Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X

IJCSMC, Vol. 3, Issue. 4, April 2014, pg.50 – 58

RESEARCH ARTICLE

Performance Analysis of Classification Algorithms

Payal Pahwa[1], Manju Papreja[2], Renu Miglani[3]

[1]Bhagwan Parshuram Institute of Technology, Rohini (New Delhi)-110085, INDIA
[2]GVM Institute of Technology and Management, Sonipat (Haryana)-131001, INDIA
[3]GVM Institute of Technology and Management, Sonipat (Haryana)-131001, INDIA
[1] pahwapayal@gmail.com; [2] manju.papreja@gmail.com; [3] rnkakkar@gmail.com

Abstract— ***Classification is a process of finding model for partitioning the data into different classes. It is a process of generalizing and assigning a class label to a set of unclassified cases. In drug design the classification algorithms helps to identify the class of new designed drug (test data set) on the basis of existing training data set. In this paper we analyze and compare the behavior of different kinds of classification algorithms on medical data set taking from literature.***

Keywords—Data mining; KDD; Classification; Bayesian; Decision tree

I. INTRODUCTION

Data mining, also known as Knowledge Discovery in Databases (KDD), is a process of nontrivial extraction of implicit, previously unknown and potentially useful information from data in databases. It is also known as knowledge discovery process or knowledge mining [1]. Other similar terms referring to data mining are: data dredging, knowledge extraction and pattern discovery [2].

Now a day's data mining tools and techniques are used by various organizations to take advantages of previous historical data/patterns. By using pattern recognition technologies and statistical and mathematical, data mining helps analysts recognize significant facts, relationships, trends, patterns, exceptions and anomalies that might otherwise go unnoticed. It is also a process to analyze large amount of data to obtain useful information leading to understanding of relationships within data items/chemical compounds to extract "hidden" information for decision making.

Data mining functionality are used to specify the kind of pattern to be found in data mining tasks. Classification is one of the functionality of data mining. Classification is basically a process of finding a model that describes and distinguishes data classes of test data set based upon set of training data.

This paper describes and analyses the performance of various classification algorithms on dataset taken from literature for identification classes of new drug designed by researchers.

II. DATA CLASSIFICATION ALGORITHMS

A. Naive Bayesian

The naive Bayesian classifier is a statistical classifier used for supervised learning. It is one of the fastest learning algorithms and can deal with any number of features or classes [2]. The naive Bayesian learning uses Bayes theorem to calculate the most likely class label of the new instance. Assume that each tuple is represented by n-dimensional attribute vector, X={x1,x2,……..,xn) with feature values of a instance {a1,a2,…..,an). Let *C* be the target feature which represents the class value, and {c1,c2,…,cm} represents the m-values that *C* can take. A test data tuple *d* is classified to the class with the maximum posterior probability. This can be derived from Bayes' theorem as given

$$P(C_i|\mathbf{X})=\frac{P(\mathbf{X}|C_i)P(C_i)}{P(\mathbf{X})}$$

The classification on *d* is defined as follows [2]:

$$C(d) = max \ \ [P(ci)P(x1, x_2 \ldots\ldots, ax_n|ci)$$

In naive Bayesian learning, since all features P(ci) are Constant ,So the class value given test data tuple,also known as new instance, is computed as

$$P(x1, x_2 \ldots\ldots, x_n|ci) = \prod_{i=1}^{n} P(x_i|c)$$

B. *Decision Tree Induction*

C4.5 is a program that creates a decision tree based on a set of labeled input data was developed by Ross Quinlan. The decision trees generated by C4.5 can be used for classification. These algorithms adopt a greedy (Backtracking) approach for construction of decision tree in top-down recursive divide and conquer manner. J48 is an open source Java implementation of the C4.5 algorithm in the Weka data mining tool [3].

The basic algorithm for inducing a decision tree from training tuples for classification as [4].

- Create a node N
- If tuples in D are all of the same class, C then
  Return N as a leaf node labeled with the class C
- If attribute list is Empty then
  Return N as a leaf node labeled with the majority class in D
- Apply Attribute_selection method to find best splitting criterion.
- Label node N with splitting criterion
- If splitting attribute is discrete valued and multiway splits allowed then
  Attribute list←attribute list-splitting attribute
- For each outcome j of splitting criterion
  let Dj be the set of data tuples in D satisfying outcome j
- If Dj is empty then
  attach a leaf labeled with the majority class in D to the node N.

C. *Multilayer Perceptron*

A multilayer perceptron (MLP) is a feedforward artificial neural network(ANN) model that maps sets of input data onto a set of appropriate outputs. A MLP consists of multiple layers of nodes in a weighted directed graph, with

each layer fully connected to the next one. It consists of three or more layers :an input and an output layer with one or more hidden layers[5]. A graphical representation of an MLP is shown below:

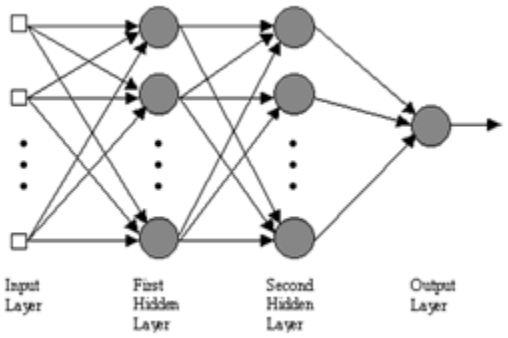

Figure 1: Graphical representation of MLP

Each layer has one or more neurons. Every neuron i is connected to the j neurons of the next layer by a set of weighted links denoted by W1i, . . . , Wji. At the input layer, {a1, a2, . . . , am} represent m input signals associated with the m attributes. At the hidden and output layers, each neuron j receives the input signals as a linear combination of the output given by:

$$v_j = \sum_{i=0}^{m} w_{ji} a_i$$

The linear combinations are transformed into output signals using an activation function $\varphi(v_j)$. These signals are sent in a forward direction layer by layer to the output layer which delivers an output yj for each output neuron j. In classification, each class is associated with an output neuron and the prediction is typically given by the one with the highest activation level [6].

The goal is to define the values for the connections weights that return the outputs which lowest error, i.e., the output is most similar to the desired value, d(n). One method to learn the weights is BP, which propagates errors in a backward direction from the output layer to the input layer, updating the weight connections if an error is detected at the output layer. A weight correction on the nth training example is defined in terms of the error signals $c_j(n)$ for each output neuron j. Considering a sequential mode in which the weights are updated after every training example, the predicted output $y_j(n)$is compared with the desired target $d_j(n)$ and the individual error $e_j(n)$ is estimated as follows: $e_j(n) = d_j(n) - y_j(n)$. In a typical NN, the error signal is equal to the individual error, because the predicted output is directly compared with the target. The correction is given by$\Delta w_{ji} = n\delta j(n) y_i(n)$, where η is the learning rate, $y_i(n)$ is the output signal of the previous neuron *i* and the local gradient δ_j is defined by$\delta_j = e_j \delta'\left(v_j(n)\right)$. For a hidden neuron *i*, the local gradient is defined in a recursive form by $\delta_i(n) = \delta'_i\left(v_i(n)\right) \sum_j \delta_j(n) w_{ij}(n)$. [6]

III. RADIAL BASIS FUNCTION

Radial Basis Function (RBF) Network is a type of Artificial Neural Network for supervised learning [7]. approximation. It uses RBF as a function which is usually Gaussian and the outputs are inversely proportional to the distance from the center of the neuron [7]. The idea of Radial Basis Function (RBF) Networks

derives from the theory of function.The Radial Basis Function (RBF) procedure produces a predictive model for one or more dependent (target) variables based on values of predictor variables.
The RBF network consists of one hidden layer of basis functions, or neurons. At the input of each neuron, the distance between the neuron center and the input vector is calculated. The output of the neuron is then formed by applying the basis function to this distance. The RBF network output is formed by a weighted sum of the neuron outputs and the unity bias shown.

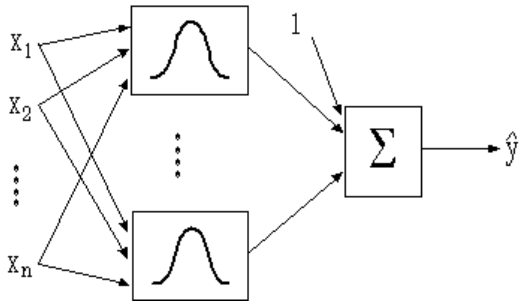

Figure 2: An RBF network with one output.

For an RBF to form a successful model of any function, the network structure needs to dispose of many radial neurons. If there is a sufficient number of radial neurons, each important detail of a modeled function can have the needed radial neuron attached, which guarantees that the obtained solution shall genuinely reproduce the given function. Radial networks consist of neurons, whose activation functions are[8]:

$x \to \varphi(\|x - c\|, x \in R^n$
Where $(\|.\|)$ is the Euclidean norm.
Functions$\varphi(\|x - c\|)$ are called radial base functions. Their values change radially from the centre c. A radial neuron is defined by its centre and a parameter defined as "ray". Neurons in the hidden layer are defined by formula. A neuron in output layer stands for the operation of the weighed sum of signals of output neurons in the hidden layer, and can be expressed as [8]:

$$y = \sum_i w_i \varphi_i = \sum_i w_i \varphi(\|x - c_i\|).$$

IV. AGGREGATING ONE- DEPENDENCE ESTIMATORS (AODE)

AODE (Aggregating One-Dependence Estimators) is considered one of the most interesting representatives of the Bayesian classifiers, taking into account not only the low error rate it provides but also its efficiency. Until now, all the attributes in a dataset have had to be nominal to build an AODE classifier [9].

The AODE classifier is considered an improvement on NB and a good alternative to other attempts such as Lazy Bayesian Rules (LBR) and Super-Parent TAN (SP-TAN), as they offer similar accuracy ratios, but AODE is significantly more efficient at classification time compared to the first one and at training time compared to the second [10].

V. SIMPLE LOGISTIC

Logistic Regression is an popular analytic tool. It is used to predict the probability that the 'event of interest' will occur as a linear function of one (or more) continuous and/or dichotomous independent variables.. The logistic

function is used to estimate, as a function of unit changes in the independent variable, the probability that the event of interest will occur. This function is often called the link function in that it connects, or 'links' changes in values of the independent variables to increasing (or decreasing) probability of occurrence of the event being modeled by the dependent variable.

VI. DATABASE SOURCES

The data base from literature is taken and applied on Weka tool for comparison and analysis of various Classification algorithms. The Data base used in our paper is shown below along with no.of attributes along with no.of tuples and possible class label for a tuple.

Sr. No.	**Dataset**	**No. of Tuples**	**No. of attributes**	**Class**
1	*PTP 1B INHIBITORS*	47	4	Molecules_type: Type A, Type-B,Type-C
2	*Selective Inhibitors*	435	8	INHIBITORS: IA,IB,IC,ID
3	Drugs	1750	5	Diabetes mellitus,Antidiabetic drugs

VII. EXPERIMENTS

Experiments were conducted under the framework of Weka to study the various kinds of Classification Algorithms on four datasets. The experiments compares various results in terms of classification measured by percentage accuracy of no. of correctly classified instances .The environmental variables are same for each algorithm and dataset. The algorithms are compared by using various parameters like tprate, fprate, precision, recall, time taken etc. Confusion matrix is a useful tool for analyzing how well the classification algorithm can recognize the tuples of different classes. A Confusion Matrix with m classes is of order m*m.

TABLE 1
CONFUSION MATRIX WITH 2 CLASSES

	C1	C2
C1	True positive(TP)	False negative(FN)
C2	False positive(FP)	True negative(TN)

TP rate is the true positive rate and the FP rate is the false positive rate. Precision is the ratio of the number of true positives(i.e. no.of tuples that are correctly identified) to the total number of irrelevant and relevant records retrieved. Recall is the ratio of the number of relevant records retrieved to the total number of relevant records in the database.

$$\textbf{Precision} = \frac{\textbf{TP}}{\textbf{TP+FP}} \times \quad \textbf{100}\%$$

$$\textbf{Recall} = \frac{\textbf{TP}}{\textbf{TP+FN}} \times \quad \textbf{100}\%$$

Where, TP, TN, FP, and FN are as represented in the confusion matrix.

A. *DataSet-1*

TABLE 2
PTP 1B INHIBITORS DATA SET APPLIED ON VARIOUS ALGORITHMS.

Algorithm	TP Rate	FP Rate	Precision	Recall	Classification Accuracy (in per)	Time (in sec)
Decision Tree Induction	0.964	0.045	0.955	0.960	95.5	0.01
AODE	0.929	0.059	0.940	0.936	94.0	0
Navie Bayes	0.90	0.125	0.878	0.88	88.0	0
RBF Network	0.92	0.070	0.929	0.930	93.0	0.1
Simple Logistic	0.942	0.034	0.965	0.962	96.5	4.39
Multilayer Perceptron	0.998	0.004	0.978	0.980	97.8	2.32

B. *DataSe. t-2*

TABLE 3
SELECTIVE INHIBITORS DATA SET APPLIED ON VARIOUS ALGORITHMS.

Algorithms	TP Rate	FP Rate	Precision	Recall	Classification Accuracy (in per)	Time (in sec)
Decision Tree Induction	0.972	0.031	0.969	.968	97.0	0.01
AODE	0.949	0.047	0.953	.95	95.4	0
Navie Bayes	0.903	0.089	0.910	.900	91.0	0
RBF Network	0.929	0.062	0.937	.941	93.8	0.06
Simple Logistic	0.963	0.041	0.959	.96	96.0	0.43
Multilayer Perceptron	1	0	1	1	100	0.03

C. *.DataSet-3*

TABLE 4
DRUG DATA SET APPLIED ON VARIOUS ALGORITHMS

Algorithms	TP Rate	FP Rate	Precision	Recall	Classification Accuracy (in per)	Time (in sec)
Decision Tree Induction	1	0	1	1	100	0
AODE	0.978	0.0181	0.982	0.97	98.2	0
Navie Bayes	0.918	0.08	0.920	0.91	91.9	0
RBF Network	1	0	1	1	100	0.01
Simple Logistic	0.929	0.129	0.878	0.90	88.0	0.04
Multilayer Perceptron	1	0	1	1	100	0.03

The Comparison between various classifications algorithms on given datasets are represented in the form of graphs.

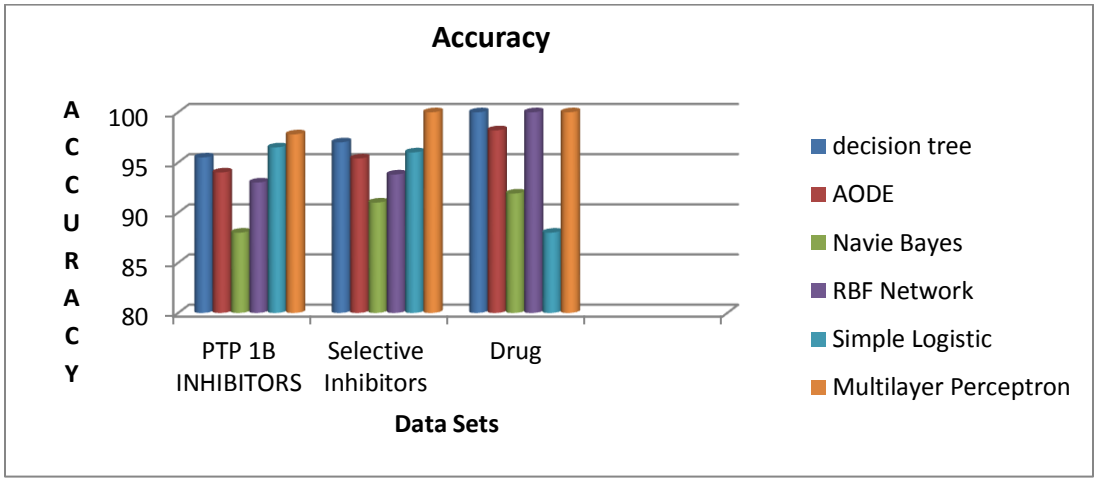

Figure 3: Graph shows accuracy of various algorithms using different dataset.

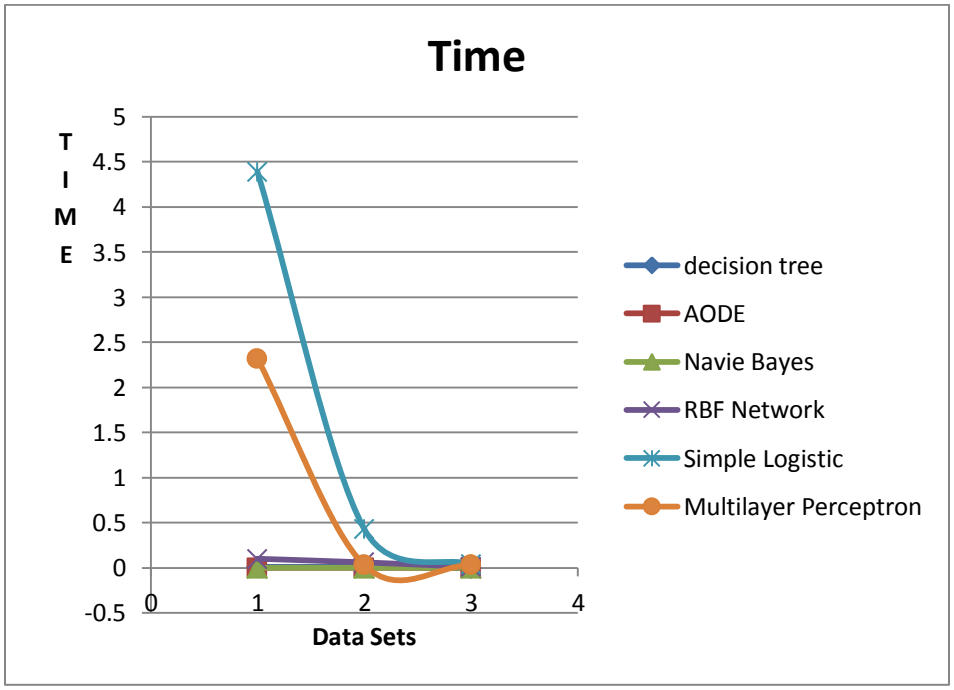

Figure 4: Graph shows time of various algorithms using different dataset.

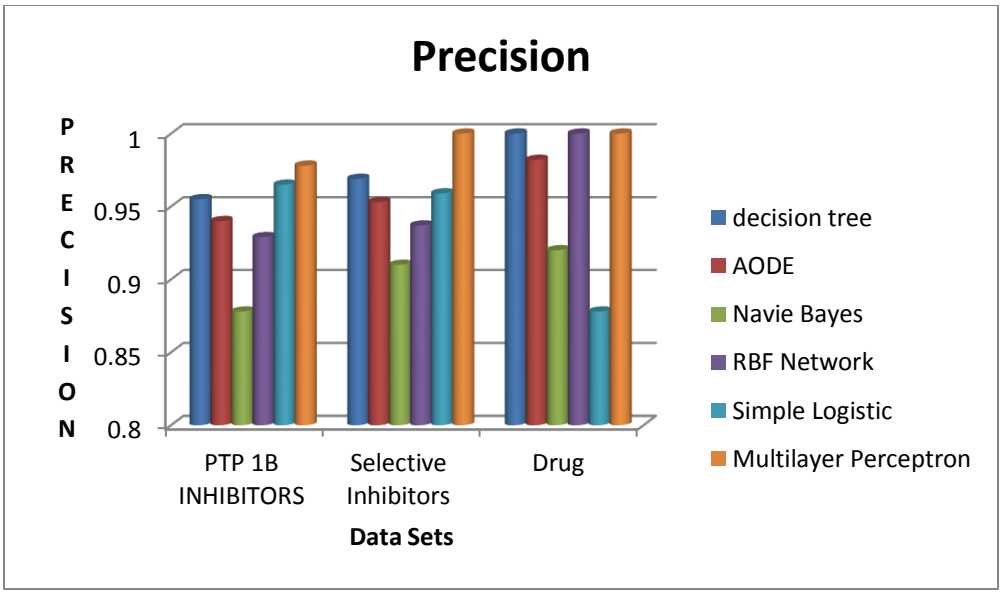

Figure 5: Graph shows Precision of various algorithms using different dataset.

VIII. CONCLUSION

In our paper, we use Weka Tool for analysis and comparison of various classification algorithms on three data sets namely:*PTP 1B INHIBITORS, Selective Inhibitors and Drugs.*After comparative analysis we concluded that MLP algorithm is more accurate that other algorithms but it takes more time for classification. The time taken by Algorithms AODE and naïve Bayesian classification is minimum. We also analyze that the precision and Recall value of MLP technique is maximum .In our paper we also represent this analysis graphically and finally concluded that the performance of MLP is better than all other algorithms.

REFERENCES

[1] Osmar R. Zaïane et. al., "*Introduction to Data Mining*", 1999 CMPUT690 Principles of Knowledge Discovery in Databases.

[2] Chang-Hwan Lee et. al., "*Calculating FeatureWeights in Naive Bayes with Kullback-Leibler Measure*", 2011 11th IEEE International Conference on Data Mining.

[3] Gholap, Jay. "*Performance tuning of J48 algorithm for prediction of soil fertility*." arXiv preprint arXiv:1208.3943 (2012).

[4] Anshul Goyal et. al. , "*Performance Comparison of Naïve Bayes and J48 Classification Algorithms*", International Journal of Applied Engineering Research, ISSN 0973-4562 Vol.7 No.11 (2012), © Research India Publications; http://www.ripublication.com/ijaer.htm.

[5] Ribeiro, Geraldina, et al. "*Multilayer perceptron for label ranking*." Artificial Neural Networks and Machine Learning–ICANN 2012. Springer Berlin Heidelberg, 2012. 25-32.

[6] Z. Caiqing, Q. Ruonan, and Q. Zhiwen, "*Comparing BP and RBF Neural Network for Forecasting the Resident Consumer Level by MATLAB*," International Conference on Computer and Electrical ngineering, 2008 (ICCEE 2008), 20-22 Dec. 2008, pp.169-172.

[7] Izabela Rojek et. al. , "*Neural networks as performance improvement models in intelligent CAPP systems*" Control and Cybernetics vol. 39 (2010) No. 1.

[8] M. Julia Flores et. al., "*GAODE and HAODE: Two Proposals based on AODE to Deal with Continuous Variables*", Appearing in Proceedings of the 26 th International Conference on Machine Learning, Montreal, Canada, 2009.

[9] Webb, G. I., Boughton, J. R., & Wang, Z. (2005). Not So Naive Bayes: Aggregating One-Dependence Estimators. Mach. Learn., 58, 5–24.

[10] Andrew H. Karp et. al., "*Getting Started With Proc Logistic*", http://www2.sas.com/proceedings/sugi26/p248-26.pdf.

[11] Provost, F., Fawcett, T., &Kohavi, R., "*The case against accuracy estimation for comparing classifiers*".In Proceedings of the 15th International Conference on Machine Learning. San Francisco, CA: Morgan Kaufmann (1998).

