International Journal for Research in Engineering Application & Management (IJREAM)
ISSN : 2454-9150 Vol-03, Issue-12, Mar 2018

# TSA: Scalable Machine Learning Online Service for Big-Data Real Time Analysis Using HDFS and Python Jupyter Notebook

**Mr. Mahesh B. Shelke**

**Asst. Prof Department of Computer Science and Engineering, CSMSS CSCOE, Aurangabad, Maharashtra**

***Mahesh_shelke21@hotmail.com***

***Abstract:*** **This work describes a proposal for developing and testing a scalable machine learning design ready to provide real time predictions or analytics as a service over domain-independent massive information, working on the top of Hadoop eco-system and providing real time analytics as a service through API. A systems implementing this design might offer company with on-demand tools facilitating the tasks of storing, analyzing, understanding and reacting to their information, either in batch or stream fashion; and will change into a valuable improvement for increasing the business performance and be a key market during this fast pace surroundings. So as to validate the proposed design, two systems are developed, every one providing classical machine-learning services in several domains: the primary one involves a recommender system for Internet advertisement, whereas the second consists in an exceedingly prediction system that learns from gamers' behavior and tries to predict future events comparable to purchases or churning. Associate analysis is done out on these systems, and results show however each services are able to offer quick responses even once variety of concurrent requests are created, and within the specific case of the second system, results clearly prove that computed predictions considerably outperform those obtained if random guess was used.**

**Keywords: Big Data, HDFS, Representational state Transfer, Jupyter Notebook, and Sentimental Analysis.**

## I. INTRODUCTION

Each day, the amount of data and the number of changing data sources continue to grow. As companies are collecting vast amounts of data from the Internet, their own web sites, social media channels, customer information, call center reports or financial transactions; the need for analytical tools able to leverage knowledge behind all these data is imperative. Even more important than this is the fact that this growth is going to increase exponentially in the future, as there are other emerging areas which are about to come such as Smart Cities and Internet of Things, where the number of potential devices capable of generating high volumes of data is going to be multiplied as a result of what is known as M2M (machine-to-machine interaction). All these incoming changes will require an adequate scaled infrastructure which allows storing, processing and responding to an increasing number of batch and stream requests.

This vast amount of information, if conveniently processed, can reveal relevant insights about each business. For instance, analysis of the former data may serve to predict the upcoming friendships or interests of a social network user suggest related products in which a customer may be interested to purchase or adapt the content and structure of an online course to better fit the students' needs. At this point it is where machine learning techniques can help to analyze big data sources and extract the important trends, links, rules or in other words: knowledge. This field has been studied since the first appearance of the Knowledge Discovery in Databases (KDD) concept, but depending on the data sources and the domains, different approaches and techniques were used, such as association, clustering, classification, prediction, sequential patterns identification, decision trees or, what is more usual, a hybrid approach resulting from a combination of these approaches. However, when a big data framework involves real-time analytics, specific software architecture is needed. Typically, a distinction is made when considering how this data is analyzed with regard to time constraints:

- Batch processing, where a set (typically a very big one) of data is processed to retrieve some statistics of other information. This processing is not required to happen in real-time, as the expected result is not needed within a strong time constraint. This processing is the most adequate for those machine learning techniques or algorithms that require running periodic training and updating processes.

- Stream processing, where new data must be processed in real-time, in many times considering the historical data as well, in order to generate a value. Most often, it involves the use of previously trained models, in order to avoid too much processing and ultimately reduce response times.

Typical examples of batch processing would involve computing trends or extracting patterns from customer's activity during a period that can be categorized per product, website, geographical distribution or user profiles in a social network. For this type of operations, frameworks such as the MapReduce [1] paradigm are suitable, as they enable distributed processing of the data over a cluster of inexpensive nodes. However, additional value can be obtained when stream processing comes into play, as it unveils new possibilities such as providing real-time recommendations to a customer. For instance, these systems can dynamically interact with customers offering specific products and empowering their engagement by means of an accurate prediction of when they are about to leave the site. This prediction enables generating the appropriate events to modify the customer behavior.

## II. CONTRIBUTIONS OF THIS PAPER

This paper describes the implementation of scalable machine learning online services for Big Data with real time data analysis which uses the social media like twitter from which we will be extracting the tweets of various users (like tweets on effects of demonetization from which we can collect the users data whether it has created positive impact or negative impact.) which leads to collecting users opinion on particular product and etc.

## III. WORKING OF SENTIMENTAL ANALYSIS

In the era of Digitization people are so much expressive through the various Medias like social networking sites, blogs, forums, tweets and other various Medias. This type of comments, discussion, views gives the opinions on various topics like effects of demonetization, awareness about social responsibilities, and so on. This gives the opportunity to business, politics, finance and public actions to grow fast by analyzing the activities of user. Just like if user likes any product which he has brought to home and expresses his reviews about that product over social media or e-commerce sites which leads to increase/decrease of sales product depending upon the review of user.

Here we are implementing sentiment analysis for the twitter which gives the positive, negative and neutral reactions, views, and opinions of user.

Following are the some steps for understanding things that can be done using this script:

- Finding out reactions to the News by the users.
- Finding out reviews of product over other competitors.

We use various machine learning classifiers to define perfect sentiment analysis of twitter. And experimentation is completed with some training samples. And we have implemented a classifier using different machine learning algorithm like decision trees, Maximum Entropy and Naïve Byes.

### *Machine Learning Techniques*

We employed classification methods which is polarity based using set of positive, negative and neutral tweets provided by Twitter4j API. Polarity is given by ratio of probability of a word appeared in set of positive or negative statement which makes the word positive or negative. The classifiers we are using are based on the concept of polarity.

Polarity = P (Postive_Words)/P (Total_Words)

P (Negative_Words)/P (Total_Words)

If the feature is independent and based only on Standard English Dictionary then only this technique works. This method fails when we tried to record the sentiment shown with respect to comparison. Further, the polarity based technique also fails to record query related sentiment. In order to fulfil the requirement of classification we involved machine learning techniques.

### *Naïve Bayes*

The Naïve Bayes classifier in one of the simplest probabilistic model works positively on text categorization and employed on Bayes rule with self-supporting feature collection works positively on text categorization and employed on Bayes rule with self-supporting feature collection. It is flexible in way of handling with any number of classes or attributes. For a given tweet d, C* is a class variable which defines the sentiment given by

$$C* = argMaxc(C|D) \quad (2)$$

Bayes Probability PNB(C|D) described as

$PNB\ (C|D) = ((c)\Sigma\ P(f|c)ni(d)\ mi\ =1\ )\ /\ P(d)$ Here, f is feature and ni(d) is feature count found in d, m represents total number of features and P(c) and P(f|c) are found through maximum likelihood estimates.

During classification phase we found a word which was not found in training phase then we will give zero as probability for positive, negative and neutral classes. To end this problem we tend to make probability equal using Laplacian smoothing constant k=1.

$$term\_count+k/Total\_Terms+k|c|$$

## IV. PERFORMANCE MEASURE

To calculate the accuracy of classifier we required measure on which accuracy can be obtained. There are two measures on which accuracy can be dependent: Precision, Recall and Accuracy.

Let's take collection of M documents, MP denotes the number of document which belongs to the true positive class and MN denotes the number of documents which belongs to the true negative class. TP documents had rightly classified whereas FP documents are wrongly classified, similarly FN documents are wrongly classified and TN documents are rightly classified.

**Precision:** It is the ratio of documents of rightly classified under positive prediction class to all documents under positive prediction class.

$$\text{Precision} = TP/TP + FP$$

**Recall:** It is the ratio of documents of rightly classified under positive prediction class to the documents that are positive in the negative prediction class.

$$\text{Recall} = TP/TP+FP$$

| | | True Class: Positive | True Class: Negative |
|---|---|---|---|
| Prediction Class | Positive | True Positive (TP) | False Positive (FP) |
| | Negative | False Negative (FN) | True Negative (TN) |

Fig. Confusion Matrix

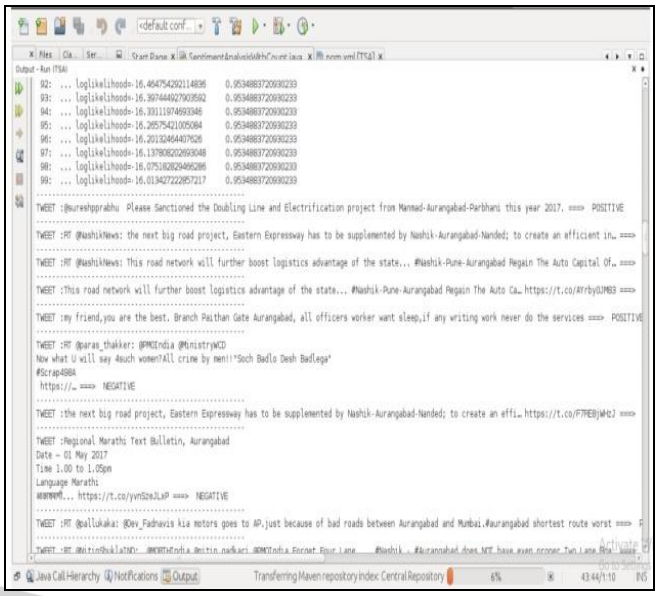
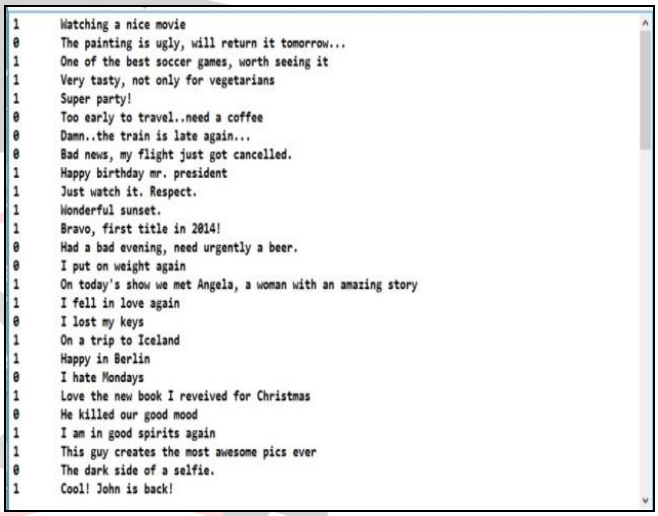


**Figure 1 Shows Tweets Extracted from Twitter**



| | |
|---|---|
| 1 | Watching a nice movie |
| 0 | The painting is ugly, will return it tomorrow... |
| 1 | One of the best soccer games, worth seeing it |
| 1 | Very tasty, not only for vegetarians |
| 1 | Super party! |
| 0 | Too early to travel..need a coffee |
| 0 | Damn..the train is late again... |
| 0 | Bad news, my flight just got cancelled. |
| 1 | Happy birthday mr. president |
| 1 | Just watch it. Respect. |
| 1 | Wonderful sunset. |
| 1 | Bravo, first title in 2014! |
| 0 | Had a bad evening, need urgently a beer. |
| 0 | I put on weight again |
| 1 | On today's show we met Angela, a woman with an amazing story |
| 1 | I fell in love again |
| 0 | I lost my keys |
| 1 | On a trip to Iceland |
| 1 | Happy in Berlin |
| 0 | I hate Mondays |
| 1 | Love the new book I reveived for Christmas |
| 0 | He killed our good mood |
| 1 | I am in good spirits again |
| 1 | This guy creates the most awesome pics ever |
| 0 | The dark side of a selfie. |
| 1 | Cool! John is back! |

**Figure 2 Shows Training Data Set**

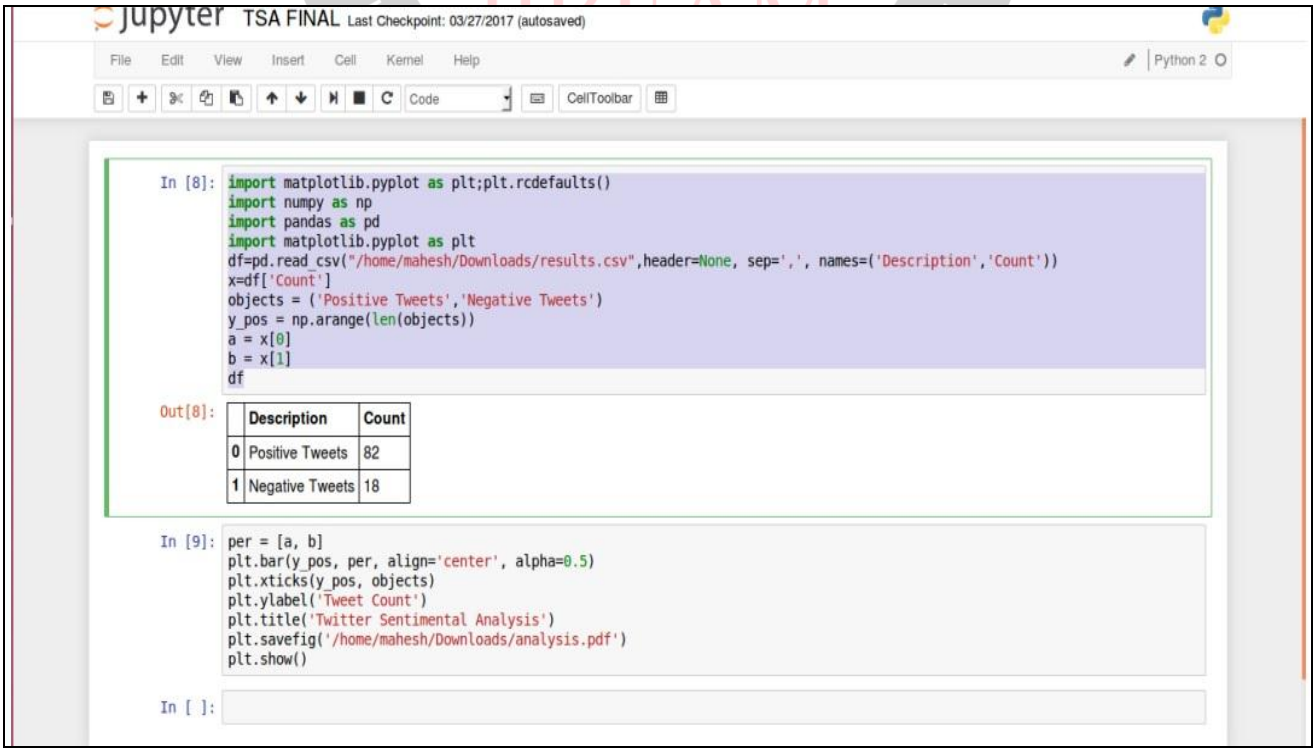



**Figure 3 Shows Loading Data using Jupyter**

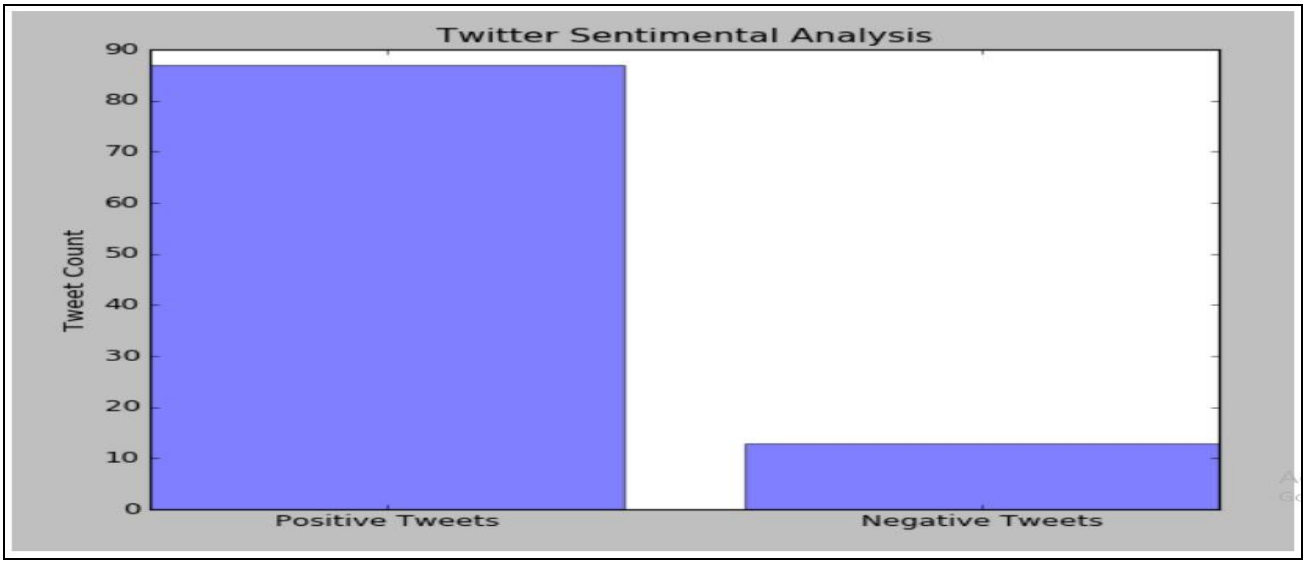



**Figure 4 shows Final Results**

## V. CONCLUSION

In this thesis, we have done comparative analysis on classifier algorithm Naïve Bayes using Python Jupyter Notebook feature. There is need to do sentiment analysis as texts in form of messages or posts to find the whether the sentiment is negative or positive. We had extracted data from twitter i.e. hashtag Aurangabad for sentiment prediction using machine-learning algorithms. First we extracted the data from twitter using twitter API. Then in pre-processing, we clean the data and make the data available to train using classifiers. We have created 100 tweets in training set with positive and negative mark and used with 100 tweets from twitter account. Further we have shown the output in form of bars with number of positive and negative tweets. It is concluded that Naïve Bayes gives best result while working with the limited number of tweets.

## ACKNOWLEDGEMENTS

The completion of my paper, Would not have been so easy without the guidance of various individuals, who gave an appropriate direction for my efforts.

## REFERENCES

[1] B. Pang, L. Lee and S. Vaithyanathan, "Thumbs up?: sentiment classification using machine learning techniques", Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10, pp. 79--86, 2002.

[2] L. Jiang, M. Yu, M. Zhou, X. Liu and T. Zhao, "Target-dependent twitter sentiment classification", Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, vol. 1, pp. 151--160, 2011.

[3] C. Tan, L. Lee, J. Tang, L. Jiang, M. Zhou and P. Li, "User-level sentiment analysis incorporating social networks", Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining, pp. 1397--1405, 2011.

[4] L. Chen, C. Liu "A neural network based approach for sentiment classification in the blogosphere", Journal of Informetrics, vol. 5, no. 2, pp. 313-322, 2011.

[5] M.Anjaria and R. Guddeti, "Influence factor based opinion mining of Twitter data using supervised learning", Communication Systems and Networks (COMSNETS), 2014th International Conference on, pp. 1--8, 2014.