**WINTER RESEARCH INTERNSHIP REPORT**

on

**An Ensembling study of various Classification Algorithms**

Supervised By

Prof. Dhruba K. Bhattacharyya

Professor and Dean

School of Engineering,

Tezpur University

&

Submitted by

Hillul Saikia

3rd Year, Computer Science and Engineering

NIT Silchar



**Department of Computer Science and Engineering**

**Tezpur University**

**Tezpur, India**

Department of

COMPUTER SCIENCE & ENGINEERING

TEZPUR UNIVERSITY

(A central University)

NAPAAM, TEZPUR- 784028

ASSAM: INDIA

CERTIFICATE

This is to certify that Hillul Saikia has completed one month Winter Research Internship programme(December 2017- January 2018), entitled **“An Ensembling study of various Classification Algorithms** “ under my supervision in the Department of Computer Science and Engineering, Tezpur University.

**Prof Dhurba K. Bhattacharyya**

Professor

Department of Computer Science and Engineering

Tezpur University

Tezpur, India

**ACKNOWLEDGEMENT:**

We have taken efforts in the projects. However, it would not have been possible without the kind support and help of varied people. We would like to extend sincere thanks to all of them.

We are highly indebted to our project guide, Prof. D.K.Bhattacharyya, for his guidance as well as for providing necessary information regarding the project & also for his support in completing the project. Without his also guidance, the project could not have taken its shape.

We would like to express special gratitude and thanks to various faculty members of the Department of Computer Science and Engineering, for giving us such motivation, attention and time.

Our thanks and an appreciation also goes to the people who have willingly helped us out with their abilities

HILLUL SAIKIA

3rd Year, Computer Science & Engineering

NIT Silchar

**ABSTRACT:**

The role of Classifier is significant. Classifier selection methods have been used in various applications of machine learning, bio-informatics, pattern recognition and network traffic analysis. In high dimensional datasets, a learning method consumes a significant amount of time and reduces the performance of the model. To overcome these problems, we compare the classifier first on the basis of accuracy and then on the basis of execution time. Classifier selected for datasets are unstable in nature i.e., for different datasets, different classifier selected gives different accuracy.

In this paper we first study the different classifier in machine learning. Then, we compare 5 different classifiers (KNN, J48, Random Forest, SVM and Naive Bayes) on the basis of accuracy and execution time. For Comparing we use 10 datasets. Different classifier gives different accuracy, for a specified datasets. As such, we ensemble all the five classifier and study its accuracy and execution time.

**TABLE OF CONTENTS**

**Title Page**

1. Introduction 1

1.1 Types of Algorithm in ML

1.1.1 Supervised Learning 3

* + 1. Unsupervised Learning 7
    2. Ensemble learning techniques 10

1. Review of Literature 12
2. Experimental Analysis 14

3.1 Comparison among Random Forest, J48, IBk, 14

SMO, Naïve Bayes

3.2 Datasets Used 14

3.3 Accuracy Measurement 14

* 1. Execution Time Measurement 19

3.5 Average Accuracy Measurement 20

4. Result Analysis 21

5. Conclusion and Future Work 22

Reference 23

1. **INTRODUTION 1**

Machine learning is an application of artificial intelligence (AI) that provides systems the ability to automatically learn and improve from experience without being explicitly programmed**. Machine learning focuses on the development of computer programs** that can access data and use it learn for themselves.

These days, machine learning algorithms deal with large amount of data. These data contain tens of thousands of instances, each instance being represented by hundreds of features. While dealing with large datasets, it is often all the classifier has different accuracy for specified dataset. So, it is utmost important to study the all the algorithm and find the best classifier on the basis of accuracy and resolution time.

**1.1 Types of Algorithms in ML**

Machine Learning algorithms come in (at least) three major flavors:

**A.** **Supervised learning**:

Supervised learning can be explained as follows: use labeled training data to learn the mapping function from the input variables (X) to the output variable (Y).

*Y = f (X)*

Supervised learning problems can be of two types:

*a.* **Classification**: To predict the outcome of a given sample where the output variable is in the form of categories. Examples include labels such as male and female, sick and healthy.

*b***. Regression**: To predict the outcome of a given sample where the output variable is in the form of real values. Examples include real-valued labels denoting the amount of rainfall, the height of a person.

The 1st 5 algorithms that we cover in this blog– Linear Regression, Logistic Regression, CART, Naïve Bayes, KNN are examples of supervised learning.

**2**

Ensembling is a type of supervised learning. It means combining the predictions of multiple different weak ML models to predict on a new sample. Algorithms 9-10 that we cover– Bagging with Random Forests, Boosting with XGBoost are examples of ensemble techniques

**B. Unsupervised learning**:

Unsupervised learning problems possess only the input variables (X) but no corresponding output variables. It uses unlabeled training data to model the underlying structure of the data.

Unsupervised learning problems can be of two types:

**a. Association**: To discover the probability of the co-occurrence of items in a collection. It is extensively used in market-basket analysis. Example: If a customer purchases bread, he is 80% likely to also purchase eggs.

**b. Clustering**: To group samples such that objects within the same cluster are more similar to each other than to the objects from another cluster.

**c. Dimensionality Reduction**: True to its name, Dimensionality Reduction means reducing the number of variables of a dataset while ensuring that important information is still conveyed. Dimensionality Reduction can be done using Feature Extraction methods and Feature Selection methods. Feature Selection selects a subset of the original variables. Feature Extraction performs data transformation from a high-dimensional space to a low-dimensional space. Example: PCA algorithm is a Feature Extraction approach.

Algorithms 6-8 that we cover here - Apriori, K-means, PCA are examples of unsupervised learning.

**C. Reinforcement learning:**

Reinforcement learning is a type of machine learning algorithm that allows the agent to decide the best next action based on its current state, by learning behaviors that will maximize the reward.

**3**

Reinforcement algorithms usually learn optimal actions through trial and error. They are typically used in robotics – where a robot can learn to avoid collisions by receiving negative feedback after bumping into obstacles, and in video games – where trial and error reveals specific movements that can shoot up a player’s rewards. The agent can then use these rewards to understand the optimal state of game play and choose the next action.

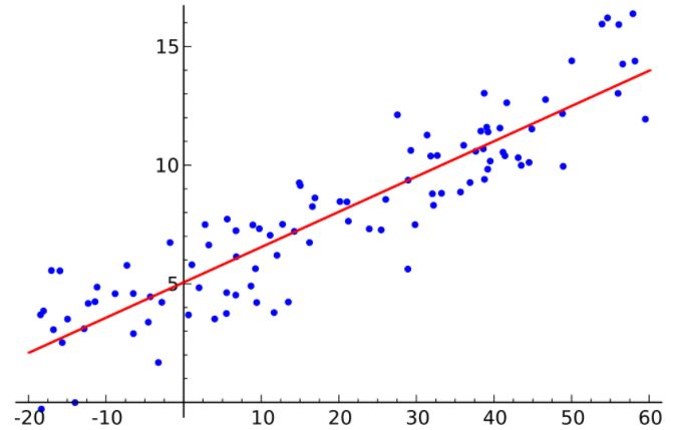
**1.1.1. Supervised learning algorithms**

**1.1.1.1 Linear Regression**

Linear regression is a statistical modelling technique, which attempts to model the relationship between an explanatory variable and a dependent variable by fitting the observed data points on a linear equation. For eg: Modelling the BMI of individuals using weight.

A linear regression is used if there is relationship or significant association between the variables. This can be checked by scatterplots. If no association appears between the variables, fitting a linear regression model to the data will not provide useful model.

A linear regression line has equation in the following form:

Y = a + bX,

Where, X = explanatory variable and

Y = dependent variable.

b = slope of the line

a = intercept (the value of y when x = 0).

Thus, the goal of linear regression is to

find out the values of coefficients a and b.

Here, a is the intercept and b is the slope Figure (1.a)

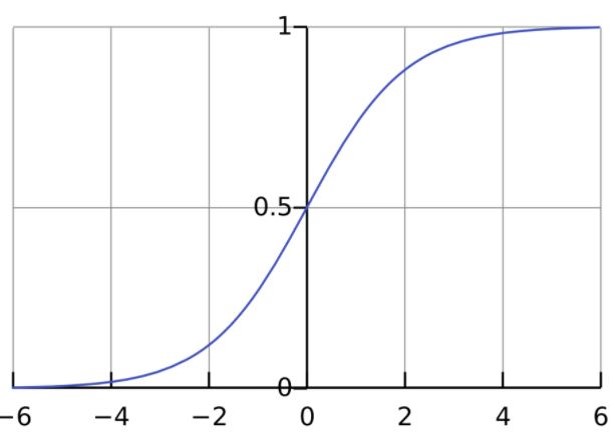
of the line.

The goal is to fit a line that is nearest to most of the points. This would reduce the distance (‘error’) between the y value of a data point and the line.

**4**

**1.1.1.2. Logistic Regression**

Logistic regression is the technique to find relationship between a set of input variables and a output variable (just like any regression) but the output variable in this case would be a binary outcome (think of 0/1 or yes/no).For eg: Will there be traffic jam in a certain location in Bangalore is a binary variable. The output is a categorical Yes or no.

The probability of occurrence of traffic jam can be dependent on attributes like weather condition, day of week and month, time of day, number of vehicles etc. Using logistic regression, we can find the best fitting model that explains the relationship between independent attributes and traffic jam occurrence rates and predicts probability of jam occurrence.

The logistic regression equation

*P(x) = e ^ (b0 +b1\*x) / (1 + e^(b0 + b1\*x))*

 can be transformed into *ln(p(x) / 1-p(x))*

*= b0 + b1\*x*.The goal of logistic regression

is to use the training data to find the

values of coefficients b0 and b1 such

that it will minimize the error between the (Figure 1.b)

predicted outcome and the actual outcome. These coefficients are estimated using the technique of Maximum Likelihood Estimation.

**1.1.1.3. CART**

Classification and Regression Trees (CART) is an implementation of Decision Trees, among others such as ID3, C4.5.

The non-terminal nodes are the root node and the internal node. The terminal nodes are the leaf nodes. Each non-terminal node represents a single input variable (x) and a splitting point on that variable; the leaf nodes represent the output variable (y). The model is used as follows to make predictions: walk the splits of the tree to arrive at a leaf node and output the value present at the leaf node.

**5**

The decision tree in Figure3 classifies whether a person will buy a sports car or a minivan depending on their age and marital status. If the person is over 30 years and is not married, we walk the tree as follows : ‘over 30 years?’ -> yes -> ’married?’ -> no. Hence, the model outputs a sportscar.

**1.1.1.4. Naive Bayes Classifier**

To calculate the probability that an event will occur, given that another event has already occurred, we use Bayes’ Theorem. To calculate the probability of an outcome given the value of some variable, that is, to calculate the probability of a hypothesis(h) being true, given our prior knowledge(d), we use Bayes’ Theorem as follows:

*P(h|d)= (P(d|h) \* P(h)) / P(d)*

where

* P(h|d) = Posterior probability. The probability of hypothesis h being true, given the data d, where P(h|d)= P(d1| h)\* P(d2| h)\*....\*P(dn| h)\* P(d)
* P(d|h) = Likelihood. The probability of data d given that the hypothesis h was true.
* P(h) = Class prior probability. The probability of hypothesis h being true (irrespective of the data)
* P(d) = Predictor prior probability. Probability of the data (irrespective of the hypothesis)

This algorithm is called ‘naive’ because it assumes that all the variables are independent of each other, which is a naive assumption to make in real-world examples.

* + - 1. **SVM**

“Support Vector Machine” (SVM) is a supervised machine learning algorithm which can be used for both classification or regression challenges. However, it is mostly used in classification problems. In this algorithm, we plot each data item as a point in n-dimensional space (where

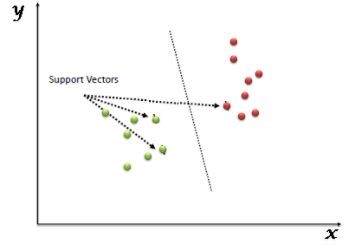
**6**

n is number of features you have) with the value of each feature being the value of a particular coordinate. Then, we perform classification by finding the hyper-plane that differentiate the two classes very well (look at the below snapshot). **Identify the right hyper-plane.**Here, we have three

hyper-planes (A, B and C). Now, identify the right hyper-plane to classify star and circle.**(Scenario-1):** You need to remember a thumb rule to identify the right hyper-plane : “ Select the hyper-plane which segregates the two classes better”. In this scenario, hyper-plane “B” has excellently performed this job.

**(Scenario-2):** Here, maximizing the distances between nearest data point (either class) and hyper-plane will help us to decide the right hyper-plane. This distance is called as **Margin**. Let’s look at the below snapshot:

Above, you can see that the margin for hyper-plane C is high as compared to both A and B. Hence, we name the right hyper-plane as C. Another lightning reason for selecting the hyper-plane with higher margin is robustness. If we select a hyper-plane having low margin then there is high chance of miss-classification.



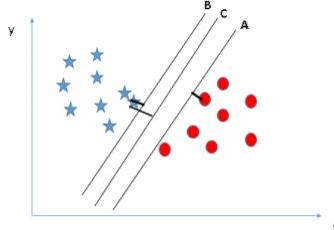
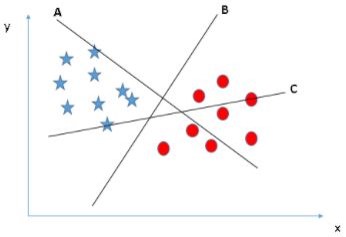


Fig 1.c.1 Fig 1.c.2 Fig 1.c.3

**1.1.1.6. KNN Classifier**

The k-nearest neighbors algorithm uses the entire dataset as the training set, rather than splitting the dataset into a training set and test set.

When an outcome is required for a new data instance, the KNN algorithm goes through the entire dataset to find the k-nearest instances to the new instance, or the k number of instances most similar to the new record, and

**7**

then outputs the mean of the outcomes (for a regression problem) or the mode (most frequent class) for a classification problem. The value of k is user-specified. The similarity between instances is calculated using measures such as Euclidean distance and Hamming distance.

**1.1.2. Unsupervised Learning Algorithm**

**1.1.2.1. Apriori**

The Apriori algorithm is used in a transactional database to mine frequent item sets and then generate association rules. It is popularly used in market basket analysis, where one checks for combinations of products that frequently co-occur in the database. In general, we write the association rule for ‘if a person purchases item X, then he purchases item Y’ as : X -> Y.

Example: if a person purchases milk and sugar, then he is likely to purchase coffee powder. This could be written in the form of an association rule as: {milk, sugar} -> coffee powder. Association rules are generated after crossing the threshold for support and confidence. The Support measure helps prune the number of candidate itemsets to be considered during frequent item set generation. This support measure is guided by the Apriori

principle. The Apriori principle states that if an itemset is frequent, then all of its subsets must also be frequent.

**1.1.2.2. K-means Algorithm (Clustering)**

K-means is an iterative algorithm that groups similar data into clusters. It calculates the centroids of k clusters and assigns a data point to that cluster having least distance between its centroid and the data point. Randomly initialize K cluster centroids u1,u2,u3,..,, um ϵ Rn

Repeat {

for i =1to m, c(i) :=index(from 1 to K) of cluster centroid closest to x(i) [minimize J(…)w.r.t c1,c2,c3…cm holding u1,u2,u3,..,, um fixed Where c1,c2,c3…cm index of cluster, m is the number of objects u1,u2,u3,..,, um  cluster centroid of cluster]

For i = 1 to K

**8**

uk :=average (mean) of points

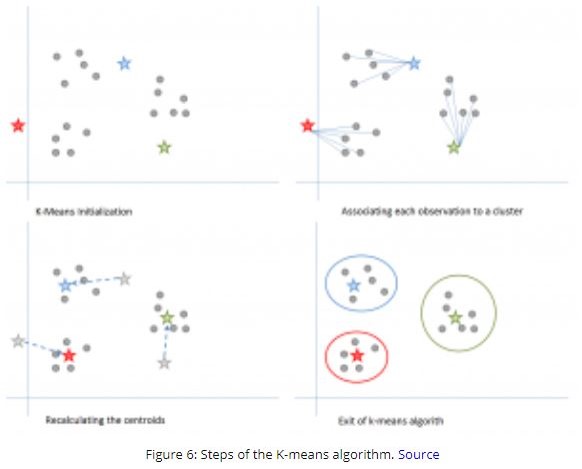
assigned to cluster

k=index of cluster

K=number of cluster

[minimize J(…)w.r.t u1,u2,u3,

.., um]}



****

**1.1.2.3. PCA**

The main idea of principal component analysis (PCA) is to reduce the dimensionality of a data set consisting of many variables correlated with each other, either heavily or lightly, while retaining the variation present in the dataset, up to the maximum extent. The same is done by transforming the variables to a new set of variables, which are known as the principal components (or simply, the PCs) and are orthogonal, ordered such that the

**9**

retention of variation present in the original variables decreases as we move down in the order. So, in this way, the 1st principal component retains maximum variation that was present in the original components. The principal components are the eigenvectors of a covariance matrix, and hence they are orthogonal.

Importantly, the dataset on which PCA technique is to be used must be scaled. The results are also sensitive to the relative scaling. As a layman, it is a method of summarizing data. Imagine some wine bottles on a dining table. Each wine is described by its attributes like color, strength, age, etc. But redundancy will arise because many of them will measure related properties.

So what PCA will do in this case is summarize each wine in the stock with less characteristics. Intuitively, Principal Component Analysis can supply the user with a lower-dimensional picture, a projection or "shadow" of this object when viewed from its most informative viewpoint.

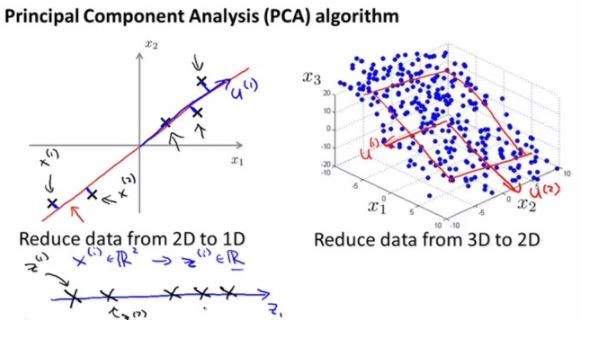
****

Figure 1.e

**1.1.3 Ensemble learning techniques 10**

Ensembling means combining the results of multiple learners (classifiers) for improved results, by voting or averaging. Voting is used during classification and averaging is used during regression. The idea is that ensembles of learners perform better than single learners.

There are 3 types of ensembling algorithms: Bagging, Boosting and Stacking. We are not going to cover ‘stacking’ here, but if you’d like a detailed explanation of it, let me know in the comments section below, and I can write a separate blog on it.

**1.1.3.1. Bagging with Random Forests**

Random Forest (multiple learners) is an improvement over bagged decision trees (a single learner).

Bagging: The first step in bagging is to create multiple models with datasets created using the Bootstrap Sampling method. In Bootstrap Sampling, each generated training set is composed of random subsamples from the original dataset. Each of these training sets is of the same size as the original dataset, but some records repeat multiple times and some records do not appear at all. Then, the entire original dataset is used as the test set. Thus, if the size of the original dataset is N, then the size of each generated training set is also N, with the number of unique records being about (2N/3); the size of the test set is also N.

The second step in bagging is to create multiple models by using the same algorithm on the different generated training sets. In this case, let us discuss Random Forest. Unlike a decision tree, where each node is split on the best feature that minimizes error, in random forests, we choose a random selection of features for constructing the best split. The reason for randomness is: even with bagging, when decision trees choose a best feature to split on, they end up with similar structure and correlated predictions. But bagging after splitting on a random subset of features means less correlation among predictions from subtrees.

**11**

The number of features to be searched at each split point is specified as a parameter to the random forest algorithm.

Thus, in bagging with Random Forest, each tree is constructed using a random sample of records and each split is constructed using a random sample of predictors.

**1.1.3.2 Boosting with AdaBoost**

a) Bagging is a parallel ensemble because each model is built independently. On the other hand, boosting is a sequential ensemble where each model is built based on correcting the misclassifications of the previous model.

b) Bagging mostly involves ‘simple voting’, where each classifier votes to obtain a final outcome– one that is determined by the majority of the parallel models; boosting involves ‘weighted voting’, where each classifier votes to obtain a final outcome which is determined by the majority– but the sequential models were built by assigning greater weights to misclassified instances of the previous models.

Adaboost stands for Adaptive Boosting.

1. **REVIEW OF LITERATURE 12**

Perhaps one of the earliest work on ensemble systems is Dasarathy and Sheela’s 1979 paper (Dasarathy 1979), which first proposed using an ensemble system in a divide-and-conquer fashion, partitioning the feature space using two or more classifiers. Over a decade later, Hansen and Salamon (Hansen 1990) showed the variance reduction property of an ensemble system, and that the generalization performance of a neural network can be improved by using an ensemble of similarly configured neural networks. But it was Schapire's work that put the ensemble systems at the center of machine learning research, as he proved that a strong classifier in [probably approximately correct](http://www.scholarpedia.org/w/index.php?title=Probably_approximately_correct&action=edit&redlink=1) (PAC) sense can be generated by combining weak classifiers through a procedure he called [boosting](http://www.scholarpedia.org/w/index.php?title=Boosting&action=edit&redlink=1), (Schapire 1990). Boosting was the predecessor of the AdaBoost family of algorithms - which arguably became one of the most popular machine learning algorithms in recent times. Since these seminal works, research in ensemble systems have expanded rapidly, appearing often in the literature under many creative names and ideas. The long list includes composite classifier systems (Dasarathy 1979), mixture of experts (Jacobs 1991, Jordan 1994), stacked generalization (Wolpert 1992), combination of multiple classifiers (Ho 1994, Rogova 1994, Lam 1995, Woods 1997), dynamic classifier selection (Woods 1997), classifier fusion (Cho 1995, Kuncheva 2001), classifier ensembles, among many others. These approaches usually differ from each other primarily in two ways: i) specific procedure used for generating individual classifiers; and/or ii) the strategy employed for combining the classifiers. Ensemble systems can also be categorized based on whether classifiers are selected or fused (Woods 1997, Kuncheva 2001, Kuncheva 2002, Ho 2000): In *classifier selection*, each classifier is trained to become an expert in some local area of the feature space (as in Figure 3). The combination of the classifiers is then based on the given instance: the classifier trained with data closest to the vicinity of the instance, according to some distance metric, is given the highest credit. One or more local experts can be nominated to make the decision (Jacobs 1991, Woods 1997, Alpaydin 1996, Giacinto 2001).

**13**

In *classifier fusion*, all classifiers are trained over the entire feature space. In this case, the classifier combination involves merging the individual (usually weaker and/or diverse) classifiers to obtain a single (stronger) expert of superior performance. Examples of this approach include bagging predictors (Breiman 1996), boosting (Schapire 1990), AdaBoost (Freund 2001) and their many variations. The combination may apply to classification labels only, or to the class-specific continuous valued outputs of the individual experts (Kittler 1998). In the latter case, classifier outputs are often normalized to the [0, 1] interval, and these values are interpreted as the support given by the classifier to each class, or as class-conditional posterior probabilities. Such interpretation allows forming an ensemble through algebraic combination rules (majority voting, maximum / minimum / sum / product or other combinations of posterior probabilities) (Kittler 1998, Kuncheva 2002, Roli 2002), fuzzy integral (Cho 1995), the Dempster-Shafer based fusion (Rogova 1994), and more recently, the decision templates (Kuncheva 2001). A sample of the immense literature on classifier combination can be found in Kuncheva’s recent book (Kuncheva 2005), the first text devoted to theory and implementation of ensemble based classifiers, and references therein. Two recent tutorials written by the current curator of this article also provide a comprehensive overview of ensemble systems (Polikar 2006, Polikar 2007).

1. **Experimental Analysis 14**

**3.1. Comparison among IBk, J48, Random Forest, SVM, Naïve Bayes by Using Weka**

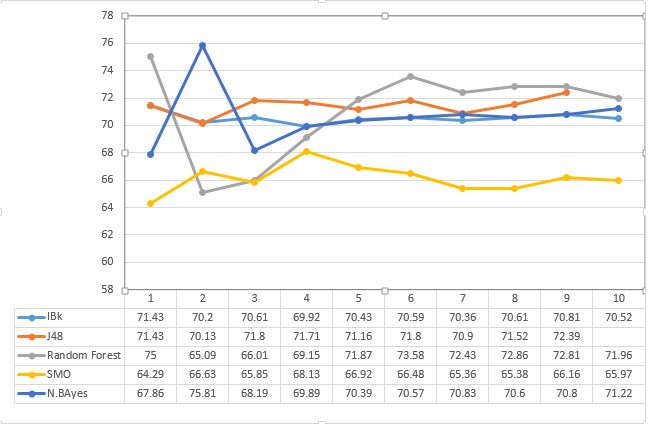
Here, for comparison we have consider classifier of different types. We have consider decision tree classifiers i.e. Random Forest, J48. Probability classifier Naïve Bayes is also considered. SVM is a functional classifier and KNN is a lazy classifier.

**3.2 Dataset Used:** 10 Datasets from weka datasets Software has been selected. The 10 datasets are breast cancer, diabetes, ionospheres**,** iris data, soybean, supermarket, labor, glass, unbalanced, vote

**3.3. Accuracy Measurement**

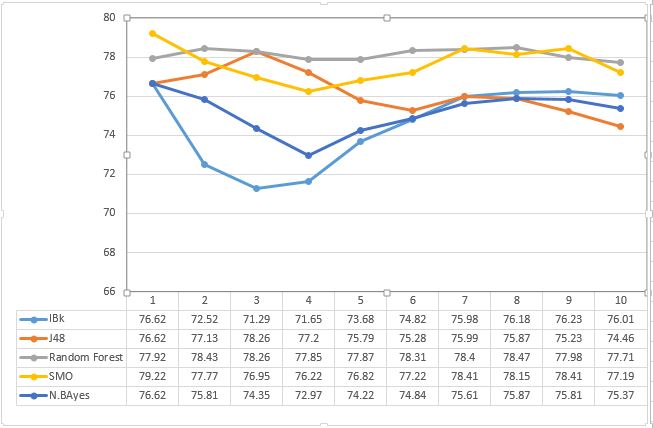
For accuracy comparison, Table and Graphs Made from Different Data Analysis are as Follows taking repetition as independent variable.

**Breast-Cancer**: Both Random Forest and J48 gives maximum accuracy partially.

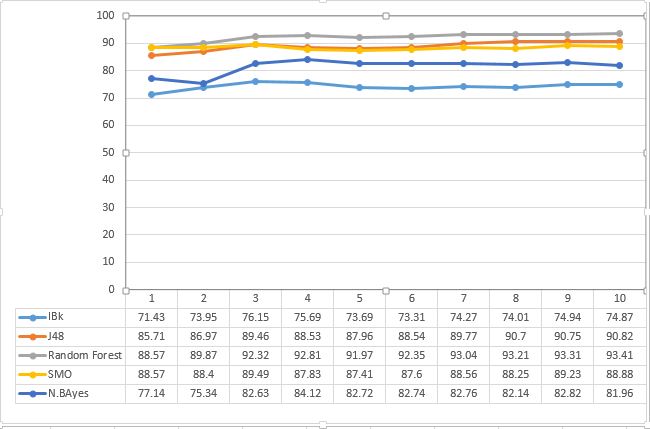


**15**

**Diabetes: Random Forest has maximum accuracy**

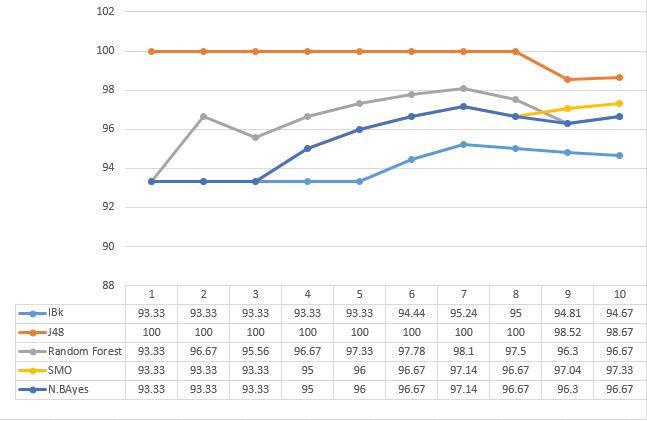


**Ionospheres: Random Forest has maximum accuracy**

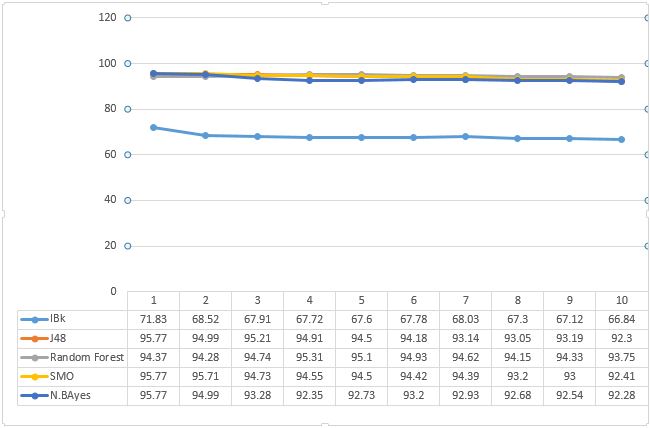


**16**

**Iris Data: J48 has maximum accuracy**

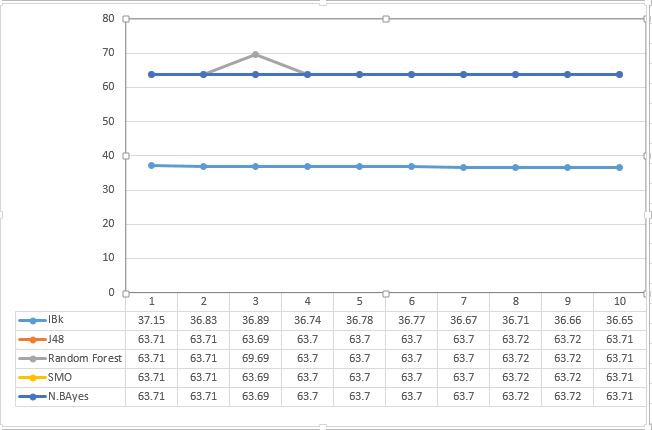
****

**Soybean: J48, Random Forest and SMO has maximum accuracy**

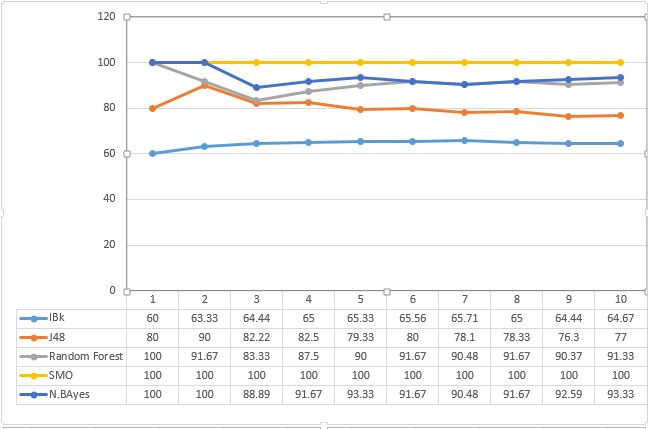
****

**17**

**Supermarket: Both Random Forest and J48 has maximum accuracy in partial instances**

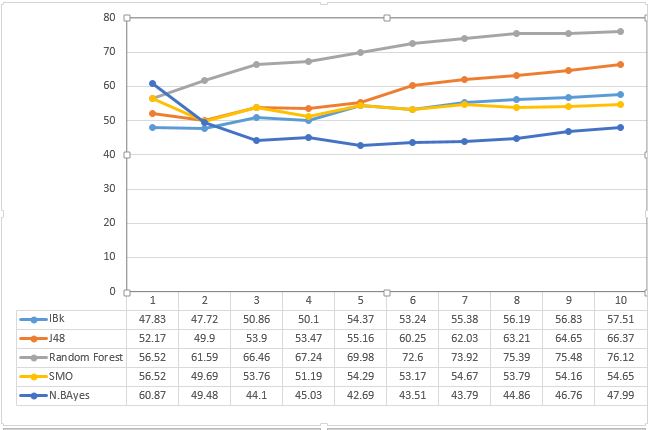


**Labor: SMO has the maximum accuracy**

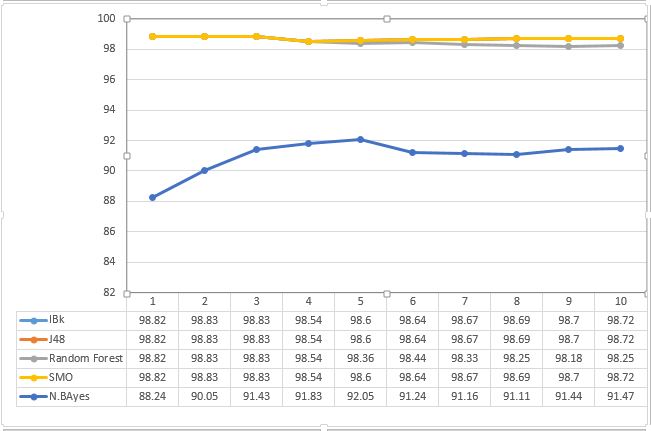
 

**18**

**Glass: Random forest has maximum accuracy.**

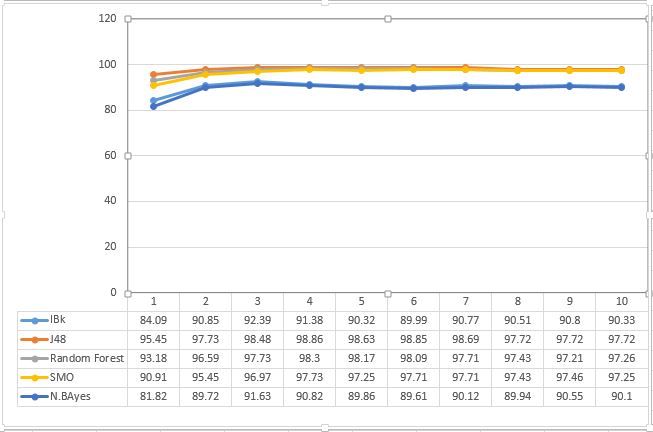
****

**Unbalanced: J48 and SMO has maximum accuracy**

****

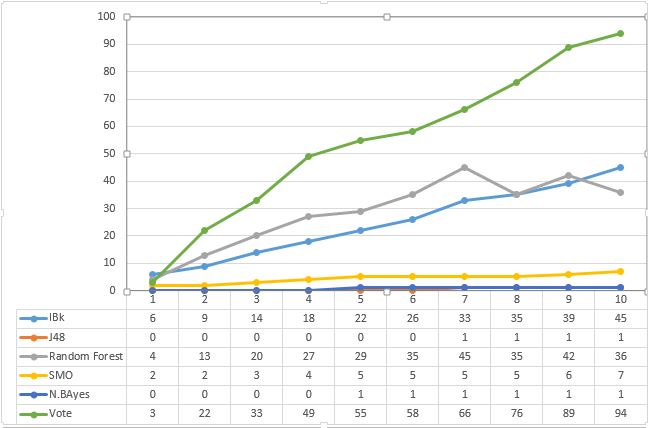
**19**

**Vote: J48 has maximum accuracy**

****

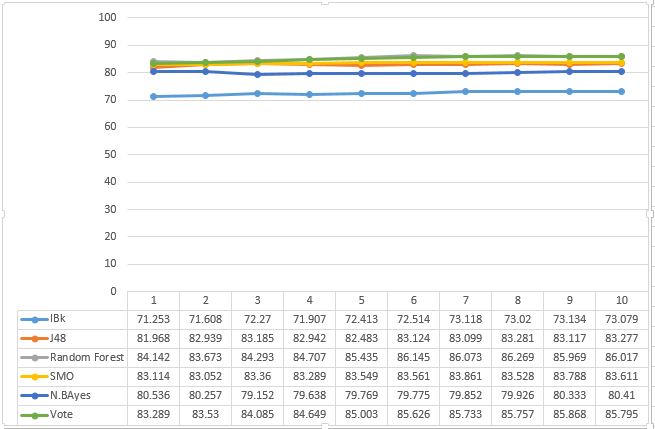
**3.4. Execution Time**

J48 takes least time

****

**20**

**3.5. Average Accuracy:**

****For different datasets, different classifier has different maximum accuracy. So average accuracy is found, where J48 and Random Forest outperforms the IBk, SMO and Naïve Bayes algorithm. Between Random Forest and J48, Random Forest leads by small margin. ****

1. **RESULT ANALYSIS 21**

The result obtained from the graphs are as follows.

* For different datasets, different classifier gives different accuracy.
* For comparison we have consider classifiers of different types. We have consider decision tree classifiers i.e. Random Forest, J48. Probability classifier Naïve Bayes is also considered. SVM is a functional classifier and KNN is a lazy classifier.
* None of classifier is found to be absolute winner. It has been seen that different classifier dominates in the maximum accuracy. For Example: Random Forest dominates Diabetes, Ionospheres, Glass datasets.

J48 dominates in the Glass and Vote datasets. SMO dominates in the case of Labor datasets. In the remaining cases different classifiers partially have the maximum accuracy.

* On comparing the maximum accuracy among J48 and Random Forest outperforms the IBk, SMO and Naïve Bayes algorithm. Between Random Forest and J48, Random Forest leads by small margin.

So, it can be said random forest is the best classifier on the basis of accuracy.

* Execution time plays a crucial role. If the execution time is high then it not a good classifier. Taking execution time into consideration, J48 outperforms the remaining classifier.
* Taking accuracy and execution time in consideration, J48 is found to be the best classifier, as Random Forest takes high execution time.
* As different classifier gives different accuracy, Ensembling of classifier is done through majority voting method. By using Ensembling, maximum accuracy time is obtain, in spite of the fact it takes huge amount of time

1. **CONCLUSION AND FUTURE WORK 22**

**Conclusion**: In this report we first study the different classifier in machine learning. Then, we compare 5 different classifiers (KNN, J48, Random Forest, SVM and Naive Bayes) on the basis of accuracy in which Random Forest dominates followed by J48 with a very small gay of accuracy. On the basis of execution time, J48 gives the least execution time. Therefore, on the basis of both accuracy and execution time J48 is the best. For Comparing we use 10 datasets.

Different classifier gives different accuracy, for specified datasets. As such, we ensemble the all the five classifier and study its accuracy and resolution time where we found that ensemling gives the best accuracy in spite of the fact that it takes huge amount of time compared to J48.

**Future Work:** In case where datasets is very huge (i.e. in TB, PB) ensembling does not give the accurate result. So, parallelization and dispersion of dataset is needed in those cases.

In case of big data, accuracy given by ensembling is not up to the mark. So, better methods like Neural Network can be applied to these datasets.

**REFERENCES 23**

1. Opitz, D.; Maclin, R. (1999). "Popular ensemble methods: An empirical study". [Journal of Artificial Intelligence Research](https://en.wikipedia.org/wiki/Journal_of_Artificial_Intelligence_Research). **11**: 169– 198. [doi](https://en.wikipedia.org/wiki/Digital_object_identifier):[10.1613/jair.614](https://doi.org/10.1613%2Fjair.614).

2. Polikar, R. (2006). "Ensemble based systems in decision making". IEEE Circuits and Systems Magazine. **6** (3): 21–45. [doi](https://en.wikipedia.org/wiki/Digital_object_identifier):[10.1109/MCAS.2006.1688199](https://doi.org/10.1109%2FMCAS.2006.1688199).

3.Rokach, L. (2010). "Ensemble-based classifiers". Artificial Intelligence Review. **33** (1-2): 1–39. [doi](https://en.wikipedia.org/wiki/Digital_object_identifier):[10.1007/s10462-009-9124-7](https://doi.org/10.1007%2Fs10462-009-9124-7).

4. Kuncheva, L. and Whitaker, C., Measures of diversity in classifier ensembles, Machine Learning, 51, pp. 181-207, 2003

5.Sollich, P. and Krogh, A., Learning with ensembles: How overfitting can be useful, Advances in Neural Information Processing Systems, volume 8, pp. 190-196, 1996.

6. Brown, G. and Wyatt, J. and Harris, R. and Yao, X., Diversity creation methods: a survey and categorisation, Information Fusion, 6(1), pp.5-20, 2005.

7. [Accuracy and Diversity in Ensembles of Text Categorisers](http://www.clei.cl/cleiej/papers/v8i2p1.pdf). J. J. García Adeva, Ulises Cerviño, and R. Calvo, CLEI Journal, Vol. 8, No. 2, pp. 1 - 12, December 2005.

8.Ho, T., Random Decision Forests, Proceedings of the Third International Conference on Document Analysis and Recognition, pp. 278-282, 1995.

9. Gashler, M. and Giraud-Carrier, C. and Martinez, T., [Decision Tree Ensemble: Small Heterogeneous Is Better Than Large Homogeneous](http://axon.cs.byu.edu/papers/gashler2008icmla.pdf), The Seventh International Conference on Machine Learning and Applications, 2008, pp. 900-905., [DOI 10.1109/ICMLA.2008.154](http://ieeexplore.ieee.org/search/wrapper.jsp?arnumber=4796917)

10.R. Bonab, Hamed; Can, Fazli (2016). [A Theoretical Framework on the Ideal Number of Classifiers for Online Ensembles in Data Streams](http://dl.acm.org/citation.cfm?id=2983907). CIKM. USA: ACM. p. 2053.

**24**

11. R. Bonab, Hamed; Can, Fazli (2017). [Less Is More: A Comprehensive Framework for the Number of Components of Ensemble Classifiers](https://arxiv.org/pdf/1709.02925.pdf) (PDF). TNNLS. USA: IEEE.

12.[Tom M. Mitchell](https://en.wikipedia.org/wiki/Tom_M._Mitchell), Machine Learning, 1997, pp. 175

13.Breiman, L., Bagging Predictors, Machine Learning, 24(2), pp.123-140, 1996.

14.[Hoeting, J. A.](https://en.wikipedia.org/wiki/Jennifer_A._Hoeting); Madigan, D.; Raftery, A. E.; Volinsky, C. T. (1999). "Bayesian Model Averaging: A Tutorial". Statistical Science.***14****(4): 382–401.*[doi](https://en.wikipedia.org/wiki/Digital_object_identifier):[10.2307/2676803](https://doi.org/10.2307%2F2676803). [JSTOR](https://en.wikipedia.org/wiki/JSTOR) [2676803](https://www.jstor.org/stable/2676803).

15.David Haussler, Michael Kearns, and Robert E. Schapire. Bounds on the sample complexity of Bayesian learning using information theory and the VC dimension. Machine Learning, 14:83–113, 1994

16.Domingos, Pedro (2000). [Bayesian averaging of classifiers and the overfitting problem](http://www.cs.washington.edu/homes/pedrod/papers/mlc00b.pdf)*(PDF).* Proceedings of the 17th[International Conference on Machine Learning (ICML)](https://en.wikipedia.org/wiki/International_Conference_on_Machine_Learning). pp. 223––230.

17.Minka*, Thomas (2002*), [Bayesian model averaging is not model combination](http://research.microsoft.com/en-us/um/people/minka/papers/minka-bma-isnt-mc.pdf) (PDF)