Sentiment Analysis on Reviews

Manish Mondal
Department of Computer Science
University of Central Missouri
Warrensburg, Missouri 64093, USA
mxm38640@ucmo.edu

Hardik Bharatkumar Somaiya
Department of Computer Science
University of Central Missouri
Warrensburg, Missouri 64093, USA
hxs68380@ucmo.edu

***Abstract*—This article will examine the advancement and use of natural language processing methods on text-based Yelp.com review data. In-depth analyses of each phase of the data science methodology will be provided in this article, along with a discussion of the effectiveness of the statistical model that was created using data from several classification tests and how it might be used in a business setting.**

With the advent of online shopping and e-commerce technology, information about products has exploded. Users often rely on reviews and ratings. However, with thousands of reviews, it's often difficult to read them all. Some reviews are repeated and you may miss useful information in between. Therefore, users only read parts or follow review ratings only. In any case a significant portion of information is ignored.This necessitates the need for a framework that can provide a quick accurate summary of information which can help to make informed decision while making a purchase decision.This summary should make use of the full textual information generate and unbiased classification of the product sentiment as a whole.This article will examine the advancement and use of natural language processing methods on text-based Yelp.com review data. In-depth analyses of each phase of the data science methodology is provided in this article, along with a discussion of the effectiveness of various supervised learning models that were applied using data from several classification tests and how it might be used in for sentiment analysis.This article provides overview of novel methods used in text processing by leveraging natural language processing techniques to remove redundancy from the data and focus on important words to derive the product sentiment as a whole.We also cover the exploratory data analysis to understand the type and count of words and their significance in the finding the opinion related to a product.We leverage several machine learning models to achieve this task of sentiment analysis. There are many techniques and complex algorithms used to instruct and train machines to perform sentiment analysis. Each has advantages and disadvantages. But when used with NLTK and other text pre processing methods together, they give excellent results.The goal is to design a robust model for a one category first and once the model is trained on one product category such as electronics the same can be applied to to other product review categories such as automobile, personal care, furniture etc. Additionally, it can be applied to hotel and restaurant reviews. As data on the Internet grows exponentially, this type of model helps extract useful information and save users time.

***Index Terms*—Keywords: Natural Language Processing, Sentiment Analysis, Machine Learning, Text Classification, Bag of Words**

GitHub Source Code Link

I. Introduction

The use of digital applications for various purposes and the rising popularity of e-commerce have increased users' reliance on reviews and ratings when making decisions. But given that there are thousands of evaluations online, it is frequently difficult to look through them all. They may occasionally be redundant and fail to provide reviews that are focused on particular features. Going through every single one of them is typically a waste of time. Users only read a portion of them as a result, or they base their decisions simply on numerical ratings. In any scenario, a substantial amount of the information is being disregarded, which may result in a biased decision.A quantitative and textual summary of these reviews would therefore aid in making the most of the available data.

The task of summarizing product reviews can have a substantial impact on enterprises in a real-world setting. It gives them meaningful consumer feedback in a condensed format that they can use to enhance their products. The primary reason for studying this issue is that the methods discovered while resolving it can be used to other contexts, such as summarizing news stories, etc. Due to the abundance of lengthy articles on the internet, this kind of activity is useful for gathering pertinent information and saving users' time.

For contemporary applications, it has always been difficult to quantify a notion or an opinion. Today, the majority of websites allow visitors to rate products or services using text and a five-star scale. Even while the star rating system is effective, it is not entirely accurate in capturing people's true feelings. A product with a four-star rating may be considered fantastic by some people while only being above ordinary by others. There's a reason why so many websites employ this system, both people and computer programs can easily understand it. It is preferable to read the full-text review in order to obtain a more accurate representation of these reviewers' opinions.

Using sentiment analysis, we may benefit from text-based data. Approximately 80% of the data on the internet is unstructured. The words on a screen are simple for a human to interpret, but not so simple for a computer program. On the other hand, it is difficult for humans to look through millions

of lines of code in the same amount of time as a program can. Even the most sophisticated models for sentiment analysis have some degree of error.

People have the option to leave reviews of restaurants and other establishments using a five-star rating system in addition to an open-ended text review.However these ratings can be subjective in nature and can distort the overall sentiment about the product. The entire focus of this investigation will be on reviews that are written in text.

As explained previously sentiment analysis can significantly help to derive information in this age of unstructured data world.Sentiment analysis is used to identify the affect or emotion (positive, negative, or neutral) of the data. For a business, it is a simple way to determine customers' reactions towards the product or service and to quickly pick up on any change of emotion that may require immediate attention. We propose the most basic approach to this problem which is to use supervised learning. We can have actual humans to determine and label the sentiment of our data and treat it like a text classification problem. However labelling huge amount of data manually becomes a challenging task. To address the problem of labelling reviews as positive or negative we can leverage the natural language processing libraries. These libraries have various lexical resources that can help to perform the tasks of classification, categorization, and numerous other tasks. TextBlob is a straightforward library that provides intricate textual analysis and processing. A sentiment is identified by its semantic orientation and the force of each word in the sentence for lexicon-based techniques. This calls for a pre-defined dictionary that divides words into negative and positive categories. A text message will typically be represented by a bag of words. Following the individual scoring of each word, the ultimate sentiment is determined by performing a pooling procedure, such as averaging all the sentiments. For straightforward situations, TextBlob typically works just well. We can then proceed to the feature extraction phase after labeling the data.

The primary issue with language processing is that machine learning methods cannot be used to directly process raw text. So, to turn text into a matrix (or vector) of features, we need certain feature extraction techniques. Among the most widely used techniques for feature extraction are: Bag-of-Words & TF-IDF. Bag-of-Words is one of the most basic processes for converting tokens into a collection of features is the use of words. Each word is utilized as a feature to train the classifier in the BoW model, which is used to categorize documents. For instance, the presence of words like "fabulous" and "excellent" in a sentiment analysis task based on reviews indicates a positive review, while the presence of words like "annoying" and "poor" indicates a negative review. As we generate a vector of tokens in random order, one significant disadvantage of utilizing this model is that the order in which words occur is lost. Term frequency-inverse document frequency is referred to as TF-IDF. It draws attention to a particular problem that might not come up often in our corpus but is really significant. The TF-IFD score rises in direct proportion to the frequency of a word in the document and falls in direct proportion to the number of documents in the corpus that use the term.

The most logical place to start is using supervised learning as the challenge entails categorizing text reviews based on the overall sentiment. In order to train models to produce the required output, we will employ a training set. Which is how we feel in this instance. The correct sentiment labels and feature vector inputs present in this training dataset enable the model to develop over time. The loss function serves as a gauge for the algorithm's accuracy, and iterations are made until the error is sufficiently reduced. We can experiment with various configurations of machine learning algorithms, such as Logistic Regression, Naive Bayes, Random Forest, Support Vector Classifier, and Gradient Boost Classifier with Cross Different Validation Values , and assess the results.

II. Motivation

In order to distinguish between texts with positive and negative sentiment, sentiment analysis uses machine learning and natural language processing. It is utilized by brands and organizations to gauge brand reputation, identify client wants, and extract user feelings from user feedback. Customers today express their opinions about businesses, goods, and services more openly than ever before. Additionally, they can communicate their ideas through a variety of websites, forums, and social media outlets. Consequently, sentiment analysis becomes a crucial tool for tracking customer feedback. Textual data analysis can be automated, including chats from social media and reviews, among other things. It might aid in your understanding of the features of your product that customers appreciate and those that irritate them. These information can be used to create customized goods and services that meet customer expectations. Through improved customer experiences and service outcomes, sentiment analysis can increase customer retention and loyalty.

A customer might ask for support via chat or email, for instance. The text from the customer's email or chat session is subjected to sentiment analysis using an algorithm created by combination of Natural Language Processing and machine learning model. The program recognizes the emotional condition of the customer. The customer is agitated in this situation. The customer care representative is prepared for the right reaction thanks to business standards that relate to this emotional state. In this instance, a customer service agent is contacted right away and the support request is immediately upgraded to the highest priority.

Finally, the customer relationship is improved and the problem is satisfactorily resolved when the service person is aware of the customer's emotional condition. This results in a more

sympathetic response than a typical one. We used sentiment analysis in our project primarily to validate the rating based on the review text.

III. Main Contributions & Objectives

Below, we'll list every single subtask that involves dataset creation, model research, model development, and evaluation analysis.

A. Downloaded yelp dataset.

B. Data was Uploaded to Google Drive.

C. Configured Google Colab to seamlessly collaboration on the project work.

D. Preprocessing and data cleaning was done to take small chunk of the correct data for further analysis.

E. Worked on the models like, logistic regression, support vector classifier, Gradient Boost, Naive Bayes, and random forest and selected the best model.

F. At last, we identified discrepancy if any between rating and the actual review.

IV. Related Work

Early summarizing studies concentrated on extracting and measuring the emotion toward the primary features of the reviewed entity (Snyder and Barzilay [2007], Titov and McDonald [2008]). These aspect-based summaries are quite informative, but the lack of justifications and explanations makes it difficult for the user to comprehend why a specific aspect obtained a certain rating. Another area of research, called multi-document summarizing, tries to extract textual summaries from input reviews. While (Brainskas et al. [2020]) apply an unsupervised framework for Summarization as copycat-review generation, (Ganesan et al. [2010]) employs a graph-based approach to summarize extremely duplicated viewpoints. These methods can provide more information, but they don't offer a quantitative analysis of the data. Additionally, the prominence of each piece in the summary is not stated, making it impossible to assess their relative importance. A summary should preferably state the percentage of positive vs. negative evaluations for each position to capture contrasting points of view. Recently, a novel extractive summarization framework called Key Point Analysis (KPA) has been presented (Bar-Haim et al. [2020a]) (Bar- Haim et al. [2020b]) to solve the shortcomings of the aforementioned approaches. KPA extracts the key ideas from a group of texts and compares the input sentences to these key ideas (KPS).The collection of important points is chosen from a group of candidates, comprising brief input sentences of strong argumentative quality, in order to maximize coverage while minimizing redundancy. Both textual and numerical versions of the data are shown in the summary that results. This approach, which relies on the successful identification of key point candidates and employs a rule-based methodology to ensure that only succinct single phrases are chosen, is likely to fail if there are insufficient candidate reviews in the dataset.

Many different sentiment lexica have been developed over the years, varying greatly in the number of annotated words they contain (from a few hundred to hundreds of thousands), the values they assign to a word (from binary to multi-class to finer-grained scales), and the method used to gather these ratings (manually or automati- cally). Here, we just provide a few noteworthy and realistic examples.

Sentiment analysis on Yelp.com is a highly common study topic that has been used in a variety of business applications and on a number of joint projects over the years (Rosenthal et al., 2017). The effects of intensifiers and words that switch their polarity in sentiment dictionaries have been the subject of numerous research (Flekova et al., 2015).

Sentiment analysis has been investigated on several levels: Document Level, Sentence Level, Phrase Level, and Aspect Level.

A. Document level sentiment analysis

A single polarity is assigned to the entire document after doing sentiment analysis at the document level. Not many companies utilize sentiment analysis of this kind. A book's chapters or pages can be categorized as good, negative, or neutral using this method. At this level, the material can be classified using both supervised and unsupervised learning techniques (Bhatia et al. 2015). The two most significant problems in document-level sentiment analysis are cross-domain and cross-language sentiment analysis. (2021 Saunders) It has been demonstrated that domain-specific sentiment analysis can reach exceptional accuracy while yet being substantially domain-sensitive. The feature vector in these tasks is a set of words that must be constrained and domain-specific.

B. Sentence level sentiment analysis

Each sentence is examined at this level of analysis in order to determine its polarity. When a document is associated with a diverse range and mixture of emotions, this is really helpful (Yang and Cardie 2014). This classification level has a subjective classification component (Rao et al. 2018). With more training data and processing resources, each sentence's polarity will be decided separately using the same approaches as the document level. Each sentence's polarity can either be utilized individually or combined to determine the document's overall attitude. Sometimes, document-level sentiment analysis is insufficient for some applications (Behdenna et al. 2018).

C. Phrase level sentiment analysis

Opinion words at the phrase level are mined as part of sentiment analysis, and classification is done. There could be one or many components in each phrase. It is noted that a single element is conveyed in a phrase in these product reviews, which may be helpful as they cover several lines (Thet et al. 2010). It has recently been a popular research topic. While sentence-level analysis is more helpful because

a text contains both positive and negative claims, document-level analysis focused on classifying the entire document as subjective, either positively or negatively. The simplest unit of language is the word, and the subjectivity of the sentence or document in which it appears has a direct bearing on the word's polarity. Adjectives significantly increase the likelihood that a sentence will be subjective. (Kim and Fredriksen-Goldsen, 2017) The term chosen for expression also reflects the demographic traits of people, such as their age and gender, as well as their aspirations, social status, and personalities, as well as other psychological and social traits (Flek 2020). Therefore, term provides the framework for text sentiment analysis.

D. Aspect level sentiment analysis

The aspect level of sentiment analysis is used. Sentiment analysis at the aspect level is necessary since each statement could include numerous aspects. Priority is given to each element utilized in the sentence, each element is given a polarity, and then an aggregate sentiment is produced for the entire statement (Schouten and Frasincar 2015; Lu et al. 2011).

E. Unsupervised Sentiment Analysis

The source of all the information was Yelp. The objective was to collect at least 10,000 reviews from a wide range of well-known eateries. the NYC Open Data directory of eateries was used to make the selection. Eateries were transformed to valid URLs using Yelp's Fusion API. The training dataset was built using 11067 reviews from 220 different eateries in total. We had to remove any malicious HTML tags from the initial web scrape in order to retrieve the data in a format that could be used. The restaurant name, location, and other identifying information were then added to a pandas data frame along with the reviews as strings. Then, a CSV file of this data frame was exported for additional analysis.

Yelp served as the primary informational source. The goal was to gather at least 10,000 reviews from a variety of renowned restaurants. The restaurants were chosen from the NYC Open Data directory. The Yelp Fusion API was used to convert restaurants' URLs into legitimate ones. A total of 11067 reviews from 220 distinct restaurants were used to build the training dataset. To get the data from the first site scrape in a usable manner, we had to remove any dangerous HTML tags. The reviews were then uploaded as strings to a pandas data frame together with the restaurant's name, address, and other identifying information. This data frame was then exported as a CSV file for further examination.

The model testing procedure for this analysis will look like this:
Determine a pipeline, choose a model type, and then use Grid Search to determine the ideal parameters.
- Save the accuracy, model, and f1 scores.
- Evaluate the outcomes using confusion matrices.
Author tested ten different models, including Logistic Regression, Random Forest, Support Vector Classification, Gradient Boosted Classifier, two text-based Naive Bayes, two text-based Gradient Boosted Classifiers, one Naive Bayes with dense and sparse text matrices, and one stacked model, in order to obtain a good variety of results. The Naive Bayes model's probability and the Gradient Boosted Classifier's feature performance were combined to create the stacked model, which had a 92% accuracy rate.

V. Proposed Framework

The objective of this project is to analyse and gather review data from different businesses on Yelp using a combination of natural language processing and data processing techniques. Once we have this information, we may analyze it and, using the content of the evaluations, ascertain the sentiment of any location. After performing few statistical analysis methods, the next task is to identify features that can be used to feed the supervised machine learning models. The final steps involve applying various evaluation metrics to select the best performing model.

A. Data gathering

This dataset contains data that Yelp collected about companies, reviews, customers, check-ins, suggestions, and images. The file is in json format and is 8.69 gigabytes in size (6 json files including business.json, review.json, user.json, checkin.json, tip.json and photo.json). If we want to know what kind of business each review is for, we would need to integrate the review file with the business file because the review.json does not provide business information. We keep the dataset of only the companies that are still operating. By first using categories to search for the pertinent companies, we may later find the relevant reviews we need. Restaurants, food, shopping, home services, and spas are just a few of the example categories. However, in this illustration, we are only concerned with companies that deal with electronics products. For the category that we are working with, there are about 36,000 records.

B. Preprocessing

We need to reduce the data to a more manageable size now that we have it.Stop words like "the," "a," "for," or "so" aren't very meaningful when it comes to sentiment analysis. Eliminating them will increase overall efficiency, reduce filler, and make the general meaning of each review easier to understand. Now that we have this, we apply it to our reviews dataset. To make the analysis process easier, we add two more columns: one for lowercase reviews and the other for lowercase text with no punctuation.

C. Exploratory Data Analysis

To beign with , we first plotted the word count in each review to get the idea of frequency distribution as shorn in "Fig. 1" & "Fig. 2". We found out that there are close to 2000 reviews out of 36000 having word count more than 400 which can be ignored to simplify the model training process.

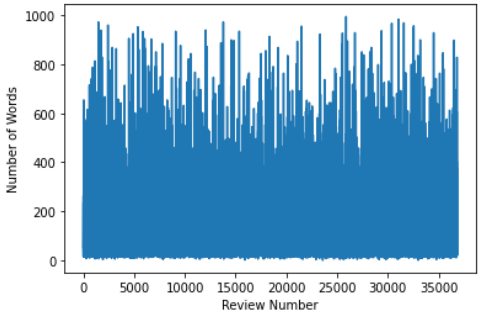

Fig. 1. Word counts of each Review

store	18115
service	14269
phone	12980
get	11910
one	11158
would	10515
time	9920
customer	9271
back	9014
go	8349
new	8155
like	8014
great	7519
even	6703
dont	6558
walmart	6409
told	6324
said	6072
screen	5980
went	5802
dtype: int64	

Fig. 2. Frequent Words

We now count the stop words in the original review's words. Stop words are an useful initial step for eliminating meaningless terms, but we check through the entire text to identify any words that are repeatedly used. Now that we don't have stop words as shown in "Fig. 3", we can create a word cloud without any filler terms. Instead of seeing only stop words, this will help the more frequently used words with significance stand out more. Stemming is a crucial component of natural language processing; it effectively strips each word variant of all its suffixes to reveal the original root word. For instance: After the "ing" is removed, going changes to going, and so on. As a result, the reviews have less uncommon terms, which will be advantageous when we calculate the frequency of sentimental words. There are other stemming techniques, but the Snowball Stemmer, an improved variant of the well-known Porter Stemmer, is the technique we think makes the most sense for this analysis. Now that we have our data, we need to reduce it into a more useful format. Lemmatization ensures that each word is properly formed but, unlike stemming, it does not render the text unusable. For this reason, we created a column for both.

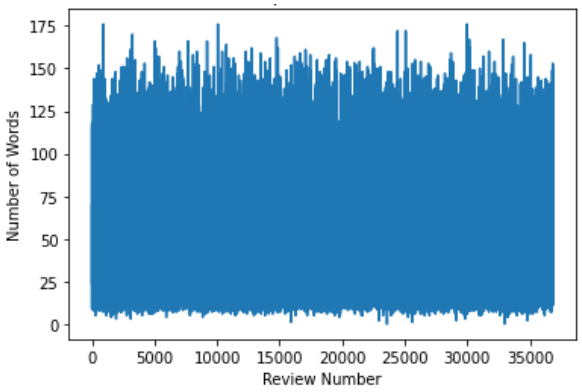

Fig. 3. Word counts After Stop word removed from each Review

Polarity(-1,1) and Subjectivity(0,1) can be extracted from text using TextBlob's extremely helpful library. As it will help us categorize the reviews we are currently working with and train the model, polarity is essentially the only useful factor for this project, though subjectiveness would be worth investigating as well. We now have various metrics to use when engaging with these reviews and a much better grasp of our data. We may utilize polarity and information about the total word count, the percentage of stop words and positive words, and the percentage of stop words to categorize the reviews into sentiments. In the future, it will be wise to decide whether we want to create more categories, such as a five-star system, or just keep with positive and negative. If we choose the first option, we must establish, among other things, where the cutoff for a 4 star vs. a 5 star might be. But for now, we proceed with the modeling.

D. Machine Learning

We looked into the connections in our data and discovered Polarity, which aids in labeling our training data. We prepare our data in this notebook so that we can effectively use it as a train/test split for creating a model in the subsequent notebook.

We perform following actions:

- Label and classify the data.
- Clean up the reviews and create vector data.
- Making use of a text vectorization pipeline.

We must first organize our data into groups that our model can sort by before we can begin to use it. Because we're performing a sentiment analysis in this situation, we'll categorize our data into "positive" and "negative" sentiment reviews based on the TextBlob Polarity of each review. We can see that the majority of our reviews are positive. We further organize it for better outcomes.

Now that we have organized our data, we must turn our tokenized reviews into word vectors using one of two methods: TFIDF and CountVectorizer.

- **CountVectorizer :**
  The CountVectorizer converts text to word counts and, more significantly, a vector using the bag of words method. Then, by comparing the vector for each word, sentence, or paragraph to other vectors, we may determine which ones are most similar.
- **TFIDF:**
  Named after the term frequency. Inverse Document Frequency allows us to gauge a word's usage across all lines of text in addition to gauging its frequency inside each review. This is highly significant when choosing extremely original words.

Now that we have these new features we can generate a dataset so that we can train our model . We gathered the information, examined it, and identified the key characteristics. We will test a variety of models in this project. Although there are numerous models we can test, we will initially focus on the common machine learning models like logistic regression, support vector classifier, Gradient Boost, Naïve Bayes, and random forest.

Along the process, we will gather accuracy and f1 scores to discover which model works best with this data.

We have followed the following procedure:

- Select a model type.
- Create a pipeline, then use GridSearch to determine the optimum parameters
- Run the model's accuracy and f1 scores after loading it.

VI. Data Description

We have downloaded yelp dataset.

A. Business data set

We are planning to use following columns:

- **business**: string, 22-character unique string business id.
- **name**: string, the business's name
- **address**: string, the full address of the business
- **city**: string, the city
- **state**: string, 2-character
- **state code**: string, 2-character state code
- **postal code**: string, the postal code

B. Review dataset

We are planning to use following columns in Review

- **review_id**: string, 22-character unique review id
- **business_id**: string, 22-character business id, maps to business in business.json
- **user_id**: string, 22-character unique user id
- **stars**: integer, star rating
- **date**: string, date formatted YYYY-MM-DD
- **state code**: string, 2-character state code
- **text**: "Great place to hang out after work: the prices are decent, and the ambience is fun. It's a bit loud, but very lively. The staff is friendly, and the food is good. They have a good selection of drinks."

We have done data filtering and the data cleanings. We have used about 7,000,000 total data in our project. This total data is consists of business categories of 'Doctors, Traditional Chinese Medicine, Naturopathic/Holistic, Acupuncture, Health & Medical, Nutritionists, Shipping Centers, Local Services, Notaries, Mailbox Centers, Printing Services, Department Stores, Shopping, Fashion, Home & Garden, Electronics, Furniture Stores, Shopping, Jewelry, Piercing, Toy Stores, Beauty & Spas, Accessories, Fashion, Fitness/Exercise Equipment, Eyewear & Opticians, Shopping, Sporting Goods, Bikes, Beauty & Spas, Permanent Makeup, Piercing, Tattoo'.

But we have filtered out the "Electronics" to get the smaller and precise data. This data set is now 36,858 records. We have also created the data new data set for the further sentiment analysis.

VII. Results & Analysis

After extracting the features using TfidfVectorizer we will train our models using different classification algorithms. "Fig 4" shows the distribution of stemmed words which are used for extracting the final features.

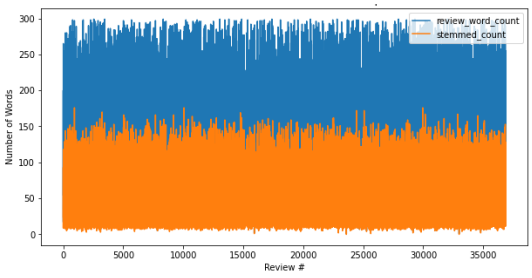

Fig. 4. Stemmed with no Stop Words

After cleaning the data and extracting features the data set was divided into train test split. Machine learning algorithms that are applicable for prediction-based algorithms and applications are evaluated using the train-test split. We can compare the output of our own machine learning model to that of other machines using this quick and simple process. In ideal approach, the training set contains 70% of the real data, whereas the test set contains 30% of the real data.

To assess how well our machine learning model works, we divided a dataset into train and test sets using 80:20 split. The statistics of the train set are known and are utilized to fit the model. The test data set, which is the second set, was utilized just for predictions.

We restricted total data set to 10000 to speed up the testing and evaluation process as processing original dataset of 36000 was causing google colab session crash even with

the GPU. We used Sklearn library for splitting the data and for modelling purpose.

There are two distinct ways to use Scikit-Learn to implement the TF-IDF in Sklearn. The first method involves using the TfidfVectorizer class, while the second method involves using the TfidfTransformer class.

There is actually no difference between the two implementations theoretically. Practically speaking, if we want to use TfidfTransformer, we have to write more code. TfidfVectorizer performs both term frequency and inverse document frequency, whereas TfidfTransformer requires you to use the CountVectorizer class from Scikit-Learn to perform term frequency. This is the main distinction between the two implementations. We created a TF-IDF feature matrix from lemmatized text reviews. Our machine learning model was trained using the features that were taken using TfidfVectorizer to predict any of Slightly Positive, Slightly Negative, Positive, Negative classes. We used 80% of 10000 records for training and remaining 20% as a test data set. We also tried different combinations of cross validation sets to verify the robustness of the model for different test data sets. We performed the testing on 9 combinations of machine learning algorithms like Logistic regression, Naïve Bayes, Random Forest, Support Vector Classifier and Gradient Boost Classifier with Cross validation values of 3,5 and 10. We were able to improve the accuracy from 60% of Logistic Regression to 70% using gradient boost classifier.
"Fig. 5" shows the distribution of classes. It can be seen that the data set is imbalanced towards positive side.

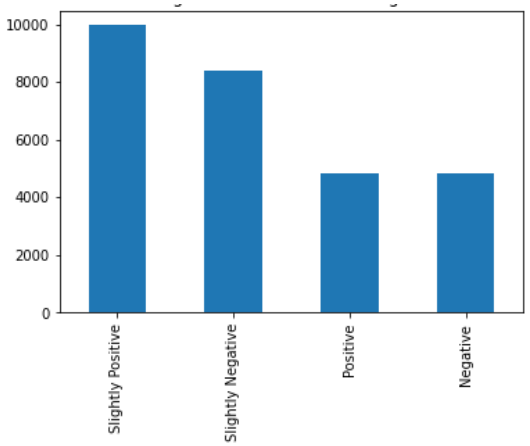

Fig. 5. Assigned Sentiment of Training Data

"Fig 6" shows a confusion metrix which can help to determine the performance of the model.The confusion matrix corresponds to a baseline model that we trained using logistic regression which gave us a head start with the accuracy score of 60%.
We tried different combinations of cross validation sets to verify the robustness of the model for different test data sets.

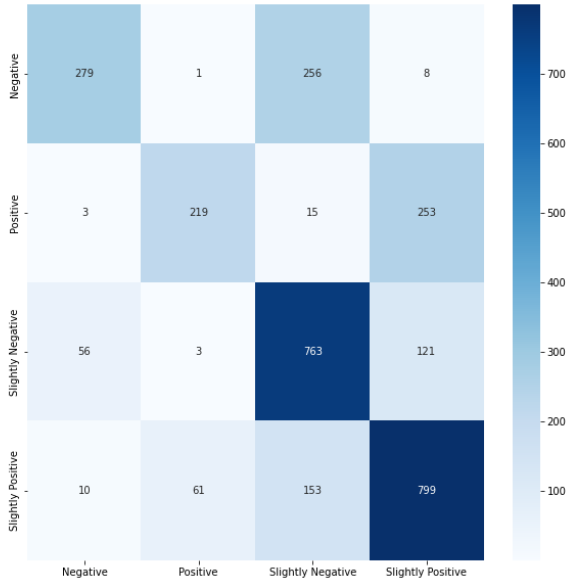

Fig. 6. Confution Matrix

Model	Accuracy	F1 Macro	F1 Micro	F1 Weighted
Logistic regression (Cross validation- 5)	60%	59%	60%	60%
Logistic regression (Cross validation- 10)	60%	59%	60%	60%
Naive Bayes with CountVectorizer	61%	60%	61%	61%
Naive Bayes with TFIDF	61%	56%	61%	59%
Random Forest (Cross validation- 5)	63%	60%	63%	62%
Random Forest (Cross validation- 10)	62%	59%	62%	61%
Support Vector Classifier (Cross validation- 3)	62%	61%	62%	62%
Gradient Boost (Cross validation- 5)	70%	68%	70%	70%
Gradient Boost without grid search	70%	68%	68%	69%

Fig. 7. Results

We performed the testing on 9 combinations using machine learning algorithms like Logistic regression, Naive Bayes, Random Forest, Support Vector Classifier and Gradient Boost Classifier with Cross validation values of 3,5 and 10.We were able to improve the accuracy of base line Logistic Regression model from 60% to 70% using gradient boost classifier as shown in "Fig. 7". Accuracy can further be improved using better labelling of classes and increasing training dataset size. Also word embeddings methods and unsupervised learning methods can also be tried as a future scope for this project.

REFERENCES

[1] I. Smeureanu and M. Zurini, "Spam Filtering For Optimization in Internet Promotions Using Bayesian Analysis," J. Appl. Quant. Methods, vol. 5, no. 2, 2010.
[2] H. Wang, D. Can, A. Kazemzadeh, F. Bar, and S. Narayanan, "A System for Real-time Twitter Sentiment Analysis of 2012 U.S. Presidential Election Cycle," Proc. 50th Annu. Meet. Assoc. Comput. Linguist., no. July, pp. 115–120, 2012.
[3] P. Goncalves, B. Fabrício, A. Matheus, and C. Meeyoung, "Comparing and Combining Sentiment Analysis Methods Categories and Subject Descriptors," Proc. first ACM Conf. Online Soc. networks, pp. 27–38, 2013.

[4] B. Pang, L. Lee, and S. Vaithyanathan, “Thumbs up?: sentiment classification using machine learning techniques,” Proc. Conf. Empir. Methods Nat. Lang. Process., pp. 79–86, 2002.

[5] S. J. M. Modha, Jalaj S. , Gayatri S. Pandi, “Automatic Sentiment Analysis for Unstructured Data,” Int. J. Adv. Res. Comput. Sci. Softw. Eng., vol. 3, no. 12, pp. 91–97, 2013.

[6] K. N. Devi and V. M. Bhaskarn, “Online forums hotspot prediction based on sentiment analysis,” J. Comput. Sci., vol. 8, no. 8, pp. 1219–1224, 2012.

[7] K. Mouthami, K. N. Devi, and V. M. Bhaskaran, “Sentiment analysis and classification based on textual reviews,” 2013 Int. Conf. Inf. Commun. Embed. Syst., pp. 271–276, 2013.

[8] G. Eason, B. Noble, and I. N. Sneddon, “On certain integrals of Lipschitz-Hankel type involving products of Bessel functions,” Phil. Trans. Roy. Soc. London, vol. A247, pp. 529–551, April 1955.

[9] J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73.

[10] I. S. Jacobs and C. P. Bean, “Fine particles, thin films and exchange anisotropy,” in Magnetism, vol. III, G. T. Rado and H. Suhl, Eds. New York: Academic, 1963, pp. 271–350.

[11] K. Elissa, “Title of paper if known,” unpublished.

[12] R. Nicole, “Title of paper with only first word capitalized,” J. Name Stand. Abbrev., in press.

[13] Y. Yorozu, M. Hirano, K. Oka, and Y. Tagawa, “Electron spectroscopy studies on magneto-optical media and plastic substrate interface,” IEEE Transl. J. Magn. Japan, vol. 2, pp. 740–741, August 1987 [Digests 9th Annual Conf. Magnetics Japan, p. 301, 1982].

[14] M. Young, The Technical Writer’s Handbook. Mill Valley, CA: University Science, 1989.

[15] P. S. Earle, D. C. Bowden, and M. Guy, “Twitter earthquake detection: Earthquake monitoring in a social world,” Ann. Geophys., vol. 54, no. 6, pp. 708–715, 2011.

[16] M. Cheong and V. C. S. Lee, “A microblogging-based approach to terrorism informatics: Exploration and chronicling civilian sentiment and response to terrorism events via Twitter,” Inf. Syst. Front., vol. 13, no. 1, pp. 45–59, 2011.

[17] A. Go, R. Bhayani, and L. Huang, “Twitter Sentiment Classification using Distant Supervision,” Processing, vol. 150, no. 12, pp. 1–6, 2009.

[18] B. Pang and L. Lee, “Opinion mining and sentiment analysis,” Found. Trends Inf. Retr., vol. 2, no. 1–2, pp. 1–135, 2008.

[19] L. Breiman, “Random forests,” Mach. Learn., vol. 45, no. 1, pp. 5–32, 2001.

[20] P. K. Singh and M. Shahid Husain, “Methodological Study Of Opinion Mining And Sentiment Analysis Techniques,” Int. J. Soft Comput., vol. 5, no. 1, pp. 11–21, 2014.

[21] J. Khairnar and M. Kinikar, “Machine Learning Algorithms for Opinion Mining and Sentiment Classification,” Int. J. Sci. Res. Publ., vol. 3, no. 6, pp. 1–6, 2013.

[22] T. M. S. Akshi Kumar, A. Kumar, and T. M. Sebastian, “Sentiment Analysis on Twitter,” IJCSI Int. J. Comput. Sci. Issues, vol. 9, no. 4, pp. 372–378, 2012.