Large Scale Hierarchical Classification

Adarsh Khalique
Department of Computer Science; SZABIST Institute
Karachi, Pakistan
edarsh@hotmail.com

Rahim Hasnani
Department of Computer Science, SZABIST Institute
Karachi, Pakistan
rhasnani@yahoo.com

Abstract— **This study elucidates various algorithms used for document or text classification challenge. A sample data is used in this study on which various algorithms like Support Vector Machines (SVM), Naïve Bayes, Neural Networks and K-Nearest Neighbor are used in order to analyze their performances and accuracies. This study tries to identify the limitations and strength of these algorithms on the given sample data that how optimally they can perform classification. Different validations are used in this study to examine the accuracies regarding the classification can be identified. Validations include Split-Validation, X-Validation and Bootstrapping. Different ways and methods are discussed through which classification is made possible in large hierarchy. Finally this study concludes on the basis of results obtained that which machine learning technique or classifier performed excellent on the provided sample data set and achieved higher accuracy as compared to others.**

Keywords-component; Text Classification, Document clustering, Support Vector Machines(SVM), K-Nearest Neighbor, Refinement, Deep Classification on large scale hierarchy, cross-validation, split validation, BootstrappingValidation, training classifiers.

I. INTRODUCTION

Online commerce has gained huge success and popularity over past decades [1]. People are more and more towards shopping online and save their time [1]. Companies like eBay.com and Amazon.com that are considered as the giant companies for ecommerce business contains a very large and long-tail inventory with trillions of items or products entered into market palace every day [1]. These items are managed into categories defined which ultimately help users to find the desired product or item easily [1]. For example, eBay have structure of approximately 20,000 leaf categories which represents all of the goods that can be legally traded all over the world. Item categorization is fundamental concept followed by the e-commerce websites [1]. Properly assigning an item to its most suitable category is the major task that required to be done [1].

Yahoo directory or DMOZ are the other examples that follow the concept of classification but in terms of documents rather than items [2][5]. For example in DMOZ directory there are number of categories mentioned, the task is to assign a webpage to its most suitable category manually [2]. Performing this task manually needs lots of effort for the user whereas it require methods that can automate this task in more effective and efficient manner [2]. With the increase amount of data that are coming from different sources automatic classification has become the need of online businesses (especially ecommerce sites) [2]. To perform the task of classification a document classifier is used [3]. The basic purpose of this classifier is to extract the features of the document and then properly assign that document to the most appropriate category [3]. If the document classifier has the larger set of classes it indicates that it has the more potential for making more appropriate or precise classification as compared to the one with the fewer classes [3]. For large sets of classes a hierarchical structures are applied to increase the usability of the classes [3]. A hierarchical structure for a classifier is termed as "taxonomy" [3]. It is observed that the hierarchical structure or taxonomy is constructed manually and then users access to them for accessing documents from those categories [3]. Systems that automatically classify the documents into hierarchical taxonomies are called hierarchical classifiers [3]. In machine learning, classification of data is a task which is very common [4]. For example if the system given some data which can be part of any class or it can belong to more than one class as well, final task would be to decide that to which class this data will be assigned[4].

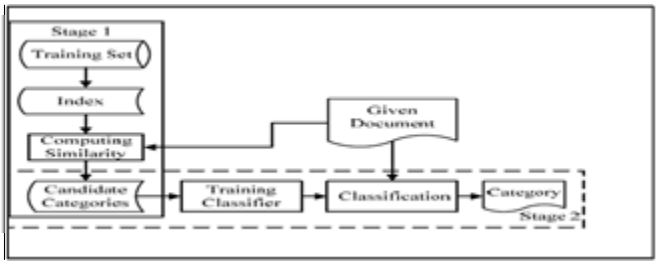
II. RELATED WORK

It is observed that when there is deep hierarchy of classes linked with each other than for a given a document it is expected that it will be assigned to the most appropriate category deep inside the hierarchy [4]. Increase in number of classes dramatically decreases the predictive accuracy that makes classifier fail [4]. Second problem is that it requires long time for training if large size taxonomy exists [4]. Third difficulty is that categories are organized in the hierarchical structure, which indicates the complex relationships parent-child [4]. Classification problem can be solved using various approaches which includes first the "big bang approach" and secondly "Top-down classification" [4].

A. Classification in Large-scale Text Hierarchies

1) Two-Stage Approach

In the internal working of first step hierarchy is prepared into the flat categories and then the related categories are extracted related to the given document [4]. Categories are ranked and then best suitable categories are considered as the candidate categories [4]. This way the large hierarchy is reduced in size [4]. In next step a model is trained on minor set (set belongs to the original data) and documents are classified in that small subset [4].



Flow Chart of Deep Classification [4]

2) Old-fashioned Text Classification

SVM has worst performance (using the flat classification) as compared to the Top down based SVM. Another system is proposed for large scale flat classification of data [4]. To reduce the possible categories to a minor set that can be manageable feature based filtering is used. It helps to improve the performance [4].

3) Applying Text Classification (Hierarchical type)

Two approaches are followed in Hierarchical Text Classification:

One classifier is used for entire hierarchical structure of the categories [4]. This method has been developed with the combination of rule-based classifier etc. [4]. The period of execution taken by this approach is greater as compared to top-down hierarchical approach this is called the Big Bang Approach [4].

Top-down approach is designed on the bases of Bayesian and SVM classifiers [4]. When this approach was tested on Yahoo directory, it was found that the performance is lowers (40%) at level 5th [4].

4) Deep classification

In this approach if a document is provided than the classes can be divided into two different types or forms, one the related categories and second the unrelated categories [4]. Relevant classes or types remain the major interest in this approach [4]. Small subsets of the related categories are extracted from large scale hierarchy [4]. Finally the classification is done on mined sets obtained with considering actual hierarchy [4].

B. Search stage Strategies

In this step or stage there are few plans are trailed to discovery the category candidates for a document. These two strategies are known as "Document-based" and "Category-based". These strategies are discussed below briefly:

1) Strategy on the basis of Document

In this type of strategy for a particular document related documents are searched in the training set [4]. All of these documents are characterized as frequency vector [4]. An evaluation in done for a particular document and documents present in the training set using the cosine resemblance amount [4]. Finally, highest N documents are nominated as the most alike documents to the given text [4].

2) Strategy on the basis Categories

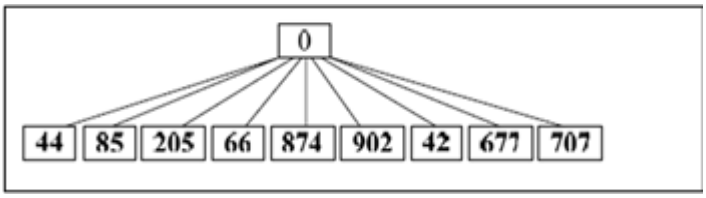
In this type of strategy, categories are represented with the text inside the categories and finally resemblance is calculated between two components (document and category) [4]. There exist few leaves or node in the classes, a course of time occurrence can be built for each class [4]. Again cosine between the vectors can be computed for these categories pages and given document [4].

C. Strategies in classification stage

There are few strategies for training data are known as "Flat Strategy", "Pruned Top-down Strategy" and "Ancestor-Assistant Strategy". These strategies are discussed below briefly:

1) Flat Strategy

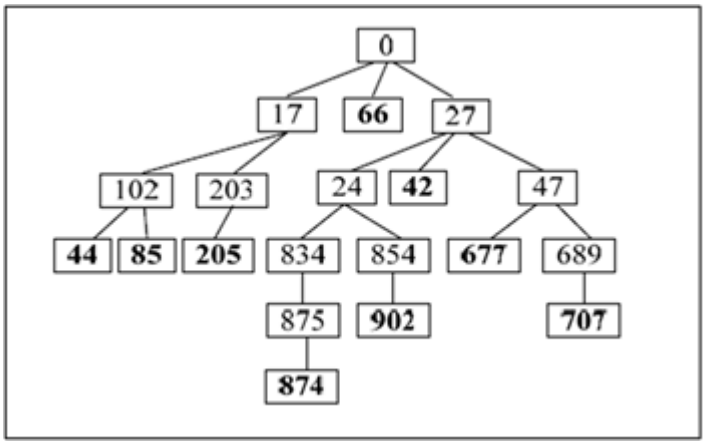
In flat strategy all the class contestants are positioned right at origin [4]. In the candidate categories there are web pages on which the classifiers are trained [4].



Flat Strategy [4]

2) Pruned Top-down strategy

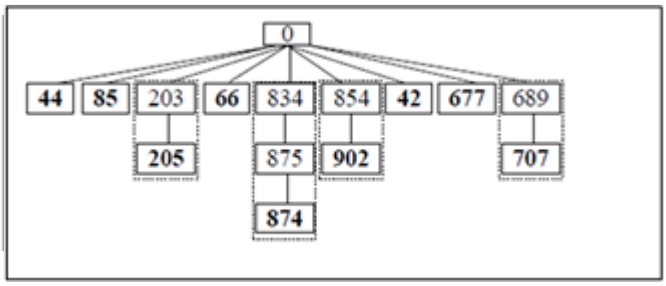
In pruned top-down strategy the classifier first classifies the document at the root node then it moved deeper to the candidate nodes, it moves deeper until it reaches the most suitable category for the given document [4].



Pruned Top-down strategy [4]

3) Ancestor-Assistant Strategy

In this strategy there are two things highly focused and tried to overcome this issue [4]. First the size of preparation data can be inadequate (category candidate) which requires to be obtained elsewhere [4]. Second, the training data may be too general for the parent nodes to imitate the properties of inner child candidate [4]. By combining both strategies' training data the result excludes the nodes which are not shared commonly among the ancestors [4]. The height is limited to two-level higher to prevent data unbalanced and performance [4].



Ancestor-Assistant strategy [4]

D. Classification in Large-scale Text Hierarchies

1) Two-Stage Approach

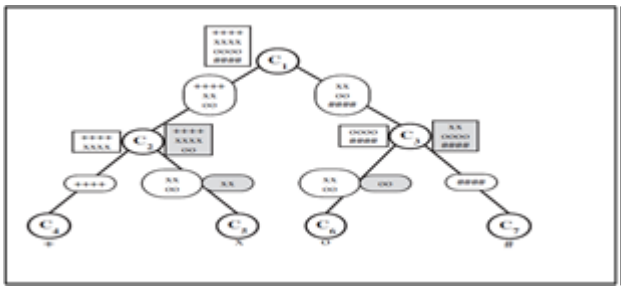
In the internal working of first step hierarchy is prepared into the flat categories and then the related categories are extracted related to the given document [4]. Categories are ranked and then best suitable categories are considered as the candidate categories [4]. This way the large hierarchy is reduced in size [4]. In next step a model is trained on minor set (set belongs to the original data) and documents are classified in that small subset [4].

E. Problems Associated with Classification

As the number of websites and their pages increases day by day it indicates that there should be some methods that can work automatically for assigning the webpages to the categories they belong to [5]. This is not an easy task to accomplish [5]. Data scarcity problem or challenge is usually faced in hierarchical classification [5]. There are other two most common challenges faced in hierarchical classification [5].

1) Error Propagation and Refinement

Classifiers can be trained in two ways, flattened approach and hierarchical approach [5]. Document can either belong to any class which reflects the concept of positive data and it can also happen that document does not belong to the class which indicates the negative data, this comes under the flattened approach and it is true for the training data [5]. Whereas in other approach know as hierarchical approach, document can either belong to any class which reflects the concept of positive data versus documents that fit to the parent (negative data) [5]. Data classification runs in the top-down fashion [5]. In Pachinko Model, first the document is classified to the most upper level and if the classifier predicts positive then classifying of document moves to the next lower level [5]. It can also happen that a document can belong to multiple children for which binary classifier can be built for each topic [5]. In the practical environment, as the classification progresses down the classifier are predicting the classes for the documents [5]. First document that should be the part of the lower branch is omitted (false rejections) and other document that ought not to be the part of that child category has been included (false positives) [5]. In the case of false positives, let's recognize them early and do not let them pass on [5]. "Refinement" can be considered as the perfect term here to indicate this problem [5]. A modest approach of using "CROSS VALIDATION" is the key [5]. Cross-Validation did over the preparation data and expected labels are used over it train data to a node [5]. In text classification it is highly observed that the amounts of documents that are allocated to the class are lesser [5]. But the classifier depends on the quantity of non-negative documents that belong to that class rather than the total number of documents presents [5]. The below figure 5 we can see that the examples from C5 and C6 are jumbled among themselves [5]. Figure tells about the 16 examples whose association is stable among the kid nodes [5]. Alteration understands examples alike to fault cases in the case of training and it can also learn that halt propagation [5]. In the case of parallel training, the extensive categorizations classically trust on the growth, it makes sure the existing amount of processors is speedily drenched [5]. The main computational price of this method is in the n-fold cross validation which points towards linear cost of n times the standard approach [5].



Refinement uses predicted behavior [5]

III. Machine Learning Techniques

F. Support Vector Machines (SVM)

It is one the generally used approaches for grouping and deterioration study for data as it follows strong mathematical foundations. There are two salient features of SVM these are discussed below:

1) Marin Maximization

In the theory of ML (Machine Learning), to make the most of the broad view presentation for a given sample of the training data is referred as classification edge gatherings of SVMs Maximize [6].

2) Non-Linear Transformation of the feature space

A non-linear classification can be handled by SVM which can work efficiently by means of the kernel trick [6]. The characteristics of SVM are not suitable or desirable for large quantity of data because it has complex functionality in terms of training data [6]. For any data set is extremely reliant on its magnitude [6]. Different kinds of SVMs are introduced in order to increase the training efficiency [6].

3) Clustering-Based SVM (CB-SVM)

It is based on the idea of handling large data sets as SVM does not achieve good results with large data (for training purpose) as compared to a well worth of examples of the records set [6]. SVMs work in a manner that they maximize the performance level by selection the drill data in selective sampling practices [6]. But in order to achieve this task it requires several examinations of the whole data set [6]. In contrast, CB-SVM uses the same idea but it put on a graded micro-clustering procedure that truly examines the data set for the one time only and delivers an SVM with excellent examples

which carries the statistical synopses [6]. Hence it increases the value of knowledge of the SVM [6].

4) Clustering-Based SVM

There are few steps followed by CB-SVM [6]. The very initial step taken by CB-SVM is that it forms negative and positive preparation data it builds double micro-cluster trees [6]. The nodes on the higher level represent the children nodes [6]. In the end it starts teaching the SVM to the end nodes [6]. Using the tree structure, the data is selected and de-clustered (which lies near to the boundary from the root nodes) [6]. CB-SVM keeps on working in the same manner until it reach to the leaf level [6].

5) SVM

If the limited training data set is given then the optimal class boundary function (considered as hypothesis also) is deliberated the one which delivers the finest overview performance which marks the finest performance on the hidden data [6]. To boast up the generalization by increasing of margin the SVMs comes under good techniques [6]. Furthermore over fitting and under fitting is also avoided by SVMS as it uses advanced kernels [6]. In feature space it is noticed that the space from the margin to the closest point or found data is known as margin in SVMs [6].

A. K-Nearest Neighbors Algorithm

It is one of the methods for classifying data into the appropriate categories [7]. It follows the concept of categorizing items on the foundation of nearby training instances discovered in the feature space [7]. This algorithm or method is considered as the modest algorithms as compared to others, an object is classified by the votes given to it by its neighbours [7]. If an object is assigned to any class so it reflects that it is closest to k adjacent neighbours where K is characteristically minor in value and a positive integer [7]. Nearest neighbour problem is thoroughly studied under the name of closest pair of point's problem [7].

B. Naive Bayes

This classifier's concept or main idea is based on the probabilistic approach which is quite simple not too complex one [8]. Naïve Bayes classifier works in a manner that it assumes the existence or non-existence of any characteristic is dissimilar to the information provided by the class variable [8].

Naïve Bayes can also be explained with an example that a fruit can be considered to be an apple if it has these features, if it is red, round and about 3 in diameter. Naïve Bayes consider each of the features independently calculating probability that the given fruit is an apple on the bases of the presence or absence of features [8].

C. Neural Networks

Neural Networks traditionally used to refer to networks. It is also referred as circuit of biological neurons [9]. In Neural Networks a hidden layer is used between the two extremes i.e. input layer and output layer to create a model [9]. The artificial networks can be utilized for predictive modelling, adaptive control etc. [9].

D. Data Set

The sample data which is taken from the web "Kaggle.com" is drawn from web classification. The data is spare and high dimensional data with a million features. The file contains 50,000 data points for training and testing. Another file is provided with 50,000 labels that must be embedded with data points in order to process further.

E. Data Set

The sample data which is taken from the web "Kaggle.com" is drawn from web classification. The data is spare and high dimensional data with a million features. The file contains 50,000 data points for training and testing. Another file is provided with 50,000 labels that must be embedded with data points in order to process further.

F. Data Set

The sample data which is taken from the web "Kaggle.com" is drawn from web classification. The data is spare and high dimensional data with a million features. The file contains 50,000 data points for training and testing. Another file is provided with 50,000 labels that must be embedded with data points in order to process further.

IV. Experimental Analysis

Following are the few different validations applied to the above mentioned algorithms on the given kaggle's dataset. Different behaviour or results are obtained in this experiment. Every algorithm has different way of dealing with the given data which results variation in results. Results are mentioned below, and about the sample data that has been taken bit information about it also discussed below:

A. Data Set

The sample data which is taken from the web "Kaggle.com" is drawn from web classification. The data is spare and high dimensional data with a million features. The file contains 50,000 data points for training and testing. Another file is provided with 50,000 labels that must be embedded with data points in order to process further.

B. Tools

The data points are kept into Excel file (.csv), hence Microsoft Excel is used. And most importantly "Rapid Miner" tool is used for further classification work and obtaining the result set. The labels file indicates that it is binominal data.

C. Methods to process data

There are various methods available for classification of data, few of them are listed below (these are the algorithms used in this research):

1. Neural Networks
2. KNN
3. Naïve Bayes
4. Support Vector Machines (SVM).

D. Validation

Validation is done with the help of different type of validation types applied to the sample data. 70% of the data is used for the purpose of training and rest 30% is used for testing. Types of validation applied are listed below:-

1. Split Validation
2. X-Validation
3. Bootstrapping

E. Accuracy

Accuracy is measured on the bases of predicted true or false assignment of a value to a category. Accuracy can be better understood by considering the table below:-

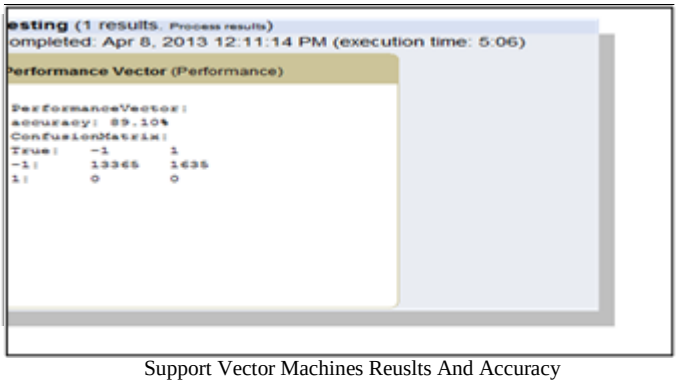
		Predicted Values	
		True	False
Observed Values	True	True Positive (TP)	False Negative (FN)
	False	False Positive (FP)	True Negative (TN)

Predicated Values Table

F. Split Validation

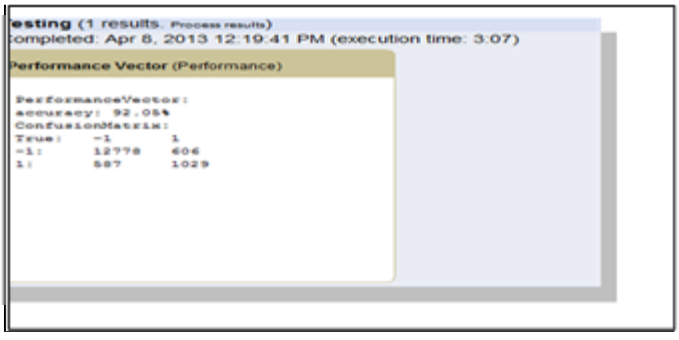
Starting with the Split Validation which is applied to the following algorithms and different results are obtained.

1) Support Vector Machines



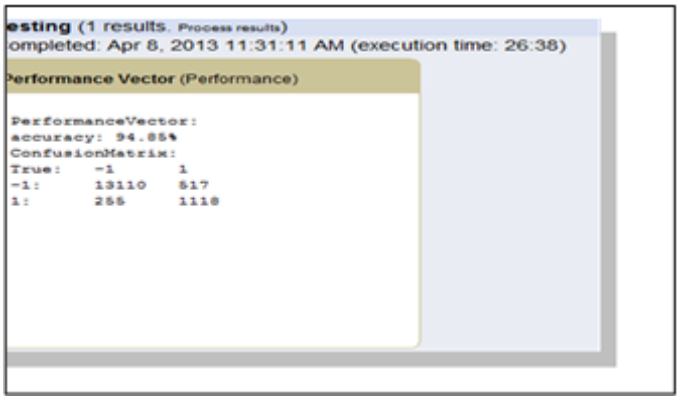
Support Vector Machines Reuslts And Accuracy

2) KNN



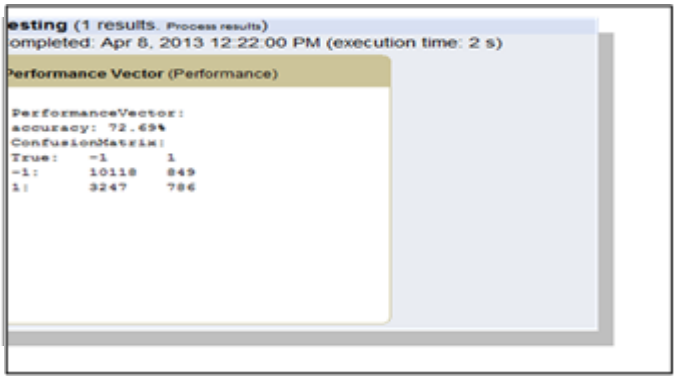
KNN Reuslts And Accuracy

3) Neural Networks



KNN Reuslts And Accuracy

4) Naïve Bayes



Naïve Bayes Reuslts And Accuracy

From the above results the following results cab obtained in terms of accuracy and execution time taken by these techniques.

1. Execution time taken by SVM is 5min 06sec and accuracy is 89.10%.
2. Execution time taken by KNN is 3min 07sec and accuracy is 92.05%.
3. Execution time taken by Neural Networks is 26min 38sec and accuracy is 94.85%.
4. Execution time taken by Naïve Bayes is 02sec and accuracy is 72.69%.

G. X-Validation

Following are the results obtained using X-Validation and the value of K=10:

1. Execution time taken by SVM is 1hour 02min 35sec and accuracy is 89.15%.
2. Execution time taken by KNN is 12min 56sec and accuracy is 91.93%.
3. Execution time taken by Neural Networks is 6hours 03min 04sec and accuracy is 95.00%.
4. Execution time taken by Naïve Bayes is 05sec and accuracy is 72.67%.

Following are the results obtained using X-Validation and the value of K=3:

1. Execution time taken by SVM is 21min 02sec and accuracy is 89.15%.
2. Execution time taken by KNN is 10min 01sec and accuracy is 91.67%.
3. Execution time taken by Neural Networks is 1hours 48min 29sec and accuracy is 94.84%.
4. Execution time taken by Naïve Bayes is 03sec and accuracy is 72.79%.

H. Bootstrapping Validation

Following are the results obtained using Bootstrapping-Validation where sample ratio is 0.7:

1. Execution time taken by SVM is 18min 51sec and accuracy is 89.13%.
2. Execution time taken by KNN is 10min 08sec and accuracy is 91.81%.
3. Execution time taken by Neural Networks is 01hours 19min 09sec and accuracy is 94.92%.
4. Execution time taken by Naïve Bayes is 04sec and accuracy is 73.13%.

V. Conclusion And Discussion

Four algorithms were tested on the same sample data using three types of validations (data taken from kaggle), and few results are obtained as mentioned under the topic Results and Analysis.

A. Best Performer

According to the analysis done above it is found that on the given sample data KNN and Naïve Bayes have less execution time and high accuracy. KNN is better in terms of accuracy with little more time consumed as compared to Naïve Bayes. Whereas Naïve Bayes is good in execution time but it compromises its accuracy (less accurate results as compared to KNN). In all three types of validations these two algorithms (KNN and Naïve Bayes) performed optimally.

B. Worst Performer

According to the results obtained SVM and Neural Networks seems like worst performer on given sample data. It is observed that Neural Networks have the highest execution time in all type of validations and it has given the highest accuracy as compared to other classification algorithms. On the other hand, SVM have taken less time for execution then Neural Networks but in terms of accuracy it is less accurate then Neural Network in all type of validations.

C. Other Findings

The purpose of using different validation on the each machine learning technique was to identify that whether these validations are effecting the execution time of any algorithm or not. It is noticed in the above findings that using different validations affects the execution time of each machine learning technique, although the accuracies produced by each of them are quite similar. And finally it is observed that from all types of validations applied to the given sample data, Split Validation helped each machine learning technique to produce results in the minimum execution time with suitable accuracy (as accuracies of each algorithm are almost similar when different validations were applied).

Changing the K value in X-Validation shows that it affects the execution time consumed by each algorithm whereas the accuracies produced by it on the given data seem almost similar. In Bootstrapping validation the sample ratio was kept 0.7 which indicates that for training 70% of data was used and rest 30% of data was used for testing. In Spit Validation 70% data was used for training and 30% for testing. The accuracy rates of Split Sampling, X Validation and Bootstrapping shows that the results are internally validated with good accuracies and estimates for external validation are also quite encouraging.

VI. Future Direction

In this study an attempt is made to analyse the performance of different classification algorithms on a given data. Different execution time and accuracies are measured with the help of using different validations. This study can be further extended into number of directions. Each algorithm can be tested on different kind of data sets available on internet so that different behaviours can be measured on the basis of various sample data sets. Changing sample data sets can help to analyse the weaknesses and strong points of an algorithm that on what kind of data set an algorithm performs well and give the highest accuracy. In deep hierarchies to what extend an algorithm can classify a given data more accurately can be measured so that the level of best classification provided by algorithms can be measured. Each algorithm's weaknesses can be pointed out and few suggestions can be added to them in order to increase the level of performance and accuracy. Finally, after knowing the working criteria of each algorithm a technique can be proposed which can work as a classification algorithm best suitable for deep hierarchies.

Acknowledgement

Foremost, praise to Allah, the Almighty Who enabled me to complete this very project by bestowing that zeal and commitment that was ever needed and for giving us the courage and strength to complete the project on time.

Secondly, I would like to thank our Teacher, Mr. Rahim Hasnani, for guiding us in the field and helping us explore a new dimension of technology. I would also like to thank him for providing us an opportunity to show our aptitude and providing us with all the necessary guidance and assistance throughout research.

To add to this, we are thankful to all the friend we met daily who beamed us with bright smiles that brightened our day and for their help during the entire project.

References

[1] D.Shen, J.D.Ruvini and B.Sarwar, "Large-scale Item Categorization for e-Commerce," eBay Research Labs 2145 Hamilton Avenue San Jose, CA 95032, USA, In proceedings of the 21st ACM international conference on Information and knowledge management, pp 595-604, 2012.

[2] R.Babbar, I.Partalas, E.Gaussier, and C.Amblard, "On Empirical Tradeoffs in Large Scale Hierarchical Classification," LIG, Université Joseph Fourier.

[3] K.Nitta, "Improving Taxonomies for Large-Scale Hierarchical Classifiers of Web Documents," Yahoo Japan Research Tokyo, Japan, In proceedings of the 19th ACM international conference on Information and knowledge management, pp 1649-1652, 2010.

[4] G.Xue1, D.Xing, Q.Yang and Y.Yu, "Deep Classification in Large-scale Text Hierarchies," Shanghai Jiao-Tong University, pp 619-626, Jul 24, 2008.

[5] N.Bennett and N.Nguyen, "Refined Experts: Improving Classification in Large Taxonomies," Microsoft Research, pp 11-18, Jul 19, 2009.

[6] H.Yu, J.Yang and J.Han, "Classifying Large Data Sets Using SVMs with Hierarchical Clusters", University of Illinois UrbanaChampaign,IL 61801 USA, 2003.

[7] T. Hitendra Sarma, P. Viswanath, D. Reddy and Sri.Raghava, "An Improvement to K-Nearest Neighbor Classifier", Department of Computer Science and Information Technology NRI Institute of Technology-Guntur, Guntur.

[8] L.Chen and S.Wang, "Automated Feature Weighting in Naive Bayes for High-dimensional Data Classification", Department of Computer Science University of Sherbrooke Quebec, J1K 2R1, Canada, In Proceedings of the 21st ACM international conference on Information and knowledge management, pp 1243-1252, 2012.

[9] R.Sethukkarasi, U.Keerthika and A. Kannan, "A Self Learning Rough Fuzzy Neural Network Classifier for Mining Temporal patterns", Department of Information Science and Technology Anna University Tamil Nadu, India, In Proceedings of the International Conference on Advances in Computing, Communications and Informatics, pp 111-117, 2012.