# Recommendation System Through Sentiment Analysis Of Twitter Data

[1]Samika Rastogi, [2]G.R. Smitha
[1]Student, [2]Assistant Professor,
RV College Of Engineering

***Abstract* - Twitter is widely used by several people for networking. It is a platform where people openly express their opinions on any and every subject. The data can be further extracted from twitter to draw meaningful insights on a specific subject. This data, drawn from twitter can be consider a legitimate feedback from the customers or user, regarding a specific product. Customer product reviews play a vital part in the customer's judgement to purchase a product or use a facility. In this paper a recommender system which is constructed on sentiment analysis on online tweets is proposed. The purpose of this system is to create the most accurate recommendation system by also performing sentiment analysis on tweets procured, using relevant keywords described in a config file .The models are also able to make the predictions within an order of a 30 milliseconds which was a very crucial specification for the onboard embedded systems to operate. The models are also able to achieve a considerable level of accuracy of about 90%, which was required for its proper functioning.**

***keywords* - Recommendation System, Sentiment Analysis, Twitter Data**

## I. INTRODUCTION

Twitter is widely used by several people for networking. It is a platform where people openly express their opinions on any and every subject. The data can be further extracted from twitter to draw meaningful insights on a specific subject**.** This data, drawn from twitter can be consider legitimate feedback from the customers or user, regarding a specific product. Customer product reviews play a vital part in the customer's judgement to purchase a product or use a facility. A customer's inclinations and views are also affected by the feedbacks they come across through these social networking platforms. This system proposes a recommender system which is constructed on sentiment analysis on online tweets. The purpose of this system is to create the most accurate recommendation system by also performing sentiment analysis on tweets procured, using relevant keywords. The suggested design is implemented through a python script. The system studies the competitor's product by performing sentimental analysis on the obtained data and observing the accountable user's action and profile.

The system is designed to provide relevant data so as to analyse the scope, popularity of the beneficiary's product and its faults. This data is analysed in order to provide meaningful feedbacks regarding customers. The data procured by the system is vital information regarding the potential customers as well as the existing customers. The sales team then loops in the potential customers. It also highlights how some of the existing customers might be dissatisfied with the beneficiary's product and might now renew their subscription.

The system is designed to provide relevant data so as to analyse the scope, popularity of the beneficiary's product and its faults. This data is analysed in order to provide meaningful feedbacks regarding customers. The analysed data is later sent to the sales team. The data procured by the system is vital information regarding the potential customers as well as the existing customers. The sales team then loops in the potential customers. It also highlights how some of the existing customers might be dissatisfied with the beneficiary's product and might now renew their subscription

## II. METHODOLOGY

The system is a recommendation which operates through sentimental analysis of twitter data of some specific users. This analysis is done using a script developed on python, which uses Text Blob a built-in python library for the purpose of sentiment analysis and cleaning of text[1].

The system is a recommendation which operates through sentimental analysis of twitter data of some specific users. This analysis is done using a script developed on python, which uses Text Blob a built-in python library for the purpose of sentiment analysis and cleaning of text[1]. These specific users are targeted on the basis of a config file holding certain keywords. Whenever any of the keywords mentioned in the config file are identified, that particular tweet is extracted, and a sentiment analysis is performed on that tweet. Once, the twitter data is extracted and processed, the output is stored in 3 tables:

USER
TWEET
WORKPLACE

For further analysis this data, which is stored as CSV files or JSON files are brought to the AWS S3 bucket. This data is then compared with the already existing data in the dataset. Amazon Simple Storage Service is storage for the Internet. It is designed to make web-scale computing easier for developers[2].

The Figure. 3.1 depicts the different actors involved with the recommendation system and also flow of data within the recommendation system, through a use-case diagram. Algorithm used here , for sentiment analysis and for calculating the

polarity for each sentiment is The Naive Bayes algorithm. TextBlob is a Python (2 and 3) library for processing textual data[3]. It provides a simple API for diving into common natural language processing (NLP) tasks such as part-of-speech tagging, noun phrase extraction, sentiment analysis, classification, translation, and more.


**Figure. 1.1 Methodology**

## III. DESIGN

Design plays a vital role in giving an overview of the implementation of the complete model designed.Various modules are tested implemented and brought in together to give an entire system .Providing a high-level design overview of the model explains the basic working of all the sub-modules .Using Use Case diagrams , we can get the interaction between different actors involved with the system as well as their interaction with the system .This diagram also elaborates on the internal functioning of the system.

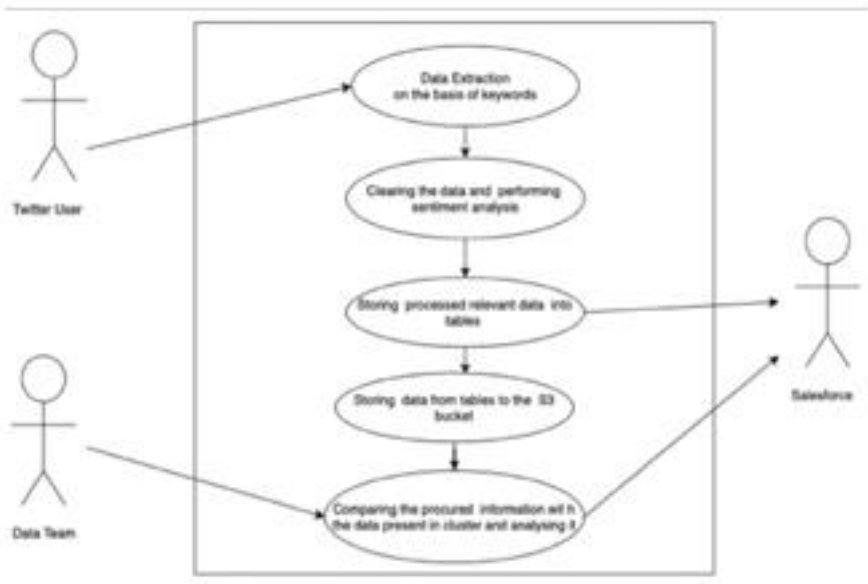


**Figure. 3.1 Use Case Diagram for Recommendation System**

The input here are the tweets extracted using keywords and TextBlob(to cleanse the text and for performing sentiment analysis).Within the system, this input is further categorized into 3 different tables . This data originally stored in csv as well as JSON files can is further received by the data engineers. These engineers decide, the relevant data which is to be sent to the cluster for comparison with the existing database. One the data is analyzed it is sent to the sales team to draw meaningful insights from it. Figure. 3.1 depicts the different actors involved with the recommendation system and also flow of data within the recommendation system, through a use-case diagram. The internal working of each of the modules is explained in the detailed design. It also describes the software component and subcomponent of the system.

This Module is responsible for pre-processing the entire dataset and partitioning it into training, testing and validation dataset.The flowchart shown in the Figure 3.2 explains the Recommendation system. The module starts by extracting the data on the basis of keywords. This data is then fed into a Python based library – Text Blob. The data is later cleansed, and sentiment polarity is assigned to it using Naïve Bayes Algorithm.


**Figure. 3.2 Pre-processing Module flowchart**

The training module explains how data is finally getting captured by the target tables. The training module is explains the entire the workflow via a detailed flowchart in Figure.3.3.

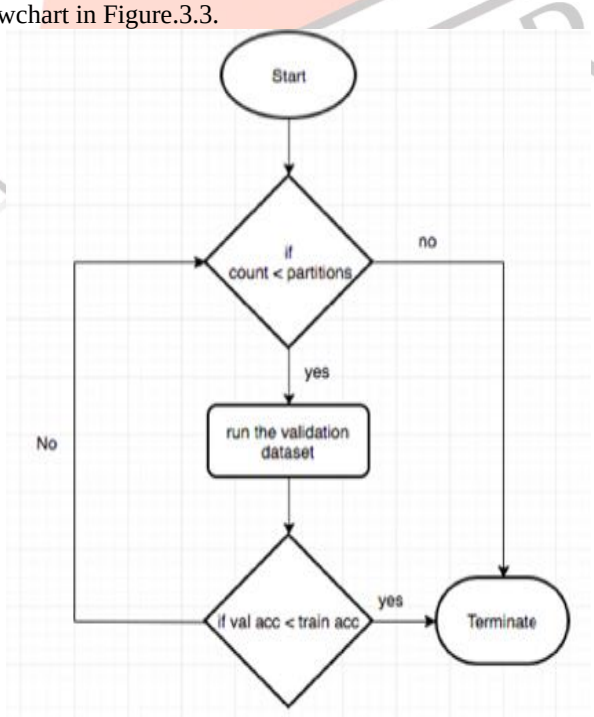


**Figure 3.3. Training Module flowchart**

The third and the final module explains the flow of events in the validation module, meant for testing the entire framework after development. The flowchart shown in the Figure.3.3 explains the procedure followed in the validation module. The trained

datasets are divided into partitions. The validation module validates every partition until the training accuracy achieved is more than the validation accuracy.

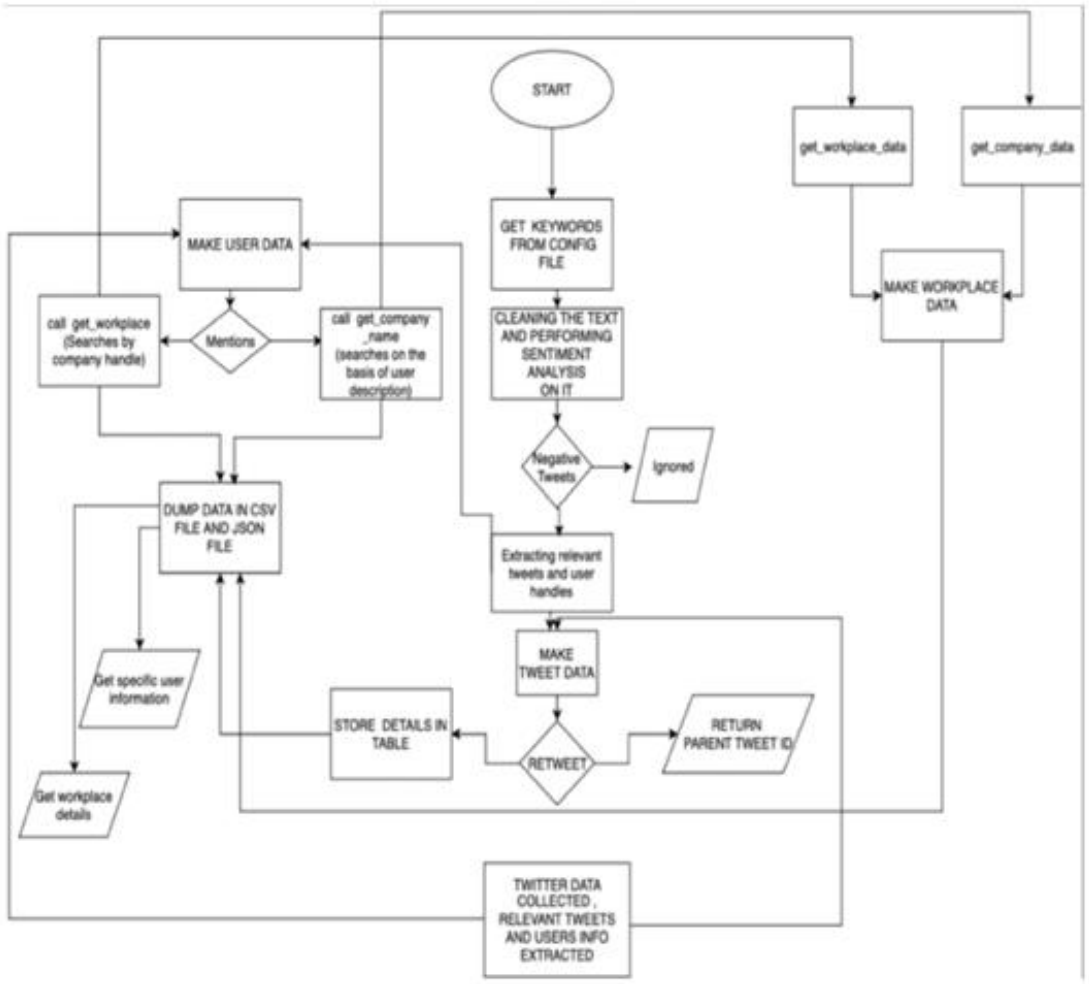


**Figure. 3.3 Validation Module flowchart**

## IV. VALIDATION AND VERIFICATION

To test the different modules , different testing techniques for the module need to be employed [4] .The basic testing framework used to validate each module individually is by performing basic test on each module .Basic testing includes checks null checks , checking the data for unwanted special characters , verification of the individual columns as per the customized requirements .After performing the basic tests on each module of the system , functional tests or integration testing needs to be performed to determine the interaction between all the modules . The functional tests basically compare the source table's data with the target's data [5]. It is performed to make sure that target's data is in consistence with the source's table. The functional tests generate logs giving the mismatches between the source and the target .

Once it is verified if the individual models functioning properly or not and the if interaction between each module is as desired , a final system test performed by combining all the modules [6].The system tests verify the final data .The system check in this scenario is a type of a data check[7].

## V. EVALUATION METRICS AND EXPERIMENTAL DATASET

Evaluation metrics are the criteria for testing different algorithms. The behaviour of the algorithms or techniques can be determined using these metrics. Some techniques satisfy some of the metrics. In this project, the outputs that are obtained from the different inputs given to the system are compared with the ideal output as per requirements to check whether the metrics are satisfied. These are suitable measures of performance since the two important evaluation criteria for a space application is accuracy and computation time. The experimental datasets used thus comprised of various possible inputs that could be given to the system [8]. The robustness of the model is highly critical in this scenario since the procured information is then supplied to the sales team, working to loop in more customers and making sure that existing customers end up renewing their subscription. This mandates that the experimental data set should be an accurate analogy of the actual data to be obtained. The output data is essentially stored in 3 tables, sentimental analysis is performed on the extracted data. Every time the python script is run, it creates partitions for that particular date as well as the time. Data is stored in these partitions as CSV files as well as JSON files.

The results obtained consisted of all the results expected at the beginning of the project. The tweets here, are extracted on the basis of the twitter id from the first tweet extracted in that week. Every time the basic python script runs, it provides the last

twitter id which is extracted in that particular week and this id is also stored in a config file. Once, the python script is initiated, an array is created, and all the relevant tweets made since the twitter id in the config file are extracted.
The extracted data is then stored into tables after predicting the false positives as well as the false negatives.
The LSTM model also worked on the same lines, though with 11% lesser accuracy. Thus, the system is overall in an efficient and ideal functional flow. The results obtained consisted of all the results expected at the beginning of the project. The trajectory prediction worked with no errors and an acceptable accuracy which is high with a fast processing time.

***Conclusion***

In this paper we have described a user recommender system for Twitter. Our work emphasizes the use of implicit sentiment analysis in order to improve the performance of the recommendation process.

Although the working system is efficient enough as per requirements, there are a few cases of usage which will lead to loopholes in the functioning. Some of these limitations of the project are:

- Data regarding a specific feature of a specific product, generated by the model deployed isn't 100% accurate. Since the issue is being resolved on the basis of hardcoded keywords, this system is unable to extract feedback regarding specific products. This issue can be resolved by using some RNN models such as LSTM
- The system is also incapable of recognizing the sentiment behind slang words and hence doesn't provide a distinctively positive or a distinctively negative sentiment polarity to the tweets holding such words.
- Since , twitter data is being used for various analytical purposes , it can be assumed that validation of a user's workplace can't be done accurately .While , in most cases the system is capable of predicting the users workplace , provided they mention it on their profile but in some cases where workplace is mentioned in the description field , the system's prediction isn't entirely reliable .
- This model can, at a time fetch data for a week. Later, this data is stored into incremental tables (in s3 bucket). But since, the free Twitter API is being used, it only permits a person, holding the twitter developer's account, to fetch data for a week every time the path is called. Besides, this system doesn't allow extraction of data, which is older than a week, at any particular time.

To overcome the limitations of the project, steps will be taken up in the future. The time required to process a prediction has to be optimized for practical use in devices with low processing capacity. New sentiment analysis methods can be used [9]. More than 1 week of relevant data should be fetched from last tweet onwards. Creating a dashboard suggesting negative as well as positive tweets along with their user handle and also reflecting polarity of all these sentiments against each tweet.Scheduling an OOZIE job to fetch the data on a regular basis and populating it in the s3 bucket, into incremental tables [10].

This section gives an overlook of the entire recommendation system and briefly states the limitation of the project and future enhancements to grow and overcome the limitations of the system built during the course of the project.

**REFERENCES**

[1] F. Abel, Q. Gao, G.-J. Houben, and K. Tao. Analyzing temporal dynamics in twitter profiles for personalized recommendations in the social web. In Proceedings of ACM WebSci '11, 3rd International Conference on Web Science, Koblenz, Germany. ACM, June 2011.
[2] A. Agarwal, B. Xie, I. Vovsha, O. Rambow, and R. Passonneau. Sentiment analysis of twitter data. In Proceedings of the Workshop on Languages in Social Media, LSM '11, pages 30–38, Stroudsburg, PA, USA, 2011. Association for Computational Linguistics.
[3] G. Arru, D. Feltoni Gurini, F. Gasparetti, A. Micarelli, and G. Sansonetti. Signal-based user recommendation on twitter. Social Recommender Systems 2013, 2013.
[4] M. Bank and J. Franke. Social networks as data source for recommendation systems. In F. Buccafurri and G. Semeraro, editors, E-Commerce and Web Technologies, volume 61 of Lecture Notes in Business Information Processing, pages 49–60. Springer Berlin Heidelberg, 2010.
[5] C. Biancalana, F. Gasparetti, A. Micarelli, and G. Sansonetti. An approach to social recommendation for context-aware mobile services. ACM Trans. Intell. Syst. Technol., 4(1):10:1–10:31, Feb. 2013.
[6] C. Biancalana, F. Gasparetti, A. Micarelli, and G. Sansonetti. Social semantic query expansion. ACM Trans. Intell. Syst. Technol., 4(4), 2013. Forthcoming issue.
[7] C. Biancalana and A. Micarelli. Social tagging in query expansion: A new way for personalized web search. In CSE (4), pages 1060–1065. IEEE Computer Society, 2009.
[8] J. Chen, W. Geyer, C. Dugan, M. Muller, and I. Guy. Make new friends, but keep the old: recommending people on social networking sites. In Proceedings of the 27th International Conference on Human Factors in Computing Systems, CHI '09, pages 201–210, New York, NY, USA, 2009. ACM.
[9] S. Faridani. Using canonical correlation analysis for generalized sentiment analysis, product recommendation and search. In Proceedings of the fifth ACM Conference on Recommender systems, RecSys '11, pages 355–358, New York, NY, USA, 2011. ACM.
[10] J. Freyne, M. Jacovi, I. Guy, and W. Geyer. Increasing engagement through early recommender intervention. In Proceedings of the third ACM Conference on Recommender Systems, RecSys '09, pages 85–92, New York, NY, USA, 2009. ACM.