International Journal of Scientific Research & Engineering Trends
Volume 6, Issue 1, Jan-Feb-2020, ISSN (Online): 2395-566X

Sentimental Analysis on Amazon Fine Food Reviews

Kartikay Thakkar, Sidharth Sharma, Ujjwal Chhabra, Asst. Prof. Ms Charu Gupta
Bhagwan Parshuram Institute of Technology
Guru Gobind Singh Indraprastha University(GGSIPU)
Delhi, India

***Abstract* -In the era of social media and internet, sentimental analysis is extremely useful in social media monitoring as it allows us to gain an overview of the wider public opinion behind certain topics. Our research is focused on to make prediction model where we will be able to predict whether a recommendation is positive or negative. in this analysis , we will focus on score as well as positive/negative sentiment of the recommendation. The analysis proves that logistic algorithm provides the best sentimental analysis result. Our results were further verified by the amazing accuracy of the Logistic regression classifier as well.**

***Keywords*- Machine Learning, Natural Language Processing, Deep Learning, Text Mining, Sentimental Analysis, Data Analysis,Artificial Intelligence.**

I. INTRODUCTION

As with all sectors of machine learning, innovation around sentiment analysis is happening at a lightning pace and the scope for its use is vast. As an extremely valuable tool for social media companies, business owners and advertisers, sentiment analysis is already providing insights that help drive effective business decisions, strategies and objectives across a range of sectors. These insights range from the analysis of reviews of your brand and the competition to comparison of your product's reception in new, international markets.

This paper aims at analyzing an answer for the sentiment classification at a fine-grained level, specifically the sentence level during which polarity of the sentence may be given by three categories as positive, negative and neutral. In this work, the goal is to predict the score of food reviews on a scale of 1 to 5 with two recurrent neural networks that are carefully tuned. As for baseline, we train a simple RNN for classification. Then we extend the baseline to modified RNN and GRU. In addition, we present two different methods to deal with highly skewed data, which is a common problem for reviews.

Models are evaluated using accuracies. We've looked at one such popular microblog called Reviews and build models for classifying "reviews" into positive, negative sentiment. We build models for two classification tasks: a binary task of classifying sentiment into positive and negative classes and a 3-way task of classifying sentiment into positive, negative and neutral classes. We experiment with three types of models: unigram model, a feature based model and a tree kernel based model. For the feature based model we use some of the features proposed In past literature and propose new features. For the tree kernel based model we design a new tree representation for reviews. We use a unigram model, previously shown to work well for sentiment analysis for reviews data, as our baseline. Our experiments show that a unigram model is indeed a hard baseline achieving over 20% over the chance baseline for both classification tasks. Our feature based model that uses only 100 features achieves similar accuracy as the unigram model that uses over 10,000 features. Our tree kernel based model outperforms both these models by a significant margin.

We also experiment with a combination of models: combining unigrams with our features and combining our features with the tree kernel. Both these combinations outperform the unigram baseline by over 4% for both classification tasks. In this paper, we present extensive feature analysis of the 100 features we propose. Our experiments show that features that have to do with review-specific features (emoticons, hashtags etc.) add value to the classifier but only marginally. Features that combine prior polarity of words with their parts-of-speech tags are most important for both the classification tasks.

Thus, we see that standard natural language processing tools are useful even in a genre which is quite different from the genre on which they were trained (newswire). Furthermore, we also show that the tree kernel model performs roughly as well as the best feature based models, even though it does not require detailed feature engineering. We use manually annotated reviews data for our experiments. One advantage of this data, over previously used data-sets, is that the reviews are collected in a streaming fashion and therefore represent a true sample of actual reviews in terms of language use and

content. Our new data set is available to other researchers.We introduce two resources which are available i.e Hand annotated dictionary for emoticons and an acronym dictionary collected from the web with English translations of over 5000 frequently used acronyms.

II. DATASET

The dataset was picked from kaggle [1] and given a review , we have to determine whether the review is positive (4 or 5) or negative(1 or 2). The data span over 16 years, including all ~500,000 reviews from Oct 1997 up to October 2012. Reviews include product and user information, ratings, and a plain text review. It also includes reviews from all other Amazon categories. All of the data is in 2 files, Train and Test.

1. Train.csv contains 5 columns: ProductId, Time, Title, Summary, Text
2. Test.csv contains the same columns which we have to predict.
3. Size of Train.csv: 6.75GB
4. Size of Test.csv: 2GB
5. Number of rows in Train.csv = 568,454

1. Data includes

1. Reviews from Oct 1997 - Oct 2012
2. 568,454 reviews
3. 256,059 users
4. 74,258 products
5. 260 users with > 50 reviews

2. Attribute Information

1. Id
2. ProductId - unique identifier for the product
3. UserId - unqiue identifier for the user
4. ProfileName
5. HelpfulnessNumerator - number of users who found the review helpful
6. HelpfulnessDenominator - number of users who indicated whether they found the review helpful or no
7. Score - rating between 1 and 5
8. Time - timestamp for the review
9. Summary - brief summary of the review
10. Text - text of the review

III. PROPOSED METHODOLOGY

Fig.1. Word Cloud of the entire dataset.

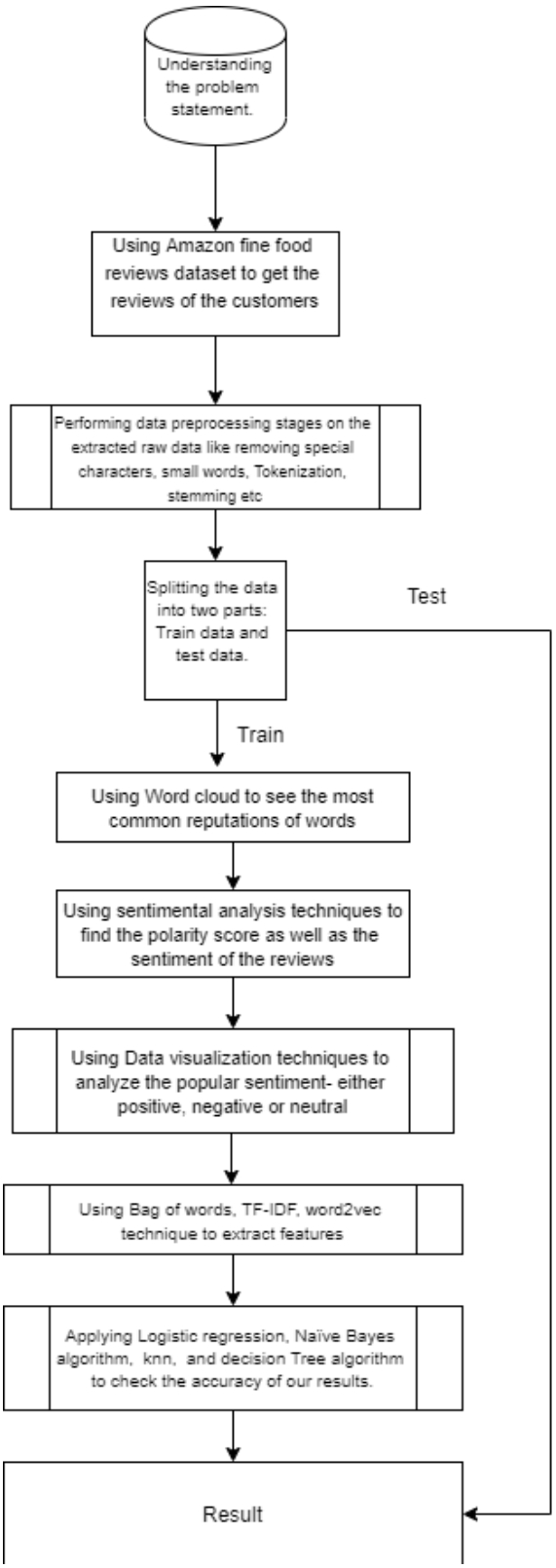

Fig 2. Flow chart of the approach used

International Journal of Scientific Research & Engineering Trends
Volume 6, Issue 1, Jan-Feb-2020, ISSN (Online): 2395-566X

Fig.3 . Word Cloud of the Positive Reviews.

Fig.4 Word Cloud of the Negative Reviews.

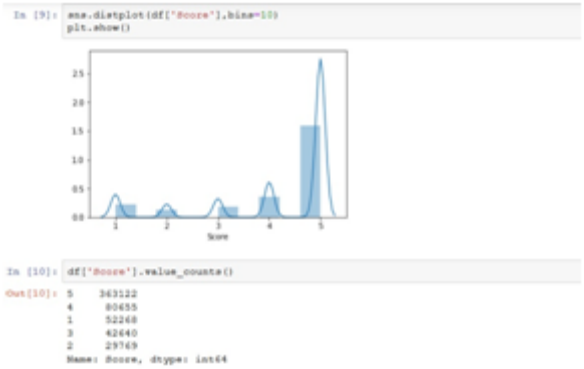

Fig.5 Seaborn distibution plot on the dataset.

1. Data PreProcessing

Data pre-processing could be a very crucial step in our analysis because it can have a grave impact on the results. An unprocessed dataset can cause wrong results and may ruin the analysis; therefore, it's necessary to pre-process the information before applying any data mining operation. In data preprocessing we tend to remove the unwanted tags, web links and special symbols (@ # ^ * "/ : >, < \ |?), that may lead to wrong results. The following procedure was followed to process the data:

1. Deduplication of data.
2. Removing Reviews Handles
3. Removing Punctuations, Numbers, Special Characters
4. Removing Stop Words
5. Tokenization
6. Stemming

2. Feature Extraction

A couple different techniques were used in this paper to extract features namely bag of words, tf-idf, and TNSE word2vec. Bag of words- A unigram of BOW was made with 2 different perplexity to build the vocabulary for machine learning algos to check accuracy. The unigram with a perplexity of 30 is shown in fig 2.1 and fig 2.2 with a perplexity of 50**.**

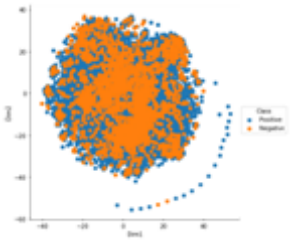

Fig.6 BOW Perplexity=30.

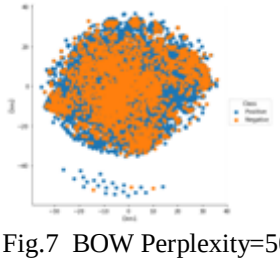

Fig.7 BOW Perplexity=50.

Further a bigram of BOW was also calculated for both test and training data and AUC was plotted. Rests are shown in fig 2.3

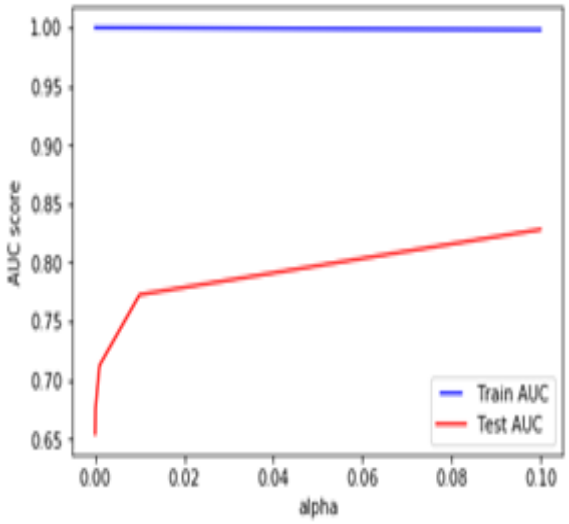

Fig.8 AUC curve on the BOW bigram.

Later in this paper the accuracy of feature extracted using BOW is shown on decision tree algo.

TF-IDF: Similar to BOW two different perplexities 20 & 30 with 10K points each were plotted which are shown in Fig.6 and Fig 7 respectively.

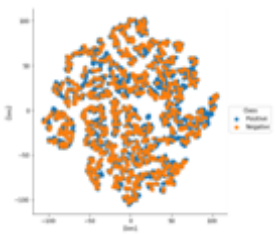

Fig.8 . TFIDF with perplexity 20.

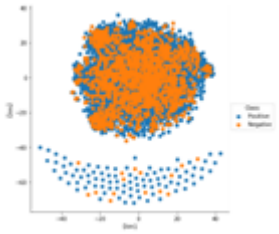

Fig .9 . TFIDF with perplexity 30.

TNSE word2vec:TNSE word to vector was used to find similar subsets of the same word and graph was plotted which can be seen in Fig 8.

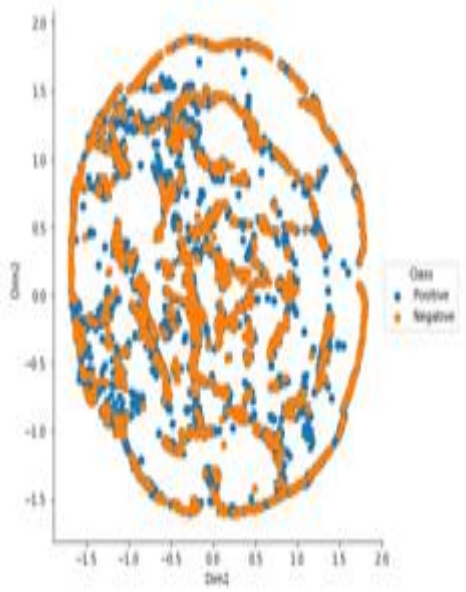

Fig.10. TSNE on the word2vec.

Conclusions from TSNE plots Most of TSNE plot shows that data is quite overlapping hence we can't be sure that data is linearly separable but as TSNE is an approximation algorithm we can't be sure for this claim Hence we need to make models and test it ourselves If the data from the TSNE plot would had been seen to separable it would have easily separable using any linear model.

IV. RESULTS

We applied four vectorization techniqueson the dataset The goal of using a Machine Learning algorithms is to create a training model that can be used to predict the class or value of the target variable by learning simple decision rules inferred from prior data(training data). BOW on Uni-gram, bi-gram and tfidf would have taken forever if had taken all the dimensions as it had huge dimension and hence tried with max 300 as max_depth Bi-gram Featurization (max_depth=73) gave the best results with accuracy of 85.11% and F1-score of 0.513 Plotted feature importance for Uni-gram, bi-gram and tfidf but not for Avg Word2Vec and Tfidf Avg Word2Vec as Word2Vec featurizations are highly correlated hence can't directly get the feature importance. Fig. 11 shows feature importance of on BOW unigram using decision trees, Fig. 12 on bigram BOW, and Fig. 4.3 on TF-IDF.

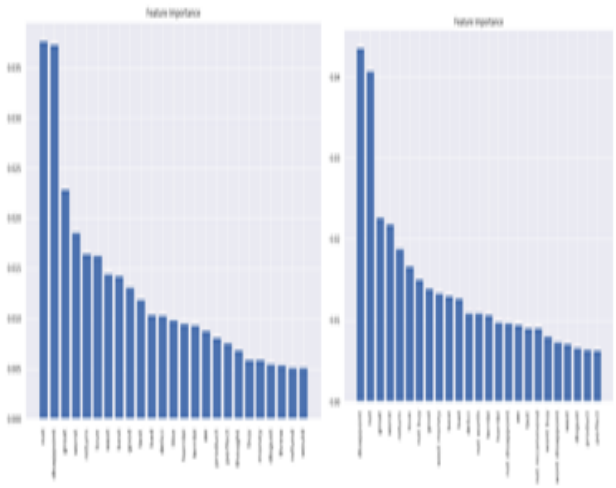

Fig. 11. BOW unigram using decision trees Fig.12 BOW bigram.

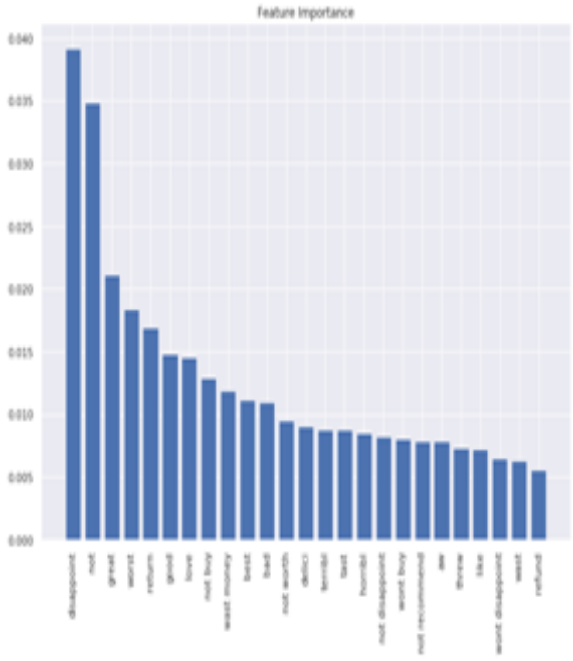

Fig. 13 Feature importance on the dataset.

KNN: Best Accuracy of 85.107% is achieved by avg. Word2Vec Featurization. The kd-tree and brute implementation of KNN gives relatively similar results KNN is a very slow Algorithm compared to others takes a

lot of time to train. KNN did not fair in terms of precision and F1-score. Overall KNN was not that good for this dataset. Fig. 13 shows accuracy of KNN on test set and plots a heatmap.

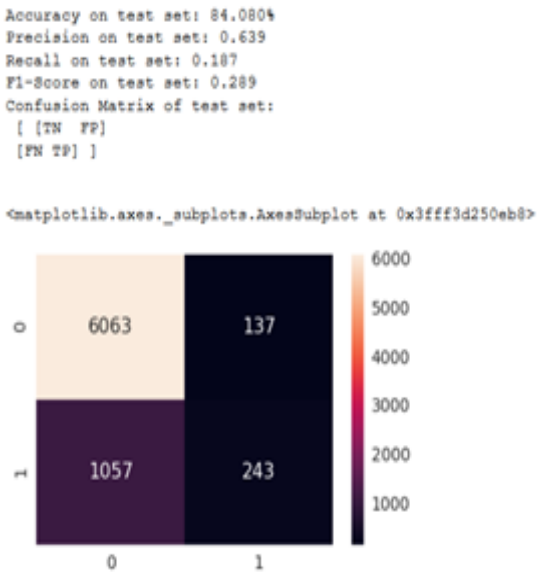

Fig.13 HeatMap on the KNN avg word2vec.

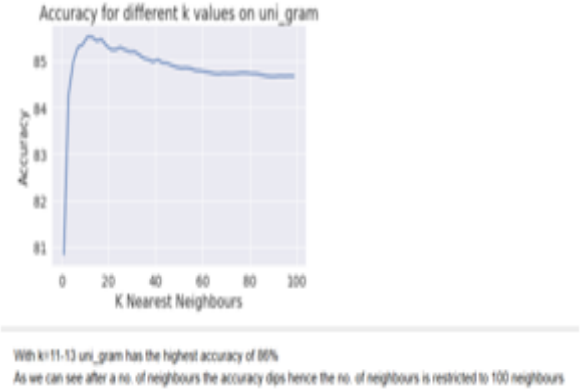

Fig.14 Accuracy for different k values.

Random Forest: The following steps were followed and conclusions :
Step 1 : Text Preprocessing
Step 2: Time-based splitting of whole dataset into train_data and test_data.
Step 3: Training the vectorizer on train_data and later applying same vectorizer on both train_data and test_data to transform them into vectors
Step 4: Using Random Forest as an estimator in GridSearchCV in order to find optimal value of base_learners.
Step 5: Once , we get optimal value of base_learners then train Random Forest again with this optimal value of base_learners and make predictions on test_data
Step 6: Draw Cross_Validation Error VS Base_Learners(n_estimators) plot.
Step 7 : Evaluate : Accuracy , F1-Score , Precision , Recall
Step 8: Draw Seaborn Heatmap for Confusion Matrix .
Step 9: Using GBDT as an estimator in GridSearchCV in order to find optimal value of base_learners , depth and learning_rate.
Step 10: Once , we get optimal values (of base_learners ,depth and learning_rate) , then train GBDT again with thess optimal values (of base_learners , depth , learning_rate) and after that make predictions on test_data
Step 11: Draw Cross-Validation Error vs tuples of (Learning_rate,Max_depth,N_estimators) graph
Step 12: Evaluate : Accuracy ,F1-Score, Precision, Recall
Step 13: Draw Seaborn Heatmap for Confusion Matrix
Repeat from Step 3 to Step 13 for each of these four vectorizers: Bag Of Words(BoW)(Fig. 14), TFIDF(Fig. 15), Avg Word2Vec(Fig. 4.8) and TFIDF Word2Vec(Fig. 15)

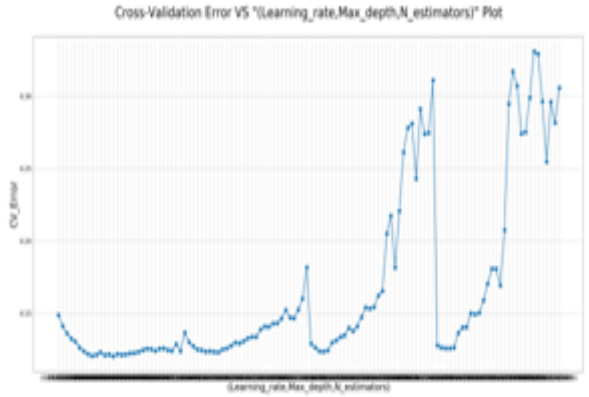

Fig.15 Cross validation Error vs (Learning Rate,Max depth , N_estimators) Plot.

Table 1 KNN.

0	Vectorization	Algorithm	Accuracy	F1 Score
1.	Unigram	Brute kd-tree	84.08 84.347	0.71 0.668
2.	Bi-gram	Brute kd-tree	84.613 84.613	0.723 0.702
3.	Tfidf	Brute kd-tree	84.48 84.48	0.743 0.732
4.	Avg word2vec	Brute kd-tree	85.107 85.107	0.695 0.609
5.	Tfidf-word2vec	Brute kd-tree	84.92 84.92	0.586 0.5879

Table 2 Logistic Regression.

S.No	Featurization	Cv	Accuracy
1.	Unigram	Gridsearch Cv Randomizedsearch Cv	91.95 91.96
2.	Bi-Gram	Gridsearch Cv Randomizedsearch Cv	93.704 93.704
3.	Tfidf	Gridsearch Cv Randomizedsearch Cv	93.615 93.66
4	Avg Word2vec	Gridsearch Cv Randomizedsearch Cv	89.28 89.264
5.	Tfidf-Word2vec	Gridsearch Cv Randomizedsearch Cv	88.027 88.093

Table 3 SVMTable

S.No	Vectori-Zation	Cv	ACCURACY	F1-Score
1.	Unigram	Gridsearch Cv Randomized Search Cv	90.46 89.96	0.7 0.688
2.	Bi-Gram	Gridsearch Cv Randomized Search Cv	90.64 90.707	0.7 0.702
3.	Tfidf	Gridsearch Cv Randomized Search Cv	91.667 91.64	0.733 0.732
4.	Avg Word2vec	Gridsearch Cv Randomized Search Cv	89.57 88.8	0.645 0.609
5.	Tfidf-Word2vec	Gridsearch Cv Randomized Search Cv	88.707 88.84	0.546 0.579

Table 4 Decision Tree.

S.No	Vectorization	Accuracy	F1-Score
1.	Unigram	83.42	0.485
2.	Bi-Gram	85.11	0.513
3.	Tfidf	84.46	0.529
4	Avg Word2vec	79.38	0.355
5.	Tfidf-Word2vec	77.09	0.304

Table 5 Naïve Bayes

S.no	Model	Accuracy	F1 - score
1.	Bernoullinb For Bow	89.36	0.7
2.	Multinomialnb For Bow	90.19	0.702
3.	Bernoullinb For Tfidf	89.36	0.733
4.	Multinomialnb For Bow	88.04	0.609

Table 6 Naïve Bayes

S.NO	FEATURIZATION	CV	ACCURACY
1.	UNIGRAM	GRIDSEARCH CV RANDOMIZEDSEARCH CV	88.79 88.35
3.	TFIDF	GRIDSEARCH CV RANDOMIZEDSEARCH CV	88.35
4	AVG WORD2VEC	GRIDSEARCH CV RANDOMIZEDSEARCH CV	89.13
5.	TFIDF-WORD2VEC	GRIDSEARCH CV RANDOMIZEDSEARCH CV	67.73

Table 7 LSTM

S.NO	VECTORIZATION	ACCURACY
1.	RNN With 1 LSTM Layer	0.919315860009658
2.	RNN With 2 LSTM Layers	0.919661884570 5999
3.	RNN With 3 LSTM Layers	0.921716062130 6325
4	RNN With 4 LSTM Layers	0.921792956479 109

V. CONCLUSION

Using machine learning supervised approach helps to obtain the results.Logistic Regression,naïve Bayes gave the best accuracies in machine learning models.whereas LSTM ran the battle since it's a deep learning model and gave an accuracy of 92.1% .There were sentiments of all sorts but the overwhelming majority had a positive sentiment. Our results were further verified by the amazing accuracy of the Logistic regression classifier as well.

REFERENCES

[1]. https://www.kaggle.com/snap/amazon-fine-food-reviews
[2]. Hutto, C. J. & Gilbert, E. (2014). VADER: A Parsimonious Rule-based Model for Sentiment Analysis of Social Media Text. AAAI 2014

[3]. Liu, B. (2012). Sentiment Analysis and Opinion Mining. San Rafael, CA: Morgan & Claypool.
[4]. Pang, B., & Lee, L. (2008). Opinion mining and sentiment analysis. Foundations & Trends in Information Retrieval, 2(1), 1–135.
[5]. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V. and Vanderplas, J., 2011. Scikit-learn: Machine learning in Python.Journal of machine learning research,12(Oct), pp.2825-2830
[6]. Lilleberg, J., Zhu, Y. and Zhang, Y., 2015, July. Support vector machines and word2vec for text classification with semantic features. In2015 IEEE14th International Conference on Cognitive Informatics & Cognitive Computing (ICCI* CC)(pp. 136-140). IEEE.
[7]. Rehurek, R. and Sojka, P., 2010. Software framework for topic modelling with large corpora. In Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks.pp. 45-50. Valletta, Malta, May 2010. ELRA.
[8]. Rong, X., 2014. word2vec parameter learning explained.arXiv preprint arXiv:1411.2738.
[9]. Zahra Nazari, Dongshik Kang & M.Reza Asharif "A New Hierarchical Clustering Algorithm" ICIIBMS 2015, Track2: Artificial Intelligence, Robotics, and Human-Computer Interaction, Okinawa, Japan
[10]. Prafulla Bafna, Dhanya Pramod and Anagha Vaidya "Document Clustering: TF-IDF approach"International Conferenceon Electrical,Electronics, and OptimizationTechniques (ICEEOT) -2016 University)"
[11]. Kim, S. M., & Hovy, E. (2004, August). Determining the sentiment of opinions. In Proceedings of the international conference on Computational Linguistics (p.1367).Association for Computational Linguistics.
[12]. McAuley, J., & Leskovec, J. (2013). Hidden Factors and Hidden Topics: Understanding Rating Dimensions with Review Text. In Proceedings of the 7th ACM Conference on Recommender Systems (pp. 165-172). New York, NY, USA:ACM. doi:10.1145/2507157.2507163.
[13]. G. Ganu, N. Elhadad, and A. Marian. Beyond the stars: Improving rating predictions using review text content. In WebDB, 2009.
[14]. Frank and M. Hall. Additive regression applied to a large-scale collaborative filtering problem. In AI 2008: Advances in Artificial Intelligence, pages 435-446. Springer, 2008.