 | 1.45 |
| (MIS224-M)(MIS492-H) | (MIS481-H) | 153 | 0.92 | 0.14 | 37 | 0.86 | 1.01 |
| (MIS211-H)(MIS492-H) | (MIS481-H) | 160 | 0.91 | 0.15 | 39 | 0.92 | 1.14 |
| (MIS111-M)(MIS492-H) | (MIS481-H) | 153 | 0.91 | 0.16 | 40 | 0.83 | 1.06 |
| (MIS113-H)(MIS481-H) | (MIS492-H) | 160 | 0.89 | 0.25 | 32 | 1.00 | 1.19 |
| (MIS424-H)(MIS481-H) | (MIS492-H) | 240 | 0.88 | 0.15 | 52 | 0.90 | 1.00 |
| (MIS211-H)(MIS481-H) | (MIS212-H) | 160 | 0.86 | 0.46 | 10 | 0.60 | 1.70 |
| (MIS211-H)(MIS481-H) | (MIS492-H) | 160 | 0.86 | 0.21 | 38 | 0.95 | 1.14 |
| (MIS111-M)(MIS481-H) | (MIS492-H) | 153 | 0.86 | 0.23 | 38 | 0.87 | 1.06 |
| (MIS463-H)(MIS481-H) | (MIS492-H) | 155 | 0.86 | 0.06 | 23 | 1.00 | 1.05 |
| (MIS212-H)(MIS481-H) | (MIS492-H) | 200 | 0.85 | 0.24 | 29 | 0.97 | 1.07 |
| (MIS124-M) | (MIS481-H) | 157 | 0.84 | 0.06 | 0 | -1.00 | 1.19 |
| (MIS213-M)(MIS481-H) | (MIS492-H) | 166 | 0.81 | 0.16 | 35 | 0.91 | 1.22 |
| (MIS224-M)(MIS481-H) | (MIS492-H) | 153 | 0.81 | 0.15 | 35 | 0.91 | 1.01 |
| (MIS211-H) | (MIS212-H) | 220 | 0.79 | 0.56 | 20 | 0.70 | 1.81 |
| (MIS213-M) | (MIS481-H) | 204 | 0.78 | 0.07 | 57 | 0.93 | 1.11 |
| (MIS144-M) | (MIS481-H) | 153 | 0.73 | 0.11 | 49 | 0.92 | 1.27 |
| (MIS411-H) | (MIS481-H) | 172 | 0.73 | 0.19 | 0 | -1.00 | 1.12 |
| (MIS211-H)(MIS212-H) | (MIS481-H) | 160 | 0.73 | 0.05 | 29 | 0.93 | 1.70 |
| (MIS481-H)(MIS492-H) | (MIS424-H) | 240 | 0.72 | 0.25 | 63 | 0.68 | 1.00 |
| (MIS131-H) | (MIS212-H) | 174 | 0.71 | 0.55 | 32 | 0.31 | 1.85 |
| (MIS131-H) | (MIS481-H) | 171 | 0.70 | 0.23 | 47 | 0.94 | 1.13 |
| (MIS212-H)(MIS481-H) | (MIS211-H) | 160 | 0.68 | 0.38 | 6 | 0.67 | 1.70 |
| (MIS212-H)(MIS481-H) | (MIS424-H) | 159 | 0.68 | 0.21 | 25 | 0.72 | 1.08 |
| (MIS336-M) | (MIS321-M) | 156 | 0.68 | 0.14 | 46 | 0.78 | 1.02 |
| (MIS213-M) | (MIS492-H) | 176 | 0.67 | 0.04 | 40 | 0.93 | 1.14 |
| (MIS313-M) | (MIS321-M) | 154 | 0.67 | 0.13 | 48 | 0.79 | 1.11 |
| (MIS143-M) | (MIS481-H) | 155 | 0.65 | 0.05 | 55 | 0.85 | 1.06 |
| (MIS131-H) | (MIS211-H) | 157 | 0.64 | 0.38 | 49 | 0.35 | 1.79 |
| (MIS111-M) | (MIS492-H) | 169 | 0.64 | 0.07 | 46 | 0.87 | 1.03 |
| (MIS212-H) | (MIS211-H) | 220 | 0.63 | 0.53 | 20 | 0.60 | 1.81 |
| (MIS211-H) | (MIS492-H) | 175 | 0.63 | 0.05 | 42 | 0.95 | 1.10 |
| (MIS212-H) | (MIS492-H) | 214 | 0.62 | 0.09 | 31 | 0.97 | 1.01 |
| (MIS113-H) | (MIS492-H) | 170 | 0.61 | 0.10 | 34 | 1.00 | 1.11 |
| (MIS213-M) | (MIS321-M) | 157 | 0.60 | 0.14 | 52 | 0.75 | 1.23 |
| (MIS481-H)(MIS492-H) | (MIS212-H) | 200 | 0.60 | 0.12 | 8 | 0.13 | 1.07 |
| (MIS492-H) | (MIS212-H) | 214 | 0.59 | 0.12 | 10 | 0.10 | 1.01 |
| (MIS424-H)(MIS481-H) | (MIS212-H) | 159 | 0.58 | 0.04 | 6 | 0.17 | 1.08 |
| (MIS424-H) | (MIS212-H) | 171 | 0.57 | 0.03 | 7 | 0.14 | 1.02 |
| (MIS211-H) | (MIS131-H) | 157 | 0.56 | 0.48 | 34 | 0.47 | 1.79 |
| (MIS211-H) | (MIS113-H) | 152 | 0.55 | 0.36 | 31 | 0.61 | 1.57 |
| (MIS113-H) | (MIS211-H) | 152 | 0.54 | 0.29 | 50 | 0.48 | 1.57 |
| (MIS113-H) | (MIS212-H) | 151 | 0.54 | 0.18 | 27 | 0.41 | 1.36 |
| (MIS321-M) | (MIS213-M) | 157 | 0.53 | 0.14 | 48 | 0.48 | 1.23 |
| (MIS321-M) | (MIS336-M) | 156 | 0.53 | 0.19 | 69 | 0.54 | 1.02 |
| (MIS321-M) | (MIS313-M) | 154 | 0.52 | 0.18 | 69 | 0.58 | 1.11 |
| (MIS212-H) | (MIS131-H) | 174 | 0.50 | 0.56 | 17 | 0.47 | 1.85 |
| (MIS481-H) | (MIS213-M) | 204 | 0.50 | 0.14 | 37 | 0.57 | 1.11 |

Appendix B – Top 45 Rules For AD Department

<u>4 Grade Dataset</u>

| Left Side | Right Side | S | C | I | TS | TC | L |
|---|---|---|---|---|---|---|---|
| (AD104-H)(AD353-H) | (AD341-H) | 184 | 0.88 | 0.31 | 59 | 0.80 | 3.34 |
| (AD104-H)(AD320-H) | (AD341-H) | 279 | 0.87 | 0.32 | 60 | 0.65 | 2.83 |
| (AD214-H)(AD320-H) | (AD104-H) | 191 | 0.85 | 0.26 | 0 | -1.00 | 2.46 |
| (AD104-H)(AD351-H) | (AD341-H) | 263 | 0.85 | 0.31 | 21 | 0.95 | 3.53 |
| (AD104-H)(AD311-H) | (AD341-H) | 272 | 0.84 | 0.31 | 74 | 0.69 | 2.71 |
| (AD220-H)(AD408-H) | (AD104-H) | 162 | 0.84 | 0.23 | 1 | 1.00 | 2.98 |
| (AD213-H)(AD320-H) | (AD104-H) | 161 | 0.83 | 0.24 | 0 | -1.00 | 2.40 |
| (AD104-H)(AD452-H) | (AD401-H) | 169 | 0.82 | 0.45 | 80 | 0.69 | 2.27 |
| (AD104-H)(AD408-H) | (AD341-H) | 230 | 0.82 | 0.26 | 28 | 0.71 | 2.55 |
| (AD353-H) | (AD341-H) | 279 | 0.82 | 0.26 | 66 | 0.77 | 2.21 |
| (AD220-H)(AD320-H) | (AD104-H) | 181 | 0.82 | 0.23 | 1 | 1.00 | 3.01 |
| (AD202-H)(AD214-H) | (AD104-H) | 245 | 0.81 | 0.26 | 18 | 0.28 | 2.58 |
| (AD401-H)(AD452-H) | (AD104-H) | 169 | 0.81 | 0.23 | 1 | 1.00 | 2.27 |
| (AD131-H)(AD214-H) | (AD104-H) | 198 | 0.81 | 0.21 | 18 | 0.39 | 2.85 |
| (AD214-H)(AD231-H) | (AD104-H) | 264 | 0.80 | 0.25 | 22 | 0.41 | 2.20 |
| (AD104-M)(AD311-L) | (AD353-L) | 150 | 0.80 | 0.37 | 17 | 0.71 | 3.72 |
| (AD104-M)(AD351-L) | (AD353-L) | 244 | 0.80 | 0.41 | 34 | 0.79 | 4.13 |
| (AD213-H)(AD401-H) | (AD104-H) | 211 | 0.80 | 0.23 | 0 | -1.00 | 2.44 |
| (AD104-H)(AD403-H) | (AD401-H) | 167 | 0.80 | 0.43 | 53 | 0.64 | 2.55 |
| (AD202-H)(AD220-H) | (AD104-H) | 227 | 0.80 | 0.24 | 18 | 0.39 | 3.06 |
| (AD320-H) | (AD341-H) | 422 | 0.79 | 0.27 | 65 | 0.62 | 1.86 |
| (AD104-H)(AD403-H) | (AD341-H) | 165 | 0.79 | 0.33 | 67 | 0.64 | 2.07 |
| (AD104-H)(AD202-H) | (AD341-H) | 307 | 0.79 | 0.30 | 72 | 0.82 | 2.59 |
| (AD104-H)(AD312-H) | (AD341-H) | 284 | 0.79 | 0.28 | 71 | 0.66 | 2.39 |
| (AD202-H)(AD213-H) | (AD104-H) | 208 | 0.79 | 0.23 | 17 | 0.18 | 2.52 |
| (AD213-H)(AD442-H) | (AD104-H) | 178 | 0.79 | 0.26 | 1 | 1.00 | 1.88 |
| (AD351-H) | (AD341-H) | 413 | 0.79 | 0.28 | 27 | 0.89 | 2.40 |
| (AD150-H)(AD320-H) | (AD104-H) | 172 | 0.79 | 0.19 | 0 | -1.00 | 1.68 |
| (AD214-H)(AD408-H) | (AD104-H) | 167 | 0.78 | 0.21 | 1 | 1.00 | 2.34 |
| (AD214-H)(AD220-H) | (AD104-H) | 235 | 0.78 | 0.23 | 22 | 0.41 | 2.74 |
| (AD213-H)(AD231-H) | (AD104-H) | 224 | 0.78 | 0.22 | 43 | 0.40 | 2.06 |
| (AD213-H)(AD220-H) | (AD104-H) | 233 | 0.78 | 0.22 | 34 | 0.44 | 2.75 |
| (AD214-H)(AD401-H) | (AD104-H) | 232 | 0.78 | 0.22 | 1 | 1.00 | 2.32 |
| (AD150-H)(AD408-H) | (AD104-H) | 155 | 0.78 | 0.19 | 0 | -1.00 | 1.66 |
| (AD220-H)(AD231-H) | (AD104-H) | 245 | 0.78 | 0.23 | 50 | 0.44 | 2.50 |
| (AD131-H)(AD213-H) | (AD104-H) | 171 | 0.78 | 0.18 | 24 | 0.33 | 2.64 |
| (AD341-H)(AD452-H) | (AD104-H) | 156 | 0.78 | 0.23 | 2 | 1.00 | 1.78 |
| (AD150-H)(AD202-H) | (AD104-H) | 218 | 0.78 | 0.19 | 11 | 0.36 | 1.72 |
| (AD220-H)(AD401-H) | (AD104-H) | 225 | 0.77 | 0.21 | 1 | 1.00 | 2.90 |
| (AD131-M)(AD202-H) | (AD341-H) | 170 | 0.77 | 0.23 | 28 | 0.68 | 1.83 |
| (AD214-H)(AD232-H) | (AD104-H) | 248 | 0.77 | 0.22 | 23 | 0.30 | 2.05 |
| (AD202-H)(AD401-H) | (AD104-H) | 233 | 0.77 | 0.26 | 1 | 1.00 | 2.45 |
| (AD350-L) | (AD353-L) | 175 | 0.77 | 0.29 | 1 | 1.00 | 2.46 |
| (AD131-H)(AD202-H) | (AD104-H) | 211 | 0.77 | 0.22 | 11 | 0.36 | 3.28 |
| (AD202-H)(AD408-H) | (AD104-H) | 172 | 0.77 | 0.24 | 1 | 1.00 | 2.54 |

8 Grade Dataset

| Left Side | Right Side | S | C | I | TS | TC | L |
|---|---|---|---|---|---|---|---|
| (AD353-AA)(AD231-AA) | (AD341-AA) | 76 | 0.87 | 0.52 | 26 | 0.85 | 8.94 |
| (AD311-AA)(AD202-AA) | (AD341-AA) | 85 | 0.86 | 0.51 | 18 | 0.83 | 10.55 |
| (AD311-AA)(AD353-AA) | (AD341-AA) | 80 | 0.83 | 0.50 | 13 | 0.92 | 17.04 |
| (AD231-AA)(AD311-AA) | (AD341-AA) | 91 | 0.80 | 0.51 | 30 | 0.77 | 6.03 |
| (AD232-AA)(AD353-AA) | (AD341-AA) | 76 | 0.78 | 0.50 | 29 | 0.83 | 10.20 |
| (AD353-AA)(AD351-AA) | (AD341-AA) | 76 | 0.78 | 0.48 | 9 | 1.00 | 19.94 |
| (AD401-AA)(AD311-AA) | (AD341-AA) | 86 | 0.78 | 0.47 | 13 | 0.69 | 7.92 |
| (AD311-AA)(AD232-AA) | (AD341-AA) | 93 | 0.78 | 0.47 | 36 | 0.64 | 7.06 |
| (AD311-AA)(AD351-AA) | (AD341-AA) | 95 | 0.77 | 0.47 | 10 | 0.80 | 14.10 |
| (AD231-AA)(AD320-AA) | (AD341-AA) | 84 | 0.77 | 0.49 | 21 | 0.81 | 7.14 |
| (AD220-AA)(AD311-AA) | (AD341-AA) | 75 | 0.77 | 0.49 | 26 | 0.69 | 10.61 |
| (AD231-AA)(AD351-AA) | (AD341-AA) | 101 | 0.75 | 0.49 | 11 | 0.91 | 8.28 |
| (AD411-AA) | (AD412-AA) | 75 | 0.75 | 0.59 | 0 | -1.00 | 2.77 |
| (AD311-AA)(AD214-AA) | (AD341-AA) | 82 | 0.75 | 0.42 | 28 | 0.79 | 9.58 |
| (AD351-AA)(AD202-AA) | (AD341-AA) | 80 | 0.74 | 0.44 | 10 | 0.90 | 12.23 |
| (AD401-AA)(AD351-AA) | (AD341-AA) | 87 | 0.74 | 0.45 | 8 | 0.88 | 9.66 |
| (AD220-AA)(AD488-AA) | (AD341-AA) | 78 | 0.74 | 0.49 | 28 | 0.61 | 6.95 |
| (AD202-AA)(AD488-AA) | (AD341-AA) | 75 | 0.73 | 0.49 | 18 | 0.72 | 7.04 |
| (AD220-AA)(AD401-AA) | (AD341-AA) | 80 | 0.73 | 0.44 | 16 | 0.69 | 9.65 |
| (AD232-AA)(AD351-AA) | (AD341-AA) | 94 | 0.72 | 0.46 | 13 | 0.77 | 8.79 |
| (AD401-AA)(AD202-AA) | (AD341-AA) | 85 | 0.71 | 0.42 | 14 | 0.86 | 8.07 |
| (AD442-AA)(AD311-AA) | (AD341-AA) | 90 | 0.69 | 0.48 | 9 | 0.78 | 6.56 |
| (AD231-AA)(AD401-AA) | (AD341-AA) | 103 | 0.69 | 0.45 | 23 | 0.61 | 5.24 |
| (AD104-AA)(AD202-AA) | (AD231-AA) | 84 | 0.69 | 0.60 | 27 | 0.93 | 4.61 |
| (AD232-AA)(AD401-AA) | (AD341-AA) | 94 | 0.68 | 0.43 | 28 | 0.54 | 5.45 |
| (AD232-AA)(AD131-AA) | (AD231-AA) | 96 | 0.68 | 0.57 | 32 | 0.97 | 7.57 |
| (AD507-AA) | (AD521-AA) | 81 | 0.68 | 0.17 | 15 | 0.47 | 1.20 |
| (AD312-AA)(AD351-AA) | (AD341-AA) | 76 | 0.68 | 0.41 | 5 | 1.00 | 10.31 |
| (AD353-AA) | (AD341-AA) | 128 | 0.68 | 0.45 | 29 | 0.83 | 3.59 |
| (AD214-AA)(AD401-AA) | (AD341-AA) | 79 | 0.68 | 0.40 | 14 | 0.86 | 7.22 |
| (AD214-AA)(AD220-AA) | (AD341-AA) | 80 | 0.67 | 0.40 | 42 | 0.71 | 9.27 |
| (AD202-AA)(AD220-AA) | (AD341-AA) | 80 | 0.67 | 0.42 | 36 | 0.72 | 11.35 |
| (AD488-AA)(AD401-AA) | (AD341-AA) | 86 | 0.67 | 0.40 | 10 | 0.70 | 6.43 |
| (AD351-F)(AD313-F) | (AD353-DD) | 79 | 0.67 | 0.42 | 2 | 1.00 | 4.89 |
| (AD401-AA)(AD351-AA) | (AD231-AA) | 79 | 0.67 | 0.55 | 3 | 1.00 | 7.70 |
| (AD351-DD)(AD313-F) | (AD353-DD) | 98 | 0.67 | 0.44 | 1 | 1.00 | 9.20 |
| (AD104-AA)(AD401-AA) | (AD341-AA) | 90 | 0.67 | 0.42 | 24 | 0.63 | 4.80 |
| (AD231-AA)(AD202-AA) | (AD341-AA) | 96 | 0.67 | 0.43 | 43 | 0.67 | 6.51 |
| (AD442-AA)(AD351-AA) | (AD341-AA) | 97 | 0.66 | 0.45 | 5 | 0.80 | 9.15 |
| (AD104-AA)(AD442-AA) | (AD341-AA) | 82 | 0.66 | 0.45 | 19 | 0.63 | 3.50 |
| (AD202-AA)(AD220-AA) | (AD231-AA) | 78 | 0.66 | 0.54 | 31 | 1.00 | 9.61 |
| (AD312-AA)(AD401-AA) | (AD341-AA) | 78 | 0.66 | 0.39 | 11 | 0.73 | 6.46 |
| (AD220-AA)(AD232-AA) | (AD341-AA) | 75 | 0.64 | 0.39 | 66 | 0.62 | 7.02 |
| (AD231-AA)(AD104-AA) | (AD341-AA) | 103 | 0.64 | 0.40 | 109 | 0.45 | 3.80 |
| (AD341-AA)(AD213-AA) | (AD214-AA) | 78 | 0.64 | 0.68 | 25 | 0.84 | 8.65 |

## Appendix C – Top 45 Rules Having Highest Confidence

### 4 Grade Dataset

| Left Side | Right Side | S | C | I | TS | TC | L |
|---|---|---|---|---|---|---|---|
| (FLED311-H)(FLED403-H)(FLED416-H) | (FLED412-H) | 156 | 0.98 | 0.30 | 28 | 0.96 | 2.30 |
| (FLED306-H)(FLED416-H) | (FLED412-H) | 151 | 0.97 | 0.29 | 26 | 0.96 | 1.65 |
| (FLED304-H)(FLED403-H)(FLED416-H) | (FLED412-H) | 169 | 0.96 | 0.28 | 23 | 0.96 | 2.45 |
| (FLED401-H)(FLED416-H) | (FLED412-H) | 161 | 0.96 | 0.22 | 34 | 0.94 | 1.62 |
| (FLED415-H)(FLED416-H) | (FLED412-H) | 159 | 0.96 | 0.23 | 36 | 0.97 | 1.65 |
| (FLED303-H)(FLED403-H)(FLED416-H) | (FLED412-H) | 177 | 0.96 | 0.28 | 22 | 1.00 | 2.28 |
| (FLED312-H)(FLED416-H) | (FLED412-H) | 169 | 0.95 | 0.34 | 51 | 0.92 | 1.38 |
| (FLED403-H)(FLED411-H)(FLED416-H) | (FLED412-H) | 199 | 0.95 | 0.24 | 26 | 0.96 | 2.18 |
| (FLED303-H)(FLED403-H)(FLED412-H) | (FLED416-H) | 177 | 0.95 | 0.29 | 28 | 0.79 | 2.28 |
| (FLED308-H)(FLED403-H)(FLED412-H) | (FLED416-H) | 175 | 0.95 | 0.35 | 16 | 1.00 | 2.07 |
| (MIS463-H)(MIS492-H) | (MIS481-H) | 155 | 0.95 | 0.04 | 24 | 0.96 | 1.05 |
| (FLED304-H)(FLED411-H)(FLED416-H) | (FLED412-H) | 153 | 0.95 | 0.26 | 22 | 0.95 | 2.52 |
| (FLED304-H)(FLED403-H)(FLED412-H) | (FLED416-H) | 169 | 0.95 | 0.28 | 25 | 0.88 | 2.45 |
| (MIS424-H)(MIS492-H) | (MIS481-H) | 240 | 0.95 | 0.09 | 49 | 0.96 | 1.00 |
| (FLED412-H)(FLED415-H) | (FLED416-H) | 159 | 0.95 | 0.22 | 38 | 0.92 | 1.65 |
| (FLED201-H)(FLED416-H) | (FLED412-H) | 159 | 0.95 | 0.30 | 45 | 0.96 | 1.55 |
| (FLED103-M)(FLED403-H)(FLED416-H) | (FLED412-H) | 169 | 0.94 | 0.32 | 24 | 0.96 | 1.75 |
| (FLED304-H)(FLED416-H) | (FLED412-H) | 202 | 0.94 | 0.30 | 32 | 0.94 | 1.68 |
| (FLED104-M)(FLED403-H)(FLED416-H) | (FLED412-H) | 151 | 0.94 | 0.29 | 22 | 0.95 | 1.59 |
| (MIS213-M)(MIS492-H) | (MIS481-H) | 166 | 0.94 | 0.16 | 36 | 0.89 | 1.22 |
| (FLED403-H)(FLED416-H) | (FLED412-H) | 276 | 0.94 | 0.33 | 42 | 0.93 | 1.53 |
| (MIS113-H)(MIS492-H) | (MIS481-H) | 160 | 0.94 | 0.16 | 33 | 0.97 | 1.19 |
| (FLED311-H)(FLED416-H) | (FLED412-H) | 174 | 0.94 | 0.30 | 45 | 0.96 | 1.47 |
| (FLED403-H)(FLED414-M)(FLED416-H) | (FLED412-H) | 158 | 0.94 | 0.21 | 18 | 0.94 | 1.67 |
| (FLED311-H)(FLED403-H)(FLED412-H) | (FLED416-H) | 156 | 0.94 | 0.28 | 31 | 0.87 | 2.30 |
| (MIS212-H)(MIS492-H) | (MIS481-H) | 200 | 0.93 | 0.18 | 29 | 0.97 | 1.07 |
| (FLED212-H)(FLED416-H) | (FLED412-H) | 151 | 0.93 | 0.35 | 41 | 0.95 | 1.51 |
| (FLED306-H)(FLED412-H) | (FLED416-H) | 151 | 0.93 | 0.27 | 27 | 0.93 | 1.65 |
| (FLED308-H)(FLED403-H)(FLED416-H) | (FLED412-H) | 175 | 0.93 | 0.33 | 17 | 0.94 | 2.07 |
| (FLED403-H)(FLED411-H)(FLED412-H) | (FLED416-H) | 199 | 0.93 | 0.24 | 30 | 0.83 | 2.18 |
| (MIS212-H)(MIS424-H) | (MIS481-H) | 159 | 0.93 | 0.16 | 23 | 0.96 | 1.08 |
| (MIS233-M) | (MIS481-H) | 159 | 0.92 | 0.01 | 10 | 1.00 | 1.45 |
| (FLED201-H)(FLED412-H) | (FLED416-H) | 159 | 0.92 | 0.36 | 49 | 0.88 | 1.55 |
| (FLED103-M)(FLED403-H)(FLED412-H) | (FLED416-H) | 169 | 0.92 | 0.31 | 27 | 0.85 | 1.75 |
| (FLED403-H)(FLED412-H) | (FLED416-H) | 276 | 0.92 | 0.33 | 46 | 0.85 | 1.53 |
| (FLED311-H)(FLED403-H) | (FLED412-H) | 166 | 0.92 | 0.28 | 33 | 0.94 | 1.64 |
| (FLED212-H)(FLED412-H) | (FLED416-H) | 151 | 0.92 | 0.35 | 42 | 0.93 | 1.51 |
| (FLED104-M)(FLED403-H)(FLED412-H) | (FLED416-H) | 151 | 0.92 | 0.29 | 24 | 0.88 | 1.59 |
| (FLED303-H)(FLED416-H) | (FLED412-H) | 209 | 0.92 | 0.29 | 34 | 1.00 | 1.54 |
| (FLED311-H)(FLED412-H) | (FLED416-H) | 174 | 0.92 | 0.29 | 48 | 0.90 | 1.47 |
| (FLED401-H)(FLED403-H) | (FLED412-H) | 150 | 0.92 | 0.19 | 28 | 0.93 | 1.77 |
| (FLED312-H)(FLED403-H) | (FLED412-H) | 155 | 0.92 | 0.30 | 36 | 0.86 | 1.48 |
| (MIS224-M)(MIS492-H) | (MIS481-H) | 153 | 0.92 | 0.14 | 37 | 0.86 | 1.01 |
| (MIS211-H)(MIS492-H) | (MIS481-H) | 160 | 0.91 | 0.15 | 39 | 0.92 | 1.14 |
| (FLED403-H)(FLED412-H)(FLED414-M) | (FLED416-H) | 158 | 0.91 | 0.20 | 20 | 0.85 | 1.67 |

8 Grade Dataset

| LeftSide | RightSide | S | C | I | TS | TC | L |
|---|---|---|---|---|---|---|---|
| (OTL107-AA) | (OTL108-AA) | 109 | 1.00 | 0.06 | 0 | -1.00 | 1.07 |
| (OTL135-AA)(OTL107-AA) | (OTL108-AA) | 84 | 1.00 | 0.02 | 0 | -1.00 | 1.47 |
| (CET462-AA)(CET451-AA) | (CET150-AA) | 82 | 0.99 | 0.08 | 0 | -1.00 | 1.27 |
| (OTL102-AA) | (OTL108-AA) | 89 | 0.98 | 0.18 | 0 | -1.00 | 1.27 |
| (OTL135-AA) | (OTL108-AA) | 116 | 0.97 | 0.26 | 0 | -1.00 | 1.31 |
| (ELT126-AA)(ELT119-AA) | (ELT125-AA) | 77 | 0.97 | 0.28 | 0 | -1.00 | 2.74 |
| (MIS131-AA)(MIS211-AA)(MIS481-AA) | (MIS212-AA) | 78 | 0.96 | 0.53 | 3 | 0.33 | 6.28 |
| (OTL258-AA)(OTL135-AA) | (OTL108-AA) | 76 | 0.96 | 0.14 | 0 | -1.00 | 1.45 |
| (MIS492-AA)(MIS411-AA) | (MIS481-AA) | 90 | 0.95 | 0.09 | 0 | -1.00 | 1.37 |
| (CHE433-BB) | (CHE492-AA) | 103 | 0.94 | 0.04 | 12 | 1.00 | 1.03 |
| (CHE334-AA) | (CHE492-AA) | 85 | 0.93 | 0.08 | 7 | 1.00 | 1.09 |
| (CHE302-AA) | (CHE492-AA) | 107 | 0.92 | 0.07 | 17 | 0.94 | 1.06 |
| (CE343-CB) | (CE492-AA) | 92 | 0.92 | 0.04 | 0 | -1.00 | 1.56 |
| (CHE462-BA) | (CHE492-AA) | 87 | 0.92 | 0.03 | 6 | 1.00 | 1.01 |
| (ELT126-AA)(ELT201-AA) | (ELT125-AA) | 87 | 0.92 | 0.23 | 0 | -1.00 | 2.10 |
| (CHE353-AA) | (CHE492-AA) | 78 | 0.91 | 0.07 | 7 | 1.00 | 1.05 |
| (MIS131-AA)(MIS211-AA) | (MIS212-AA) | 96 | 0.91 | 0.57 | 4 | 0.50 | 5.46 |
| (MIS411-AA)(MIS212-AA) | (MIS481-AA) | 93 | 0.90 | 0.06 | 0 | -1.00 | 2.10 |
| (ELT126-AA) | (ELT125-AA) | 102 | 0.90 | 0.29 | 0 | -1.00 | 1.55 |
| (CHE353-BB) | (CHE492-AA) | 138 | 0.90 | 0.07 | 13 | 0.92 | 1.14 |
| (CET150-AA)(CET462-AA) | (CET451-AA) | 82 | 0.90 | 0.41 | 15 | 1.00 | 1.27 |
| (CHE202-AA) | (CHE492-AA) | 83 | 0.89 | 0.11 | 3 | 1.00 | 1.21 |
| (CET462-AA) | (CET451-AA) | 83 | 0.89 | 0.14 | 17 | 0.94 | 1.09 |
| (MIS463-BA) | (MIS481-AA) | 83 | 0.88 | 0.06 | 10 | 0.90 | 1.01 |
| (MIS131-AA)(MIS492-AA) | (MIS481-AA) | 83 | 0.87 | 0.24 | 5 | 0.80 | 1.54 |
| (AD353-AA)(AD231-AA) | (AD341-AA) | 76 | 0.87 | 0.52 | 26 | 0.85 | 8.94 |
| (MIS424-AA)(MIS492-AA) | (MIS481-AA) | 115 | 0.87 | 0.07 | 8 | 0.88 | 1.36 |
| (ELT201-AA)(ELT119-AA) | (ELT125-AA) | 94 | 0.87 | 0.20 | 0 | -1.00 | 2.35 |
| (MIS212-AA)(MIS492-AA) | (MIS481-AA) | 127 | 0.87 | 0.17 | 12 | 0.92 | 1.27 |
| (CHE202-BA) | (CHE492-AA) | 104 | 0.87 | 0.10 | 6 | 0.83 | 1.15 |
| (GER202-AA)(GER301-AA)(GER101-AA) | (GER201-AA) | 122 | 0.87 | 0.53 | 23 | 0.78 | 9.75 |
| (MIS211-AA)(MIS492-AA) | (MIS212-AA) | 95 | 0.86 | 0.48 | 0 | -1.00 | 2.56 |
| (MIS492-AA)(MIS211-AA) | (MIS481-AA) | 95 | 0.86 | 0.04 | 21 | 0.90 | 1.28 |
| (ELT216-AA) | (ELT201-AA) | 82 | 0.86 | 0.12 | 0 | -1.00 | 1.27 |
| (MIS492-AA)(MIS481-AA)(MIS211-AA) | (MIS212-AA) | 82 | 0.86 | 0.40 | 0 | -1.00 | 2.80 |
| (MIS492-AA)(MIS212-AA)(MIS211-AA) | (MIS481-AA) | 82 | 0.86 | 0.04 | 10 | 0.90 | 2.80 |
| (GER202-AA)(GER411-AA) | (GER412-AA) | 82 | 0.86 | 0.53 | 14 | 1.00 | 2.21 |
| (CHE334-BB) | (CHE492-AA) | 118 | 0.86 | 0.04 | 11 | 1.00 | 1.07 |
| (MIS424-AA) | (MIS481-AA) | 149 | 0.86 | 0.08 | 13 | 0.85 | 1.02 |
| (MIS211-AA)(MIS481-AA) | (MIS212-AA) | 110 | 0.86 | 0.54 | 9 | 0.33 | 2.05 |
| (AD311-AA)(AD202-AA) | (AD341-AA) | 85 | 0.86 | 0.51 | 18 | 0.83 | 10.55 |
| (CE242-CB) | (CE492-AA) | 90 | 0.86 | 0.01 | 0 | -1.00 | 1.55 |
| (JP201-AA)(JP102-AA) | (JP101-AA) | 78 | 0.86 | 0.21 | 7 | 0.86 | 3.16 |
| (ELT125-AA)(ELT119-AA) | (ELT201-AA) | 94 | 0.85 | 0.23 | 0 | -1.00 | 2.35 |
| (ELT126-AA)(ELT125-AA) | (ELT201-AA) | 87 | 0.85 | 0.18 | 0 | -1.00 | 2.10 |

Appendix D – Top 45 Rules Having Highest Support Count

4 Grade Dataset

| LeftSide | RightSide | S | C | I | TS | TC | L |
|---|---|---|---|---|---|---|---|
| (TK222-H) | (TK221-H) | 8443 | 0.66 | 0.32 | 1872 | 0.81 | 1.17 |
| (TK221-H) | (TK222-H) | 8443 | 0.76 | 0.24 | 1836 | 0.81 | 1.17 |
| (EC101-M) | (EC102-M) | 4167 | 0.51 | 0.12 | 1063 | 0.55 | 1.40 |
| (EC102-M) | (EC101-M) | 4167 | 0.53 | 0.12 | 865 | 0.65 | 1.40 |
| (PE102-H) | (PE101-H) | 3331 | 0.81 | 0.21 | 0 | -1.00 | 1.15 |
| (PE101-H) | (PE102-H) | 3331 | 0.84 | 0.16 | 0 | -1.00 | 1.15 |
| (PHYS201-M) | (PHYS101-M) | 2819 | 0.56 | 0.09 | 271 | 0.63 | 2.76 |
| (PHYS201-L) | (PHYS101-M) | 1856 | 0.51 | 0.02 | 228 | 0.66 | 2.47 |
| (PHYS102-M) | (PHYS101-M) | 1757 | 0.66 | 0.13 | 108 | 0.64 | 2.60 |
| (CHEM102-M) | (CHEM101-M) | 1562 | 0.68 | 0.13 | 0 | -1.00 | 2.11 |
| (CHEM101-M) | (CHEM102-M) | 1562 | 0.63 | 0.18 | 0 | -1.00 | 2.11 |
| (PHYS102-M) | (PHYS201-M) | 1479 | 0.56 | 0.21 | 162 | 0.38 | 2.64 |
| (PHYS130-M) | (PHYS101-M) | 1444 | 0.60 | 0.06 | 323 | 0.57 | 2.36 |
| (GER102-H) | (GER101-H) | 1266 | 0.66 | 0.23 | 136 | 0.81 | 1.56 |
| (PHYS130-M) | (PHYS201-M) | 1227 | 0.51 | 0.17 | 588 | 0.53 | 2.54 |
| (EC203-M) | (EC102-M) | 1132 | 0.52 | 0.05 | 181 | 0.52 | 1.40 |
| (EC203-M) | (EC101-M) | 1131 | 0.52 | 0.07 | 176 | 0.63 | 1.38 |
| (PHYS101-M)(PHYS102-M) | (PHYS201-M) | 1096 | 0.62 | 0.21 | 104 | 0.42 | 5.51 |
| (PHYS102-M)(PHYS201-M) | (PHYS101-M) | 1096 | 0.74 | 0.12 | 37 | 0.70 | 5.51 |
| (PHYS130-L) | (PHYS101-M) | 1067 | 0.58 | 0.02 | 247 | 0.66 | 2.22 |
| (EC351-M) | (EC102-M) | 942 | 0.55 | 0.05 | 77 | 0.57 | 1.21 |
| (EC351-M) | (EC101-M) | 927 | 0.54 | 0.05 | 84 | 0.56 | 1.17 |
| (EC351-M) | (EC203-M) | 927 | 0.54 | 0.06 | 218 | 0.51 | 1.32 |
| (GER201-H) | (GER102-H) | 911 | 0.65 | 0.36 | 150 | 0.71 | 1.60 |
| (GER201-H) | (GER101-H) | 860 | 0.61 | 0.23 | 92 | 0.87 | 1.42 |
| (PHYS101-M)(PHYS130-M) | (PHYS201-M) | 831 | 0.58 | 0.19 | 352 | 0.56 | 4.77 |
| (PHYS130-M)(PHYS201-M) | (PHYS101-M) | 831 | 0.68 | 0.08 | 125 | 0.58 | 4.77 |
| (PHYS202-M) | (PHYS201-M) | 829 | 0.59 | 0.15 | 136 | 0.51 | 2.45 |
| (FR102-H) | (FR101-H) | 816 | 0.70 | 0.28 | 106 | 0.76 | 1.63 |
| (EC305-M) | (EC101-M) | 785 | 0.51 | 0.05 | 0 | -1.00 | 1.32 |
| (PHYS102-L) | (PHYS201-L) | 784 | 0.51 | 0.28 | 121 | 0.55 | 3.21 |
| (GER201-H) | (GER202-H) | 782 | 0.55 | 0.55 | 154 | 0.77 | 1.53 |
| (GER202-H) | (GER201-H) | 782 | 0.72 | 0.38 | 167 | 0.63 | 1.53 |
| (EC204-M) | (EC102-M) | 781 | 0.55 | 0.08 | 0 | -1.00 | 1.72 |
| (EC204-M) | (EC203-M) | 775 | 0.55 | 0.10 | 1 | 0.00 | 1.86 |
| (EC305-M) | (EC102-M) | 770 | 0.50 | 0.05 | 0 | -1.00 | 1.33 |
| (CHEM103-M) | (CHEM109-H) | 770 | 0.58 | 0.00 | 250 | 0.92 | 1.20 |
| (CHEM109-H) | (CHEM103-M) | 770 | 0.60 | 0.12 | 413 | 0.59 | 1.20 |
| (EC204-M) | (EC101-M) | 761 | 0.54 | 0.05 | 0 | -1.00 | 1.65 |
| (AD442-M) | (AD341-M) | 739 | 0.50 | 0.22 | 91 | 0.74 | 1.11 |
| (CHEM103-M) | (CHEM104-M) | 735 | 0.56 | 0.16 | 317 | 0.63 | 2.17 |
| (CHEM104-M) | (CHEM103-M) | 735 | 0.64 | 0.11 | 240 | 0.72 | 2.17 |
| (PHYS101-L)(PHYS201-F) | (PHYS201-L) | 706 | 0.51 | 0.20 | 397 | 0.45 | 4.28 |
| (PHYS201-F)(PHYS201-L) | (PHYS101-L) | 706 | 0.50 | 0.36 | 81 | 0.32 | 4.28 |
| (PHYS101-L)(PHYS201-L) | (PHYS201-F) | 706 | 0.51 | 0.27 | 258 | 0.55 | 4.28 |

8 Grade Dataset

| LeftSide | RightSide | S | C | I | TS | TC | L |
|---|---|---|---|---|---|---|---|
| (TK221-AA) | (TK222-AA) | 4215 | 0.62 | 0.32 | 1304 | 0.70 | 1.39 |
| (HTR311-AA) | (HTR312-AA) | 3000 | 0.53 | 0.37 | 1075 | 0.68 | 1.49 |
| (PE101-AA) | (PE102-AA) | 2407 | 0.74 | 0.14 | 0 | -1.00 | 1.19 |
| (PE102-AA) | (PE101-AA) | 2407 | 0.69 | 0.17 | 0 | -1.00 | 1.19 |
| (AE111-AA) | (AE112-AA) | 1748 | 0.57 | 0.35 | 381 | 0.61 | 1.58 |
| (AE112-AA) | (AE111-AA) | 1748 | 0.52 | 0.47 | 400 | 0.61 | 1.58 |
| (GER102-AA) | (GER101-AA) | 665 | 0.57 | 0.32 | 88 | 0.75 | 2.00 |
| (GER201-AA) | (GER102-AA) | 485 | 0.55 | 0.54 | 86 | 0.70 | 2.25 |
| (PHYS130-DD) | (PHYS201-F) | 485 | 0.51 | 0.31 | 289 | 0.63 | 1.93 |
| (GER201-AA) | (GER101-AA) | 453 | 0.51 | 0.30 | 50 | 0.84 | 1.78 |
| (PHYS102-DD) | (PHYS201-F) | 417 | 0.54 | 0.35 | 70 | 0.71 | 1.95 |
| (FR102-AA) | (FR101-AA) | 413 | 0.60 | 0.41 | 72 | 0.71 | 2.19 |
| (MATH102-F)(MATH201-F) | (MATH101-F) | 406 | 0.65 | 0.46 | 151 | 0.46 | 1.45 |
| (GER202-AA) | (GER201-AA) | 400 | 0.61 | 0.50 | 127 | 0.48 | 2.04 |
| (PHYS101-F)(PHYS130-F) | (PHYS201-F) | 343 | 0.56 | 0.33 | 222 | 0.69 | 1.58 |
| (GER202-AA) | (GER102-AA) | 339 | 0.52 | 0.41 | 70 | 0.49 | 2.03 |
| (GER102-AA)(GER201-AA) | (GER101-AA) | 319 | 0.66 | 0.32 | 38 | 0.84 | 4.97 |
| (GER101-AA)(GER201-AA) | (GER102-AA) | 319 | 0.70 | 0.60 | 52 | 0.75 | 4.97 |
| (GER301-AA) | (GER302-AA) | 303 | 0.56 | 0.41 | 47 | 0.77 | 1.50 |
| (GER302-AA) | (GER301-AA) | 303 | 0.54 | 0.50 | 71 | 0.51 | 1.50 |
| (GER301-AA) | (GER201-AA) | 278 | 0.51 | 0.37 | 46 | 0.63 | 1.85 |
| (FR201-AA) | (FR102-AA) | 278 | 0.57 | 0.50 | 76 | 0.42 | 2.13 |
| (MATH101-F)(MATH202-F) | (MATH201-F) | 275 | 0.56 | 0.54 | 135 | 0.48 | 1.38 |
| (MATH201-F)(MATH202-F) | (MATH101-F) | 275 | 0.57 | 0.44 | 34 | 0.26 | 1.38 |
| (MATH251-F)(MATH151-F) | (MATH152-F) | 259 | 0.55 | 0.48 | 0 | -1.00 | 1.69 |
| (MATH251-F)(MATH152-F) | (MATH151-F) | 259 | 0.51 | 0.50 | 0 | -1.00 | 1.69 |
| (FR201-AA) | (FR101-AA) | 255 | 0.52 | 0.30 | 62 | 0.45 | 1.74 |
| (GER202-AA)(GER102-AA) | (GER201-AA) | 254 | 0.75 | 0.50 | 51 | 0.78 | 5.53 |
| (GER201-AA)(GER202-AA) | (GER102-AA) | 254 | 0.64 | 0.44 | 32 | 0.81 | 5.53 |
| (GER201-AA)(GER102-AA) | (GER202-AA) | 254 | 0.52 | 0.58 | 54 | 0.81 | 5.53 |
| (GER202-AA)(GER101-AA) | (GER201-AA) | 235 | 0.73 | 0.53 | 64 | 0.64 | 4.24 |
| (PHYS130-DD)(PHYS101-F) | (PHYS201-F) | 235 | 0.71 | 0.41 | 142 | 0.71 | 2.82 |
| (GER202-AA)(GER201-AA) | (GER101-AA) | 235 | 0.59 | 0.27 | 21 | 0.90 | 4.24 |
| (HIST496-AA) | (HIST495-AA) | 235 | 0.85 | 0.18 | 48 | 0.94 | 1.15 |
| (GER201-AA)(GER101-AA) | (GER202-AA) | 235 | 0.52 | 0.59 | 55 | 0.78 | 4.24 |
| (HIST495-AA) | (HIST496-AA) | 235 | 0.67 | 0.29 | 54 | 0.93 | 1.15 |
| (TKL102-AA) | (TKL101-AA) | 232 | 0.58 | 0.43 | 47 | 0.77 | 2.58 |
| (SCED271-AA) | (SCED272-AA) | 230 | 0.58 | 0.26 | 0 | -1.00 | 1.46 |
| (SCED272-AA) | (SCED271-AA) | 230 | 0.67 | 0.18 | 0 | -1.00 | 1.46 |
| (GER202-AA)(GER101-AA) | (GER102-AA) | 228 | 0.71 | 0.51 | 37 | 0.73 | 4.51 |
| (GER102-AA)(GER202-AA) | (GER101-AA) | 228 | 0.67 | 0.29 | 25 | 0.88 | 4.51 |
| (MATH202-F)(MATH102-F) | (MATH101-F) | 227 | 0.62 | 0.48 | 28 | 0.25 | 1.15 |
| (HTR406-AA) | (HTR405-AA) | 227 | 0.54 | 0.32 | 40 | 0.60 | 1.39 |
| (PHYS101-F)(PHYS102-F) | (PHYS201-F) | 224 | 0.59 | 0.34 | 58 | 0.81 | 1.73 |
| (MATH202-F)(MATH102-F) | (MATH201-F) | 220 | 0.60 | 0.58 | 109 | 0.53 | 1.40 |

Appendix E – Top 45 Rules Having Highest Importance

4 Grade Dataset

| LeftSide | RightSide | S | C | I | TS | TC | L |
|---|---|---|---|---|---|---|---|
| (JP101-H) | (JP102-H) | 282 | 0.53 | 0.99 | 38 | 0.68 | 1.29 |
| (PHYS101-H)(PHYS130-H) | (PHYS201-H) | 184 | 0.56 | 0.94 | 53 | 0.43 | 50.98 |
| (AD412-H) | (AD411-H) | 155 | 0.55 | 0.93 | 0 | -1.00 | 1.45 |
| (TR202-H) | (TR201-H) | 173 | 0.64 | 0.80 | 33 | 0.64 | 2.07 |
| (RUS101-H) | (RUS102-H) | 180 | 0.56 | 0.80 | 36 | 0.81 | 1.36 |
| (FR301-H) | (FR302-H) | 219 | 0.55 | 0.79 | 78 | 0.83 | 1.51 |
| (JP102-H) | (JP201-H) | 174 | 0.53 | 0.76 | 33 | 0.73 | 1.30 |
| (PHYS130-H)(PHYS201-H) | (PHYS101-H) | 184 | 0.74 | 0.70 | 27 | 0.63 | 50.98 |
| (JP101-H)(JP102-H) | (JP201-H) | 151 | 0.54 | 0.67 | 24 | 0.75 | 1.85 |
| (PHYS202-H) | (PHYS201-H) | 306 | 0.58 | 0.64 | 50 | 0.66 | 9.37 |
| (FLED412-H) | (FLED416-H) | 362 | 0.77 | 0.63 | 86 | 0.84 | 1.15 |
| (AD214-H) | (AD213-H) | 361 | 0.62 | 0.62 | 90 | 0.77 | 1.96 |
| (FR202-H)(FR301-H) | (FR302-H) | 155 | 0.58 | 0.62 | 57 | 0.84 | 2.85 |
| (AD353-L) | (AD351-L) | 486 | 0.54 | 0.61 | 97 | 0.57 | 2.47 |
| (AD213-H) | (AD214-H) | 361 | 0.70 | 0.61 | 97 | 0.71 | 1.96 |
| (FR201-H)(FR202-H) | (FR301-H) | 193 | 0.50 | 0.60 | 85 | 0.79 | 2.73 |
| (FR102-H)(FR201-H)(FR202-H) | (FR301-H) | 153 | 0.53 | 0.59 | 61 | 0.82 | 5.67 |
| (CHEM101-F)(CHEM101-L) | (CHEM102-F) | 152 | 0.59 | 0.59 | 0 | -1.00 | 5.27 |
| (PHYS121-H) | (PHYS201-H) | 244 | 0.53 | 0.58 | 101 | 0.57 | 9.24 |
| (FR101-H)(FR201-H) | (FR202-H) | 270 | 0.53 | 0.58 | 79 | 0.76 | 2.65 |
| (AD104-H)(AD351-H) | (AD320-H) | 191 | 0.61 | 0.57 | 20 | 0.70 | 4.01 |
| (TR201-H) | (TR202-H) | 173 | 0.77 | 0.57 | 27 | 0.78 | 2.07 |
| (FLED201-H) | (FLED212-H) | 164 | 0.60 | 0.57 | 63 | 0.70 | 1.77 |
| (FLED212-H) | (FLED313-H) | 157 | 0.62 | 0.57 | 61 | 0.92 | 1.29 |
| (FR101-H)(FR102-H)(FR201-H) | (FR202-H) | 236 | 0.58 | 0.56 | 62 | 0.79 | 5.66 |
| (MIS212-H) | (MIS131-H) | 174 | 0.50 | 0.56 | 17 | 0.47 | 1.85 |
| (MIS211-H) | (MIS212-H) | 220 | 0.79 | 0.56 | 20 | 0.70 | 1.81 |
| (FLED304-H) | (FLED313-H) | 151 | 0.51 | 0.56 | 40 | 0.70 | 1.38 |
| (FR102-H)(FR201-H) | (FR202-H) | 290 | 0.54 | 0.56 | 76 | 0.75 | 3.03 |
| (AD104-H)(AD213-H) | (AD214-H) | 266 | 0.76 | 0.55 | 59 | 0.76 | 2.78 |
| (AD353-H) | (AD351-H) | 219 | 0.64 | 0.55 | 63 | 0.29 | 3.66 |
| (GER201-H) | (GER202-H) | 782 | 0.55 | 0.55 | 154 | 0.77 | 1.53 |
| (MIS131-H) | (MIS212-H) | 174 | 0.71 | 0.55 | 32 | 0.31 | 1.85 |
| (AD104-H)(AD220-H) | (AD202-H) | 227 | 0.59 | 0.55 | 75 | 0.60 | 3.06 |
| (CHEM204-H) | (CHEM203-H) | 244 | 0.60 | 0.55 | 35 | 0.40 | 1.89 |
| (AD104-H)(AD311-H) | (AD351-H) | 188 | 0.58 | 0.54 | 71 | 0.23 | 4.00 |
| (FR302-H) | (FR301-H) | 219 | 0.72 | 0.54 | 76 | 0.80 | 1.51 |
| (AD104-H)(AD214-H) | (AD213-H) | 266 | 0.65 | 0.54 | 62 | 0.73 | 2.78 |
| (AD104-H)(AD320-H) | (AD351-H) | 191 | 0.59 | 0.54 | 63 | 0.21 | 4.01 |
| (FLED416-H) | (FLED412-H) | 362 | 0.89 | 0.53 | 76 | 0.95 | 1.15 |
| (AD104-H)(AD202-H) | (AD351-H) | 211 | 0.54 | 0.53 | 73 | 0.23 | 3.79 |
| (AD104-H)(AD202-H) | (AD320-H) | 206 | 0.53 | 0.53 | 63 | 0.65 | 2.80 |
| (AD104-H)(AD214-H) | (AD202-H) | 245 | 0.60 | 0.53 | 60 | 0.63 | 2.58 |
| (AD202-H) | (AD351-H) | 287 | 0.50 | 0.53 | 89 | 0.22 | 2.50 |
| (MIS212-H) | (MIS211-H) | 220 | 0.63 | 0.53 | 20 | 0.60 | 1.81 |

8 Grade Dataset

| LeftSide | RightSide | S | C | I | TS | TC | L |
|---|---|---|---|---|---|---|---|
| (TR202-AA) | (TR201-AA) | 83 | 0.55 | 1.04 | 17 | 0.47 | 3.71 |
| (MATH101-AA)(MATH201-AA)(MATH102-AA) | (MATH202-AA) | 85 | 0.61 | 1.01 | 26 | 0.88 | 1081.4 |
| (MATH101-AA)(MATH201-AA) | (MATH202-AA) | 106 | 0.53 | 1.01 | 38 | 0.71 | 99.08 |
| (MATH152-AA)(MATH151-AA) | (MATH251-AA) | 75 | 0.51 | 1.00 | 0 | -1.00 | 138.1 |
| (MATH151-AA)(MATH152-AA) | (MATH252-AA) | 83 | 0.56 | 1.00 | 0 | -1.00 | 128.32 |
| (MATH102-AA)(MATH201-AA) | (MATH202-AA) | 131 | 0.53 | 0.99 | 42 | 0.76 | 84.1 |
| (MATH102-AA)(MATH101-AA)(MATH202-AA) | (MATH201-AA) | 85 | 0.67 | 0.95 | 31 | 0.61 | 1081.4 |
| (MATH102-AA)(MATH202-AA) | (MATH201-AA) | 131 | 0.59 | 0.95 | 55 | 0.49 | 84.1 |
| (MATH101-AA)(MATH202-AA) | (MATH201-AA) | 106 | 0.61 | 0.94 | 47 | 0.49 | 99.08 |
| (PHYS202-AA) | (PHYS201-AA) | 140 | 0.54 | 0.89 | 32 | 0.66 | 18.64 |
| (MATH251-AA)(MATH151-AA) | (MATH252-AA) | 84 | 0.62 | 0.87 | 0 | -1.00 | 102.41 |
| (MATH151-AA)(MATH251-AA) | (MATH152-AA) | 75 | 0.56 | 0.85 | 0 | -1.00 | 138.1 |
| (MATH201-AA)(MATH101-AA) | (MATH102-AA) | 140 | 0.70 | 0.85 | 46 | 0.59 | 92.44 |
| (MATH152-AA)(MATH252-AA) | (MATH151-AA) | 83 | 0.55 | 0.83 | 0 | -1.00 | 128.32 |
| (MATH151-AA)(MATH252-AA) | (MATH251-AA) | 84 | 0.54 | 0.83 | 0 | -1.00 | 102.41 |
| (MATH102-AA)(MATH201-AA) | (MATH101-AA) | 140 | 0.57 | 0.83 | 40 | 0.60 | 92.44 |
| (MATH252-AA)(MATH251-AA) | (MATH151-AA) | 84 | 0.51 | 0.82 | 0 | -1.00 | 102.41 |
| (MATH252-AA)(MATH151-AA) | (MATH152-AA) | 83 | 0.53 | 0.81 | 0 | -1.00 | 128.32 |
| (TR201-AA) | (TR202-AA) | 83 | 0.73 | 0.81 | 12 | 0.67 | 3.71 |
| (MATH252-AA)(MATH152-AA) | (MATH251-AA) | 82 | 0.54 | 0.81 | 0 | -1.00 | 91.99 |
| (MATH152-AA)(MATH251-AA) | (MATH252-AA) | 82 | 0.54 | 0.80 | 0 | -1.00 | 91.99 |
| (MATH201-AA)(MATH102-AA)(MATH202-AA) | (MATH101-AA) | 85 | 0.65 | 0.79 | 15 | 0.53 | 1081.4 |
| (MATH101-AA)(MATH202-AA) | (MATH102-AA) | 126 | 0.72 | 0.79 | 28 | 0.68 | 90.22 |
| (MATH252-AA)(MATH251-AA) | (MATH152-AA) | 82 | 0.50 | 0.78 | 0 | -1.00 | 91.99 |
| (FR201-AA)(FR101-AA) | (FR202-AA) | 139 | 0.55 | 0.78 | 41 | 0.56 | 5.81 |
| (MATH202-AA)(MATH201-AA)(MATH101-AA) | (MATH102-AA) | 85 | 0.80 | 0.77 | 13 | 0.85 | 1081.4 |
| (AD213-AA) | (AD214-AA) | 135 | 0.58 | 0.76 | 48 | 0.77 | 3.15 |
| (MATH202-AA)(MATH102-AA) | (MATH101-AA) | 126 | 0.57 | 0.75 | 33 | 0.48 | 90.22 |
| (FR101-AA)(FR201-AA)(FR102-AA) | (FR202-AA) | 107 | 0.58 | 0.75 | 28 | 0.68 | 18.43 |
| (FR201-AA)(FR102-AA) | (FR202-AA) | 148 | 0.53 | 0.75 | 36 | 0.69 | 7.00 |
| (MATH201-AA)(MATH202-AA) | (MATH102-AA) | 131 | 0.61 | 0.74 | 25 | 0.72 | 84.1 |
| (AD341-AA)(AD214-AA) | (AD213-AA) | 78 | 0.55 | 0.74 | 30 | 0.70 | 8.65 |
| (AD104-AA)(AD231-AA) | (AD202-AA) | 84 | 0.52 | 0.73 | 82 | 0.30 | 4.61 |
| (EC102-AA)(EC351-AA) | (EC203-AA) | 83 | 0.53 | 0.73 | 23 | 0.48 | 12.18 |
| (AD341-AA)(AD202-AA) | (AD220-AA) | 80 | 0.51 | 0.72 | 26 | 0.81 | 11.35 |
| (ELT125-AA) | (ELT126-AA) | 102 | 0.63 | 0.72 | 0 | -1.00 | 1.55 |
| (AD220-AA)(AD231-AA) | (AD202-AA) | 78 | 0.56 | 0.71 | 65 | 0.48 | 9.61 |
| (AD341-AA)(AD353-AA) | (AD311-AA) | 80 | 0.63 | 0.71 | 25 | 0.56 | 17.04 |
| (ED101-AA)(ED221-AA) | (ED213-AA) | 91 | 0.58 | 0.71 | 46 | 0.65 | 4.71 |
| (AD231-AA)(AD202-AA) | (AD220-AA) | 78 | 0.54 | 0.71 | 37 | 0.84 | 9.61 |
| (AD231-AA)(AD401-AA) | (AD202-AA) | 76 | 0.51 | 0.71 | 11 | 0.64 | 6.32 |
| (AD341-AA)(AD353-AA) | (AD351-AA) | 76 | 0.59 | 0.70 | 22 | 0.41 | 19.94 |
| (AD214-AA)(AD341-AA) | (AD311-AA) | 82 | 0.58 | 0.70 | 43 | 0.56 | 9.58 |
| (AD341-AA)(AD213-AA) | (AD214-AA) | 78 | 0.64 | 0.68 | 25 | 0.84 | 8.65 |
| (AD341-AA)(AD214-AA) | (AD220-AA) | 80 | 0.57 | 0.68 | 31 | 0.74 | 9.27 |

## Appendix F – Top 45 Rules Having Highest Lift

### 4 Grade Dataset

| LeftSide | RightSide | S | C | I | TS | TC | L |
|---|---|---|---|---|---|---|---|
| (PHYS101-H)(PHYS130-H) | (PHYS201-H) | 184 | 0.56 | 0.94 | 53 | 0.43 | 50.98 |
| (PHYS130-H)(PHYS201-H) | (PHYS101-H) | 184 | 0.74 | 0.70 | 27 | 0.63 | 50.98 |
| (PHYS101-L)(PHYS202-L) | (PHYS201-L) | 175 | 0.68 | 0.32 | 23 | 0.83 | 10.25 |
| (PHYS101-F)(PHYS102-L)(PHYS201-L) | (PHYS101-L) | 160 | 0.59 | 0.30 | 12 | 0.25 | 9.71 |
| (PHYS101-F)(PHYS101-L)(PHYS102-L) | (PHYS201-L) | 160 | 0.62 | 0.24 | 31 | 0.58 | 9.71 |
| (PHYS202-H) | (PHYS201-H) | 306 | 0.58 | 0.64 | 50 | 0.66 | 9.37 |
| (PHYS121-H) | (PHYS201-H) | 244 | 0.53 | 0.58 | 101 | 0.57 | 9.24 |
| (PHYS101-L)(PHYS102-L) | (PHYS201-L) | 429 | 0.63 | 0.29 | 57 | 0.60 | 8.70 |
| (PHYS102-L)(PHYS201-L) | (PHYS101-L) | 429 | 0.55 | 0.33 | 32 | 0.41 | 8.70 |
| (PHYS101-F)(PHYS101-L)(PHYS130-L) | (PHYS201-L) | 163 | 0.51 | 0.24 | 115 | 0.50 | 8.50 |
| (PHYS101-F)(PHYS130-L)(PHYS201-L) | (PHYS101-L) | 163 | 0.57 | 0.27 | 18 | 0.33 | 8.50 |
| (PHYS102-L)(PHYS202-L) | (PHYS101-L) | 153 | 0.60 | 0.33 | 10 | 0.50 | 8.43 |
| (PHYS101-L)(PHYS202-L) | (PHYS102-L) | 153 | 0.60 | 0.37 | 15 | 0.53 | 8.43 |
| (EC203-H)(EC204-H) | (EC102-H) | 203 | 0.57 | 0.38 | 0 | -1.00 | 8.24 |
| (EC102-H)(EC204-H) | (EC203-H) | 203 | 0.62 | 0.46 | 0 | -1.00 | 8.24 |
| (PHYS102-L)(PHYS202-L) | (PHYS201-L) | 170 | 0.67 | 0.32 | 23 | 0.78 | 8.17 |
| (EC101-H)(EC204-H) | (EC102-H) | 204 | 0.68 | 0.42 | 0 | -1.00 | 7.46 |
| (EC102-H)(EC204-H) | (EC101-H) | 204 | 0.62 | 0.39 | 0 | -1.00 | 7.46 |
| (EC203-H)(EC204-H) | (EC101-H) | 193 | 0.55 | 0.36 | 0 | -1.00 | 7.44 |
| (EC101-H)(EC204-H) | (EC203-H) | 193 | 0.64 | 0.47 | 0 | -1.00 | 7.44 |
| (EC102-H)(EC203-H) | (EC101-H) | 391 | 0.67 | 0.40 | 49 | 0.76 | 7.20 |
| (EC101-H)(EC203-H) | (EC102-H) | 391 | 0.67 | 0.41 | 46 | 0.76 | 7.20 |
| (PHYS101-F)(PHYS101-L)(PHYS130-L) | (PHYS201-F) | 231 | 0.73 | 0.43 | 115 | 0.70 | 7.11 |
| (PHYS101-L)(PHYS130-L)(PHYS201-F) | (PHYS101-F) | 231 | 0.51 | 0.39 | 38 | 0.24 | 7.11 |
| (PHYS101-F)(PHYS130-L)(PHYS201-F) | (PHYS101-L) | 231 | 0.57 | 0.28 | 23 | 0.39 | 7.11 |
| (PHYS101-L)(PHYS130-L) | (PHYS201-L) | 397 | 0.52 | 0.29 | 206 | 0.49 | 6.92 |
| (PHYS130-L)(PHYS201-L) | (PHYS101-L) | 397 | 0.50 | 0.26 | 60 | 0.23 | 6.92 |
| (EC204-H)(EC351-H) | (EC203-H) | 227 | 0.62 | 0.47 | 0 | -1.00 | 6.88 |
| (EC203-H)(EC204-H) | (EC351-H) | 227 | 0.64 | 0.37 | 0 | -1.00 | 6.88 |
| (PHYS101-F)(PHYS101-L)(PHYS102-L) | (PHYS201-F) | 184 | 0.71 | 0.41 | 31 | 0.87 | 6.58 |
| (PHYS101-F)(PHYS102-L)(PHYS201-F) | (PHYS101-L) | 184 | 0.62 | 0.32 | 20 | 0.40 | 6.58 |
| (EC102-H)(EC204-H) | (EC351-H) | 208 | 0.63 | 0.37 | 0 | -1.00 | 6.48 |
| (EC204-H)(EC351-H) | (EC102-H) | 208 | 0.56 | 0.37 | 0 | -1.00 | 6.48 |
| (EC204-H)(EC351-H) | (EC232-H) | 190 | 0.51 | 0.46 | 0 | -1.00 | 6.22 |
| (EC204-H)(EC232-H) | (EC351-H) | 190 | 0.80 | 0.33 | 0 | -1.00 | 6.22 |
| (PHYS101-L)(PHYS102-F) | (PHYS201-L) | 290 | 0.51 | 0.17 | 59 | 0.42 | 6.12 |
| (PHYS102-F)(PHYS201-L) | (PHYS101-L) | 290 | 0.62 | 0.36 | 8 | 0.63 | 6.12 |
| (PHYS101-F)(PHYS101-L)(PHYS201-L) | (PHYS201-F) | 331 | 0.68 | 0.37 | 125 | 0.62 | 6.00 |
| (PHYS101-F)(PHYS201-F)(PHYS201-L) | (PHYS101-L) | 331 | 0.57 | 0.27 | 24 | 0.25 | 6.00 |
| (PHYS101-F)(PHYS101-L)(PHYS201-F) | (PHYS201-L) | 331 | 0.52 | 0.19 | 228 | 0.44 | 6.00 |
| (EC204-L)(EC361-L) | (EC331-L) | 159 | 0.69 | 0.22 | 1 | 0.00 | 5.87 |
| (EC204-L)(EC331-L) | (EC361-L) | 159 | 0.68 | 0.18 | 0 | -1.00 | 5.87 |
| (PHYS101-F)(PHYS101-L)(PHYS130-L) | (PHYS130-F) | 180 | 0.57 | 0.32 | 98 | 0.40 | 5.79 |
| (PHYS101-F)(PHYS130-F)(PHYS130-L) | (PHYS101-L) | 180 | 0.57 | 0.26 | 29 | 0.48 | 5.79 |
| (PHYS101-L)(PHYS130-F)(PHYS130-L) | (PHYS101-F) | 180 | 0.58 | 0.40 | 52 | 0.27 | 5.79 |

8 Grade Dataset

| LeftSide | RightSide | S | C | I | TS | TC | L |
|---|---|---|---|---|---|---|---|
| (MATH202-AA)(MATH201-AA)(MATH101-AA) | (MATH102-AA) | 85 | 0.80 | 0.77 | 13 | 0.85 | 1081.4 |
| (MATH201-AA)(MATH102-AA)(MATH202-AA) | (MATH101-AA) | 85 | 0.65 | 0.79 | 15 | 0.53 | 1081.4 |
| (MATH102-AA)(MATH101-AA)(MATH202-AA) | (MATH201-AA) | 85 | 0.67 | 0.95 | 31 | 0.61 | 1081.4 |
| (MATH101-AA)(MATH201-AA)(MATH102-AA) | (MATH202-AA) | 85 | 0.61 | 1.01 | 26 | 0.88 | 1081.4 |
| (MATH152-AA)(MATH151-AA) | (MATH251-AA) | 75 | 0.51 | 1.00 | 0 | -1.0 | 138.10 |
| (MATH151-AA)(MATH251-AA) | (MATH152-AA) | 75 | 0.56 | 0.85 | 0 | -1.0 | 138.10 |
| (MATH152-AA)(MATH252-AA) | (MATH151-AA) | 83 | 0.55 | 0.83 | 0 | -1.0 | 128.32 |
| (MATH252-AA)(MATH151-AA) | (MATH152-AA) | 83 | 0.53 | 0.81 | 0 | -1.0 | 128.32 |
| (MATH151-AA)(MATH152-AA) | (MATH252-AA) | 83 | 0.56 | 1.00 | 0 | -1.0 | 128.32 |
| (MATH151-AA)(MATH252-AA) | (MATH251-AA) | 84 | 0.54 | 0.83 | 0 | -1.0 | 102.41 |
| (MATH251-AA)(MATH151-AA) | (MATH252-AA) | 84 | 0.62 | 0.87 | 0 | -1.0 | 102.41 |
| (MATH252-AA)(MATH251-AA) | (MATH151-AA) | 84 | 0.51 | 0.82 | 0 | -1.0 | 102.41 |
| (MATH101-AA)(MATH202-AA) | (MATH201-AA) | 106 | 0.61 | 0.94 | 47 | 0.49 | 99.08 |
| (MATH101-AA)(MATH201-AA) | (MATH202-AA) | 106 | 0.53 | 1.01 | 38 | 0.71 | 99.08 |
| (MATH201-AA)(MATH101-AA) | (MATH102-AA) | 140 | 0.70 | 0.85 | 46 | 0.59 | 92.44 |
| (MATH102-AA)(MATH201-AA) | (MATH101-AA) | 140 | 0.57 | 0.83 | 40 | 0.60 | 92.44 |
| (MATH252-AA)(MATH152-AA) | (MATH251-AA) | 82 | 0.54 | 0.81 | 0 | -1.0 | 91.99 |
| (MATH252-AA)(MATH251-AA) | (MATH152-AA) | 82 | 0.50 | 0.78 | 0 | -1.0 | 91.99 |
| (MATH152-AA)(MATH251-AA) | (MATH252-AA) | 82 | 0.54 | 0.80 | 0 | -1.0 | 91.99 |
| (MATH101-AA)(MATH202-AA) | (MATH102-AA) | 126 | 0.72 | 0.79 | 28 | 0.68 | 90.22 |
| (MATH202-AA)(MATH102-AA) | (MATH101-AA) | 126 | 0.57 | 0.75 | 33 | 0.48 | 90.22 |
| (MATH201-AA)(MATH202-AA) | (MATH102-AA) | 131 | 0.61 | 0.74 | 25 | 0.72 | 84.10 |
| (MATH102-AA)(MATH202-AA) | (MATH201-AA) | 131 | 0.59 | 0.95 | 55 | 0.49 | 84.10 |
| (MATH102-AA)(MATH201-AA) | (MATH202-AA) | 131 | 0.53 | 0.99 | 42 | 0.76 | 84.10 |
| (AD353-AA)(AD351-AA) | (AD341-AA) | 76 | 0.78 | 0.48 | 9 | 1.00 | 19.94 |
| (AD341-AA)(AD353-AA) | (AD351-AA) | 76 | 0.59 | 0.70 | 22 | 0.41 | 19.94 |
| (EC101-AA)(EC203-AA) | (EC102-AA) | 131 | 0.62 | 0.66 | 21 | 0.76 | 19.36 |
| (EC102-AA)(EC203-AA) | (EC101-AA) | 131 | 0.65 | 0.65 | 23 | 0.74 | 19.36 |
| (PHYS202-AA) | (PHYS201-AA) | 140 | 0.54 | 0.89 | 32 | 0.66 | 18.64 |
| (FR201-AA)(FR202-AA)(FR102-AA) | (FR101-AA) | 107 | 0.72 | 0.32 | 5 | 0.60 | 18.43 |
| (FR101-AA)(FR201-AA)(FR102-AA) | (FR202-AA) | 107 | 0.58 | 0.75 | 28 | 0.68 | 18.43 |
| (FR102-AA)(FR202-AA)(FR101-AA) | (FR201-AA) | 107 | 0.79 | 0.57 | 20 | 0.80 | 18.43 |
| (FR202-AA)(FR201-AA)(FR101-AA) | (FR102-AA) | 107 | 0.77 | 0.44 | 5 | 0.80 | 18.43 |
| (AD311-AA)(AD353-AA) | (AD341-AA) | 80 | 0.83 | 0.50 | 13 | 0.92 | 17.04 |
| (AD341-AA)(AD353-AA) | (AD311-AA) | 80 | 0.63 | 0.71 | 25 | 0.56 | 17.04 |
| (AD351-AA)(AD341-AA) | (AD311-AA) | 95 | 0.53 | 0.67 | 12 | 0.75 | 14.10 |
| (AD311-AA)(AD351-AA) | (AD341-AA) | 95 | 0.77 | 0.47 | 10 | 0.80 | 14.10 |
| (GER301-AA)(GER102-AA)(GER201-AA) | (GER202-AA) | 137 | 0.75 | 0.51 | 25 | 0.84 | 13.24 |
| (GER102-AA)(GER201-AA)(GER202-AA) | (GER301-AA) | 137 | 0.54 | 0.53 | 33 | 0.88 | 13.24 |
| (GER202-AA)(GER301-AA)(GER201-AA) | (GER102-AA) | 137 | 0.71 | 0.43 | 12 | 0.92 | 13.24 |
| (GER202-AA)(GER102-AA)(GER301-AA) | (GER201-AA) | 137 | 0.83 | 0.48 | 21 | 0.86 | 13.24 |
| (GER202-AA)(GER201-AA)(GER102-AA) | (GER101-AA) | 183 | 0.72 | 0.32 | 17 | 0.94 | 13.01 |
| (GER202-AA)(GER102-AA)(GER101-AA) | (GER201-AA) | 183 | 0.80 | 0.53 | 39 | 0.79 | 13.01 |
| (GER101-AA)(GER201-AA)(GER102-AA) | (GER202-AA) | 183 | 0.57 | 0.61 | 40 | 0.83 | 13.01 |
| (GER202-AA)(GER101-AA)(GER201-AA) | (GER102-AA) | 183 | 0.78 | 0.52 | 25 | 0.84 | 13.01 |

Appendix G – Top 45 Rules Having Highest Test Support

4 Grade Dataset

| LeftSide | RightSide | S | C | I | TS | TC | L |
|---|---|---|---|---|---|---|---|
| (TK222-H) | (TK221-H) | 8443 | 0.66 | 0.32 | 1872 | 0.81 | 1.17 |
| (TK221-H) | (TK222-H) | 8443 | 0.76 | 0.24 | 1836 | 0.81 | 1.17 |
| (EC101-M) | (EC102-M) | 4167 | 0.51 | 0.12 | 1063 | 0.55 | 1.40 |
| (EC102-M) | (EC101-M) | 4167 | 0.53 | 0.12 | 865 | 0.65 | 1.40 |
| (PHYS130-M) | (PHYS201-M) | 1227 | 0.51 | 0.17 | 588 | 0.53 | 2.54 |
| (PHYS101-F)(PHYS201-F) | (PHYS201-L) | 584 | 0.51 | 0.19 | 435 | 0.44 | 2.20 |
| (CHEM109-H) | (CHEM103-M) | 770 | 0.60 | 0.12 | 413 | 0.59 | 1.20 |
| (PHYS101-L)(PHYS201-F) | (PHYS201-L) | 706 | 0.51 | 0.20 | 397 | 0.45 | 4.28 |
| (PHYS101-M)(PHYS130-M) | (PHYS201-M) | 831 | 0.58 | 0.19 | 352 | 0.56 | 4.77 |
| (CHEM109-H) | (CHEM104-M) | 691 | 0.54 | 0.22 | 330 | 0.54 | 1.31 |
| (PHYS130-M) | (PHYS101-M) | 1444 | 0.60 | 0.06 | 323 | 0.57 | 2.36 |
| (CHEM103-M) | (CHEM104-M) | 735 | 0.56 | 0.16 | 317 | 0.63 | 2.17 |
| (PHYS121-M) | (PHYS201-M) | 472 | 0.52 | 0.17 | 313 | 0.55 | 2.06 |
| (PHYS201-M) | (PHYS101-M) | 2819 | 0.56 | 0.09 | 271 | 0.63 | 2.76 |
| (PHYS101-F)(PHYS101-L) | (PHYS201-F) | 632 | 0.57 | 0.35 | 267 | 0.70 | 2.61 |
| (PHYS101-F)(PHYS101-M) | (PHYS201-F) | 594 | 0.54 | 0.33 | 264 | 0.70 | 1.40 |
| (PHYS101-L)(PHYS201-L) | (PHYS201-F) | 706 | 0.51 | 0.27 | 258 | 0.55 | 4.28 |
| (CHEM103-M) | (CHEM109-H) | 770 | 0.58 | 0.00 | 250 | 0.92 | 1.20 |
| (PHYS101-F)(PHYS201-L) | (PHYS201-F) | 584 | 0.64 | 0.37 | 249 | 0.59 | 2.20 |
| (AD232-H) | (AD341-H) | 534 | 0.61 | 0.20 | 249 | 0.53 | 1.49 |
| (PHYS101-L)(PHYS130-F) | (PHYS201-F) | 373 | 0.51 | 0.29 | 247 | 0.58 | 2.78 |
| (PHYS130-L) | (PHYS101-M) | 1067 | 0.58 | 0.02 | 247 | 0.66 | 2.22 |
| (CHEM104-M) | (CHEM103-M) | 735 | 0.64 | 0.11 | 240 | 0.72 | 2.17 |
| (PHYS130-L)(PHYS201-L) | (PHYS201-F) | 418 | 0.53 | 0.29 | 234 | 0.51 | 3.89 |
| (PHYS130-F)(PHYS201-L) | (PHYS201-F) | 322 | 0.59 | 0.32 | 230 | 0.50 | 2.18 |
| (PHYS201-L) | (PHYS101-M) | 1856 | 0.51 | 0.02 | 228 | 0.66 | 2.47 |
| (PHYS101-F)(PHYS101-L)(PHYS201-F) | (PHYS201-L) | 331 | 0.52 | 0.19 | 228 | 0.44 | 6.00 |
| (AD231-H) | (AD341-H) | 552 | 0.66 | 0.26 | 225 | 0.54 | 1.54 |
| (PHYS130-F)(PHYS130-M) | (PHYS201-F) | 279 | 0.51 | 0.27 | 224 | 0.49 | 1.36 |
| (PHYS101-F)(PHYS130-F) | (PHYS201-F) | 343 | 0.56 | 0.33 | 222 | 0.69 | 1.59 |
| (PHYS101-F)(PHYS130-L) | (PHYS201-F) | 403 | 0.69 | 0.45 | 220 | 0.72 | 2.57 |
| (EC351-M) | (EC203-M) | 927 | 0.54 | 0.06 | 218 | 0.51 | 1.32 |
| (PHYS130-F)(PHYS130-L) | (PHYS201-L) | 294 | 0.51 | 0.25 | 216 | 0.51 | 3.10 |
| (PHYS130-F)(PHYS130-L) | (PHYS201-F) | 381 | 0.66 | 0.42 | 216 | 0.68 | 2.37 |
| (PHYS101-L)(PHYS130-L) | (PHYS201-F) | 454 | 0.60 | 0.39 | 206 | 0.66 | 4.66 |
| (PHYS101-L)(PHYS130-L) | (PHYS201-L) | 397 | 0.52 | 0.29 | 206 | 0.49 | 6.92 |
| (CHEM103-M)(CHEM109-H) | (CHEM104-M) | 483 | 0.63 | 0.20 | 201 | 0.62 | 2.28 |
| (AD220-H) | (AD341-H) | 492 | 0.52 | 0.25 | 200 | 0.63 | 1.58 |
| (PHYS101-F)(PHYS130-F) | (PHYS130-L) | 314 | 0.51 | 0.19 | 199 | 0.47 | 2.09 |
| (EC361-M) | (EC351-M) | 445 | 0.54 | 0.08 | 196 | 0.64 | 1.32 |
| (PHYS130-F)(PHYS201-F) | (PHYS130-L) | 381 | 0.55 | 0.20 | 192 | 0.48 | 2.37 |
| (EC233-M) | (EC351-M) | 519 | 0.52 | 0.13 | 187 | 0.58 | 1.40 |
| (EC203-M) | (EC102-M) | 1132 | 0.52 | 0.05 | 181 | 0.52 | 1.40 |
| (PHYS101-F)(PHYS130-L) | (PHYS130-F) | 314 | 0.54 | 0.32 | 181 | 0.36 | 2.09 |
| (AD232-H) | (AD231-H) | 476 | 0.54 | 0.35 | 178 | 0.85 | 1.40 |

8 Grade Dataset

| LeftSide | RightSide | S | C | I | TS | TC | L |
|---|---|---|---|---|---|---|---|
| (TK221-AA) | (TK222-AA) | 4215 | 0.62 | 0.32 | 1304 | 0.70 | 1.39 |
| (HTR311-AA) | (HTR312-AA) | 3000 | 0.53 | 0.37 | 1075 | 0.68 | 1.49 |
| (AE112-AA) | (AE111-AA) | 1748 | 0.52 | 0.47 | 400 | 0.61 | 1.58 |
| (AE111-AA) | (AE112-AA) | 1748 | 0.57 | 0.35 | 381 | 0.61 | 1.58 |
| (PHYS130-DD) | (PHYS201-F) | 485 | 0.51 | 0.31 | 289 | 0.63 | 1.93 |
| (PHYS101-F)(PHYS130-F) | (PHYS201-F) | 343 | 0.56 | 0.33 | 222 | 0.69 | 1.58 |
| (MATH102-F)(MATH201-F) | (MATH101-F) | 406 | 0.65 | 0.46 | 151 | 0.46 | 1.45 |
| (PHYS130-F)(PHYS101-DD) | (PHYS201-F) | 218 | 0.51 | 0.25 | 147 | 0.59 | 3.34 |
| (PHYS130-DD)(PHYS101-F) | (PHYS201-F) | 235 | 0.71 | 0.41 | 142 | 0.71 | 2.82 |
| (PHYS101-CC)(PHYS130-F) | (PHYS201-F) | 216 | 0.52 | 0.27 | 136 | 0.60 | 2.18 |
| (MATH101-F)(MATH202-F) | (MATH201-F) | 275 | 0.56 | 0.54 | 135 | 0.48 | 1.38 |
| (GER202-AA) | (GER201-AA) | 400 | 0.61 | 0.50 | 127 | 0.48 | 2.04 |
| (MATH101-DD)(MATH201-F) | (MATH102-F) | 186 | 0.52 | 0.48 | 114 | 0.59 | 3.24 |
| (PHYS130-F)(PHYS101-DC) | (PHYS201-F) | 186 | 0.54 | 0.27 | 111 | 0.60 | 2.69 |
| (AD232-AA)(AD104-AA) | (AD341-AA) | 76 | 0.51 | 0.28 | 110 | 0.48 | 3.21 |
| (AD231-AA)(AD104-AA) | (AD341-AA) | 103 | 0.64 | 0.40 | 109 | 0.45 | 3.80 |
| (MATH202-F)(MATH102-F) | (MATH201-F) | 220 | 0.60 | 0.58 | 109 | 0.53 | 1.40 |
| (AD232-AA)(AD231-AA) | (AD341-AA) | 117 | 0.58 | 0.39 | 103 | 0.53 | 4.74 |
| (ED213-AA)(ED101-AA) | (ED221-AA) | 91 | 0.53 | 0.51 | 102 | 0.42 | 4.71 |
| (MATH201-F)(MATH102-DD) | (MATH101-F) | 210 | 0.63 | 0.38 | 95 | 0.33 | 2.65 |
| (CHEM103-BB) | (CHEM109-AA) | 195 | 0.51 | 0.14 | 94 | 0.76 | 1.42 |
| (AD231-AA)(AD104-AA) | (AD232-AA) | 82 | 0.51 | 0.41 | 93 | 0.75 | 3.00 |
| (GER102-AA) | (GER101-AA) | 665 | 0.57 | 0.32 | 88 | 0.75 | 2.00 |
| (GER201-AA) | (GER102-AA) | 485 | 0.55 | 0.54 | 86 | 0.70 | 2.25 |
| (PHYS101-CC)(PHYS130-DD) | (PHYS201-F) | 159 | 0.53 | 0.27 | 83 | 0.63 | 4.16 |
| (GER412-AA) | (GER411-AA) | 196 | 0.56 | 0.50 | 83 | 0.77 | 1.11 |
| (PHYS101-F)(PHYS130-CC) | (PHYS201-F) | 117 | 0.52 | 0.26 | 82 | 0.61 | 1.39 |
| (PHYS101-F)(PHYS130-DC) | (PHYS201-F) | 172 | 0.66 | 0.38 | 82 | 0.72 | 2.33 |
| (AD104-AA)(AD231-AA) | (AD202-AA) | 84 | 0.52 | 0.73 | 82 | 0.30 | 4.61 |
| (MATH101-F)(MATH102-F)(MATH202-F) | (MATH201-F) | 157 | 0.69 | 0.60 | 80 | 0.55 | 2.54 |
| (AD104-AA)(AD232-AA) | (AD231-AA) | 82 | 0.55 | 0.49 | 80 | 0.91 | 3.00 |
| (CHEM103-BA) | (CHEM109-AA) | 120 | 0.53 | 0.14 | 78 | 0.87 | 1.34 |
| (AD231-AA)(AD220-AA) | (AD341-AA) | 89 | 0.64 | 0.38 | 76 | 0.58 | 7.24 |
| (FR201-AA) | (FR102-AA) | 278 | 0.57 | 0.50 | 76 | 0.42 | 2.13 |
| (PHYS130-DD)(PHYS101-DD) | (PHYS201-F) | 165 | 0.65 | 0.35 | 75 | 0.76 | 6.56 |
| (FR102-AA) | (FR101-AA) | 413 | 0.60 | 0.41 | 72 | 0.71 | 2.19 |
| (MATH101-F)(MATH202-DD) | (MATH201-F) | 155 | 0.52 | 0.47 | 72 | 0.47 | 1.82 |
| (GER302-AA) | (GER301-AA) | 303 | 0.54 | 0.50 | 71 | 0.51 | 1.50 |
| (AD214-AA) | (AD341-AA) | 141 | 0.50 | 0.30 | 70 | 0.60 | 2.16 |
| (PHYS102-DD) | (PHYS201-F) | 417 | 0.54 | 0.35 | 70 | 0.71 | 1.95 |
| (GER202-AA) | (GER102-AA) | 339 | 0.52 | 0.41 | 70 | 0.49 | 2.03 |
| (AD104-AA)(AD220-AA) | (AD341-AA) | 81 | 0.59 | 0.41 | 69 | 0.55 | 5.72 |
| (PHYS101-DC)(PHYS130-DD) | (PHYS201-F) | 126 | 0.58 | 0.31 | 69 | 0.74 | 4.74 |
| (MATH102-DC)(MATH201-F) | (MATH101-F) | 180 | 0.62 | 0.39 | 67 | 0.39 | 2.31 |
| (AD220-AA)(AD232-AA) | (AD341-AA) | 75 | 0.64 | 0.39 | 66 | 0.62 | 7.02 |

Appendix H – Top 45 Rules Having Highest Test Confidence

<u>4 Grade Dataset</u>

| LeftSide | RightSide | S | C | I | TS | TC | L |
|---|---|---|---|---|---|---|---|
| (CHEM104-H) | (CHEM109-H) | 261 | 0.63 | 0.03 | 73 | 1.00 | 1.35 |
| (GER202-H)(GER411-H) | (GER412-H) | 164 | 0.91 | 0.29 | 43 | 1.00 | 1.16 |
| (AD320-H) | (AD231-H) | 327 | 0.61 | 0.36 | 43 | 1.00 | 1.56 |
| (AD104-H)(AD320-H) | (AD231-H) | 222 | 0.69 | 0.40 | 42 | 1.00 | 2.41 |
| (AD353-H) | (AD232-H) | 215 | 0.63 | 0.31 | 40 | 1.00 | 1.80 |
| (GER201-H)(GER411-H) | (GER412-H) | 164 | 0.91 | 0.29 | 39 | 1.00 | 1.27 |
| (GER302-H)(GER411-H) | (GER412-H) | 187 | 0.86 | 0.22 | 39 | 1.00 | 1.01 |
| (GER101-H)(GER411-H) | (GER412-H) | 164 | 0.91 | 0.38 | 38 | 1.00 | 1.25 |
| (GER301-H)(GER302-H)(GER412-H) | (GER411-H) | 150 | 0.82 | 0.24 | 36 | 1.00 | 1.54 |
| (GER102-H)(GER411-H) | (GER412-H) | 159 | 0.88 | 0.32 | 35 | 1.00 | 1.38 |
| (MIS113-H) | (MIS492-H) | 170 | 0.61 | 0.10 | 34 | 1.00 | 1.11 |
| (FLED303-H)(FLED416-H) | (FLED412-H) | 209 | 0.92 | 0.29 | 34 | 1.00 | 1.54 |
| (MIS113-H)(MIS481-H) | (MIS492-H) | 160 | 0.89 | 0.25 | 32 | 1.00 | 1.19 |
| (GER301-H)(GER411-H) | (GER412-H) | 166 | 0.86 | 0.20 | 31 | 1.00 | 1.06 |
| (FLED303-H)(FLED411-H) | (FLED412-H) | 165 | 0.85 | 0.20 | 31 | 1.00 | 1.47 |
| (GER301-H)(GER302-H)(GER411-H) | (GER412-H) | 150 | 0.90 | 0.20 | 29 | 1.00 | 1.54 |
| (FLED303-H)(FLED403-H) | (FLED412-H) | 186 | 0.89 | 0.24 | 28 | 1.00 | 1.61 |
| (AR202-H) | (AR201-H) | 164 | 0.58 | 0.08 | 28 | 1.00 | 1.16 |
| (CHEM363-M) | (CHEM402-H) | 170 | 0.60 | 0.04 | 28 | 1.00 | 1.10 |
| (GER102-H)(GER201-H)(GER202-H) | (GER101-H) | 438 | 0.78 | 0.20 | 25 | 1.00 | 4.90 |
| (MIS463-H)(MIS481-H) | (MIS492-H) | 155 | 0.86 | 0.06 | 23 | 1.00 | 1.05 |
| (FLED303-H)(FLED403-H)(FLED416-H) | (FLED412-H) | 177 | 0.96 | 0.28 | 22 | 1.00 | 2.28 |
| (AD408-H) | (AD232-H) | 261 | 0.53 | 0.24 | 20 | 1.00 | 1.32 |
| (AD104-H)(AD401-H) | (AD232-H) | 240 | 0.57 | 0.21 | 19 | 1.00 | 1.99 |
| (GER102-H)(GER301-H) | (GER101-H) | 329 | 0.72 | 0.16 | 18 | 1.00 | 2.34 |
| (AD104-H)(AD408-H) | (AD232-H) | 178 | 0.64 | 0.25 | 18 | 1.00 | 2.07 |
| (FLED303-H)(FLED308-H) | (FLED412-H) | 154 | 0.76 | 0.16 | 17 | 1.00 | 1.62 |
| (EL205-M)(EL310-M) | (EL202-M) | 202 | 0.64 | 0.07 | 17 | 1.00 | 1.33 |
| (FLED308-H)(FLED403-H) | (FLED416-H) | 188 | 0.89 | 0.32 | 17 | 1.00 | 1.47 |
| (FLED308-H)(FLED403-H)(FLED412-H) | (FLED416-H) | 175 | 0.95 | 0.35 | 16 | 1.00 | 2.07 |
| (GER101-H)(GER102-H)(GER302-H) | (GER202-H) | 230 | 0.80 | 0.34 | 16 | 1.00 | 3.63 |
| (GER102-H)(GER201-H)(GER301-H) | (GER101-H) | 292 | 0.78 | 0.19 | 16 | 1.00 | 4.62 |
| (FLED308-H)(FLED411-H) | (FLED412-H) | 172 | 0.82 | 0.23 | 16 | 1.00 | 1.55 |
| (GER101-H)(GER301-H)(GER302-H) | (GER202-H) | 242 | 0.81 | 0.36 | 15 | 1.00 | 3.06 |
| (GER201-H)(GER301-H)(GER302-H) | (GER202-H) | 289 | 0.81 | 0.32 | 15 | 1.00 | 3.55 |
| (FLED308-H)(FLED411-H)(FLED416-H) | (FLED412-H) | 156 | 0.90 | 0.28 | 15 | 1.00 | 2.09 |
| (GER101-H)(GER201-H)(GER302-H) | (GER202-H) | 232 | 0.81 | 0.35 | 14 | 1.00 | 3.41 |
| (GER102-H)(GER202-H)(GER301-H) | (GER101-H) | 270 | 0.78 | 0.19 | 13 | 1.00 | 4.01 |
| (AD351-H) | (AD232-H) | 301 | 0.57 | 0.30 | 12 | 1.00 | 1.85 |
| (AD351-H) | (AD214-H) | 272 | 0.52 | 0.48 | 12 | 1.00 | 2.39 |
| (GER301-H)(GER411-H)(GER412-H) | (GER302-H) | 150 | 0.90 | 0.28 | 12 | 1.00 | 1.54 |
| (GER301-H)(GER412-H) | (GER302-H) | 182 | 0.88 | 0.29 | 12 | 1.00 | 1.40 |
| (JP201-H) | (JP101-H) | 167 | 0.80 | 0.12 | 11 | 1.00 | 1.25 |
| (EC234-M)(EC331-L) | (EC101-M) | 190 | 0.52 | 0.01 | 11 | 1.00 | 1.72 |
| (AD104-H)(AD351-H) | (AD214-H) | 204 | 0.66 | 0.44 | 10 | 1.00 | 3.46 |

<u>8 Grade Dataset</u>

| LeftSide | RightSide | S | C | I | TS | TC | L |
|---|---|---|---|---|---|---|---|
| (CHEM104-AA) | (CHEM109-AA) | 114 | 0.61 | 0.19 | 40 | 1.00 | 1.83 |
| (AD202-AA)(AD220-AA) | (AD231-AA) | 78 | 0.66 | 0.54 | 31 | 1.00 | 9.61 |
| (GER201-AA)(GER411-AA) | (GER412-AA) | 79 | 0.80 | 0.48 | 23 | 1.00 | 2.31 |
| (AD353-AA) | (AD232-AA) | 97 | 0.51 | 0.52 | 22 | 1.00 | 2.73 |
| (GER302-AA)(GER411-AA) | (GER412-AA) | 88 | 0.76 | 0.44 | 22 | 1.00 | 1.51 |
| (GER411-AA)(GER301-AA) | (GER412-AA) | 82 | 0.82 | 0.43 | 19 | 1.00 | 1.83 |
| (AD353-AA)(AD341-AA) | (AD232-AA) | 76 | 0.59 | 0.55 | 17 | 1.00 | 10.20 |
| (AD320-AA) | (AD231-AA) | 109 | 0.50 | 0.45 | 16 | 1.00 | 1.96 |
| (AD353-AA)(AD341-AA) | (AD231-AA) | 76 | 0.59 | 0.49 | 16 | 1.00 | 8.94 |
| (CHE353-CB) | (CHE492-AA) | 124 | 0.81 | 0.01 | 15 | 1.00 | 1.03 |
| (GER302-AA)(GER101-AA)(GER201-AA) | (GER301-AA) | 109 | 0.75 | 0.42 | 15 | 1.00 | 7.16 |
| (CET150-AA)(CET462-AA) | (CET451-AA) | 82 | 0.90 | 0.41 | 15 | 1.00 | 1.27 |
| (CHE211-BA) | (CHE492-AA) | 77 | 0.75 | 0.04 | 14 | 1.00 | 1.18 |
| (GER202-AA)(GER411-AA) | (GER412-AA) | 82 | 0.86 | 0.53 | 14 | 1.00 | 2.21 |
| (GER102-AA)(GER301-AA) | (GER101-AA) | 159 | 0.64 | 0.24 | 14 | 1.00 | 4.12 |
| (GER201-AA)(GER301-AA) | (GER101-AA) | 162 | 0.58 | 0.24 | 13 | 1.00 | 3.84 |
| (AD401-AA)(AD231-AA) | (AD232-AA) | 79 | 0.53 | 0.52 | 13 | 1.00 | 3.99 |
| (AD232-AA)(AD401-AA) | (AD231-AA) | 79 | 0.57 | 0.46 | 13 | 1.00 | 3.99 |
| (CHE433-BB) | (CHE492-AA) | 103 | 0.94 | 0.04 | 12 | 1.00 | 1.03 |
| (GER102-AA)(GER301-AA)(GER201-AA) | (GER101-AA) | 129 | 0.71 | 0.27 | 12 | 1.00 | 12.02 |
| (AD341-AA)(AD320-AA) | (AD231-AA) | 84 | 0.63 | 0.53 | 12 | 1.00 | 7.14 |
| (AE111-AA)(AE231-AA) | (AE112-AA) | 97 | 0.75 | 0.10 | 11 | 1.00 | 2.01 |
| (CHE334-BB) | (CHE492-AA) | 118 | 0.86 | 0.04 | 11 | 1.00 | 1.07 |
| (GER101-AA)(GER202-AA)(GER302-AA) | (GER301-AA) | 97 | 0.75 | 0.49 | 10 | 1.00 | 5.97 |
| (EE338-AA) | (EE492-AA) | 132 | 0.77 | 0.29 | 10 | 1.00 | 1.69 |
| (AD341-AA)(AD401-AA) | (AD232-AA) | 94 | 0.50 | 0.47 | 10 | 1.00 | 5.45 |
| (GER301-AA)(GER102-AA)(GER202-AA) | (GER101-AA) | 116 | 0.70 | 0.28 | 10 | 1.00 | 10.18 |
| (AD401-AA)(AD341-AA) | (AD231-AA) | 103 | 0.55 | 0.48 | 9 | 1.00 | 5.24 |
| (GER202-AA)(GER201-AA)(GER301-AA) | (GER101-AA) | 122 | 0.64 | 0.26 | 9 | 1.00 | 9.75 |
| (AD353-AA)(AD351-AA) | (AD341-AA) | 76 | 0.78 | 0.48 | 9 | 1.00 | 19.94 |
| (AD351-AA)(AD231-AA) | (AD401-AA) | 79 | 0.59 | 0.61 | 8 | 1.00 | 7.70 |
| (ED382-AA)(ED262-AA) | (ED221-AA) | 91 | 0.55 | 0.49 | 8 | 1.00 | 5.06 |
| (JP201-AA) | (JP102-AA) | 91 | 0.71 | 0.37 | 8 | 1.00 | 1.64 |
| (AD231-AA)(AD351-AA) | (AD232-AA) | 82 | 0.61 | 0.51 | 7 | 1.00 | 6.68 |
| (CHE353-AA) | (CHE492-AA) | 78 | 0.91 | 0.07 | 7 | 1.00 | 1.05 |
| (AD351-AA)(AD232-AA) | (AD231-AA) | 82 | 0.63 | 0.52 | 7 | 1.00 | 6.68 |
| (AD312-AA)(AD341-AA) | (AD442-AA) | 76 | 0.51 | 0.34 | 7 | 1.00 | 5.30 |
| (AD401-AA)(AD202-AA) | (AD231-AA) | 76 | 0.64 | 0.52 | 7 | 1.00 | 6.32 |
| (CHE334-AA) | (CHE492-AA) | 85 | 0.93 | 0.08 | 7 | 1.00 | 1.09 |
| (AD351-AA)(AD232-AA) | (AD442-AA) | 76 | 0.58 | 0.40 | 6 | 1.00 | 7.21 |
| (CHE462-BA) | (CHE492-AA) | 87 | 0.92 | 0.03 | 6 | 1.00 | 1.01 |
| (GER102-AA)(GER201-AA)(GER302-AA) | (GER202-AA) | 120 | 0.73 | 0.56 | 6 | 1.00 | 11.43 |
| (JP201-AA)(JP101-AA) | (JP102-AA) | 78 | 0.80 | 0.41 | 6 | 1.00 | 3.16 |
| (AD442-AA)(AD232-AA) | (AD231-AA) | 86 | 0.55 | 0.55 | 6 | 1.00 | 3.66 |
| (AD312-AA)(AD351-AA) | (AD341-AA) | 76 | 0.68 | 0.41 | 5 | 1.00 | 10.31 |

Appendix I - Rule Creator Codes

DBCon.java

```
package creator;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;

public class DBCon {
	public Connection conn;
	public Statement stmt;
	/**
	 * This Class makes the connection to the database.
	 * It has the simplified functions to make the necessary operations.
	 */
	public DBCon(String source)
	{
		try
		{
			conn =
DriverManager.getConnection("jdbc:odbc:"+source);
			stmt = conn.createStatement();
		}
		catch (SQLException e)
		{
			e.printStackTrace();
		}
	}
	public ResultSet runResult(String query)
	{
		try
		{
			return stmt.executeQuery(query);
		}
		catch (SQLException e)
		{
			e.printStackTrace();
		}
		return null;
	}
	public void run(String query) throws Exception
	{
		stmt.executeUpdate(query);
	}
}
```

DBFunctions.java

```
package creator;

import java.sql.ResultSet;
```

```
import java.util.Vector;

public class DBFunctions {
	/**
	 * This function gets a condition as a parameter and returns the 
list of students' IDs which satisfy that condition.
	 */
	public static Vector<Integer> getListOfStudents(String condition)
	{
		try
		{
			DBCon con = new DBCon(Settings.dataSet);
			try
			{
				Vector<Integer> list = new Vector<Integer>();
				ResultSet rs = con.runResult("select distinct 
OgrenciID from MainTable where "+condition+" order by OgrenciID asc");
				while(rs.next())
				{
					list.add(rs.getInt(1));
				}
				con.conn.close();
				return list;
			}
			catch (Exception e)
			{
				System.out.println(condition +" Caused An 
Error");
				
				e.printStackTrace();
			}
			con.conn.close();
		}
		catch (Exception e)
		{
			e.printStackTrace();
		}
		return null;
	}
	/**
	 * This function gets a condition as a parameter and returns the 
number of students which satisfy that condition.
	 */
	public static int getCountOfStudents(String condition)
	{
		try
		{
			DBCon con = new DBCon(Settings.dataSet);
			try
			{
				ResultSet rs = con.runResult("select 
count(OgrenciID) from MainTable where "+condition);
				if(rs == null)
				{
					con.conn.close();
					return 0;
				}
				rs.next();
```

```
				int count = rs.getInt(1);
				con.conn.close();
				return count;
			}
			catch (Exception e)
			{
				System.out.println(condition +" Caused An 
Error");
				con.conn.close();
				e.printStackTrace();
			}
		}
		catch (Exception e)
		{
			e.printStackTrace();
		}
		return 0;
	}
	/**
	 * This function gets the ID of course and returns the name of the 
course
	 */
	
	public static String getNameOfCourse(int id)
	{
		try
		{
			DBCon con = new DBCon(Settings.dataSet);
			ResultSet rs =  con.runResult("select course from 
courses where ID="+id);
			rs.next();
			String s = rs.getString(1);
			con.conn.close();
			return 	s;
		}
		catch (Exception e)
		{
			e.printStackTrace();
		}
		return "unfound course";
	}
}
```

<u>GeneralFunctions.java</u>

```
package creator;

import java.util.Vector;

public class GeneralFunctions
{
	/**
	 * Gets 2 vectors and finds their intersection.
	 * The elements which are contained in both vectors are added to a 
result vector.
	 * The result vector is returned.
	 */
```

```java
	public static Vector<Integer> combineTwoVectors(Vector<Integer> a, Vector<Integer> b)
	{
		// The Vectors Are Considered As Ordered
		if(a==null || b == null)
			return null;
		Vector<Integer> v = new Vector<Integer>();
		int i = 0;
		int j = 0;
		while(i<a.size() && j<b.size())
		{
			if(a.get(i)<b.get(j))
			{
				i++;
			}
			else if(a.get(i)>b.get(j))
			{
				j++;
			}
			else if(a.get(i).equals(b.get(j)))
			{
				v.add(a.get(i));
				i++;
				j++;
			}
		}
		return v;
	}
	/**
	 * Gets 2 vectors and finds the difference the first one from second one.
	 * The elements which are contained by the first vector but not the second one are added to a result vector.
	 * The result vector is returned.
	 */
	public static Vector<Integer> differTwoVectors(Vector<Integer> a,Vector<Integer> b)
	{
		Vector<Integer> v = new Vector<Integer>();
		for(Integer i : a)
		{
			if(!b.contains(i))
			{
				v.add(i);
			}
		}
		return v;
	}
}
```

<u>Grade.java</u>

```java
package creator;

import java.util.Vector;

/**
```

```
 * The entity class to hold the grades.
 */
public class Grade
{

	private int courseID;
	private int gradeID;

	public Grade(int cID,int gID)
	{
		this.courseID = cID;
		this.gradeID = gID;
	}
	/**
	 * Returns the number of students who have taken this grade.
	 */
	public int countStudentsTakenThisGrade()
	{
		return DBFunctions.getCountOfStudents("CourseID="+courseID+"
and NoteID="+gradeID);
	}
	/**
	 * Returns the number of students who have taken the course which
this grade is about.
	 */
	public int countStudentsTakenThisCourse()
	{
		return DBFunctions.getCountOfStudents("CourseID="+courseID);
	}
	/**
	 * Returns the list of students who have taken this grade.
	 */
	public Vector<Integer> getListOfStudentsTakenThisGrade()
	{
		return DBFunctions.getListOfStudents("CourseID="+courseID+"
and NoteID="+gradeID);
	}
	/**
	 * Returns the list of students who have taken the course which this
grade is about.
	 */
	public Vector<Integer> getListofStudentTakenThisCourse()
	{
		return DBFunctions.getListOfStudents("CourseID="+courseID);
	}
	/**
	 * Returns the number of students who have taken this grade
satisfying given condition.
	 */
	public int countStudentsTakenThisGrade(String rule)
	{
		return DBFunctions.getCountOfStudents("CourseID="+courseID+"
and NoteID="+gradeID+" and "+rule);
	}
	/**
	 * Returns the number of students who have taken the course this
grade is about satisfying given condition.
```

```
	 */
	public int countStudentsTakenThisCourse(String rule)
	{
		return DBFunctions.getCountOfStudents("CourseID="+courseID+"
and "+rule);
	}
	/**
	 * Returns the list of students who have taken this grade satisfying
given condition.
	 */
	public Vector<Integer> getListOfStudentsTakenThisGrade(String rule)
	{
		return DBFunctions.getListOfStudents("CourseID="+courseID+"
and NoteID="+gradeID+" and "+rule);
	}
	/**
	 * Returns the list of students who have taken the course this grade
is about satisfying the given condition.
	 */
	public Vector<Integer> getListofStudentTakenThisCourse(String rule)
	{
		return DBFunctions.getListOfStudents("CourseID="+courseID+"
and "+rule);
	}
	public boolean equals(Object arg0) {
		Grade g = (Grade) arg0;
		return g.courseID == this.courseID && g.gradeID ==
this.gradeID;
	}
	public int getCourseID() {
		return courseID;
	}
	public void setCourseID(int courseID) {
		this.courseID = courseID;
	}
	public int getGradeID() {
		return gradeID;
	}
	public void setGradeID(int gradeID) {
		this.gradeID = gradeID;
	}
	public String toString()
	{
		return DBFunctions.getNameOfCourse(this.courseID)+"-
"+this.gradeID;
	}
}
```

<u>Set.java</u>

```
package creator;

import java.util.Vector;

public class Set {
	private int id;
	private static int idCounter;
```

```
	public Vector<Grade> list= new Vector<Grade>();
	Vector<Rule> allRules = new Vector<Rule>();
	int count=0;
	public Set()
	{
		this.id=idCounter++;
	}
	public Set(Grade g)
	{
		list.add(g);
		this.id=idCounter++;
	}
	public void addGrade(Grade g)
	{
		list.add(g);
	}
	/**
	 * Returns the count of the students who have taken all the courses 
which the grades in the set are about.
	 */
	public int countCourses()
	{
		Vector<Integer> temp = 
list.get(0).getListofStudentTakenThisCourse(Settings.training);
		for(Grade g: this.list)
		{
			temp=GeneralFunctions.combineTwoVectors(temp, 
g.getListofStudentTakenThisCourse());
		}
		return temp.size();
	}
	/**
	 * Returns the list of courses which the grades in the set are 
about.
	 */
	public Vector<Integer> getCoursesList(String s)
	{
		Vector<Integer> temp=null;
		if(list.size()>0)
		{
			temp = list.get(0).getListofStudentTakenThisCourse(s);
			for(Grade g: this.list)
			{
				temp=GeneralFunctions.combineTwoVectors(temp, 
g.getListofStudentTakenThisCourse());
			}
		}
		return temp;
	}
	/**
	 * Returns the count of the students who have taken all the grades 
in the set.
	 */
	public void countList(String s)
	{
		try
		{
```

```
			if(this.list.size()==1)
			{
				this.count = 
list.get(0).countStudentsTakenThisGrade(s);
				return;
			}
			Vector<Integer> temp = 
list.get(0).getListOfStudentsTakenThisGrade(s);
			if(temp==null)
				return;
			for(Grade g:this.list)
			{
	
	temp=GeneralFunctions.combineTwoVectors(temp,g.getListOfStudentsTake
nThisGrade(s));
			}
			if(temp == null)
				this.count =0;
			else
				this.count= temp.size();
		}
		catch (Exception e)
		{
			e.printStackTrace();
		}
	}
	/**
	 * Returns the list of the students who have taken all the grades in 
the set.
	 */
	public Vector<Integer> getList(String s)
	{
		Vector<Integer> temp=null;
		if(list.size()>0)
		{
			temp = list.get(0).getListOfStudentsTakenThisGrade(s);
			for(Grade g:this.list)
			{
	
	temp=GeneralFunctions.combineTwoVectors(temp,g.getListOfStudentsTake
nThisGrade(s));
			}
			this.count= temp.size();
		}
		return temp;
	}
	public boolean equals(Object obj) {
		Set s = (Set) obj;
		if(this.list.size() != s.list.size())
			return false;
		for(Grade g:this.list)
		{
			if(!s.list.contains(g))
				return false;
		}
		return true;
	}
```

```java
	/**
	 * Creates all possible rules in this set.
	 */
	public void createAllRules()
	{
		for(Grade g:this.list)
		{
			this.createRule(new Set(g));
		}
	}
	/**
	 * Gets a subset of this set as a parameter.
	 * Sets this subset as the LeftSide of the rule.
	 * Collects remaining sets in another sets and sets it as the
rightside of the rule.
	 * Works recursively to create all possible sets.
	 */
	public void createRule(Set s)
	{
		if(s.list.size()==this.list.size())
		{
			return;
		}
		for(Grade g:this.list)
		{
			Set otherSide = new Set();
			if(s.list.contains(g))
				continue;
			otherSide.list.add(g);
			Rule r = new Rule(s, otherSide, this.count);
			if(this.allRules.contains(r))
				continue;
			if(r.getImportance()>0 &&
r.getConfidence()>Settings.minConf && r.lift>1)
				r.insert();
		}
		for(Grade g:this.list)
		{
			if(s.list.contains(g))
				continue;
			Set temp = (Set) s.clone();
			temp.list.add(g);
			this.createRule(temp);
		}
	}

	protected Object clone()
	{
		Set s = new Set();
		for(Grade g:this.list)
			s.addGrade(g);
		return s;
	}
	public String toString()
	{
		String s="";
```

```java
		for(Grade g:list)
		{
			s+="("+g+")";
		}
		return s;
	}
	public int getId() {
		return id;
	}
	public void setId(int id) {
		this.id = id;
	}
}
```

Rule.java

```java
package creator;

import java.sql.ResultSet;
import java.util.Vector;
/**
 * The entity class to hold the Rules .
 */
public class Rule {
	int id;
	Set right;
	Set left;
	double confidence;
	int total;
	double importance;
	double testConf;
	int testSupport;
	double lift;

	public Rule(Set l,Set r,int total)
	{
		this.right=r;
		this.left =l;
		this.total = total;
		this.calculateAll();
	}

	public void calculateAll()
	{
		if(total!=0)
		{
			this.left.countList(Settings.training);
			this.total = total;
			this.confidence=((double)total/this.left.count);
			if(this.confidence<Settings.minConf)
				return;
			this.importance = this.calculateImportance();
			if(this.importance<0)
				return;
			this.calculateLift();
			if(this.lift<1)
				return;
```

```
                this.calculateTestConfidence();

            }
        }
        /**
         * Calculates the Importance value for the rule.
         * The formula for the Importance is:
         * Confidence/(Support(LeftSide & notRightSide)/Support(RightSide))
         */
        public double calculateImportance()
        {
            try
            {
                double importance = confidence;
                Vector<Integer> leftSideNot =
GeneralFunctions.differTwoVectors(this.left.getCoursesList(Settings.traini
ng),this.left.getList(Settings.training));
                Vector<Integer> rightSide =
this.right.getList(Settings.training);
                Vector<Integer> a =
GeneralFunctions.combineTwoVectors(leftSideNot, rightSide);
                double x = (double)a.size()/leftSideNot.size();
                importance/=x;
                return Math.log10(importance);
            }
            catch (Exception e)
            {
                e.printStackTrace();
            }
            return 0;
        }
        /**
         * Calculates the TestConfidence for the Rule.
         * The Formula for TestConfidence is:
         * Support(Right side After 2008)/Support(Right side course after
2008 & Left side )
         */
        public double calculateTestConfidence()
        {
            System.out.println("Testing "+this);
            Vector<Integer> temp=this.left.getList("1=1");
            Vector<Integer> temp1 =
this.right.getCoursesList(Settings.test);
            Vector<Integer> temp2 = this.right.getList(Settings.test);
            Vector<Integer> temp3=
GeneralFunctions.combineTwoVectors(temp, temp1);
            Vector<Integer> temp4=
GeneralFunctions.combineTwoVectors(temp, temp2);
            if(temp3.size()==0)
            {
                this.testConf = -1;
                return -1;
            }
            this.testConf = (double) temp4.size()/temp3.size();
            this.testSupport=temp3.size();
            System.out.println("tested  "+this);
            return testConf;
```

```
	}

	/**
	 * Calculates the lift of the rule.
	 * Formula for the lift is:
	 * Support(Left Side & Right Side)/(Support(Left Side)*Support(Right
Side)).
	 */
	public void calculateLift()
	{
		double lift=1;
		Vector<Integer>
studentsTakenCourses=this.left.list.get(0).getListofStudentTakenThisCourse
();
		for(Grade g: left.list)
		{
			lift *=
(double)g.countStudentsTakenThisGrade()/g.countStudentsTakenThisCourse();
		}
		for(Grade g: right.list)
		{
			lift *=
(double)g.countStudentsTakenThisGrade()/g.countStudentsTakenThisCourse();
		}
		for(Grade g: left.list)
		{
			studentsTakenCourses =
GeneralFunctions.combineTwoVectors(studentsTakenCourses,
g.getListofStudentTakenThisCourse());
		}
		for(Grade g: right.list)
		{
			studentsTakenCourses =
GeneralFunctions.combineTwoVectors(studentsTakenCourses,
g.getListofStudentTakenThisCourse());
		}
		this.lift = ((double)total/studentsTakenCourses.size())/lift;

	}

	/**
	 * Updates the Rules in the Database as the test values are
calculated.
	 */
	public void updateRuleInDB()
	{
		try
		{
			DBCon con = new DBCon(Settings.dataSet);
			con.run("update rules set
testConfidence="+this.testConf+" where ID="+this.id);
			con.run("update rules set
testSupport="+this.testSupport+" where ID="+this.id);
			con.conn.close();
		}
		catch (Exception e)
		{
```

```
                e.printStackTrace();
            }
        }
        /**
         * Inserts the rules into the database.
         */
        public void insert()
        {
            try
            {
                DBCon con = new  DBCon(Settings.dataSet);
                ResultSet rs = con.runResult("select max(ID) from
Rules");
                rs.next();
                int newID = rs.getInt(1)+1;
                con.run("insert into
rules(ID,confidence,Importance,Support,Lift,testConfidence,testsupport)
values
("+newID+","+confidence+","+importance+","+total+","+lift+","+this.testCon
f+","+testSupport+")");
                for(Grade g:left.list)
                {
                    con.run("insert into ruleParts
(RuleID,courseID,GradeID,side) values
("+newID+","+g.getCourseID()+","+g.getGradeID()+",yes)");
                }
                for(Grade g:right.list)
                {
                    con.run("insert into ruleParts
(RuleID,courseID,GradeID,side) values
("+newID+","+g.getCourseID()+","+g.getGradeID()+",no)");
                }
                System.out.println(this);
            }
            catch (Exception e)
            {
                e.printStackTrace();
            }
        }
        public boolean equals(Object arg0)
        {
            Rule r = (Rule) arg0;
            if(this.right.list.size()!=r.right.list.size())
                return false;
            if(this.left.list.size()!=r.left.list.size())
                return false;
            for(Grade g : this.right.list)
            {
                if(!r.right.list.contains(g))
                    return false;
            }
            for(Grade g : this.left.list)
            {
                if(!r.left.list.contains(g))
                    return false;
            }
            return true;
```

```
	}
	public Set getRight() {
		return right;
	}
	public void setRight(Set right) {
		this.right = right;
	}
	public Set getLeft() {
		return left;
	}
	public double getConfidence() {
		return confidence;
	}
	public void setConfidence(double confidence) {
		this.confidence = confidence;
	}
	public double getImportance() {
		return importance;
	}
	public void setImportance(double importance) {
		this.importance = importance;
	}
	public double getTestConf() {
		return testConf;
	}
	public void setTestConf(double testConf) {
		this.testConf = testConf;
	}
	public void setLeft(Set left) {
		this.left = left;
	}
	public String toString() {
		//return ""+this.left+"--->"+this.right+"\t
Sup="+this.total+"\tC="+this.confidence+"\t I="+this.importance+"\t TS
="+this.testSupport+"\t TC="+this.testConf+"\t
Lift="+String.format("%.5f", this.lift);
		return
String.format("%s\t%s\t%d\t%.2f\t%.2f\t%d\t%.2f\t%.2f",
this.left,this.right,this.total,this.confidence,this.importance,this.testS
upport,this.testConf,this.lift);
	}
	/**
	 * Gets a Department Code as a parameter.
	 * Returns if there is any grade in this rule from that department.
	 */
	public boolean contains(String dept)
	{
		for(Grade g:left.list)
		{
			if(!g.toString().startsWith(dept))
				return false;
		}
		for(Grade g:right.list)
		{
			if(!g.toString().startsWith(dept))
				return false;
		}
```

```
                return true;
        }
}
```

Settings.java

```
package creator;

public class Settings {
        /**
         * The General Variables to be used in whole software.
         */
        public static String training = "Term<6024";
        public static String test = "Term>=6024";
        public static int support =75;
        public static double minConf=.187;
        public static String dataSet = "TezDB";
        public static String[]
gradesN={"AA","BA","BB","CB","CC","DC","DD","F"};
        public static String[] gradesL= {"HIGH(AA-BA)","MID(BB-CC)","LOW(DC-
DD)","FAIL"};
        public static String[] grades;

        public static void setNormal()
        {
                dataSet = "tezDB";
                minConf = .187;
                grades = gradesN;
                support = 75;
        }
        public static void setLessGrade()
        {
                dataSet = "LessDB";
                minConf = .388;
                grades = gradesL;
                support = 150;
        }
}
```

Driver.java

```
package creator;

import java.sql.ResultSet;
import java.util.Vector;

public class Driver {
        // listOfSets contains all the sets created by the algorithm.
        Vector<Set>[] listOfSets=new Vector[10];

        public Driver()
        {
                listOfSets[1]=new Vector<Set>();
        }

        // Creates all possible sets
        public void createAllSets()
```

```
	{
		for(int i=2;i<10;i++)
		{
			createAllSetsLevel(i);
		}
	}

	// Gets the Set Size as a parameter and creates all the sets
	public void createAllSetsLevel(int n)
	{
		System.out.println("creating sets #"+n);
		n--;
		listOfSets[n+1]=new Vector<Set>();
		Vector<Set> templist = (Vector<Set>) listOfSets[n].clone();
		for(int i =0;i<listOfSets[n].size()-1;i++)
		{
			boolean check = false;
			Set s1 = listOfSets[n].get(i);
			for(int j=i+1;j<templist.size();j++)
			{
				Set s2 = templist.get(j);
				System.out.println("Combining Sets : " + s1 +"--
-"+ s2);

				Set s = combineSets(s1, s2);
				if(s!=null)
				{
					if(listOfSets[n+1].contains(s))
						continue;
					s.countList(Settings.training);
					if(s.count>=Settings.support)
					{
						listOfSets[n+1].add(s);
						System.out.println(s);
						s.createAllRules();
					}
					check=true;
				}
				// if the sets are not suitable for combining
the loop gets to the next set.
				if(s1.list.get(0)!=s2.list.get(0) && check &&
n>1)

					break;
			}
		}
	}
	// Checks all created sets and creates all the rules for them.
	public void createAllRules()
	{
		for(int i=0;i<10;i++)
		{
			for(Set s: listOfSets[i])
			{
				s.createAllRules();
			}
		}
	}
	/**
```

```
         * Gets two sets as parameters and returns the combination of those
two.
         * If they are not suitable for combining, the function returns null
         */
        public Set combineSets(Set s1,Set s2)
        {
                if(s1.getId()>s2.getId())
                        return null;
                Set s = new Set();
                if(s1.list.size() != s2.list.size())
                        return null;
                int count=s1.list.size();
                for(int i =0 ;i<count-1;i++)
                {
                        if(s1.list.get(i).equals(s2.list.get(i)))
                        {
                                s.addGrade(s1.list.get(i));
                                continue;
                        }
                        return null;
                }
                if(s1.list.get(count-1).getCourseID()==(s2.list.get(count-
1).getCourseID()))
                        return null;

                if(s1.list.get(count-1).equals(s2.list.get(count-1)))
                        return null;
                else
                {
                        s.addGrade(s1.list.get(count-1));
                        s.addGrade(s2.list.get(count-1));
                        return s;
                }
        }

        public static void main(String[] args) {
                Settings.setLessGrade();
                runDept("%", 0);
                Settings.setNormal();
                runDept("%", 0);
        }
        /**
         * Gets Department code and the ID of the grade to start as the
parameter.
         * Runs the algorithm on the given department starting from given
ID.
         */
        public static void runDept(String dept,int number)
        {
                try {
                        DBCon ccc = new DBCon(Settings.dataSet);
                        ccc.conn.close();
                        Driver d=new Driver();
                        Vector<Grade> grades = new Vector<Grade>();
                        DBCon con = new DBCon((Settings.dataSet));
                        System.out.print("Reading Grades...");
```

```java
                   ResultSet rs = con.runResult("select mt.courseID
,mt.NoteID from Maintable mt,Courses c where coursedept like '"+dept+"'
and c.ID>="+number+"and c.ID=mt.courseID and mt.term<6024 and noteID<=8
group by mt.courseID, mt.NoteID having count(*)>"+Settings.support );
                   while(rs.next())
                   {
                         Grade g = new Grade(rs.getInt(1),rs.getInt(2));
                         grades.add(g);
                   }
                   System.out.println("DONE");
                   con.conn.close();
                   System.out.println("Creating Sets...");
                   for(Grade g:grades)
                   {
                         System.out.println(g);
                         try
                         {
                               Set s =new Set(g);
                               s.countList(Settings.training);
                               if(s.count>Settings.support)
                               {
                                     d.listOfSets[1].add(s);
                               }
                         }
                         catch (Exception e)
                         {
                               e.printStackTrace();
                         }
                   }
                   System.out.println("DONE");
                   d.createAllSets();
             }
             catch (Exception e)
             {
                   e.printStackTrace();
             }

       }
}
```

## Appendix J – User Tool Codes

### Suggestion.java

```java
package user;

import creator.Set;

public class Suggestion {
       /**
        * Entity class for holding suggestions.
        */

       Set g;
       double probability;
       String reason;
```

```
	public Suggestion(Set g,double p)
	{
		probability=p;
		this.g=g;
	}
	public String toString() {
		return g+" "+probability+" - " +reason;
	}
}
```

UserFunctions.java

```
package user;

import java.util.Vector;
import creator.*;

public class UserFunctions {
	public static Vector<Grade> gradeList = new Vector<Grade>();
	public static String[] suggestions;

	public static void addGrade(Grade g)
	{
		gradeList.add(g);
	}
	public static String[] getAllGradesInArray()
	{
		String[] a = new String[gradeList.size()];
		for(int i=0;i<gradeList.size();i++)
			a[i]=gradeList.get(i).toString();
		return a;
	}

	public static void findSuggestions()
	{
		suggestions = new String[20];
		if(gradeList.size()==0)
		{
			Suggestion[] ss = UserDB.findBaseSuggestion();
			for(int i =0;i<ss.length;i++)
			{
				suggestions[i]=ss[i].toString();
			}
		}
		else
		{
			Suggestion[] ss = UserDB.findBestRules(gradeList);
			for(int i =0;i<ss.length;i++)
			{
				if(ss[i]==null)
					break;
				suggestions[i]=ss[i].toString();
			}
		}
	}
}
```

UserUI.java

```
package user;

import java.awt.event.ActionEvent;
import java.awt.event.ActionListener;

import javax.swing.JButton;
import javax.swing.JFrame;
import javax.swing.JList;
import javax.swing.JScrollPane;
import javax.swing.ListSelectionModel;
import javax.swing.event.ListSelectionEvent;
import javax.swing.event.ListSelectionListener;

import creator.*;
public class UserUI extends JFrame{
	private JScrollPane deptPane;
	private JList<String> deptList;
	private JScrollPane coursePane;
	private JList<String> courseList;
	private JScrollPane gradePane;
	private JList<String> gradeList;
	private JList<String> takenCoursesList;
	private JScrollPane takenCoursesPane;
	private JButton gradeAddButton;
	private JButton gradeRemoveButton;
	private JList<String> suggestCoursesList;
	private JScrollPane suggestCoursesPane;
	private JButton suggestButton;

	public UserUI()
	{
		UserFunctions.findSuggestions();
		this.setDefaultCloseOperation(EXIT_ON_CLOSE);
		this.setLayout(null);
		this.setSize(500, 500);
		deptList=new JList<String>();
		deptList.setListData(UserDB.getDepartmentsArray());

	deptList.setSelectionMode(ListSelectionModel.SINGLE_SELECTION);
		deptPane = new JScrollPane(deptList);
		deptPane.setSize(100, 100);
		deptPane.setLocation(10, 10);
		deptList.addListSelectionListener(new ListSelectionListener()
{
			public void valueChanged(ListSelectionEvent e) {

	courseList.setListData(UserDB.getCoursesByDept(deptList.getSelectedV
alue()));
			}
		});
		courseList = new JList<String>();
		String[] tempString={"Select Dept"};
		courseList.setListData(tempString);

	courseList.setSelectionMode(ListSelectionModel.SINGLE_SELECTION);
```

```
		coursePane = new JScrollPane(courseList);
		coursePane.setSize(100,100);
		coursePane.setLocation(120, 10);
		gradeList = new JList<String>();
		gradeList.setListData(Settings.grades);
		
	gradeList.setSelectionMode(ListSelectionModel.SINGLE_SELECTION);
		gradePane = new JScrollPane(gradeList);
		gradePane.setSize(100,100);
		gradePane.setLocation(230, 10);
		suggestCoursesList = new JList<String>();
		suggestCoursesList.setListData(UserFunctions.suggestions);
		
	suggestCoursesList.setSelectionMode(ListSelectionModel.SINGLE_SELECT
ION);
		suggestCoursesPane = new JScrollPane(suggestCoursesList);
		suggestCoursesPane.setSize(320,150);
		suggestCoursesPane.setLocation(10, 320);
		takenCoursesList = new JList<String>();
		String[] tempString1={"No Courses Taken"};
		takenCoursesList.setListData(tempString1);
		
	takenCoursesList.setSelectionMode(ListSelectionModel.SINGLE_SELECTIO
N);
		takenCoursesPane = new JScrollPane(takenCoursesList);
		takenCoursesPane.setSize(320,150);
		takenCoursesPane.setLocation(10, 150);
		gradeAddButton = new JButton("Add Grade");
		gradeAddButton.setSize(150,20);
		gradeAddButton.setLocation(10,120);
		gradeAddButton.addActionListener(new ActionListener() {
			public void actionPerformed(ActionEvent arg0) {
				Grade g = new
Grade(UserDB.findCourseID(deptList.getSelectedValue()+courseList.getSelect
edValue()),UserDB.findGradeID(gradeList.getSelectedValue()));
				UserFunctions.addGrade(g);
				refreshTakenCourses();
			}
		});
		gradeRemoveButton = new JButton("Remove Grade");
		gradeRemoveButton.setSize(150,20);
		gradeRemoveButton.setLocation(180,120);
		gradeRemoveButton.addActionListener(new ActionListener() {
			public void actionPerformed(ActionEvent e) {
				
	UserFunctions.gradeList.remove(takenCoursesList.getSelectedIndex());
				refreshTakenCourses();
			}
		});
		suggestButton = new JButton("Make Suggestions");
		suggestButton.setSize(320,20);
		suggestButton.setLocation(10,480);
		suggestButton.addActionListener(new ActionListener() {
			public void actionPerformed(ActionEvent arg0) {
				refreshSuggestions();
			}
		});
```

```
            this.add(deptPane);
            this.add(coursePane);
            this.add(gradePane);
            this.add(takenCoursesPane);
            this.add(gradeAddButton);
            this.add(gradeRemoveButton);
            this.add(suggestCoursesPane);
            this.add(suggestButton);
            this.setTitle("Course Selection DSS");
      }
      public void refreshTakenCourses()
      {
      
      this.takenCoursesList.setListData(UserFunctions.getAllGradesInArray(
));
      }
      public void refreshSuggestions()
      {
            UserFunctions.findSuggestions();
      
      this.suggestCoursesList.setListData(UserFunctions.suggestions);
      }
}
```

UserMain.java

```
package user;

import creator.Settings;

public class UserMain {
      public static void main(String[] args)
      {
            Settings.setNormal();
      //    Settings.setLessGrade();
            UserUI ui = new UserUI();
            ui.setVisible(true);
      }
}
```

REFERENCES

Agarwal, R. & Srikant, R. (1994). Fast algorithms for mining association rules in large databases. Proceedings of the 20th International Conference on Very Large Data Bases, VLDB, Santiago, Chile, 487-499.

Agarwal, R., Imieliński, T. & Swami, A.(1993). Mining Association Rules Between Sets of Items in Large Databases, ACM SIGMOD Record, 22, 2.

Atkinson, A. B. (1970). On the measurement of inequality. Journal Of Economic Theory 2, 244-263.

Berka, P. & Bruha, I. (1998). Discretization and grouping: Preprocessing steps for data mining, PKDD, 239-245.

Dass, R. (2006). Data mining in banking and finance: A note for bankers. Technical Note, Indian Institute of Management, Ahmedabad. Note No: CISG88.

Ensel, C. (2001). A scalable approach to automated service dependency modeling in heterogeneous environments. Enterprise Distributed Object Computing Conference, 2001. EDOC '01. Proceedings. Fifth IEEE International, 128 – 139.

Fayyad, U. , Piatetsky-Shapiro, G. & Smyth, P. (1996). From data mining to knowledge discovery: An overview. Advances in Knowledge Discovery and Data Mining, MIT Press, Cambridge, Mass., 1-36.

Geng, L. & Hamilton, H. J. (2006). Interestingness measures for data mining: A survey. ACM Computing Surveys (CSUR), 38, 3, Art. No: 9. doi:10.1145/1132960.1132963

Guvenc, E. (2001). Data Mining and Knowledge Management in Higher Education. Retrieved from Bogazici University Library.

Han, J., Kamber, M. and Pei,J. (2012), Data mining: Concepts and techniques. London : Morgan Kaufmann

Phyu, T.N. (2009). Survey of Classification Techniques in Data Mining. Proceedings of the International MultiConference of Engineers and Computer Scientists, Hong Kong, 1.

Roiger, J.R., Azarod, C. and Sant, R.R.(1997).A majority rules approach to data mining. IASTED International Conference on Intelligent Information Systems '97,100.

Seidman, C.(2001). Data Mining with Microsoft SQL Server 2000 : Technical Reference.Microsoft Press.

Shaw, M.J., Subramaniam, C., Tan,G.W. & Welge,M.E. (2001). Knowledge management and data mining for marketing. Decision Support Systems, 31, 127–137