STUDY AND EVALUATION OF CLASSIFICATION ALGORITHMS IN DATA MINING

Archit Verma[1]

[1] *M.Tech, Computer Science and Engineering, from United College of Engineering and Research, Allahabad Uttar Pradesh, India*

---***---

Abstract – *Data Mining is the process of discovering hidden or unknown patterns in huge datasets that are potentially useful and ultimately understandable. The goal of data mining is to extract useful information from huge data sets and to transform it into an understandable structure, using combined techniques of statistics, machine learning and database systems. Data Mining algorithms learns from training datasets to build the model, that can be used on unknown data for prediction. Here we will discuss classification technique that is often called as supervised learning technique. Classification is used to predict the categorical class label of a data instance , so as to classify it into one of the predefined classes. Classification is a two step process, in first step classification algorithm uses training dataset to build a classifier , and then in the second step this classifier is used to predict the class label of an unlabeled data instance . The classifier is like a function that maps a data instance to a label. The classifier that we will discuss here are Naive Bayes, Multilayer Neural Network, Decision tree , Random forest , Support Vector Machine and K-NN. Building an accurate and an efficient classifier is an important task in data mining. We will evaluate these algorithms on various datasets using evaluation metrics such as accuracy, precision, recall, F1-Score and time by WEKA. WEKA is a machine learning tool written in Java.*

Key Words: Data Mining, Supervised Learning, Classification Algorithms, Naive Bayes, Multilayer Neural Network, Decision Tree, Random Forest, Support Vector Machine, K-NN, Accuracy, Precision, Recall, F1-Score and WEKA

1. INTRODUCTION

Data Mining is the process of discovering hidden or unknown patterns in huge datasets involving combined techniques of machine learning, statistics and database systems.
Data Mining is the process of discovering hidden or unknown patterns in huge datasets that are potentially useful and ultimately understandable. The goal of data mining is to extract useful information from huge data sets and to transform it into an understandable structure. Data mining makes use of machine learning that is an application of artificial intelligence. Machine learning aims to build the systems that has the ability to learn without being explicitly programmed, these systems learns from training datasets. Data Mining algorithms learns from training datasets to build the model, that contains knowledge or summary of whole dataset, that can be used for future prediction. The two most important techniques in data mining are classification and clustering, that are often called as supervised and unsupervised learning technique respectively. Supervised learning is the machine learning task of learning a function from labeled training datasets consisting of a set of training examples, that consists of input-output pairs, this function maps an input to output. Classification is used to predict the categorical class label of a data instance. Classification algorithm finds a function that assigns an item to one of the predefined target categories or classes. Data instance for training is represented as attribute/feature vector say X and its label y, represented as (X,y) ,where X=(x_1,x_2,x_3,......,x_n) such that x_1,x_2,x_3,......,x_n are the values of attributes A_1,A_2,A_3,.....,A_n in the dataset, and y represents the value of label . Classification is a two step process, in first step classification algorithm uses training data to build a classifier, and then in second step it uses this classifier to predict class label of an unlabeled data instance.
The classifier that we will discuss here are, Naive Bayes classifier that is based on Bayes' theorem of posterior probability and assumption of conditional independence between attributes, Multilayer Neural Network that uses back propagation learning algorithm which finds sets of weight that can model data by applying gradient descent method to minimize the mean squared error between predicted and actual class label, Decision tree learning algorithm that uses greedy approach in attribute selection for internal nodes, and constructs tree in top-down recursive divide and conquer manner with attribute selection at each recursive step based on any one of the measures that is information gain/gain ratio/gini index fixed throughout the algorithm. Random forest classifier that creates a number of decision trees at training time, each such individual decision tree is created from the tuples obtained by random sample with replacement of training dataset, and random selection of attributes at each split, after that it predicts the class label of unknown data instances from these decision trees and outputs the majority class label ,Support Vector Machine that works by finding an optimal separating hyperplane between classes ,and works for both of linear and non-linear data, and K-

nearest neighbor that predicts the class label of unknown data instance by taking majority of the class labels of K-nearest data instances.
In this paper we will evaluate these classification algorithm in WEKA on evaluation metrics such as accuracy, precision, recall, F1-Score and time on various datasets.

2. ISSUES REGARDING CLASSIFICATION

The following issues are important to be considered during classification for data preprocessing and comparing algorithm:

2.1 Processing Data before Classification

[1]The following preprocessing steps are applied to the data to achieve better ,accuracy, efficiency and scalability of the classification process:-

1.)Data cleaning: This refers to the preprocessing of data with the view of removing or reducing noise(applying smoothing technique) ,and treatment of missing values(by replacing missing value with most commonly occurring value , or a probable value calculated from statistical technique).

2.)Relevance analysis: The dataset may contain redundant attributes. The techniques like correlation analysis can be used to find out if any two attributes are statistically related. As an example, the high correlation between attribute A_1 and A_2 ,would result in removal of one of the attribute. Another relevance analysis is Attribute subset selection that finds a reduced set of attributes, such that the attained probability distribution of data classes is as near as possible to the original probability distribution using all attributes. This is how we detect attributes that do not contribute to classification. In theory, the time consumed on relevance analysis, when added to the time consumed on learning from the derived reduced set of attributes, should be less than the time consumed in learning from original set of attributes. Hence this step can improve upon classification efficiency and scalability.

3.)Data Transformation and Reduction: Sometimes the dataset ,may be required to be transformed by normalization, especially when algorithm like "back propagation of neural network" is used which requires numerical values to be provided into input layer ,or algorithm like "K-NN" which requires distance measure. Normalization is done by scaling all values of an attribute under consideration , so that they lie within a specified range, such as -1.0 to 1.0 or 0.0 to 1.0. This prevents attributes that are initially large range like "salary" from outweighing the effect of small range binary attributes.
Generalizing the data to a higher level concept is also a way of transforming data. To do it we apply knowledge of concept hierarchy. Let us take an example of continuous valued attribute like “salary” that has numeric values, these numeric values can be generalized to discrete ranges , such as below average ,average and above average. In a similar manner attribute like street can be generalized to a higher level like city. Since generalization is compressing data, a small number of input/output operations are involved during learning.

2.2 Criteria of Comparing Classification

[1]1.)Accuracy: The accuracy of the classifier refers to the ability of a given classifier to correctly predict the class label of unseen data.

2.)Speed: This refers to the computational cost involved in building model, and using it.

3.)Robustness: This is the ability of the classifier to make correct prediction, even if the data is noisy, and have missing value.

4.)Scalability: It is the ability to construct the classification model efficiently, given large amount of data.

5.)Interpretability: It refers to the ease or extent of understanding provided by the classifier.

3. CLASSIFICATION ALGORITHMS

3.1 Naïve Bayes

Naive bayes classification algorithm is based on Bayes' theorem of posterior probability. This algorithm works by predicting probability that a given data instance belongs to a particular class. Data instance is represented by a vector $X=(x_1,x_2,x_3,......,x_n)$ where $x_1,x_2,x_3,....,x_n$ are the values of the n attributes $A_1,A_2,A_3,.....,A_n$ in the dataset. The probability that a given data vector will belong to class C, is called posterior probability and is denoted by P(C|X) ,probability of C conditioned on X, similarly the probability that a given data instance X is known to be in class C is called likelihood and is denoted by P(X|C), probability of X conditioned on C, the prior probability of class C is denoted by P(C),and the prior probability of X is denoted by P(X).The computation of P(X|C) is done as below from the assumption of conditional independence between attributes:

$$P(X|C)=\prod_{i=1}^{i=n}P(x_i|C) \tag{1}$$

In the above equation of computing likelihood the term $P(x_i|C)$ denotes the probability that a data instance in training data set has value of attribute $A_i=x_i$ and belongs to class C.

Now by Bayes' theorem, the probability that a given data instance belongs to class C is given as:

$$P(C|X) = \frac{P(X|C)P(C)}{P(X)} \tag{2}$$

So, if there are m classes , we find $P(C_1|X), P(C_2|X), \ldots\ldots, P(C_m|X)$ and then we classify X into class C_i such that $P(C_i|X) > P(C_j|X)$ for all $j \epsilon \{1,2,3\ldots m\}$ and $j \neq i$. Since P(X) is constant we have to work with P(X|C)P(C) in our calculation.
So final formula is Y =C_i for some i as under:

$$Y = \underset{i \in 1,2,3,..m}{\arg\max}\ P(C_i) \prod_{j=1}^{j=n} P(x_j|C_i) \tag{3}$$

This is also called maximum posteriori hypothesis.

In case of categorical attribute say A_j the value of P $(x_j|C_i)$ is simply calculated by counting number of tuples that has value of attribute A_j as x_j and belongs to class C_i, divided by number of tuples of class C_i in dataset DS.
In case of continuous valued attribute we apply Gaussian distribution.

$$P(x_j|C_i) = \frac{1}{\sqrt{2\pi\sigma_{C_i}}} e^{-\frac{(x_j - \mu_{C_i})^2}{2\sigma_{C_i}{}^2}} \tag{4}$$

[1]Many empirical studies have shown that Naive Bayes is comparable to Decision Tree and Neural network. In theory Bayesian classifiers have minimum error rate but this is not always in practice as some inaccuracies might be their due to assumption of conditional independence between attributes. It is used in text categorization.

3.2 Multilayer Neural Network classifier based on Backpropagation Algorithm

[2]Neural network is a set of connected nodes that is an input/output units, and each connection has a weight associated with it. The network learns by adjusting its weight according to training data vectors. The network adjusts its weight so as to minimize the mean squared error between predicted class label and actual class label, for this it employs gradient descent method. Neural network learning is also called connectionist learning due to connections between nodes.
Backpropagation algorithm works by iteratively processing a data set of training tuples. It compares network predicted value for each tuple with the actual known target value.
The input layer nodes simply transfer the received input to output port of that node, where as nodes of hidden layers and output layers apply sigmoid functions
f(x)= $\frac{1}{1+e^{-x}}$ to the received input $in_j = \sum_k W_{k,j} a_k$ and outputs value $a_j = f(in_j)$.

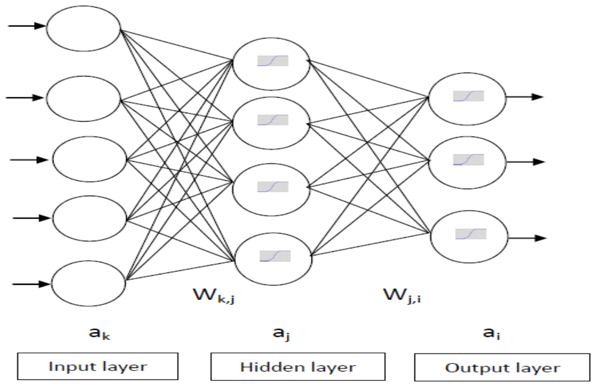

Fig-1: Diagram of Multilayer Neural network for dataset with five attributes and three classes

We use sigmoid function as it is a non-linear and differentiable function. It can be shown that f'(x)=f(x)(1-f(x)).The weights are updated according to gradient descent rule with learning rate denoted by η.

$$W = W - \eta \frac{\partial E}{\partial W} \tag{5}$$

Let us derive the formula for updating weights between hidden layer and output layer, and weights between input layer and hidden layer. Consider the squared error E on a single tuple, which is sum of squared differences between target value and predicted value of nodes in the output layer:

$$E = \frac{1}{2}\sum_i (y_i - a_i)^2 \tag{6}$$

$$\frac{\partial E}{\partial W_{j,i}} = -(y_i - a_i)\frac{\partial a_i}{\partial W_{j,i}} = -(y_i - a_i)\frac{\partial f(in_i)}{\partial W_{j,i}}$$

$$= -(y_i - a_i) f'(in_i)\frac{\partial in_i}{\partial W_{j,i}} = -(y_i - a_i) f(in_i)(1 - f(in_i))\frac{\partial}{\partial W_{j,i}}\sum_j W_{j,i} a_j$$

$$= -(y_i - a_i) f(in_i)(1 - f(in_i)) a_j = -a_j \delta_i \tag{7}$$

$$\text{where } \delta_i = (y_i - a_i) f(in_i)(1 - f(in_i))$$

Similarly

$$\frac{\partial E}{\partial W_{k,j}} = -\sum_i (y_i - a_i)\frac{\partial a_i}{\partial W_{k,j}} = -\sum_i (y_i - a_i)\frac{\partial f(in_i)}{\partial W_{k,j}}$$

$$= -\sum_i (y_i - a_i) f'(in_i)\frac{\partial in_i}{\partial W_{k,j}} = -\sum_i (y_i - a_i) f(in_i)(1 - f(in_i))\frac{\partial in_i}{\partial W_{k,j}} = -\sum_i \delta_i \frac{\partial}{\partial W_{k,j}}\sum_j W_{j,i} a_j$$

$$= -\sum_i \delta_i W_{j,i}\frac{\partial a_j}{\partial W_{k,j}} = -\sum_i \delta_i W_{j,i}\frac{\partial f(in_j)}{\partial W_{k,j}} = -\sum_i \delta_i W_{j,i} f'(in_j)\frac{\partial in_j}{\partial W_{k,j}}$$

$$= -\sum_i \delta_i W_{j,i} f(in_j)(1 - f(in_j))\frac{\partial in_j}{\partial W_{k,j}}$$

$$= -\sum_i \delta_i W_{j,i} f(in_j)(1-f(in_j)) \frac{\partial}{\partial W_{k,j}} \sum_k W_{k,j} a_k$$

$$= -\sum_i \delta_i W_{j,i} f(in_j)(1-f(in_j)) a_k = -a_k \delta_j \qquad (8)$$

$$where \quad \delta_j = \sum_i \delta_i W_{j,i} f(in_j)(1-f(in_j))$$

$$= f(in_j)(1-f(in_j)) \sum_i \delta_i W_{j,i}$$

Once we have computed $\frac{\partial E}{\partial W_{j,i}}$ and $\frac{\partial E}{\partial W_{k,j}}$ we update the weight as under:

$$W_{j,i} = W_{j,i} + \eta a_j \delta_i \qquad (9)$$

$$W_{k,j} = W_{k,j} + \eta a_k \delta_j \qquad (10)$$

From the above derivation it is clear that the neural network adjusts its weight by back propagating errors, as δ_i values for output layer is used in computation of δ_j values for hidden layer.
In the above diagram we have shown a neural network that can classify data instance into three classes. The input training data instances is simply feeded into the input layer in the form of numerical values [x_1, x_2, x_3, x_4, x_5]. If the data instance belongs to class c_1 then output layer should output [1,0,0], similarly if it belongs to class c_2 it should output [0,1,0], and similarly if it belongs to class c_3 it should output [0,0,1], this is how training data instances are used for training. For a dataset DS with |DS| tuples and a neural network with N_w weights, time complexity of each epoch(one iteration through the training set) is O(|DS|×N_w). [3]Neural network has the ability to handle noisy data. It works well with continuous numeric valued attributes. It also has ability to classify unknown patterns.

3.3 Decision Tree Classifier

[1]Decision tree learning algorithm is a top-down , recursive divide and conquer, greedy algorithm. Decision tree is a tree like structure in which internal nodes are labeled with attributes and denotes test on an attribute, outgoing edges denotes outcome of the test, and leaf nodes denotes classes. Nodes labeled with attributes divides the training dataset into two or more subsets based on the values of the attribute, as an example if we have a node labeled with an attribute age, that has three possible values young, middle-aged and senior, then this node will split the training dataset into three subsets, tuples with age= young, then tuples with age= middle-aged, and then tuples with age=senior, these values of age are shown on the outgoing edges. The attribute selection at each stage is done by an attribute selection measure. It is a heuristic for selecting the splitting criterion, that divides the training dataset best, into distinct classes. The three attribute selection measure are: (Information gain used in ID3),(Gain ratio used in C 4.5) and (Gini index used in CART).Then decision tree algorithm works recursively on subsets of data, and remaining attributes. The recursion stops when any one of the following terminating conditions is true, if we get all tuples of the same class, then we label leaf node with that class, or if we get attribute list empty, then we label leaf node with majority class of tuples, or if there are no tuples left, then we label leaf node with the majority class of the node's parent training tuples. The time complexity of decision tree algorithm for dataset DS with n attributes and |DS| tuples is O(n×|DS|×log(|DS|)).

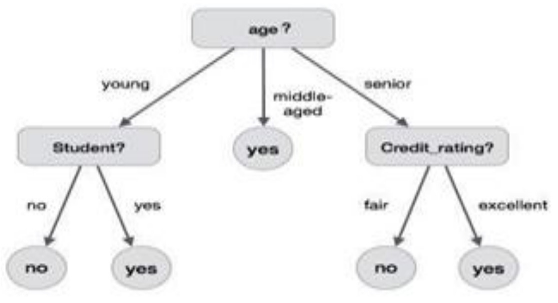

Fig-2: Diagram representing Decision Tree[9]

Now we discuss attribute selection measure:

1.)ID3(Information Gain)

Information gain is an attribute selection measure. It is based on concept of entropy. Entropy is a measure of surprise or randomness in the dataset. It also signifies the average amount of information needed to classify a tuple in the dataset. If we are certain about the event with probability p=1 , then entropy will be zero means no surprise. As an example, if all tuples belong to the same class, then entropy is zero. The formula for calculating entropy of dataset DS that has |C| classes:

$$Entropy(DS) = \sum_{i=1}^{i=|C|} -(p_i) log_2(p_i) \qquad (11)$$

, where p_i is the probability that an arbitrary tuple belongs to a class C_i.
The idea behind this approach is to find an attribute that has the most information gain. Information gain is the measure of the difference between entropy before and weighted average entropy after the dataset DS is split on an attribute A. It measures how much uncertainty is reduced by splitting. Now equation for computing information gain of an attribute A that has v distinct values $A_{val_1}, A_{val_2}, \ldots, A_{val_v}$ is as under:

$$Gain(A) = Entropy(DS) - \sum_{j=1}^{j=v} \frac{|DS_j|}{|DS|} Entropy(DS_j) \quad (12)$$

It should be noted that |DS| denotes number of tuples in dataset DS , while $|DS_j|$ refers to number of tuples that has value of attribute $A=A_{valj}$
We choose the attribute with maximum Gain(A)

2.) C.4.5(Gain Ratio)

[1] Information Gain is biased towards selecting attribute with many outcomes. Sometimes this can be useless if we have a primary key like attribute in our training dataset. So C 4.5 is developed as an extension of ID3.
The equations for calculation to Gain ratio of an attribute A is as under:

$$SplitInfo(A) = -\sum_{j=1}^{j=v} \frac{|DS_j|}{|DS|} log_2 \frac{|DS_j|}{|DS|} \quad (13)$$

$$GainRatio(A) = \frac{Gain(A)}{SplitInfo(A)} \quad (14)$$

We select the attribute with maximum GainRatio(A).
Although gain ratio adjusts for biases, it tends to prefer unbalanced splits in which one partition is much smaller than others.

3.)CART(Gini Index)

[19]Gini impurity is a measure of how often a randomly chosen instance from the dataset would be assigned a class incorrectly if it was randomly assigned a class according to the distribution of classes in the dataset.
The Gini impurity can be computed by taking summation over the probability p_i of a data item with class label i being chosen times the probability $\sum_{k\neq i} p_k = 1 - p_i$ of assigning incorrect class to that data item.
So to compute gini index of a dataset DS that has |C| classes, we compute:

$$Gini(DS) = \sum_{i=1}^{i=|C|} p_i \sum_{k\neq i} p_k = \sum_{i=1}^{i=|C|} p_i(1-p_i) = \sum_{i=1}^{i=|C|} (p_i - p_i^2)$$

$$= \sum_{i=1}^{i=|C|} p_i - \sum_{i=1}^{i=|C|} p_i^2 = 1 - \sum_{i=1}^{i=|C|} p_i^2$$

$$Gini(DS) = 1 - \sum_{i=1}^{i=|C|} p_i^2 \quad (15)$$

For each attribute A a binary split is considered , for example if an attribute A has three values {a,b,c} then its subsets are φ ,{a},{b},{c},{a,b},{b,c},{a,c},{a,b,c} out of which φ and {a,b,c} are not considered so there are 2^v-2 ways for binary split that partitions the dataset DS into two datasets DS_1 and DS_2,and gini index of DS for a given partitioning is computed as:

$$Gini_A(DS) = \frac{|DS_1|}{|DS|} Gini(DS_1) + \frac{|DS_2|}{|DS|} Gini(DS_2) \quad (16)$$

Next we compute reduction in impurity.

$$\Delta Gini(A) = Gini(DS) - Gini_A(DS) \quad (17)$$

The attribute that maximizes the reduction in impurity is selected for splitting.

[1]Gini index is biased towards multivalued attributes. It has problem when number of classes is large. It has a tendency to favor tests that produces equal-sized partitions with purity in both partitions.

3.4 Random Forest Classifier

[4]Random forests or random decision forests are an ensemble learning method for classification, that operate by constructing a number of decision trees at training time and outputting the class that is the majority of the classes outputted from individual trees. Random decision forests correct for decision tree's problem of overfitting to their training data sets. The individual decision trees are created by random selection of tuples from training datasets and random selection of attributes at each split.

Random Forests are built using bagging combined with random selection of attributes.

The training algorithm for random forests applies the general technique of bootstrap aggregating, or bagging, to tree learners. Given a training set X = x_1,.....,x_N with classes Y =c_1,....,c_N, bagging repeatedly (m times) selects a random sample with replacement of the training set and construct trees from these samples:

For i = 1 to m:

1.) Sample, with replacement, N training tuples from X,Y; call these X_i, Y_i.

2.) Build a classification tree say T_i from X_i, Y_i.

After training, predictions for unknown tuples x' can be made by taking the majority vote.

The above method explains the actual bagging algorithm for trees, but random forest uses one more approach, at each split in process of learning, it selects random subset of features, also known as "feature bagging". This is done to reduce correlation between trees if one or a few features are very strong predictors for the class.

An analysis of how bagging and random subspace method improves accuracy under various conditions is given by Ho.
A classification problem, dataset that has p features, $\lfloor\sqrt{p}\rfloor$ features are used in each split.

3.5 Support Vector Machine

[10]Support vector machine SVM is a supervised learning based, non-probabilistic algorithm for classification. A support vector machine constructs a hyperplane which can be used for classification. This algorithm can also work on non-linear classification by mapping data points to a higher dimension by kernel technique. New data instance is then mapped into that same space and predicted to belong to a class based on which side of the gap they fall. Any hyperplane can be written as the set of points $\vec{x}$ satisfying, $\vec{w}.\vec{x}+w_0=0$, where $\vec{w}$ is the weight vector, $\vec{w}$ is normal vector to the hyperplane. SVM is trained using samples from both classes. The points are classified according to following equations:
$\vec{w}.\vec{x}+w_0>0$ for points above hyperplane (18)
$\vec{w}.\vec{x}+w_0<0$ for points below hyperplane (19)

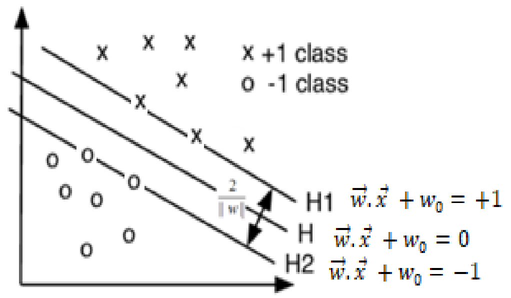
x +1 class
o -1 class
$\frac{2}{\|w\|}$
H1 $\vec{w}.\vec{x}+w_0=+1$
H $\vec{w}.\vec{x}+w_0=0$
H2 $\vec{w}.\vec{x}+w_0=-1$

Fig-3: Diagram representing hyperplanes in SVM

The weights can be adjusted so that the hyperplanes defining the "sides" of the margin can written as:

$H1:\vec{w}.\vec{x}_i+w_0\geq+1$ for $y_i=+1$ (20)

$H2:\vec{w}.\vec{x}_i+w_0\leq-1$ for $y_i=-1$ (21)

Any training tuples that fall on hyperplanes H_1 or H_2 are called support vectors. Combining both the inequalities we get:-
$y_i(\vec{w}.\vec{x}_i+w_0)\geq 1 \quad \forall i$ (22)

Finally optimization technique is used for maximizing $\frac{2}{\|w\|}$ that is distance between H_1 and H_2, this equation can be written in constrained(convex) quadratic optimization problem. Hyperplane with larger margin is found to be more accurate than hyperplane with smaller margin.
Note that $||w||=\sqrt{w^T w}$
Since maximizing $\frac{2}{\|w\|}$ is same as minimizing $\frac{w^T w}{2}$ the quadratic programming problem in primal form becomes:

$$Minimize \ \frac{w^T w}{2} \ , subject \ \ to \ \ y_i(w^T x_i+w_0)\geq 1 \ \ \forall i \tag{23}$$

For solving this problem we have the following lagrangian formulation with α_i as lagranges multiplier where $\alpha_i\geq0$ for all i ϵ {1, 2, ….., N}, N is number of data points:

$$L=\frac{w^T w}{2}-\sum_i \alpha_i(y_i(w^T x_i+w_0)-1) \tag{24}$$

Now we solve $\frac{\partial L}{\partial w}=0$ and $\frac{\partial L}{\partial w_0}=0$ which gives

$$w=\sum_i \alpha_i y_i x_i \tag{25}$$

$$\sum_i \alpha_i y_i=0 \tag{26}$$

The same quadratic programming problem in dual form can be obtained by substituting the values of equation (25) and (26) into L :

$$Maximize \ \ \sum_i \alpha_i-\frac{1}{2}\sum_i\sum_j \alpha_i\alpha_j y_i y_j(x_i\cdot x_j) \ \ , \quad subject \ \ to \ \ \alpha_i\geq0 \ \ and \ \ \sum_i \alpha_i y_i=0 \tag{27}$$

Once α_i has been calculated, the classification function is:

$$h(x)=sign(\sum_i \alpha_i y_i(x_i\cdot x)+w_0) \tag{28}$$

After solving for α_i , we see that most α_i are zero, and only a small percentage of $\alpha_i > 0$. The data points x_i whose $\alpha_i > 0$ are the support vectors. The value of w_0 is calculated using $w_0=y_k-w.x_k$ for any k where $\alpha_k>0$.
How to handle linearly inseparable data?
This involves two main steps:
1.)The original input data points is transformed to a

higher dimension space using a non-linear mapping. 2.)Once the data is transformed the algorithm searches for a linear separating hyperplane in that new space.

As an example consider a mapping from X to Z where X is a three dimensional vector $X = (x_1, x_2, x_3)$,and Z is a six dimensional vector $Z = (\phi_1(X), \phi_2(X), \phi_3(X), \phi_4(X), \phi_5(X), \phi_6(X))$,where $\phi_1(X) = x_1, \phi_2(X) = x_2, \phi_3(X) = x_3, \phi_4(X) = x_1^2, \phi_5(X) = x_1x_2, \phi_6(X) = x_1x_3$.

Now the decision hyperplane becomes d(Z) =WZ+W$_0$, which is linear. As we have seen that in solving quadratic programming optimization problem, the training tuples appear only in form of dot products, $\phi(X_i) \cdot \phi(X_j)$,where $\phi(X)$ is a nonlinear mapping function applied to transform the training tuples.
It is mathematically equivalent to applying kernel function K(X$_i$,X$_j$),to the original input data.

$$K(X_i, X_j) = \phi(X_i) \cdot \phi(X_j) \quad (29)$$

So $\phi(X_i) \cdot \phi(X_j)$ is replaced with K(X$_i$,X$_j$) in our calculation, in this way all calculation are done in original input space which is of potentially of less dimension. Some kernel functions are:
Polynomial Kernel of degree h

$$K(X_i, X_j) = (X_i \cdot X_j + 1)^h \quad (30)$$

Gaussian radial basis function kernel

$$K(X_i, X_j) = e^{\frac{-\|X_i - X_j\|^2}{2\sigma^2}} \quad (31)$$

Sigmoid Kernel

$$K(X_i, X_j) = tanh(\kappa X_i \cdot X_j - \delta) \quad (32)$$

Now the quadratic programming problem in dual form is:

$$\begin{aligned} &Maximize \quad \sum_i \alpha_i - \frac{1}{2}\sum_i \sum_j \alpha_i \alpha_j y_i y_j K(x_i, x_j) \quad , \\ &subject \quad to \quad \alpha_i \geq 0 \quad and \quad \sum_i \alpha_i y_i = 0 \end{aligned} \quad (33)$$

And classification function becomes:

$$h(x) = sign(\sum_i \alpha_i y_i K(x_i, x) + w_0) \quad (34)$$

Now let us consider soft margin SVM :

In this we introduce non-negative slack variables $\xi_i \geq 0$ in the equation :

$$H1: \vec{w}.\vec{x}_i + w_0 \geq +1 - \xi_i \quad \text{for } y_i = 1 \quad (35)$$

$$H2: \vec{w}.\vec{x}_i + w_0 \leq -1 + \xi_i \quad \text{for } y_i = -1 \quad (36)$$

$$y_i(\vec{w}.\vec{x}_i + w_0) \geq 1 - \xi_i \quad (37)$$

Let C denotes the penalty factor, then the quadratic optimization problem becomes:

$$\begin{aligned} &Minimize \quad \frac{w^T w}{2} + C\sum_i \xi_i \quad , \\ &subject \quad to \quad y_i(w^T x_i + w_0) \geq 1 - \xi_i \quad and \quad \xi_i \geq 0 \quad \forall i \end{aligned} \quad (38)$$

And finally, this equation is also solved in similar manner.

[1]The complexity of learned Support Vector Machine classifier is characterized by the number of support vectors rather than the dimensionality of the data.

SVM has applications in hand written digit recognition, identification of speaker, object recognition and text classification.
WEKA uses SMO(Sequential Minimal Optimization) technique for optimization.

3.6 K-Nearest Neighbor

[18]K-nearest neighbor (K-NN) predicts the class label of unknown data instance by taking majority of the class labels of K-nearest data instances, the distance between the instances is measured by distance measure formula. K-NN is a type of instance-based learning, or lazy learning, where all computation are done only when the test tuple is given for classification.

$$dist(X, Y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2} \quad (39)$$

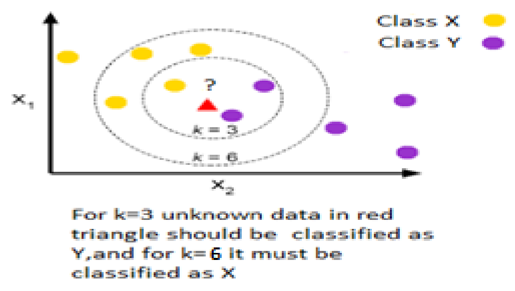
Class X
Class Y
X₁
X₂
?
k = 3
k = 6
For k=3 unknown data in red triangle should be classified as Y,and for k=6 it must be classified as X

Fig-4:[15]Diagram representing K-NN for k=3 and k=6

[1]K-NN is based on learning by analogy, it compares test tuple with training tuples that are similar to it. These tuples are described by n attributes. All training tuples are stored in n-dimensional space. When we are given a test tuple the algorithm searches for k training tuples that are closest to the test tuple, these k tuples are called nearest

neighbors. It then labels this test tuple with the majority of the class labels of these K-nearest data instances. We also normalize value of each attribute, so that large value attribute do not outweights small weight attribute. This is done by min-max normalization.

$$v' = \frac{v - min_A}{max_A - min_A} \tag{40}$$

This normalization equation transforms the value v of attribute A to v' in range[0,1].

Now we see how to handle distance computation case for attribute value that is not numeric but categorical, such as color, in this case we take difference between different color to be 1 and same color to be 0. Another case is how to handle missing values. In this maximum possible difference is assumed. For nominal attributes, if either one or both of the values are missing, then difference is taken to be 1. For a numeric attribute with normalized values, if both values are missing, then the difference is taken to be 1. If only one value is missing and the other is v'(normalized) then difference is taken to be max(|1-v'|,|0-v'|).

The value of k is determined practically beginning from k=1 and finally incrementing k. The value of k that gives minimum error rate is selected. Since distance computation has assigned equal weight to each attribute, this algorithm has poor accuracy when provided with noisy or useless attributes.
So, this method has been modified to include attribute weighting, and the pruning of noisy data tuples.
It is used in face recognition.

4. EVALUATION METRICS

[1][16]For evaluation we split the data set into two sets, training dataset and test dataset. We use training data set to build the classifier and then we use this classifier on test data set for prediction. This splitting is normally 80% training dataset and 20% test dataset.

Confusion Matrix

Actual\Predicted	C	¬C
C	TP	FN
¬C	FP	TN

TP: True Positive, FN: False Negative, FP: False Positive TN: True Negative

The evaluation metrics are, Accuracy, Precision, Recall, F1-Score, and Time.

Accuracy: It is percentage of test tuples that are correctly classified.

$$Accuracy = \frac{TP+TN}{Total\ tuples\ in\ test\ dataset} \tag{41}$$

Precision: It is the measure of exactness, which is the percentage of tuples in test dataset that are labeled as positive, are actually positive.

$$Precision = \frac{TP}{TP+FP} \tag{42}$$

Recall: It is the measure of completeness, which is the percentage of positive tuples in test dataset that the classifier labeled as positive.

$$Recall = \frac{TP}{TP+FN} \tag{43}$$

F1-Score: It is the harmonic mean of precision and recall.

$$F1 - Score = \frac{2*Precision*Recall}{Precision+Recall} \tag{44}$$

Time: Time taken in building model and prediction.

Algorithm that has high accuracy, precision, recall, F1-score, and takes least time is considered to be better than others.

5. WEKA

[5][17]Waikato Environment for Knowledge Analysis (WEKA) is a machine learning tool written in Java, developed at the University of Waikato. It is free software licensed under the GNU General Public License.
WEKA provides many data mining algorithms and visualization tools for data analysis and predictive modeling, with graphical user interfaces that helps user to easily run these algorithms on datasets. WEKA supports several standard data mining tasks, that are, data preprocessing, classification, regression, clustering, feature selection, and visualization. The classification algorithms that we will evaluate here are: Decision Tree (C 4.5) by J48,Naive Bayes, Random Forest, Multilayer Neural Network by Multilayer Perceptron, SVM by SMO(PolyKernel,Normalized PolyKernel), and K-Nearest neighbor by Instance based learning(IBk). WEKA outputs accuracy. It also outputs precision, recall, and F1-Score of all classes separately and combined weighted average precision, weighted average recall ,and weighted average F1-Score of all classes. The input datasets are provided in ARFF file format. ARFF stands for Attribute-Relation File Format.

6. DATASETS TO BE USED IN EVALUATION

1.)[11]Iris Dataset: This dataset has 4 attributes and 3 classes,150 instances ,attributes are: sepallength REAL, sepalwidth REAL, petallength REAL,petalwidth REAL, and classes are :Iris-setosa,Iris-versicolor,Iris-virginica.

2.)[12]Ionosphere Dataset:This dataset has 34 attributes and 2 classes,351 instances.Attributes are: a01 numeric, a02 numeric,…………., a34 numeric,and classes are b,g.

3.)[13]Soybean disease dataset:This dataset has 35 attributes ,and 19 classes,683 instances.Attributes are:date,plant-stand,precip,temp,hail,crop-hist,area-damaged,severity ,seed-tmt,germination,plant-growth ,leaves.leafspots-halo ,leafspots-marg,leafspot-size,leaf-shread ,leaf-malf ,leaf-mild ,stem ,lodging ,stem-cankers ,canker-lesion ,fruiting-bodies ,external-decay ,mycelium int-discolor ,sclerotia ,fruit-pods,fruit-spots,seed ,mold-growth ,seed-discolor,seed-size ,shriveling ,roots and classes are: diaporthe-stem-canker, charcoal-rot, rhizoctonia-root-rot, phytophthora-rot, brown-stem-rot, powdery-mildew, downy-mildew, brown-spot, bacterial-blight, bacterial-pustule, purple-seed-stain, anthracnose, phyllosticta-leaf-spot, alternarialeaf-spot, frog-eye-leaf-spot, diaporthe-pod-&-stem-blight, cyst-nematode, 2-4-d-injury, herbicide-injury

4.)[14]Labor Dataset:This dataset has 16 attributes and 2 classes,57 instances.Attributes areduration numeric, wage-increase-first-year numeric,wage-increase-second-year numeric,wage-increase-third-year numeric,cost-of-living-adjustment string, working-hours numeric,pension string, standby-pay numeric, shift-differential numeric, education-allowance string,statutory-holidays numeric, vacation string,longterm-disability-assistance string, contribution-to-dental-plan string,bereavement-assistance string,contribution-to-health-plan string,and classes are: bad,good

7. RESULT OF EVALUATION FROM WEKA

Iris Dataset

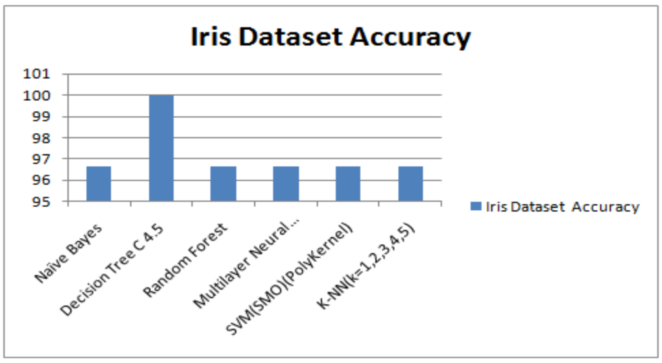

Chart-1: Iris dataset accuracy

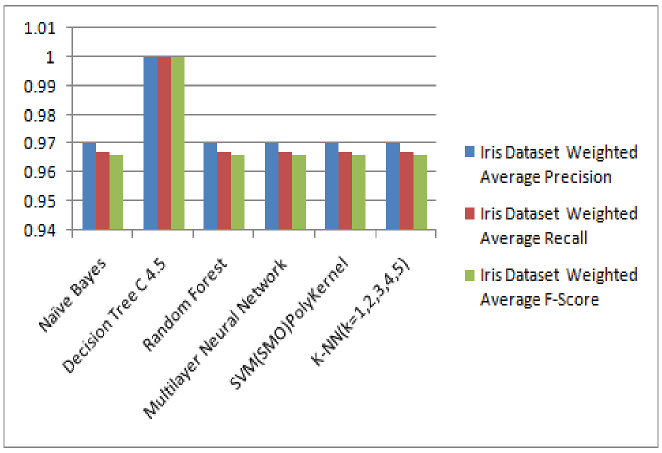

Chart-2: Iris dataset weighted average precision, weighted average recall, and weighted average F-score

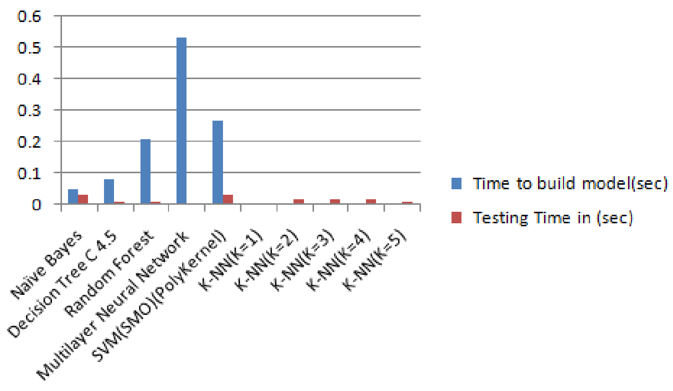

Chart-3: Iris dataset time

Ionosphere Dataset

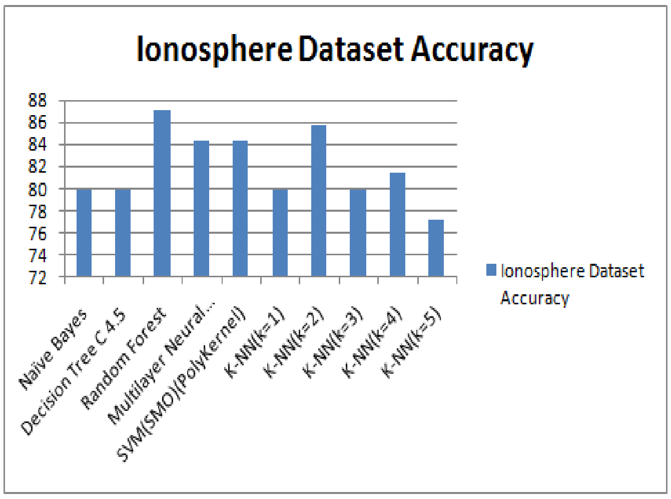

Chart-4: Ionsophere dataset accuracy

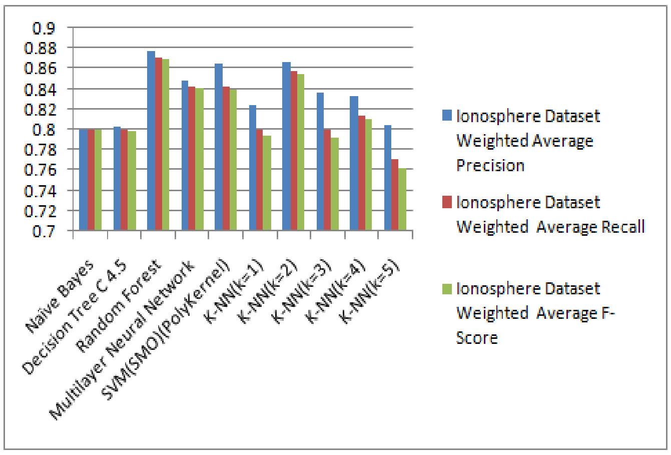

Chart-5: Ionosphere dataset weighted average precision, weighted average recall, and weighted average F-score

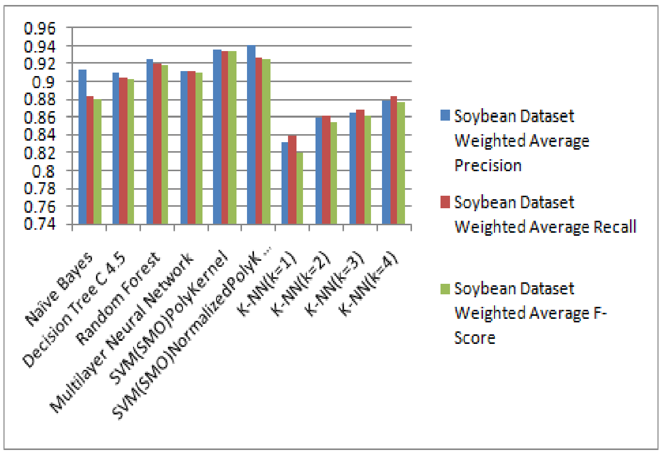

Chart-8: Soybean dataset weighted average precision, weighted average recall,and weighted average F-score

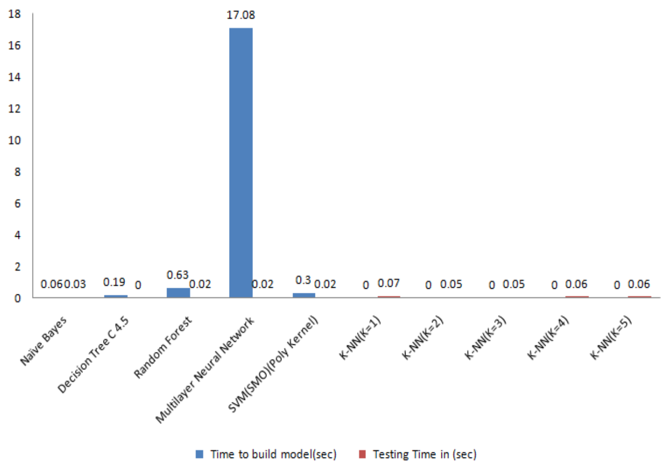

Chart-6: Ionosphere dataset time

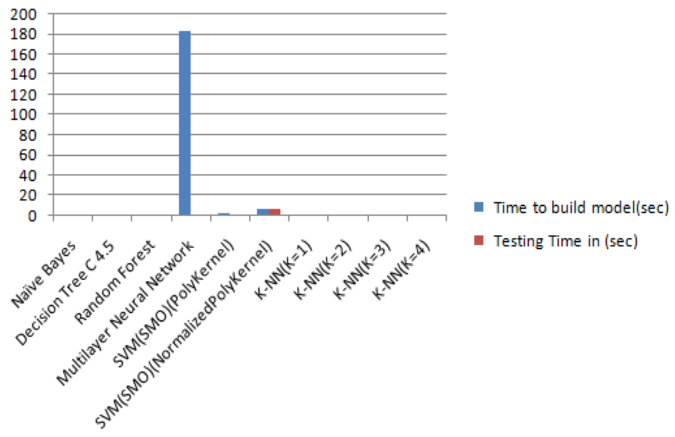

Chart-9: Soybean dataset time

Labor Dataset

Soybean dataset

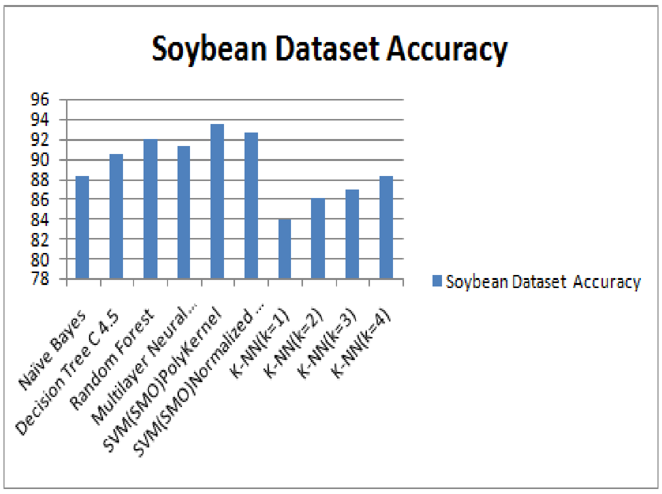

Chart-7: Soybean dataset accuracy

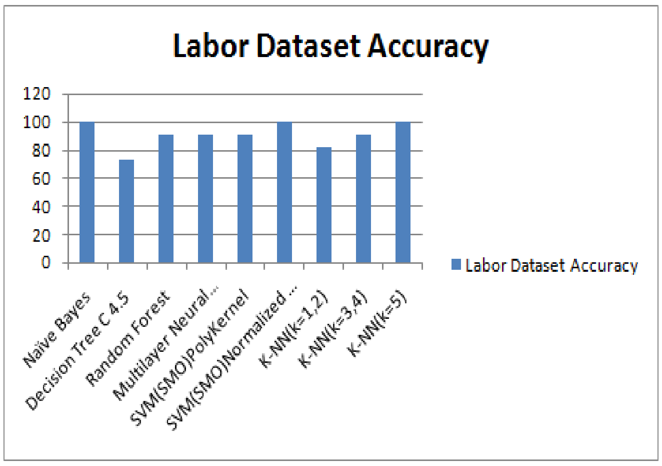

Chart-10: Labor dataset accuracy

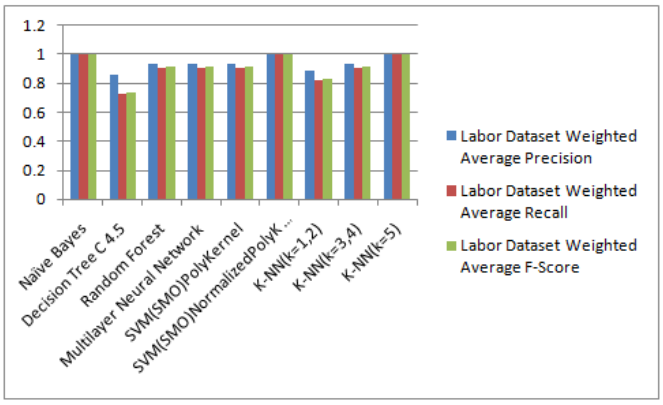

Chart-11: Labor dataset weighted average precision, weighted average recall,and weighted average F-score

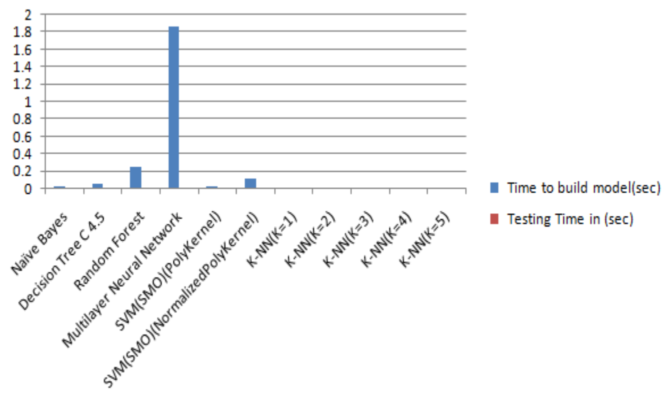

Chart-12: Labor dataset time

8. IMPORTANT POINTS FROM EVALUATION RESULTS

1.) From result of evaluation of iris dataset we see that Decision Tree C4.5 has better accuracy, precision, recall ,and F1-score compared to others.Chart-1,Chart-2

2.) From result of evaluation of ionosphere dataset we see that Random Forest has better accuracy, precision, recall , and F1-score compared to others.Chart-4,Chart-5

3.) From result of evaluation of soybean dataset we see that SVM(SMO)with PolyKernel has better accuracy , recall ,and F1-score compared to others.Chart-7,Chart-8

4.) From result of evaluation of labor dataset we see that Naive Bayes, SVM(SMO)with Normalized PolyKernel, and K-NN with k=5 has better accuracy ,precision ,recall , and F1-score compared to others.Chart-10,Chart-11

5.)From Chart-3,Chart-6,Chart-9, and Chart-12 we see that neural network takes the most time in each case.

9. CONCLUSION

From this paper we have seen that classification algorithm has a wide use case from lot of available datasets. It is a supervised learning technique. Each algorithm has its own pros and cons. Naive Bayes performs well if there is a conditional independence between attributes, which is not always the case, and takes less time. Multilayer neural network works well with numeric data and has ability to handle noisy data, but takes most time. Decision tree provides understandability in representing classification model and takes less time. Random forest improves accuracy of decision tree by correcting decision tree problem of overfitting to their training datasets. In Support vector machine the complexity of the learned classifier is characterized by the number of support vectors rather than the dimensionality of the data, it give better accuracy, and works for linearly inseparable data as well. K-nearest neighbor is a lazy learner and gives poor accuracy in case it is provided with noisy or useless attributes, so this method has been modified to include attribute weighting, and the pruning of noisy data tuples.

REFERENCES

[1] Jiawei Han,Micheline Kamber “Data Mining Concepts and Techniques” , Second Edition

[2] Stuart J. Russell and Peter Norvig , “Artificial Intelligence ,A Modern Approach”, Third Edition

[3] Margaret H Dunham, “Data Mining: Introductory and Advanced Topics”

[4] https://en.wikipedia.org/wiki/Random_forest

[5] https://en.wikipedia.org/wiki/Weka_(machine_learning)

[6] N. Chandra Sekhar Reddy, K. Sai Prasad and A. Mounika,Classification Algorithms in Datamining: A study,Internation Journal of Computational Intelligence Research,2017

[7] Mr.Sudhir, M.Gorade ,Prof. Ankit Deo ,Prof. Preetesh Purohit, A study of some Data Mining Classification Techniques, International Research Journal of Engineering and Technology, IRJET 2017

[8] Nikhil.N Salvithal ,Dr.RB Kulkarni, Evaluating Performance of Data Mining classification Algorithm in WEKA, International Journal of Application or Innovation in Engineering and Management(IJAIEM) 2013

[9] https://www.tutorialspoint.com/data_mining/dm_dti.htm

[10] http://web.cs.iastate.edu/~honavar/smo-svm.pdf

[11] https://archive.ics.uci.edu/ml/datasets/iris

[12] https://archive.ics.uci.edu/ml/datasets/ionosphere

[13] https://archive.ics.uci.edu/ml/datasets/Soybean+(Large)

[14] https://archive.ics.uci.edu/ml/datasets/Labor+Relations

[15] http://adataanalyst.com/machine-learning/knn/

[16] https://en.wikipedia.org/wiki/Precision_and_recall

[17] https://www.cs.waikato.ac.nz/ml/weka/arff.html

[18] https://en.wikipedia.org/wiki/K-nearest_neighbors_algorithm

[19] https://en.wikipedia.org/wiki/Decision_tree_learning

BIOGRAPHY

Archit Verma

I am UGC-NET qualified in Computer Science and Applications in January 2018. M.Tech(C.S.E.) from United College of Engineering and Research, Allahabad in 2017, affiliated to Dr. A.P.J. Abdul Kalam Technical University, Lucknow. B.Tech(C.S.E.) from Institute of Engineering and Rural Technology, Allahabad in 2013, affiliated to Dr. A.P.J. Abdul Kalam Technical University, Lucknow. My area of interest includes Data Mining, Artificial Intelligence, Distributed Systems and Big Data Analytics.