Sentiment Analysis Using Machine Learning for Twitter

Akshay Chavhan[1], Prof. Sneha A. Khaire[2], Ankit Kumar[3], Saurabh Mate[4], Vipul Thakare[5]
[1,2,3,4,5] *Department of Information Technology, SITRC, Nashik, India*

***Abstract*- With the advancement of Web technology and its growth, there's a large volume of knowledge present within the Web for Internet users and plenty of knowledge is generated too. Social networking sites like Twitter, Facebook, Google+ are rapidly gaining popularity as they permit people to share and express their views about topics, have discussion with different communities, or post messages across the planet. There has been plenty of labor worn out the sphere of sentiment analysis of twitter data. we have mainly thought on sentiment analysis of twitter data which is beneficial to research the information available on the tweets where opinions are highly unstructured, heterogeneous and are may be positive or negative, or neutral accordingly. We've also discussed general challenges and applications of sentiment analysis on Twitter.**

***Index terms*- Social media, Twitter data, Machine learning, Random Forest**

I.INTRODUCTION

In recent times, people are the usage of social networking websites for expressing their sentiments, views, evaluations etc. And the critiques of other peoples have always been crucial to us in many ways. Over 1.five billion registered accounts are there on social media and billions of images and films posts and messages send thru social web sites. Social media is chargeable for generating massive quantity of data.

Twitter History:
Twitter is one of the leading social site in world and ranked at function 4th. Twitter created on March 2006 by Jack Dorsey and his associates at New York University. Twitters headquarter is placed at San Francisco, California, USA. Twitter have extra than 500 million registered customers and 336 monthly lively customers (MAUs).

II. RELATED WORK

A. Sentiment Analysis

Opinion mining (sometimes referred to as sentiment evaluation and emotion AI) refers to the use of herbal language processing, text analysis, computational linguistics and biometrics to systematically identify, extract, quantify, and study active states and subjective information. Sentiment evaluation is widely implemented to voice of the customer fabric such as opinions and survey responses, on line and social media, and fitness care materials for packages that variety from marketing to customer support to clinical medicine
.It aims to determine the mind-set of a speaker, author or other concern with recognize to some subjects or the overall contextual polarity or emotional reaction to a document, interaction or occasion. There are unique strategies for sentiment analysis like NB classifier, SVM algorithm, NBSVM set of rules etc. For the sentiment analysis specific researchers have done one of a kind work in those domains. They might be real time, event like earthquake detection the use of social sensors, occasion summarization, and interpretation of the general public sentiment versions on twitter and so on. These all are the advancement because the time is going on. That is maximum in all likelihood a reason to end up sentimental evaluation extra famous area for research work.

B. Introduction to Problem

Every day massive amount of knowledge is generated by social media users which may be wont to analyze their opinion about any event, movie, product or politics. Conventional tools like Apache Storm analyze stream in micro-batch whereas novel tools like Apache Spark process data in real time making analyzing and processing of real time data possible.

C. Platform and Technologies

There are technologies and tools implemented within the project. These are introduced below. Apache

Spark: it's an open source lightning fast cluster computing platform to retrieve streaming data and forwarding to storage system like HDFS, Database Server. It's built on top of Map Reduce and might integrate well with other Apache software. Apache spark is an in memory fast processing system used for giant scale processing. It's come up as a complicated version of Hadoop. Though it implements the Map Reduce technology but it processes data even 100 times faster by partitioning on memory and 10 times faster on disk across different nodes.

Its structure relies on Resilient Distributed datasets (RDD) which is read only, multi sets of knowledge partitioned and distributed across different node, to make sure fault intolerance and scalability factors. It overcomes the limitation of Map Reduce within which data after reducing was stored into a disk by implementing iterative algorithms who fetch data from multiple datasets during a loop thereby implementing repeated database-style querying of knowledge. During this way, the latency involved is reduced thereby making it faster. RDD is largely an abstraction feature which before processing lays down the execution plan then later depicts computation using Direct Acyclic Graph (DAG).

Further, it's a far better edge over other technologies because it is sort of easy to implement because of multiple available APIs. Also, the opposite benefits include high level libraries. it's not only a High Level Functional but also supports Object Oriented programing language model. This provides it a grip over Java which requires more code to be written for the identical task as compared to Scala. The main success of Scala is that Apache Spark is itself implemented in Scala. Thus, we proceeded with implementation in Scala as compared to Python.

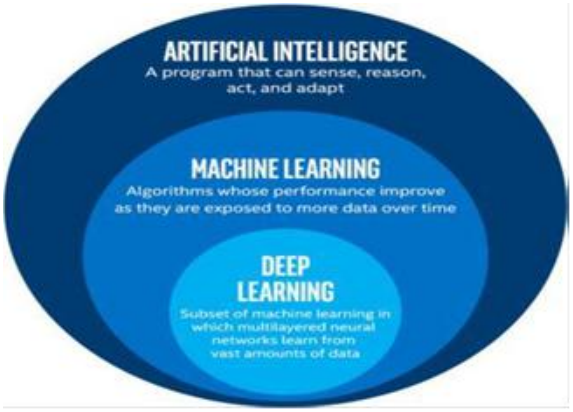


Figure1. Relationship between Artificial Intelligence, Machine Learning and Deep Learning

Data Flow Diagram:

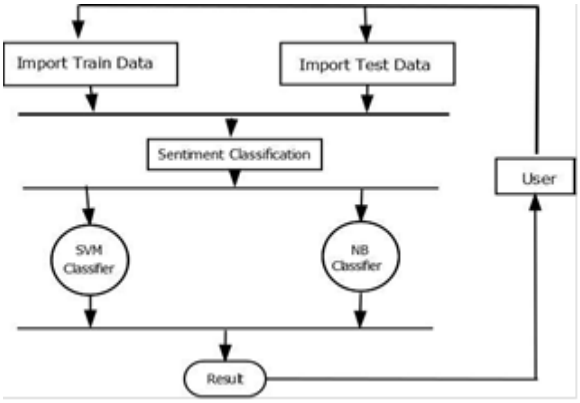


Figure 2: Data Flow Diagram for Sentiment Analysis Using Machine Learning for Twitter

Class Diagram:

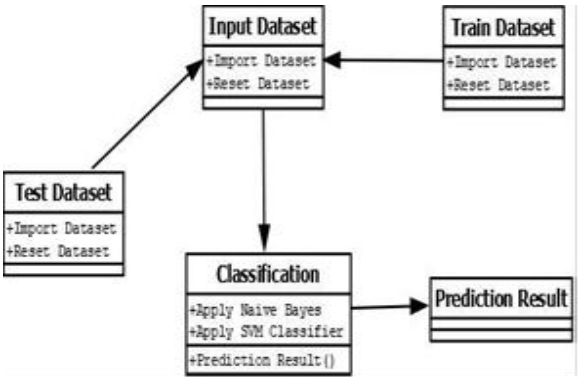


Figure 3: Class Diagram for Sentiment Analysis Using Machine Learning for Twitter

Activity Diagram:

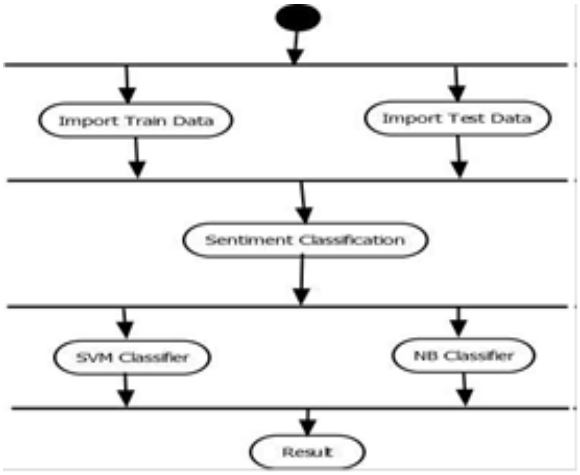


Figure 4:Activity Diagram for Sentiment Analysis Using Machine Learning for Twitter

III. PROPOSED SYSTEM

This section focuses on random forest classifier, hyperparamters of random forest, their impact on accuracy and a few of the features of random forest classifier. Random Forests [14] was the primary paper which brought the concept of ensemble of decision trees which is understood Random Forest, which consists by combining multiple decision trees. While handling the only tree classifier there could also be the matter of noise or outliers which can possibly affect the results of the general classification method, whereas Random Forest is a type of classifier which is very much robust to noise and outliers because of randomness it provides [28]. Random Forest classifier provides two types of randomness, first is with respect to data and second is with respect to features.

Random Forest works as shown below-
Algorithm 1. Random Forest Input: B = Number of Trees, N = Training Data, F = Total- Features, f = Subset of Features Output: Bagged class label for the input data.

1. For each tree available in Forest B:
a) Select a bootstrap sample S for size N from given data. b) Create the tree Tb by recursively repeating the following steps for each internal node of the tree. i. Choose f at random from the F. ii. Select the best among f. iii. Split the node.

2. Once B Trees are created, Test instance will be passed to each tree and class label will be assigned based on majority of votes. Bootstrapping is considered as technique for improving the quality of estimators, in which predefined amount of portion of total dataset will be selected and that will be used for training, so the classifier will not actually get to see the overall data but a small portion of it. Whereas Bagging refers Bootstrap Aggregating which is mainly used to improve the stability and accuracy of classification algorithms, it is mainly used to get rid of variance because single tree is considered to be of high variance but to get rid of that variance number of trees can be combined and the average result of those combined trees will be free from variance.

A. Hyperparameters of Random Forest

As Random Forest is that the combination of decision Trees, it deals with multiple number of hyperparameters which are: • Number of Trees to construct for the choice Forest • Number of features to pick randomly • Depth of every trees. All these hyperparameters are required to be set manually which will be time consuming and does not guarantee that it will give good results for the parameter that we have set manually. First hyperparameter is Number of Trees within the forest, increase in number of trees linearly increase accuracy of the model. Larger the dimensions forest better the accuracy, but the accuracy won't be changed at certain level when even there's a rise in number of trees. Number of features also plays a very important role in classification. Random forest doesn't work on all the features but rather than that there are two values of features which are very famous within the literature and they probably may provide good accuracy results compared to other values of features, but it's worth trying random forest with other values for choosing features randomly. Depth of tree is additionally a really critical hyperparameter in random forest, if smaller value is been selected for Depth then model will suffer from under fitting. More about influence of those hyperparameters are discussed in section 4: Experiments and Results.

B. Features of Random Forest

Random Forest is considered to be an accurate and robust classifier because of following two reasons. • Robust: As Random Forest uses the concept of Bootstrapping, so each tree works on the subset of the whole training data, and because of that each tree is trained on the different value of training data. So that it is very much robust in terms of noise. • Accurate: Random Forest make use of concept of Bagging so as that output of all decision classifier are getting to be averaged, as there is a proof that when infinite number of data is provided to a single classifier then the result will not be consistent, whereas if those data is divided into number of classifier, then averaging of the result of those classifier will be consistent.

IV. EXPERIMENTAL ANALYSIS

Confusion Matrix:

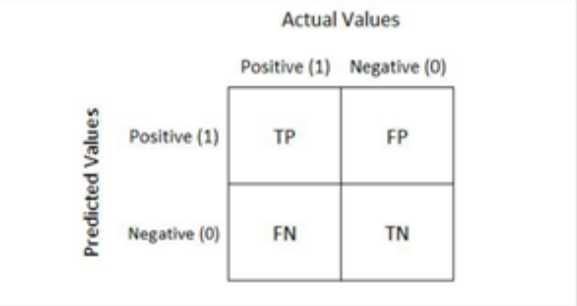


Figure 5: Table for confusion matrix

A confusion matrix might be a summary of prediction results on a classification problem. The amount of correct and incorrect predictions are summarized with count values and lessened by each class. This is often often the key to the confusion matrix.

The confusion matrix shows the ways in which your classification model seems to be confused when it makes starts predictions that gives us the true insight not only into the errors to be made by a classifier but more importantly the categories of errors that are being made to it .

Here,

Class 1: Positive

Class 2: Negative Definition of the Terms:

Positive (P): Observation is positive.

Negative (N): Observation isn't positive.

True Positive (TP): Observation that is positive, and is predicted to be positive.

False Negative (FN): Observation that is positive, but is predicted negative.

True Negative (TN): Observation is negative and it is predicted to be negative. False Positive (FP): Observation is negative but it is predicted positive.

Classification Accuracy:

Classification Rate or Accuracy is given by the relation following

$$\textbf{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

However, there are problems with accuracy. It assumes equal costs for both forms of errors. 99 percent accuracy are going to be excellent, good, mediocre, poor or terrible depending upon the matter.

Recall:

High Recall indicates the category is correctly recognized. Recall is given by the relation:

$$\textbf{Recall} = \frac{TP}{TP + FN}$$

Precision:

To get the price of precision we divide the total number of correctly classified positive examples by the total number of predicted positive examples. Precision is given by the relation:

$$\textbf{Precision} = \frac{TP}{TP + FP}$$

High recall, low precision. this suggests that nearly all of the positive examples are correctly recognized but there are many false positives. Low recall and high precision. This shows that we have been missing many positives but those we predict as positive are indeed positive also.

Let’s consider an example now, within which we've infinite data elements of sophistication B and one element of sophistication A and also the model is predicting class A against all the instances within the test data. Here,

Precision: 0.0

Recall: 1.0

Now:

Arithmetic mean: 0.5

Harmonic mean: 0.0

When taking the mean, it'd have 50 percent correct. Despite being the worst possible outcome! While taking the mean, the F-measure is 0.

Evaluation Result

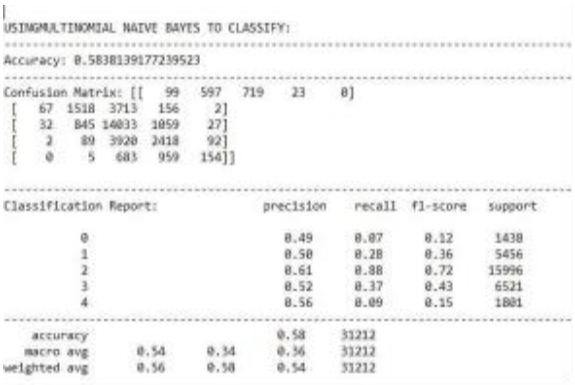
```
USING SVM TO CLASSIFY:
--------------------------------------------------------------------------------
Accuracy: 0.6312315775983596
--------------------------------------------------------------------------------
Confusion Matrix: [[  587   739   281    55     5]
 [  287  2282  2529   244    19]
 [   91  1202 13346  1231    62]
 [   11   234  2879  3128   304]
 [    6    29   247  1015   567]]
--------------------------------------------------------------------------------
Classification Report:              precision    recall  f1-score   support

           0                        0.49      0.27      0.35      1445
           1                        0.51      0.43      0.46      5361
           2                        0.69      0.84      0.76     15912
           3                        0.55      0.47      0.51      6618
           4                        0.56      0.30      0.39      1864

    accuracy                        0.63     31212
   macro avg                        0.56      0.46      0.49     31212
weighted avg                        0.61      0.63      0.61     31212

--------------------------------------------------------------------------------
```

Figure 6: Classification Result Using SVM Classifier



```
USINGMULTINOMIAL NAIVE BAYES TO CLASSIFY:
--------------------------------------------------------------------------------
Accuracy: 0.5838139177239523
--------------------------------------------------------------------------------
Confusion Matrix: [[   99   597   719    23     0]
 [   67  1518  3713   156     2]
 [   32   845 14033  1059    27]
 [    2    89  3920  2418    92]
 [    0     5   683   959   154]]
--------------------------------------------------------------------------------
Classification Report:              precision    recall  f1-score   support

           0                         0.49      0.07      0.12      1438
           1                         0.50      0.28      0.36      5456
           2                         0.61      0.88      0.72     15996
           3                         0.52      0.37      0.43      6521
           4                         0.56      0.09      0.15      1801
--------------------------------------------------------------------------------
    accuracy                         0.58     31212
   macro avg      0.54      0.34      0.36     31212
weighted avg      0.56      0.58      0.54     31212
```

Figure 7: Classification Result Using Naive Bayes Classifier

V. CONCLUSION

It is an effort to implement Sentiment Analysis Using Machine Learning for Twitter. We used two classifiers SVM and Naive Bayes. we've got achieved 63 percent accuracy using SVM and 58 percent accuracy using Naive Bayes classifier. Hence, we conclude that SVM is best classifier to be used for Sentiment Analysis Using Machine Learning for Twitter. From future perspective, we'd wish to extend this project by implementing some machine learning algorithms for applications like election results, product ratings, movies' outcomes and running the project on clusters to expand its functionality. Moreover, we'd wish to make an online application for users to input keywords and find analyzed results. During this project, we've got worked only with unigram models, but we'd wish to extend it to bigram and further which is able to increase linkage between the info and supply accurate sentiment analysis results.

REFERENCES

[1] Sentiment Analysis, available from https://en.wikipedia.org/Sentiment, accessed on 27th October 2018.

[2] Pragya Tripathi, Santosh Kr Vishwakarma, and Ajay Lala, Sentiment Analysis of English Tweet Using Rapidminer, In proc. of International Conference on Computational Intelligence and Communication Net-works, pp. 668-672, 2015.

[3] OKeefe. T and Koprinska I, Feature Selection and Weighting in Sentiment Analysis, In proc. of 14th Australasian Document Computing Symposium, Sydney, Australia, Dec 2009.

[4] K. Bhuvaneshwari and R. Parimala, Correlation Base Feature Selection for Movie Review Sentiment Classification, In proc. of IJARCCE, vol. 5, no. 7, July 2016.

[5] Mangal Singh, Md. Tabrez Nafis, and Neel Mani, Sentiment Analysis and Similarity Evaluation for Heterogeneous-Domain Product Re-views" , In proc. of IJCA, vol. 144, no. 2, June 2016.

[6] Mnahel Ahmed Ibrahim and Naomie Salim, Sentiment Analysis of Arabic Tweets: With Special Reference Restaurant Tweets, In proc. of IJCST, vol. 4, no. 3, pp. 173179, May June 2016.

[7] J. Isabella Dr. R.M.Suresh, Analysis and Evaluation of Feature Selectors in Opinion Mining, In proc. of Indian Journal of Computer Science and Engineering, (ISSN: 0976-5166), Vol., 3 Dec 2012-Jan 2013.

[8] A.Pak and P. Paroubek. Twitter as a Corpus for Sentiment Analysis and Opinion Mining" . In proc. of the Seventh Conference on International Language Resources and Evaluation, pp.1320-1326, 2010.

[9] L. Barbosa, J. Feng. Robust Sentiment Detection on Twitter from Biased and Noisy Data. In proc. of COLING, Volume, pp. 36-44, 2010.

[10] Bifet and E. Frank, Sentiment Knowledge Discovery in Twitter Streaming Data" , In proc. of 13th International Conference on Discovery Science, Berlin , Germany: Springer, pp. 1-15,2010.

[11] Agarwal, B. Xie, I. Vovsha, O. Rambow, R. Passonneau, Sentiment Analysis of Twitter Data" , In proc. of the ACL 2011Workshop on Languages in Social Media, pp. 30-38,2011.

[12] Dmitry Davidov, Ari Rappoport. Enhanced Sentiment Learning Using Twitter Hashtags and Smileys. In proc. of Coling, Volume pages 241, Beijing, August 2010.

[13] Pablo Gamallo, Marcos Garcia, Citius: A Naive-Bayes Strategy for Sentiment Analysis on English Tweets" , In proc. of 8th International Workshop on Semantic Evaluation, Dublin, Ireland, pp 171-175,Aug 2014.

[14] R. Xia, C. Zong, and S. Li, Ensemble of feature sets and classification algorithms for Sentiment classification, In proc. of Information Sciences: an International Journal, vol. 181, no. 6, pp. 11381152, 2011.

[15] Wan, X..A Comparative Study of Cross-Lingual Sentiment Classification. In proc. of the 2012 IEEE/WIC/ACM International Joint Conferences on Web Intelligence and Intelligent Agent Technology, Volume 01 (pp. 24-31).IEEE Computer Society, 2012.

[16] Li, S., Xue, Y., Wang, Z., Zhou, G.. Active learning for cross-domain sentiment classification. In proc. of the Twenty-Third international joint conference on Artificial Intelligence (pp. 2127-2133).AAAI Press, 2013.