Performance Enhancement Using Combinatorial Approach of Classification and Clustering In Machine Learning

[1]Neetu Sharma, [2]Dr. S. Niranjan

[1]Ph.D Scholar, Computer Science & Engg.,
Mewar University, Chittorgarh,
Rajasthan, India

[2]Principal,PDM college of Engg.,
Bahadurgarh, Haryana, India

ABSTRACT

Clustering and classification are two important techniques of data mining. Classification is a supervised learning problem of assigning an object to one of several pre-defined categories based upon the attributes of the object. While clustering is an unsupervised learning problem that group objects based upon distance or similarity. Each group is known as a cluster. In this paper we make use of a database 'Weather.arff ' containing 7 attributes and 14 instances to perform an integration of clustering and classification techniques of data mining. We compared results of simple classification technique (using Naive Bayes classifier) with the results of integration of clustering and classification technique, based upon various parameters using WEKA (Waikato Environment for Knowledge Analysis), a Data Mining tool. The results of the experiment show that integration of clustering and classification gives promising results with utmost accuracy rate and robustness even when the data set is containing missing values.

Keywords: Data Mining; Naïve Bayes; KMEANS; WEKA; Weather.arff.

1. INTRODUCTION

1.1. Data Mining:

Data mining is the discovery and modeling of hidden patterns in large amounts of data. Data mining is extensively used in the industry, where organizations have huge amount of data in their database. Data mining is the science of finding useful information from this huge voluminous data, which can benefit the business of the organization tremendously. Examples include: "Which customers will order how much quantity of a particular product in the coming week?" or "Will this customer cancel our service if we introduce higher fees?" Text mining is a subset of data mining, which applies the same concept, but is aimed at finding information from text rather than numeric data.

1.2. Machine Learning

Machine Learning is the ability of a machine to perform better at a given task, using its previous experience. Various algorithms like decision trees, Bayesian learning, artificial neural networks and instance-based learning algorithms are used widely in machine learning systems. A learner can take advantage of examples (data) to capture characteristics of interest of their unknown underlying probability distribution. Data can be seen as examples that illustrate relations between observed variables. A major focus of machine learning research is to automatically learn to recognize complex patterns and make intelligent decisions based on data; the difficulty lies in the fact that the set of all possible behaviors given all possible inputs is too large to be covered by the set of observed examples (training data). Hence the learner must generalize from the given examples, so as to be able to produce a useful output in new cases. The application of machine learning techniques to natural language processing (NLP) has increased tremendously in recent years. Examples are hand-writing recognition and speech recognition.

Definition[10]: A computer program is said to learn from experience E with respect to some class of tasks T and performance P, if its performance at tasks in T, as measured by P, improves with experience E. For example, a computer program built to learn to play chess, will have task T defined as playing chess, performance P defined as number of wins while playing chess matches against opponents and experience E defined as knowledge and tactics learnt by playing the game against self. A machine learning system consists of three main parts:

1. The training experience: This is the set of data, which is fed into the machine learning system and the learning algorithm processes through this data to built a knowledge base.
2. The learning algorithm: This forms the core of the machine learning system. Various types of algorithms have been invented for the varying types of learning tasks.
3. The test data: Test data is used to determine the performance of the learning system.
Each of the three parts mentioned above play an important role in determining the success of the machine learning system. Representation of training data and test data are decided depending on the type of learning task at hand.

1.3 Naive Bayes Machine Learning Algorithm

Naive Bayes text classification is a supervised and probabilistic learning method. It calculates the probability of a document d being in class c by the following formula. $P(t_k|c)$ is the conditional probability of term t_k occurring in a document of class c. P(c) is the prior probability of a document occurring in class c.

$$P(c|d) \propto P(c) \prod P(t_k|c)$$

The goal of classification is to find the best class for the document. The best class in naive bayes classification is the most likely or maximum a posteriori(MAP) class C_{map}

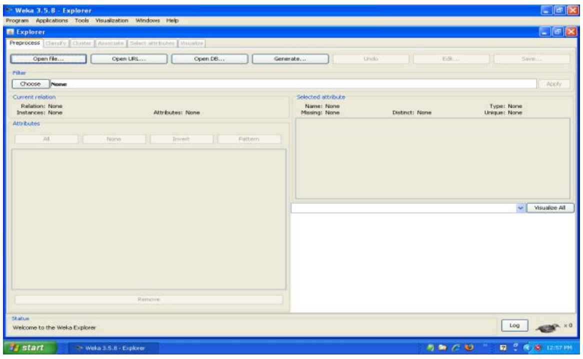
$$C_{map} = \text{argmax}_{c \in C} \; P(c|d) = \text{argmax}_{c \in C} \; P(t_k|c)$$

1.4. K-Means clusterer

Simple K-Means is one of the simplest clustering algorithms K-Means algorithm is a classical clustering method that group large datasets in to clusters. The procedure follows a simple way to classify a given data set through a certain number of clusters. It select k points as initial centroids and find K clusters by assigning data instances to nearest centroids. Distance measure used to find centroids is Euclidean distance.

1.5. WEKA

Weka stands for Waikato Environment for Knowledge Analysis.[22] It is software developed by the University of Waikato in New Zealand. Weka provides implementations of state-of –the-art are machine learning algorithms. It is freely available on the World Wide Web. The software is developed in Java, thus enabling porting and compatibility with various platforms. Weka provides a command line interface as well as a GUI interface. Its functionality can be divided into two main categories, Explorer and Experimenter. Using Explorer, one can preprocess a dataset, feed it into a learning scheme and analyze the resulting classifier and its performance. A learning method is called a classifier. Using Experimenter, one can apply several learners and compare their performance to choose a learner for prediction. Implementation of learning schemes is the most significant and valuable feature of Weka. Tools for preprocessing the data, called filters are also a useful feature of Weka. The main focus of Weka is on classifier and filter algorithms. It also has implementations of algorithms for learning association rules and for clustering data for which no class value is specified.


Weka 3.5.8 - Explorer
Program Applications Tools Visualization Windows Help
Explorer
Preprocess Classify Cluster Associate Select attributes Visualize
Open file... Open URL... Open DB... Generate... Undo Edit... Save...
Filter
Choose
Apply
Current relation
Attributes
All None Invert Pattern
Remove
Selected attribute
Visualize All
Status
Log
x 0
start
Weka 3.5.8 - Explorer
12:57 PM

Filter: None

Relation: None
Instances: None Attributes: None

Name: None Type: None
Missing: None Distinct: None Unique: None

Welcome to the Weka Explorer

Figure 1. WEKA options interface

1.6. Classification

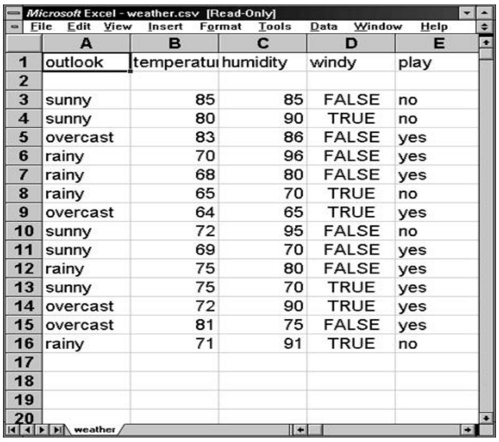
Implementations of almost all main-stream classification algorithms are included. Bayesian methods include naive Bayes, complement naive Bayes, multinomial naive Bayes, Bayesian networks,
and AODE. There are many decision tree learners: decision stumps, ID3, a C4.5 clone called “J48,” trees generated by reduced error pruning, alternating decision trees, and random trees and forests thereof.
Rule learners include OneR, an implementation of Ripper called “JRip,” decision tables, single conjunctive rules, and Prism. There are several separating hyperplane approaches like support vector machines with a variety of kernels,

logistic regression, voted perceptrons, Winnow and a multi-layer perceptron. There are many lazy learning methods like IB1, IBk, lazy Bayesian rules, KStar, and locally-weighted Learning. As well as the basic classification learning methods, so-called “metalearning” schemes enable users to combine instances of one or more of the basic algorithms in various ways: bagging, boosting(including the variants AdaboostM1 and LogitBoost), and stacking. A method called “FilteredClassifier” allows a filter to be paired up with a classifier. Classification can be made cost-sensitive, or multi-class, or ordinal-class.

Parameter values can be selected using cross-validation.

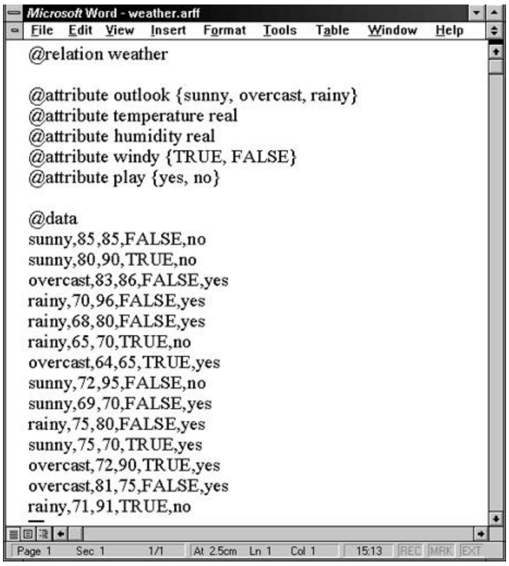
1.7. The ARFF format

Weka requires that the dataset, to which the machine learning algorithms are going to be applied, should be in ARFF format. Before one can apply any algorithm to the data, the data needs to be converted into the ARFF format. Figure 2 shows the data in csv format which is converted into ARFF format as shown in figure 3.



	A	B	C	D	E
1	outlook	temperatu	humidity	windy	play
2					
3	sunny	85	85	FALSE	no
4	sunny	80	90	TRUE	no
5	overcast	83	86	FALSE	yes
6	rainy	70	96	FALSE	yes
7	rainy	68	80	FALSE	yes
8	rainy	65	70	TRUE	no
9	overcast	64	65	TRUE	yes
10	sunny	72	95	FALSE	no
11	sunny	69	70	FALSE	yes
12	rainy	75	80	FALSE	yes
13	sunny	75	70	TRUE	yes
14	overcast	72	90	TRUE	yes
15	overcast	81	75	FALSE	yes
16	rainy	71	91	TRUE	no
17					
18					
19					
20					

Figure 2. Weather data in csv file format



```
@relation weather

@attribute outlook {sunny, overcast, rainy}
@attribute temperature real
@attribute humidity real
@attribute windy {TRUE, FALSE}
@attribute play {yes, no}

@data
sunny,85,85,FALSE,no
sunny,80,90,TRUE,no
overcast,83,86,FALSE,yes
rainy,70,96,FALSE,yes
rainy,68,80,FALSE,yes
rainy,65,70,TRUE,no
overcast,64,65,TRUE,yes
sunny,72,95,FALSE,no
sunny,69,70,FALSE,yes
rainy,75,80,FALSE,yes
sunny,75,70,TRUE,yes
overcast,72,90,TRUE,yes
overcast,81,75,FALSE,yes
rainy,71,91,TRUE,no
```

Figure 3. Weather data in aff file format

1.8. Clustering

The Cluster tab opens the process that is used to identify commonalties or clusters of occurrences within the data set and produce information for the user to analyze. There are a few options within the cluster window that are similar to those described in the Classify tab. These options are: use training set, supplied test set and percentage split. At present, only a few standard clustering algorithms are included: KMeans, EM for naive Bayes models, farthest-first clustering, and Cobweb. This list is likely to grow in the near future.

1.9. Filters

Processes that transform instances and sets of instances are called “filters,” and they are classified according to whether they make sense only in a prediction context (called “supervised”) or in any context (called “unsupervised”). We further split them into "attribute filters," which work on one or more attributes of an instance, and "instance filters," which work on entire instances. Unsupervised attribute filters include adding a new attribute, adding a cluster indicator, adding noise, copying an attribute, discretizing a numeric attribute, normalizing or standardizing a numeric attribute, making indicators, merging attribute values, transforming nominal to binary values, obfuscating values, swapping values, removing attributes, replacing missing values, turning string attributes into nominal ones or word vectors, computing random projections, and processing time series data. Unsupervised instance filters transform sparse instances into non-sparse instances and vice versa, randomize and resample sets of instances, and remove instances according to certain criteria. Supervised attribute filters include support for attribute selection, discretization, nominal to binary transformation, and re-ordering the class values. Finally, supervised instance filters resample and subsample sets of instances to generate different class distributions—stratified, uniform, and arbitrary user-specified spreads.

2. PROBLEM STATEMENT

The problem in particular is a comparative study of classification technique algorithm Naive Bayes with an integration of Simple KMeans clusterer and Naive Bayes classifier on various parameters using weather..arff data file containing 7 attributes and 14 instances.

3. PROPOSED WORK

Classification is the process of finding a set of models that describe and distinguish data classes and concepts, for the purpose of being able to use the model to predict the class whose label is unknown. Clustering is different from classification as it builds the classes (which are not known in advance) based upon similarity between object features.. Integration of clustering and classification technique is useful even when the dataset contains missing values. In this experiment, object corresponds to weather.arff file from dataset and has two object class labels corresponds to data file. Apply classification technique using Naïve Bayes classifier in WEKA tool. Classification is a two step process, first, it build classification model using training data. Every object of the dataset must be pre-classified i.e. its class label must be known, second the model generated in the preceding step is tested by assigning class labels to data objects in a test dataset The test data may be different from the training data. Every element of the test data is also classified in advance. The accuracy of the classification model is determined by comparing true class labels in the testing set with those assigned by the model. Apply clustering technique on the original data file using WEKA tool and now we are come up with a number of clusters. It also adds an attribute cluster to the data set. Apply classification technique on the clustering result data set. Then compare the results of simple classification and an integration of clustering and classification. In this paper, we identified the finest classification rules through experimental study for the task of using WEKA data mining tool.

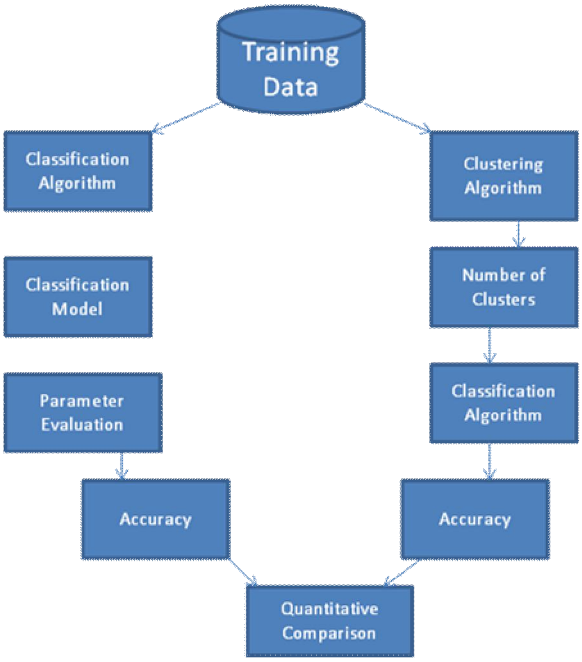


Figure 4. Proposed Model

4. IMPLEMENTATION

We have chosen the data from weather record . For each attribute, we have created a line of data specifying the values for the chosen features and also the correct data (all separated by commas). One possibility is to enter this data into a spreadsheet (one feature per column) and then export it using "Save As" with file type "Text CSV" (set the field delimiter to be a comma).

Non-numerical values can be put in double quotes ("...") and indeed it can be done for unusual characters also.

Weka needs a file called XXX.arff in order to build a classifier (set of rules). Here XXX can be any name .The file we have just created is in the correct format for the second part of an "XXX.arff" file for Weka The first part of the file is used to describe the format of data. This contains, after a name for the "relation" that is represented in the file, for each feature ("attribute") in turn (in the same order as in the data file), a specification of the possible values of the feature. For the above example, the file in arff format is:

This example shows the 3 types of features most likely to have - those with string values, those with numerical values and those with a small number of possible values. Now Weka[22] is used to build a classifier for this data file . The steps are as follows:

1. Run Weka by selecting "Applications->Programming->Weka...". Select the Explorer
2. Click "Open file..." and select weather.arff file
3. Most classification algorithms need to know, for string-valued features, all the possible values they can have.. In the Preprocess tab of Weka, we have done the following for each string-valued feature:
    1. Click "Choose" under "Filter" and select the filter called filters->unsupervised->attribute->StringToNominal. (You don't need to do this after you have used this filter once).
    2. When the name appears after the "Choose" button, click on the name and in the box that appears enter the position (e.g. 1 or 2) of the feature we want to process after "attributeIndex".
    3. Click "OK" and the box disappears.
    4. Click "Apply".
4. Go to the Classify tab.
5. Choose a classifier Naïve Bayes in the same way as chosen a filter before.
6. The attribute that is to be predicted appears in the box below the "Test options" area.
7. Click "Start".
8. Various information about what has been learned and how well it accounts for the data have been provided. For the example "weather.arff" data.

Temperature is a numeric value; therefore, you can see min, max, means, and standard deviation in ‘Selected Attribute’ window. Missing = 0 means that the attribute is specified for all instances (no missing values), Distinct = 12 means that Temperature has twelve different values, and Unique = 10 means that other attributes or instances have the same 10 value as Temperature has. Temperature is a Numeric value; therefore, we can see the statistics describing the distribution of values in the data - Minimum, Maximum, Mean and Standard Deviation. Minimum = 64 is the lowest temperature, Maximum = 85 is the highest temperature, mean and standard deviation. Compare the result with the attribute table “weather.xls”; the numbers in WEKA match the numbers in the table. We can select a class in the ‘Class’ pull-down box. The last attribute in the ‘Attributes’ window is the default class selected in the ‘Class’ pull-down box. Building “Classifiers”

Classifiers in WEKA are the models for predicting nominal or numeric quantities. The learning schemes available in WEKA include decision trees and lists, instance-based classifiers, support vector machines, multi-layer perceptrons, logistic regression, and bayes’ nets. “Meta”classifiers include bagging, boosting, stacking, error-correcting output codes, and locally weighted learning. Once we have our data set loaded, all the tabs are available to us. Click on the ‘Classify’ tab. Classify’ window comes up on the screen. Before we run the classification algorithm, we need to set test options. Set test options in the ‘Test options’ box. The test options that available are

1. Use training set. Evaluates the classifier on how well it predicts the class of the instances it was trained on.

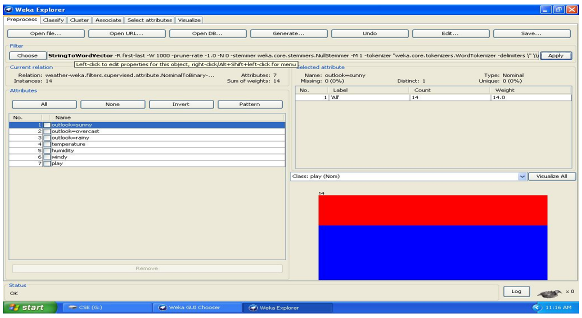
2. Supplied test set. Evaluates the classifier on how well it predicts the class of a set of instances loaded from a file. Clicking on the 'Set…' button brings up a dialog allowing choosing the file to test on.

3. Cross-validation. Evaluates the classifier by cross-validation, using the number of folds that are entered in the 'Folds' text field.

4. Percentage split. Evaluates the classifier on how well it predicts a certain percentage of the data, which is held out for testing. The amount of data held out depends on the value entered in the '%' field.

5. RESULTS AND CONCLUSIONS

This paper presents a comparison between supervised Machine Learning Algorithms NaiveBayes and combination of unsupervised machine learning algorithm K-Means clusterer and NaiveBayes Machine learning algorithm. When applied for weather data file we worked with above said two algorithms and concluded that using clustering before classification on data file weather.arff from dataset has optimized the performance. The dataset file weather.arff is used as input to WEKA for the above said algorithms. On the contrary, available datasets are represented as matrices of many columns (attributes) and usually few rows (examples) with lots of zeros (sparse information). Both the concepts and the representation appear very sparsely.



Filter: StringToWordVector -R first-last -W 1000 -prune-rate -1.0 -N 0 -stemmer weka.core.stemmers.NullStemmer -M 1 -tokenizer "weka.core.tokenizers.WordTokenizer -delimiters \" \\

Left-click to edit properties for this object, right-click/Alt+Shift+left-click for menu

Current relation
Relation: weather-weka.filters.supervised.attribute.NominalToBinary-...
Instances: 14
Attributes: 7
Sum of weights: 14

Attributes

No.	Name
1	outlook=sunny
2	outlook=overcast
3	outlook=rainy
4	temperature
5	humidity
6	windy
7	play

Selected attribute
Name: outlook=sunny
Missing: 0 (0%)
Distinct: 1
Type: Nominal
Unique: 0 (0%)

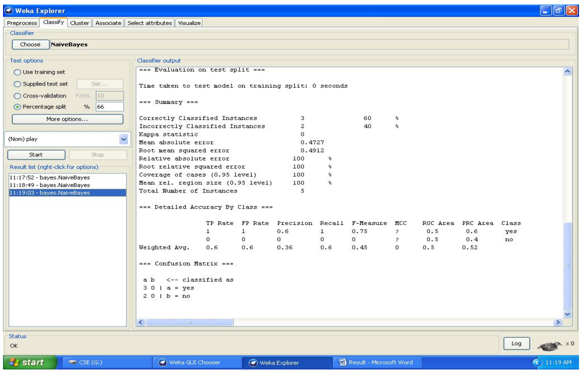
No.	Label	Count	Weight
1	'All'	14	14.0

Class: play (Nom)

14

Status
OK

Figure 5. Preprocessing of weather dataset in WEKA



Classifier: NaiveBayes

Test options
○ Use training set
○ Supplied test set
○ Cross-validation Folds 10
● Percentage split % 66

(Nom) play

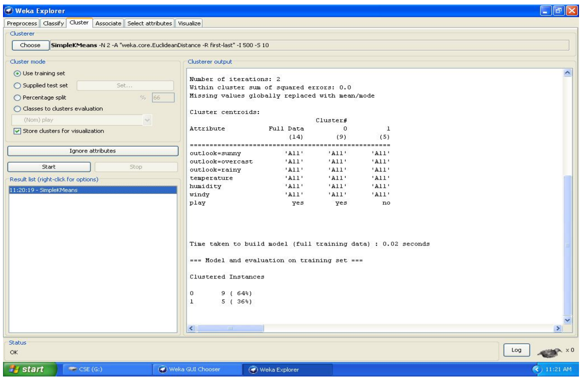
Result list (right-click for options)
11:17:52 - bayes.NaiveBayes
11:18:49 - bayes.NaiveBayes
11:19:03 - bayes.NaiveBayes

Classifier output

```
=== Evaluation on test split ===

Time taken to test model on training split: 0 seconds

=== Summary ===

Correctly Classified Instances           3               60      %
Incorrectly Classified Instances         2               40      %
Kappa statistic                          0
Mean absolute error                      0.4727
Root mean squared error                  0.4912
Relative absolute error                100      %
Root relative squared error            100      %
Coverage of cases (0.95 level)         100      %
Mean rel. region size (0.95 level)     100      %
Total Number of Instances                5

=== Detailed Accuracy By Class ===

                 TP Rate  FP Rate  Precision  Recall  F-Measure  MCC    ROC Area  PRC Area  Class
                 1        1        0.6        1       0.75       ?       0.5       0.6       yes
                 0        0        0          0       0          ?       0.5       0.4       no
Weighted Avg.    0.6      0.6      0.36       0.6     0.45       0      0.5       0.52

=== Confusion Matrix ===

 a b   <-- classified as
 3 0 | a = yes
 2 0 | b = no
```

Status
OK

Figure 6. Classify weather dataset in WEKA



Clusterer: SimpleKMeans -N 2 -A "weka.core.EuclideanDistance -R first-last" -I 500 -S 10

Cluster mode
● Use training set
○ Supplied test set
○ Percentage split % 66
○ Classes to clusters evaluation
(Nom) play
☑ Store clusters for visualization

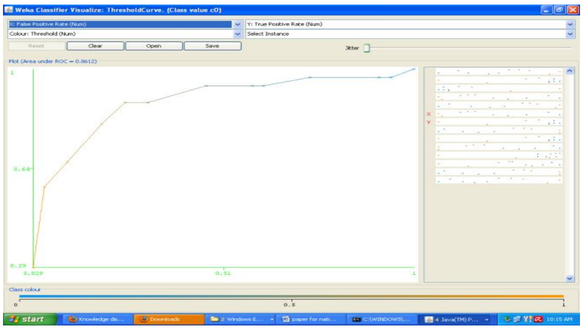
Result list (right-click for options)
11:20:19 - SimpleKMeans

Clusterer output

```
Number of iterations: 2
Within cluster sum of squared errors: 0.0
Missing values globally replaced with mean/mode

Cluster centroids:
                                   Cluster#
Attribute            Full Data          0          1
                        (14)          (9)        (5)
===================================================
outlook=sunny          'All'       'All'      'All'
outlook=overcast       'All'       'All'      'All'
outlook=rainy          'All'       'All'      'All'
temperature            'All'       'All'      'All'
humidity               'All'       'All'      'All'
windy                  'All'       'All'      'All'
play                     yes         yes         no




Time taken to build model (full training data) : 0.02 seconds

=== Model and evaluation on training set ===

Clustered Instances

0       9 ( 64%)
1       5 ( 36%)
```

Status
OK

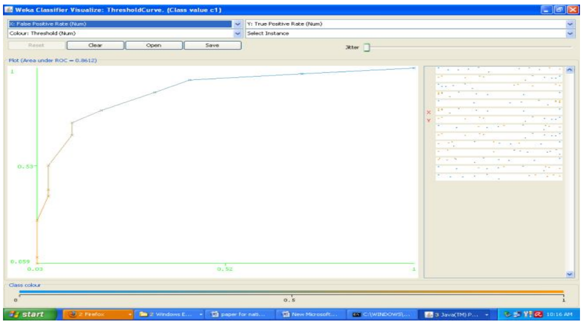
Figure 7. Clustering of weather dataset in WEKA


Weka Classifier Visualize: ThresholdCurve. (Class value c0)

X: False Positive Rate (Num)
Y: True Positive Rate (Num)
Colour: Threshold (Num)
Select Instance

Plot (Area under ROC = 0.8612)

Figure 8. Visualize threshold curve for cluster 1


Weka Classifier Visualize: ThresholdCurve. (Class value c1)

X: False Positive Rate (Num)
Y: True Positive Rate (Num)
Colour: Threshold (Num)
Select Instance

Plot (Area under ROC = 0.8612)

Figure 9. Visualize threshold curve for cluster 2

First we applied Naïve Bayes algorithm on data file weather.arff without clustering and then we combine the K-Means clustering algorithm the results are as follows:

Table 1: WEKA result on Naive Bayes Algorithm with K-Means clustering

Parameter	**Naive Bayes**	**K-Means Clusterer+NB**
Precision	0.807	0.84
Recall	0.81	0.86
Specificity	0.81	0.86
Sensitivity	0.255	0.24
F-Score	0.808	0.83
ErrorRate	0.265	0.126
Accuracy	81%	82.3%

6. OBSERVATIONS AND ANALYSIS

1.It may be observed from Table 1 that the error rate of binary classifier Naïve Bayes with Simple KMeans Clusterer is lowest i.e. 0.126 in comparison with Naïve Bayes classifier without clusterer i.e. 0.265, which is most desirable.

2. Accuracy of Naïve Bayes classifier with KMeans clusterer is high i.e. 82.3% which is highly required.

3. Sensitivity (TPR) of clusters (results of integration of classification and clustering technique) is higher than that of classes In an ideal world we want the FPR to be zero. FPR is lowest of integration of clustering and classification technique, in other words closet to the zero as compared with simple classification technique with Naïve Bayes classifier.

4. In an ideal world we want precision value to be 1.Precision value is the proportion of true positives out of all positive results. Precision value of integration of classification and clustering technique is higher than that of simple classification with Naïve Bayes classifier.

REFERENCES

[1] Banerjee, Arindam, and Langford, John. "An objective evaluation criterion for clustering", Proceedings of the 2004 ACM SIGKDD international conference on Knowledge discovery and data mining, August 22-25, 2004

[2] Dhillon, I., and Modha, D. ,"Concept decompositions for large sparse text data using clustering", *Machine Learning*, vol. 42, no. 1, pp. 143-175, 2001

[3] Zhimao Lu, Ting Liu, and Sheng Li,"Combining neural networks and statistics for Chinese word sense disambiguation", In Oliver Streiter and Qin Lu, editors, ACL SIGHAN Workshop, 2004.

[4] Leacock, C., Chodorow, M., and Miller, G. A., "Using corpus statistics and WordNet relations for sense identification", Computational Linguistics,2000

[5] Sin-Jae Kang, Jong-Hyeok Lee, "Ontology Word Sense Disambiguation by Using Semi-Automatically Constructed Ontology", Machine Translation Summit ,2001.

[6] Ramakrishanan G., Bhattacharyya P. ,"Word Sense Disambiguation using Semantic Nets based on WordNet" LREC, Spain,2002

[7] Reeve LH, Han H, Brooks AD: ,"WordNet: A Lexical Database for English**",** *Communications of the ACM,* 1995.

[8] Peng Jin, Xu Sun, Yunfang Wu, Shiwen Yu, "Word Clustering for Collocation-Based Word Sense Disambiguation", Computational Linguistics and Intelligent Text Processing, Volume: 4394, Pages: 267-274 648,2007

[9] A. Azzini, C. da Costa Pereira, M. Dragoni, and A. G. B. Tettamanzi," Evolving Neural Networks for Word Sense Disambiguation", Eighth International Conference on Hybrid Intelligent Systems, 2008.

[10] Rion Snow Sushant Prakash, Daniel Jurafsky, Andrew Y. Ng ,"Learning to Merge Word Senses", Computer Science Department Stanford University,2007.

[11]Yoong Keok Lee and Hwee Tou Ng and Tee Kiah Chia," Supervised Word Sense Disambiguation with Support Vector Machines and Multiple Knowledge Sources", Department of Computer Science National University of Singapore, 2004.

[12] G. Miller, "Wordnet: An on-line lexical database," international Journal of Lexicography, vol. 3(4), pp. 235–244, 1990.

[13] Y. G. I. Dhillon, J. Fan, "Efficient clustering of very large document collections in Data Mining for Scientific and Engineering Applications", R. N. R. Grossman, G. Kamath, Ed. Kluwer Academic Publishers, 2001.

[14] J. Sedding, "Wordnet-based text document clustering," Master's thesis,University of York, 2004.

[15] D. Yuret.," Some Experiments with a Naive Bayes WSD System",In Proceedings of the International Workshop an Evaluating Word Sense Disambiguation Systems, Senseval, 2004.

[16]Lee YK, Ng,"An empirical evaluation of knowledge sources and learning algorithms for word sense disambiguation", Proc EMNLP ,2002

[17] Arindam Chatterjee, Salil Joshii, Pushpak Bhattacharyya, Diptesh Kanojia and Akhlesh Meena, "A Study of the Sense Annotation Process: Man v/s Machine", International Conference on Global Wordnets ,Matsue, Japan,, Jan, 2012

[18] Brijesh Bhat and Pushpak Bhattacharyya, "IndoWordnet and its Linking with Ontology", International Conference on Natural Language Processing (ICON 2011), Chennai, December, 2011

[19] R. Ananthakrishnan, Jayprasad Hegde, Pushpak Bhattacharyya and M. Sasikumar, "Simple Syntactic and Morphological Processing Can Help English-Hindi Statistical Machine Translation", International Joint Conference on NLP (IJCNLP08), Hyderabad, India, Jan, 2008.

[20] Holmes, A. Donkin,I.H. Witten ," WEKA: A machine Learning work bench", In proceedings of the second Australian and New Zealand Conference on Intelligent Information Systems, 1996.

[21] Varun Kumar, Nisha Rathee," Knowledge discovery from database Using an integration of clustering and classification", (IJACSA) International Journal of Advanced Computer Science and Applications,Vol. 2, No.3, March 2011.

[22] Remco R. Bouckaert, Eibe Frank, Mark Hall, Richard Kirkby, Peter Reutemann ,"WEKA Manual for Version 3-7-5",The University of WAIKATO,October,2011.

AUTHORS

Neetu Sharma, completed BE (CSE) in 1996. M.Tech (CSE) in 2010 from Panjab University. Presently doing Ph. D. in Computer Science and Engg.. Having total teaching experience of 15 years. Presented two papers in National and three papers in International conferences sponsored by IEEE**.** Presented three papers in International Journals

S. Niranjan, did his M.Tech(Computer Engg.) from IIT Kharagpur in 1987. Completed Ph.D.(CSE) in 2004 and Ph.D.(I&CT) in 2007, Have total 26 years of teaching experience. Presently working as Principal in PDM College of Engg. Bahadurgarh(Haryana). Having 6 publications in journals and 30 other papers published in conferences.