Global Optimization Ensemble Model for Classification Methods

by

Hina Anwar

2010-NUST-MS PHD- CSE-26

MS-10 (SE)

Submitted to the Department of Computer Engineering in fulfillment of the requirements for the degree of

MASTER OF SCIENCE
in
SOFTWARE ENGINEERING

Thesis Supervisor

Dr. Usman Qamar

College of Electrical & Mechanical Engineering

National University of Sciences & Technology

بسم الله الرحمن الرحيم

In the name of Allah, the most

Beneficent and the most Merciful

DECLARATION

I hereby declare that I have developed this thesis entirely on the basis of my personal efforts under the sincere guidance of my supervisor Dr. Usman Qamar. All the sources used in this thesis have been cited and the contents of this thesis have not been plagiarized. No portion of the work presented in this thesis has been submitted in support of any application for any other degree of qualification to this or any other university or institute of learning.

Student Signature

ACKNOWLEDGEMENTS

Innumerable words of praise and thanks to Allah, the Almighty, and the Creator of the universe for carving the path for me and always helping me out in the best possible way. Without His Will and Mercy, I would not have been able to accomplish this milestone. I am grateful to my parents for their immense love, moral support, encouragement and prayers throughout my academic career.

I am deeply beholden to my supervisor, Dr. Usman Qamar, for his continuous guidance, inspiration, and patience. His ability of management and foresightedness taught me a lot of things which will be more helpful for me in my practical life. I am highly thankful to Head of Department Dr. Shoab Ahmad khan, for his continuous and valuable suggestions and guidance, especially for the provision of all kinds of facilities during my thesis work.

I gratefully acknowledge the help and guidance provided by Guidance and Examination Committee members Dr. Saad Rehman, Dr. Arslan Shaukat, Dr. Shoab Ahmad Khan. Their valuable suggestions and comments were a great source to improve the research work presented in this thesis.

DEDICATION

To my parents and teachers

ABSTRACT

Data mining is the process of knowledge discovery and extraction of useful information and pattern from raw data gathered from various resources and supervised learning is the process of data mining for deducing rule from marked training dataset. A broad array of supervised learning algorithms exists, every one of them with its own advantages and drawbacks. For supervised learning problems there is still no single algorithm that works ideally. There are some basic issues that affect the accuracy of classifier while solving a supervised learning problem like bias-variance tradeoff, dimensionality of input space and noise in the input data space. All these problems affect the accuracy of classifier and are the reason that there is no global optimal method for classification. Neither is there any generalized improvement method that can increase the accuracy of any classifier while addressing all the problems stated above. The objective of this paper is to create a global optimization ensemble model for classification methods (GMC) that can improve the overall accuracy for supervised learning problems. The experimental results on various public datasets showed that the proposed model improved the accuracy of the classification models from 1% to 30% depending upon the algorithm and dataset.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1: INTRODUCTION

CHAPTER 1: INTRODUCTION

According to Han and Kamber, “Data mining is known to be a part of knowledge discovery (KDD) process in which data is analyzed and summarized from different perspectives and converted into useful information. It helps in extracting the hidden and valid data which has the potential of being transformed into useful information. The objective of data mining process is to convert raw data in useful information that is helpful in making future predictions and informed business decisions. Data mining is carried out using various techniques, but most important and commonly used technique is Classification.” [1] It is similar to machine learning process and can also be termed as supervised learning process. Supervised learning is the process of data mining for deducing rule from marked training dataset. A broad array of supervised learning algorithms exists, every one of them with its own advantages and drawbacks. For supervised learning problems there is still no single algorithm that works ideally. In Classification the first step is to divide the data in two portions known as training set and testing set [2]. In these datasets, one attribute must be necessarily defined as class label besides all other attributes. According to Jiawei Han [2], the two steps of the classification task are model construction and model usage. In this task, the model is build with the help of trained dataset and then this trained model is used to allocate The unseen records as precisely as possible. While training data set is used to build and train the model the testing data set is use to validate and test the model accuracy [3]. Which bring us to some of the basic issues that affect the accuracy of a classifier while solving a supervised learning problem. For instance, the bias-variance tradeoff, the dimensionality curse or the noise in the dataset all contribute towards a decreasing accuracy. Bias arises when the classifier cannot represent the true function – that is, the classifier under fits the data i.e. when it is training on any data set than for a specific input value it is methodically inaccurate when predicting the right outcome for that input value. In contrast to this, variance occurs when the algorithm over fits the data and for a specific input value in a dataset it gives a different outcome every time the training dataset is changed. Another problem that can affect the accuracy of a classifier is the dimensionality or the number of attributes or features in a dataset. If we input a large number of attributes in a classification algorithm even for problems where decision depends on subset of all those attributes, than performance of the classifier will be clouded by high variance due to high dimension of dataset. Therefore if a dataset with high dimension is being used the classifier must be tuned to make a tradeoff between high bias and low variance. The classification results are also altered by the noise in data i.e. redundant records, incorrect records, missing records, outliers etc. All these problems affect the accuracy of a classifier. Usually the improvements done in a classifier or ensemble model are limited to a very narrow spectrum and they cannot be applied to another classifier under same conditions. Classification accuracy is normally improved through ensemble models like bagging (Which averages the prediction of a number of classification models), boosting (it uses the voting scheme over a number of classification models), or a combination of classifiers from different or same families as discussed in chapter 2.

Therefore, the aim of this research is to propose a global optimization model using the idea of ensemble models for classification methods and prove through experimental results that the proposed model improves the classification accuracy of various classifiers on various different public datasets.

1.1. Motivation

A lot of supervised learning methods have been developed each with its own pros and cons and classification accuracy rate. And a lot of optimization and improvement techniques and have been suggested to solve the basic problems in these supervised learning methods. However so far there is no global model available that can solve all the problems to some extent and improve the classification accuracy rate

1.2. Background

As it will be discussed in chapter 2 various researchers like Sujata Dash et al, R. Bryll et al and many others have tried to improve the classification accuracy of different classifiers from different aspects. Lin et al. focused on feature reduction, while D. W. Abbott, S.Y. Sohn et al. tried to made a better bias-variance tradeoff. M. R. Smith et al. tried to deal with the problem of noise and outliers in order to improve classification accuracy rates. However as shown in chapter 2, there exists no global model to improve the accuracy of a classifier.

1.3. Methodology

In this project we would follow the method used in ensemble models to design a new global optimized model for classification methods.

1.4. Structure of the Thesis:

Chapter 2 provides background to the current research. It starts by defining the concepts of Dimensionality, Bias, Variance, Noise etc. This is followed by a discussion on all the latest work published related to the above mentioned concepts and classification accuracy rate.

Chapter 3 describes the detailed design process for the proposed model. The aim of this research is to develop a global optimization model for classification methods.

Chapter 4 consists of the complete implementation details of the model t along with the brief description of the software along with the details of the environment in which the project was done. (Dataset details and Code were covered in more detail in the appendices.)

Chapter 5 provides the experimental results followed by a discussion of the results.

Chapter 6 summarizes the work done as well as the main results of the thesis and suggests future work that may be of interest. It also outlines various applications for the global model presented in this work.

CHAPTER 2: LITERATURE REVIEW

CHAPTER 2: LITERATURE REVIEW

2.1 Background:

As mentioned earlier that so far no global optimization ensemble model is present which can help in improving the classification and prediction accuracy for supervised learning problems which are generally affected by a spectrum of issues like dimensionality, accuracy rate with respect to bias and variance, data quality etc.

2.1.1 Dimensionality Curse:

Curse of dimensionality can be explained as: with a fixed size of training sample the prediction accuracy decreases if the dimensionality or number of feature increases. If the sample size is small and number of feature is large than results will not be accurate. If m is the sample size and b is the number of features than m must be greater than b^2 in order for the prediction to be accurate. Otherwise over fitting will occur; means the model will have high variance. This means that for a fixed sample size of data there exist a optimal number of features where the model for classification will perform better. If there are two features and sample size is 10 than we need 10^2= 100 sample size for correctly training the model similarly for 3 features it would be 10^3

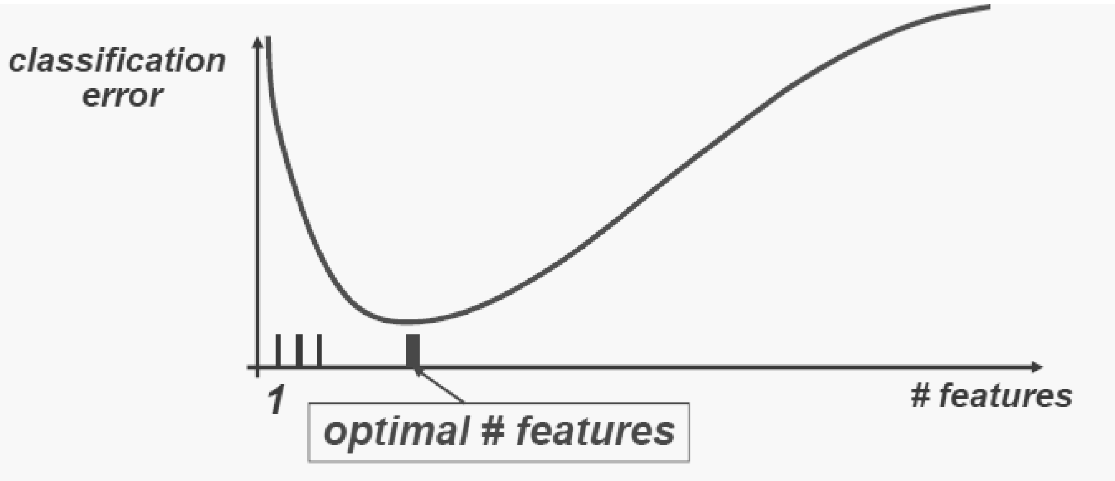


Figure 1: Graph: Classification Error and Optimal Features [1]

2.1.2 Bias-Variance Tradeoff:

When discussing the classification and prediction errors in supervised learning problems we are actually referring to the errors caused by bias and variance. There is always a tradeoff between these bias and variance and the error value depends on the ability of a model to make this trade off and avoid the over and under fitting of data. Bias is the difference between the predicted value for data point by the model and the correct predication. i. e. Bias arises when the classifier cannot represent the true function – that is, the classifier under fits the data i.e. when it is training on any data set than for a specific input value it is methodically inaccurate when predicting the right outcome for that input value. In contrast to this, error due to variance occurs when a model is predicting variable values for give data point i.e. variance occurs when the algorithm over fits the data and for a specific input value in a dataset it gives a different outcome every time the training dataset is changed. Both bias and variance are increased and decreased with respect to the model complexity, which means that every algorithm builds a different model with different complexity with respect to the values in dataset hence the bias-variance trade-off for each algorithm, is different for different dataset. But in general what the researcher are looking at is the overall error not the decomposition of error in these two components. The point or level of complexity of a model where the decrease in bias is equal to increment in variance can be termed as the optimal point for that model. Going over this point causes over-fitting or high variance and staying under this point causes under-fitting or high bias.

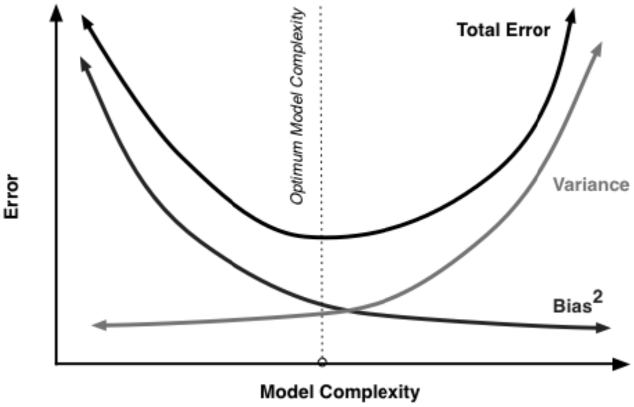


Figure 2: Bias Variance Trade-off with respect to the model complexity [2]

2.1.3 Noise in Data:

Noise cannot be defined properly as its definition varies from data to data, problem to problem and domain to domain. Noise could be the redundant data, irrelevant data, missing data, outlier etc. Noise in the data also tends to over fit the model.

2.2 Previous Work

Although, no global solution exists for the problems stated above but some other efforts have been made to resolve these issue and all of them are either algorithm specific or data specific. Every approach has tackled the problem of classification accuracy rate from a different angle and perspective. One such work is [4] where Sujata Dash et al. have proved through comparison of various classification techniques like Support Vector Machine (SVM) with polynomial Kernel, Support Vector Machine with RBF Kernel, Radial Basis Function Network (RBFN), Multi-Layer Perceptron network (MLP); with and without feature extraction. It was found that for construction of high performance classification model for microarray dataset, partial least square (PLS) regression method is the suitable feature selection method instead of hybrid dimensionality reduction scheme and feature selection combined with various classification techniques can yield better results.

Table 1: Predictive error (%) of classification algorithms, using SIMPLS Dimensionality Reduction Scheme [4]

Dataset	RBFN	Polynomial SVM	RBF SVM	MLP
Leukemia	0	0.45	28.22	0.41
Colon Cancer	10.95	0	23.33	0.31
Lung Cancer	11.55	0	16	0.95

The classification error rate of all the three dataset indicate that all dataset responded favorably to variable pre-selection for all the classifiers except few exceptions and the predictive accuracy is extremely high something like 100% for SIMPLS-SVM- Polynomial model for Colon data set, SIMPLS-RBFN model for Leukemia data set and SIMPLS-SVM-

Polynomial model for Lung data set. The Leukemia and Colon cancer datasets indicate they were not largely affected by variable pre-selection for SIMPLS-SVM-RBF model and Colon and Lung data sets for SIMPLS-RBFN model and achieve predictive accuracy of approximately72% and 88% respectively

Lin et al. in [5] combined PSO (particle swarm optimization) -based approach with commonly used classification technique LDA (Linear Discriminant Analysis). This research also emphasizes the importance of feature selection and its positive effect on classification accuracy. Author of this study have compared the performance of this combined model called PSOLDA with many other feature selection techniques like forward selection, back propagation selection etc and shown through experimental results that for many public datasets the proposed combined model (PSOLDA) have higher classification accuracy rate. The results in this research work were obtained by comparing the classification accuracy rates obtained by LDA without feature selection, LDA with forward feature selection, LDA with backward feature selection, LDA with PCA-based feature selection, LDA feature selection by exhaustive enumeration and PSOLDA. The data sets used for testing included Australian dataset, Bioinformatics Dataset, Boston Housing Dataset, Heart Dataset, Cancer Dataset etc. and it was found that PSOLDA classification accuracy rate for each dataset was found to be the most optimal as shown in table 2 below.

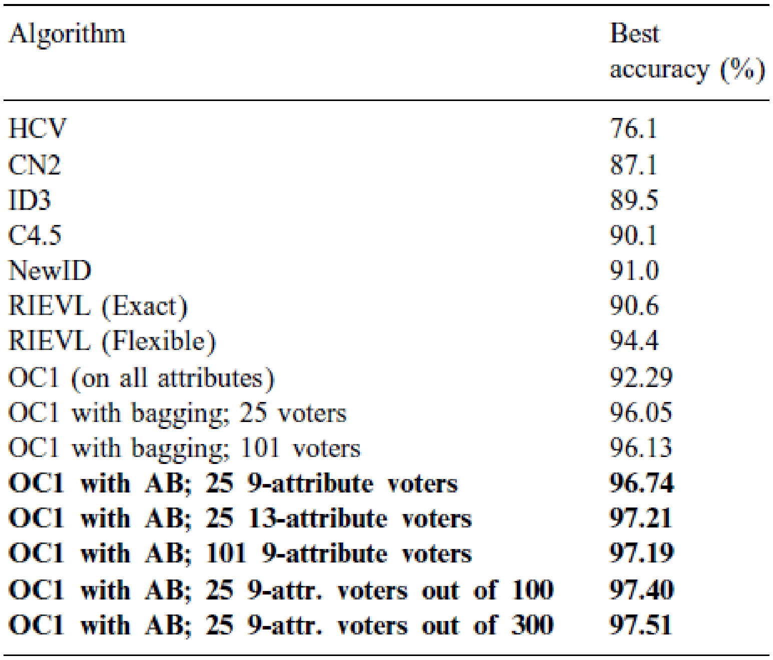
Table 2: Comparison of classification accuracy: LDA with and without forward selection, backward selection, PCA, Exhaustive enumeration and PSOLDA [5]

Dataset	LDA without feature selection	LDA with forward feature selection	LDA with backward feature selection	LDA with PCA-based feature selection	LDA with feature selection by exhaustive enumeration	PSOLDA	Time (s)
Australian	83.0%	80.4%	84.2%	82.1%	84.5%	84.5%	19.09
Bioinformatics	80.4%	79.3%	81.6%	80.4%	-	84.4%	18.23
Boston housing	83.8%	82.8%	84.3%	83.8%	85.2%	85.2%	20.10
Breast Cancer	96.1%	95.4%	95.8%	91.7%	96.5%	96.5%	10.04
Bupa live	63.5%	61.1%	64.3%	60.5%	65.2%	65.2%	3.48
Car Evaluation	79.3%	71.2%	78.0%	81.8%	78.1%	78.1%	29.00
Cleveland Heart	74.2%	78.9%	83.3%	77.6%	84.7%	84.7%	7.29
Dermatology	81.6%	96.5%	97.0%	93.3%	-	98.4%	37.09
Ecoli	42.5%	79.7%	79.7%	72.3%	80.1%	80.1%	6.00

Table 3: Number of features selected by each technique [5]

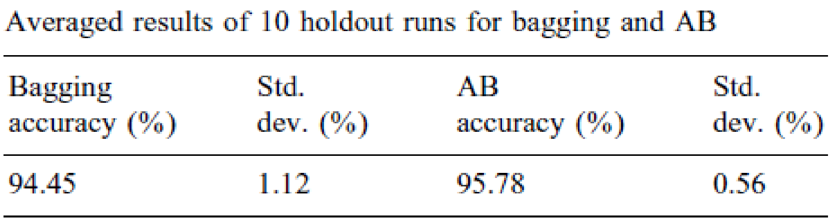
Dataset	No. of original features	LDA with forward feature selection	LDA with backward feature selection	LDA with PCA-based feature selection	PSOLDA
Australian	15	5.6	13	13.2	11.4
Bioinformatics	20	9.5	18.2	19.5	15.7
Boston housing	13	5.1	9.9	13.9	7.7
Breast Cancer	10	6.0	6.7	9.7	6.6
Bupa live	6	6.3	4.7	5.6	4.6
Car Evaluation	6	4.9	5.6	5.3	5.4
Cleveland Heart	13	6.1	11.7	12.7	9.5
Dermatology	34	17.7	26.9	28.4	22.3
Ecoli	7	5.5	5.6	6.2	5.6

R. Bryll et al [6] developed a new wrapper method AB (Attribute Bagging) to improve the classification accuracy implementing a two stage method in which first a suitable size was provided for training data and then randomly a subset of attributes were selected for voting scheme. This method was compared with bagging which was used with some decision tree algorithms and some rule induction algorithms, and it was found the AB performs better in terms of accuracy and constancy. And authors conclude that attribute partitioning is better than data partitioning for improving the accuracy in an ensemble method. Holdout method has been used instead of cross-validation. Means in each run the training and test data points are same only the attributes are randomly selected. The results are shown in figure 1. OC1 (oblique Classifier) is an algorithm for building oblique decision trees. The results show that OC1 in combination with the attribute bagging gives best accuracy rate as compared to when OC1 is used with bagging. Also using one single technique with full range of attributes does not yield best result as far as classification accuracies are concerned. Average results are recorded after 10 runs of hold out methods for attribute bagging and simple bagging as shown in figure 2. While average of 5 runs of hold out method is recorded for simple OC1 and attribute bagging; as shown in figure 3.



Algorithm	Best accuracy (%)
HCV	76.1
CN2	87.1
ID3	89.5
C4.5	90.1
NewID	91.0
RIEVL (Exact)	90.6
RIEVL (Flexible)	94.4
OC1 (on all attributes)	92.29
OC1 with bagging; 25 voters	96.05
OC1 with bagging; 101 voters	96.13
OC1 with AB; 25 9-attribute voters	**96.74**
OC1 with AB; 25 13-attribute voters	**97.21**
OC1 with AB; 101 9-attribute voters	**97.19**
OC1 with AB; 25 9-attr. voters out of 100	**97.40**
OC1 with AB; 25 9-attr. voters out of 300	**97.51**

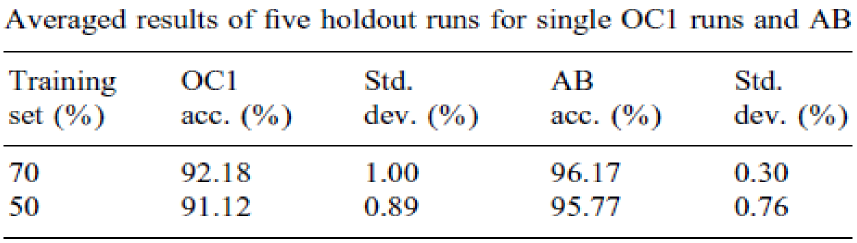
Figure 3: Comparison of best accuracies achieved by various algorithms on hand-pose database [6]



Averaged results of 10 holdout runs for bagging and AB

Bagging accuracy (%)	Std. dev. (%)	AB accuracy (%)	Std. dev. (%)
94.45	1.12	95.78	0.56

Figure 4: Results of Attribute bagging and simple bagging [6]



Averaged results of five holdout runs for single OC1 runs and AB

Training set (%)	OC1 acc. (%)	Std. dev. (%)	AB acc. (%)	Std. dev. (%)
70	92.18	1.00	96.17	0.30
50	91.12	0.89	95.77	0.76

Figure 5: Results of simple OC1 and attribute bagging [6]

D. W. Abbott [7] compared boosting with an ensemble of models across the algorithm families. These combined models used voting as the selection scheme and authors report that boosting performs better because it focus on complicated cases in data and take into account the confidence value of a particular classification decision.

S.Y. Sohn et al. [8] tried to improve the classification accuracy of algorithms like neural network and decision trees by applying different approaches including bagging, boosting and clustering. However for the particular problem of road traffic accident classification clustering leading to classification was found to be more effective.

Algorithm	Accuracy (%)	The number of classifier
Decision Tree	72.30	1
Neural network	70.86	1
Dempster-Shafer	72.79	2
Bayesian	71.23	2
Logistic fusion	72.30	2
Bagging (neural net)	72.70	5
Bagging (Decision tree)	74.78	5
Clustering method(neural net	73.94	3
Clustering method (decision tree)	76.10	3

Figure 6: Classification accuracy of algorithms [8]

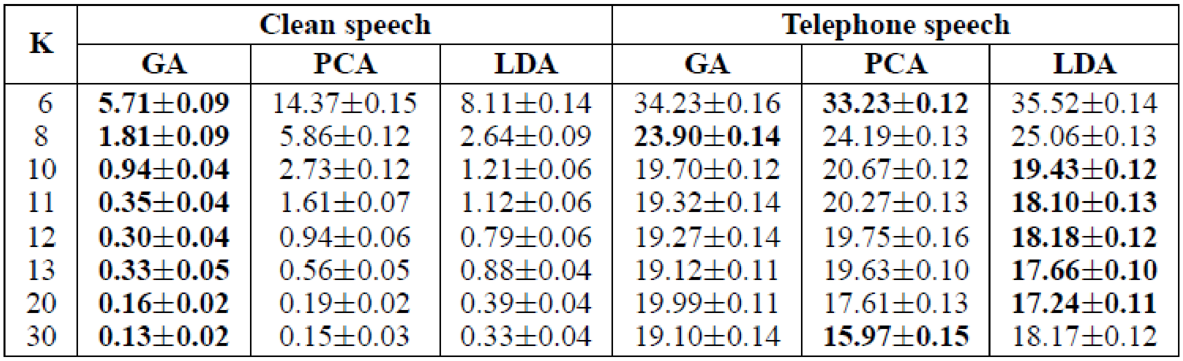
M. R. Smith et al. [9] suggested that outliers and noise should be eliminated from the dataset as it will yield better results in terms of classification accuracy. Because by removing or filtering these instances the dataset becomes clean of all the cases that could be misclassified. As there is no general definition or guide available as to what noise is and what an outlier is therefore the identification of these two elements in any dataset is difficult. Furthermore PRISM was found to be one of the best algorithms for finding cases that could be outliers. Dimensionality reduction problem has been an interesting topic for researchers in a diverse spectrum of fields like image detection, voice detection, microarrays, neural network patterns etc.

Table 4: The average classification accuracy for each learning algorithm trained with and without filtering [9]

	Orig	Dist	LOF	ECODB	RENN	PRISM
C4.5	0.803	0.794	0.802	**0.807**	**0.805**	**0.809**
IB1	0.771	0.773	0.773	0.784	**0.809**	0.797
IB5	0.791	0.789	0.793	0.802	**0.822**	0.814
MLP	0.813	0.814	0.814	0.822	**0.829**	**0.831**
NB	0.765	**0.733**	0.767	**0.722**	**0.774**	**0.776**
Percept	0.801	0.803	0.798	**0.808**	**0.811**	**0.812**
RBFNet	0.796	0.791	0.792	0.797	**0.807**	**0.806**
RIPPER	0.787	0.787	0.788	0.792	0.790	**0.798**
SVM	0.805	0.803	0.801	**0.810**	0.808	**0.814**
Overall	0.792	0.792	0.792	0.799	**0.806**	**0.806**

The increase in accuracy was about 1.3%. However, on data sets where more than 10% of the instances are ISMs (instances that should be misclassified), the increase on average is 2.8% compared to 1.2% for data sets with less than 10% ISMs. Rather than focusing on correctly classifying the instances that should be misclassified and arbitrarily adjusting the classification boundary, removing the ISMs for training allows the learning algorithms to focus on the instances that can be correctly classified. Removing the ISMs allows a more appropriate decision surface to be discovered since the ISMs do not arbitrarily pull the decision surface from its more optimal position. This leads to higher classification accuracy.

As discussed by Zamalloayz et al [10], B. Liu et al. [11] and Michael L. Raymer et al. [12] Genetic Algorithm (GA) is quite a popular method under research and is found to be quite effective for feature selection and classification accuracy improvement. All these researches related to GA are data specific or algorithm specific. In [10] the performance of GA is compared with other feature reduction and extraction techniques like Liner Discriminant Analysis (LDA), Principle Component Analysis (PCA) for one dataset GA was found to perform better while for the other dataset LDA and PCA showed promising results.



K	Clean speech			Telephone speech		
	GA	PCA	LDA	GA	PCA	LDA
6	**5.71±0.09**	14.37±0.15	8.11±0.14	34.23±0.16	**33.23±0.12**	35.52±0.14
8	**1.81±0.09**	5.86±0.12	2.64±0.09	**23.90±0.14**	24.19±0.13	25.06±0.13
10	**0.94±0.04**	2.73±0.12	1.21±0.06	19.70±0.12	20.67±0.12	**19.43±0.12**
11	**0.35±0.04**	1.61±0.07	1.12±0.06	19.32±0.14	20.27±0.13	**18.10±0.13**
12	**0.30±0.04**	0.94±0.06	0.79±0.06	19.27±0.14	19.75±0.16	**18.18±0.12**
13	**0.33±0.05**	0.56±0.05	0.88±0.04	19.12±0.11	19.63±0.10	**17.66±0.10**
20	**0.16±0.02**	0.19±0.02	0.39±0.04	19.99±0.11	17.61±0.13	**17.24±0.11**
30	**0.13±0.02**	0.15±0.03	0.33±0.04	19.10±0.14	**15.97±0.15**	18.17±0.12

Figure 7: Mean error rates and 95% confidence intervals in speaker recognition experiments on test data for clean and telephone speech, using the optimal K-feature sets provided by GA, PCA, and LDA, for k=6, 8, 10, 11, 12, 13, 20 and 30 [10]

GA outperformed PCA and LDA only when dealing with clean speech, whereas PCA and LDA outperformed GA in most cases when dealing with telephone speech, probably due to some kind of noise compensation implicit in linear transforms, which cannot be accomplished just by selecting a subset of features.

In [11] the Genetic Algorithm is combined with the boosting technique in order to improve accuracy of classification. The improved version assigns higher weight to the misclassified instances in order to shift the focus on them in the next iteration. This process tends to achieve higher accuracy with less number of evaluations than the original GA.

Table 5: Mean Prediction Accuracy of GA and BoostGA [11]

	Accuracy		Evaluations	
Data set	GA	Boost GA	GA	BoostGA
Breast Cancer	90.6 1.9%	93.1+2.2%	20,000	7,500
Tic-tac-toe	70.0+1.4%	84.7+1.8%	20,000	7,500

In [12] Genetic Algorithm is implemented in combination with K-nearest neighbor classifier and feature extraction, reduction and classifier training are all done simultaneously and results are compared with other industry standard feature extraction and reduction technique like Liner Discriminant Analysis and Sequential Floating Forward Feature Selection.

Despite all this extensive work on ensemble methods and feature reduction problem and various classification algorithms for improving the accuracy rate in classification. There still not any research that focus on the improvement of an algorithm in more than one aspect or we can say that mostly the improvement are data and problem specific. There is no global optimization ensemble model suggested so far that can improve the accuracy of classification methods with any dataset. Therefore in this thesis we design and implement such a global optimization model and detail design for this model is given in the section 3.

The data source used in most of the previous research work are public dataset (e.g Cleveland heart disease dataset, breast cancer dataset, wine dataset, sonar data etc available free of cost online for use in research work. Therefore in this project we will also be using some of these datasets for testing.

CHAPTER 3: METHODOLOGY

CHAPTER 3: METHODOLOGY

The aim of this chapter is to provide the design of Global Optimization Model. The idea was to implement the concept of ensemble model in order to create global model for optimization.

3.1 Flow Chart of Global Optimization Ensemble Model for Classification Methods (GMC):

The flow chart of GMC is given below. It consists of four layers.

Layer 1: Providing Antidote for Dimensionality Curse

Layer 2: Cross Validation

Layer 3: Bagging (Bias-variance Trade off)

Layer 4: Classification

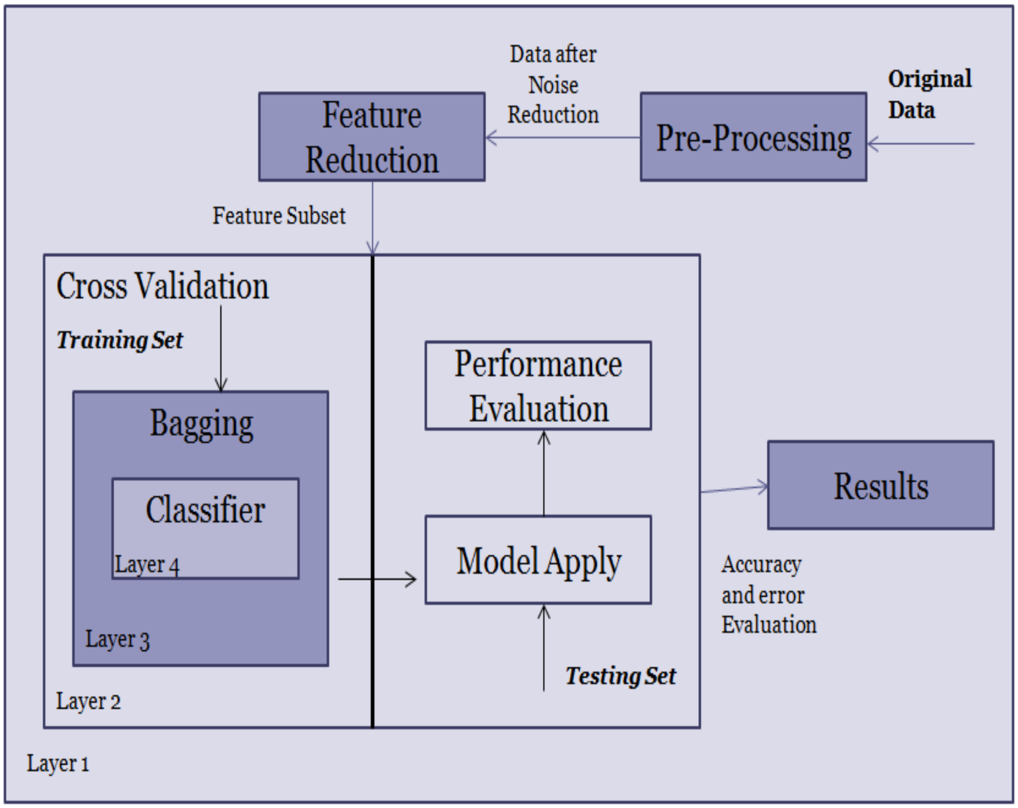


Figure 8: Design of Global Optimization Ensemble Model for Classification Methods (GMC)

3.2 Layer 1: Providing antidote for Dimensionality curse:

As discussed in the literature review the dimensionality reduction or feature reduction is necessary in order to improve the classification accuracy. Therefore in our model the first layer contains the data set, pre-processing operator and a feature reduction operator. According to [12] Genetic Algorithm (GA) is better than other feature reduction techniques so therefore we have implemented GA in our proposed model.

3.2.1 The Use of GA in GMC model:

The process that the GA is following is as under:

1. The initial population of individuals is selected randomly.
2. Than the fitness of each individual in this initial population is evaluated.
3. In this design the Maximal fitness is set to infinity which implies that there is no absolute maxima for this function and the algorithm will keep on checking the populations for best of best until the maximum number of generations are executed.
4. Repeat until termination (i.e. maximum number of generations):
    - The best-fit individuals are selected for reproduction
    - New individuals are bred, first through mutation and then a crossover operation is performed to give birth to offspring.
    - The fitness of individual is evaluated and new individuals are selected through roulette wheel selection scheme (as maximal fitness is infinity so the process will repeat again)
    - Least-fit population is replaced with new individuals

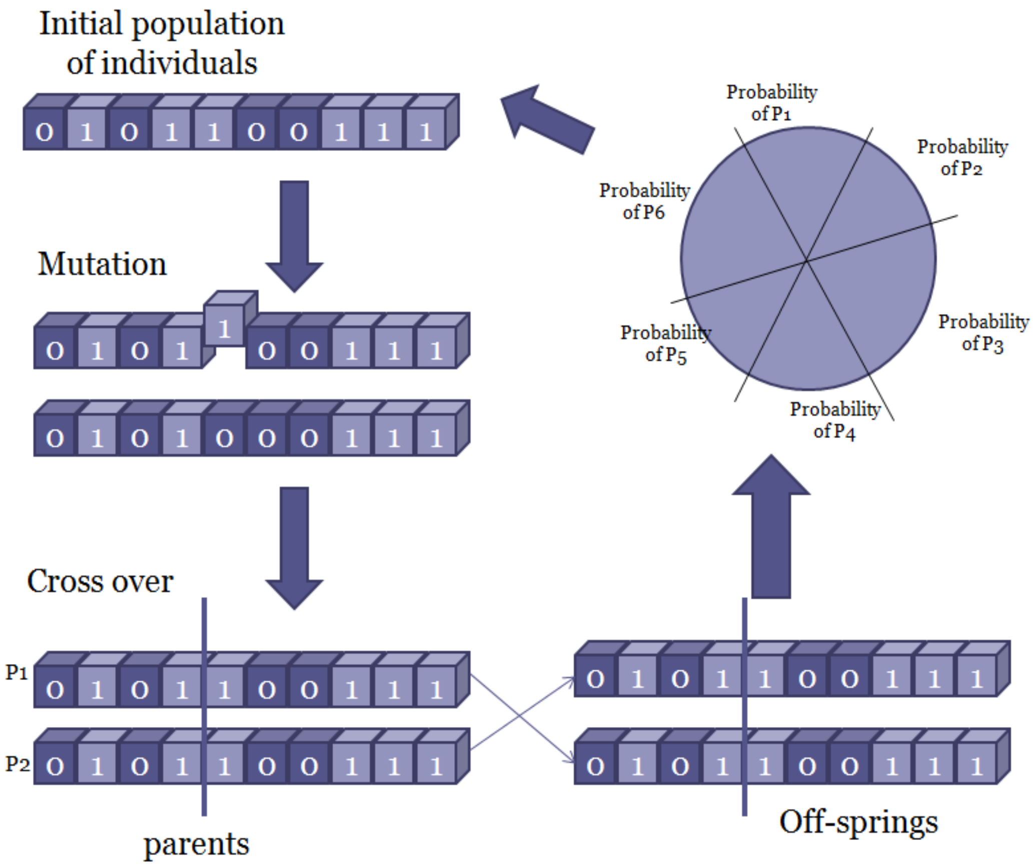


Figure 9: Use of Genetic Algorithm (GA) in GMC model

Some parameters of the algorithm were optimized using grid search method as shown in Table 7, while others were set as follows:

Table 6: Parameter Values in GA

Parameter	Value
Selection Scheme	Roulette Wheel
Cross over type	Shuffle
Probability of Cross over	0.5
Probability for initial population	0.5
Probability of mutation	1/number of attribute
Maximal Fitness	Infinity

Maximal fitness is set to infinity as there is no absolute maxima for the fitness function which means the GA will keep on selecting the best of best until the stop criteria is met which in this case is the maximum number of generation. Roulette wheel selection scheme was used for selecting individuals because it has the obvious advantage that it does not ignore or discard any individuals and each individual is given a chance of being chosen as even the weakest of individuals might be hiding valuable information. And as we are striving for a global solution therefore a selection method that preserves diversity and is fast to converge sounds good. Our experimental results (presented in chapter 5) show that this selection produces good results.

Cross over type was set to shuffle because shuffle crossover is related to uniform crossover. A single crossover position (as in single-point crossover) is selected. But before the variables are exchanged, they are randomly shuffled in both parents. After recombination, the variables in the offspring are un shuffled. This removes positional bias as the variables are randomly reassigned each time crossover is performed.

3.3 Parameter Optimization:

Parameter optimization for the operators in each layer was done by implementing Global optimization operator using Grid search. “This methodology involves setting up of grids in the decision space and evaluating the values of the objective function at each grid point. The point which corresponds to the best value of the objective function is considered to be the optimum solution.”[13] In all the layers total 5 parameters were optimized using Grid search optimizations. From each attribute’s grid 11 combinations were proposed; this means for optimizing these 5 parameters total 161051 combinations were tested. Table 7 shows all the parameter and there optimized values.

Table 7: Parameter optimization using Grid Search

Parameter	Operator	Grid Range	Combination	Optimal Value
Population size	GA-Layer 1	2-100	2,3,6,11,18, 27, 37, 50, 65, 81, 100	6
Maximum no. of generation	GA-Layer 1	1-50	1, 6, 11, 16, 21, 26, 30, 35, 40, 45, 50	16
Number of iterations	CV-Layer 2	2-50	2, 4, 6, 10, 14, 19, 26, 33, 41, 50	10
Sampling Size	Bagging- Layer 3	0-1.0	0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1	0.6
Number of iterations	Bagging- Layer 3	1-100	1, 2, 5, 10, 17, 26, 37, 50, 64, 81, 100	10

3.4 Layer 2: X-Fold Cross Validation:

In Layer 2 Partition of Training and testing Data set was done using X-Fold cross validation. **"**The data set is divided into *n* subsets, and the holdout method is repeated *k* times. Each time, one of the *n* subsets is used as the test set and the other *n-1* subsets are put together to form a training set. Then the average error across all *n* trials is computed. The advantage of this method was that it matters less how the data gets divided. Every data point gets to be in a test set exactly once, and gets to be in a training set *n-1* times. Besides, the variance of the resulting estimate is also reduced as *n* is increased." [14] Stratified sampling scheme was used in CV with number of iteration set to 10 as shown in Table 7. In stratified sampling the random subsets are created but the distribution of class in those subsets is same as the whole dataset. Thus this type of sampling reduces variance. For example we have a data set of 180 employees and we want a sample set of 40 employees. The first step is to calculate the percentage of male female in each group. i.e

- Percentage of male members in full-time category= 90 / 180 = 50%
- Percentage of male members in part-time category = 18 / 180 = 10%
- Percentage of female members in full-time category = 9 / 180 = 5%
- Percentage of female members in part-time category= 63 / 180 = 35%

This calculation tell us that of our desired sample of 40 employees, 50 percent should be male (full time), 10 percent should be male (part-time), 5 percent should be female (full-time), 35 percent should be female (part-time). This means that we need to calculate the 50% of 40 which is 20. Similarly 10% of 40 is 4, 5% of 40 is 2, 35% of 40 is 14. This is the final ratio of records in each category in our sample of 40 employees.

3.5 Layer 3: Bias-Variance Tradeoff:

Layer 3 did an optimal Bias-Variance Trade-off. Accuracy improvement is done by implementing bootstrap aggregation (bagging). Bagging is a machine learning ensemble meta-algorithm which reduces both bias and variance in order to help avoid over fitting. “Although it is usually applied to decision tree models, it can be used with any type of model. Bagging is a special case of the model averaging approach”.[15] Parameter setting for bagging is shown in Table 7. We are using bagging instead of boosting because **Error = Noise error + Bias + Variance** Bagging can reduce both bias and variance but mostly it reduces just variance and it hardly ever increase error. For high-bias classifiers, it can reduce bias and for high-variance classifiers, it can reduce variance. While boosting in the early iterations; is primary a bias-reducing method. In later iterations, it appears to be primarily a variance-reducing method. It may increase error and margins and is not good with data with noise. That is the reason that we chose bagging instead of boosting for bias and variance tradeoff.

3.6 Layer 4: Classification

Classifiers were placed in layer 4 with parameters configuration done according to the dataset. All classifier parameters were set to obtain the optimal model in order to reduce the bias. The setting used for each classifier is shown in the following table.

Table 8: Parameter configuration for Classifiers

Operator Name	Parameter Configuration
ID 3	Criterion: Information_gain
	Minimal size of split: 4
	Minimal leaf size: 2
	Minimal gain: 0.1
Decision Tree	Criterion: Information_gain
	Minimal size for split: 4
	Minimal leaf size: 2
	Minimal gain: 0.1
	Maximal depth: 20
	Confidence: 0.5
Random forest	Number of trees: 10
	Criterion: Information_gain
	Minimal leaf size: 2
	Minimal gain: 0.1
	Maximum depth: 20
	Confidence:0.5
Rule Induction	Criterion: Information_gain
	Sample ratio: 0.7
	Pureness:0.6
	Minimal prune benefit: 0.6
K-NN	K nearest neighbors : 11
	Weighted Vote : True
	Measure Type: NominalMeasures
	Nominal Measure: DiceSimilarity
Naïve Bayes	Laplace Correction : True
W-AODE	Frequency for super parents: 1.0
W-PART	Confidence Threshold: 0.5
	Minimum Objects Per Leaf: 2.0
W-J48	Confidence Threshold: 0.5
	Minimum Objects Per Leaf: 2.0

CHAPTER 4: IMPLEMENTATION

CHAPTER 4: IMPLEMENTATION

The aim of this chapter is to provide a detailed view of how GMC was implemented. Implementation and testing is done using core i3 processor with 4GB RAM, while coding is done using XML. Pre-processing is performed on every dataset according to requirements of the classifier used in order to remove noise from data and do type conversations. The model is implemented and tested in RapidMiner5.

“RapidMiner, formerly YALE (Yet Another Learning Environment), is an environment for machine learning, data mining, text mining, predictive analytics, and business analytics. It is used for research, education, training, rapid prototyping, application development, and industrial applications. In a poll by K Dnuggets, a data-mining newspaper, RapidMiner ranked second in data mining/analytic tools used for real projects in 2009 and was first in 2010.It is distributed under the AGPL open source license and has been hosted by Source Forge since 2004.” [16]

Step 1: Algorithm Selection

As we are optimizing the model for supervised learning problems therefore following liner and non-liner classifiers were selected and implemented and tested.

- **KNN** “is a method for classifying objects based on closest training examples in the feature space. An object is classified by a majority vote of its neighbors, with the object being assigned to the class most common amongst its k nearest neighbors (k is a positive integer, typically small). If k = 1, then the object is simply assigned to the class of its nearest neighbor.” [17, 18]
- **Decision Tree** “works similar to Quinlan's C4.5 or CART. Roughly speaking, the tree induction algorithm works as follows. Whenever a new node is created at a certain stage, an attribute is picked to maximize the discriminative power of that node with respect to the examples assigned to the particular sub tree. This discriminative power is measured by a criterion which can be selected by the user (information gain, gain ratio, gini index, etc.). The algorithm stops in various cases:

    - No attribute reaches a certain threshold (minimum gain).
    - The maximal depth is reached.
    - There are less than a certain number of examples (minimal_size_for_split) in the current sub tree.

Finally, the tree is pruned, i.e. leaves that do not add to the discriminative power of the whole tree are removed”. [19, 20]

- **ID3** “It’s a decision tree learner which learns without pruning and works only for nominal attributes. Its an implementation of Quinlan’s ID 3. The ID3 algorithm can be summarized as follows:
    - Take all unused attributes and count their entropy concerning test samples
    - Choose attribute for which entropy is minimum (or, equivalently, information gain is maximum)

Make node containing that attribute.” [21, 22]

- **Random Forest** “Learns a set of random trees, i.e. for each split only a random subset of attributes is available. The resulting model is a voting model of all trees.” [23, 24]
- **Logistic Regression** “Logistic regression allows one to predict a discrete outcome, such as group membership, from a set of variables that may be continuous, discrete, dichotomous, or a mix of any of these. Generally, the dependent or response variable is dichotomous, such as presence/absence or success/failure”. [25, 26]
- **Rule induction** “It’s an implementation of algorithm RIPPER which is a rule based learner. It grows iteratively and prunes the nodes until there are no positive examples left.” [27]
- **W-AODE:** “It’s a WEKA implementation of Naïve Bayes learner which is a simple probabilistic classifier based on applying Bayes' theorem with strong (naive) independence assumptions. Parameter estimation for naive Bayes models uses the method of maximum likelihood. It cannot handle numerical attributes and missing values and numeric label.
- **W-PART:** It classifies using separate and conquer rule and builds a partial C 4.5

- **W-Prism:** It's an implementation of PRISM algorithm and can only deal with nominal dataset. Prim's algorithm is a greedy algorithm that finds a minimum spanning tree for a connected weighted undirected graph. This means it finds a subset of the edges that forms a tree that includes every vertex, where the total weight of all the edges in the tree is minimized.
- **W-J48** It's a WEKA implementation of C 4.5 decision tree algorithm. At each node of the tree, C4.5 chooses one attribute of the data that most effectively splits its set of samples into subsets enriched in one class or the other. Its criterion is the normalized information gain (difference in entropy) that results from choosing an attribute for splitting the data. The attribute with the highest normalized information gain is chosen to make the decision. The C4.5 algorithm then recourses on the smaller sub lists." [28].

Step 2: Data Set Selection

For this research public datasets were selected that are usually used in other researches as well for testing supervised learning methods (see chapter 2). Details related to each dataset along with meta-data view and description of attributes is given in appendices C. A brief summary is given here in Table 8.

Table 9: Data set Details

Data Set	# of cases	# of attribute	# of classes	Attribute Characteristics	Missing Values
Cancer Dataset:	699	9	2	numeric	yes
Diabetes Dataset	768	9	2	Integer, real	No
Heart Disease Dataset	303	14	2	Categorical, integer, real	yes
Adult Income dataset	1000	15	2	Integer, Nominal	No
Wine Dataset	178	13	3	Real , Integer	No
Sonar Dataset	208	61	2	Real, nominal	yes
Educational progress Dataset	50	9	3	Nominal	No

Suitable classifier for each dataset is as under:

Table 10: Data set & Suitable Classifiers

Dataset	Classifier	Capabilities
All Datasets	K-NN	Polynomial, numerical ,binomial attributes & labels, Can handle missing values
All Datasets	Decision Tree	Polynomial, numerical, binomial attributes. Cannot handle numeric labels, Can handle missing values
Heart, wine, Educational and Sonar Dataset	Rule induction	
Cancer, Heart, Adult Income Dataset	ID3	Can only handle binomial and polynomial labels and attributes, and cannot handle missing values
All Datasets	W-AODE	
All Datasets	W-Prism	
Educational Progress, Sonar and Adult Income Dataset	Random Forest,	Polynomial, numerical, binomial attributes. Cannot handle numeric labels, Cannot handle missing values
All Datasets	W-PART	
All Datasets	W-J48	
Sonar, Diabetes, Cancer, Adult Income Dataset	Logistic Regression	Numerical attributes and binomial labels, cannot handle missing values

Step 3: Simple Classification using Validation technique

First each dataset is classified using the classifier mentioned for each dataset and the results are validated using the X-fold cross-validation technique. Where x=10 for all classifiers and sampling technique used for validation is “Shuffled sampling”. Results consisting of classification accuracy, classification error and execution time are recorded for each classifier (as shown in chapter 5).

Step 4: Classification using global optimization ensemble model for classification methods (GMC)

All the classifiers are now encapsulated in the proposed generic optimization ensemble model and executed for results. Parameters of all the classifiers are same as in step 3 and as specified in chapter 3. Now the improved results consisting of optimized classification accuracy, classification results and execution time is recorded for every classifier and compared with the previous result in order to calculate the improvement percentage. (as shown in chapter 5).

CHAPTER 5: RESULTS AND DISCUSSION

CHAPTER 5: RESULTS AND DISCUSSION

This chapter provides the results for each data set and the corresponding accuracy comparison between simple classification and GMC model are given in this section.

5.1 Cancer Dataset:

Details related to this dataset can be seen in Appendices C. For this particular dataset a preprocessing operator “Numerical to binomial” was used. This operator converts the specified columns of the numerical data to binomial data (y/n). We converted the numerical label of this dataset for further processing because CV operator in RapidMiner cannot handle numerical labels. As the label only contains two types of values that is “2” or “4” so they were easily converted to binomial label with “2” set to “N” and “4” set to “Y”. Noise was present in form of missing values which was removed using the “Replace missing values” operator. It replaced the missing values in each column with the average value for the column. First the classification results were recorded using simply each classifier and cross validating there results. Than these classifiers were place inside the GMC model and results were recorded as shown in table below.

Table 11: Results for Cancer Dataset: Comparison of Optimized Classification Accuracy using GMC Model with Simple Classification using different classifiers.

Algorithm	Classification Accuracy	Classification Error	Execution Time	Optimized Classification Accuracy	Classification Error	Execution Time	Improvement %
K-NN	66.81%	33.19%	0s	96.71%	3.43%	53s	**29.9%**
Decision Tree	94.42%	5.58%	0s	96.71%	3.29%	6:o8s	**2.29%**
ID3	66.52%	33.48%	0s	85.27%	14.73%	15:08s	**18.52%**
W-PART	94.71%	5.29%	0s	97.28%	2.72%	1:00s	**2.57%**
W-Prism	90.13%	9.87%	0s	96.28%	3.72%	2:36s	**6.15%**
W-J48	94.71%	5.29%	0s	96.71%	3.29%	1:47s	**2%**
W-AODE	97.00%	3.00%	0s	100%	0%	11s	**3%**
Logistic Regression	95.01%	4.01%	0s	96.14%	3.86%	3:59s	**1.13%**

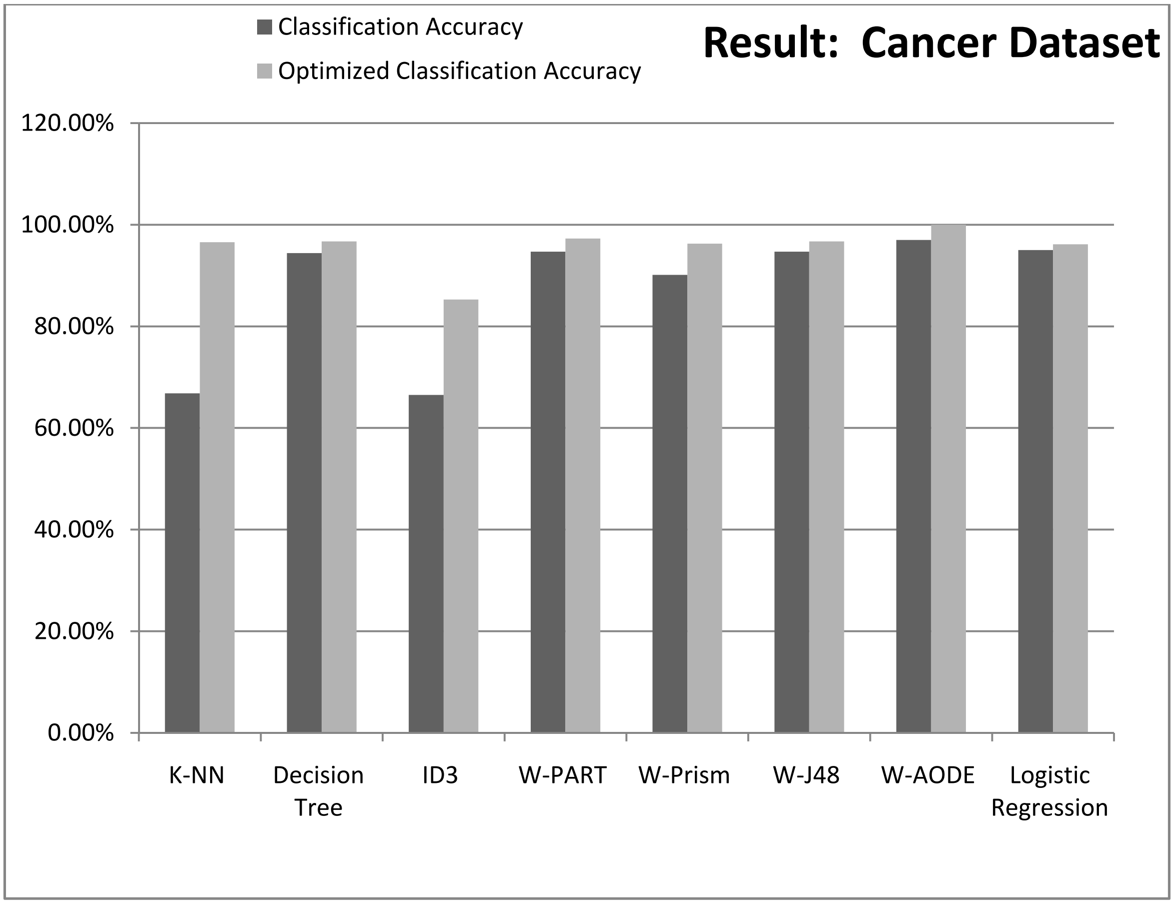


Figure 10: Cancer Dataset: Optimized Classification Accuracy

Table 11 shows that using the GMC model for optimization the classification accuracy for cancer dataset has improved from 1.13% to 29.76% depending on the classifier and the bias-variance trade off each model makes.

5.2 Heart Disease Dataset:

Details related to this dataset can be seen in Appendices C. For this particular dataset a preprocessing operator “Numerical to polynomial” was used. This operator converts the specified columns of the numerical data to polynomial data. We converted the numerical label of this dataset for further processing because CV operator in RapidMiner cannot handle numerical labels. First the classification results were recorded using simply each classifier and cross validating there results. Than these classifiers were place inside the GMC model and results were recorded as shown in table below.

Table 12: Results for Heart Disease Dataset: Comparison of Optimized Classification Accuracy using GMC Model with Simple Classification using different classifiers.

Algorithm	Classification Accuracy	Classification Error	Execution Time	Optimized Classification Accuracy	Classification Error	Execution Time	Improvement %
K-NN	50.82%	49.18%	0s	59.75%	40.25%	19s	**8.93%**
Decision Tree	44.89%	55.11%	0s	59.43%	40.57%	3:27	**14.54%**
ID3	47.52%	52.48%	0s	55.48%	44.24%	5:11	**8.24%**
W-PART	50.52%	49.48%	0s	60.08%	39.92%	2:10	**9.56%**
W-Prism	47.51%	52.49%	0s	56.09%	43.91%	1:12	**8.58%**
W-AODE	55.47%	44.53%	0s	61.13%	38.87%	36s	**5.66%**
W-J48	49.87%	50.13%	0s	61.05%	38.95%	2:12	**11.18%**
Rule Induction	57.72%	42.28%	0s	59.76%	40.24%	20:48	**2.4%**

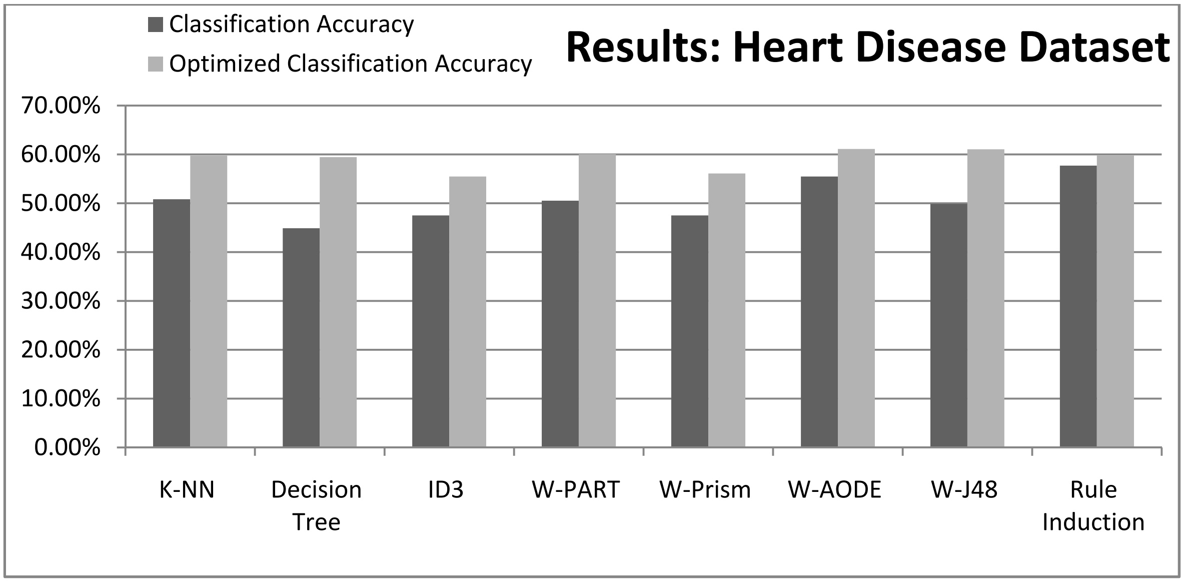


Figure 11: Heart Disease Dataset: Optimized Classification Accuracy

Table 12 shows that using the GMC model for optimization the classification accuracy for Heart Disease dataset has improved from 2.4% to 14.54% depending on the classifier and the bias-variance trade off made by model.

5.3 Wine Dataset:

Details related to this dataset can be seen in Appendices C. For this particular dataset a preprocessing operator “Numerical to polynomial” was used. This operator converts the specified columns of the numerical data to polynomial data. We converted the numerical label of this dataset for further processing because CV operator in RapidMiner cannot handle numerical labels. First the classification results were recorded using simply each classifier and cross validating there results. Than these classifiers were place inside the GMC model and results were recorded as shown in table below.

Table 13: Results of Wine Dataset: Comparison of Optimized Classification Accuracy using GMC Model with Simple Classification using different classifiers.

Algorithm	Classification Accuracy	Classification Error	Execution Time	Optimized Classification Accuracy	Classification Error	Execution Time	Improvement %
K-NN	70.75%	29.25%	0s	90.42%	9.58%	11s	**19.67%**
Decision Tree	91.57%	8.43%	0s	95.49%	4.51%	1:40	**3.92%**
W-PART	90.42%	9.58%	0s	96.67%	3.33%	28s	**6.25%**
W-Prism	52.32%	47.68%	0s	61.27%	38.73%	39s	**8.95%**
W-AODE	71.34%	28.66%	0s	75.26%	24.74%	2:46s	**3.92%**
W-J48	90.46%	9.54%	0s	96.63%	3.37%	4:46	**6.17%**
Rule induction	86.37%	13.63%	0s	93.27%	6.73%	4:01s	**6.9%**

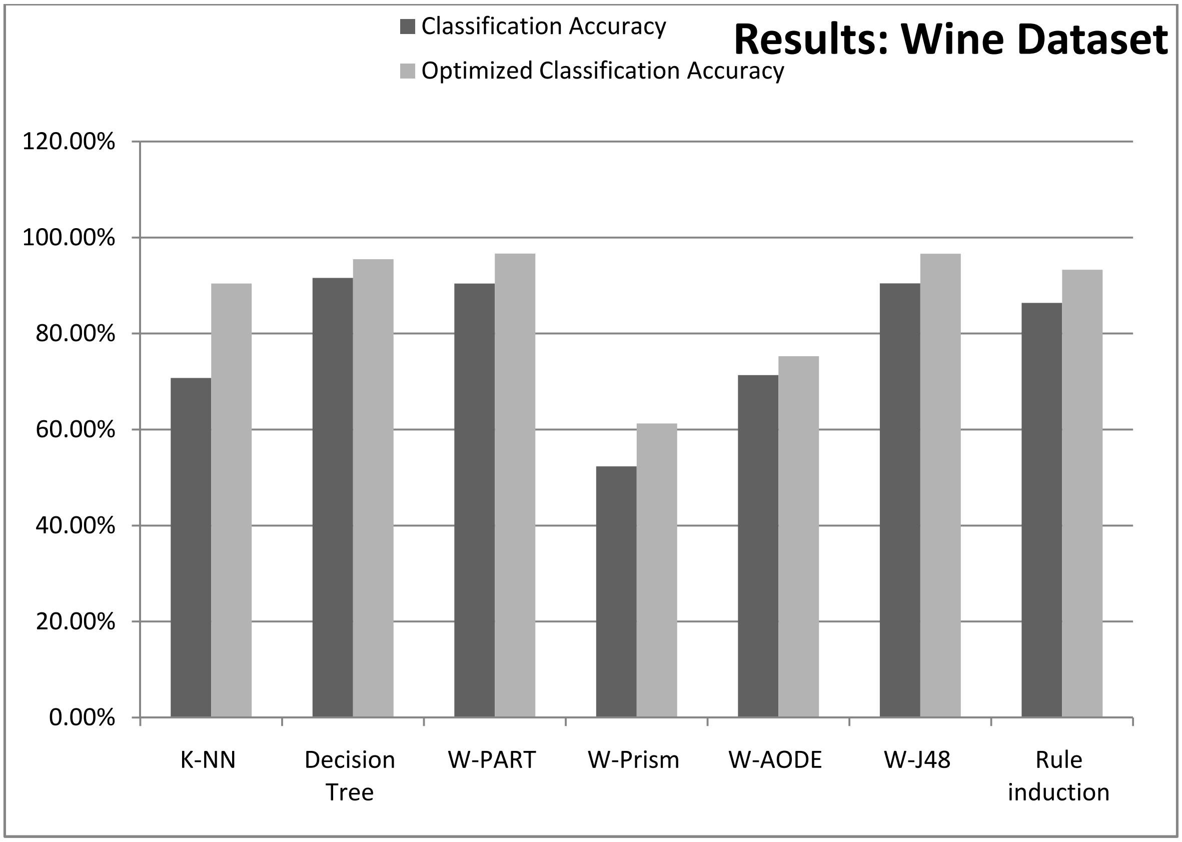


Figure 12: Wine Dataset : Optimized Classification Accuracy

Table 13 shows that using the GMC model for optimization the classification accuracy for Wine dataset has improved from 3.92% to 19.67% depending on the classifier and the bias-variance trade off made by the model.

5.4 Adult Income Dataset:

Details related to this dataset can be seen in Appendices C. For this particular dataset no pre-processing operator was used. The class label is already a binomial column and there was no noise in the data. Therefore first the classification results were recorded using simply each classifier and cross validating there results. Than these classifiers were place inside the GMC model and results were recorded as shown in table below.

Table 14: Results of Adult Income Dataset: Comparison of Optimized Classification Accuracy using GMC Model with Simple Classification using different classifiers.

Algorithm	Classification Accuracy	Classification Error	Execution Time	Optimized Classification Accuracy	Classification Error	Execution Time	Improvement %
K-NN	76.70%	23.30%	0s	83.20%	16.80%	2:02	**6.5%**
Decision Tree	80.00%	20.00%	1s	82.20%	17.80%	11:06	**2.20%**
ID3	75.60%	24.40%	1s	78.60%	21.40%	1:15:23	**3%**
W-PART	81.00%	19.00%	0s	83.50%	16.50%	5:02s	**2.4%**
W-Prism	81.10%	18.09%	0s	82.20%	17.80%	9:38s	**1.1%**
W-AODE	80.80%	19.20%	0s	82.60%	17.40%	9:42s	**1.8%**
W-J48	81.50%	18.50%	0s	83.00%	17.00%	2:46s	**1.5%**
Random Forest	76.10%	23.90%	0s	77.30%	22.70%	36:58s	**1.2%**
Logistic Regression	79.00%	20.40%	1s	80.00%	20.00%	8:56s	**1%**

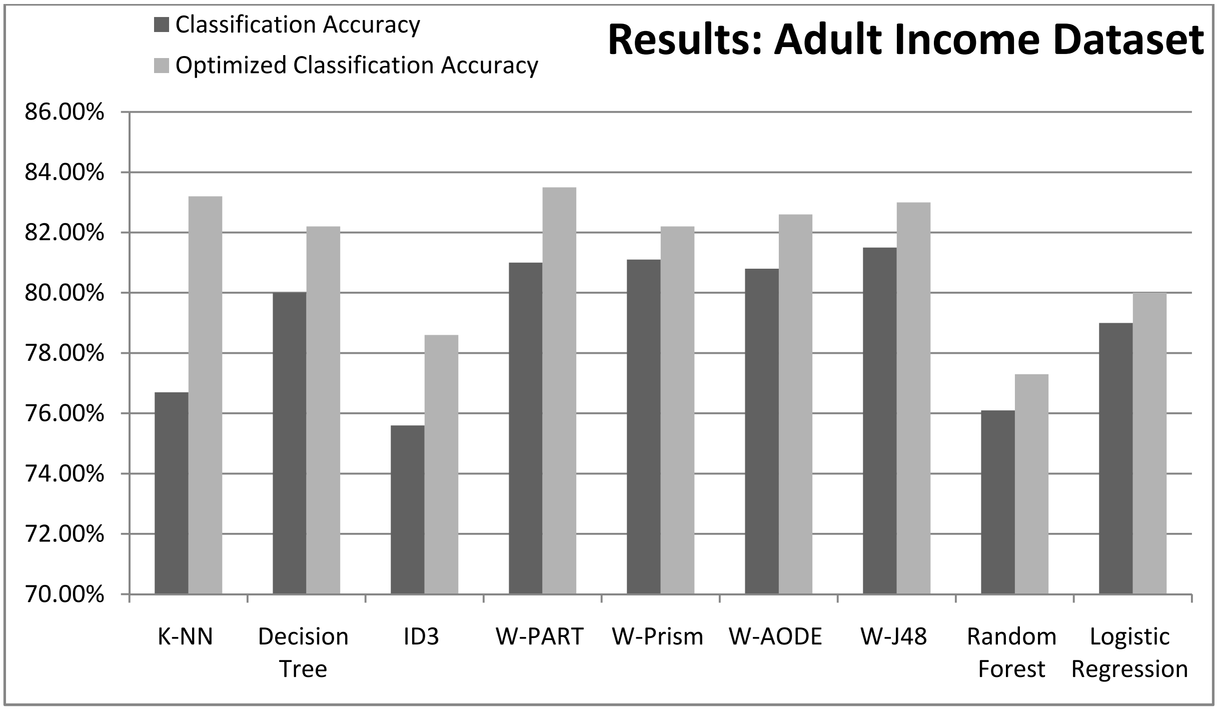


Figure 13: Adult Income Dataset: Optimized Classification Accuracy

As shown in Table 14, using the GMC model for optimization the classification accuracy for Adult Income dataset has improved from 1% to 6.5% depending on the classifier and the bias-variance trade off made by the model.

5.5 Sonar Dataset:

Details related to this dataset can be seen in Appendices C. For this particular dataset no pre-processing operator was used. The class label is already a binomial column and there was no noise in the data. Therefore first the classification results were recorded using simply each classifier and cross validating there results. Than these classifiers were place inside the GMC model and results were recorded as shown in table below.

Table 15: Results of Sonar Dataset: Comparison of Optimized Classification Accuracy using GMC Model with Simple Classification using different classifiers.

Algorithm	Classification Accuracy	Classification Error	Execution Time	Optimized Classification Accuracy	Classification Error	Execution Time	Improvement %
K-NN	69.71%	30.92%	0s	74.57%	25.43%	24s	**4.86%**
Decision Tree	73.57%	26.43%	1s	83.67%	16.33%	17:10s	**10.1%**
W-PART	75.48%	24.52%	0s	83.17%	16.83%	3:10s	**7.69%**
W-Prism	48.02%	51.98%	0s	63.38%	36.62%	2:53	**15.36%**
W-J48	70.24%	29.76%	0s	82.21%	17.79%	2:56	**11.97%**
Rule induction	71.66%	28.40%	0s	76.48%	23.525	2:44	**4.82%**
Random Forest	68.26%	31.74%	0s	75.36%	21.64%	20:41s	**7.1%**
Logistic Regression	74.55%	25.45%	0s	80.29%	19.71%	1:45s	**5.74%**

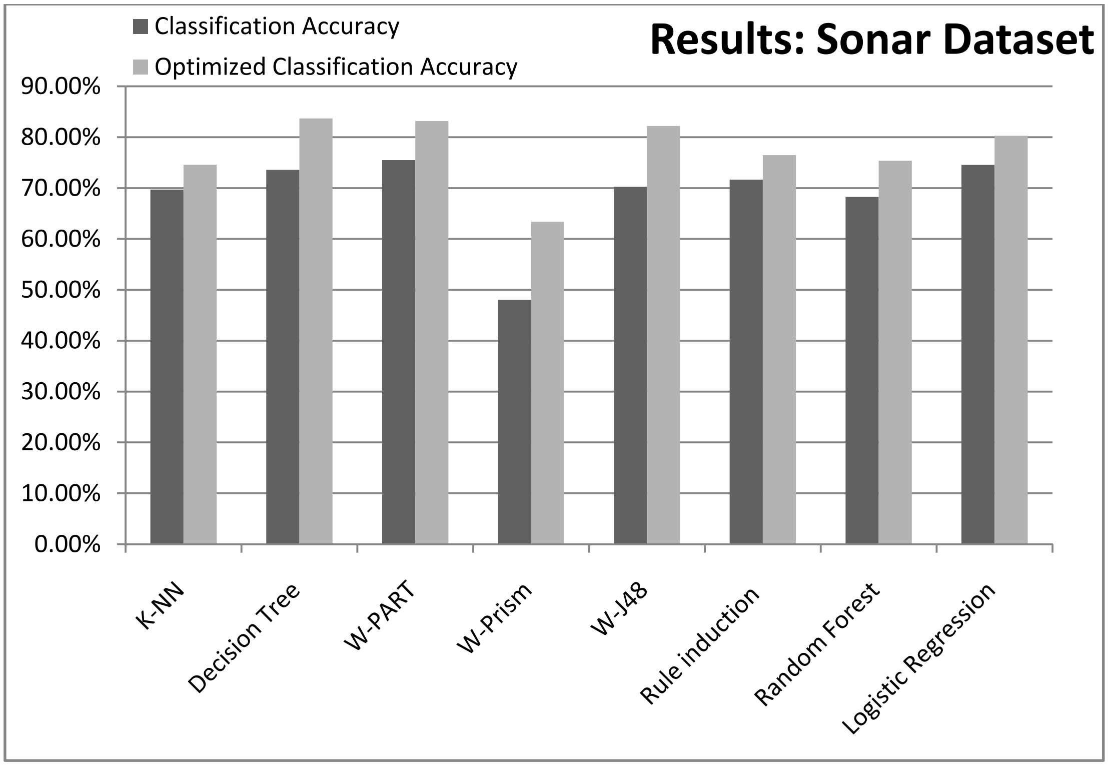


Figure 14: Sonar Dataset: Optimized Classification Accuracy

As shown in Table 15, using the GMC model for optimization the classification accuracy for Sonar dataset has improved from 4.82% to 15.36% depending on the classifier and the bias-variance trade off made by the model.

5.6: Educational Dataset:

Details related to this dataset can be seen in Appendices C. For this particular dataset no pre-processing operator was used. The class label is already a polynomial column and there was no noise in the data. Therefore first the classification results were recorded using simply each classifier and cross validating there results. Than these classifiers were place inside the GMC model and results were recorded as shown in table below.

Table 16: Results of Educational Dataset: Comparison of Optimized Classification Accuracy using GMC Model with Simple Classification using different classifiers.

Algorithm	Classification Accuracy	Classification Error	Execution Time	Optimized Classification Accuracy	Classification Error	Execution Time	Improvement %
K-NN	46%	54%	0s	54%	46%	4s	**8%**
Decision Tree	42%	58%	0s	56%	44%	13s	**14%**
ID3	20%	80%	0s	44%	56%	21s	**24%**
W-PART	32%	68%	0s	54%	46%	8s	**22%**
W-Prism	24%	76	1s	50%	50%	5s	**26%**
W-J48	44%	56%	0s	58%	42%	8s	**14%**
W-AODE	46%	54%	0s	56%	44%	6s	**10%**
SVM	60%	40	0s	76%	24%	25s	**16%**
Random Forest	48%	52%	1s	58%	42%	1:02	**12%**
Rule Induction	44%	56%	17s	54%	46%	2:07s	**10%**

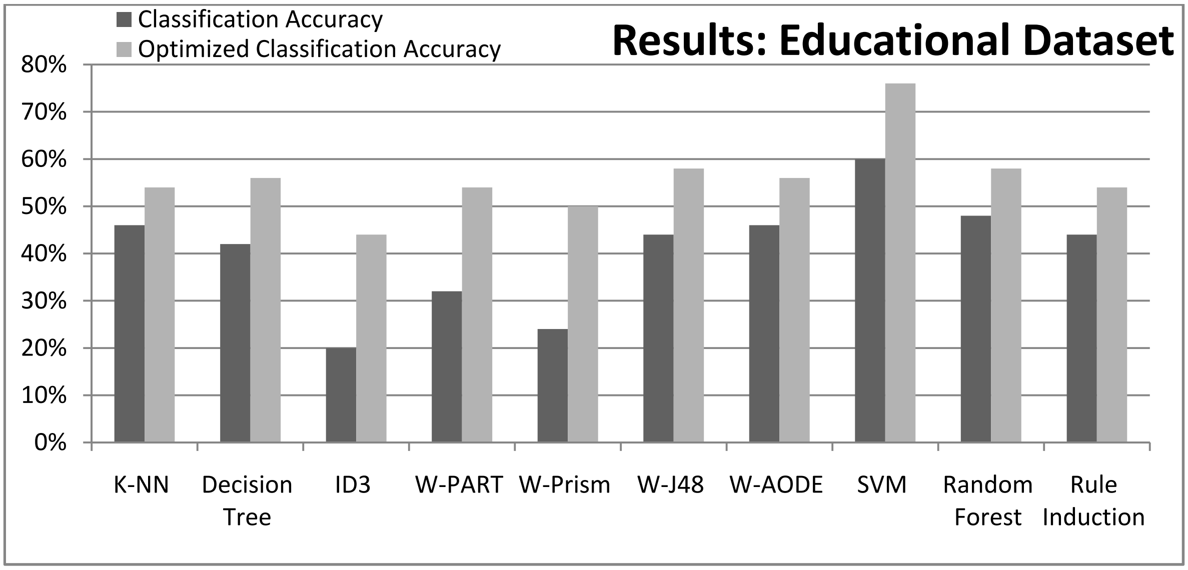


Figure 15: Educational Dataset: Optimized Classification Accuracy

As shown in Table 16, using the GMC model for optimization the classification accuracy for Educational dataset has improved from 8% to 26% depending on the classifier and the bias-variance trade off made by the model.

5.7 Diabetes Dataset:

Details related to this dataset can be seen in Appendices C. For this particular dataset a preprocessing operator "Numerical to binomial" was used. This operator converts the specified columns of the numerical data to binomial data (y/n). We converted the numerical label of this dataset for further processing because CV operator in RapidMiner cannot handle numerical labels. As the label only contains two types of values i.e. "0" or "1"; so they were easily converted to binomial label with "0" set to "N" and "1" set to "Y". Noise was not present in this dataset. First the classification results were recorded using simply each classifier and cross validating there results. Than these classifiers were place inside the GMC model and results were recorded as shown in table below.

Table 17: Results of Diabetes Dataset: Comparison of Optimized Classification Accuracy using GMC Model with Simple Classification using different classifiers.

Algorithm	Classification Accuracy	Classification Error	Execution Time	Optimized Classification Accuracy	Classification Error	Execution Time	Improvement %
K-NN	73.70%	26.30%	0s	77.48%	22.52%	37s	**4%**
Decision tree	74.0%	265	0s	75.39%	24.61	2:41	**1.39%**
W-PART	73.83%	26.17%	0s	77.34%	22.66%	1:13s	**3.51%**
W-Prism	57.42%	42.58%	0s	67.97%	32.03%	4:21	**10.55%**
W-J48	74.08%	25.92%	0s	77.22%	22.78%	1:53	**3.14%**
W-AODE	66.54%	33.46%	0s	69.14%	30.86%	3:13s	**2.6%**
Logistic Regression	76.00%	23.05%	0s	77.95%	22.65%	3:03s	**1.95%**

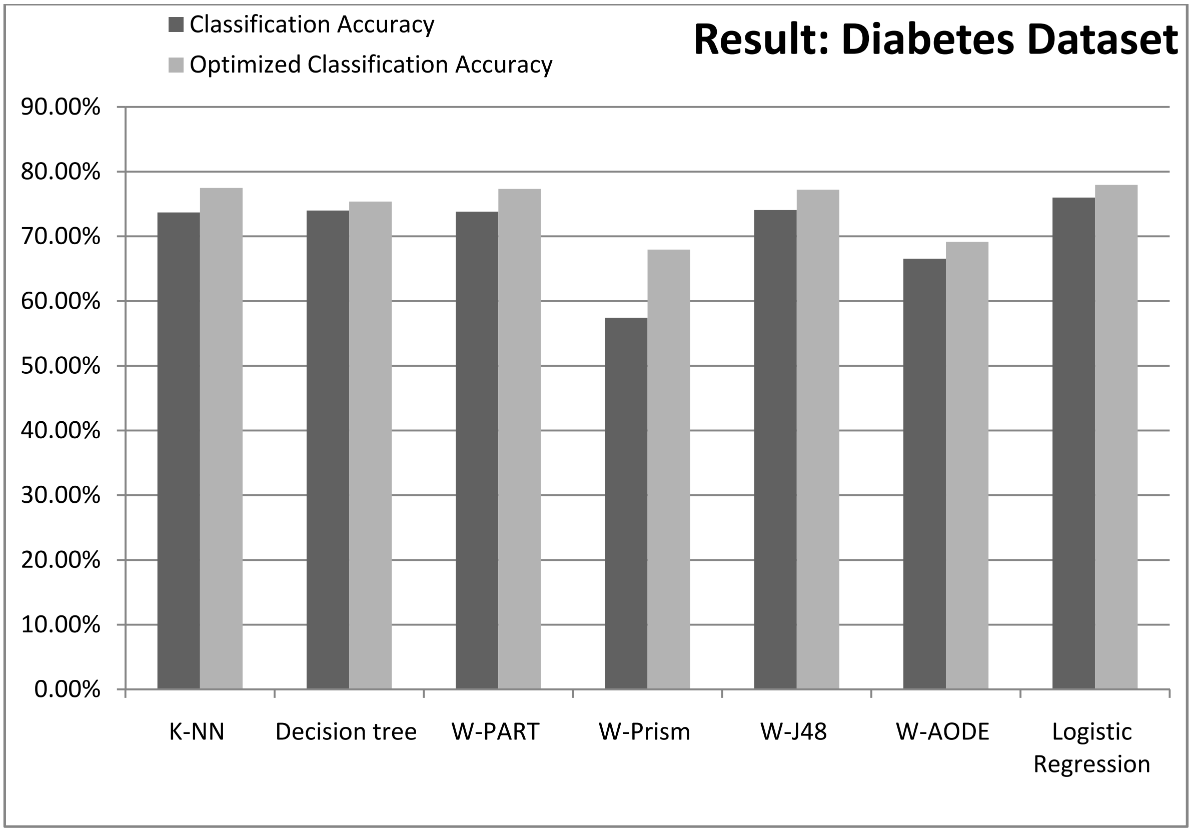


Figure 16: Diabetes Dataset: Optimized Classification Accuracy

As shown in Table 17, using the GMC model for optimization the classification accuracy for Diabetes dataset has improved from 1.39% to 10.55% depending on the classifier and the bias-variance trade off made by the model.

CHAPTER 6: CONCLUSION AND FUTURE WORK

CHAPTER 6: CONCLUSION AND FUTURE WORK

6.1 Conclusion

Data mining is the process of extracting useful information and pattern form raw data gathered from various resources. Supervised learning is a process in data mining which helps in extracting patterns and information form labeled data. A lot of techniques, methods and algorithms are available each with its own pros and cos. Each method tries to solve a different supervised learning problem. In order to solve the basic issues of supervised learning problems like dimensionality reduction, bias-variance tradeoff and noise; many researchers have tackled these problems from various aspects as discussed in chapter 2. But not a single research tackles all these problems at once. Or no single optimization technique or model is proposed for this purpose. Therefore we used the concept of ensemble models to design an optimized global ensemble model for classification methods (GMC). The model was designed in layers with each layer is solving one of the basic issues of supervised learning. Layer 1 solved the issue of dimensionality curse. We used the genetic algorithm for this purpose as it was stated to be the best techniques for feature reduction for static databases in [10, 11, and 12] so therefore in layer one we implemented genetic algorithm; like all other layers the parameter optimization was done for genetic algorithm. In which the most discussed parameter in literature like population size, maximum number of generations and selection scheme were optimized. Grid search method is used for optimization of parameters. In Layer 2 x- fold cross validation was performed. Cross validation technique divides the data into testing and training set and for each iteration of the validation process a different sample set is used for training and the rest of the sample set are used for validation. The parameters optimized in this layer were the number of iterations and the sampling scheme type. Stratified sampling was used in order to preserve the ratio of all types of data records. . In Layer 3 mostly reduction in variance was done by using bagging. Which is a meta algorithm and a special case of model averaging. Parameter optimization was done for this layer as well and number of iteration and sample size were optimized. So up till now using grid search optimization in total 5 parameters were optimized and in total 161051 combinations were tested. While layer 4 reduced bias by optimizing the parameter of classifiers selected for the testing. All classifiers were tuned to give best performance as discussed in chapter 3. We proved through experimentation that if classifiers are enclosed in our model there accuracy improves form 1%-30% depending upon the data set, model complexity and it capability of handling bias and variance. Our model yielded better results than when the classifiers were used alone or in combination. In total 7 different dataset were used in training and testing of GMC model. All these dataset are public dataset and have been used in many other data mining researches. The software used for implementation of GMC model was RapidMiner 5 which is industry standard open source software for data mining. It contains over 250 different data mining algorithms. RapidMiner comes with various different extensions related to text mining and web mining etc. One such extension is WEKA and we have also tested some of the WEKA implementations of different classifiers in this research work as well.

6.2 Further work:

The model can be further optimized for extremely large data set in real time. In that case the optimization will focus on the reduction of execution time as well as further improvement in accuracy. Parallel processing can be introduced into the model for minimizing time. Parallel processing operators can be implemented using various data mining tools. RapidMiner also support parallel processing in some cases. So further research can be carried out on the usage of threads and their impact on the model optimization and classification accuracy rate; keeping in view the execution time. These parallel processing models can then be tested for large scale real time dataset. In which data is dynamic and changing with time. The research carried out in this thesis was for static datasets and all the techniques and methods used were selected accordingly. For dynamic dataset the techniques, especially the optimization techniques might vary. There are a lot of optimization techniques available a separate research and comparison can be carried out between all those techniques and the effect of those techniques on the global model for optimization. Furthermore, research can be carried out on this model for unsupervised learning problems with data sets related to more diverse fields.

As this model is applicable to all the fields in which supervised learning methods are used like image processing, medicine, statistics, education, transactional databases, facial recognition, voice recognition, video recognition etc. Very large scale real time dataset exist for all these fields therefore further research is required in order to make a better global optimization ensemble model that can deal with large datasets as well.

REFERENCES

REFERENCES

[1] Han, J. & Kamber, M. (2006), Data Mining: Concepts & Techniques (2nd Ed).

[2] Jiawei Han, Micheline Kamber, Jian Pei, “Data Mining: Concepts and Techniques”, 2nd edition, 2005, Morgan Kaufmann, ISBN 1558609016

[3] S. Kotsiantis, D. Kanellopoulos, P. Pintelas, "Data Preprocessing for Supervised Leaning", International Journal of Computer Science, 2006, Vol 1 N. 2, pp 111–117.

[4] Sujata Dash, Bichitrananda Patra, B.K. Tripathy “A Hybrid Data Mining Technique for Improving the Classification Accuracy of Microarray Data Set”, I.J. Information Engineering and Electronic Business, 2012, 2, 43-50

[5] Shih-Wei Lin, Shih-Chieh Chen, “PSOLDA: A particle swarm optimization approach for enhancing classification accuracy rate of linear discriminant analysis”, Applied Soft Computing 9 (2009) 1008–1015

[6] R. Bryll et al. “Attribute bagging: improving accuracy of classifier ensembles by using random feature subsets”, Pattern Recognition 36 (2003) 1291 – 1302

[7] Dean W. Abbott, “Combining Models to Improve Classifier Accuracy and Robustness”, 1999, Proceedings of Second International Conference on Information Fusion, Fusion ’99, Volume I, pages 289–295.

[8] S.Y. Sohn, S.H. Lee, “Data fusion, ensemble and clustering to improve the classification accuracy for the severity of road traffic accidents in Korea”, Safety Science 41 (2003) 1–14

[9] Michael R. Smith and Tony Martinez, “Improving Classification Accuracy by Identifying and Removing Instances that Should Be Misclassified”, (IJCNN 2011). 2690-2697.

[10] Zamalloayz, M., Rodriguez-Fuentesy, L.J., Penagarikanoy, M., Bordely, G., Uribez, J.P.: Feature Dimensionality Reduction Through Genetic Algorithms For Faster Speaker Recognition. In: EUSIPCO 2008 16th European Signal Processing Conference (2008)

[11] B. Liu; B. McKay and H. A. Abbass, “Improving Genetic Classifiers with a Boosting Algorithm,” The Congress on Evolutionary Computation, (CEC’03), Vol. 4, 8-12 December 2003, pp. 2596-2602.

[12] Michael L. Raymer et al., “Dimensionality Reduction Using Genetic Algorithms”, IEEE Transactions on Evolutionary Computation, Vol. 4, No. 2, July 2000

[13] D Nagesh Kumar, Optimization Methods: Advanced Topics in Optimization - Direct and Indirect Search Methods, http://www.nptel.iitm.ac.in/courses/Webcourse-contents/IIScBANG/OPTIMIZATION%20METHODS/pdf/Module_8/M8L4_LN.pdf

[14] Paul Bartholomew, “Mapping and Modeling Chlorophyll-a Concentrations in the Lake Manassas Reservoir Using Landsat Thematic Mapper Satellite Imagery”, Thesis submitted to the Faculty of Virginia Polytechnic Institute and State University, 2002

[15]Sandhya Joshi et al., “Classification and treatment of different stages of alzheimer’s disease using various machine learning methods”, International Journal of Bioinformatics Research, ISSN: 0975–3087, Volume 2, Issue 1, 2010, pp-44-52

[16] RapidMiner, published online: November 2012,http://en.wikipedia.org/wiki/RapidMiner

[17] UET Texila, online notes, http://web.uettaxila.edu.pk/cms/sp2012/mtaibs/notes%5CKNN%20Classifier.pdf, 2008

[18] Ola Söder, 2008, http://www.fon.hum.uva.nl/praat/manual/kNN_classifiers_1__What_is_a_kNN_classifier_.html

[19] SR Safavian – 1991, ftp://piekraste.daba.lv/pub/Grafika/atteelu_analiize/.../SMC91.pdf

[20] Jia Li, Department of Statistics, The Pennsylvania State University, http://sites.stat.psu.edu/~jiali/course/stat597e/notes2/trees.pdf

[21] Quinlan, J. R., “Induction of Decision Trees.” Mach. Learn. 1, 1 Mar. 1986, 81-106

[22] Mitchell, Tom M. “Machine Learning”. McGraw-Hill, 1997. pp. 55-58

[23] Breiman, Leo, "Random Forests". Machine Learning 45 (1): 5–32. , 2001

[24] Ho, Tin Kam, "Random Decision Forest". Proceedings of the 3rd International Conference on Document Analysis and Recognition, Montreal, QC, August 1995. pp. 278–282.

[25] Hosmer, David W.; Lemeshow, Stanley “Applied Logistic Regression”, (2nd Ed.). Wiley, 2000), ISBN 0-471-35632-8

[26] Cohen, Jacob; Cohen, Patricia; West, Steven G.; Aiken, Leona S. , “Applied Multiple Regression/Correlation Analysis for the Behavioral Sciences”, (3rd Ed.), Routledge, (2002), ISBN 978-0-8058-2223-6

[27] Quinlan, J. R., "Generating production rules from decision trees". In McDermott, John. Proceedings of the Tenth International Joint Conference on Artificial Intelligence, 1987, pp. 304–307

[28] Rapid miner Tutoiral, Oct 2009, rapid-i.com/downloads/tutorial/rapidminer-4.6-tutorial.pdf

[29] F. Aiolli and A. Sperduti , “A Preference Optimization Based Unifying Framework for Supervised Learning Problems”, Preference Learning, DOI 10.1007/978-3-642-14125-6_2,_Springer-Verlag Berlin Heidelberg, 2010

[30] Datasets used for classification, “http://www.is.umk.pl/projects/datasets.html”

APPENDICES

APPENDIX A: Definitions

- **Support Vector Machine(SVM)**:

  The basic SVM algorithm takes a set of input data and predicts, for each given input, which of two possible classes forms the output, making it a non-probabilistic binary linear classifier. Given a set of training examples, each marked as belonging to one of two categories, an SVM training algorithm builds a model that assigns new examples into one category or the other. An SVM model is a representation of the examples as points in space, mapped so that the examples of the separate categories are divided by a clear gap that is as wide as possible. New examples are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall on. SVM can be used for classification or regression.

- **Kernel Methods**:

  It's another class of algorithm for pattern recognition or analysis. Kernel methods map the data into higher dimensional spaces in the hope that in this higher-dimensional space the data could become more easily separated or better structured. There are also no constraints on the form of this mapping, which could even lead to infinite-dimensional spaces.

- **Partial Least Square (PLS)**:

  It's a statistical method that bears some relation to principal components regression; instead of finding hyper planes of minimum variance between the response and independent variables, it finds a linear regression model by projecting the predicted variables and the observable variables to a new space. The PLS regression model attempts to find a small number of linear combinations of the original independent variables which maximize the covariance between the dependent variable and the PLS components.

- **Multi-Layer perceptron Network (MLP)**:

  Error back propagation neural network is a feed forward multilayer perceptron (MLP) that is applied in many fields due to its powerful and stable learning algorithm. The neural network learns the training examples by adjusting the synaptic weight according to the error occurred on the output layer. The back propagation algorithm has two main advantages: local for updating the synaptic weights and biases, and efficient for computing all the partial derivatives of the cost function with respect to these free parameters. A perceptron is a simple pattern classifier.

- **RBFN**:

  The radial basis function (RBF) network is a special type of neural networks with several distinctive features. A RBF network consists of three layers, namely the input layer, the hidden layer, and the output layer. The input layer broadcasts the coordinates of the input vector to each of the units in the hidden layer. Each unit in the hidden layer then produces an activation based on the associated radial basis function. Finally, each unit in the output layer computes a linear combination of the activations of the hidden units. How a RBF network reacts to a given input stimulus is completely determined by the activation functions associated with the hidden units and the weights associated with the links between the hidden layer and the output layer.

- **PSO**:

  Particle swarm optimization is a computational method that optimizes a problem by iteratively trying to improve a candidate solution with regard to a given measure of quality. PSO optimizes a problem by having a population of candidate solutions, here dubbed particles, and moving these particles around in the search-space according to simple mathematical formulae over the particle's position and velocity. PSO is a meta-heuristic as it makes few or no assumptions about the problem being optimized and can search very large spaces of candidate solutions. However, meta-heuristics such as PSO do not guarantee an optimal solution is ever found. PSO can therefore also be used on optimization problems that are partially irregular, noisy, change over time, etc.

- **LDA**:

  Linear discriminant analysis is a method used in statistics, pattern recognition and machine learning to find a linear combination of features which characterizes or separates two or more classes of objects or events. The resulting combination may be used as a linear classifier or, more commonly, for dimensionality reduction before later classification.

- **PCA**:

  It is a way of identifying patterns in data, and expressing the data in such a way as to highlight their similarities and differences. Since patterns in high dimension data are hard to find, where the luxury of graphical representation is not available. Once these patterns are found in the data, it can be compressed i.e. by reducing the number of dimensions, without much loss of information. This is done by computing the eigen vectors and covariance matrix, then sorting them according to the corresponding eigen values, in descending order, and finally building the projection matrix.

- **Bagging:**

  Bootstrap aggregating (bagging) is a machine learning ensemble meta-algorithm to improve machine learning of statistical classification and regression models in terms of stability and classification accuracy. It also reduces variance and helps to avoid over fitting. Although it is usually applied to decision tree models, it can be used with any type of model. Bagging is a special case of the model averaging approach.

- **Boosting:**

  It is a machine learning meta-algorithm for reducing bias in supervised learning. It is the process of turning a weak learner into a strong learner.

- **Cross-Validation:**

  The data set is divided into *k* subsets, and the holdout method is repeated *k* times. Each time, one of the *k* subsets is used as the test set and the other *k-1* subsets are put together to form a training set. Then the average error across all *k* trials is computed. The advantage of this method is that it matters less how the data gets divided. Every data point gets to be in a test set exactly once, and gets to be in a training set *k-1* times. The variance of the resulting estimate is reduced as *k* is increased.

- **Holdout method:**

  The simplest kind of cross validation. It has fixed training and testing dataset partitions. The advantage of this method is that it is usually preferable to the residual method and takes no longer to compute. However, its evaluation can have a high variance. The evaluation may depend heavily on which data points end up in the training set and which end up in the test set, and thus the evaluation may be significantly different depending on how the division is made.

- **Meta-heuristic**:

  Designates a computational method that optimizes a problem by iteratively trying to improve a candidate solution with regard to a given measure of quality. Meta-heuristics make few or no assumptions about the problem being optimized and can search very large spaces of candidate solutions.

- **Ensemble Methods:**

  Use multiple models to obtain better predictive performance than could be obtained from any of the constituent models.

- **Clustering**:

  It is the task of grouping a set of objects in such a way that objects in the same group (called cluster) are more similar (in some sense or another) to each other than to those in other groups (clusters)

- **Decision Trees:**

  Decision Trees are used in statistics, data mining and machine learning, uses a decision tree as a predictive model which maps observations about an item to conclusions about the item's target value. More descriptive names for such tree models are classification trees or regression trees

- **Neural Networks:**

  An artificial neural network (ANN) learning algorithm, usually called "neural network" (NN), is a learning algorithm that is inspired by the structure and functional aspects of biological neural networks. Computations are structured in terms of an interconnected group of artificial neurons, processing information using a connectionist approach to computation. Modern neural networks are non-linear statistical data modeling tools. They are usually used to model complex relationships between inputs and outputs, to find patterns in data, or to capture the statistical structure in an unknown joint probability distribution between observed variables.

- **Genetic Algorithm**:

  This is the most popular type of EA. One seeks the solution of a problem in the form of strings of numbers (traditionally binary, although the best representations are usually those that reflect something about the problem being solved), by applying operators such as recombination and mutation (sometimes one, sometimes both). This type of EA is often used in optimization problems.

- **K-nearest Neighbor:**

  It is amongst the simplest of all machine learning algorithms: an object is classified by a majority vote of its neighbors, with the object being assigned to the class most common amongst its *k* nearest neighbors (*k* is a positive integer, typically small). If *k* = 1, then the object is simply assigned to the class of its nearest neighbor.

APPENDIX B: Code

XML code for Global optimization ensemble model for Classification Methods with KNN Classifier and Cancer Dataset

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<process version="5.2.006">
  <context>
    <input/>
    <output/>
    <macros/>
  </context>
  <operator activated="true" class="process" compatibility="5.2.006" expanded="true" name="Process">
    <process expanded="true" height="404" width="709">
      <operator activated="true" class="retrieve" compatibility="5.2.006" expanded="true" height="60" name="Retrieve" width="90" x="45" y="255">
        <parameter key="repository_entry" value="breast-cancer-dataset"/>
      </operator>
      <operator activated="true" class="numerical_to_binominal" compatibility="5.2.006" expanded="true" height="76" name="Numerical to Binominal" width="90" x="112" y="120">
        <parameter key="attribute_filter_type" value="single"/>
        <parameter key="attribute" value="Class(benign/Malignant)"/>
        <parameter key="include_special_attributes" value="true"/>
        <parameter key="max" value="2.0"/>
      </operator>
      <operator activated="true" class="replace_missing_values" compatibility="5.2.006" expanded="true" height="94" name="Replace Missing Values" width="90" x="246" y="120">
        <parameter key="include_special_attributes" value="true"/>
        <list key="columns"/>
      </operator>
      <operator activated="true" class="optimize_selection_evolutionary" compatibility="5.2.006" expanded="true" height="94" name="Optimize Selection (Evolutionary)" width="90" x="380" y="75">
        <parameter key="min_number_of_attributes" value="5"/>
        <parameter key="population_size" value="6"/>
        <parameter key="maximum_number_of_generations" value="16"/>
        <parameter key="selection_scheme" value="roulette wheel"/>
        <parameter key="crossover_type" value="shuffle"/>
        <process expanded="true" height="370" width="660">
          <operator activated="true" class="x_validation" compatibility="5.2.006" expanded="true" height="112" name="Validation" width="90" x="45" y="30">
            <process expanded="true" height="388" width="314">
              <operator activated="true" class="bagging" compatibility="5.2.006" expanded="true" height="76" name="Bagging" width="90" x="112" y="30">
                <parameter key="sample_ratio" value="0.6"/>
                <process expanded="true" height="388" width="678">
                  <operator activated="true" class="k_nn" compatibility="5.2.006" expanded="true" height="76" name="k-NN" width="90" x="164" y="139">
                    <parameter key="k" value="11"/>
                  </operator>
                  <connect from_port="training set" to_op="k-NN" to_port="training set"/>
                  <connect from_op="k-NN" from_port="model" to_port="model"/>
                  <portSpacing port="source_training set" spacing="0"/>
                  <portSpacing port="sink_model" spacing="0"/>
                </process>
              </operator>
              <connect from_port="training" to_op="Bagging" to_port="training set"/>
```

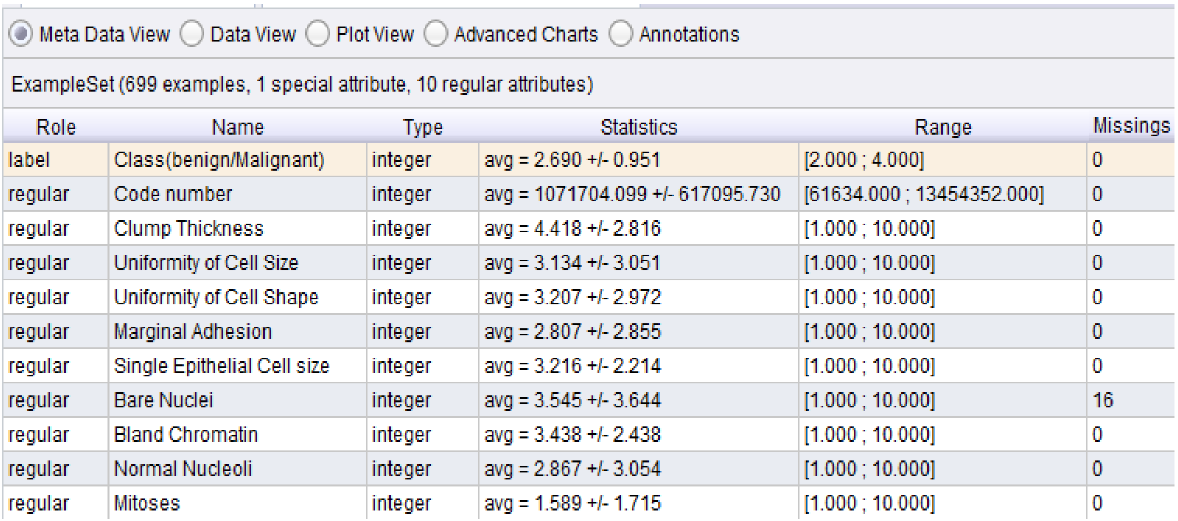
```
            <connect from_op="Bagging" from_port="model" to_port="model"/>
            <portSpacing port="source_training" spacing="0"/>
            <portSpacing port="sink_model" spacing="0"/>
            <portSpacing port="sink_through 1" spacing="0"/>
          </process>
          <process expanded="true" height="388" width="314">
            <operator activated="true" class="apply_model" compatibility="5.2.006" expanded="true" height="76" name="Apply Model" width="90" x="45" y="30">
              <list key="application_parameters"/>
            </operator>
            <operator activated="true" class="performance_classification" compatibility="5.2.006" expanded="true" height="76" name="Performance" width="90" x="179" y="30">
              <parameter key="main_criterion" value="accuracy"/>
              <parameter key="classification_error" value="true"/>
              <list key="class_weights"/>
            </operator>
            <connect from_port="model" to_op="Apply Model" to_port="model"/>
            <connect from_port="test set" to_op="Apply Model" to_port="unlabelled data"/>
            <connect from_op="Apply Model" from_port="labelled data" to_op="Performance" to_port="labelled data"/>
            <connect from_op="Performance" from_port="performance" to_port="averagable 1"/>
            <portSpacing port="source_model" spacing="0"/>
            <portSpacing port="source_test set" spacing="0"/>
            <portSpacing port="source_through 1" spacing="0"/>
            <portSpacing port="sink_averagable 1" spacing="0"/>
            <portSpacing port="sink_averagable 2" spacing="0"/>
          </process>
        </operator>
```

```
          <connect from_port="example set" to_op="Validation" to_port="training"/>
          <connect from_op="Validation" from_port="averagable 1" to_port="performance"/>
          <portSpacing port="source_example set" spacing="0"/>
          <portSpacing port="source_through 1" spacing="0"/>
          <portSpacing port="sink_performance" spacing="0"/>
        </process>
      </operator>
      <connect from_op="Retrieve" from_port="output" to_op="Numerical to Binominal" to_port="example set input"/>
      <connect from_op="Numerical to Binominal" from_port="example set output" to_op="Replace Missing Values" to_port="example set input"/>
      <connect from_op="Replace Missing Values" from_port="example set output" to_op="Optimize Selection (Evolutionary)" to_port="example set in"/>
      <connect from_op="Optimize Selection (Evolutionary)" from_port="example set out" to_port="result 1"/>
      <connect from_op="Optimize Selection (Evolutionary)" from_port="weights" to_port="result 2"/>
      <connect from_op="Optimize Selection (Evolutionary)" from_port="performance" to_port="result 3"/>
      <portSpacing port="source_input 1" spacing="0"/>
      <portSpacing port="sink_result 1" spacing="0"/>
      <portSpacing port="sink_result 2" spacing="0"/>
      <portSpacing port="sink_result 3" spacing="0"/>
      <portSpacing port="sink_result 4" spacing="0"/>
    </process>  </operator></process>
```

APPENDIX C: Dataset Samples

<u>Cancer Dataset: </u>[30]

Meta-Data View:

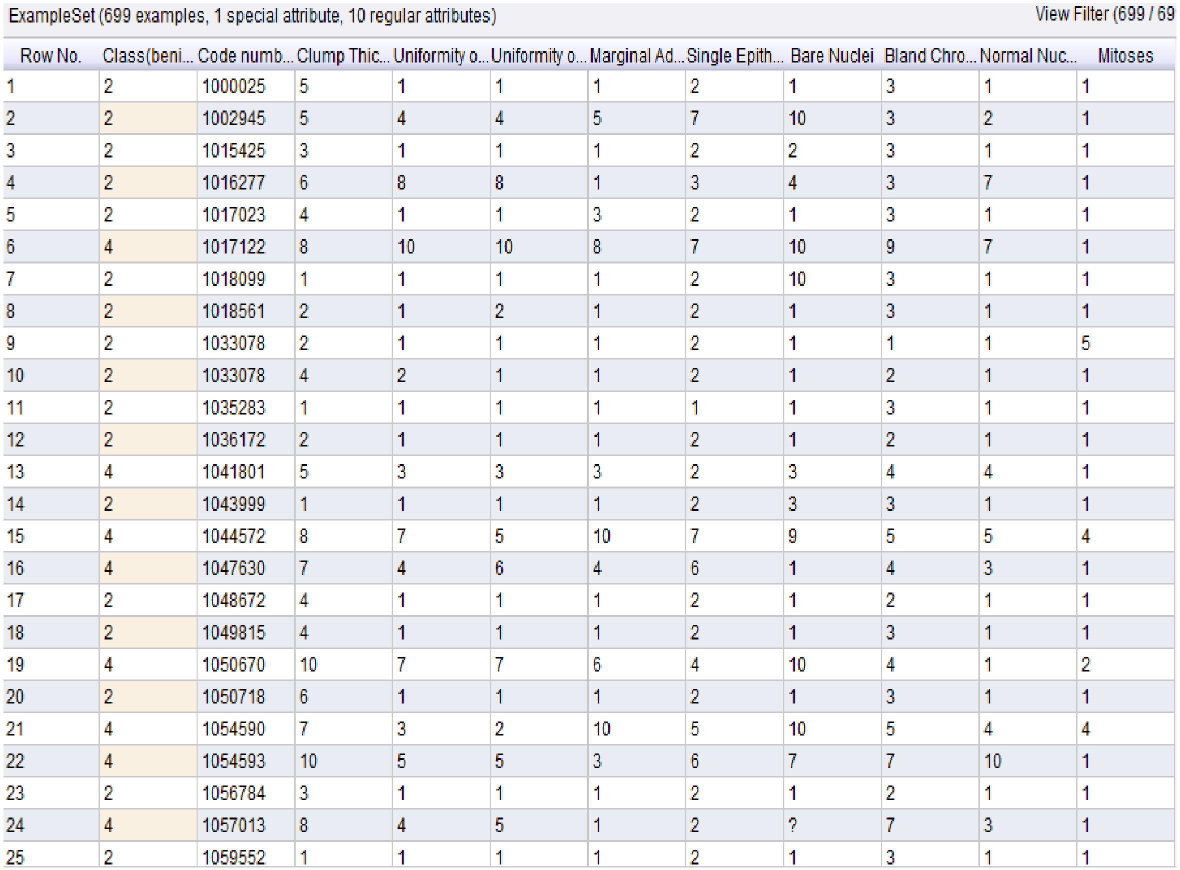


● Meta Data View ○ Data View ○ Plot View ○ Advanced Charts ○ Annotations

ExampleSet (699 examples, 1 special attribute, 10 regular attributes)

Role	Name	Type	Statistics	Range	Missings
label	Class(benign/Malignant)	integer	avg = 2.690 +/- 0.951	[2.000 ; 4.000]	0
regular	Code number	integer	avg = 1071704.099 +/- 617095.730	[61634.000 ; 13454352.000]	0
regular	Clump Thickness	integer	avg = 4.418 +/- 2.816	[1.000 ; 10.000]	0
regular	Uniformity of Cell Size	integer	avg = 3.134 +/- 3.051	[1.000 ; 10.000]	0
regular	Uniformity of Cell Shape	integer	avg = 3.207 +/- 2.972	[1.000 ; 10.000]	0
regular	Marginal Adhesion	integer	avg = 2.807 +/- 2.855	[1.000 ; 10.000]	0
regular	Single Epithelial Cell size	integer	avg = 3.216 +/- 2.214	[1.000 ; 10.000]	0
regular	Bare Nuclei	integer	avg = 3.545 +/- 3.644	[1.000 ; 10.000]	16
regular	Bland Chromatin	integer	avg = 3.438 +/- 2.438	[1.000 ; 10.000]	0
regular	Normal Nucleoli	integer	avg = 2.867 +/- 3.054	[1.000 ; 10.000]	0
regular	Mitoses	integer	avg = 1.589 +/- 1.715	[1.000 ; 10.000]	0

Data View:

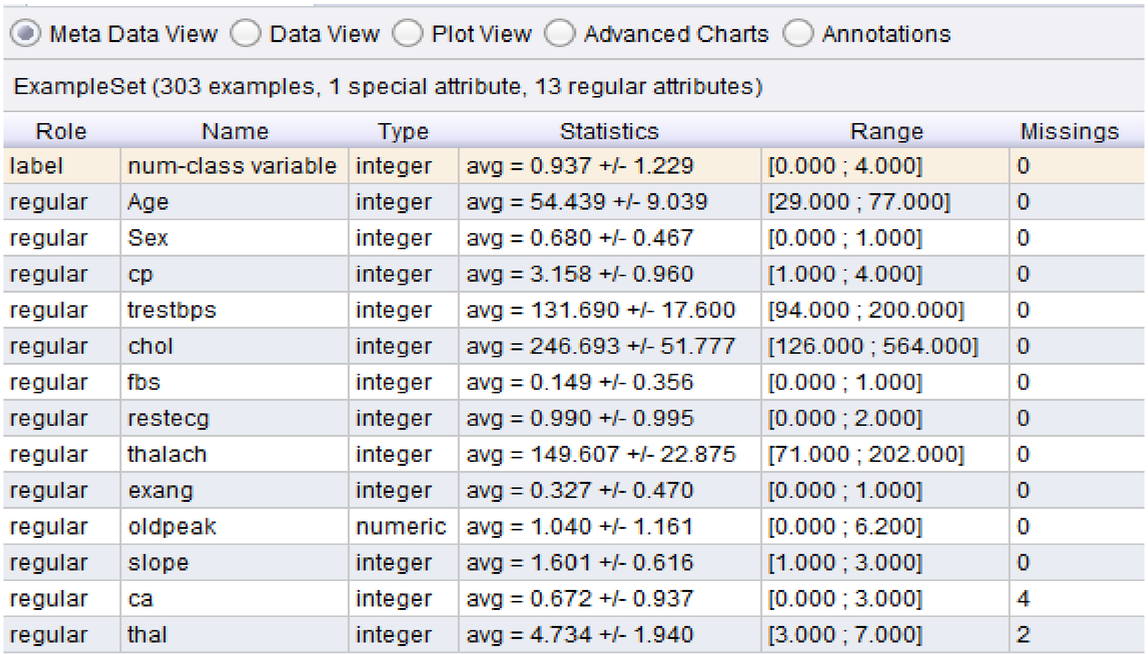


ExampleSet (699 examples, 1 special attribute, 10 regular attributes) View Filter (699 / 69

Row No.	Class(beni...	Code numb...	Clump Thic...	Uniformity o...	Uniformity o...	Marginal Ad...	Single Epith...	Bare Nuclei	Bland Chro...	Normal Nuc...	Mitoses
1	2	1000025	5	1	1	1	2	1	3	1	1
2	2	1002945	5	4	4	5	7	10	3	2	1
3	2	1015425	3	1	1	1	2	2	3	1	1
4	2	1016277	6	8	8	1	3	4	3	7	1
5	2	1017023	4	1	1	3	2	1	3	1	1
6	4	1017122	8	10	10	8	7	10	9	7	1
7	2	1018099	1	1	1	1	2	10	3	1	1
8	2	1018561	2	1	2	1	2	1	3	1	1
9	2	1033078	2	1	1	1	2	1	1	1	5
10	2	1033078	4	2	1	1	2	1	2	1	1
11	2	1035283	1	1	1	1	1	1	3	1	1
12	2	1036172	2	1	1	1	2	1	2	1	1
13	4	1041801	5	3	3	3	2	3	4	4	1
14	2	1043999	1	1	1	1	2	3	3	1	1
15	4	1044572	8	7	5	10	7	9	5	5	4
16	4	1047630	7	4	6	4	6	1	4	3	1
17	2	1048672	4	1	1	1	2	1	2	1	1
18	2	1049815	4	1	1	1	2	1	3	1	1
19	4	1050670	10	7	7	6	4	10	4	1	2
20	2	1050718	6	1	1	1	2	1	3	1	1
21	4	1054590	7	3	2	10	5	10	5	4	4
22	4	1054593	10	5	5	3	6	7	7	10	1
23	2	1056784	3	1	1	1	2	1	2	1	1
24	4	1057013	8	4	5	1	2	?	7	3	1
25	2	1059552	1	1	1	1	2	1	3	1	1

Heart Disease Dataset: [30]

Meta-Data View:

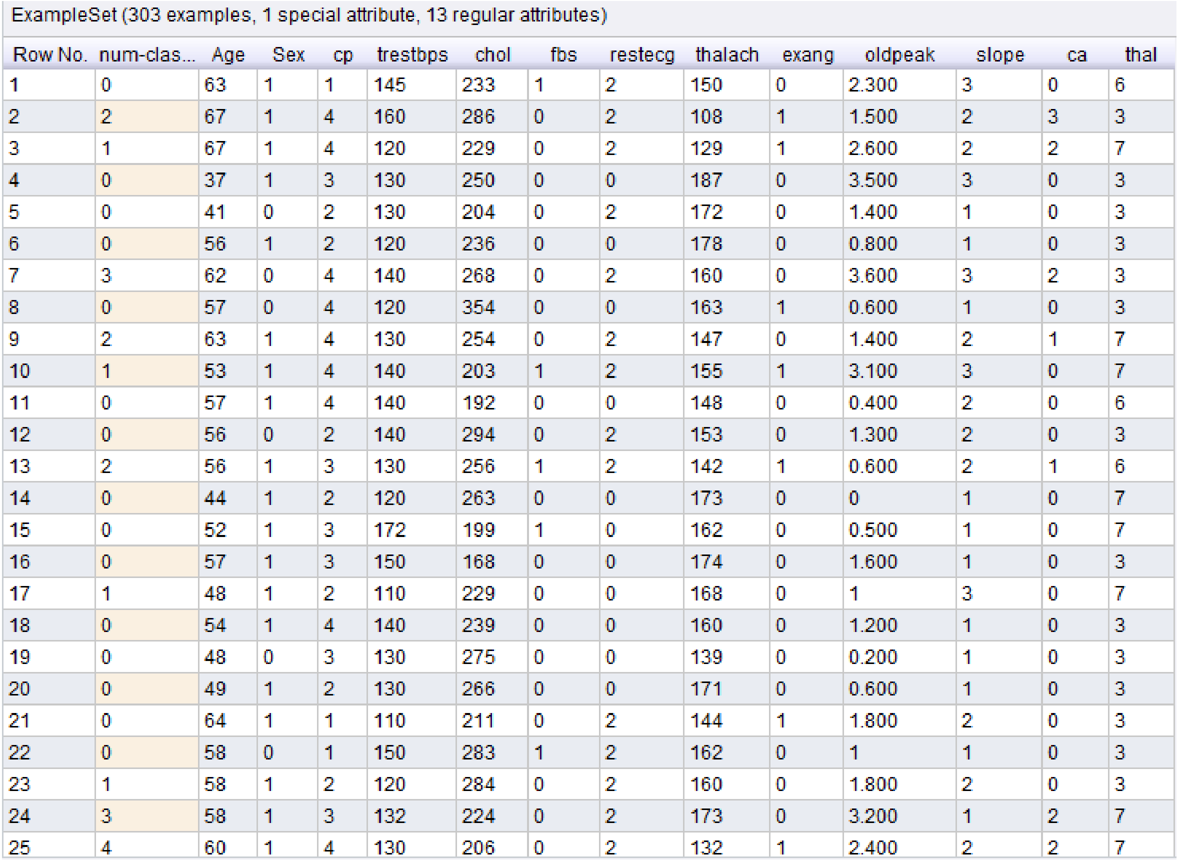


● Meta Data View ○ Data View ○ Plot View ○ Advanced Charts ○ Annotations

ExampleSet (303 examples, 1 special attribute, 13 regular attributes)

Role	Name	Type	Statistics	Range	Missings
label	num-class variable	integer	avg = 0.937 +/- 1.229	[0.000 ; 4.000]	0
regular	Age	integer	avg = 54.439 +/- 9.039	[29.000 ; 77.000]	0
regular	Sex	integer	avg = 0.680 +/- 0.467	[0.000 ; 1.000]	0
regular	cp	integer	avg = 3.158 +/- 0.960	[1.000 ; 4.000]	0
regular	trestbps	integer	avg = 131.690 +/- 17.600	[94.000 ; 200.000]	0
regular	chol	integer	avg = 246.693 +/- 51.777	[126.000 ; 564.000]	0
regular	fbs	integer	avg = 0.149 +/- 0.356	[0.000 ; 1.000]	0
regular	restecg	integer	avg = 0.990 +/- 0.995	[0.000 ; 2.000]	0
regular	thalach	integer	avg = 149.607 +/- 22.875	[71.000 ; 202.000]	0
regular	exang	integer	avg = 0.327 +/- 0.470	[0.000 ; 1.000]	0
regular	oldpeak	numeric	avg = 1.040 +/- 1.161	[0.000 ; 6.200]	0
regular	slope	integer	avg = 1.601 +/- 0.616	[1.000 ; 3.000]	0
regular	ca	integer	avg = 0.672 +/- 0.937	[0.000 ; 3.000]	4
regular	thal	integer	avg = 4.734 +/- 1.940	[3.000 ; 7.000]	2

Data View:

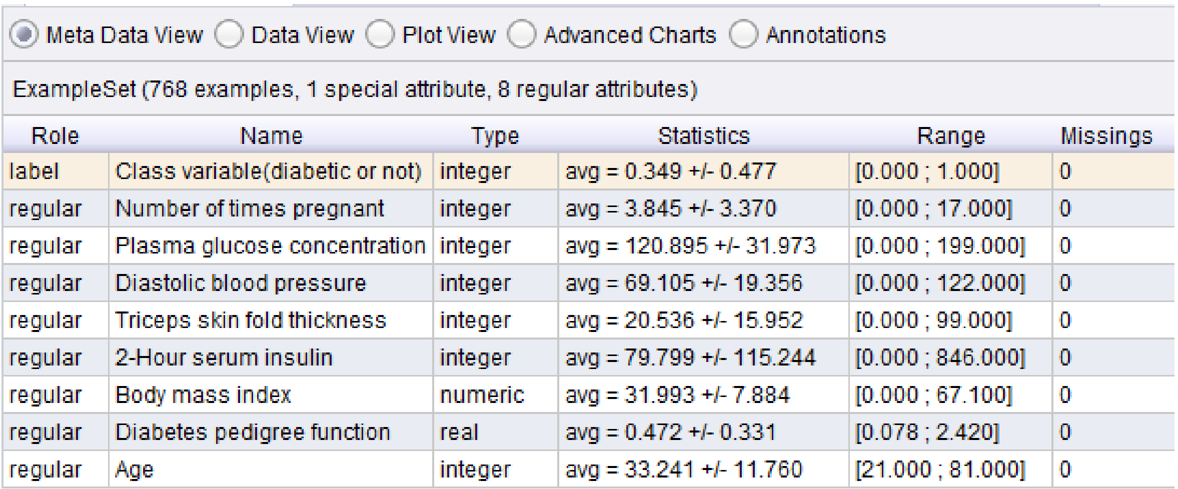


ExampleSet (303 examples, 1 special attribute, 13 regular attributes)

Row No.	num-clas...	Age	Sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal
1	0	63	1	1	145	233	1	2	150	0	2.300	3	0	6
2	2	67	1	4	160	286	0	2	108	1	1.500	2	3	3
3	1	67	1	4	120	229	0	2	129	1	2.600	2	2	7
4	0	37	1	3	130	250	0	0	187	0	3.500	3	0	3
5	0	41	0	2	130	204	0	2	172	0	1.400	1	0	3
6	0	56	1	2	120	236	0	0	178	0	0.800	1	0	3
7	3	62	0	4	140	268	0	2	160	0	3.600	3	2	3
8	0	57	0	4	120	354	0	0	163	1	0.600	1	0	3
9	2	63	1	4	130	254	0	2	147	0	1.400	2	1	7
10	1	53	1	4	140	203	1	2	155	1	3.100	3	0	7
11	0	57	1	4	140	192	0	0	148	0	0.400	2	0	6
12	0	56	0	2	140	294	0	2	153	0	1.300	2	0	3
13	2	56	1	3	130	256	1	2	142	1	0.600	2	1	6
14	0	44	1	2	120	263	0	0	173	0	0	1	0	7
15	0	52	1	3	172	199	1	0	162	0	0.500	1	0	7
16	0	57	1	3	150	168	0	0	174	0	1.600	1	0	3
17	1	48	1	2	110	229	0	0	168	0	1	3	0	7
18	0	54	1	4	140	239	0	0	160	0	1.200	1	0	3
19	0	48	0	3	130	275	0	0	139	0	0.200	1	0	3
20	0	49	1	2	130	266	0	0	171	0	0.600	1	0	3
21	0	64	1	1	110	211	0	2	144	1	1.800	2	0	3
22	0	58	0	1	150	283	1	2	162	0	1	1	0	3
23	1	58	1	2	120	284	0	2	160	0	1.800	2	0	3
24	3	58	1	3	132	224	0	2	173	0	3.200	1	2	7
25	4	60	1	4	130	206	0	2	132	1	2.400	2	2	7

Diabetes Dataset: [30]

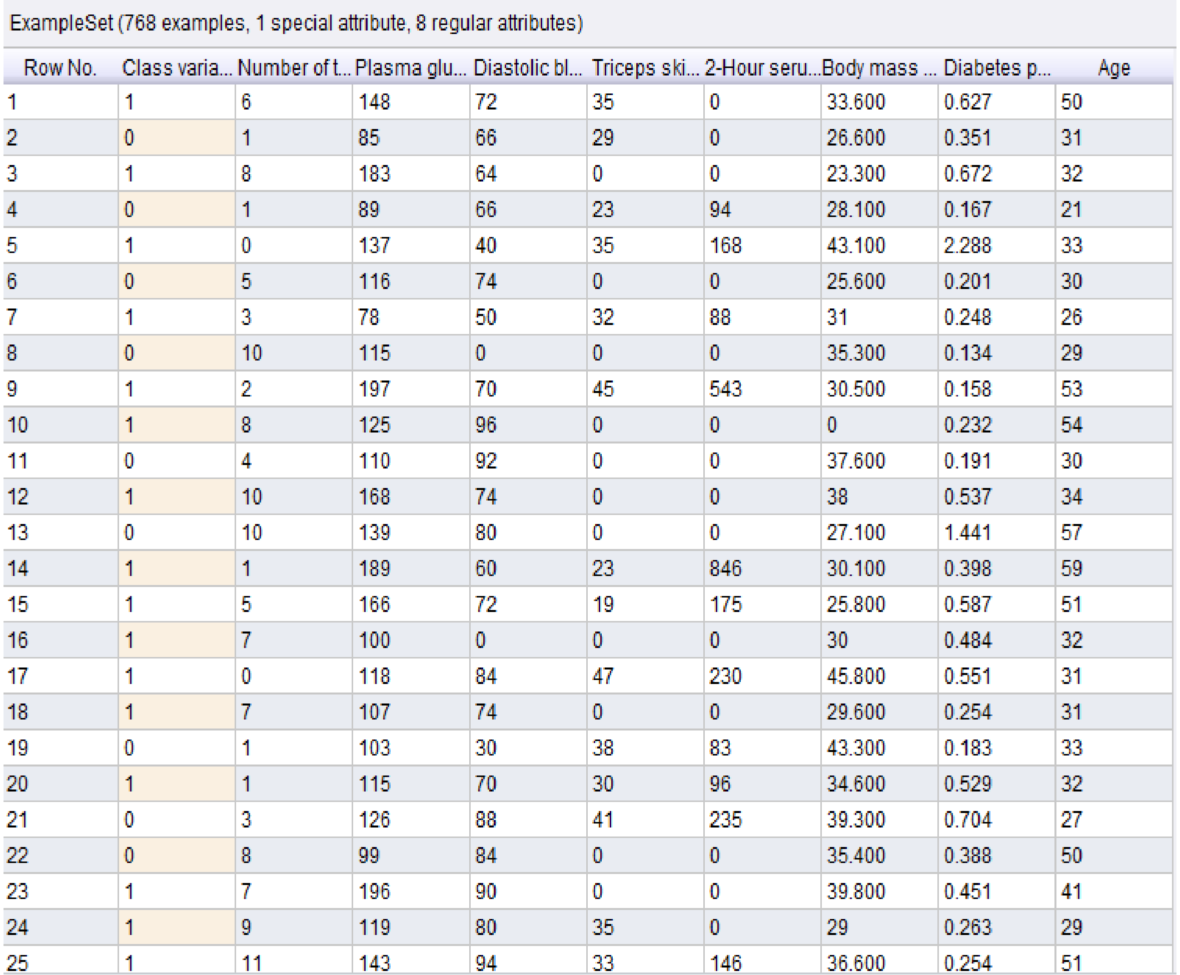
Meta-Data View:


● Meta Data View ○ Data View ○ Plot View ○ Advanced Charts ○ Annotations

ExampleSet (768 examples, 1 special attribute, 8 regular attributes)

Role	Name	Type	Statistics	Range	Missings
label	Class variable(diabetic or not)	integer	avg = 0.349 +/- 0.477	[0.000 ; 1.000]	0
regular	Number of times pregnant	integer	avg = 3.845 +/- 3.370	[0.000 ; 17.000]	0
regular	Plasma glucose concentration	integer	avg = 120.895 +/- 31.973	[0.000 ; 199.000]	0
regular	Diastolic blood pressure	integer	avg = 69.105 +/- 19.356	[0.000 ; 122.000]	0
regular	Triceps skin fold thickness	integer	avg = 20.536 +/- 15.952	[0.000 ; 99.000]	0
regular	2-Hour serum insulin	integer	avg = 79.799 +/- 115.244	[0.000 ; 846.000]	0
regular	Body mass index	numeric	avg = 31.993 +/- 7.884	[0.000 ; 67.100]	0
regular	Diabetes pedigree function	real	avg = 0.472 +/- 0.331	[0.078 ; 2.420]	0
regular	Age	integer	avg = 33.241 +/- 11.760	[21.000 ; 81.000]	0

Data View:

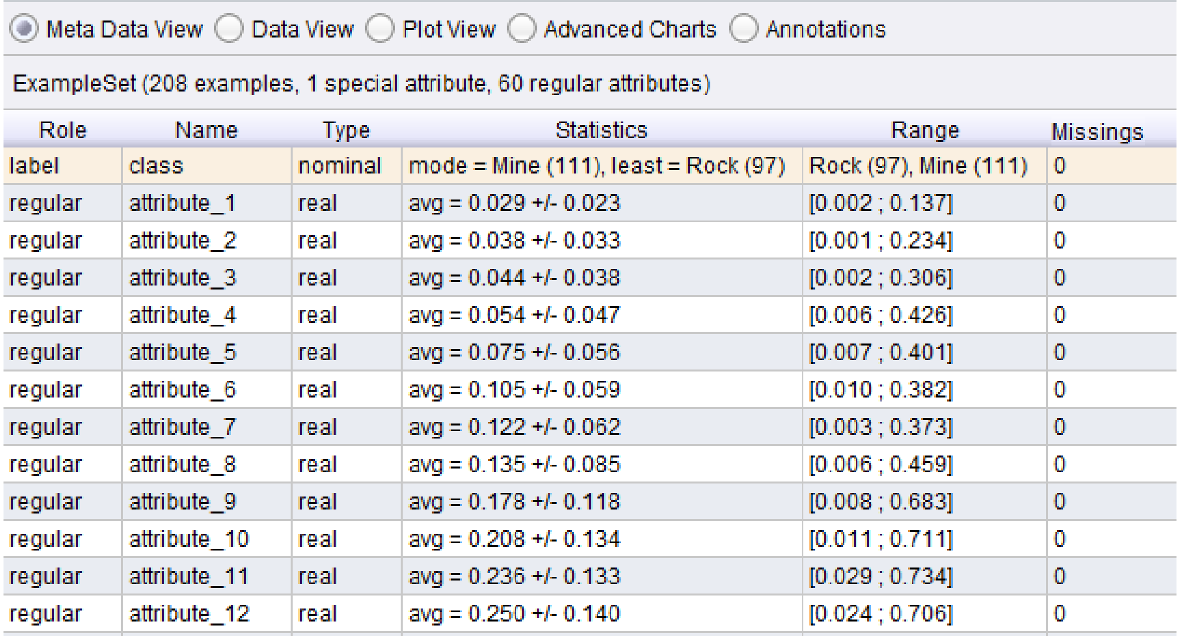


ExampleSet (768 examples, 1 special attribute, 8 regular attributes)

Row No.	Class varia...	Number of t...	Plasma glu...	Diastolic bl...	Triceps ski...	2-Hour seru...	Body mass ...	Diabetes p...	Age
1	1	6	148	72	35	0	33.600	0.627	50
2	0	1	85	66	29	0	26.600	0.351	31
3	1	8	183	64	0	0	23.300	0.672	32
4	0	1	89	66	23	94	28.100	0.167	21
5	1	0	137	40	35	168	43.100	2.288	33
6	0	5	116	74	0	0	25.600	0.201	30
7	1	3	78	50	32	88	31	0.248	26
8	0	10	115	0	0	0	35.300	0.134	29
9	1	2	197	70	45	543	30.500	0.158	53
10	1	8	125	96	0	0	0	0.232	54
11	0	4	110	92	0	0	37.600	0.191	30
12	1	10	168	74	0	0	38	0.537	34
13	0	10	139	80	0	0	27.100	1.441	57
14	1	1	189	60	23	846	30.100	0.398	59
15	1	5	166	72	19	175	25.800	0.587	51
16	1	7	100	0	0	0	30	0.484	32
17	1	0	118	84	47	230	45.800	0.551	31
18	1	7	107	74	0	0	29.600	0.254	31
19	0	1	103	30	38	83	43.300	0.183	33
20	1	1	115	70	30	96	34.600	0.529	32
21	0	3	126	88	41	235	39.300	0.704	27
22	0	8	99	84	0	0	35.400	0.388	50
23	1	7	196	90	0	0	39.800	0.451	41
24	1	9	119	80	35	0	29	0.263	29
25	1	11	143	94	33	146	36.600	0.254	51

Sonar Dataset: [30]

Meta-Data View:

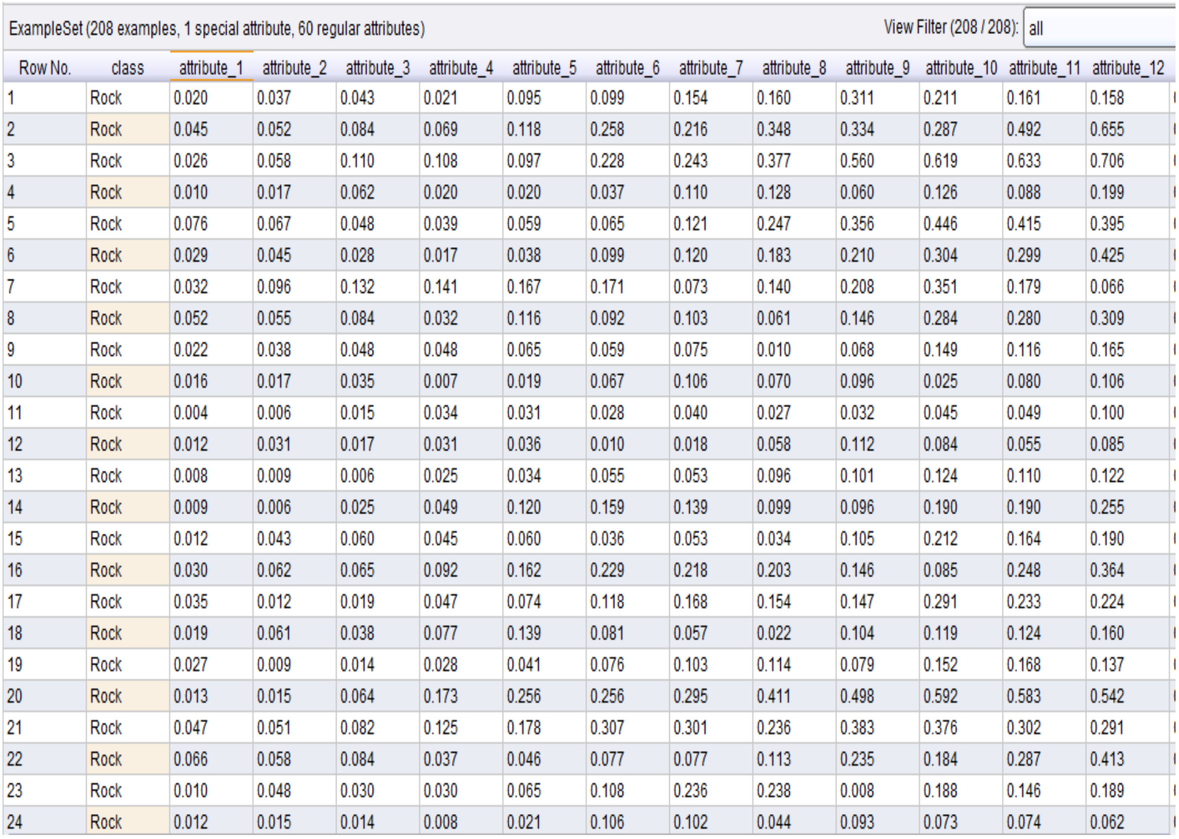


○ Meta Data View ○ Data View ○ Plot View ○ Advanced Charts ○ Annotations

ExampleSet (208 examples, 1 special attribute, 60 regular attributes)

Role	Name	Type	Statistics	Range	Missings
label	class	nominal	mode = Mine (111), least = Rock (97)	Rock (97), Mine (111)	0
regular	attribute_1	real	avg = 0.029 +/- 0.023	[0.002 ; 0.137]	0
regular	attribute_2	real	avg = 0.038 +/- 0.033	[0.001 ; 0.234]	0
regular	attribute_3	real	avg = 0.044 +/- 0.038	[0.002 ; 0.306]	0
regular	attribute_4	real	avg = 0.054 +/- 0.047	[0.006 ; 0.426]	0
regular	attribute_5	real	avg = 0.075 +/- 0.056	[0.007 ; 0.401]	0
regular	attribute_6	real	avg = 0.105 +/- 0.059	[0.010 ; 0.382]	0
regular	attribute_7	real	avg = 0.122 +/- 0.062	[0.003 ; 0.373]	0
regular	attribute_8	real	avg = 0.135 +/- 0.085	[0.006 ; 0.459]	0
regular	attribute_9	real	avg = 0.178 +/- 0.118	[0.008 ; 0.683]	0
regular	attribute_10	real	avg = 0.208 +/- 0.134	[0.011 ; 0.711]	0
regular	attribute_11	real	avg = 0.236 +/- 0.133	[0.029 ; 0.734]	0
regular	attribute_12	real	avg = 0.250 +/- 0.140	[0.024 ; 0.706]	0

Data View:



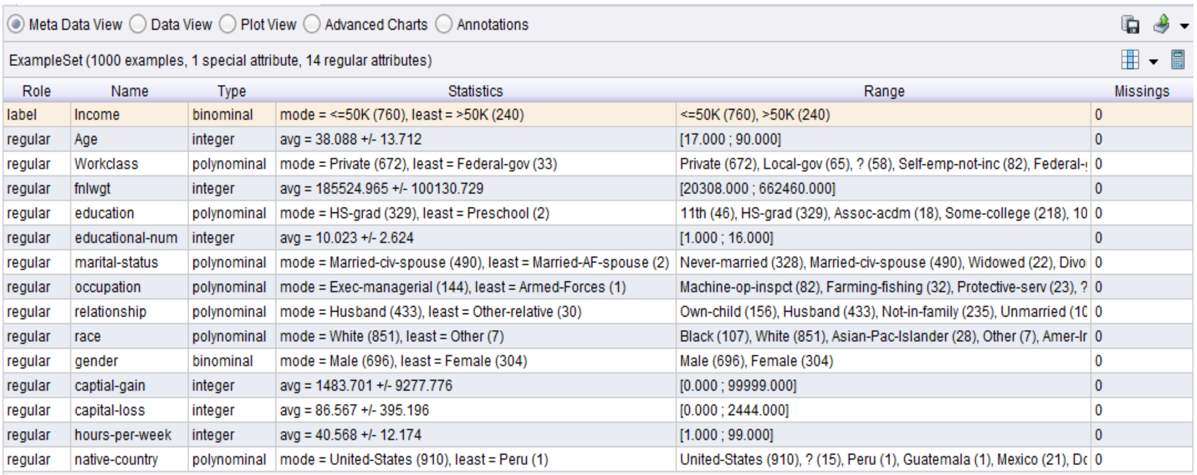
ExampleSet (208 examples, 1 special attribute, 60 regular attributes)

View Filter (208 / 208): all

Row No.	class	attribute_1	attribute_2	attribute_3	attribute_4	attribute_5	attribute_6	attribute_7	attribute_8	attribute_9	attribute_10	attribute_11	attribute_12
1	Rock	0.020	0.037	0.043	0.021	0.095	0.099	0.154	0.160	0.311	0.211	0.161	0.158
2	Rock	0.045	0.052	0.084	0.069	0.118	0.258	0.216	0.348	0.334	0.287	0.492	0.655
3	Rock	0.026	0.058	0.110	0.108	0.097	0.228	0.243	0.377	0.560	0.619	0.633	0.706
4	Rock	0.010	0.017	0.062	0.020	0.020	0.037	0.110	0.128	0.060	0.126	0.088	0.199
5	Rock	0.076	0.067	0.048	0.039	0.059	0.065	0.121	0.247	0.356	0.446	0.415	0.395
6	Rock	0.029	0.045	0.028	0.017	0.038	0.099	0.120	0.183	0.210	0.304	0.299	0.425
7	Rock	0.032	0.096	0.132	0.141	0.167	0.171	0.073	0.140	0.208	0.351	0.179	0.066
8	Rock	0.052	0.055	0.084	0.032	0.116	0.092	0.103	0.061	0.146	0.284	0.280	0.309
9	Rock	0.022	0.038	0.048	0.048	0.065	0.059	0.075	0.010	0.068	0.149	0.116	0.165
10	Rock	0.016	0.017	0.035	0.007	0.019	0.067	0.106	0.070	0.096	0.025	0.080	0.106
11	Rock	0.004	0.006	0.015	0.034	0.031	0.028	0.040	0.027	0.032	0.045	0.049	0.100
12	Rock	0.012	0.031	0.017	0.031	0.036	0.010	0.018	0.058	0.112	0.084	0.055	0.085
13	Rock	0.008	0.009	0.006	0.025	0.034	0.055	0.053	0.096	0.101	0.124	0.110	0.122
14	Rock	0.009	0.006	0.025	0.049	0.120	0.159	0.139	0.099	0.096	0.190	0.190	0.255
15	Rock	0.012	0.043	0.060	0.045	0.060	0.036	0.053	0.034	0.105	0.212	0.164	0.190
16	Rock	0.030	0.062	0.065	0.092	0.162	0.229	0.218	0.203	0.146	0.085	0.248	0.364
17	Rock	0.035	0.012	0.019	0.047	0.074	0.118	0.168	0.154	0.147	0.291	0.233	0.224
18	Rock	0.019	0.061	0.038	0.077	0.139	0.081	0.057	0.022	0.104	0.119	0.124	0.160
19	Rock	0.027	0.009	0.014	0.028	0.041	0.076	0.103	0.114	0.079	0.152	0.168	0.137
20	Rock	0.013	0.015	0.064	0.173	0.256	0.256	0.295	0.411	0.498	0.592	0.583	0.542
21	Rock	0.047	0.051	0.082	0.125	0.178	0.307	0.301	0.236	0.383	0.376	0.302	0.291
22	Rock	0.066	0.058	0.084	0.037	0.046	0.077	0.077	0.113	0.235	0.184	0.287	0.413
23	Rock	0.010	0.048	0.030	0.030	0.065	0.108	0.236	0.238	0.008	0.188	0.146	0.189
24	Rock	0.012	0.015	0.014	0.008	0.021	0.106	0.102	0.044	0.093	0.073	0.074	0.062

UCI Adult Income Dataset: [30]

Meta-Data View:

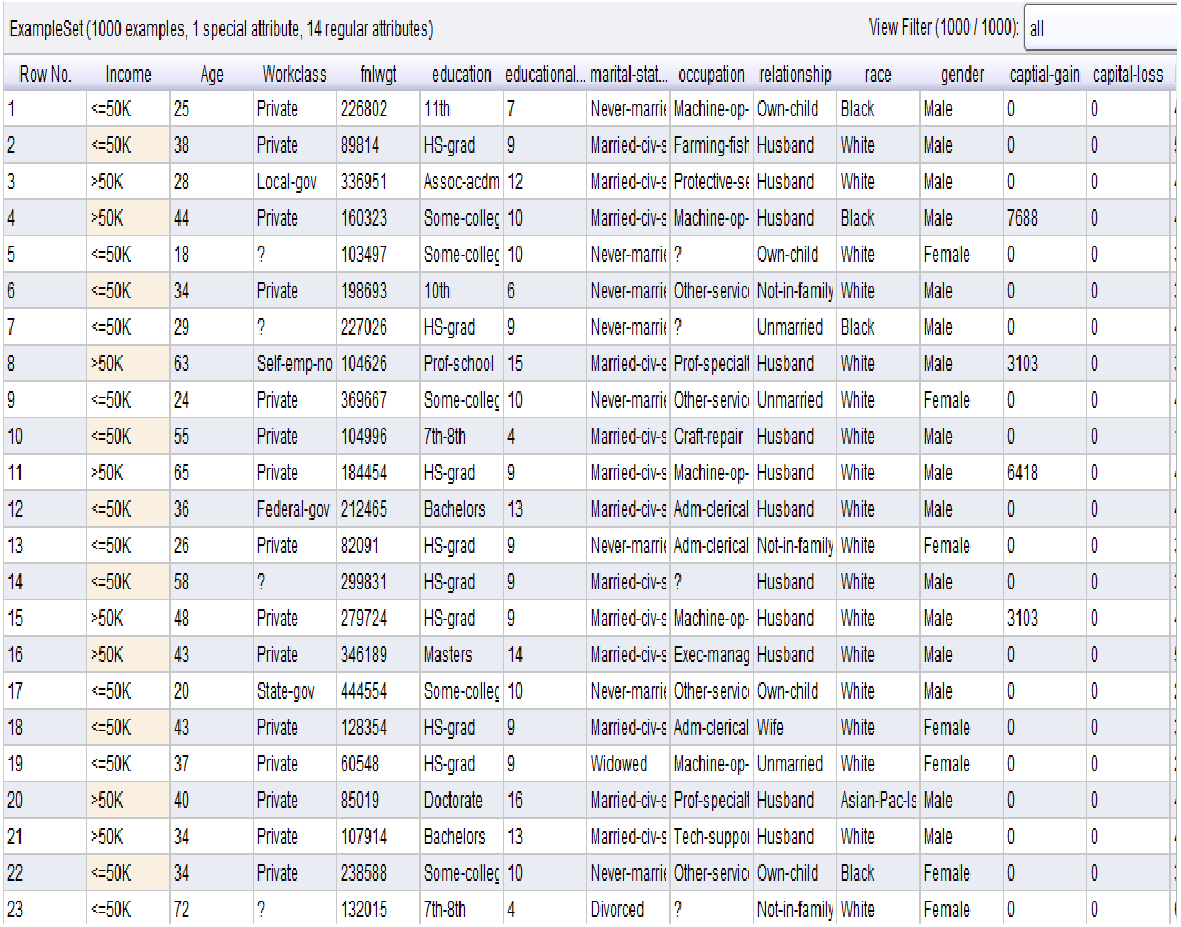


● Meta Data View ○ Data View ○ Plot View ○ Advanced Charts ○ Annotations

ExampleSet (1000 examples, 1 special attribute, 14 regular attributes)

Role	Name	Type	Statistics	Range	Missings
label	Income	binominal	mode = <=50K (760), least = >50K (240)	<=50K (760), >50K (240)	0
regular	Age	integer	avg = 38.088 +/- 13.712	[17.000 ; 90.000]	0
regular	Workclass	polynominal	mode = Private (672), least = Federal-gov (33)	Private (672), Local-gov (65), ? (58), Self-emp-not-inc (82), Federal-(	0
regular	fnlwgt	integer	avg = 185524.965 +/- 100130.729	[20308.000 ; 662460.000]	0
regular	education	polynominal	mode = HS-grad (329), least = Preschool (2)	11th (46), HS-grad (329), Assoc-acdm (18), Some-college (218), 10	0
regular	educational-num	integer	avg = 10.023 +/- 2.624	[1.000 ; 16.000]	0
regular	marital-status	polynominal	mode = Married-civ-spouse (490), least = Married-AF-spouse (2)	Never-married (328), Married-civ-spouse (490), Widowed (22), Divo	0
regular	occupation	polynominal	mode = Exec-managerial (144), least = Armed-Forces (1)	Machine-op-inspct (82), Farming-fishing (32), Protective-serv (23), ?	0
regular	relationship	polynominal	mode = Husband (433), least = Other-relative (30)	Own-child (156), Husband (433), Not-in-family (235), Unmarried (1(	0
regular	race	polynominal	mode = White (851), least = Other (7)	Black (107), White (851), Asian-Pac-Islander (28), Other (7), Amer-Ir	0
regular	gender	binominal	mode = Male (696), least = Female (304)	Male (696), Female (304)	0
regular	captial-gain	integer	avg = 1483.701 +/- 9277.776	[0.000 ; 99999.000]	0
regular	capital-loss	integer	avg = 86.567 +/- 395.196	[0.000 ; 2444.000]	0
regular	hours-per-week	integer	avg = 40.568 +/- 12.174	[1.000 ; 99.000]	0
regular	native-country	polynominal	mode = United-States (910), least = Peru (1)	United-States (910), ? (15), Peru (1), Guatemala (1), Mexico (21), Dc	0

Data View:



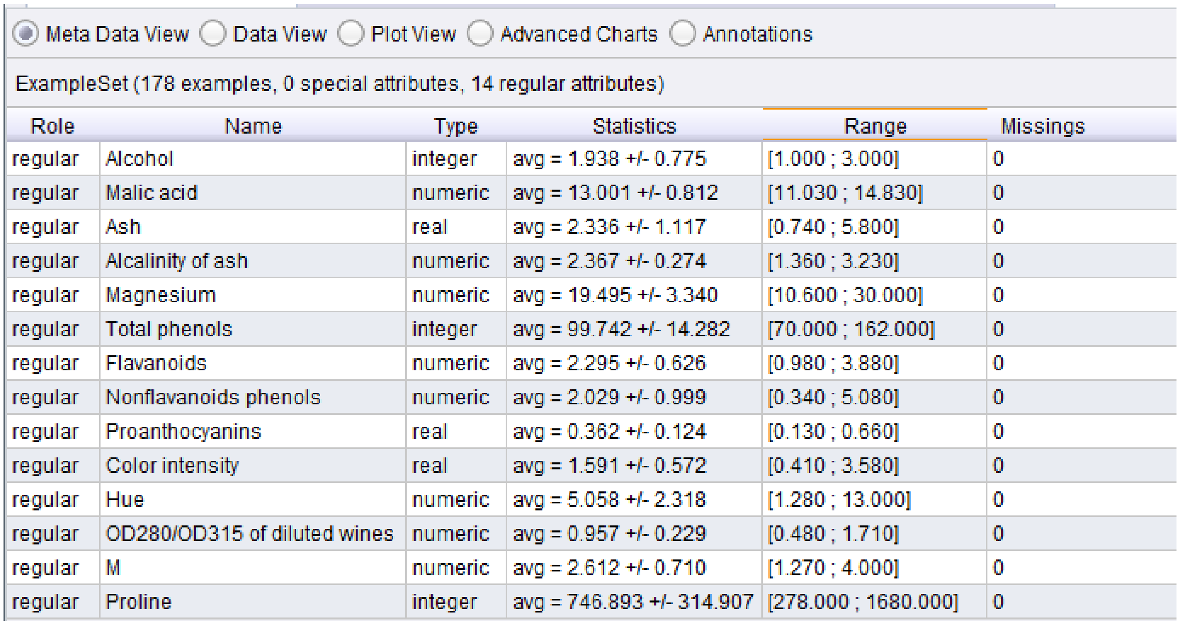
ExampleSet (1000 examples, 1 special attribute, 14 regular attributes)

View Filter (1000 / 1000): all

Row No.	Income	Age	Workclass	fnlwgt	education	educational...	marital-stat...	occupation	relationship	race	gender	captial-gain	capital-loss
1	<=50K	25	Private	226802	11th	7	Never-marrie	Machine-op-	Own-child	Black	Male	0	0
2	<=50K	38	Private	89814	HS-grad	9	Married-civ-s	Farming-fish	Husband	White	Male	0	0
3	>50K	28	Local-gov	336951	Assoc-acdm	12	Married-civ-s	Protective-se	Husband	White	Male	0	0
4	>50K	44	Private	160323	Some-colleg	10	Married-civ-s	Machine-op-	Husband	Black	Male	7688	0
5	<=50K	18	?	103497	Some-colleg	10	Never-marrie	?	Own-child	White	Female	0	0
6	<=50K	34	Private	198693	10th	6	Never-marrie	Other-servic	Not-in-family	White	Male	0	0
7	<=50K	29	?	227026	HS-grad	9	Never-marrie	?	Unmarried	Black	Male	0	0
8	>50K	63	Self-emp-no	104626	Prof-school	15	Married-civ-s	Prof-speciali	Husband	White	Male	3103	0
9	<=50K	24	Private	369667	Some-colleg	10	Never-marrie	Other-servic	Unmarried	White	Female	0	0
10	<=50K	55	Private	104996	7th-8th	4	Married-civ-s	Craft-repair	Husband	White	Male	0	0
11	>50K	65	Private	184454	HS-grad	9	Married-civ-s	Machine-op-	Husband	White	Male	6418	0
12	<=50K	36	Federal-gov	212465	Bachelors	13	Married-civ-s	Adm-clerical	Husband	White	Male	0	0
13	<=50K	26	Private	82091	HS-grad	9	Never-marrie	Adm-clerical	Not-in-family	White	Female	0	0
14	<=50K	58	?	299831	HS-grad	9	Married-civ-s	?	Husband	White	Male	0	0
15	>50K	48	Private	279724	HS-grad	9	Married-civ-s	Machine-op-	Husband	White	Male	3103	0
16	>50K	43	Private	346189	Masters	14	Married-civ-s	Exec-manag	Husband	White	Male	0	0
17	<=50K	20	State-gov	444554	Some-colleg	10	Never-marrie	Other-servic	Own-child	White	Male	0	0
18	<=50K	43	Private	128354	HS-grad	9	Married-civ-s	Adm-clerical	Wife	White	Female	0	0
19	<=50K	37	Private	60548	HS-grad	9	Widowed	Machine-op-	Unmarried	White	Female	0	0
20	>50K	40	Private	85019	Doctorate	16	Married-civ-s	Prof-speciali	Husband	Asian-Pac-Is	Male	0	0
21	>50K	34	Private	107914	Bachelors	13	Married-civ-s	Tech-suppor	Husband	White	Male	0	0
22	<=50K	34	Private	238588	Some-colleg	10	Never-marrie	Other-servic	Own-child	Black	Female	0	0
23	<=50K	72	?	132015	7th-8th	4	Divorced	?	Not-in-family	White	Female	0	0

Wine Dataset: [30]

Meta-Data View:

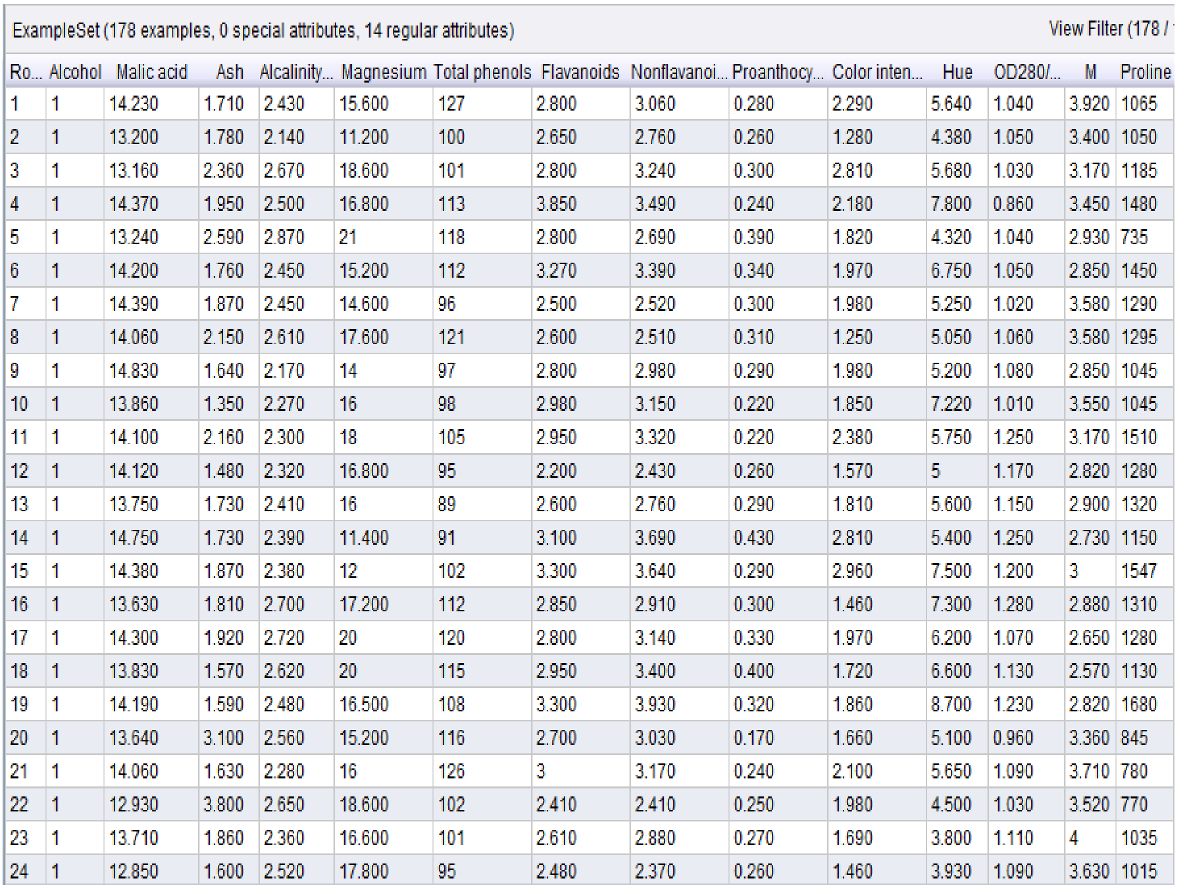


● Meta Data View ○ Data View ○ Plot View ○ Advanced Charts ○ Annotations

ExampleSet (178 examples, 0 special attributes, 14 regular attributes)

Role	Name	Type	Statistics	Range	Missings
regular	Alcohol	integer	avg = 1.938 +/- 0.775	[1.000 ; 3.000]	0
regular	Malic acid	numeric	avg = 13.001 +/- 0.812	[11.030 ; 14.830]	0
regular	Ash	real	avg = 2.336 +/- 1.117	[0.740 ; 5.800]	0
regular	Alcalinity of ash	numeric	avg = 2.367 +/- 0.274	[1.360 ; 3.230]	0
regular	Magnesium	numeric	avg = 19.495 +/- 3.340	[10.600 ; 30.000]	0
regular	Total phenols	integer	avg = 99.742 +/- 14.282	[70.000 ; 162.000]	0
regular	Flavanoids	numeric	avg = 2.295 +/- 0.626	[0.980 ; 3.880]	0
regular	Nonflavanoids phenols	numeric	avg = 2.029 +/- 0.999	[0.340 ; 5.080]	0
regular	Proanthocyanins	real	avg = 0.362 +/- 0.124	[0.130 ; 0.660]	0
regular	Color intensity	real	avg = 1.591 +/- 0.572	[0.410 ; 3.580]	0
regular	Hue	numeric	avg = 5.058 +/- 2.318	[1.280 ; 13.000]	0
regular	OD280/OD315 of diluted wines	numeric	avg = 0.957 +/- 0.229	[0.480 ; 1.710]	0
regular	M	numeric	avg = 2.612 +/- 0.710	[1.270 ; 4.000]	0
regular	Proline	integer	avg = 746.893 +/- 314.907	[278.000 ; 1680.000]	0

Data View:

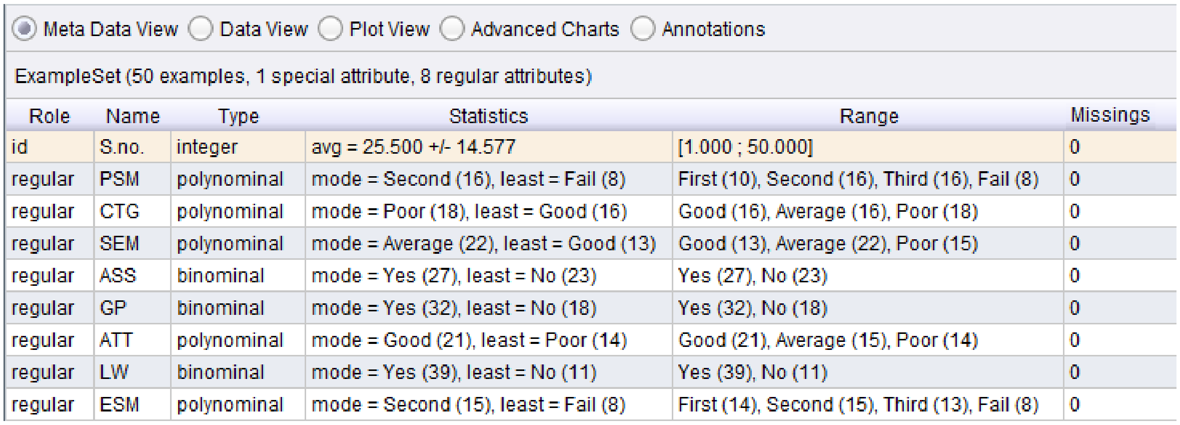


ExampleSet (178 examples, 0 special attributes, 14 regular attributes)

Ro...	Alcohol	Malic acid	Ash	Alcalinity...	Magnesium	Total phenols	Flavanoids	Nonflavanoi...	Proanthocy...	Color inten...	Hue	OD280/...	M	Proline
1	1	14.230	1.710	2.430	15.600	127	2.800	3.060	0.280	2.290	5.640	1.040	3.920	1065
2	1	13.200	1.780	2.140	11.200	100	2.650	2.760	0.260	1.280	4.380	1.050	3.400	1050
3	1	13.160	2.360	2.670	18.600	101	2.800	3.240	0.300	2.810	5.680	1.030	3.170	1185
4	1	14.370	1.950	2.500	16.800	113	3.850	3.490	0.240	2.180	7.800	0.860	3.450	1480
5	1	13.240	2.590	2.870	21	118	2.800	2.690	0.390	1.820	4.320	1.040	2.930	735
6	1	14.200	1.760	2.450	15.200	112	3.270	3.390	0.340	1.970	6.750	1.050	2.850	1450
7	1	14.390	1.870	2.450	14.600	96	2.500	2.520	0.300	1.980	5.250	1.020	3.580	1290
8	1	14.060	2.150	2.610	17.600	121	2.600	2.510	0.310	1.250	5.050	1.060	3.580	1295
9	1	14.830	1.640	2.170	14	97	2.800	2.980	0.290	1.980	5.200	1.080	2.850	1045
10	1	13.860	1.350	2.270	16	98	2.980	3.150	0.220	1.850	7.220	1.010	3.550	1045
11	1	14.100	2.160	2.300	18	105	2.950	3.320	0.220	2.380	5.750	1.250	3.170	1510
12	1	14.120	1.480	2.320	16.800	95	2.200	2.430	0.260	1.570	5	1.170	2.820	1280
13	1	13.750	1.730	2.410	16	89	2.600	2.760	0.290	1.810	5.600	1.150	2.900	1320
14	1	14.750	1.730	2.390	11.400	91	3.100	3.690	0.430	2.810	5.400	1.250	2.730	1150
15	1	14.380	1.870	2.380	12	102	3.300	3.640	0.290	2.960	7.500	1.200	3	1547
16	1	13.630	1.810	2.700	17.200	112	2.850	2.910	0.300	1.460	7.300	1.280	2.880	1310
17	1	14.300	1.920	2.720	20	120	2.800	3.140	0.330	1.970	6.200	1.070	2.650	1280
18	1	13.830	1.570	2.620	20	115	2.950	3.400	0.400	1.720	6.600	1.130	2.570	1130
19	1	14.190	1.590	2.480	16.500	108	3.300	3.930	0.320	1.860	8.700	1.230	2.820	1680
20	1	13.640	3.100	2.560	15.200	116	2.700	3.030	0.170	1.660	5.100	0.960	3.360	845
21	1	14.060	1.630	2.280	16	126	3	3.170	0.240	2.100	5.650	1.090	3.710	780
22	1	12.930	3.800	2.650	18.600	102	2.410	2.410	0.250	1.980	4.500	1.030	3.520	770
23	1	13.710	1.860	2.360	16.600	101	2.610	2.880	0.270	1.690	3.800	1.110	4	1035
24	1	12.850	1.600	2.520	17.800	95	2.480	2.370	0.260	1.460	3.930	1.090	3.630	1015

Education Dataset:

Meta-Data View:

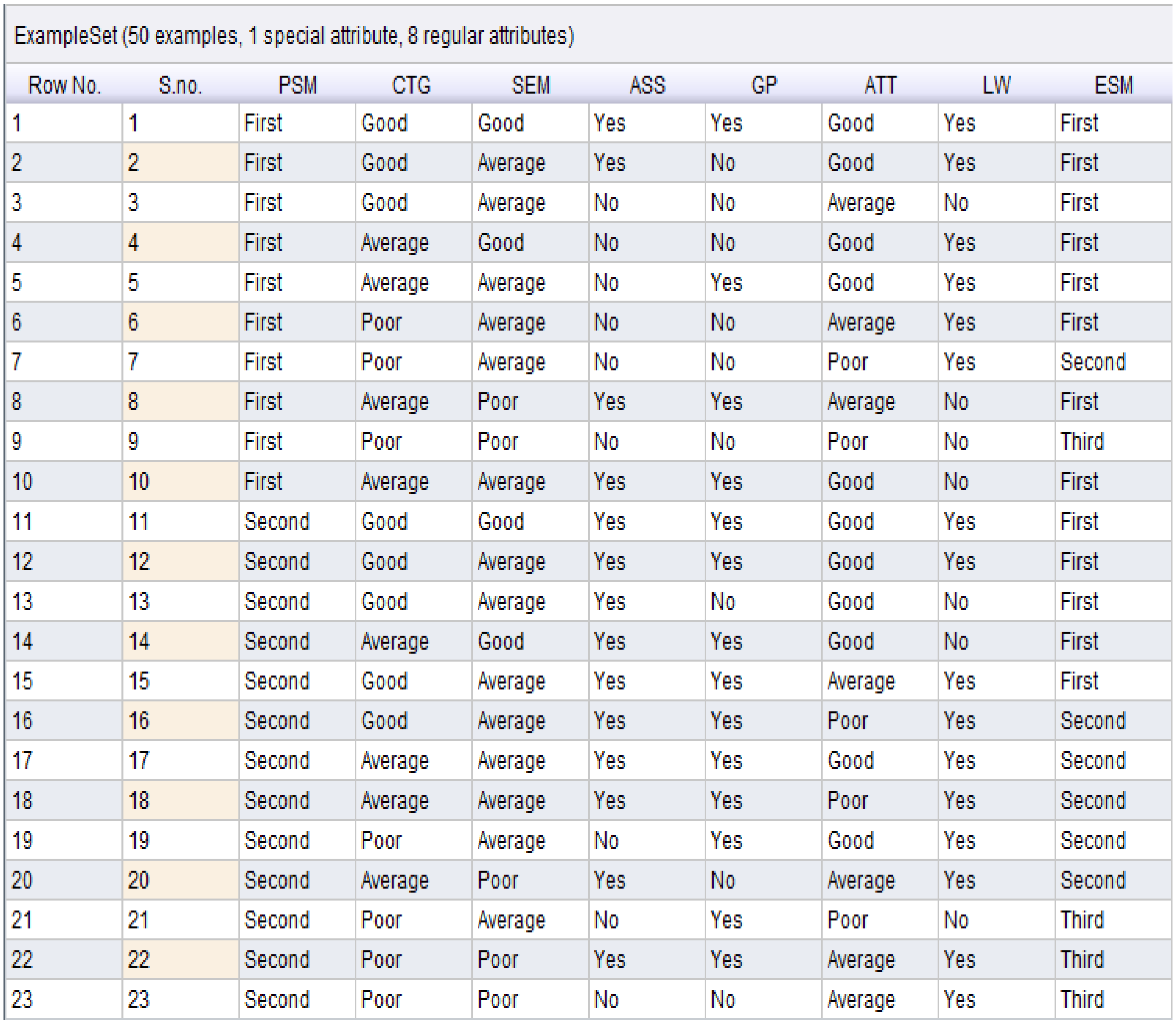


● Meta Data View ○ Data View ○ Plot View ○ Advanced Charts ○ Annotations

ExampleSet (50 examples, 1 special attribute, 8 regular attributes)

Role	Name	Type	Statistics	Range	Missings
id	S.no.	integer	avg = 25.500 +/- 14.577	[1.000 ; 50.000]	0
regular	PSM	polynominal	mode = Second (16), least = Fail (8)	First (10), Second (16), Third (16), Fail (8)	0
regular	CTG	polynominal	mode = Poor (18), least = Good (16)	Good (16), Average (16), Poor (18)	0
regular	SEM	polynominal	mode = Average (22), least = Good (13)	Good (13), Average (22), Poor (15)	0
regular	ASS	binominal	mode = Yes (27), least = No (23)	Yes (27), No (23)	0
regular	GP	binominal	mode = Yes (32), least = No (18)	Yes (32), No (18)	0
regular	ATT	polynominal	mode = Good (21), least = Poor (14)	Good (21), Average (15), Poor (14)	0
regular	LW	binominal	mode = Yes (39), least = No (11)	Yes (39), No (11)	0
regular	ESM	polynominal	mode = Second (15), least = Fail (8)	First (14), Second (15), Third (13), Fail (8)	0

Data View:



ExampleSet (50 examples, 1 special attribute, 8 regular attributes)

Row No.	S.no.	PSM	CTG	SEM	ASS	GP	ATT	LW	ESM
1	1	First	Good	Good	Yes	Yes	Good	Yes	First
2	2	First	Good	Average	Yes	No	Good	Yes	First
3	3	First	Good	Average	No	No	Average	No	First
4	4	First	Average	Good	No	No	Good	Yes	First
5	5	First	Average	Average	No	Yes	Good	Yes	First
6	6	First	Poor	Average	No	No	Average	Yes	First
7	7	First	Poor	Average	No	No	Poor	Yes	Second
8	8	First	Average	Poor	Yes	Yes	Average	No	First
9	9	First	Poor	Poor	No	No	Poor	No	Third
10	10	First	Average	Average	Yes	Yes	Good	No	First
11	11	Second	Good	Good	Yes	Yes	Good	Yes	First
12	12	Second	Good	Average	Yes	Yes	Good	Yes	First
13	13	Second	Good	Average	Yes	No	Good	No	First
14	14	Second	Average	Good	Yes	Yes	Good	No	First
15	15	Second	Good	Average	Yes	Yes	Average	Yes	First
16	16	Second	Good	Average	Yes	Yes	Poor	Yes	Second
17	17	Second	Average	Average	Yes	Yes	Good	Yes	Second
18	18	Second	Average	Average	Yes	Yes	Poor	Yes	Second
19	19	Second	Poor	Average	No	Yes	Good	Yes	Second
20	20	Second	Average	Poor	Yes	No	Average	Yes	Second
21	21	Second	Poor	Average	No	Yes	Poor	No	Third
22	22	Second	Poor	Poor	Yes	Yes	Average	Yes	Third
23	23	Second	Poor	Poor	No	No	Average	Yes	Third