ISSN:0975-9646

A Comparative Study of Classification Techniques for Fire Data Set

Rachna Raghuwanshi
M.Tech CSE
Gyan Ganga Institute of Technology & Science, Jabalpur

Abstract:Classification of data has become an important research area. The process of classifying documents into predefined categories based on their content is Text classification. It is the automated assignment of natural language texts to predefined categories. The primary requirement of text retrieval systems is text classification, which retrieve texts in response to a user query, and text understanding systems, which transform text in some way such as answering questions, producing summaries or extracting data In this paper, we study classification techniques and use of them in FIRE data set is presented. The experimental results show that the proposed system works as a successful text classifier.

Keywords: Classification, Rapid Miner tool, Decision tree, Naïve Bayes , K-NN, Fire data set, Cross Validation.

INTRODUCTION:

There are numerous text documents available in paper format such as books, news paper, magazines etc. More and more are becoming available every day. We also have FIRE data set which represents a massive amount of information that is easily accessible. In this huge collection seeking a value requires organization; data mining automated much of the work of organizing documents. The accuracy and our understanding of such systems greatly influence their usefulness. The task of data mining is to automatically classify documents into predefined classes based on their content. Many algorithms have been developed to deal with automatic text classification [10]. Various techniques of classification are ANN, Genetic Algorithms (GAs) / Evolutionary Programming (EP), Decision Tree, Naive Bayes, KNN, Clustering, Support Vector Machine, Rough set, Logistic Regression etc. But in this paper only three methods of classification
are used to calculate various results with the help of Rapid Miner tool.

Rapid Miner: Rapid Miner is the most popular open source software in the world for data mining and strongly supports text mining and other data mining techniques that are applied in combination with text mining. The power and flexibility of Rapid Miner is due to the GUI-based IDE (integrated development environment) it provides for rapid prototyping development of data mining models, as well as its strong for scripting based on XML (extensible markup language). The visual modeling in the Rapid Miner IDE is based on the defining of the data mining process in terms of operators and the flow of process through these operators. User specify the expected inputs , the delivered outputs, the mandatory and optional parameters and the core functionalities of the operators and the complete process is automatically executed by Rapid miner.

- *Decision Tree:* The decision tree is a structure that includes various nodes such as root node, branch and leaf node. Each leaf node holds the class label, each branch denotes the outcome of test and internal node denotes a test on attribute. The root node is topmost node in the tree. There are many popular decision tree algorithms CART, ID3, C4.5, CHAID, and J48. The decision tree approach is more powerful for classification problems. There are two steps in this techniques building a tree & applying the tree to the dataset.
- *Naive Bayes:* Naive Bayes classifier is based on Bayes theorem. Naïve Bayes classifier algorithm uses conditional independence properties, means it assumes that an attribute value on a given class is independent of the values of other attributes. The Bayes theorem is as follows: Let X={x_1, x_2... x_n} be a set of n attributes. In Bayesian, X is considered as evidence and H is some hypothesis means, the data of X belongs to specific class C. We have to determine P (H|X), the probability that the hypothesis H holds given evidence i.e. data sample X. According to Bayes theorem the P (H|X) is expressed as[1]
  P (H|X) = P (X| H) P (H) / P (X).
- *K-Nearest Neighbor:* The *k*-nearest neighbor's algorithm (*K*-NN) is a method for classifying objects based on closest training data in the feature space. K-NN is a type of instance-based learning. The *k*-nearest neighbor algorithm is amongst the simplest of all machine learning algorithms. But the accuracy of the *k*-NN algorithm can be severely degraded by the presence of noisy or irrelevant features, or if the feature scales are not consistent with their importance.[1]

LITERATURE REVIEW:

- ✓ Survey of Classification Techniques in Data mining By Thair Nu Phyu (2009): In this paper author explains the basic classification techniques. Various kind of classification methods such as decision tree induction, Bayesian networks, K-nearest neighbor classifier, case based reasoning, genetic algorithm and fuzzy logic. The goal of this survey is to provide comprehensive review of techniques.
- ✓ A Decision Tree Algorithm for distributed Data Mining: Towards Network Intrusion Detection By

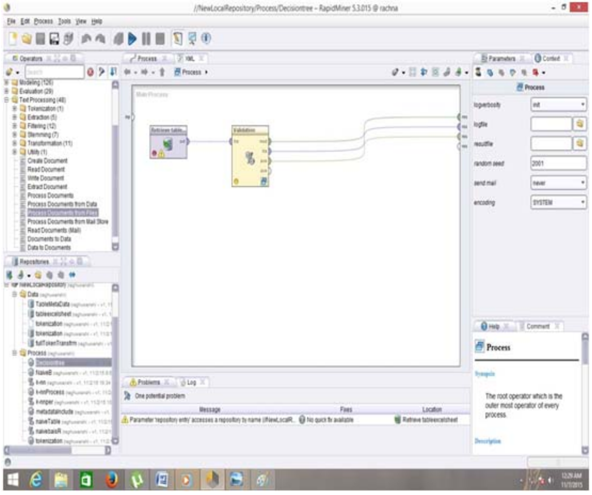
Baik, S. Bala(2004): This paper presents preliminary works on an agent-based approach for distributed learning of decision trees. The distributed decision trees approach is applied to intrusion detection domain, the interest of which is recently increasing. In this, approach a network profile is built by applying a distributed data analysis method for the collection of data from distributed hosts. Several experiments show that distributed decision tree performance is much better than the non-distributed decision tree.

- ✓ Naïve Bayes Classifiers that perform well with continuous variables by Remco R. Bouckaert(2004): There are three main methods for handling continuous variables in naïve Bayes classifiers, namely the normal method(parametric approach), the kernel method (non-parametric approach) and discretisation. In this paper comparison is performed on these three methods, which shows large mutual differences of each of the methods and no single method is better. By using continuous variable the performance of Naïve Bayes can be increases.
- ✓ Kidney Disease Prediction using Data Mining Classification Technique by Suman Bala, Krishan Kumar(2014): In this paper data mining is used for predicting disease from datasets used Analytical methodology for detecting unknown and valuable information in health data. Various techniques are compared to find best method for prediction.
- ✓ Problem of Bayesian Network Classifier By Cheng(2002): He shows the various drawbacks of Bayesian Network in his paper and also give solutions. He concluded that Bayesian network is not suitable for dataset with many features. Reason is to construct large network is not feasible in terms of time and space.
- ✓ Automatic Discovery of Similar Words, in "Survey of Text Mining: Clustering, Berry Michael W (2004),: In this paper Berry explains about the text mining and how it is similar to data mining its various aspect and operations.
- ✓ A Survey of Text Mining Techniques and Applications By Vishal Gupta & Gurpreet S. Lehal(2009): Text Mining has become an important research area. Text Mining is the discovery by computer of new, previously unknown information, by automatically extracting information from different written resources. In this paper, a Survey of Text Mining techniques and applications have been s presented.

METHODOLGY

Classification: - Classification is a very important data mining task, and the purpose of classification is to propose a classification function or classification model (called classifier).The classification model can map the data in the database to a specific class. Classification construction methods include: Decision Tree, Naive Bayes, KNN, ANN, Clustering, Support Vector Machine, Rough set, Genetic Algorithms (GAs) / Evolutionary Programming (EP), Logistic Regression etc. In this paper only three techniques are used for text preprocessing and cross validation and its sub process testing and training are also applied.

Validation- First applies validation on table which is retrieve from data set. This operator performs cross validation in order to estimate the statical performance of a learning operator. The X-validation operator is a nested operator; it has two sub processes testing and training. The X-validation operator returns performance vector. This mainly used to estimate the accuracy of particular model. Cross Validation process is shown in figure 2.



Modeling (126)
Evaluation (29)
Text Processing (48)
Tokenization (1)
Extraction (5)
Filtering (12)
Stemming (7)
Transformation (11)
Utility (1)
Create Document
Read Document
Write Document
Extract Document
Process Documents
Process Documents from Data
Process Documents from Files
Process Documents from Mail Store
Read Documents (Mail)
Documents to Data
Data to Documents

logverbosity: init
logfile:
resultfile:
random seed: 2001
send mail: never
encoding: SYSTEM

One potential problem

Message	Fixes	Location
Parameter 'repository entry' accesses a repository by name (//NewLocalR...	No quick fix available	Retrieve tableexcelsheet

Process

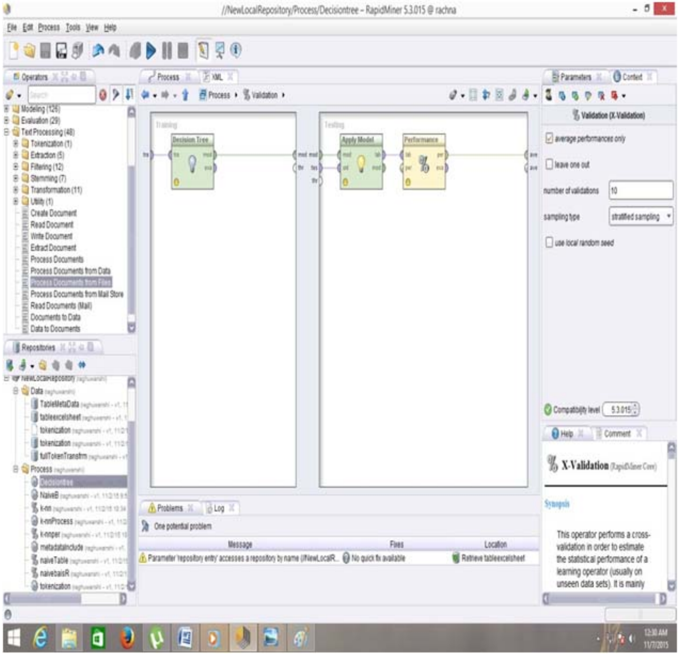
Synopsis

The root operator which is the outer most operator of every process.

Description

Fig 2- Validation operator

Testing & Training- These are the sub processes of validation operator. The training sub process is used for training a model. The trained model is then applied in the testing sub process. During the testing phase performance of the model is also measured. The testing sub process must return a Performance Vector and other results also. This is usually generated during measuring performance of the model. Two such ports are provided but more can also be used if required.



Validation (X-Validation)

☑ average performances only
☐ leave one out
number of validations: 10
sampling type: stratified sampling
☐ use local random seed
Compatibility level: 5.3.015

One potential problem

Message	Fixes	Location
Parameter 'repository entry' accesses a repository by name (//NewLocalR...	No quick fix available	Retrieve tableexcelsheet

X-Validation (RapidMiner Core)

Synopsis

This operator performs a cross-validation in order to estimate the statistical performance of a learning operator (usually on unseen data sets). It is mainly

Fig 3- Testing and Training

Decision Tree: In every recursion the DECISION TREE algorithm follows the following steps:

- An attribute A is selected to split on. Making a good choice of attributes to split on each stage is crucial to generation of a useful tree. The attribute is selected depending upon a selection criterion which can be selected by the criterion parameter.
- Examples in the FIRE Data Set are sorted into subsets, one for each value of the attribute A in case of a nominal attribute. In case of numerical attributes, subsets are formed for disjoint ranges of attribute values.
- A tree is returned with one edge or branch for each subset. Each branch has a descendant sub tree or a label value produced by applying the same algorithm recursively.

In general, the recursion stops when all the examples or instances have the same *label* value, i.e. the subset is pure. This is a generalization of the first approach; with some error threshold. However there are other halting conditions such as:

- There are less than a certain number of instances or examples in the current sub tree. This can be adjusted by using the minimal size for split parameter.
- No attribute reaches a certain threshold. This can be adjusted by using the minimum gain parameter.
- The maximal depth is reached. This can be adjusted by using the maximal depth parameter.

For splitting the attribute one of the attribute is selected for that there must be a criteria. It can have one of the following values:

- **information_gain**: All attributes entropy is calculated. The attribute with minimum entropy is selected for split. This method has a bias towards selecting attributes with a large number of values.
- **gain_ratio**: It is a variant of information gain. It adjusts the information gain for each attribute to allow the breadth and uniformity of the attribute values.
- **gini_index**: This is a measure of impurity of an Fire data set. Splitting on a chosen attribute gives a reduction in the average gain index of the resulting subsets.
- **Accuracy**: To maximizes the accuracy of the whole Tree an attribute is selected for split

Naïve Bayes: Studies comparing classification algorithms have fond a simple Baiyesian classifier known as the naïve Bayesian classifier to be comparable in performance with decision tree and selected neural network classifiers. Bayesian classifiers as comparison to other have also exhibited high accuracy and speed when applied to large databases.

Naïve Bayesian classifiers assume that the effect of an attribute value on a given class is independent of the values of the other attributes. This assumption is called class conditional independence. It is made to simplify the computations involved and, in this sense, is considered “naive”. Bayesian belief networks are graphical models subsets of attributes. can also be used for classification.

Naïve Bayes Operator: - This operator generates a Naive Bayes classification model.

K-Nearest Neighbor: The k-nearest-neighbor method was first described in early 1950s, when increased computing power become available. Ti has since widely used in the area of pattern recognition.

It is based on learning by analogy, that is by comparing a given test tuple with training tuples that are similar to it. The Training tuples are described by n attributes each tuple represents a point in an n-dimensional space. In this way, all of the training tuples are stored in an n-dimensional space. When given an unknown tuple, a k-nearest-neighbor classifier searches the pattern space for the k training tuples that are closest to the unknown tuple. These k training tuples are the k “nearest-neighbor” of the unknown tuple.

“Closeness” is defined in terms of a distance metric, such as Euclidean distance. The Euclidean distance between two pints or tuples, say $X_1=(x_{11,}\ x_{12}, \ldots\ldots, x_{1n})$ and $X_2=(x_{21,}\ x_{22}, \ldots\ldots, x_{2n})$, is

$$dist(X_1,X_2)=\sqrt{\sum}(x_{1i}-x_{2i})^2 \qquad (1)$$

In other words, for each numeric attribute, we take the difference between the corresponding values of that attribute in tuple X_1 and in tuple $X_{2,}$ square this difference, and accumulate it. The square root is taken of the total accumulated distance count. Typically, we normalize the values of each attribute before using Equation (1).

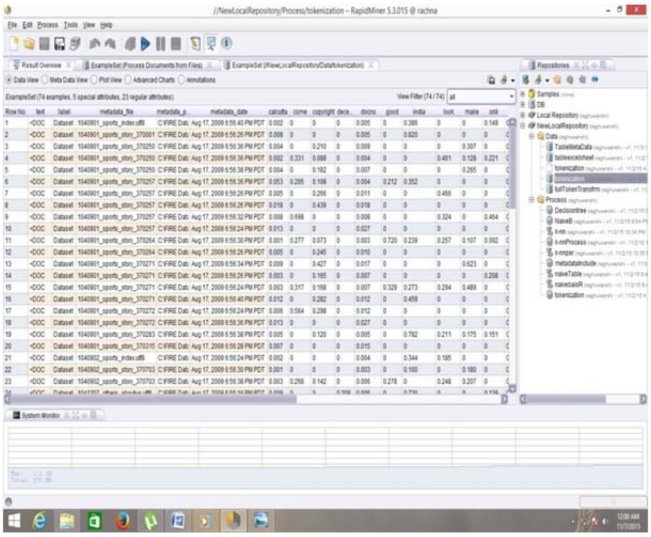
K-NN Operator- This operator generates a k-Nearest Neighbor model from the input ExampleSet. This model can be a classification or regression model depending on the input ExampleSet.

RESULT:

On the newspaper data first process document from file then text processing operators are used such as tokenization, filter tokens, steaming, transformation and stop word. After that result is shown in data view format in rapid miner. As shown in fig 4. Then convert the data into excel table, in rapid miner import excel sheet and save it into the repository. After that various method of classification are applied. First apply validation on table which is retrieve from data set, cross validation is perform to estimate the statical performance of learning. The X-validation operator is a nested operator; it has two sub processes testing and training. These are the sub processes of validation operator. The training subprocess is used for training a model. The trained model is then applied in the testing subprocess. The performance of the model is also measured during the testing phase. The testing subprocess must return a Performance Vector.

Decision Tree Result

Now decision tree method is applied on the excel sheet and it also include validation and its sub process testing and training. Results of decision tree are shown in the below figures 5, 6, 7. It is the example set of decision tree. In rapid miner various types of view can see of data such as data view, Meta data view, plot view, advanced charts, annotations. In this figure data view of the data set is shown Example set consist (24 examples, special attribute and 5 regular attribute). First row of the table shows row no. second real, third name of attribute, fourth type of attribute etc. Figure 4 shows decision tree graph view in this graph data set text type is shown there are two types of words in data set, Real and text. Figure 5 shows the performance vector of decision tree, accuracy rate of decision tree is 88.33%

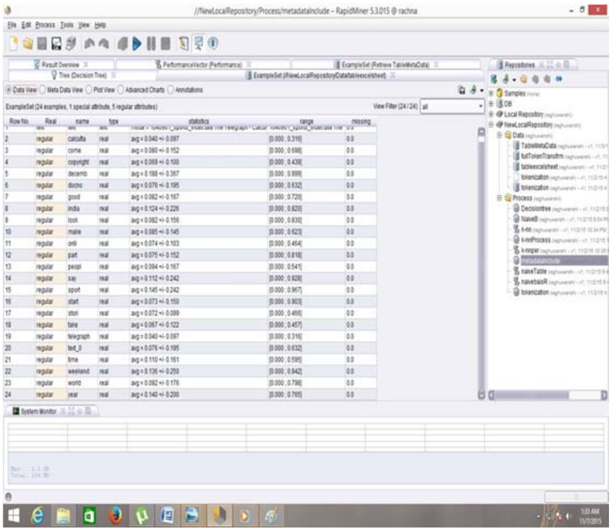

//NewLocalRepository/Process/tokenization – RapidMiner 5.3.015 @ rachna
File Edit Process Tools View Help
Result Overview
ExampleSet (Process Documents from Files)
ExampleSet (//NewLocalRepository/Data/tokenization)
○ Data View ○ Meta Data View ○ Plot View ○ Advanced Charts ○ Annotations
View Filter (74 / 74): all
Repositories
System Monitor

ExampleSet (74 examples, 5 special attributes, 23 regular attributes)

Row No.	text	label	metadata_file	metadata_p...	metadata_date	calcutta	come	copyright	decx...	docno	good	india	look	make	onli
1	<DOC	Dataset	1040901_sports_index.utf8	C:\FIRE Dat	Aug 17, 2009 6:56:40 PM PDT	0.002	0	0	0	0.005	0	0.386	0	0	0.149
2	<DOC	Dataset	1040901_sports_story_370001	C:\FIRE Dat	Aug 17, 2009 6:56:26 PM PDT	0.008	0	0	0	0.005	0	0.820	0	0	0
3	<DOC	Dataset	1040901_sports_story_370250	C:\FIRE Dat	Aug 17, 2009 6:56:36 PM PDT	0.004	0	0.210	0	0.009	0	0	0	0.307	0
4	<DOC	Dataset	1040901_sports_story_370250	C:\FIRE Dat	Aug 17, 2009 6:56:36 PM PDT	0.002	0.331	0.088	0	0.004	0	0	0.461	0.128	0.221
5	<DOC	Dataset	1040901_sports_story_370250	C:\FIRE Dat	Aug 17, 2009 6:56:38 PM PDT	0.004	0	0.182	0	0.007	0	0	0	0.265	0
6	<DOC	Dataset	1040901_sports_story_370257	C:\FIRE Dat	Aug 17, 2009 6:56:36 PM PDT	0.053	0.295	0.108	0	0.004	0.212	0.352	0	0	0
7	<DOC	Dataset	1040901_sports_story_370257	C:\FIRE Dat	Aug 17, 2009 6:56:26 PM PDT	0.005	0	0.266	0	0.011	0	0	0.466	0	0
8	<DOC	Dataset	1040901_sports_story_370257	C:\FIRE Dat	Aug 17, 2009 6:56:26 PM PDT	0.018	0	0.439	0	0.018	0	0	0	0	0
9	<DOC	Dataset	1040901_sports_story_370257	C:\FIRE Dat	Aug 17, 2009 6:56:32 PM PDT	0.008	0.698	0	0	0.008	0	0	0.324	0	0.464
10	<DOC	Dataset	1040901_sports_story_370257	C:\FIRE Dat	Aug 17, 2009 6:56:24 PM PDT	0.013	0	0	0	0.027	0	0	0	0	0
11	<DOC	Dataset	1040901_sports_story_370264	C:\FIRE Dat	Aug 17, 2009 6:56:30 PM PDT	0.001	0.277	0.073	0	0.003	0.720	0.239	0.257	0.107	0.092
12	<DOC	Dataset	1040901_sports_story_370264	C:\FIRE Dat	Aug 17, 2009 6:56:26 PM PDT	0.005	0	0.245	0	0.010	0	0	0	0	0
13	<DOC	Dataset	1040901_sports_story_370271	C:\FIRE Dat	Aug 17, 2009 6:56:34 PM PDT	0.009	0	0.427	0	0.017	0	0	0	0.623	0
14	<DOC	Dataset	1040901_sports_story_370271	C:\FIRE Dat	Aug 17, 2009 6:56:26 PM PDT	0.003	0	0.165	0	0.007	0	0	0	0	0.298
15	<DOC	Dataset	1040901_sports_story_370271	C:\FIRE Dat	Aug 17, 2009 6:56:24 PM PDT	0.003	0.317	0.168	0	0.007	0.329	0.273	0.294	0.489	0
16	<DOC	Dataset	1040901_sports_story_370271	C:\FIRE Dat	Aug 17, 2009 6:56:40 PM PDT	0.012	0	0.282	0	0.012	0	0.459	0	0	0
17	<DOC	Dataset	1040901_sports_story_370272	C:\FIRE Dat	Aug 17, 2009 6:56:24 PM PDT	0.006	0.564	0.298	0	0.012	0	0	0	0	0
18	<DOC	Dataset	1040901_sports_story_370272	C:\FIRE Dat	Aug 17, 2009 6:56:36 PM PDT	0.013	0	0	0	0.027	0	0	0	0	0
19	<DOC	Dataset	1040901_sports_story_370283	C:\FIRE Dat	Aug 17, 2009 6:56:38 PM PDT	0.005	0	0.120	0	0.005	0	0.782	0.211	0.175	0.151
20	<DOC	Dataset	1040901_sports_story_370315	C:\FIRE Dat	Aug 17, 2009 6:56:28 PM PDT	0.007	0	0	0	0.015	0	0	0	0	0
21	<DOC	Dataset	1040902_sports_index.utf8	C:\FIRE Dat	Aug 17, 2009 6:56:24 PM PDT	0.002	0	0	0	0.004	0	0.344	0.185	0	0
22	<DOC	Dataset	1040902_sports_story_370703	C:\FIRE Dat	Aug 17, 2009 6:56:30 PM PDT	0.001	0	0	0	0.003	0	0.100	0	0.180	0
23	<DOC	Dataset	1040902_sports_story_370703	C:\FIRE Dat	Aug 17, 2009 6:56:38 PM PDT	0.003	0.258	0.142	0	0.006	0.278	0	0.248	0.207	0

Fig 4: Result of Tokenization1

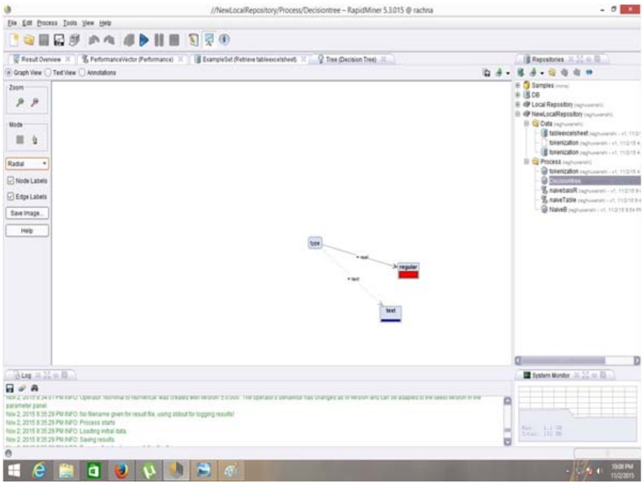
Results of Decision Tree


//NewLocalRepository/Process/metadataInclude – RapidMiner 5.3.015 @ rachna
File Edit Process Tools View Help
Result Overview
PerformanceVector (Performance)
ExampleSet (Retrieve TableMetaData)
Tree (Decision Tree)
ExampleSet (//NewLocalRepository/Data/tableexcelsheet)
○ Data View ○ Meta Data View ○ Plot View ○ Advanced Charts ○ Annotations
View Filter (24 / 24): all
Repositories
System Monitor

ExampleSet (24 examples, 1 special attribute, 5 regular attributes)

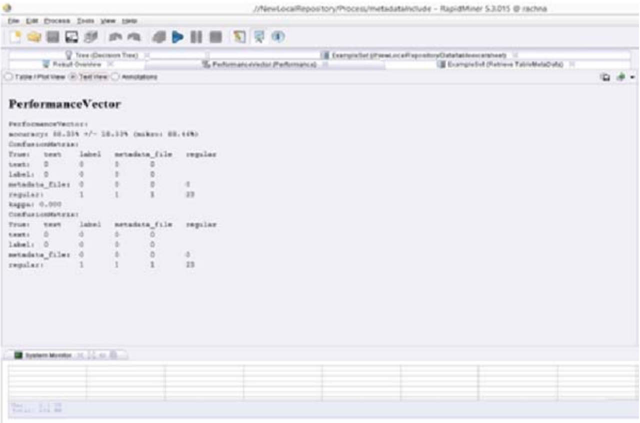
Row No.	Real	name	type	statistics	range	missing
2	regular	calcutta	real	avg = 0.040 +/- 0.097	[0.000 ; 0.316]	0.0
3	regular	come	real	avg = 0.080 +/- 0.152	[0.000 ; 0.698]	0.0
4	regular	copyright	real	avg = 0.069 +/- 0.100	[0.000 ; 0.439]	0.0
5	regular	decemb	real	avg = 0.188 +/- 0.367	[0.000 ; 0.999]	0.0
6	regular	docno	real	avg = 0.076 +/- 0.195	[0.000 ; 0.632]	0.0
7	regular	good	real	avg = 0.082 +/- 0.167	[0.000 ; 0.720]	0.0
8	regular	india	real	avg = 0.124 +/- 0.226	[0.000 ; 0.820]	0.0
9	regular	look	real	avg = 0.082 +/- 0.156	[0.000 ; 0.830]	0.0
10	regular	make	real	avg = 0.085 +/- 0.145	[0.000 ; 0.623]	0.0
11	regular	onli	real	avg = 0.074 +/- 0.103	[0.000 ; 0.464]	0.0
12	regular	part	real	avg = 0.075 +/- 0.152	[0.000 ; 0.818]	0.0
13	regular	peopl	real	avg = 0.094 +/- 0.167	[0.000 ; 0.541]	0.0
14	regular	say	real	avg = 0.112 +/- 0.242	[0.000 ; 0.928]	0.0
15	regular	sport	real	avg = 0.145 +/- 0.242	[0.000 ; 0.967]	0.0
16	regular	start	real	avg = 0.073 +/- 0.150	[0.000 ; 0.903]	0.0
17	regular	stori	real	avg = 0.072 +/- 0.099	[0.000 ; 0.466]	0.0
18	regular	take	real	avg = 0.067 +/- 0.122	[0.000 ; 0.457]	0.0
19	regular	telegraph	real	avg = 0.040 +/- 0.097	[0.000 ; 0.316]	0.0
20	regular	text_0	real	avg = 0.076 +/- 0.195	[0.000 ; 0.632]	0.0
21	regular	time	real	avg = 0.110 +/- 0.161	[0.000 ; 0.595]	0.0
22	regular	weekend	real	avg = 0.136 +/- 0.250	[0.000 ; 0.942]	0.0
23	regular	world	real	avg = 0.092 +/- 0.176	[0.000 ; 0.798]	0.0
24	regular	year	real	avg = 0.140 +/- 0.200	[0.000 ; 0.765]	0.0

Fig 5 :- Decision tree Example Set


//NewLocalRepository/Process/Decisiontree – RapidMiner 5.3.015 @ rachna
File Edit Process Tools View Help
Result Overview
PerformanceVector (Performance)
ExampleSet (Retrieve tableexcelsheet)
Tree (Decision Tree)
○ Graph View ○ Text View ○ Annotations
Zoom
Mode
Radial
☑ Node Labels
☑ Edge Labels
Save Image...
Help
type
= real
regular
= text
text
Log
System Monitor

```
parameter panel.
Nov 2, 2015 8:35:29 PM INFO: No filename given for result file, using stdout for logging results!
Nov 2, 2015 8:35:29 PM INFO: Process starts
Nov 2, 2015 8:35:29 PM INFO: Loading initial data.
Nov 2, 2015 8:35:29 PM INFO: Saving results.
```

Fig 6 -Decision Tree Graph view


//NewLocalRepository/Process/metadataInclude – RapidMiner 5.3.015 @ rachna
File Edit Process Tools View Help
Tree (Decision Tree)
ExampleSet (//NewLocalRepository/Data/tableexcelsheet)
Result Overview
PerformanceVector (Performance)
ExampleSet (Retrieve TableMetaData)
○ Table / Plot View ● Text View ○ Annotations
System Monitor

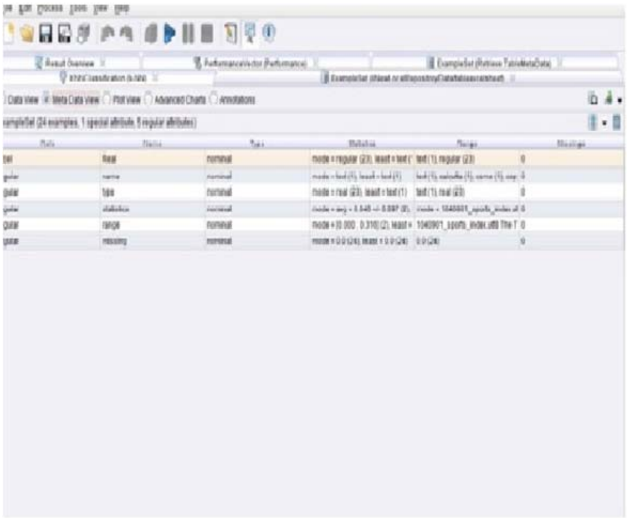
PerformanceVector

```
PerformanceVector:
accuracy: 88.33% +/- 18.33% (mikro: 88.46%)
ConfusionMatrix:
True:   text    label   metadata_file   regular
text:   0       0       0       0
label:  0       0       0       0
metadata_file:  0       0       0       0
regular:        1       1       1       23
kappa: 0.000
ConfusionMatrix:
True:   text    label   metadata_file   regular
text:   0       0       0       0
label:  0       0       0       0
metadata_file:  0       0       0       0
regular:        1       1       1       23
```

Fig 7- Decision Tree Performance Vector Text View

K-Nearest-Neighbor Classifier Results:

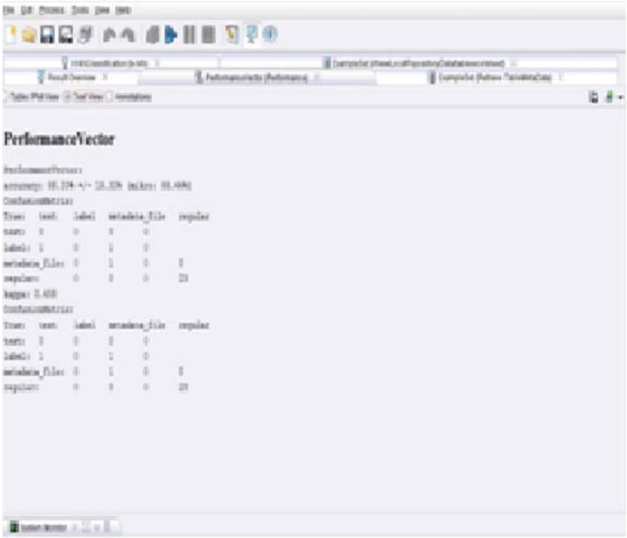
K-Nearest-Neighbor method is applied on the excel sheet and it also include validation and its sub process testing and training. Results of K-Nearest-Neighbor are shown in the below figures 8, 9.


File Edit Process Tools View Help
Result Overview
PerformanceVector (Performance)
ExampleSet (Retrieve TableMetaData)
kNNClassification (k-NN)
ExampleSet (//NewLocalRepository/Data/tableexcelsheet)
○ Data View ● Meta Data View ○ Plot View ○ Advanced Charts ○ Annotations

xampleSet (24 examples, 1 special attribute, 5 regular attributes)

Role	Name	Type	Statistics	Range	Missings
bel	Real	nominal	mode = regular (23), least = text ('	text (1), regular (23)	0
gular	name	nominal	mode = text (1), least = text (1)	text (1), calcutta (1), come (1), cop	0
gular	type	nominal	mode = real (23), least = text (1)	text (1), real (23)	0
gular	statistics	nominal	mode = avg = 0.040 +/- 0.097 (2),	mode = 1040901_sports_index.ut	0
gular	range	nominal	mode = [0.000 ; 0.316] (2), least =	1040901_sports_index.utf8 The T	0
gular	missing	nominal	mode = 0.0 (24), least = 0.0 (24)	0.0 (24)	0

Fig 8-K-NN Meta Data View


File Edit Process Tools View Help
kNNClassification (k-NN)
ExampleSet (//NewLocalRepository/Data/tableexcelsheet)
Result Overview
PerformanceVector (Performance)
ExampleSet (Retrieve TableMetaData)
○ Table / Plot View ● Text View ○ Annotations
System Monitor

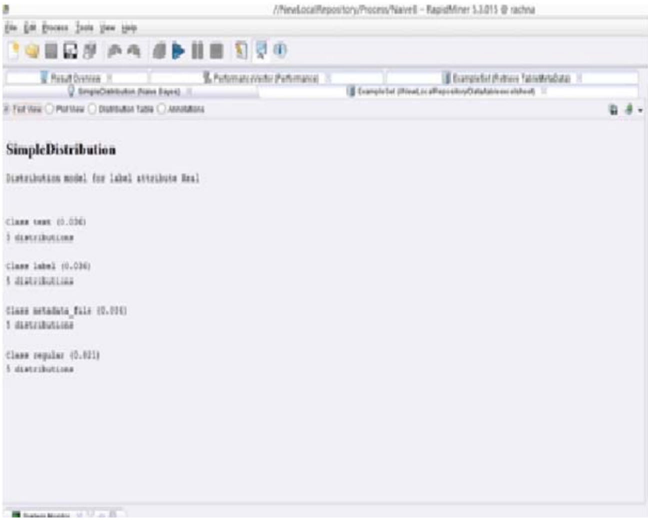
PerformanceVector

```
PerformanceVector:
accuracy: 88.33% +/- 18.33% (mikro: 88.46%)
ConfusionMatrix:
True:   text    label   metadata_file   regular
text:   0       0       0       0
label:  1       0       1       0
metadata_file:  0       1       0       0
regular:        0       0       0       23
kappa: [illegible]
ConfusionMatrix:
True:   text    label   metadata_file   regular
text:   0       0       0       0
label:  1       0       1       0
metadata_file:  0       1       0       0
regular:        0       0       0       23
```

Fig 9- K-NN Performance Vector (Text View)

Results and output of Naïve Bayes Classification-

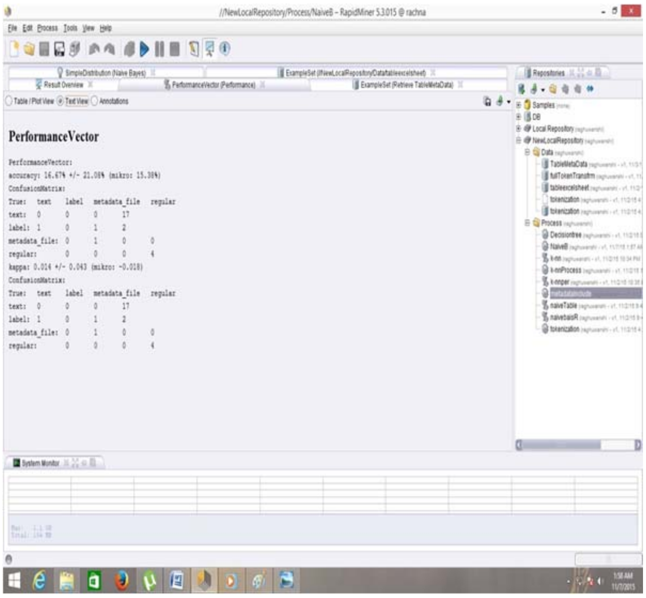
Naïve Bayes method is applied on the excel sheet and it also include validation and its sub process testing and training. Results of Naïve Bayes are shown in the below figures. Figure 10 shows simple distribution of real data and figure Performance vector with accuracy rate 16.67%.


//NewLocalRepository/Process/NaiveB – RapidMiner 5.3.015 @ rachna
File Edit Process Tools View Help
Result Overview
PerformanceVector (Performance)
ExampleSet (Retrieve TableMetaData)
SimpleDistribution (Naive Bayes)
ExampleSet (//NewLocalRepository/Data/tableexcelsheet)
Text View ○ Plot View ○ Distribution Table ○ Annotations
System Monitor

SimpleDistribution

```
Distribution model for label attribute Real

Class text (0.036)
3 distributions

Class label (0.036)
5 distributions

Class metadata_file (0.036)
5 distributions

Class regular (0.821)
5 distributions
```

Fig 10 - Naïve Bayes Simple Distribution


//NewLocalRepository/Process/NaiveB – RapidMiner 5.3.015 @ rachna
File Edit Process Tools View Help
SimpleDistribution (Naive Bayes)
ExampleSet (//NewLocalRepository/Data/tableexcelsheet)
Result Overview
PerformanceVector (Performance)
ExampleSet (Retrieve TableMetaData)
○ Table / Plot View ● Text View ○ Annotations
Repositories
Samples
DB
Local Repository
NewLocalRepository
Data
TableMetaData
fullTokenTransfrm
tableexcelsheet
tokenization
tokenization
Process
Decisiontree
NaiveB
k-nn
k-nnProcess
k-sniper
metadataincbude
naiveTable
naivebaisR
tokenization
System Monitor

PerformanceVector

```
PerformanceVector:
accuracy: 16.67% +/- 21.08% (mikro: 15.38%)
ConfusionMatrix:
True:  text   label  metadata_file  regular
text:  0      0      0      17
label: 1      0      1      2
metadata_file: 0      1      0      0
regular:       0      0      0      4
kappa: 0.014 +/- 0.043 (mikro: -0.018)
ConfusionMatrix:
True:  text   label  metadata_file  regular
text:  0      0      0      17
label: 1      0      1      2
metadata_file: 0      1      0      0
regular:       0      0      0      4
```

Fig 11- Naïve Bayes Performance Vector Text view

CONCLUSION:

The overall objective of our is to study the various data mining classification techniques available to calculate the number of word most commonly used in the FIRE dataset and to compare them to find the best method which shows exact accuracy.

- We analyzed that the most commonly used DM technique such as Decision Trees, ANN and Naïve Bayes, resulting as well-performing on the FIRE data sets to find the number of word commonly used and also avoid small ones. Which single DM technique is best is very difficult to say. Depending on concrete situations, sometime some techniques perform better than others.
- We also analyzed that there is no single classifier which produce best result for FIRE dataset. The classifier performance is evaluated using testing data set. But there is also some problem with testing data set. Some time it is easy and some time it becomes complex to classify the testing data set. To avoid these problems we used cross validation method so that every record of FIRE data set is used for both training and testing.

REFERENCES:

1) Suman Bala, Krishan Kumar A Literature Review on “Kidney Disease Prediction using Data Mining Classification Technique” IJCSMC, Vol. 3, Issue. 7, July 2014

2) Anurag Upadhayay, Suneet Shukla, Sudsanshu Kumar “ Empirical Comparison by data mining Classification algorithms (C 4.5 & C 5.0) for thyroid cancer data set.” International Journal of Computer Science & Communication Networks,Vol 3(1)

3) H. C. Koh and G. Tan, “*Data Mining Application in Healthcare*”, Journal of Healthcare Information Management, vol. 19, no. 2, 2005.

4) Brijesh Kumar Bhardwaj, Saurabh Pal “Data Mining: A prediction for performance improvement using classification” (IJCSIS) International Journal of Computer Science and Information Security, Vol. 9, No. 4, April 2011

5) AI-Radaideh,Q. A., AI-Shawakfa, E.M., and AI-Najjar, M. I.,“Mining Student Data using Decision Trees”, International Arab Conference on Information Technology(ACIT'2006),Yarmouk University, Jordan, 2006.

6) Alaa el-Halees, “Mining Students Data to Analyze e-Learning Behavior: A Case Study”, 2009.

7) Vishal Gupta, Gurpreet S. Lehal Professor & Head, Department of Computer Science, Punjabi University Patiala, India ” A Survey of Text Mining Techniques and Applications” JOURNAL OF EMERGING TECHNOLOGIES IN WEB INTELLIGENCE, VOL 1, AUGUST 2009

8) [Berry Michael W., (2004), “Automatic Discovery of Similar Words”, in “Survey of Text Mining: Clustering, Classification and Retrieval”, Springer Verlag, New York, LLC, 24-43.

9) Navathe, Shamkant B., and Elmasri Ramez, (2000), “Data Warehousing And Data Mining”, in “Fundamentals of Database Systems”, Pearson Education pvt Inc, Singapore, 841-872.

10) S. M. Kamruzzaman, Farhana Haider, Ahmed Ryadh Hasan “Text Classification Using Data Mining” ICTM 2005

11) K.Sudhakar & Dr. M. Manimekalai,” *Study of Heart Disease Prediction using Data mining*”, IJARCSSE, Volume 4, Issue 1, January 2014.

12) Frawley and Piatetsky-Shapiro, 1996. *Knowledge Discovery in Databases*: An Overview. The AAAI/MIT Press, Menlo Park, C.A.

13) DSVGK Kaladhar, Krishna Apparao Rayavarapu* and Varahalarao Vadlapudi,”*Statistical and Data Mining Aspects on Kidney Stones: A Systematic Review and Meta-analysis*”, Open Access Scientific Reports, Volume 1 • Issue 12 • 2012

14) AI-Radaideh,Q. A., AI-Shawakfa, E.M., and AI-Najjar, M. I.,“Mining Student Data using Decision Trees”, International

15) Arab Conference on Information Technology(ACIT'2006),Yarmouk University, Jordan, 2006.

16) Alaa el-Halees, “Mining Students Data to Analyze e-Learning Behavior: A Case Study”, 2009.

17) Agarwal R., Mannila H., Srikant R., Toivonan H., Verkamo, “A Fast Discovery of Association Rules,” *Advances in Knowledge Discovery and Data Mining*, 1996.