Fall 2015

Pattern-Aided Regression Modelling and Prediction Model Analysis

Naresh Avva
San Jose State University

Pattern-Aided Regression Modelling and Prediction Model Analysis

A Project
Presented to
The Faculty of the Department of Computer Science
San Jose State University

In Partial Fulfillment
of the Requirements for the Degree
Master of Science

by
Naresh Avva
Dec 2015

The Designated Project Committee Approves the Project Titled

Pattern-Aided Regression Modelling and Prediction Model Analysis

by
Naresh Avva

APPROVED FOR THE DEPARTMENT OF COMPUTER SCIENCE

SAN JOSE STATE UNIVERSITY

Dec 2015

Dr. Tsau Young Lin	Department of Computer Science
Dr. Robert Chun	Department of Computer Science
Mr. Prasad Kopanati	Advisor at ManyShip Inc.

ABSTRACT

Pattern-Aided Regression Modelling and Prediction Model Analysis

by Naresh Avva

In this research, we develop an application for generating a pattern aided regression (PXR) model, a new type of regression model designed to represent accurate and interpretable prediction model. Our goal is to generate a PXR model using Contrast Pattern Aided Regression (CPXR) method and compare it with the multiple linear regression method. The PXR models built by CPXR are very accurate in general, often outperforming state-of-the-art regression methods by big margins. CPXR is especially effective for high-dimensional data. We use pruning to improve the classification accuracy and to remove outliers from the dataset. We provide implementation details and give experimental results. Finally, we show that the system is practical and better in comparison to other available methods.

ACKNOWLEDGEMENTS

First and foremost, I would like to take this opportunity to thank my project advisor, Dr. Tsau Young Lin, for his constant guidance and trust on me. It wouldn't have been possible without his contribution throughout the project.

I would also like to thank my committee members Dr. Robert Chun and Mr. Prasad Kopanati, for their invaluable advices and crucial comments during project development.

Furthermore, I would like to thank my parents for always being there during my master's program. Finally, I would like to thank all my friends for their support throughout the completion of this project.

Table of Contents

LIST OF TABLES

LIST OF FIGURES

ACRONYMS

Table 1: Acronyms [1]

Symbol	Meaning
arr	average residual reduction
BART	Bayesian Additive Regression Trees
CPXR	Contrast Pattern Aided Regression
CPXR(LL)	CPXR using LR to build baseline models and LR to build local regression models
CPXR(LP)	CPXR using LR to build baseline models and PLR to build local regression models
EC	equivalence class
f	regression function/model
f_{PM}	regression function of PM
GBM	Gradient Boosting
LE	set of large error instances
LR	Linear Regression Algorithm
MG	minimal generator
mds	matching data set
P	pattern
PLR	Piecewise Linear Regression
PM	a PXR regression model
$PM(\{P_1, ..., P_k\}, f_d)$	$((P_1, f_{P_1}, \mathrm{arr}(P_1)), ..., (P_k, f_{P_k}, \mathrm{arr}(P_k)), f_d)$
PXR	Pattern Aided Regression
RMSE	root mean square error
R^2	R squared
SE	set of small error instances
SVR	Support Vector Regression
trr	total residual reduction

CHAPTER 1

Introduction

The contrast pattern aided regression (CPXR) method is a novel, robust, and powerful regression-based method for building prediction models [1]. CPXR provides high accuracy and works with varied predictor-response relationships [1].

CPXR generated prediction models are also representable as compared to artificial neural network models [1]. The results of many experiments conducted suggested that models developed by CPXR method are more accurate than others [1]. CPXR method gave improved performance as compared to other classifiers when applied in other areas of research [1].

The key point of CPXR is to utilize a pattern, in association of certain conditions on a limited number of predictor variables, as logical characterization of a subgroup of data, and a local regression model (corresponded to pattern) as a behavioral characterization of the predictor-response relationship for data instances of that subgroup of data [1].

CPXR can combine a pattern and a local regression model to show a distinct predictor-response relationship for subgroup of data and that's what makes CPXR powerful method [1]. CPXR's ability to choose a profoundly collaborative series of limited number of patterns to augment the entire collective prediction accuracy is one the prime reasons as well to outplay a lot of other methods [1].

The CPXR algorithm generates a pattern aided logistic regression model represented by certain patterns and certain related logistic regression models [1]. CPXR generated prediction models are clear and simple to understand. CPXR can be used in different set of areas such as clinical applications because of its capability to efficiently control data with varied predictor-response relationships [1].

The application designed for this research paper implements the functionality of CPXR by making use of statistical formulas such Root Mean Square Error, Mean Calculation, etc. The application classifies the patterns into two parts such as Large Errors and Small Errors, later mining the patterns in Large Errors group to fulfil the purpose of searching contrast patterns in Large Errors group.

CHAPTER 2

Background

This chapter consists of introduction to pattern recognition, pattern matching, data mining, and machine learning.

2.1 Pattern Recognition

Pattern Recognition comes under machine learning; it focuses on recognizing the patterns by monitoring the uniformity of data [2]. There are two ways to train Pattern recognition systems, the first one is by providing labeled “training” data (supervised learning and the other case arises when there is no labelled data available, then by using different algorithms one can determine already undiscovered patterns (unsupervised learning) [2].

2.2 Pattern Matching

Pattern matching is the methodology which follows the principle of verifying the already provided series of tokens for existence of components of some pattern [3]. In Pattern matching the match has to be identical, unlike pattern recognition [3].

2.3 Data Mining

In big datasets with millions of instances, to anticipate the results; data mining is the process used to determine irregularities, patterns and correlations [4]. By making use of several other techniques, one can utilize this useful information to boost revenues, cut costs, enhance customer relationships, lower risks and more [4].

2.4 Machine Learning

Machine learning is branch of computer science that emerged from the research of pattern recognition and computational learning theory in artificial intelligence [5]. There are advanced algorithms being developed such as to learn from and to do predictions on the dataset [5]. The two main areas in machine learning are supervised learning and unsupervised learning [5].

CHAPTER 3

Pattern-Aided Regression Modelling and Predictive Modelling Analysis

CPXR begins by developing a standard regression model by training database DT utilizing multiple linear regression technique [1]. Then, patterns are categorized on the errors of equivalent standard regression model, and aggregate error is computed [1]. A subjective value of 45 percent of aggregate error is selected to determine the cutting point, which partitions the training database DT into two collections: LE (Large Errors) and SE (Small Errors) [1]. LE consists of those patterns whose aggregate error is higher than 45 percent of the aggregate error [1]. Note that 45 percent is a perfect value established by evaluating more than 50 various databases in diverse research fields [1].

An entropy-based binning approach is utilized for distinct input variables and define elements [1]. Then CPXR analyzes entire set of patterns of LE category, because those patterns are highly repeated in LE than in SE, they are fairly to capture subgroups of data where standard regression model produces high prediction errors [1]. Certain filters are utilized to discard those patterns, which are highly identical to others [1]. For each existing contrast pattern then a local multiple linear regression model is developed [1]. At this step, several patterns and local multiple linear regression models, which fail to enhance the accuracy of predictions are discarded [1].

CPXR then employ's a double (nested) loop to find an optimal pattern [1]. For this objective, we change a pattern by another one in pattern set to lower the errors in each iteration [1].

3.1 Algorithm

Input: (1) training data $D = \{(x_i, y_i) | 1 \leq i \leq n\}$
(2) a prediction model f
(3) a number ρ to partition D into LE, SE
(4) a $minSup$ threshold on contrast patterns
Output: A PXR model

Let $r_1, ..., r_n$ denote f's residuals on $x_1, ..., x_n$;
Determine κ to minimize $|\rho - \frac{\sum_{r_i > \kappa} |r_i|}{\sum |r_i|}|$;
Let $LE = \{x_i \mid r_i > \kappa\}$, $SE = D - LE$;
Discretize each numerical variable using entropy based binning w.r.t. the LE and SE classes;
Extract all contrast patterns for $minSup$ in the LE class, and select just one pattern having shortest length from each equivalence class for inclusion in the CPS set of contrast patterns, and apply other filters (see Section 5.3) to remove patterns of little value from CPS;
For each $P \in CPS$, build the local regression model f_P for data in mds(P);
Let $PS = \{P_0\}$, where P_0 is the pattern P in CPS with highest arr;
repeat
  // add a pattern to PS
  Let P be the pattern in $CPS - PS$ that maximizes $\Delta(f_{PM(PS \cup \{P\}, f)}, f)$;
  // $\Delta(f', f)$ denotes $\frac{RMSE(f) - RMSE(f')}{RMSE(f)}$
  Let $PS^o = PS$;
  if $\Delta(f_{PM(PS \cup \{P\}, f)}, f) > 0$ **then**
    Let $PS = PS^o \cup \{P\}$;
    Call *IteratImp(CPS,PS)* to improve PS;
  end
until $\Delta(f_{PM(PS, f)}, f) - \Delta(f_{PM(PS^o, f)}, f) < 0.01$;
Let f_d be the regression model trained from $D - \cup_{P \in PS}$mds(P);
Return $PM(PS, f_d)$;

Figure 1. CPXR Algorithm [1]

Let $impval = 1$;
repeat
  for *each* $P \in PS$ **do**
    Let Q_P be a pattern P' in $CPS - PS$ maximizing $imp(PS, P, P')$;
  end
  Let P_- be a pattern $P \in PS$ maximizing $imp(PS, P, Q_P)$;
  Let $impval = imp(PS, P_-, Q_{P_-})$;
  if $impval > 0$ **then** $PS = PS - \{P_-\} \cup \{Q_{P_-}\}$;
until $impval < 0.001$;

Figure 2. The IteratImp(CPS, PS) Function [1]

3.2 Flow Diagram

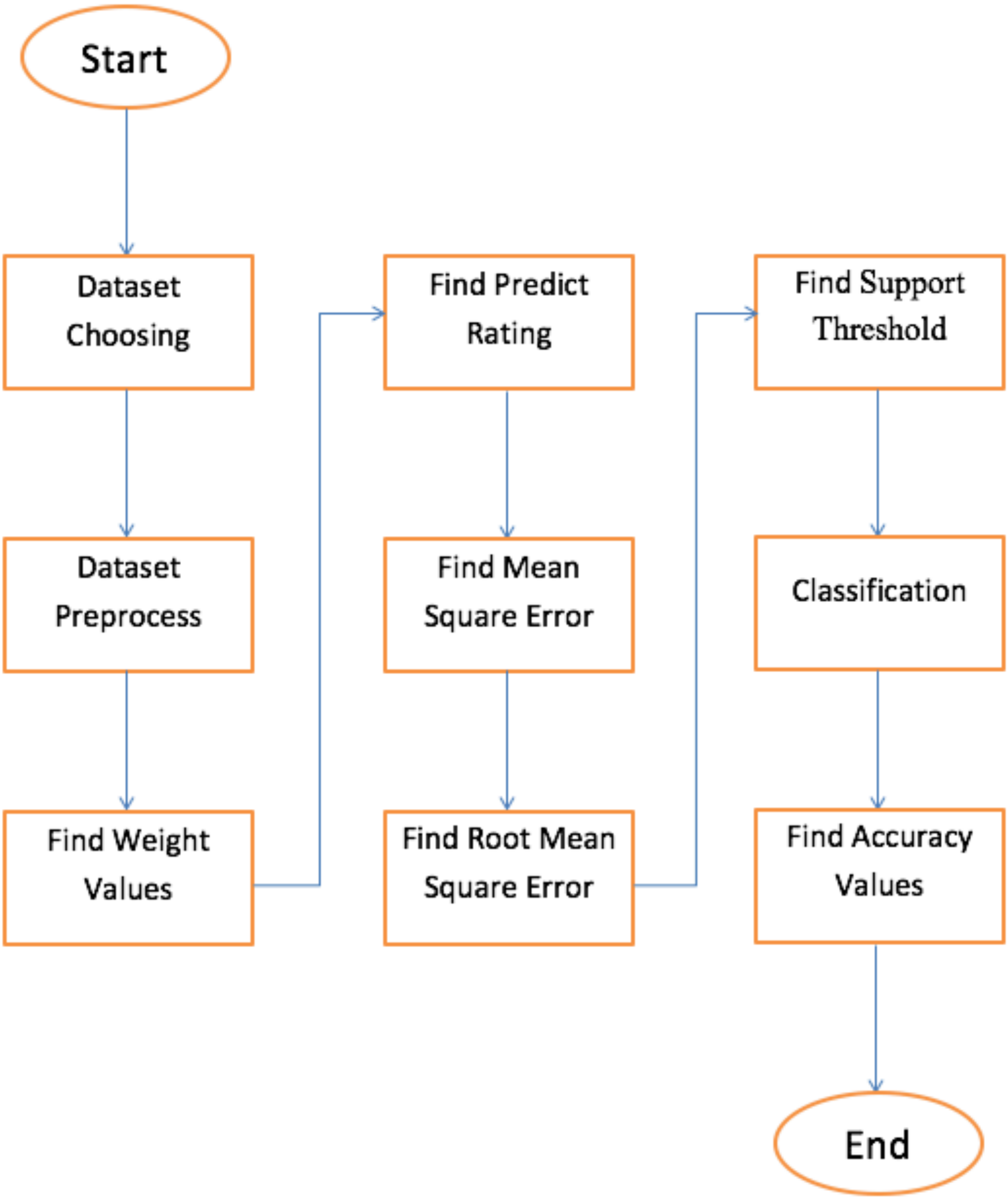


Figure 3. Flow Diagram

3.3 Use Case Diagram

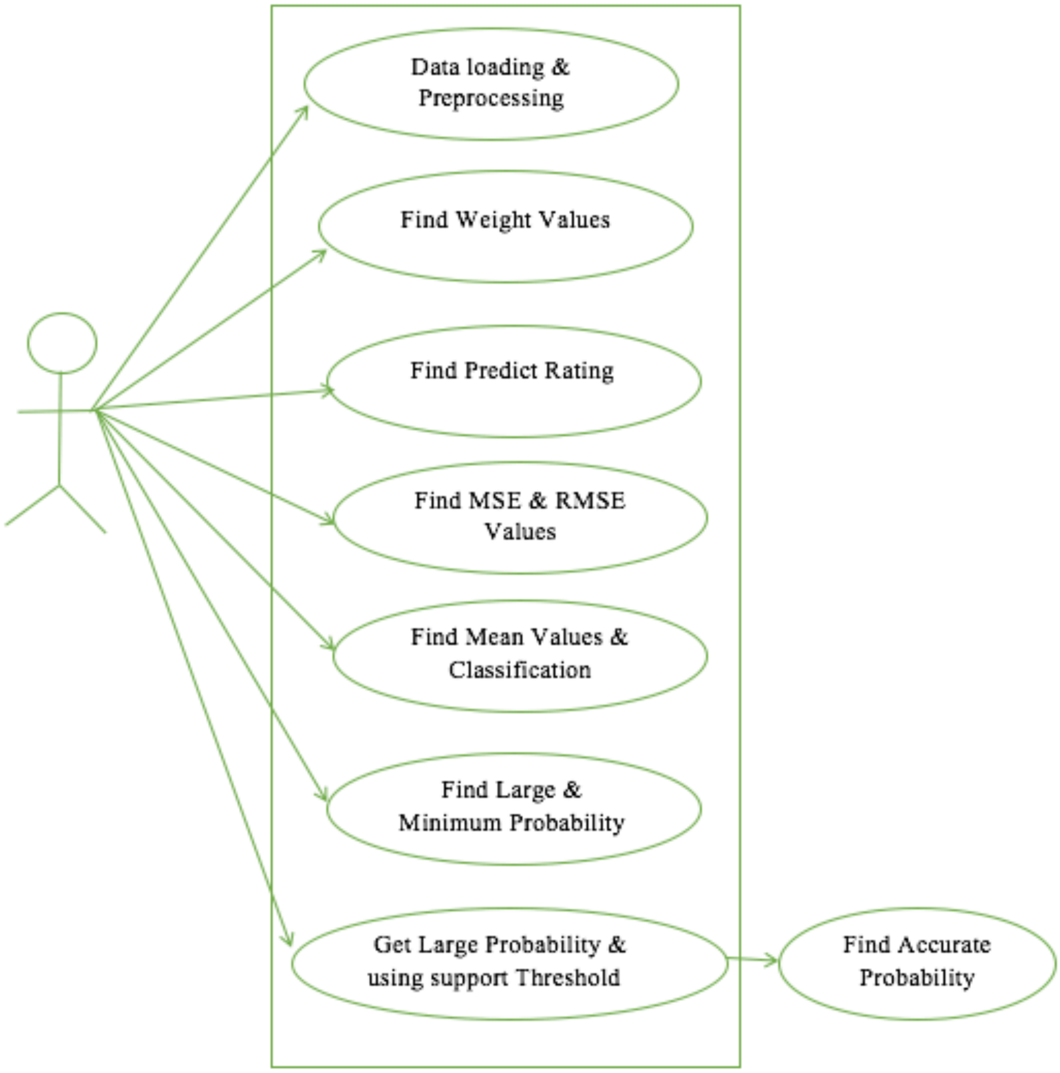


Figure 4. Use Case Diagram

3.4 Class Diagram

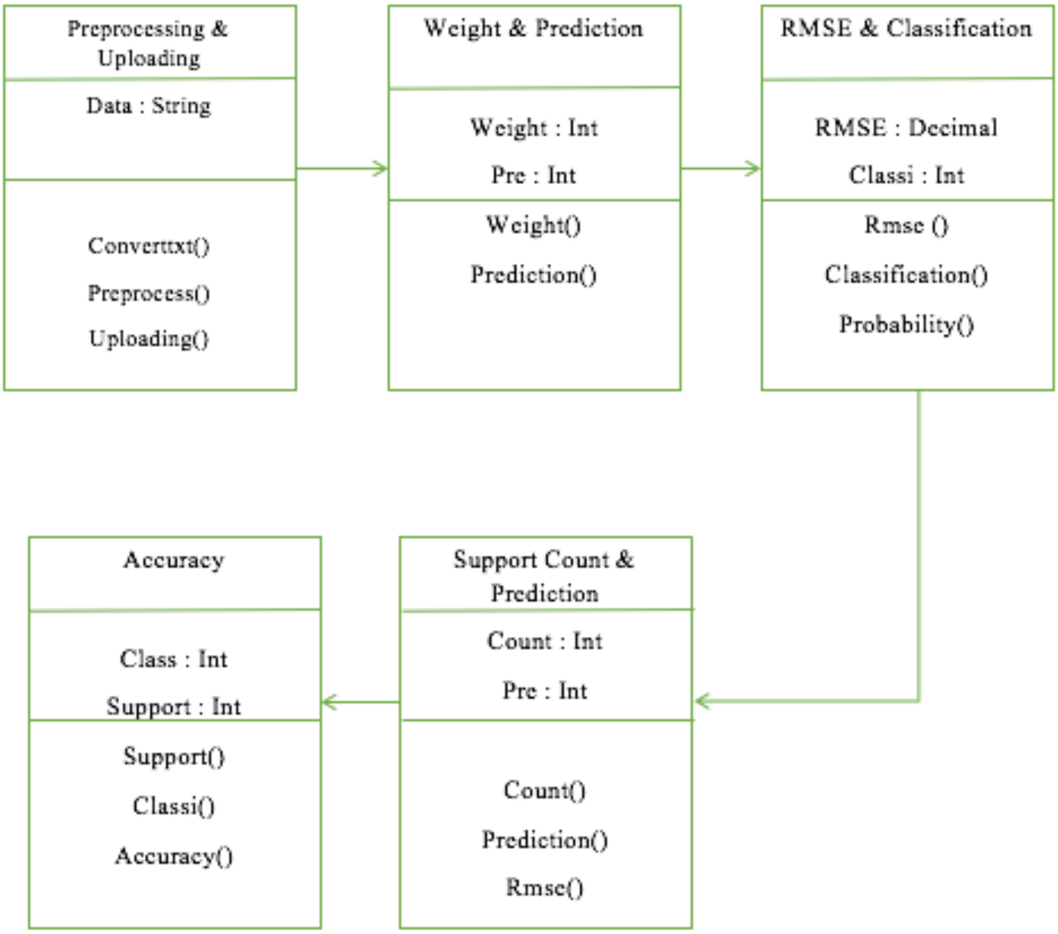


Figure 5. Class Diagram

3.5 Sequence Diagram

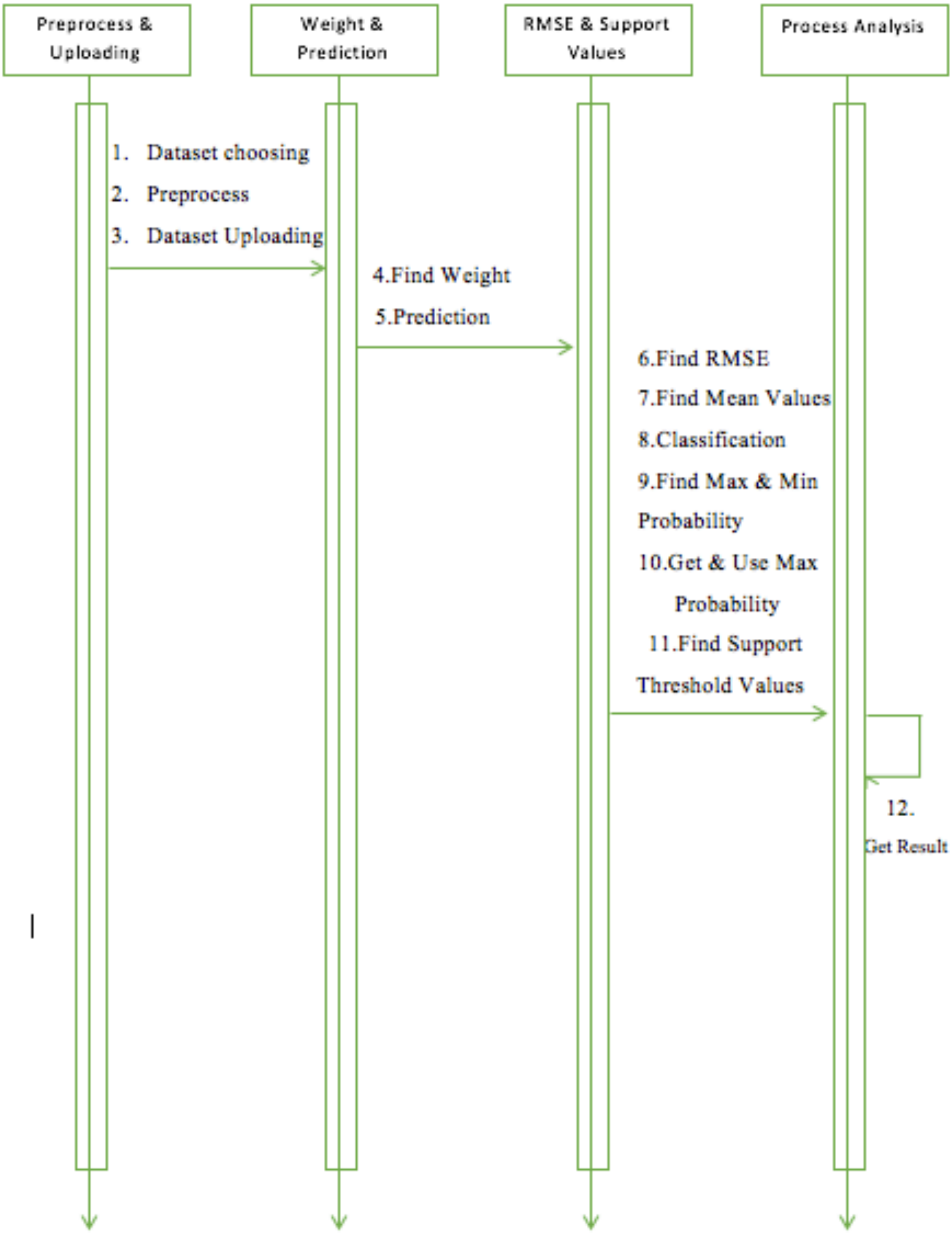


Figure 6. Sequence Diagram

3.6 Entity-Relationship Diagram

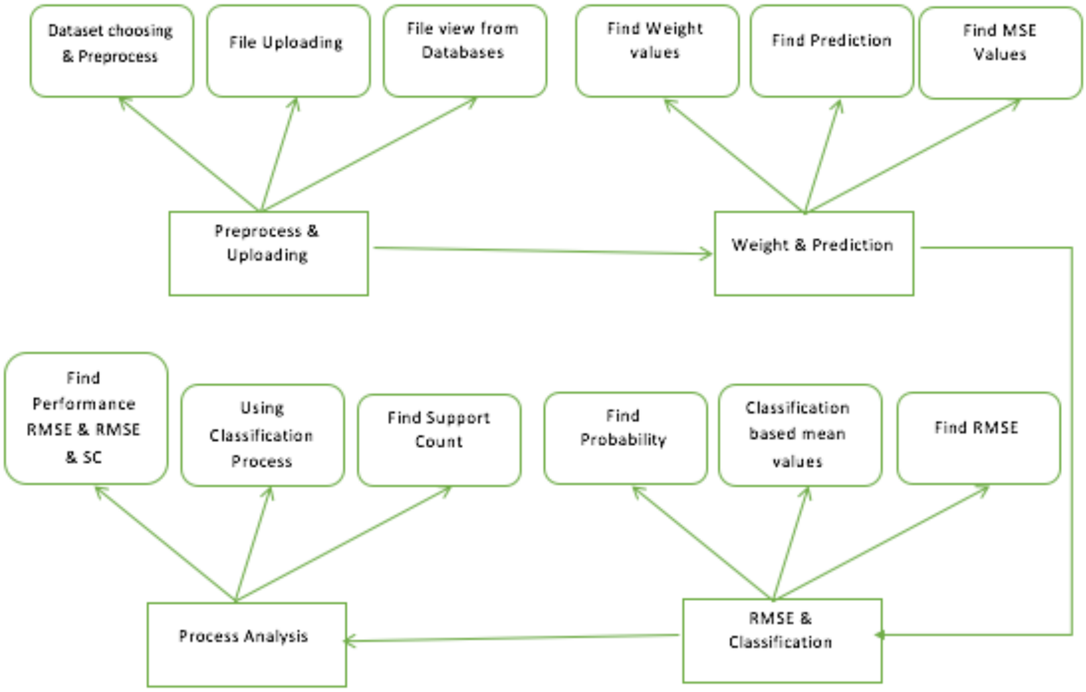


Figure 7. Entity Relationship Diagram

3.7 Modules

3.7.1 Data loading and preprocessing

It's a module consisting of 4 different phases. Data loading and preprocessing phase lets the user to select the data and load it to the application, where data preprocessing is the process which converts the loaded data to proper machine readable format by removing the extra space in the data [1]. Equivalence classes (EC) consists of set of contrast patterns that are portioned from total set to avoid redundant pattern processing [1]. Its adequate to deal with just one pattern per EC because multiple patterns possessing exact behavior can be counted as one [1]. Furthermore, it can be demonstrated that every single EC can be represented by a closed pattern (the longest in the EC) and a series of minimal-generator (MG) patterns (minimal with respect to); so an EC consists of entire set of patterns Q fulfilling the criteria "Q is a superset of some MG and Q is the subset of the closed pattern, of the EC" [1].

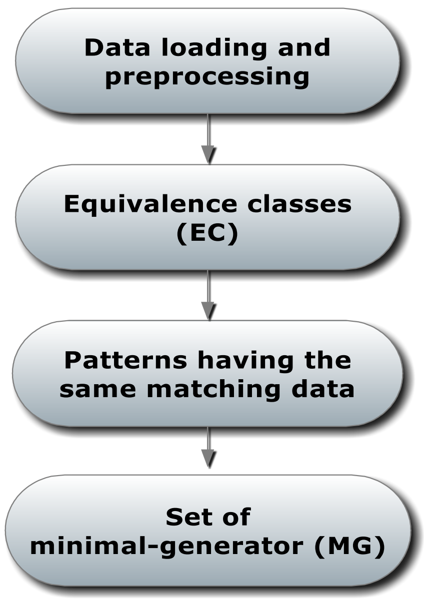


Figure 8. Data Loading & Preprocessing

3.7.2 Predict rating

In the CPXR method, the first thing we do is to divide the dataset D into two collections, LE (Large Error) and SE (Small Error), containing instances of D where f (prediction model) forms large/small prediction errors respectively [1]. CPXR then examines for a small collection of contrast patterns of LE to enhance the trr (Total residual reduction) measure, and utilizes that set to develop a PXR model [1].

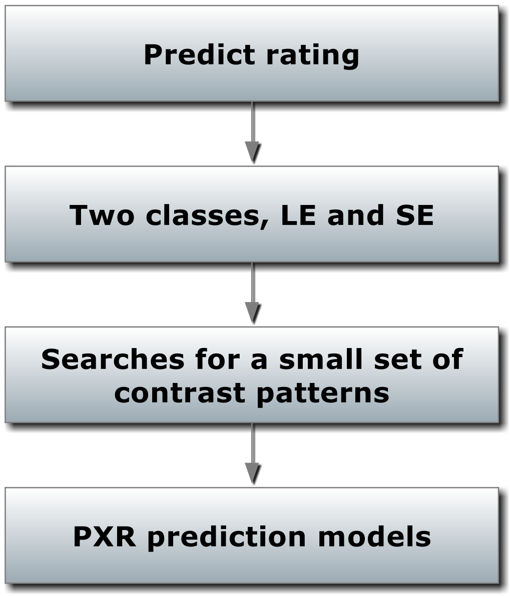


Figure 9. Predict Rating

3.7.3 Root mean square

The efficiency of a prediction model 'f' is generally calculated based on its prediction residuals [1]. The residual of 'f' on any appropriate instance x_i is $f(x_i) - y_i$, the difference between the predicted and observed response variable values [1]. The most widely used quality measure is RMSE (Root Mean Square Error), where $RMSE(f) = \sqrt{\frac{\sum_{i=1}^{n}(f(x_i)-y_i)^2}{n}}$ [1].

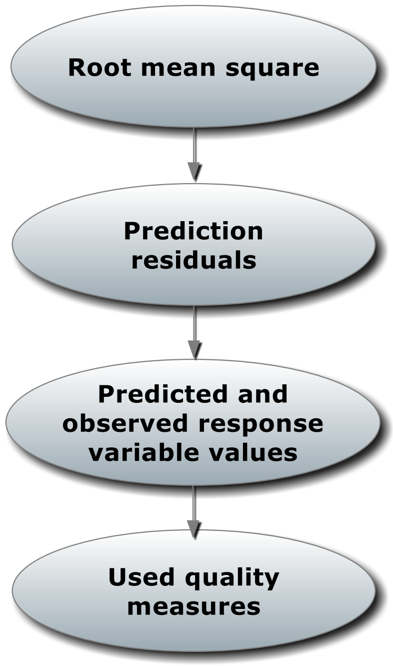


Figure 10. Root Mean Square

3.7.4 Compare accuracy

In Compare Accuracy module, we compare the RMSE reduction achieved by following two techniques CPXR(LP) and CPXR(LL) [1].

-CPXR(LP) is CPXR using LR (Linear Regression Algorithm) to build baseline models and PL (Piecewise Linear Regression) to build local regression models [1].

-CPXR(LL) is CPXR using LR (Linear Regression Algorithm) to build baseline models and LR (Linear Regression) to build local regression models [1].

CPXR(LP) attained highest average RMSE reduction (42.89%) over 80% several times during the experiments and it was found that CPXR(LL) being less accurate among the two [1].

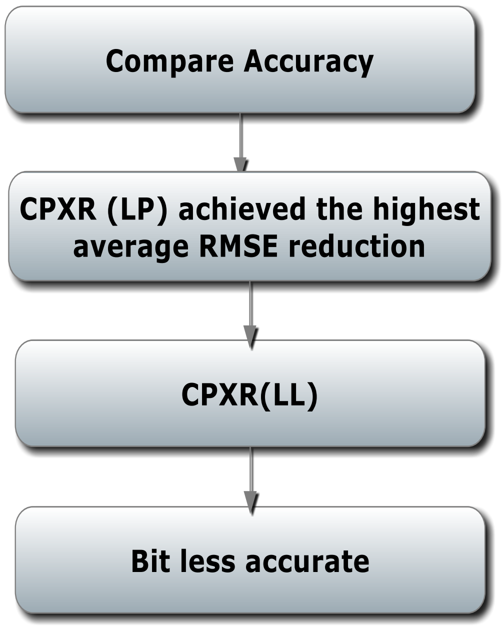


Figure 11. Compare Accuracy

3.7.5 Performance analysis

The CPXR is compared with other regression techniques, on several datasets for factors such as prediction accuracy, overfitting, and sensitivity to noise [1]. The outcomes of the comparison show that CPXR performs constantly higher than other techniques and that to with big-margins [1]. We test the effect of parameters and standard regression methods on CPXR's efficiency, computation time, and usage of memory [1]. At last we review the benefits of utilizing contrast patterns instead of frequent patterns in developing PXR models [1].

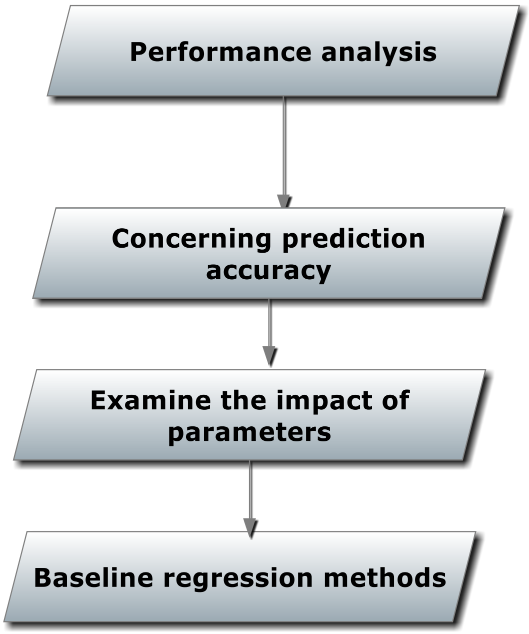


Figure 12. Performance Analysis

3.7.6 Support Values

Double pruning is the technique that can be achieved by using support values to enhance the classification accuracy and discard the outliers from the data [1]. In machine learning, pruning is the technique that is used for decreasing the data size by discarding parts of data that give limited power to categorize instances [1].

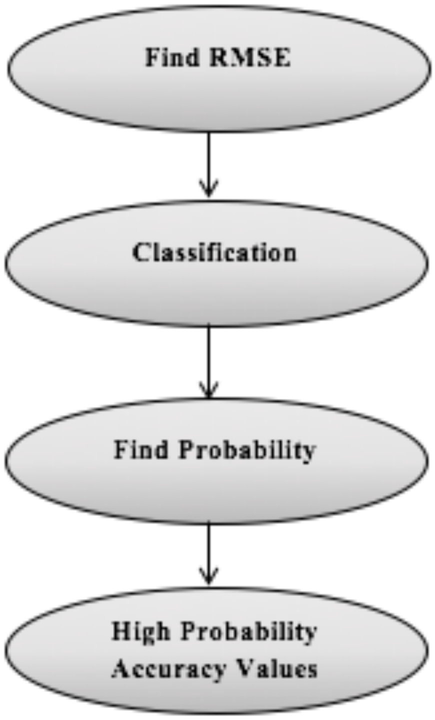


Figure 13. Support Values

CHAPTER 4

System Requirements

4.1 Software Requirements

- Operating System : Windows.
- Programing Language : Java.
- IDE : NetBeans 7.2.1
- Data Base : MySQL

4.2 Hardware Requirements

- CPU : Minimum 2.4 GHz
- Hard Disk : Minimum 160 GB
- RAM : Minimum 2GB

CHAPTER 5

Software Description

5.1 Java

Java is an object oriented programming language designed for cross platform usage [6]. Java applications are generally complied to byte code (class file) that can execute on any Java Virtual Machine (JVM) indifferent of computer architecture [6]. Java goes by the phrase "Write once, run anywhere" [6]. Java is widely used for client-server web applications [6].

5.2 Net Beans

Net beans is a framework for building Java Swing desktop applications [7]. It's designed to provide reusability and for simplification of java applications development [7]. It's an Integrated Development Environment (IDE) package for Java SE, that's essential to start Net Beans plug-in and Net beans platform based applications development, supplementary SDK is not needed [7].

5.3 WAMP Server

Windows/Apache/MySQL/PHP (WAMP), it's a web development environment [8]. The operating system for WAMP is Windows, web server is Apache, database is MySQL, scripting language is PHP [8]. WAMP server comes with an interactive GUI for MySQL database manager called 'PHPMyAdmin' [9].

5.4 MySQL

MySQL is an open source Structured Query Language for relational database management system [10]. Its written in C and C++ [10]. It's world's most widely used Query Language for its simplicity and high functionality [10]. Some of the popular applications using MySQL database are Drupal, WordPress, and TYPO3, etc. [10].

5.5 Platforms and interfaces

Lot of programming languages with language-specific APIs consists libraries for using databases [11]. This include Java Database Connectivity (JDBC) driver for Java [11]. JDBC presents approaches to query and revise data in database [11]. A JDBC-to-ODBC bridge facilitates to utilize the ODBC functionalities present in Java Virtual Machine (JVM) [11].

CHAPTER 6

Experiments and Results

The application has been trained and tested with cancer and student datasets. The objective of the research behind building this application is to analyze the contrast patterns from the dataset and produce accurate and interpretable PXR model [1]. The aim is to find the patterns from the cancer dataset which show high probability for cancer and the goal with student dataset is to predict students' performance in secondary education. The cancer dataset consists of 16 attributes and 1660 instances & the Student dataset consists of 33 attributes and 649 and 395 instances for Student-por (Student's who took Portugal language class) and Student-mat (Student's who took Math class) respectively. The application uses the statistical functions and formulas proposed in the research paper to achieve high performance output. The Results table shows the stats of the application's performance in terms of the dataset size, execution time and memory usage.

Table 2. Results

Dataset	Number of Instances	Number of attributes	Execution Time (minutes)	Memory (MB)
Cancer	1660	16	0.34	5.43
Student-por	649	33	0.21	3.72
Student-mat	395	33	0.13	2.31

CHAPTER 7

Conclusion

The research paper allowed us to successfully implement the novel concept of CPXR methods using statistical functions and formulas coupled with pruning in the application to build more accurate and interpretable PXR model as compared to other methods.

The results achieved from the experiments suggest that CPXR method predicted output variables better than traditional PXR generation techniques in both train and test phases of application. In generic, CPXR can efficiently deal with data having varied predictor-response relationships.

CHAPTER 8

Future Work

- The application can be designed to test with some other classification and regression dataset.
- The application can be designed to work in the area of medical data.
- The application can be designed in such a way that it can extract the patterns from the images and then proceed further to pattern recognition phase.
- The application in future can be combined with a retina or finger print scanning machine and store the scans from the scanner as a pattern and then run the algorithm to recognize the various patterns. This will help in biometric security.
- Big datasets with large volumes of data can be used to test the application and its performance.

LIST OF REFERENCES

[1] G. Dong and V. Taslimitehrani, 'Pattern-Aided Regression Modeling and Prediction Model Analysis', *IEEE Trans. Knowl. Data Eng.*, vol. 27, no. 9, pp. 2452-2465, 2015.

[2] Wikipedia, 'Pattern recognition', 2015. [Online]. Available: https://en.wikipedia.org/wiki/Pattern_recognition. [Accessed: 23- Nov- 2015].

[3] Wikipedia, 'Pattern matching', 2015. [Online]. Available: https://en.wikipedia.org/wiki/Pattern_matching. [Accessed: 23- Nov- 2015].

[4] Sas.com, 'What is data mining?', 2015. [Online]. Available: http://www.sas.com/en_us/insights/analytics/data-mining.html?gclid=CjwKEAiA7MWyBRDpi5TFqqmm6hMSJAD6GLeA8QTskW3N15dnwFqjJ7Y-WgAaE6Bjq_rnDuxW1NOQJhoCmDjw_wcB. [Accessed: 23- Nov- 2015].

[5] Wikipedia, 'Machine learning', 2015. [Online]. Available: https://en.wikipedia.org/wiki/Machine_learning. [Accessed: 23- Nov- 2015].

[6] Wikipedia, 'Java (programming language)', 2015. [Online]. Available: https://en.wikipedia.org/wiki/Java_(programming_language). [Accessed: 23- Nov- 2015].

[7] Wikipedia, 'NetBeans', 2015. [Online]. Available: https://en.wikipedia.org/wiki/NetBeans. [Accessed: 23- Nov- 2015].

[8] Webopedia.com, 'What is WAMP? Webopedia', 2015. [Online]. Available: http://www.webopedia.com/TERM/W/WAMP.html. [Accessed: 23- Nov- 2015].

[9] Softonic, 'WampServer', 2015. [Online]. Available: http://wampserver.en.softonic.com/. [Accessed: 23- Nov- 2015].

[10] Wikipedia, 'MySQL', 2015. [Online]. Available: https://en.wikipedia.org/wiki/MySQL. [Accessed: 23- Nov- 2015].

[11] Wikipedia, 'Java Database Connectivity', 2015. [Online]. Available: https://en.wikipedia.org/wiki/Java_Database_Connectivity. [Accessed: 23- Nov- 2015].

APPENDIX

Additional Screen-shots

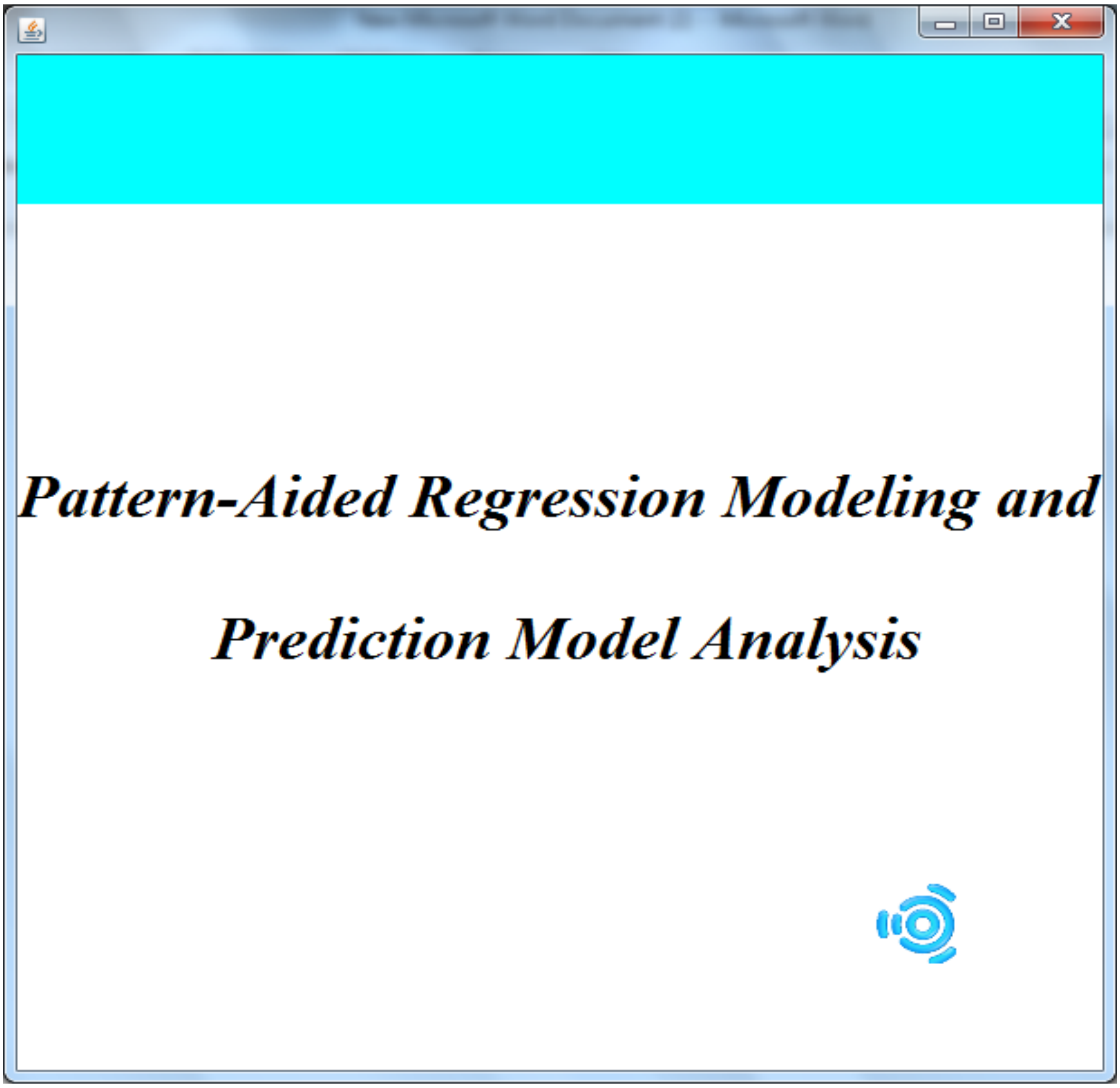


Figure 14. Application Start Frame

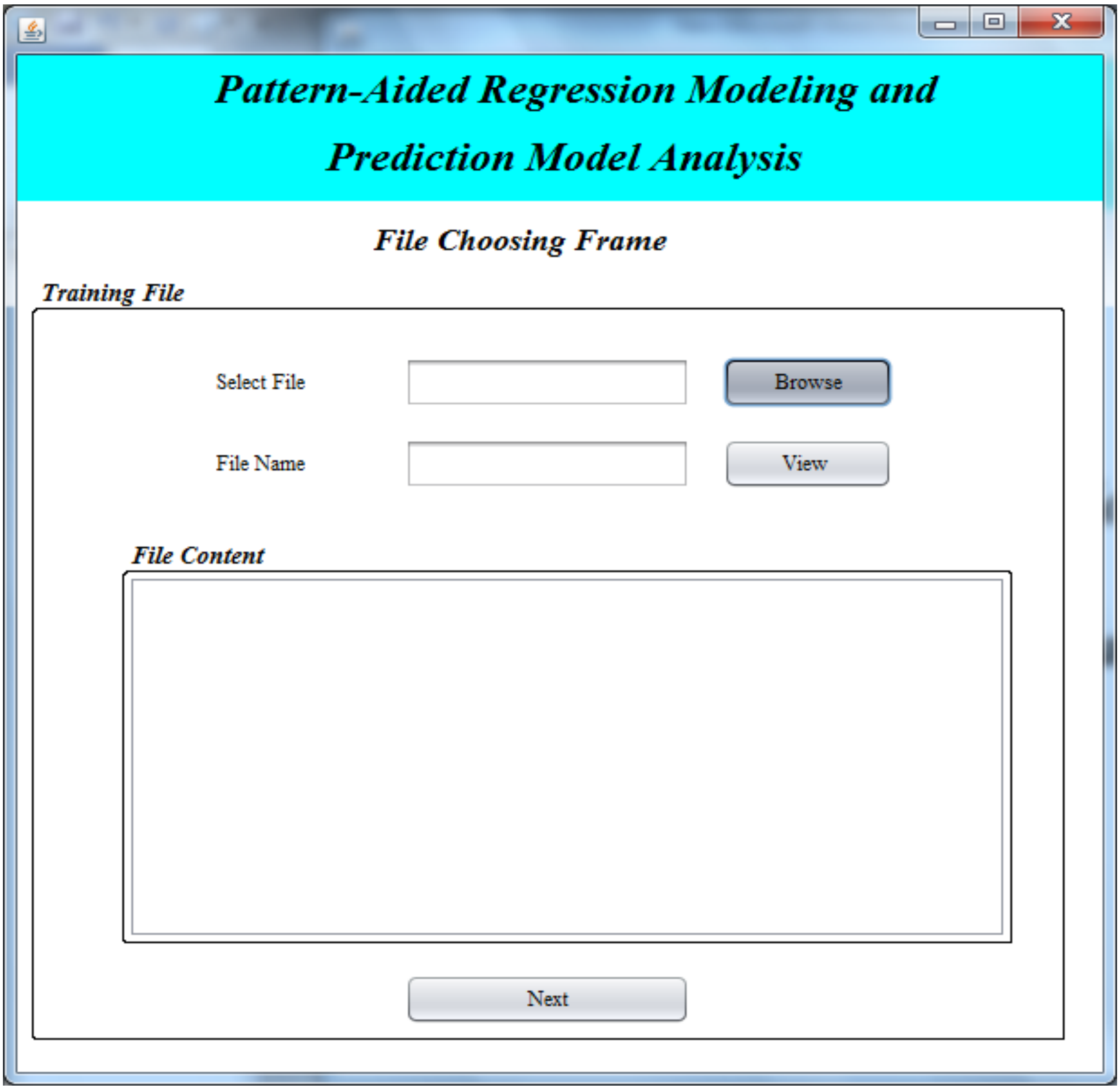


Figure 15. Data selection Frame 1

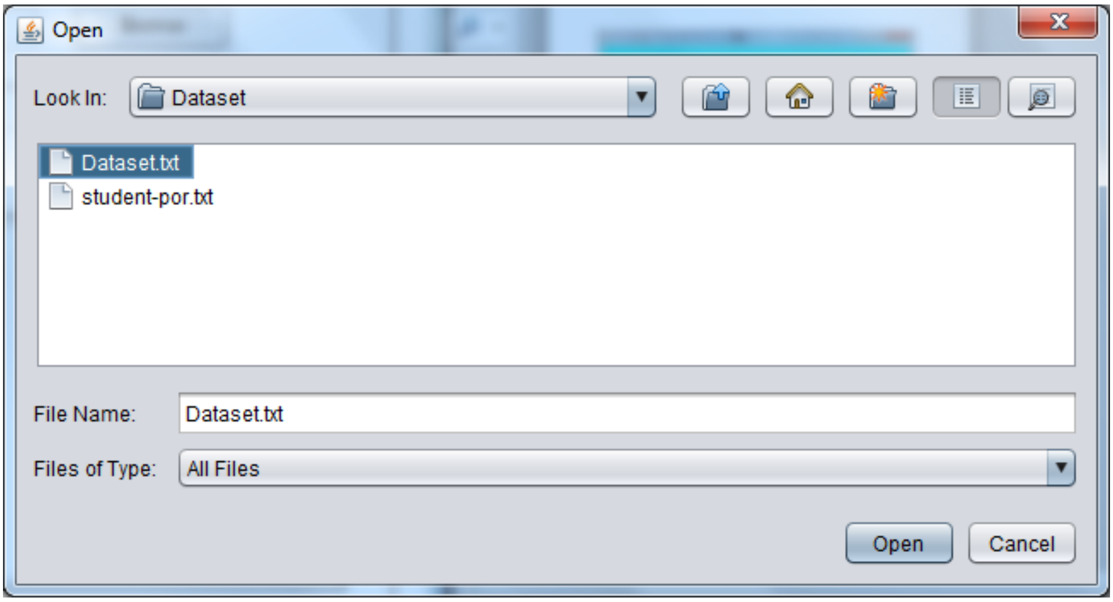


Figure 16. Data Selection Frame 2

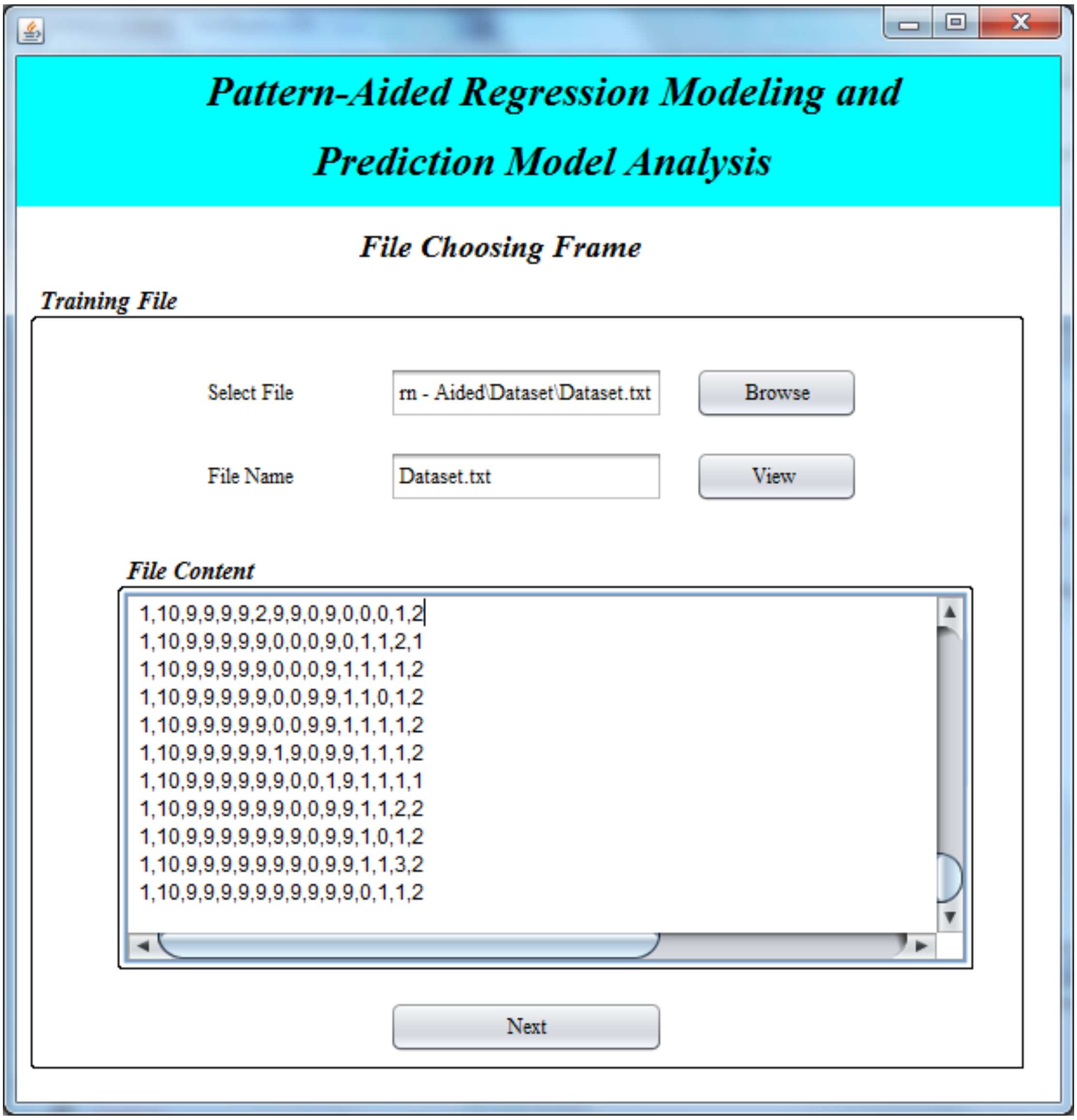


Figure 17. File Content View Frame

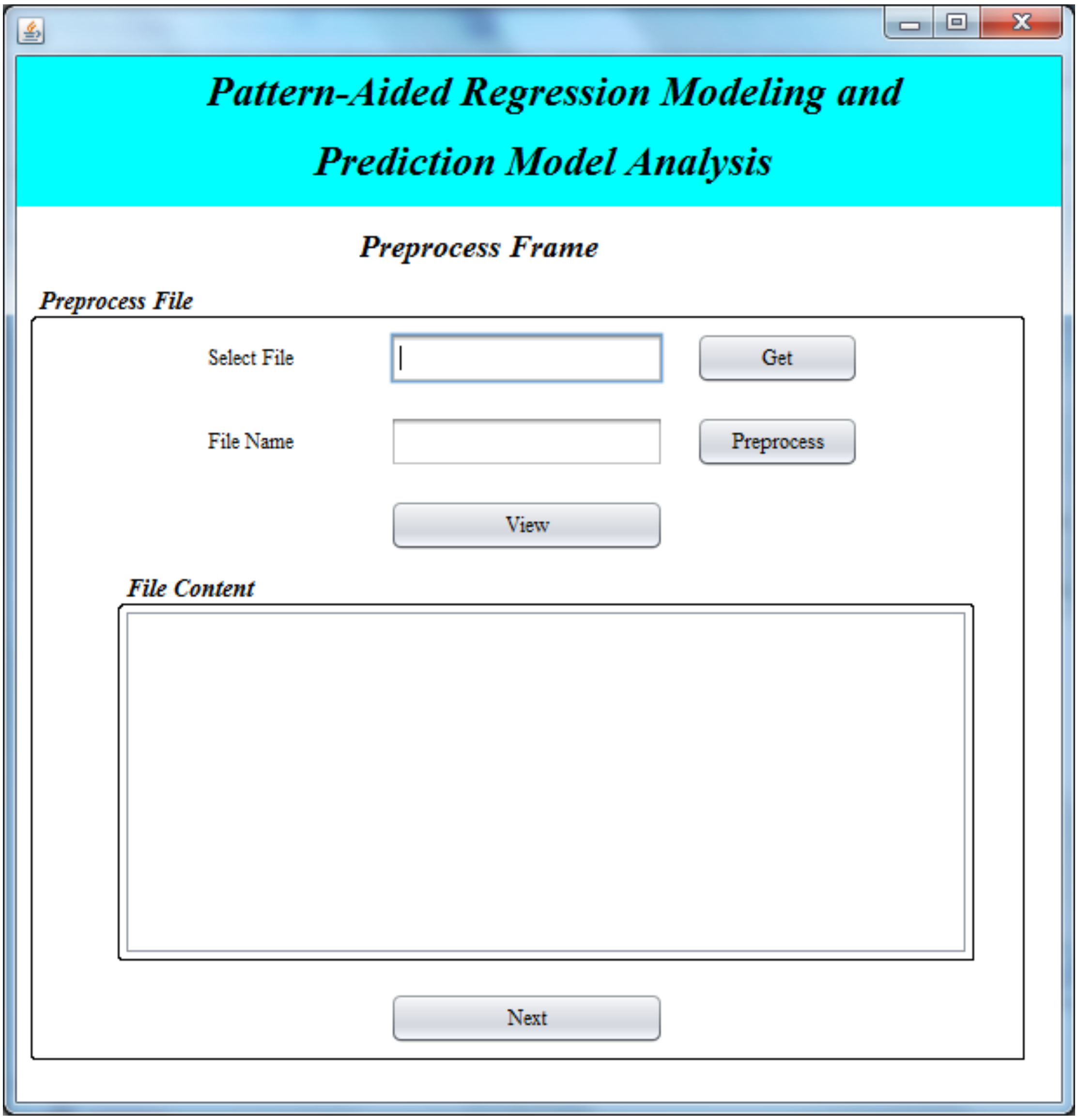


Figure 18. File Preprocess Frame 1

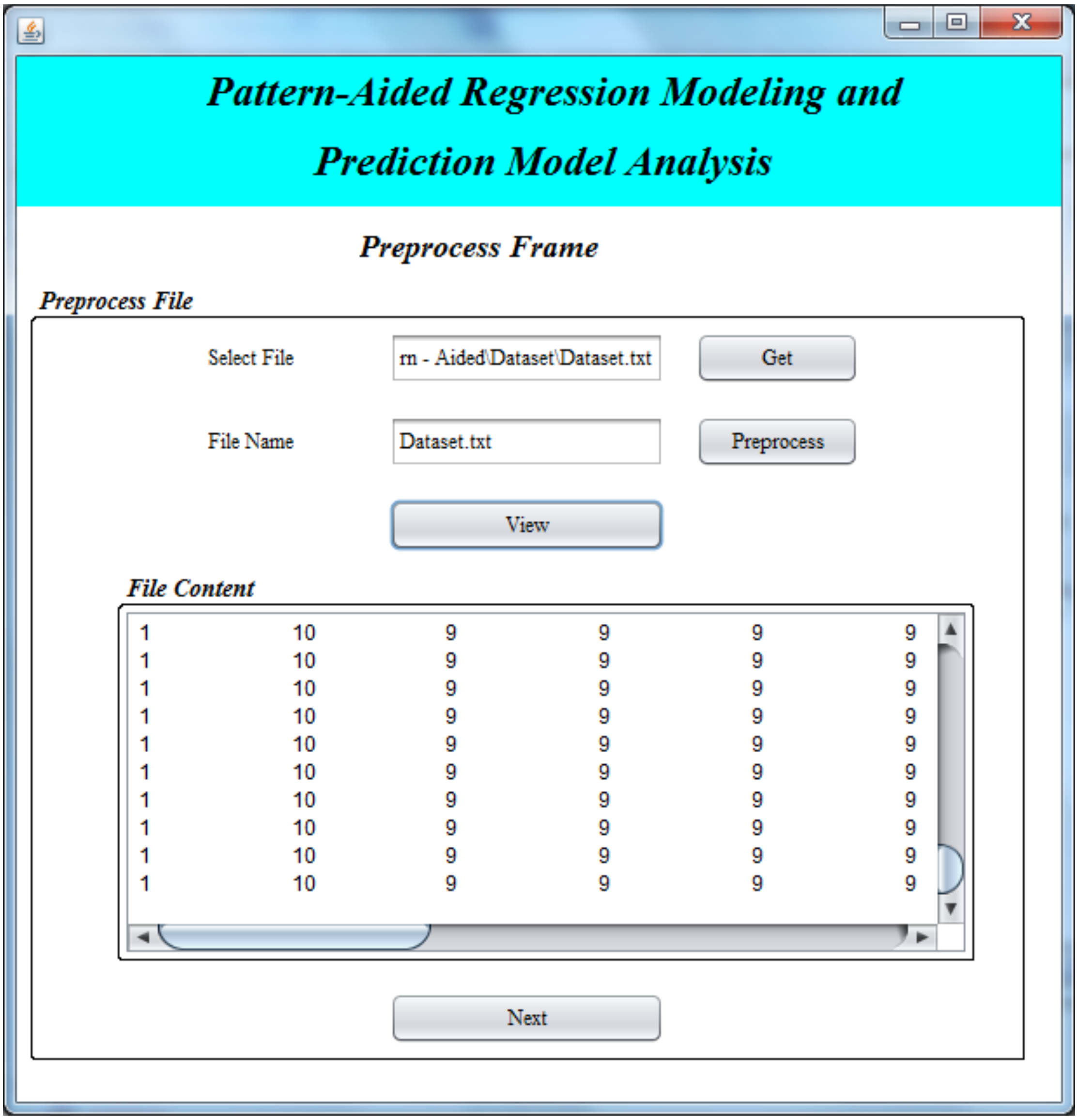


Figure 19. File Preprocess Frame 2

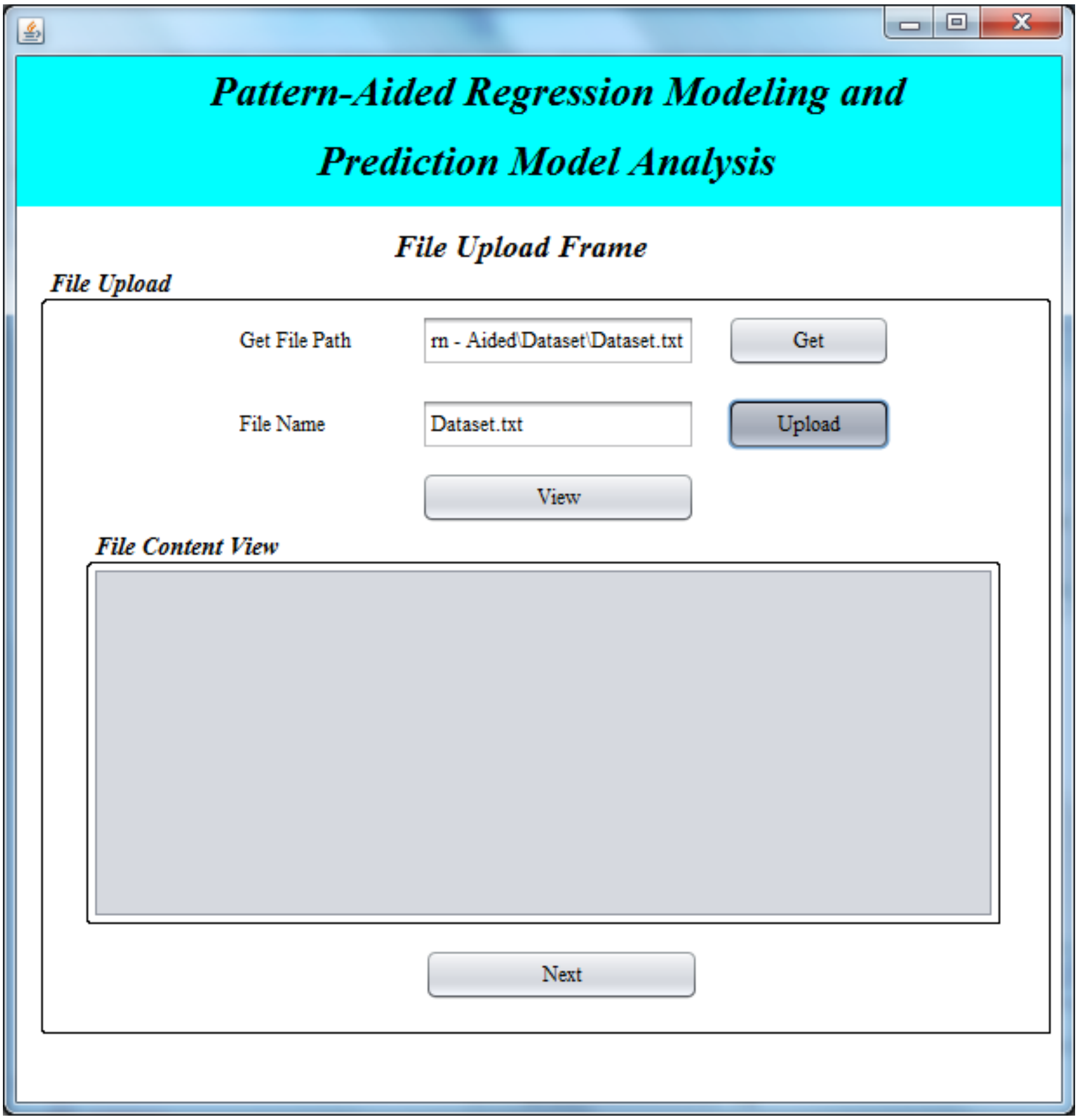


Figure 20. File Upload Frame 1

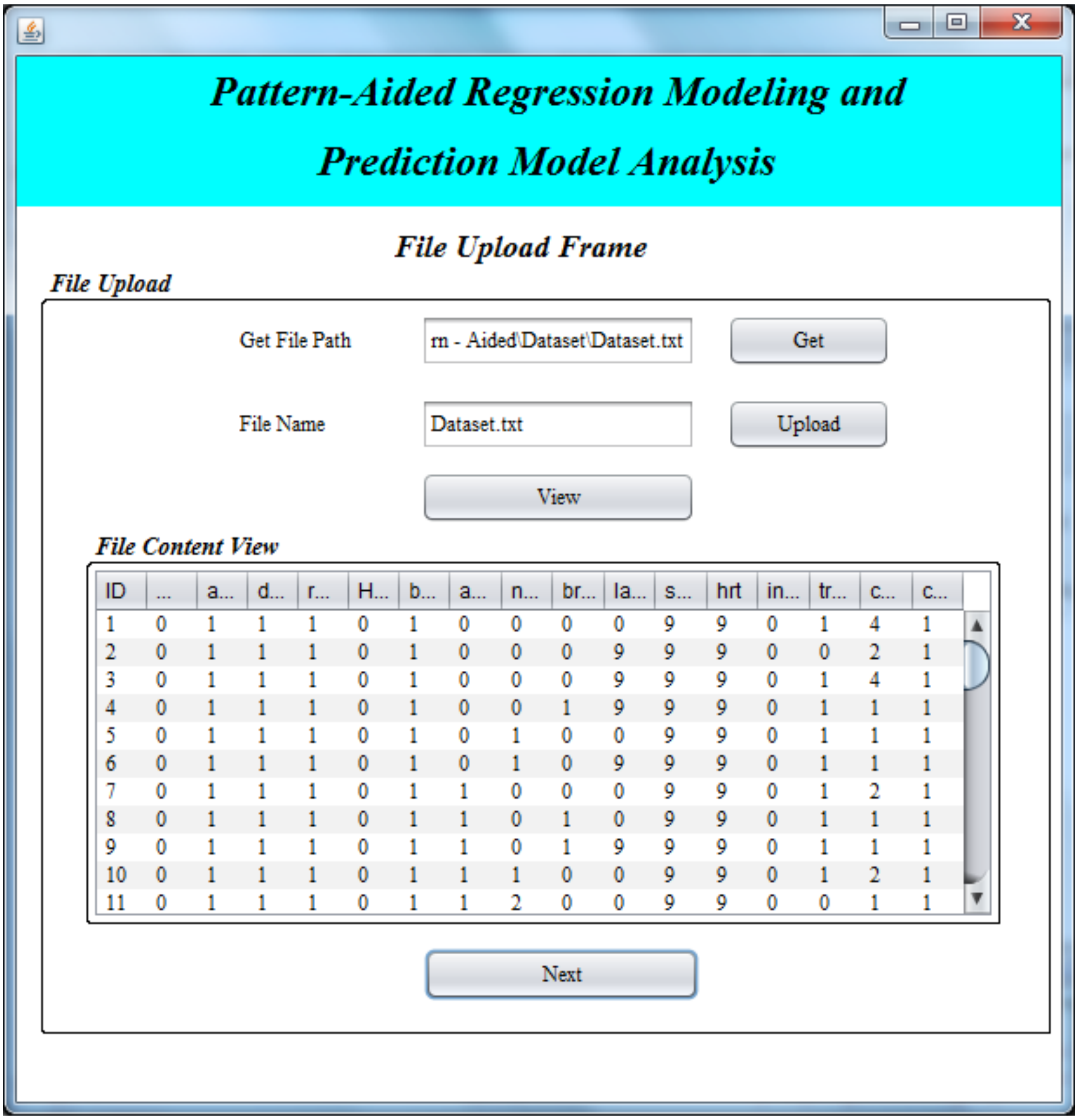


Figure 21. File Upload Frame 2

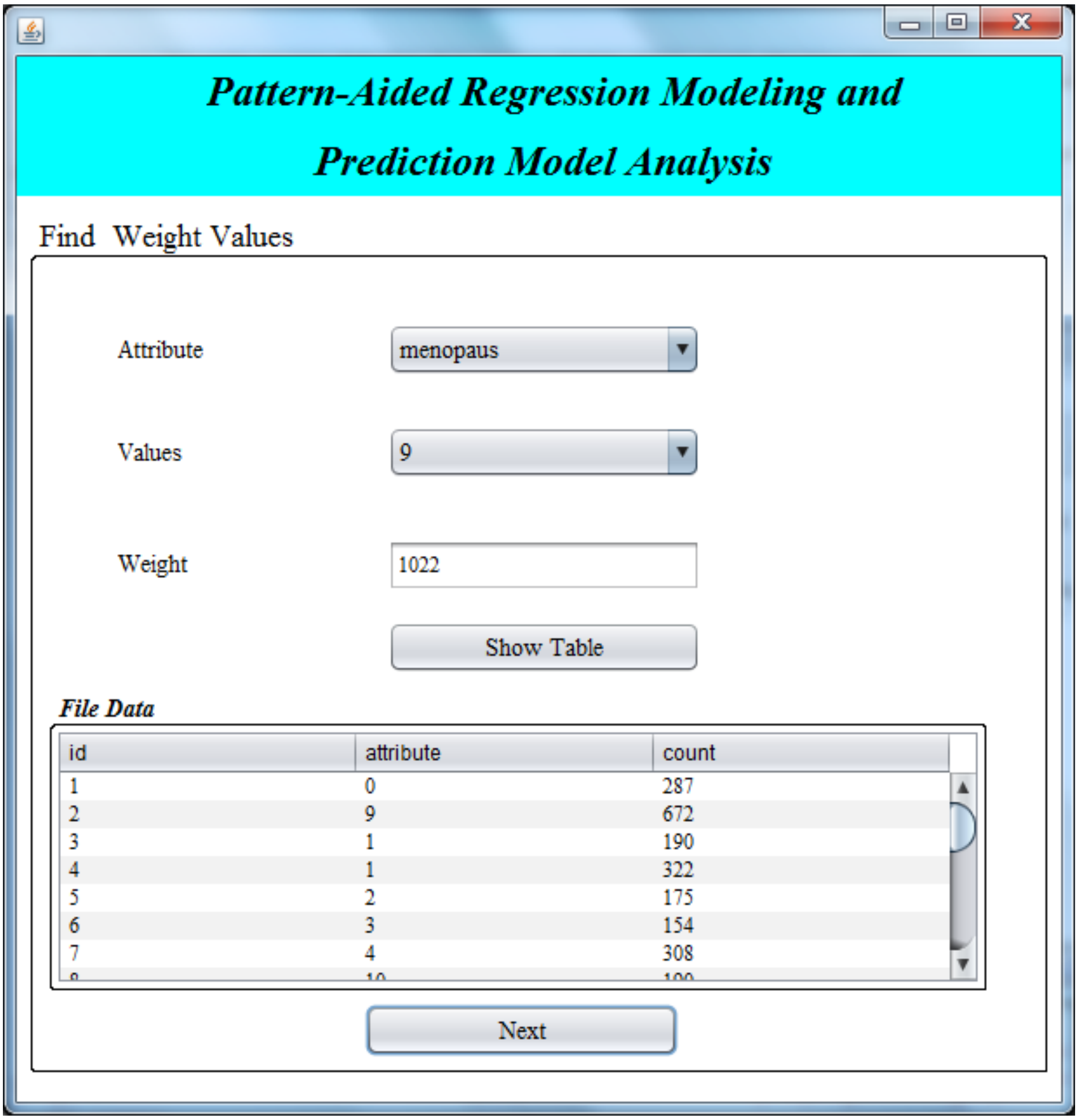


Figure 22. Find Weight Values Frame

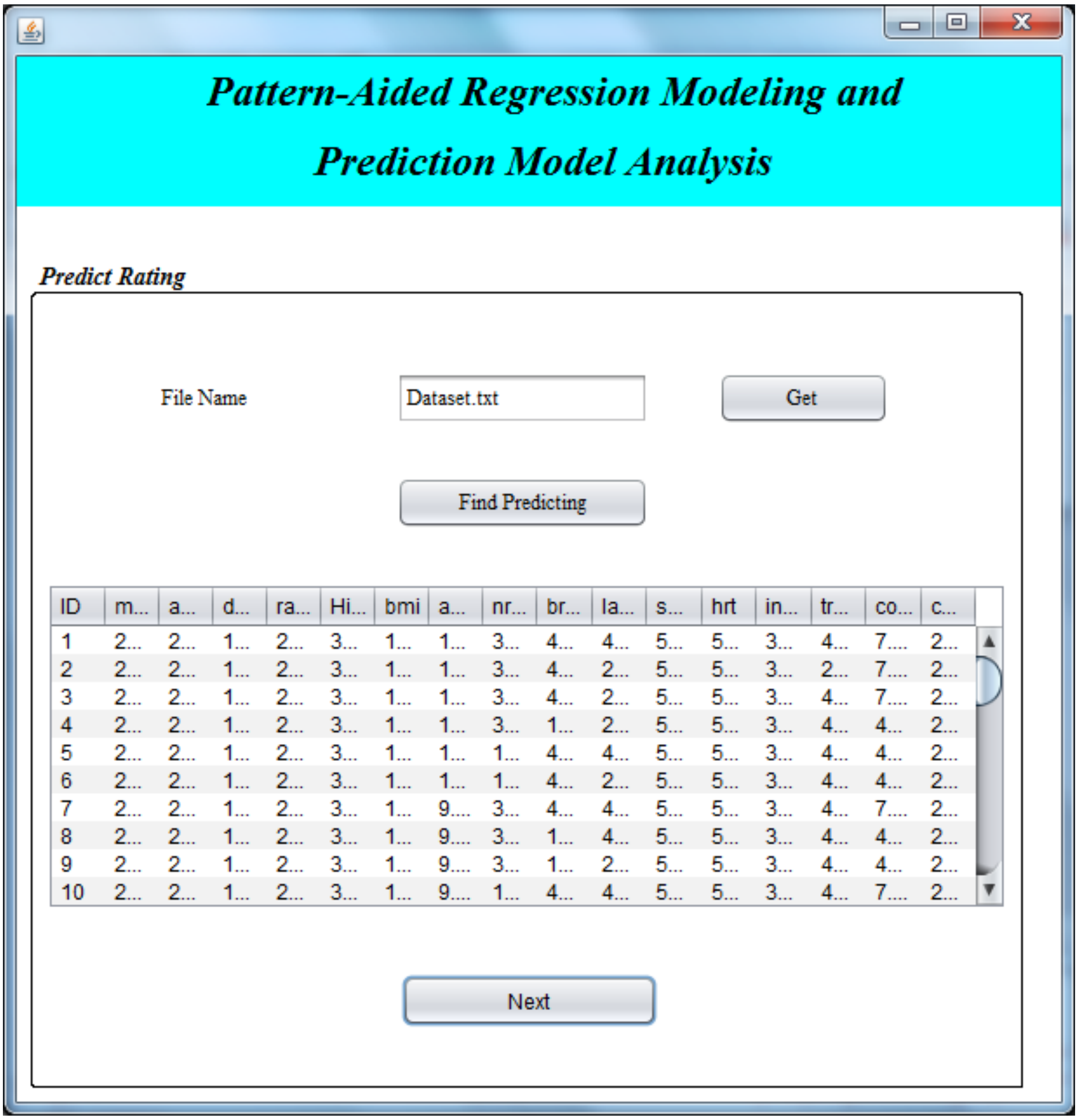


Figure 23. Find Predict Rating Frame

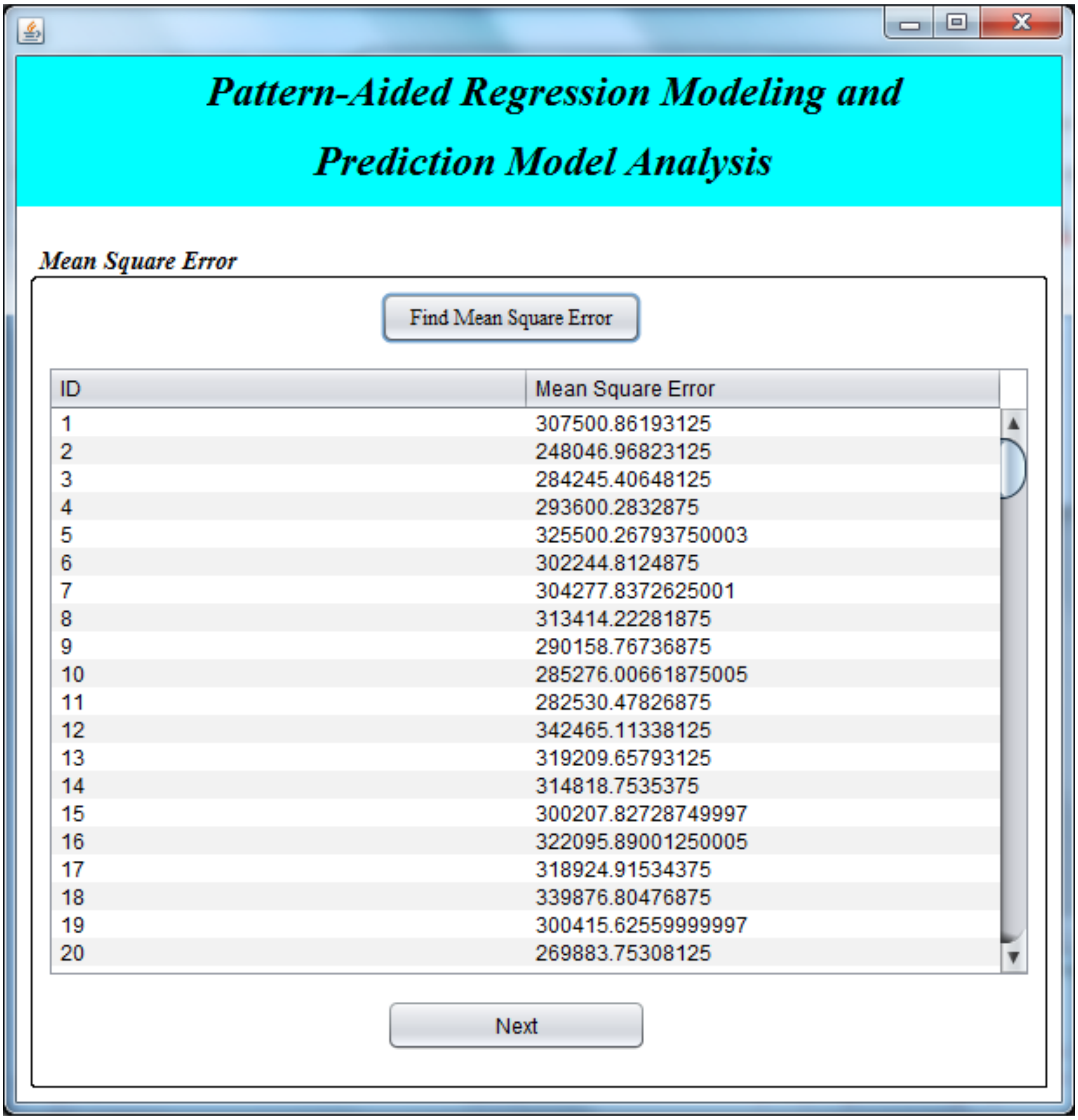


Figure 24. Find Mean Square Error Frame

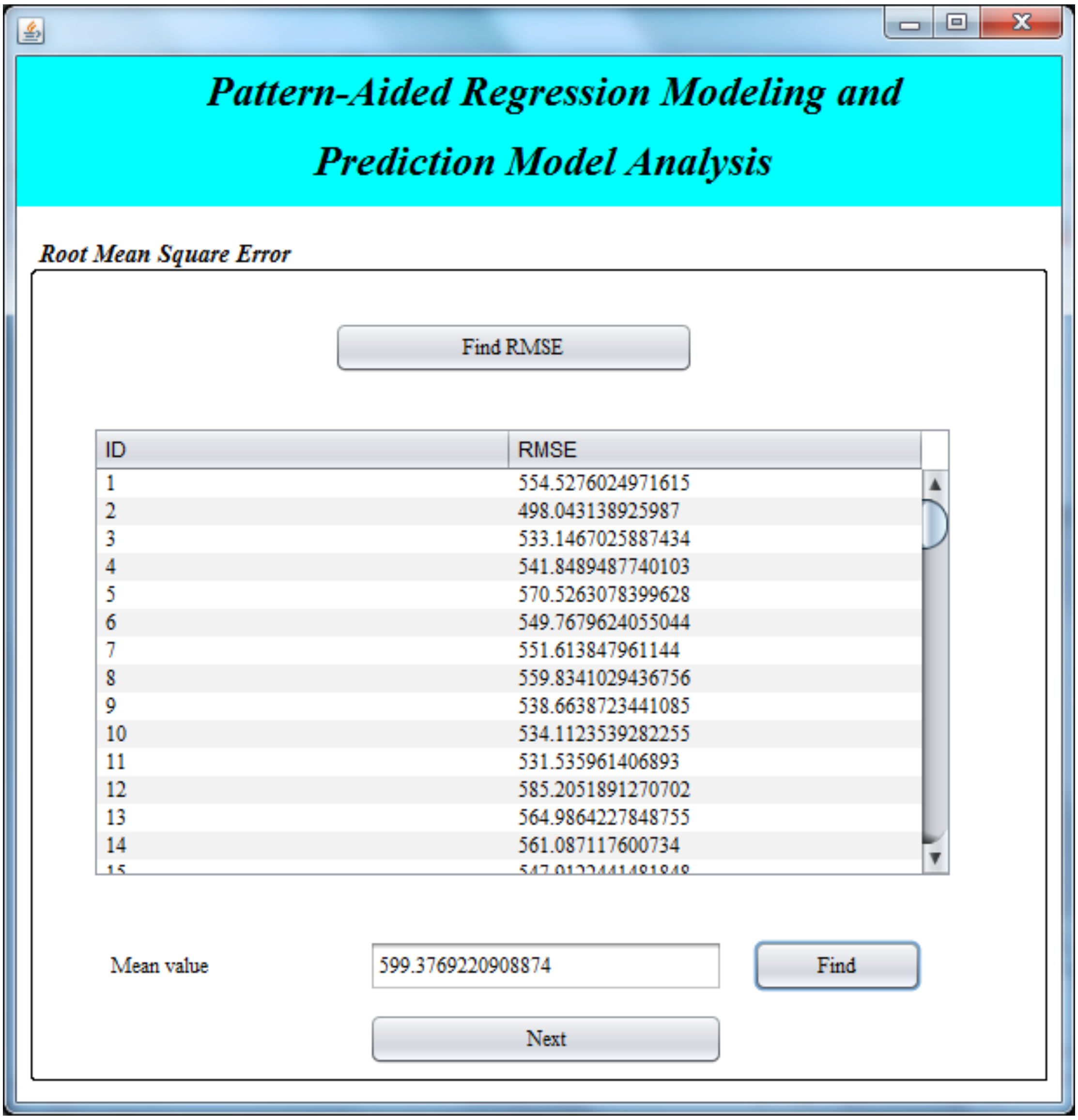


Figure 25. Find Root Mean Square Error and Mean Value Frame

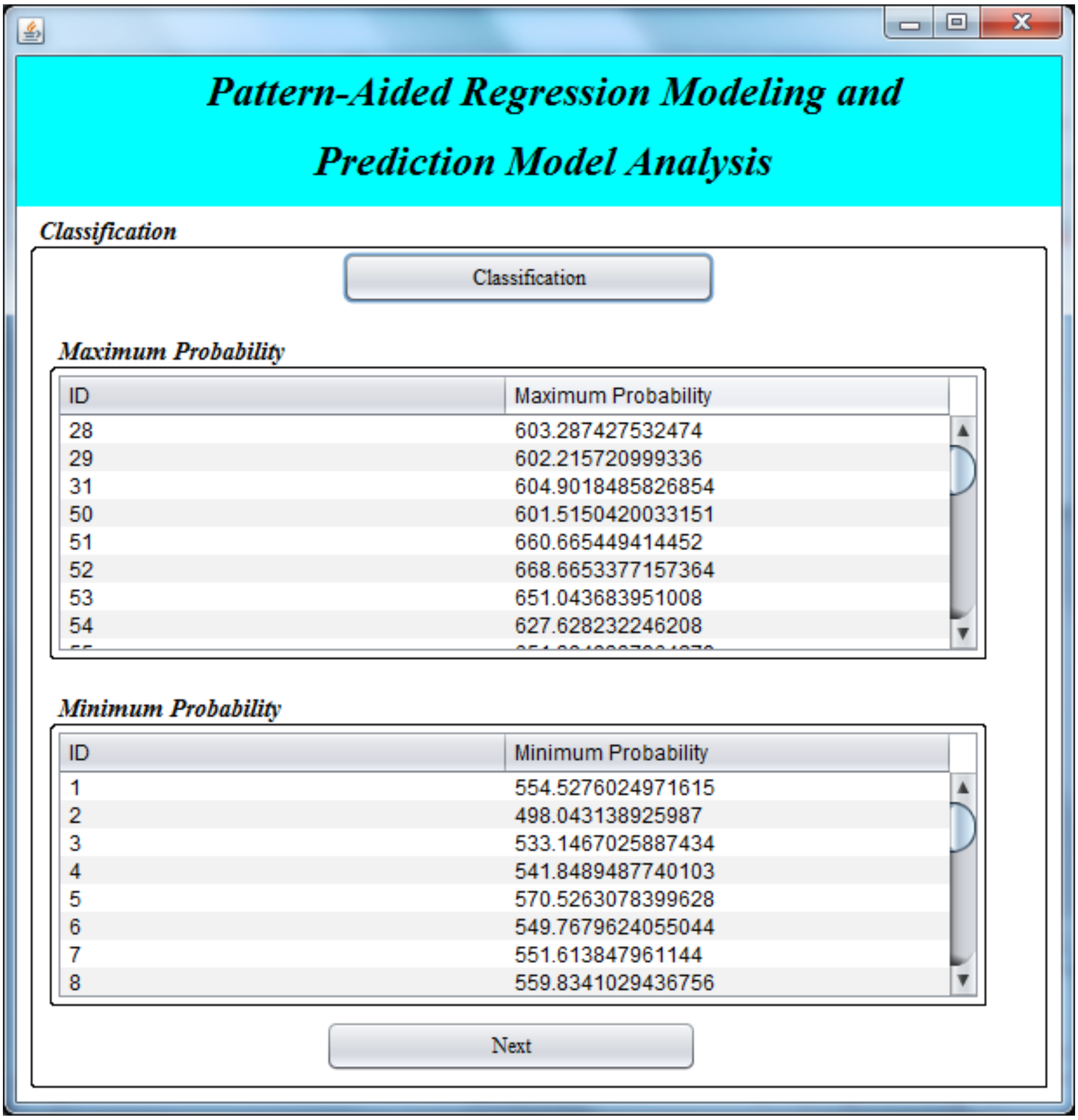


Figure 26. Find Classification Frame

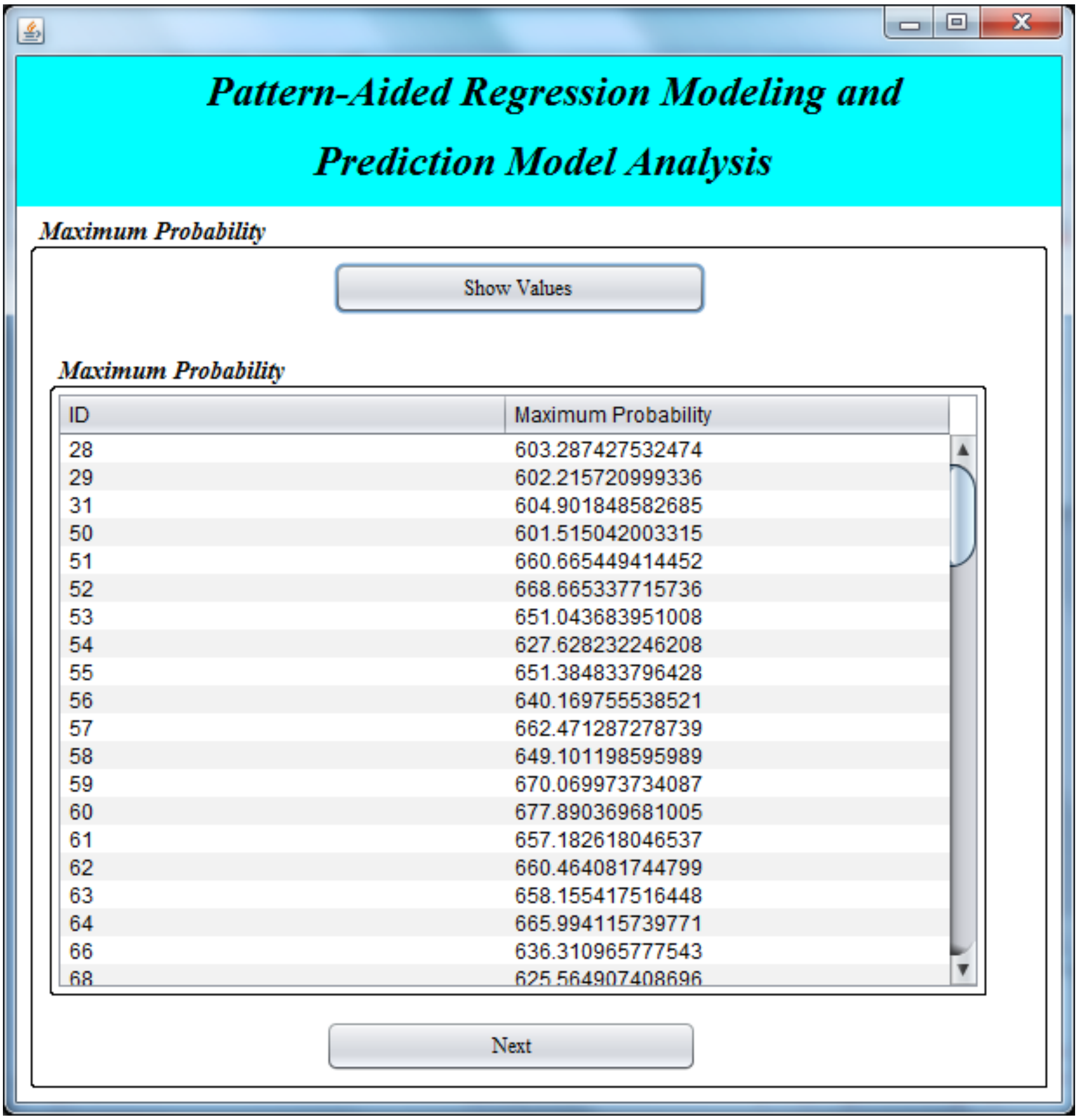
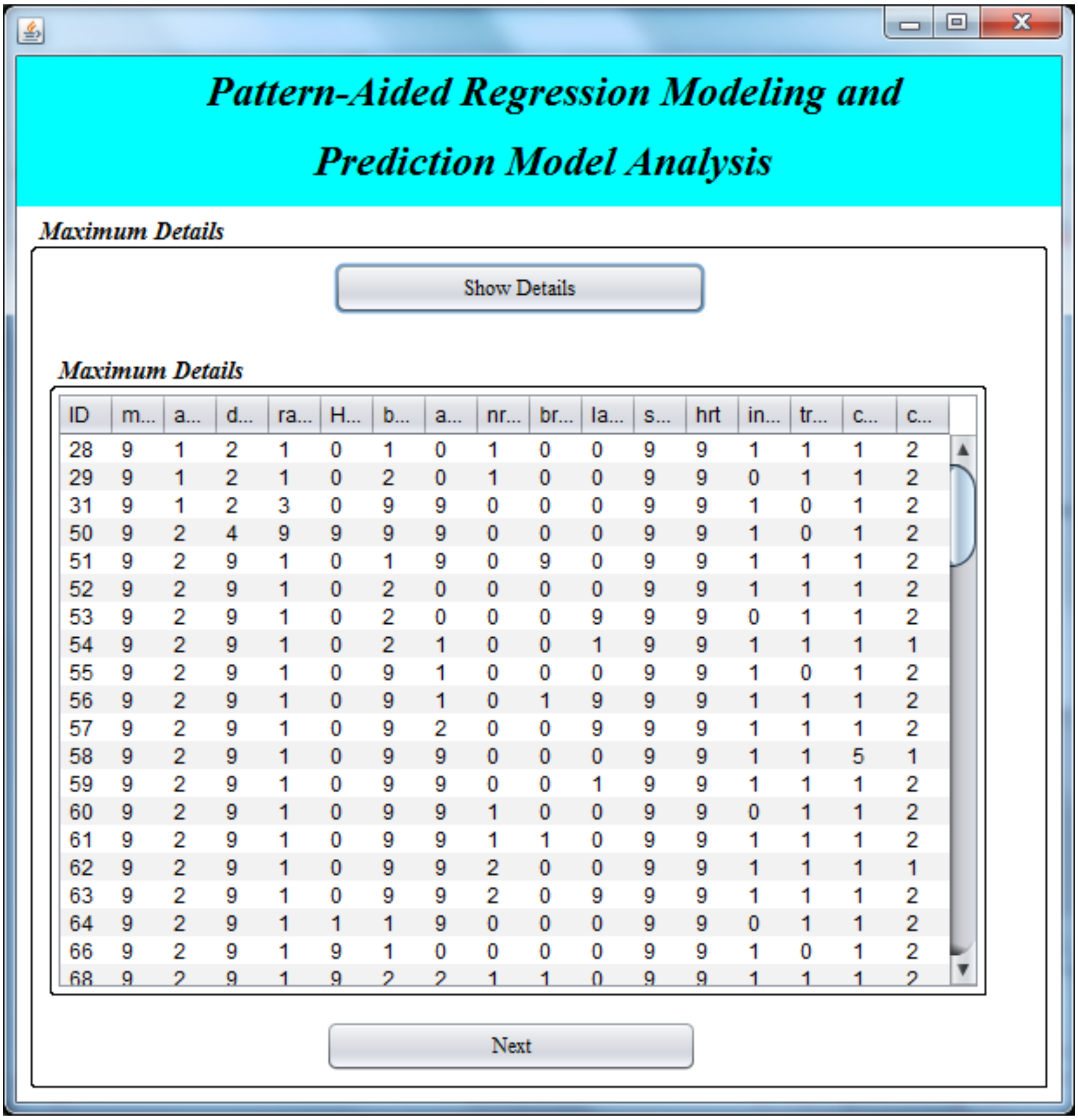


Figure 27. Maximum Probability Frame



ID	m...	a...	d...	ra...	H...	b...	a...	nr...	br...	la...	s...	hrt	in...	tr...	c...	c...
28	9	1	2	1	0	1	0	1	0	0	9	9	1	1	1	2
29	9	1	2	1	0	2	0	1	0	0	9	9	0	1	1	2
31	9	1	2	3	0	9	9	0	0	0	9	9	1	0	1	2
50	9	2	4	9	9	9	9	0	0	0	9	9	1	0	1	2
51	9	2	9	1	0	1	9	0	9	0	9	9	1	1	1	2
52	9	2	9	1	0	2	0	0	0	0	9	9	1	1	1	2
53	9	2	9	1	0	2	0	0	0	9	9	9	0	1	1	2
54	9	2	9	1	0	2	1	0	0	1	9	9	1	1	1	1
55	9	2	9	1	0	9	1	0	0	0	9	9	1	0	1	2
56	9	2	9	1	0	9	1	0	1	9	9	9	1	1	1	2
57	9	2	9	1	0	9	2	0	0	9	9	9	1	1	1	2
58	9	2	9	1	0	9	9	0	0	0	9	9	1	1	5	1
59	9	2	9	1	0	9	9	0	0	1	9	9	1	1	1	2
60	9	2	9	1	0	9	9	1	0	0	9	9	0	1	1	2
61	9	2	9	1	0	9	9	1	1	0	9	9	1	1	1	2
62	9	2	9	1	0	9	9	2	0	0	9	9	1	1	1	1
63	9	2	9	1	0	9	9	2	0	9	9	9	1	1	1	2
64	9	2	9	1	1	1	9	0	0	0	9	9	0	1	1	2
66	9	2	9	1	9	1	0	0	0	0	9	9	1	0	1	2
68	9	2	9	1	9	2	2	1	1	0	9	9	1	1	1	2

Figure 28. Maximum Probability Details Frame

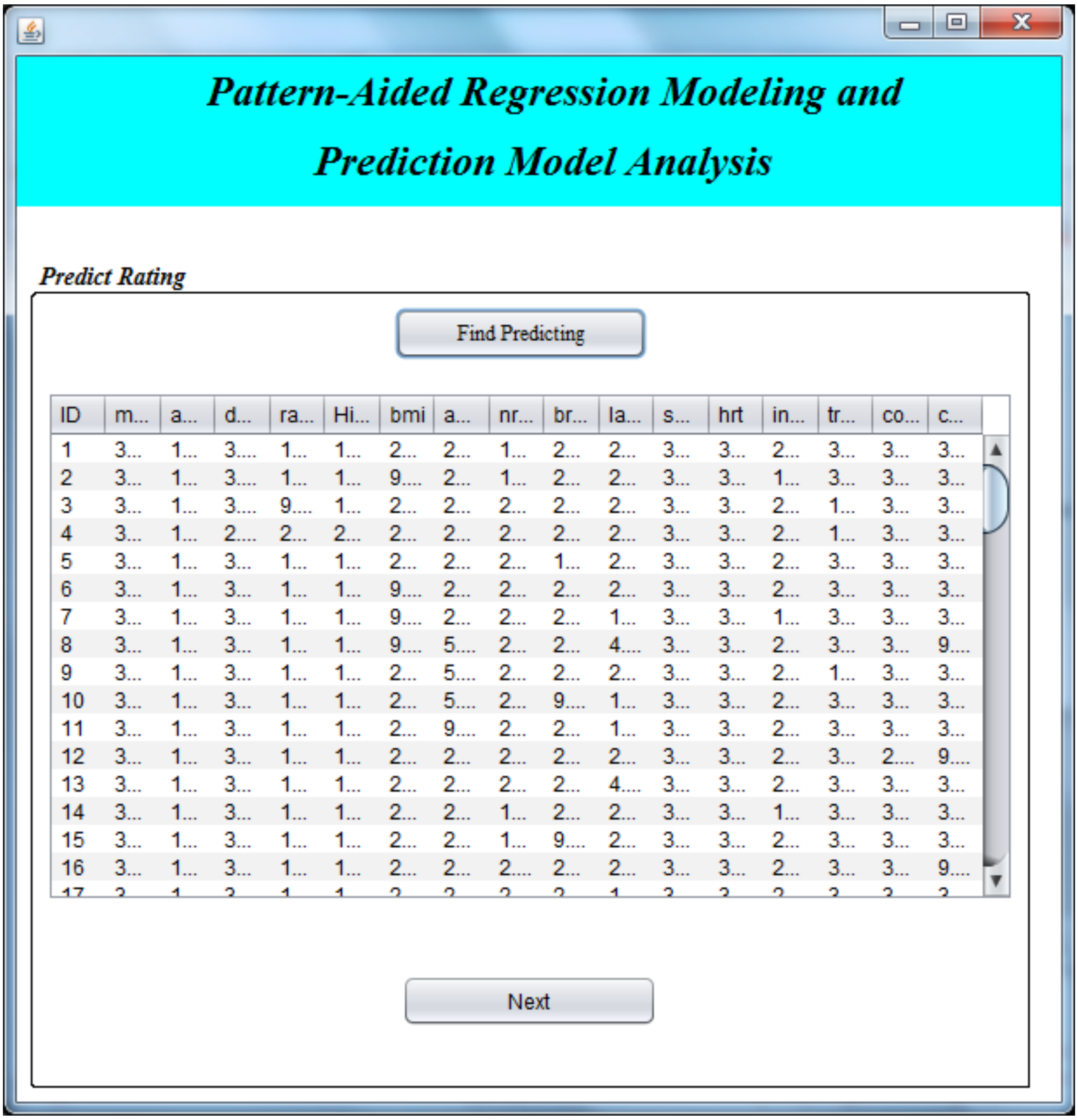


Figure 29. Find Predict Rating for Maximum Values Frame

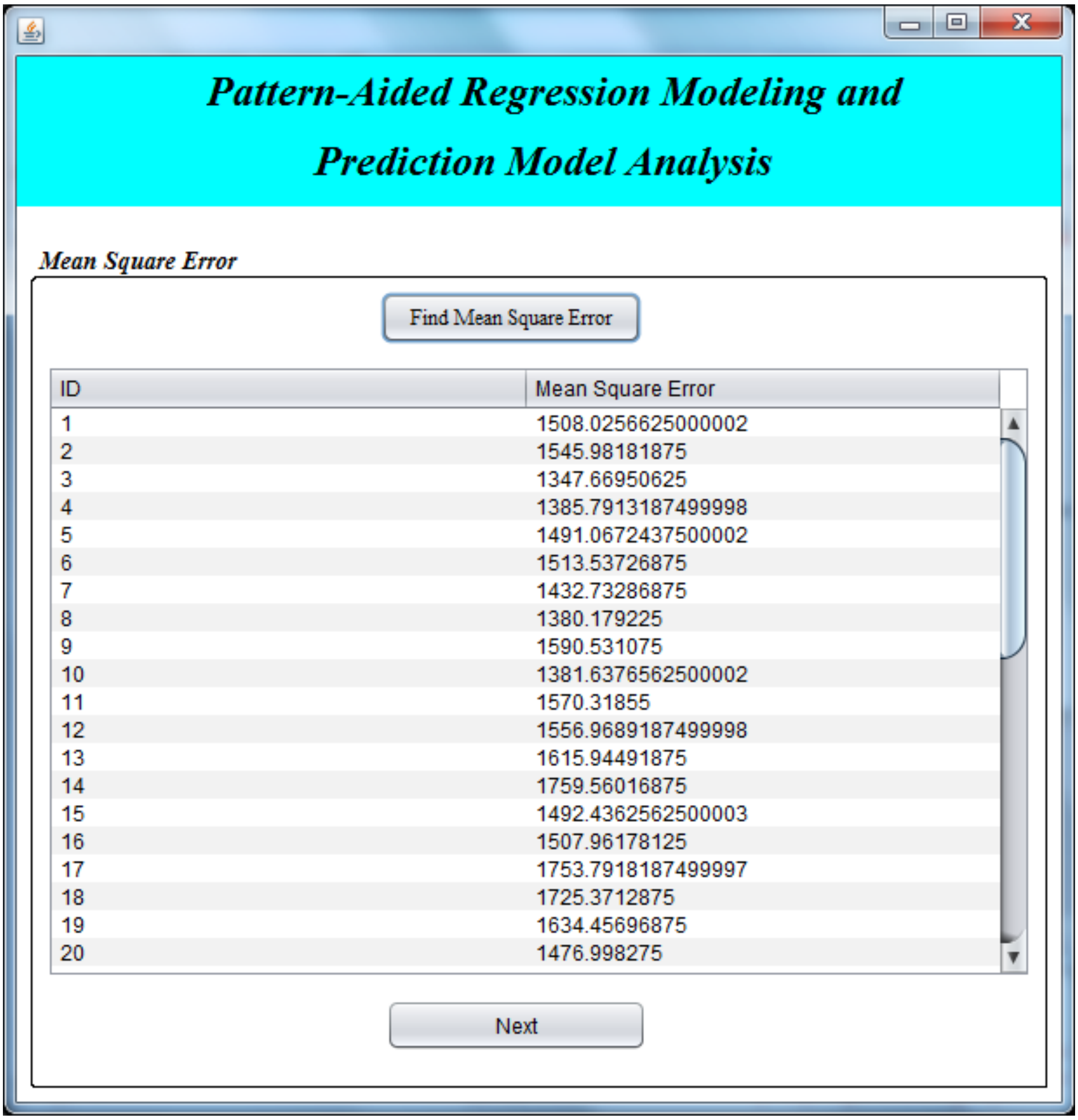


Figure 30. Find Mean Square Error for Maximum Values Frame

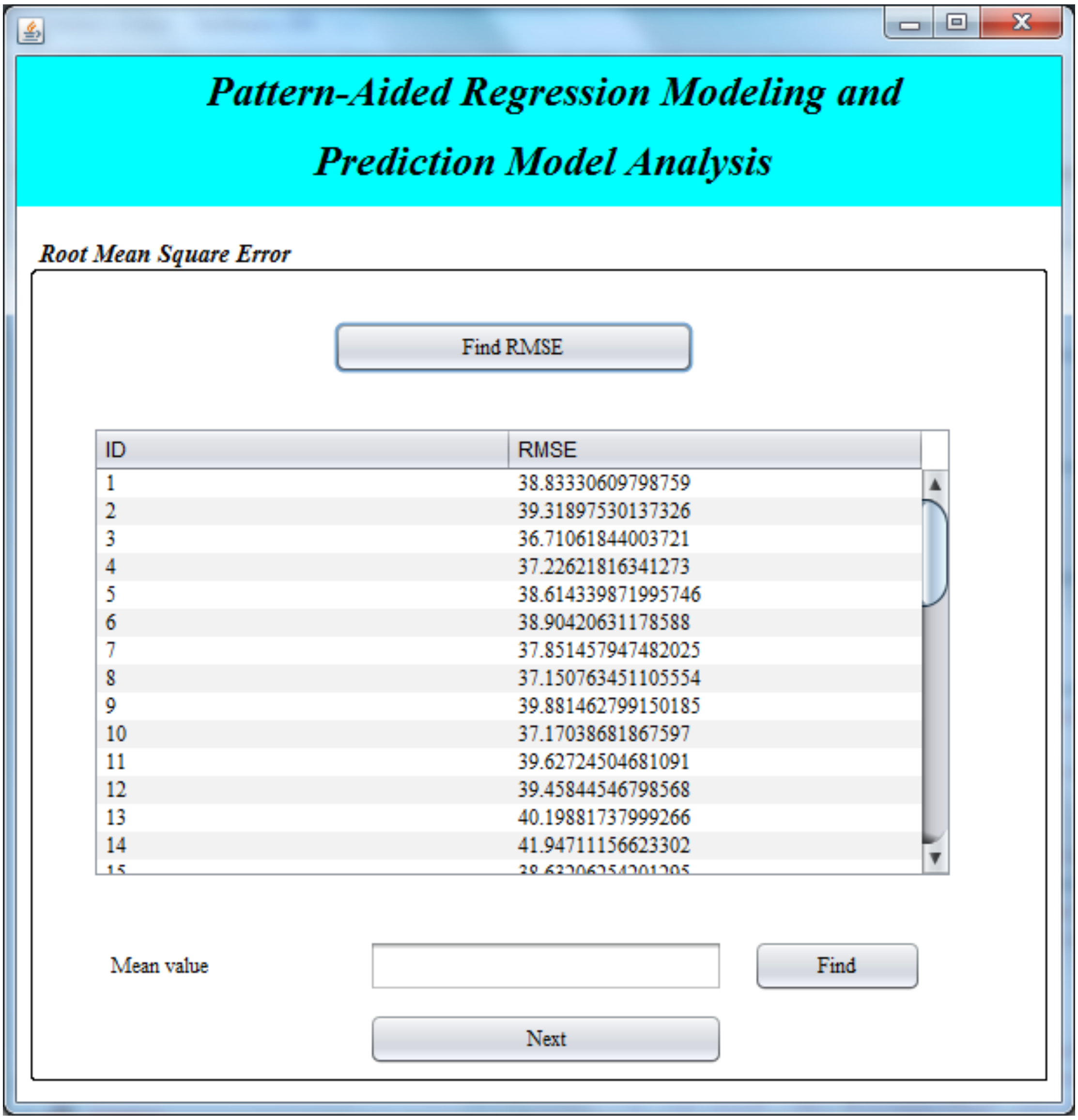


Figure 31. Find Root Mean Square for Maximum Values Frame

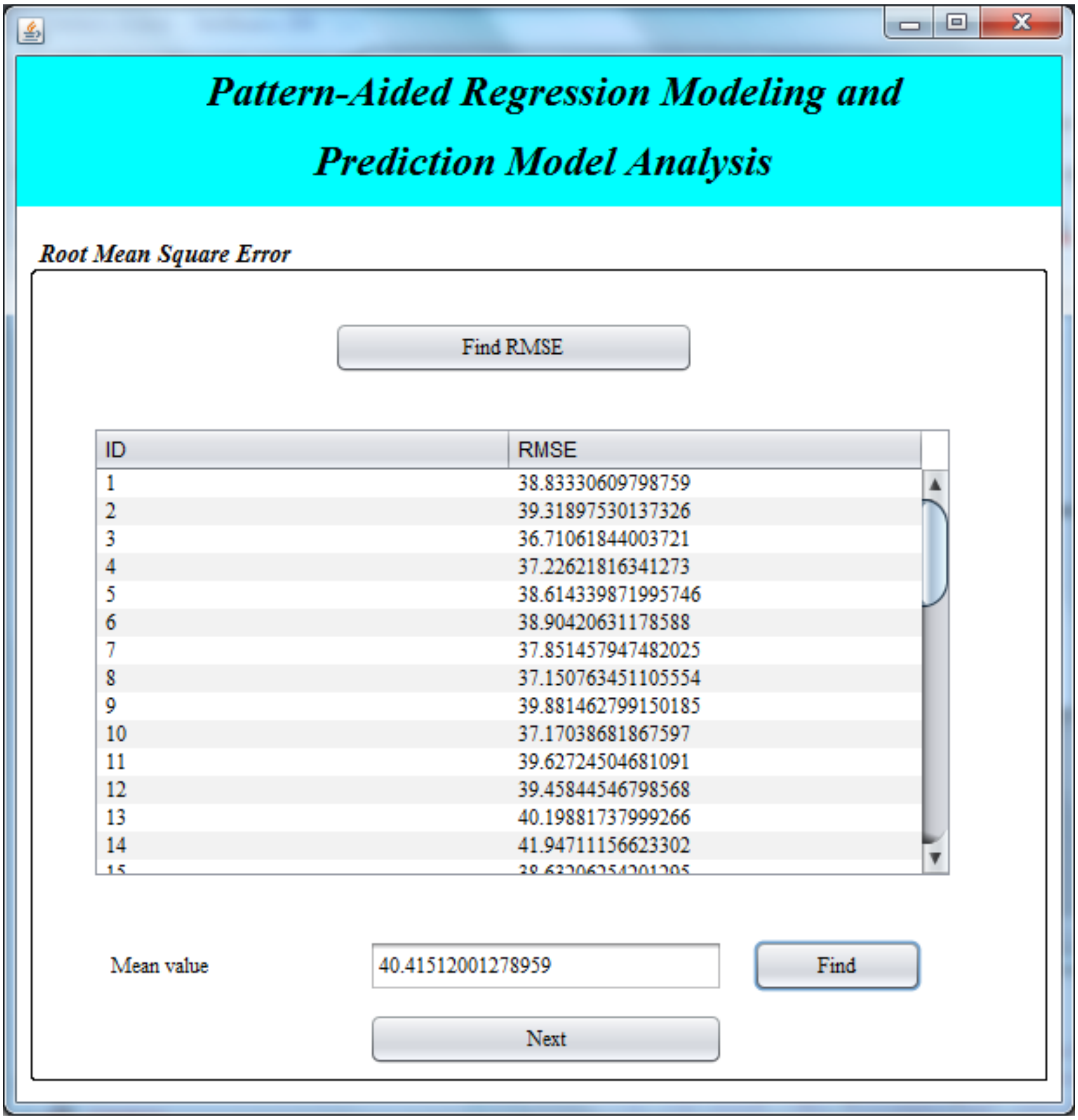


Figure 32. Find Mean Value in RMSE for Maximum Values Frame

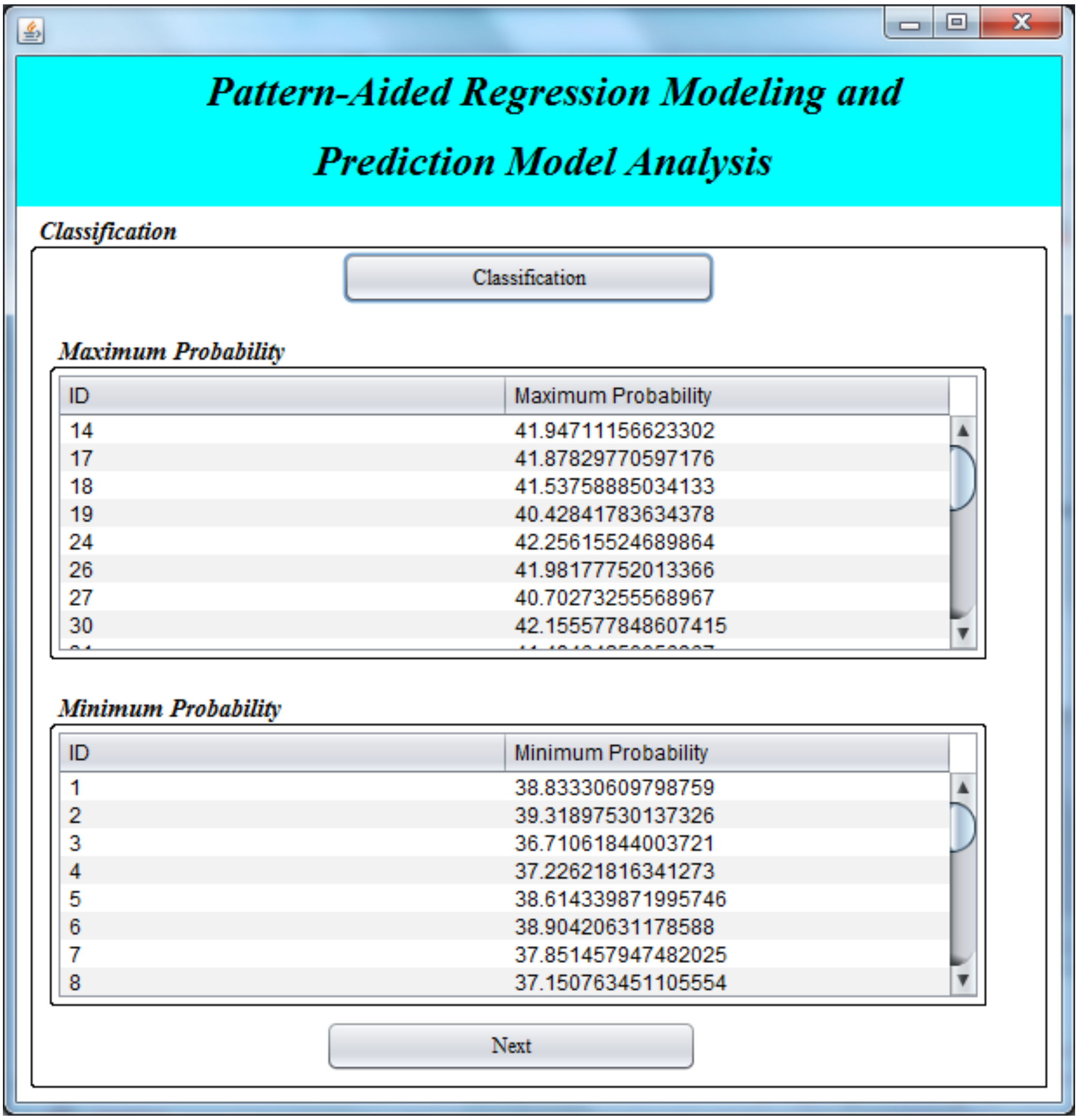


Figure 33. Find Classification for Maximum Values Frame

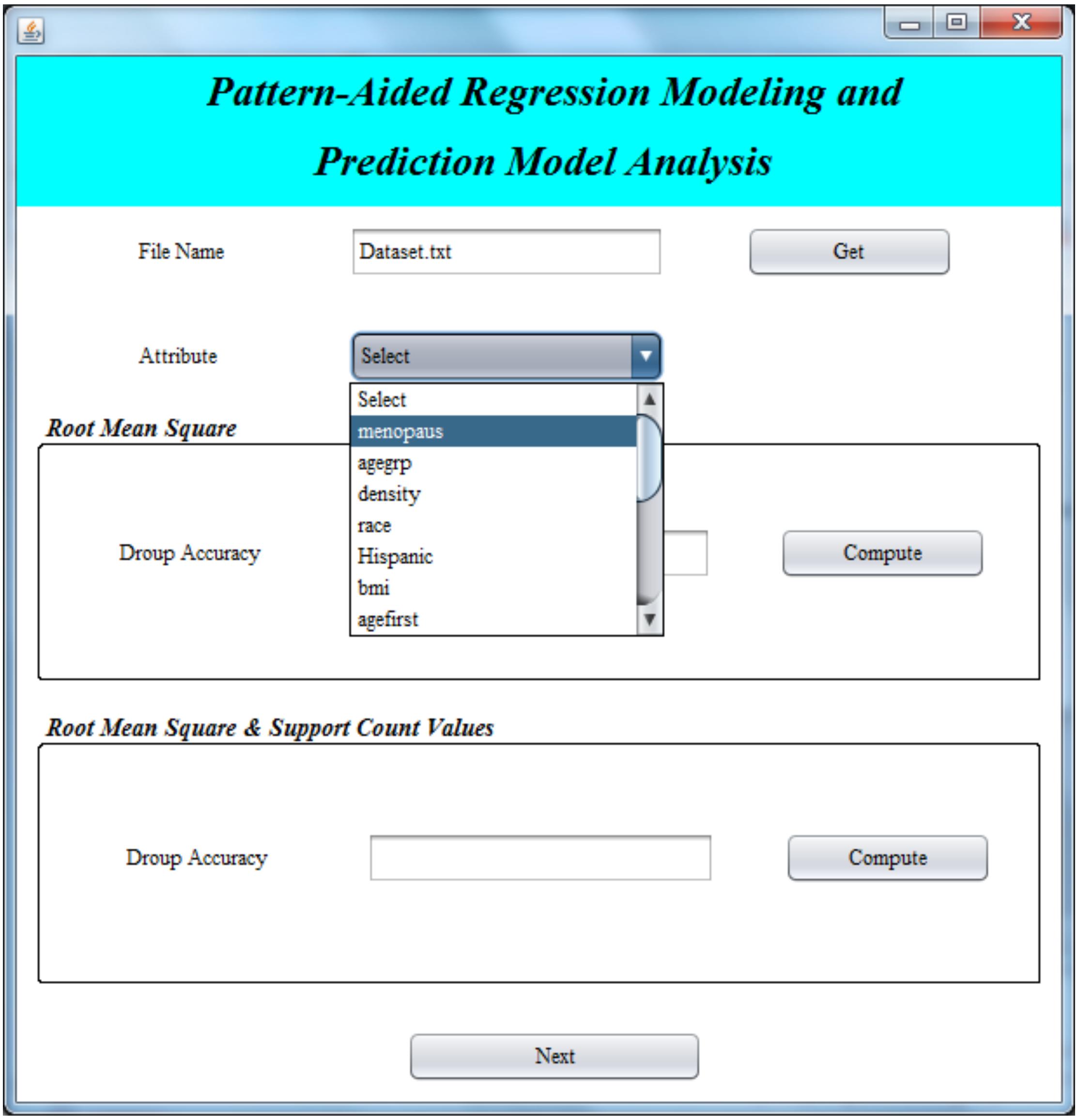


Figure 34. Find Drop in Accuracy for Attributes in RMSE and RMSE with Support Count Values Frame 1

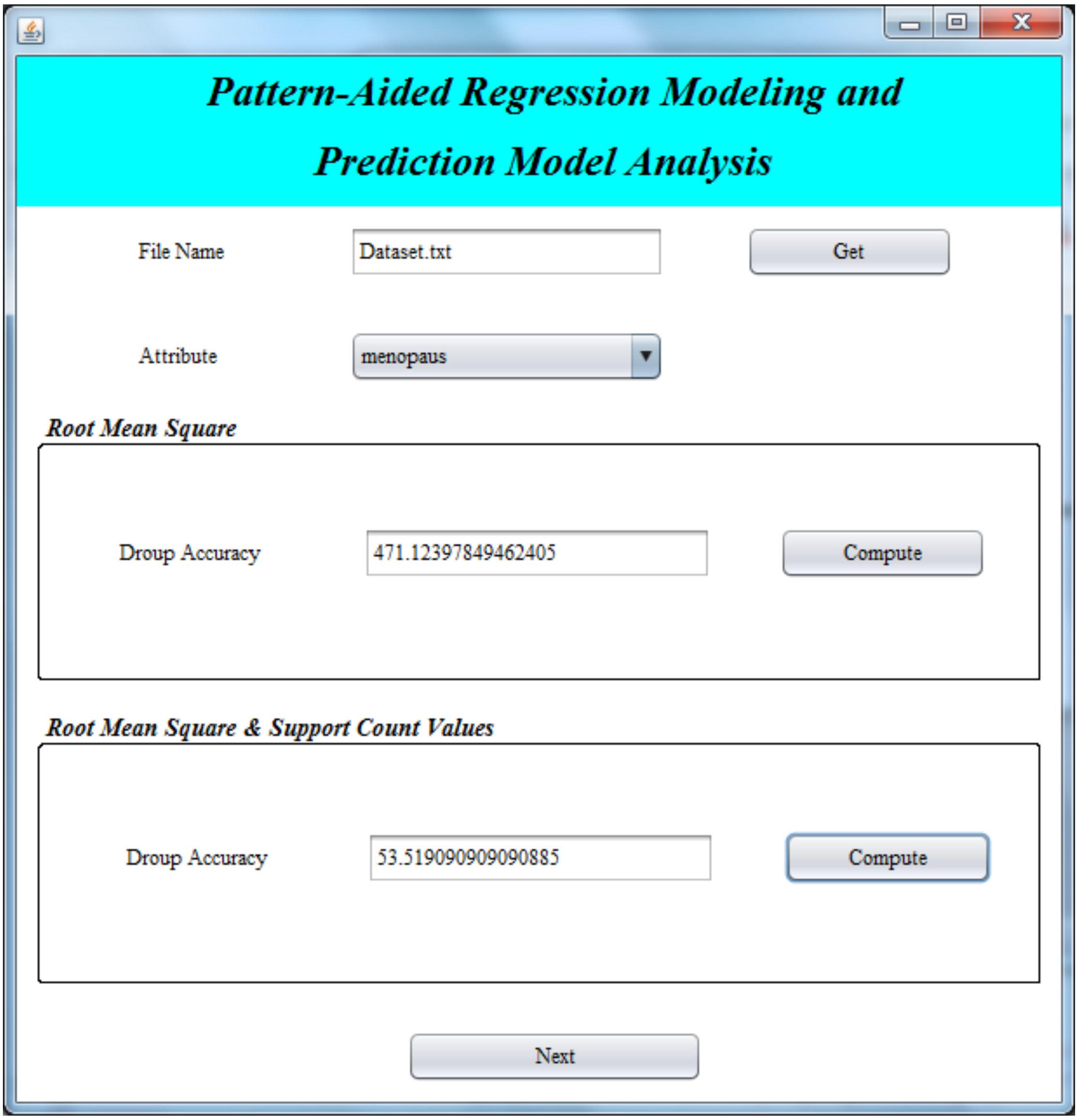


Figure 35. Find Drop in Accuracy for Attributes in RMSE and RMSE with Support Count Values Frame 2

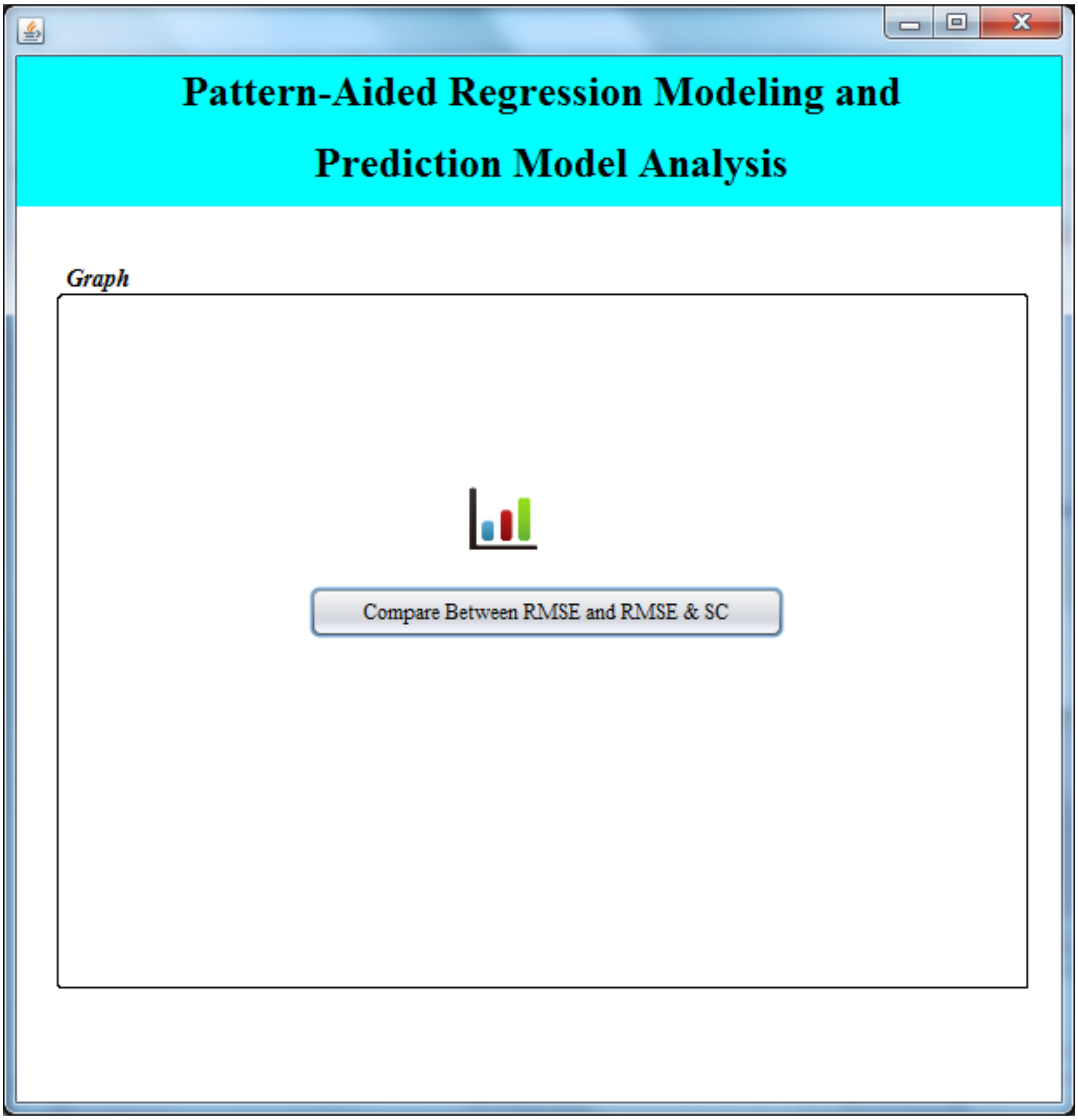


Figure 36. Find Comparison between RMSE and RMSE with Support Count Values Frame

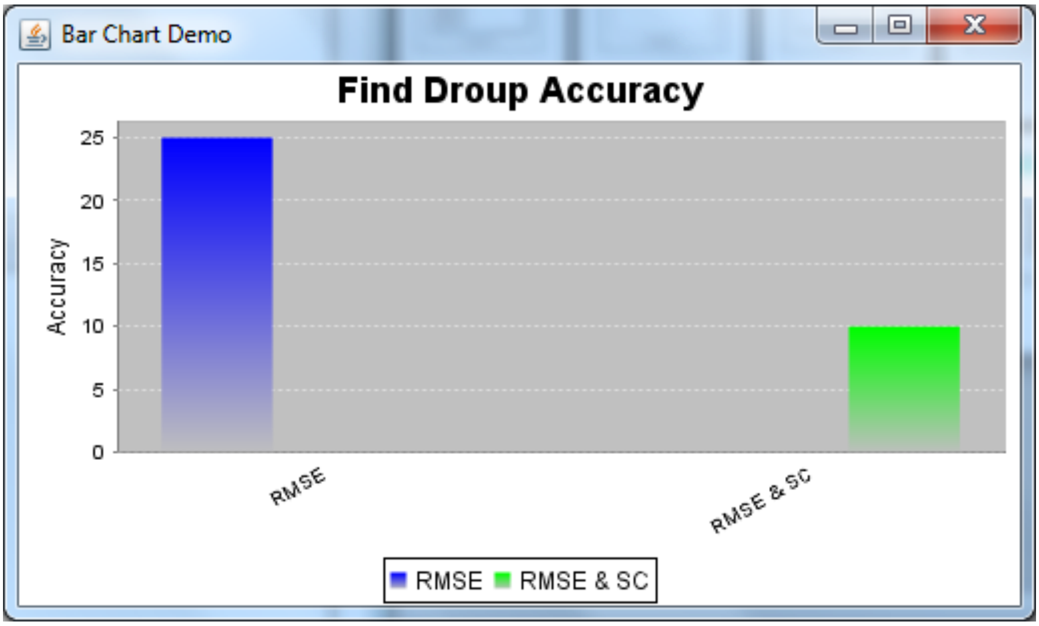


Figure 37. Comparison Graph between RMSE & RMSE with Support Count Values Frame