Ascertain the Influence of Ensemble Procedure on Categorization of YouTube Video Label by Machine Learning Stratagem

Pritom Sarker[1],NakibAman Turzo[2], Biplob Kumar[3], Amit Chakraborty[4], Niloy Kumar Shaha[5]

[1]*(Department of Computer Science and Engineering, Varendra University, Bangladesh)*
[2]*(Department of Computer Science and Engineering, Varendra University, Bangladesh)*
[3]*(Department of Computer Science and Engineering, Varendra University, Bangladesh)*
[4]*(Department of Computer Science and Engineering, Varendra University, Bangladesh)*
[5]*(Department of Computer Science and Engineering, Varendra University, Bangladesh)*

Abstract: We live in the era of information technology where plenty of data is readily available and overwhelms our capacity to analyze and absorb. Extraction of text from different platforms through computational methods and analyzing is becoming a trend. A set of root words are used in those researches. Ensemble approach is the proposed method in which collection of algorithms is used in classification. This increase the machine learning accuracy by combining arrays of specialized learners. For video denomination to respective genre we collected data from YouTube and applied this technique. Python libraries are being applied for this purpose. Different sets of variances lie in different co-ordinates. It is being concluded that execution of ensemble model gone more accurate among all other classifiers. Best performance was given by single model pre-eminently to ensemble model. For categorization schemes we can employ this classifier on videos of different genres. For all ensemble methods edification of different learners is done as opposed to single learner that is used to make all classifications.

Keywords: Bagging, Ensemble machine learning, Natural Language Processing, Pasting, Stochastic Gradient Descent, Term frequency

I. INTRODUCTION

Text categorization is the process of allocating tags or categories to text according to its content. It's one of the foundational process in Natural Language Processing (NLP). Since the institution of digital documents this kind of categorization has always been a chief application and research. Due to the plenty of dockets dealing in our routines text classification has become a necessity. Unstructured information can be found everywhere on social media, web pages, YouTube, survey responses and more. Text is a huge source of data and information but extricating insights from it can be time devouring. Different businesses are turning to text classification for conversion of data quickly and in cost efficient manner to intensify arbitration and automated response. Text can be of any cast like articles, news reports, movie reviews, advertisements etc. It is quintessential to collect most of the data from web pages or different sites with different formats and different preferred vocabularies. Data is always heterogeneous in its characteristics. A document may be allotted to different categories and different approaches got followed for this purpose. Emotional, intent and sentiment analysis of textual data is also most important part of categorization. The process of building a categorizer is not different from other classifiers. The complexity of languages and high dimensionality has made this problem a little difficult. So we investigated a different method for text classification. The results of experimental assessment of different methods led to the selection of one classifier as a solution to problems. In this paper we have aggregated the text in YouTube video titles through ensemble machine learning models. This classification approach uses two or more algorithms and calculate the mode value based on vote reference for every algorithm. It provides better accuracy in classification of labels to their respective categories.

II. LITERATURE REVIEW

In an investigation, first classification problems were divided into steps:

- Pre-requisite and setting up the environment
- Loading data set
- Extracting features from text files
- Running algorithms
- Grid search for parameter tuning

In it the classic problem in NLP was learnt [1].

In another research text classification through machine learning was done. Automated text classification has been considered vital method to manage a vast number of documents that are widespread and spontaneously increasing. Stemming is a preprocessing step. To reduce the size of initial feature set is to remove misspelled or words with stem. Stemming is taken as amplifiers of classifiers performance. Many methods are used to determine effectiveness; however, precision, recall and accuracy are frequently used. A series of experiments suggested that the use of senses results in no categorization improvement. Basic goal of feature-method is to remove dimensionality of dataset by removing features irrelevant for classification [2].

Phase of screening articles requires tiring efforts and for this purpose the goal was set for the use of automatic tools. Recent trend of text classification methods to semi-automate by providing decision support for these processes hence reducing the required efforts and time. In this work are, contribution to line of work by performing a comprehensive set of text classification experiments on corpus resulting from an actual systematic review in the area of Internet Based Randomized Controlled Trials. These experiments implemented multiple machine learning algorithms combined with several feature selection techniques. Results were generally positive in the case of overall precision. It also revealed that using only article titles provide virtually as good results [3].

A comparative analysis of machine learning models were done for quality pillar assessment of SaaS services by multi-class text classification of user’s reviews. Eleven traditional machine learning classification approaches were used along with weighted voting ensemble of these classifiers to achieve this task and performance was tested [4].

In another study the aim was to provide an overview of state of art elements of text classification. For this purpose, primary studies were selected and it was quantitatively and qualitatively analyzed. Six baseline elements like data collection, weighing, feature selection, projection and training of classification model were considered [5].

Selection of discriminative features highly relevant to class labels while having low levels of redundancy is important to improve text classification methods. A novel multi-objective algorithm for text feature selection called Multi-Objective Relative Discriminative Criterion, was proposed which balances minimal redundant feature against those maximally relevant to target class. The method proposed employs a multi-objective evolutionary framework to search through solution space. First objective function measures the relevance of text features to target class whereas other evaluates the correlation between features [6].

Text classification is a domain inspiring researchers since many years. Textual data becomes more and more abundant on the web. For this, we developed a Term-Frequency-Inverse Document Frequency parallel model to save only the most related words in the docket. Then dataset was fed to parallel Naïve Bayes classifier. Both TF-IDF parallel model and parallel Bayes classifier were implemented on Hadoop system using Map Reduce architecture and efficiency of these methods were tested [7].

In another work done, text classification was done using neural network with LSTM (long short-term memory) units. Different approaches were tested using feature vectors which are representation of documents to be classified. Firstly, conversion of words into vector representation using word2vec tool was done and sequences of these vectors’ representations were used as features of document. This approach concluded that vectors outperformed a standard [8].

It is the cornerstone of categorization of documents, personalization and document routing. An empirical comparison twelve feature selection methods evaluated on a benchmark of 229 text classification problem instances that were collected from Reuters, TREC, OHSUMED etc. Analysis were done from multiple goal perspectives-accuracy, F-measure, precision and recall since each is appropriate in different situations. New feature selection was devised called Bi-Normal Separation outperformed the others by a substantial margin in most situations [9].

Classification of text is also done based on Apache Spark distributed computer framework. The starting point of this classification is the generation of high dimensional feature vectors from documents, this task is realized with methods and tools for natural language processing [10].

Variants of Naïve Bayes and Support Vectors Machines are classification methods baseline but depending on model variants the performances varies. It was shown that the inclusion of word bigram features gives sentiment analysis gains, NB does better than SVMs and its consistency performs well across tasks [11].

KNN method which was proposed in which K0 candidate category are got through Rocchio classification method and then the representative sample parts are extracted in the k0 category training document. This method resolve problems to some extent and has improved results in classification [12].

A supervised classifier is built on the basis of training corpora containing correct label for each input. During training, each input is converted to feature set by feature extractor which is used for classification. Machine Learning algorithms got fed with pairs of feature sets and label to generate a model [13].

An ‘Unmasking method’ is also being introduced in which two input documents are chunked and cross-validation is used to check their effectiveness of machine learning method. For gaining representativeness the chunk must be long. Unmasking is ineffective for short inputs [14].

Two machine algorithms were implemented recently i.e. RIPPER and sleeping experts for phrases, on all text categorization problems. Classifiers built by these algorithms allows the word so that its presence or absence effect can be checked by it. These perform extremely well in categorization [15].

Machine learning classifiers are used for detection of automated articles. It's expensive to prepare training data set classifiers which usually consists of manually labeled relevant articles and for mitigating this challenge randomly sampled unlabeled articles. This proved a cost-effective approach to training machine learning classifiers in real life-based bio surveillance [16].

There are many hierarchical text classification approaches that are involving pre-constructed hierarchy of categories. In it feature selection is based on terms which are unsuitable for hierarchical classification. Documents are represented by two types of models i.e. Boolean and vector model. Use of named entities improved its performance [17].

Typical applications assisting categorization are end-user assistance in archiving existing documents or helping h\them in browsing existing corpus of documents. For this purpose hyper rectangular algorithm extracts the list of most of representative words in hierarchical way [18].

A comparative study of text classification approaches for personalized retrieval was done, Traditional methods based on keywords matching are insufficient. These methods developed in text mining community gave good results. These comparative techniques shows that Naïve Bayes is a better method than support vector machines in building a personalized article and retrieval system [19].

An ensemble is a classification approach by combination of two or more algorithms. Mode value based on the vote reference for every algorithm was calculated. Naïve Bayes, SVM and Ensemble algorithm were combined used in it. It provides more accuracy then other algorithms [20].

Problem of automatic classification of human language written text was considered in research and accuracy of ensemble models were assessed. These significantly improve sentiment classification for freeform text [21].

III. METHODOLOGY

We amass our information from YouTube, if we desire video denomination of relevant genre. Utilization of python is being done for scraping of this information. Two python libraries called Beautiful soup & selenium are taken into account for web scraping or amassing. We have 10094 rows along with two columns of video label and genre of video in culminated dataset.

We took videos of different subjects which include:

- Animations
- Adults
- Sports
- Kids
- Education

Each case has a norm of 2019 data.

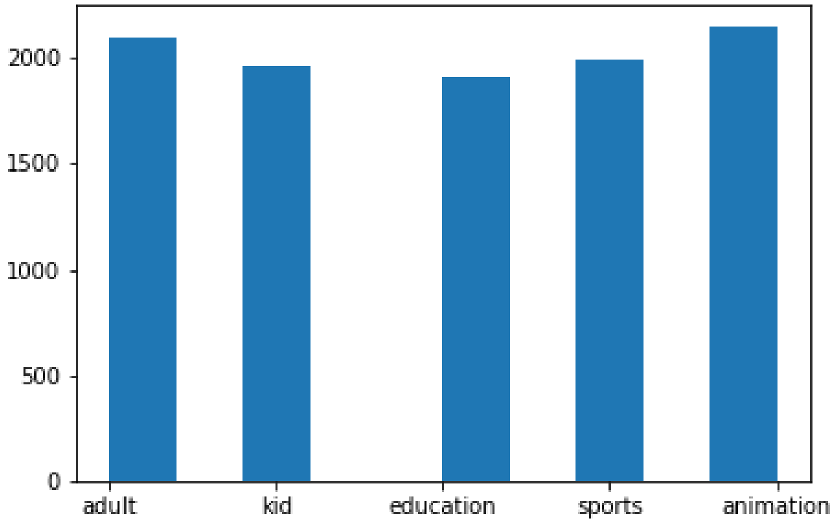

Fig. 1

We sundered this sorting of data part into three chunks.

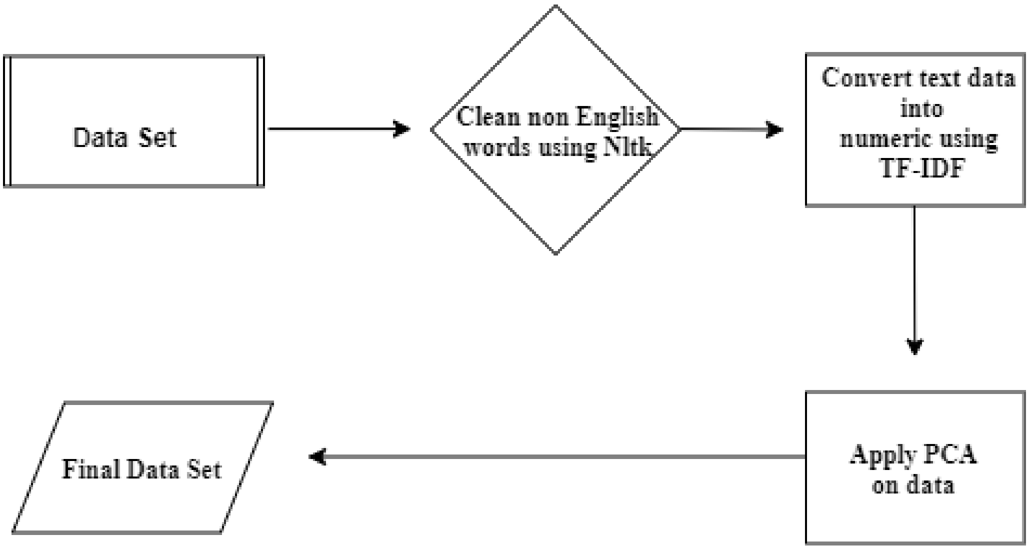
Data Set
Clean non English words using Nltk
Convert text data into numeric using TF-IDF
Apply PCA on data
Final Data Set

Fig. 2

1. Data Clean

We have non- English and stop words (which got filtered out before or after processing of natural language data) in our set of information. All the non-English words got axed from it by us. Natural Language Toolkit (NLTK) information center of python is being used for this purpose. We have all of the sentences in non-English in our information set. Ergo, after the moping through the process, on the norm we got 1983 data set.

2. Term Frequency–Inverse Document Frequency

An analytical statistic is a numerical or scientific form of statistic which is being contemplated to mirror the principal of word in a docket or corpus and is called Short Term Frequency-Inverse Document Frequency (TF-IDF). This factor has weightage in retrieving information, text mining and user modeling through hunting of this data.

Term Frequency (TF)

Frequency of a word which pops up in a docket divided by the gross number of words in the document. Every document has its own term frequency.

$$tf_{i,j} = \frac{n_{i,j}}{\sum_k n_{i,j}}$$

Inverse Data Frequency (IDF)

The log of the documents number divided by word w containing documents. Inverse data frequency determines the weight of rare words across all documents in the corpus.

$$idf(w) = \log\left(\frac{N}{df_t}\right)$$

TF-IDF is simply the TF multiplied by IDF

$$w_{i,j} = tf_{i,j} \times \log\left(\frac{N}{df_i}\right)$$

Our most work is being done from Scikit-Learn. Our text data is taken by it and converted to numeric information set. After this conversion, our data has 3394 features. We have so many less important features we can do features extraction using PCA.

3. Principal Component Analysis

A new coordinate system is being metamorphosed from data through orthogonal linear transformation so that each coordinate has greatest variance by scalar projection of data in an ordered way and so on. This is called principal component analysis. Higher variance comes to lie in first coordinate which is called first principal component and the lower variance in second coordinate. Our information set has 1678 features after application of principal component analysis. When applications of dimensions of principal component analysis got reduced and the data quality got lost.

In case of principal quality analysis, 95% caliber of data was being maintained. 95% of the quality of real data was preserved by setting value of ‘n’ components as 0.95. Our latest data has 1678 characteristics after application of principal component analysis.

IV. EXPERIMENTAL ANALYSIS

The foundation step is to do organization of single machine learning algorithm and precision evaluation. From each class data of 500 would be the norm taken from training set and to edify machine learning models we used 2500 data. Wielding of Scikit-Learn machine learning library was done for this experiment. We have done resampling procedure like cross validation after edification of our model. Cross-validation is a model validation technique for deducting into an independent data set by determining the results of a statistical assays. By edification of our model after separating and choosing randomly 80% data from dataset we used 20% from its test model and obtain accuracy. We calculated the average of all five results after replicating these steps and got result.

	SGD Classifier	Support Vector Machine	Decision Tree	Naive Bayes	Logistic Regression
Max	0.837905237	0.844611529	0.748756219	0.877192982	0.850746269
Min	0.811557789	0.814536341	0.690773067	0.802005013	0.814536341
Avg	0.826478209	0.827000766	0.718482298	0.828512025	0.833990903
F1	0.806060606	0.831	0.6818181	0.818	0.832

Table: 1

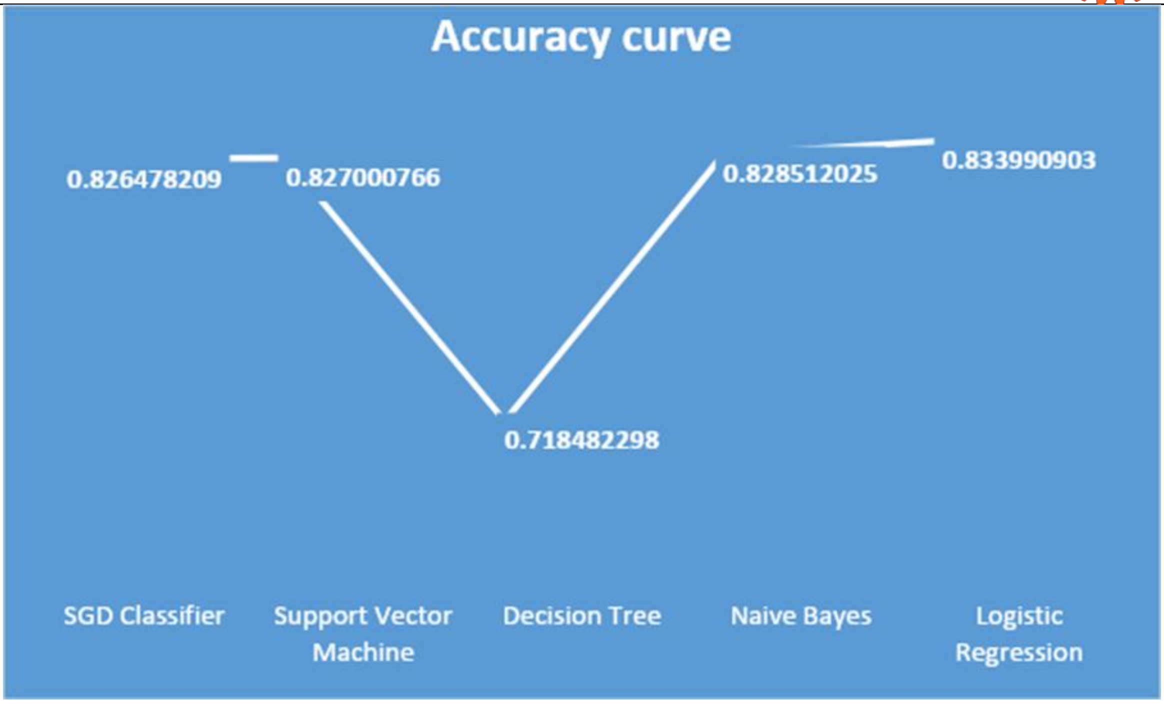

Fig: 3

Optimum precision is given by logistic regression and F1 score is highest in this regard as shown in above table. SGD classifier is Stochastic Gradient Descent (SGD) classifier which has more precision than support vector machine but accuracy is better than the support vector machine but we can see that the F1 score of SVM is higher than SGD. We also witnessed from our data that all other algorithms except decision tree are very close to each other.

We got 82% accuracy on an average from all four algorithms and got highest precision value from Logistic Regression which is 83%.

A voting classifier was made which is based on ensemble learning model. It will gather all votes after training classifiers and the end value is obtained from highest vote. This voting classifier is made with SGD classifier, support vector machine, naïve bays and logistic regression. We got 84% precision after five cross-validation. Therefore, we got better results with ensemble models than single models.

Bagging and pasting are the most common methods of ensemble learning. For application of bagging and pasting with respect to logistic regression we get best precision from single model. Adaboost is applied at the concluding part. Here is the comparison between all the ensemble models.

	Voting	Bagging Classifier	Pasting	Random Forest	Adaboost
max	0.877805	0.840399002	0.860349127	0.77235772	0.805
min	0.821608	0.803482587	0.829145729	0.74380165	0.760599
avg	0.845959	0.815486353	0.84046924	0.753260124	0.786517
F1	0.877	0.7803	0.859	0.863	0.763

Table: 2

From the above table we can say that the best accuracy we can get from the voting classifier. The four single models build with this voting classifier and on average; each model gives 82% accuracy.

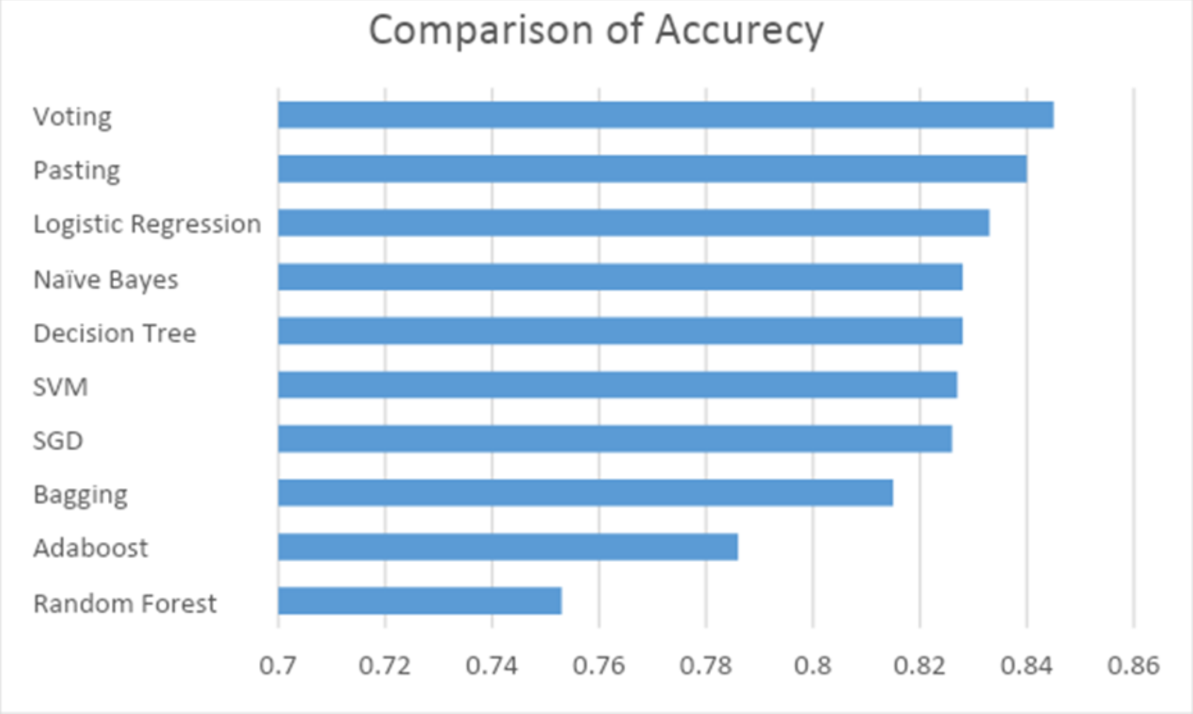

Fig: 4

Pasting ensemble model works better, we get 84.6% accuracy from that model. Among all single classifiers the ensemble methods like voting and pasting functioned best.

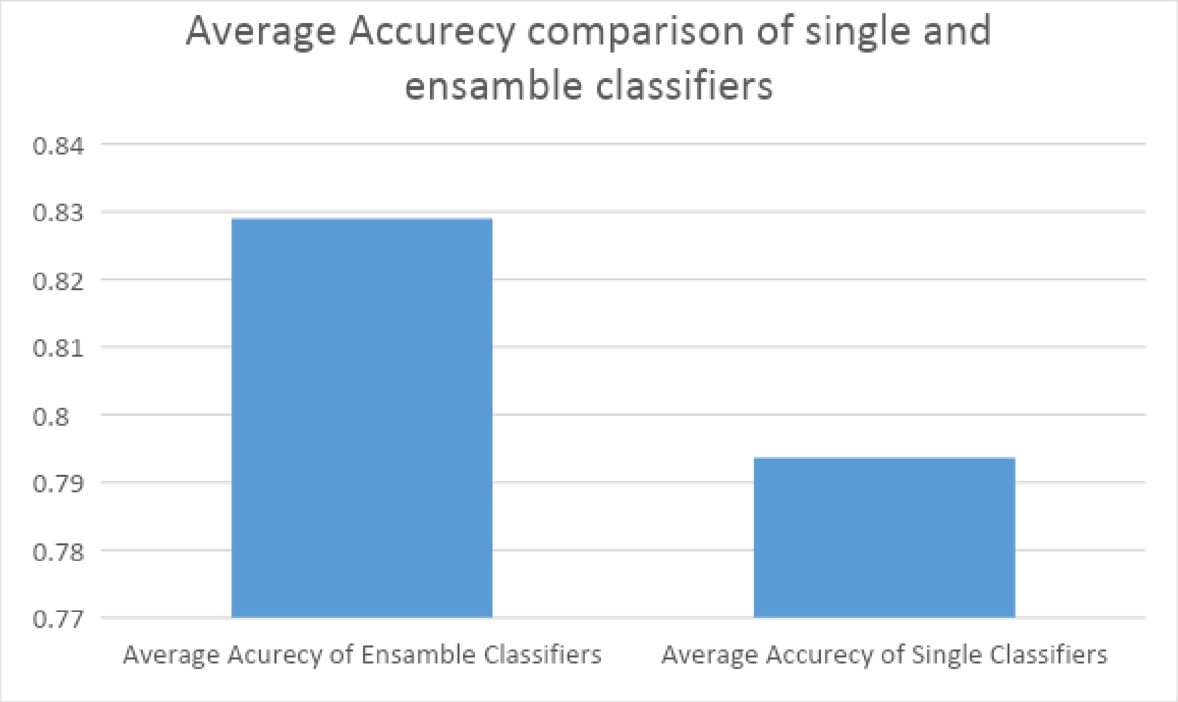

Fig: 5

In comparison to other single classifiers the ensemble accords an average of 4% more precision.

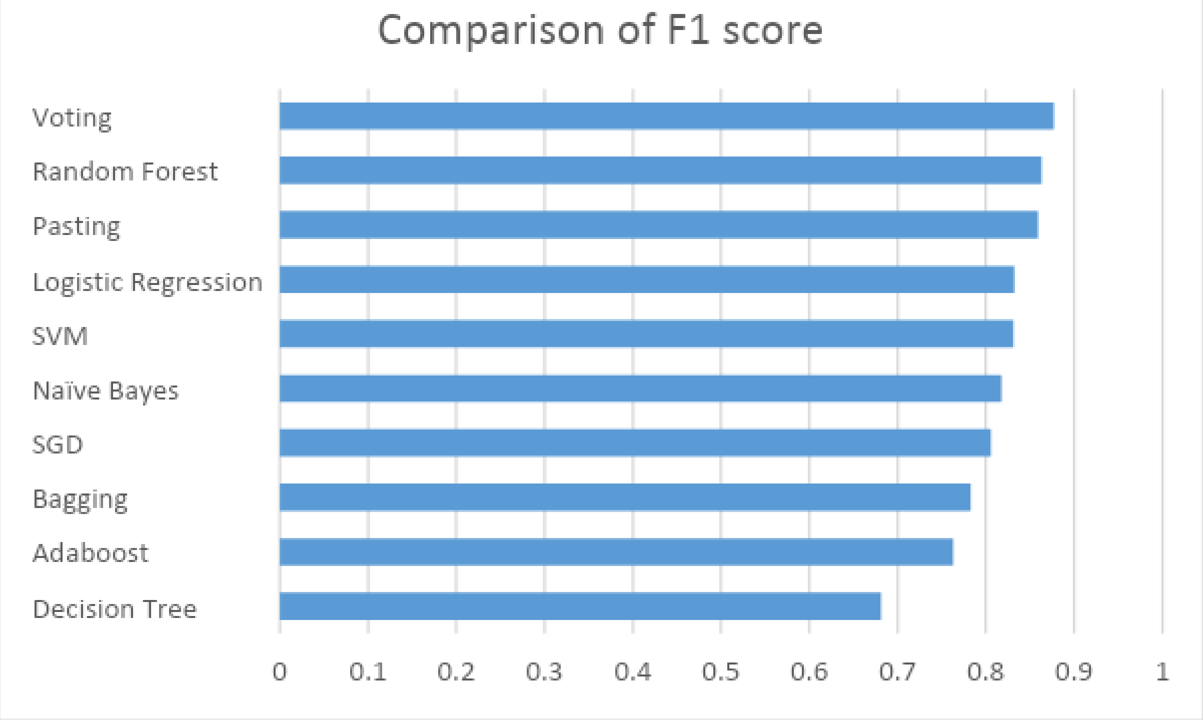

Fig: 6

With respect to other ensemble model the F1 score of this model is higher. Three ensemble methods like Voting, Random Forest and Pasting worked more precisely than single classifier. Our Voting classifier functioned matchlessly among all the classifiers.

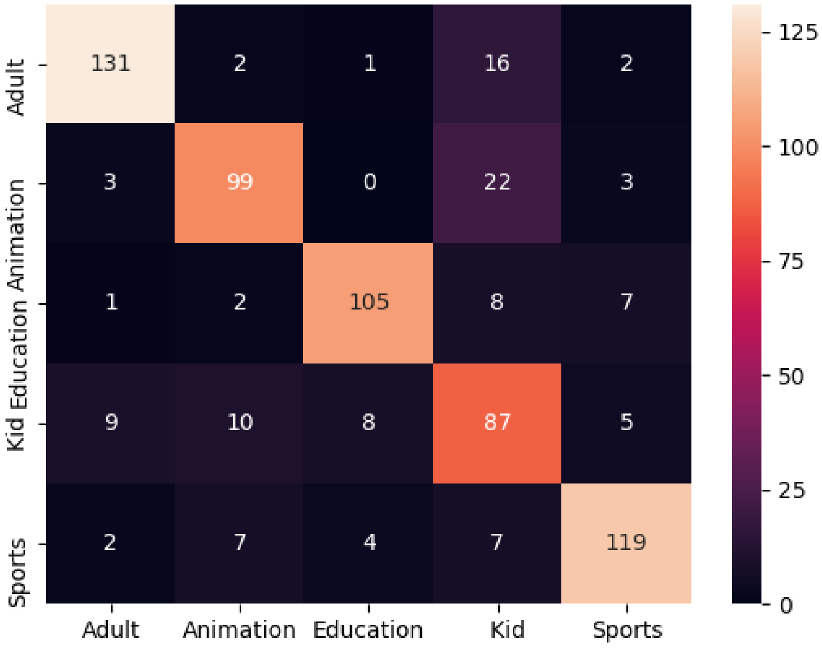

Fig: 7

With regards to video title classification our Voting classifier had given 1.5% improved precision than average classifier. In case of confusion matrix, voting method is much more precise and has reduced error. The problematic situation occurs when the video title should be about kids but it was grouped as another label like in some instances they were labeled as adult video.

V. CONCLUSION

From our experimental analysis we found that execution of ensemble model gone more precise among all other classifiers. Finest performance was given by single model as compared to ensemble model. For grading or classification schemes we can employ this classifier on videos of different genres. It's an illustration of multilevel text categorization. For all ensemble methods training of different learners is done as opposed to single learner that is used to make all classifications. In future we intend to use other ensemble models including ways to increase computational efficiency.

A conclusion section is not required. Although a conclusion may review the main points of the paper, do not replicate the abstract as the conclusion. Conclusion is giving more information of your work in short form and giving suggestion that what

Kind of suggestion want in this research.

REFERENCES

[1] Amazal, H. a. (2018). A Text Classification Approach using Parallel Naive Bayes in Big Data Context. Association for Computing Machinery, 6.

[2] BOOK, T. . (2008). Sentiment Mining Using Ensemble Classification Models. Research Gate, 509-514.

[3] Cohen, W. W. (1999). Context-Sensitive Learning Methods for Text Categorization. ACM Trans. Inf. Syst., 141–173.

[4] Forman, G. (2003). An Extensive Empirical Study of Feature Selection Metrics for Text Classification. 1289-1305.

[5] Ikonomakis, E. a. (2005). Text Classification Using Machine Learning Techniques. WSEAS transactions on computers, 966-974.

[6] Jaoua, A. H. (2015). Text Categorization Using Hyper Rectangular Keyword Extraction: Application to News Articles Classification. springer Link, 312-325.

[7] JayakumarSadhasivam, R. B. (2019). Sentiment Analysis of Amazon Products Using Ensemble Machine Learning Algorithm. International Journal of Mathematical, Engineering and Management Sciences , 508–520.

[8] KursatUysalSerkanGunal, A. (2014). Text classification using genetic algorithm oriented latent semantic features. Elsevier, 5938-5947.

[9] Loper, S. B. (2009). Natural Language Processing with Python - Analyzing Text with the Natural Language Toolkit. O'Reilly.

[10] Maciejewski, P. S. (2017). Deep learning methods for subject text classification of articles. Federated Conference on Computer Science and Information Systems , 357-360.

[11] MichałMirończukJarosławProtasiewicz, M. (2018). A recent overview of the state-of-the-art elements of text classification. Science Direct, 36-54.

[12] MichałMirończukJarosławProtasiewicz, M. (2018). A recent overview of the state-of-the-art elements of text classification. Science Direct, 36-54.

[13] P.Nelsonad2, M. T. (2011). An exploratory study of a text classification framework for Internet-based surveillance of emerging epidemics. Elsevier, 56-66.

[14] Semberecki, P. a. (2016). Distributed Classification of Text Documents on Apache Spark Platform. 621-630.

[15] Sheikh, J. (2017). Machine Learning, NLP: Text Classification using scikit-learn, python and NLTK. Medium.

[16] urRehmanc, M. K. (2019). A comparative analysis of machine learning models for quality pillar assessment of SaaS services by multi-class text classification of users' reviews. elsevier, 341-371.

[17] Wang, S. a. (2012). Baselines and Bigrams: Simple, Good Sentiment and Topic Classification. Association for Computational Linguistics, 90-94.

[18] Winter, M. K. (2014). Determining if two documents are written by the same author. Journal of the Association for Information Science and Technology, 178-187.

[19] Yang, Y. G. (2012). Hierarchical Text Classification for News Articles Based-on Named Entities. Springer Link, 318-389.

[20] Zengotitabengoa, J. A. (2014). Automatic text classification to support systematic reviews in medicine. Science direct, 1498-1508.

[21] Zhao, L. W. (2012). Improved KNN classification algorithms research in text categorization. nternational Conference on Consumer Electronics, Communications and Networks (CECNet) (pp. 1848-1852). Yichang, China: IEEE.