

SENTIMENT ANALYSIS: TEXT MINING USING LAMBDA ARCHITECTURE

Aravindan Srinivasan

2981707

Submitted in partial fulfillment for the degree of
Master of Big Data Management and Analytics

Griffith College Dublin

September, 2018

Under the supervision of Supervisor's Name

Barry Denby

Disclaimer

I hereby certify that this material, which I now submit for assessment on the programme of study leading to the Degree of Master of Science in Big Data Management and Analytics at Griffith College Dublin, is entirely my own work and has not been submitted for assessment for an academic purpose at this or any other academic institution other than in partial fulfilment of the requirements of that stated above.

Signed: _____**Date:** _____

Acknowledgements:

The satisfaction and euphoria that accompany the completion of any task would be incomplete without the mention of the people who made it possible, whose constant guidance and encouragement ground our efforts with success. We consider it a privilege to express our gratitude & respect to all those who guided us in the completion of this project.

We owe our gratitude and sincere thanks to **Mr. Aqeel Kazmi**; for his strong support, and concern during the course of our project work.

We offer our deepest gratitude to **Mr. Barry Denby**, Programme Director, Griffith College Dublin; for his unending guidance extended for the project work.

We would like to thank **Dr. Waseem Akhtar**, Head of the Computing Science faculty, Griffith College Dublin; for his encouragement and support during our project work.

Finally, we thank our family members for their support and also Almighty for his grace and blessings showered during the project.

Contents

Acknowledgements:	iii
ABSTRACT:	5
Chapter 1. Introduction	6
1.1 Sentiment Analysis:	6
1.2 Challenges faced in Sentiment Analysis:	6
1.3 Objective:	6
1.4 Overview Approach:	7
Chapter 2. Background	8
2.1 Literature Review	8
2.2 Related Work:	8
Chapter 3. Methodology	10
3.1 Support Vector Machine (SVM):	10
3.2 Naïve Bayes:	12
3.3 Gaussian Naïve Bayes:	14
3.4 KNN algorithm:	15
Chapter 4. System Design and Specifications	17
Chapter 5. Implementation	18
5.1 Scrappy:	18
5.2 Clean Text:	19
5.3 Finding Polarity and Subjectivity:	22
5.4 Improving Text Handling:	22
5.5 Lambda Architecture:	24
5.6 Apache Cassandra:	29
5.7 Visualization:	33
Chapter 6. Testing and Evaluation	34
Chapter 7. Conclusions and Future Work	38
References:	39

ABSTRACT:

Sentiment analysis on amazon product reviews using machine learning techniques and extracting the strongest positive, negative and neutral reviews using lambda architecture. Sentiment analysis fetches customer's feelings, emotions and attitudes towards the product. The need for this type of analysis has increased to a greater extent. Product manufacturers realized that sentiment analysis is the key to achieve success. The moment the company receives valuable information after sentiment analysis, they would dig deep about the issue. Companies can improve their product weakness and respond to the voice of the customer. Big data systems collect information from a variety of sources.

Therefore, the need for Architecture that should deal with the huge amount of data at high velocity is a major requirement. Extracted data should be gathered in an effective database. Using Cassandra makes our data available to do visualization technique. Hence, lambda architecture helps in data-processing to manage massive quantities of data by utilizing both stream and batch processing.

This big data system provides some solutions for solving sentiment analysis over massive amounts of data. The final data is gathered and forwarded to visualization in word cloud and text blob by displaying the strongest positive, negative and neutral words.

Chapter 1. Introduction

1.1 Sentiment Analysis:

Sentiment analysis is a machine learning in which machines analyzes the human's emotions, sentiment over the products based on stars, comment, thumbs up and down given by the user. The data we are using is collected from the amazon website. Amazon website allows any customer to give the feedback about the product. Once they post the comment, amazon verifies the comment and allows us to post the comment for that product. Sentiment is a feeling of a customer with emotions. It is also called as opinion mining because, it allows us to study about the opinion about the consumer from amazon. Day by day data is increasing enormously which allows us to do fetch the accurate result from the system. Machines should be efficient enough to understand human feelings. Sentiment is prompted by feelings with attitude, judgement etc.

1.2 Challenges faced in Sentiment Analysis:

The reviews given from the customer can either be positive, negative or neutral few reviews have the combination of both the positive and the negative comments. The extracted information should be useful for the decision making to improve the quality of the product and detect the weakness of the product from the customer perspective [8]. Reviews should be valid, there is a chance of gaining fake comments.

1.3 Objective:

The model created should face the challenges by processing the verified customer reviews in order to gain the actual problem with the product. In order to gain faster access with the data, using lambda architecture will help us on handling the massive amount of data. It improves the performance of the system by passing the strongest polarity scores in order to gain access with pushing positive, negative or neutral comments in the system. People around the world can post their own perspective or opinion of a product in various forms such as blogs, social media, comments [22]. Lambda architecture is a fastest data processing architecture that will handle massive amount of data. It takes advantage by performing both batch and stream

processing methods. This approach has high throughput, fault tolerant and latency which makes our process effective and quicker.

The processed data from the lambda architecture are pushed into cassandra, because Cassandra query language are user friendly, In big data applications, It is scalable, fault tolerant architecture [10]. It is very efficient to write and read values like scale to millions of update transactions per second. Even it has the ability to handle a single node or an entire failure of data centers at ease.

1.4 Overview Approach:

Initially we start off with scrapping the data from the amazon website. the collected data is pushed into the cleaning text phase in order to gain accurate results. Then the data is feed into our lambda architecture, and the respective data are pushed into the respective tables in Cassandra. Now the data are trained by the knn classifier and Naive Bayes algorithm in order to make use of effective data. Now WordCloud and textBlob are used to do some visualization in the model with the final set of cleaned data to extract the strongest positive, negative and neutral comments in the system.

Chapter 2. Background

2.1 Literature Review

Sentiment Analysis on online product review was done by RaheesaSafrin, K.R. Sharmila, T.S.ShriSubangi, E.A.Vimal. They used the data by collecting from consumers in the base of form. A form is created on the website with set of attributes such as name, contact number, mail id, product, and feedback. Using negative phrase identification algorithm, are used to find adjectives and adverbs words which are basically negative prefixes. This performance is evaluated by evaluation measures. Processing comments with positive or negative. Finally, producing the summary of the product used for decision-making for customers [13]. Providing the final review might fail in accuracy in defining the exact problem Because it lacks in fetching positive reviews got from the customer.

Feature Specific Sentiment Analysis for Product Reviews, In this paper Author Subhabrata Mukherjee and Pushpak Bhattacharyy finding specific expression in opinions of the reviews given by the customer. Goal of this paper is to extracting opinions by exploiting the emotions from the reviews [24]. Dependency parsing and threshold parameter allows the model to learn to set of relations to identify closely on opinion expressions [19]. Finding the emotions will lack in concluding on improvement of the products to the manufacturers.

Weakness Finder: Find product weakness from Chinese reviews by using aspects-based sentiment analysis journal paper work was done by Wenhao Zhang, Hua Xu, Wei Wan. This paper finds only `the weakness of the product to help the manufacturers. Assuming might help the manufacturers improve the quality of product. This weakness finder produces a report on weakness of product in the Chinese language [21]. Grouping features by morpheme method and similarity measure. Weakness finder working based on the Chinese language and lacks in stable with the positive feedback.

2.2 Related Work:

From the above described journal papers, the given comment is processed and resulting to either positive and negative from the opinion of the product. The given sentences (i.e) comments is segregated into words. Negative expression is identified b the algorithm namely Negation phrase identification algorithm. As there are enormous amount of data, it is used for

defining the quality of the product. Finally extracting the information and giving overall view of a particular product of the customer. Creating a website and retrieving the feedback from the users.

Data collection: The data is collected in the form of feedback. There are two ways of data collection like star-rating and textual format. It divides the adjectives as good, bad and excellent. The count 1 is considered as bad while 2 and 3 is considered as good and finally remaining stars is considered as excellent. the other way is getting like a text format.

Data preprocessing: Now the customer reviews are getting prepossessed in this system to improve the classification results. They use Part of speech tagging for fetching positive phrases and for negative reviews they use negative phrase identification algorithm. using k-means cluster to classify into 2 clusters, 1 for positive and 0 for negative. As a result, positive and negative results can be fetched [12].

Chapter 3. Methodology

3.1 Support Vector Machine (SVM):

The framework of Support vector machines is recently developed for the statistical learning theory and is successfully deployed to several applications, varying from time series prediction to the latest face recognition technology, to the biological data process of medical diagnosis. The foundations of both theory and practical research have resulted in success to encourage.

Introduction:

Furthermore, the development of their characteristics and for the future. The theory and implementation of SVM are discussed in the following. It is presented by four main sections, with the theoretical foundations of SVM, the mathematical formulation is presented along with the theory of implementation of SVM. The third section involves the experimental work and the variations of the standard SVM proposed. Finally, the last section consists of conclusions and Suggestions for future research.

Implementation:

SVM has been developed under the framework of Statistical Learning Theory, the problem of supervised learning is formulated as follows; We are given a set of l Training data $\{(x_1, y_1), \dots, (x_l, y_l)\}$ in $R^n \times R$ is represented with unknown probability distribution $P(x, y)$ and a loss function. $V(y, f(x))$ that gives the error done for a given x , $f(x)$ is “predicted” instead of the actual value y . The objective is to minimize the expectation of the error on the new data, which is to find a function f that minimizes the expected error [23].

$$\int V(y, f(x)) P(x, y) dx dy$$

Since $P(x, y)$ is unknown, so we need to a principle to induct to interfere from the training available which gives an example function the minimizes the Expected error. The Empirical Risk Management (ERM) over a set of possible Functions, called hypothesis space. In formula, $i/l = \sum_{i=1}^l V(y, f(x))$. Here, with f being restricted to be in a function, hypothesis

space – say H . A significant inquiry is the manner by which close the empirical error of the solution (minimize of the empirical error) is to the base of the normal blunder that can be accomplished with capacities from H . A focal after effect of the hypothesis expresses the conditions under which the two blunders are near one another, and gives probabilistic limits on the separation among experimental and anticipated mistakes (see hypothesis 1 beneath) [23]. These limits are given regarding a proportion of multifaceted nature of the speculation space H : is, the bigger the separation between the observational and Expected blunders are in probability.

Theorem 1 :“ If V is the VC-dimension of a hypothesis space H , then with probability $1-\eta$, the minimum of the expected error that can be achieved with functions from H , say L , and the minimum empirical error, say L_{emp} , satisfy the constraint: η independent of the distribution of the data $P(x,y)$ ”.

Problem (1) can be changed as pursues: Factors ξ_i are called slack factors, and they measure the mistake made at the point (x_i, y_i) . We see that the number of imperatives is equivalent to the quantity of preparing information, Preparing SVM, that is, taking care of compelled QP.

Problem (3), turns out to be very testing when the quantity of preparing focuses is Enormous. Various strategies for quick SVM preparing have been proposed in the writing. Decaying the QP issue into various littler ones through lumping or on the other hand deterioration calculations is one methodology proposed (for Instance observe.

A consecutive improvement technique has likewise as of late been proposed (Platt, 1998). In the methodology recommended taking care of the QP issue (3) was that of Interior Point Methods (IPM). presents a review of IPM, focusing on base double improvement techniques. These strategies consist of taking care of iterative the QP issue by moving between the plan (3) and its double detailing, which can be observed to be normally in the writing SVM again prepared by tackling the double streamlining issue. Proposes basic double IPM techniques for SVM preparing which vary from the ones regularly utilized. In the IPM talked about are likewise used to prepare learning machines other than SVM. Specifically, demonstrates how the proposed base double IPM can be utilized to prepare Artificial Neural Networks (ANN) commonly prepared to utilize back propagation One of the primary contrasts among ANN and SVM is that, as referenced in, while for ANN there can be numerous nearby ideal

arrangements, for SVM there is just a single ideal answer for an issue (3) since SVM are prepared by tackling a QP issue which has one worldwide ideal arrangement.

This is one pragmatic of SVM when contrasted and ANN.

(i) SVM is motivated through the statistical learning hypothesis. The hypothesis portrays the performance of learning machines utilizing limits on their capacity to foresee future information. One of the papers in the workshop displayed new limits on the exhibition of learning machines, and proposed a strategy to utilize them tentatively to all the more likely comprehend the learning machines (including SVM).

(ii) SVM is solved by given training to the constrained quadratic optimization problem. Among others, this implies that there is a unique optimal solution for each choice of the SVM parameters. Unlike other learning machines, such as standard Neural Networks trained using back propagation.

(iii) Primal double interior-point improvement techniques might be utilized to proficiently prepare SVM with large data sets, as depicted.

(iv) Training many local SVMs instead of a single global one can lead to significant improvement in the performance of a learning machine, as shown.

(v) SVM has been successfully used for medical diagnosis. Methods for dealing with unbalanced training data, or for biasing the performance of an SVM towards one of the classes during classification were suggested and used.

(vi) The ideas presented in the papers and discussed in the workshop suggest several future research directions. SVM's are, among others, the choice of the kernel of the SVM and the choice of the regularization parameter. On the other hand, significant improvements in the performance of SVM may be achieved if ensembles of SVM's are used.

3.2 Naïve Bayes:

Naïve Bayes is a learning algorithm that uses Bayes rule. It provides us classification accuracy. Many features merge with efficiency makes naïve Bayes widely applied in practice.

Introduction:

It helps to compare the results with sample data to estimate the probability $p(y|x)$.

(i)Computational efficiency:

Training time is linear to both number of attributes and training examples, while classification time is linear with number of attributes but it is not affected by the training examples.

(ii)Incremental learning:

Low order probability derived from training data is updated to new training data and it is stored for better results.

(iii)Robustness(noise):

Naïve Bayes uses all the attributes for predictions; therefore, it is not affected by the noise produced.

(iv)Robustness (missing value):

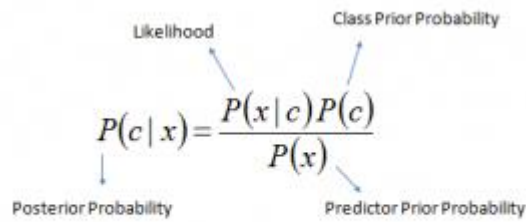
Since naïve Bayes uses all its attributes for predictions, if one attribute value is missing, it uses the value of the other and thus making it less performance loss [10]. So thus, it is insensitive to missing values.

$$P(y|x)=p(y)p(x|y)/p(x)$$

Implementation:

Naïve Bayes is based on Bayes rule. To explain in simple terms, Naïve Bayes predicting presence of one feature in a class is not related to the presence of other feature in a class. For example, fruit apple has some properties like red, round and 4 inches in diameter, these terms are dependent to one another and moreover these independent terms contributing to form a fruit apple. Bayes model is useful to assemble easily and especially valuable for exceptionally huge informational indexes [10]. Its performance is effective over sophisticated classification models.

we will be tokenizing the fetched sentences in order to find the highest scoring class in the model. our result classifies the polarity and gives update to our resulted scores. Emotions are the combinations of mind and body thoughts of behavioural, cognitive and reacting to situations. So, review posted will have those sorts of expressions either positive or negative. This emotion might be very important for the manufacturers to keep track of their product, so that they can update their product and give the best service to the consumers. Even there can be some failure in posting reviews, so fetching the strongest negative and positive reviews will be very useful for the situations.



$$P(c|x) = \frac{P(x|c)P(c)}{P(x)}$$

$$P(c|X) = P(x_1|c) \times P(x_2|c) \times \dots \times P(x_n|c) \times P(c)$$

$P(c|x)$ has the probability of class from given predictor. $P(c)$ is probability of class. $P(x)$ is probability of predictor. We have the data set of positive or negative comments. We need to find whether the comments are positive or negative. There are certain steps to achieve to find the accurate results in the system.

Initially we have to form frequency tables from the massive data sets.

One of the advantages of using naïve Bayes algorithm is fastest to have a prediction on test data sets, moreover performing well with multi class prediction.

It has a better performance on comparing with other models and needs few training data.

Comparing with numerical values and categorial variables leads to high performance by Naïve Bayes.

Bayes theorem:

It states that probability of event B given A is same to the probability of event A given B multiplied by probability of A over probability of B. It defined as

$$P(a|b) = \frac{P(b|a) \cdot P(a)}{P(b)}$$

$P(a|b)$: Probability of occurrence of event A given event B is true.

$P(a)$ and $p(b)$: probability of event a and event b respectively.

$P(b|a)$: Probability of occurrence of event B given event A is true [31].

3.3 Gaussian Naïve Bayes:

We can extend naïve Bayes to real value attributes by taking a Gaussian distribution. The extension from naïve Bayes to Gaussian distribution is known as Gaussian naïve Bayes. Gaussian is better than other functions because we going to estimate only mean and standard deviation from the training data [9].

Introduction:

One of the methods to obtain important information is to fetch sentiment from the given review. In sentiment classification, machine learning algorithms are commonly used because due to its ability to learn more in the training dataset with high accuracy. Data in the social media, web are massively increasing day by day enormously. This will be very useful for manufacturers to decide the best products in the future.

Implementation:

We going to import GaussianNB from sklearn.naive_bayes, once the module is imported we can process to the further improvement in the project. After importing the modules, we going to feed the training and testing data into our module, so that we can fetch the important information from the sentences. Then we will proceed to find the accuracy of the naïve Bayes in order to find the best algorithm in the system.

3.4 KNN algorithm:

Introduction:

Text categorization is one of the major tasks in today's world for solving problems in real world. Text categorization is a process of classifying the large amount of text data effectively and efficiently. The major problem in categorizing the text data is to find the inter-relationship between the categories in the text data [26]. Traditional text categorization model is imbalanced which leads to decrease in performance of categorizing the text data in the large dataset [25].

Implementation:

The KNN algorithm is most efficient and effective algorithm for categorizing the text data in the large dataset. Even KNN algorithm is simple and effective method for text categorization it as some major fault, which will decrease the overall performance of the model. The major issues in KNN algorithm are, complexity of finding the similarity of training sample is high, single training sample will affect the performance of the algorithm and KNN algorithm does not build any classification model for categorizing the text data [26]. Hence KNN algorithm

is not well suited for any of the application, where the data are streaming dynamically in very large amount. KNN algorithm is further enhanced by adjust weight in the number and the distribution of training data. Thus, It makes the KNN algorithm to achieve better performance than the previous model. Methods to increase the performance of the algorithm are, decreasing the dimensions of vector text, decreasing the number of training samples and accelerating the process of finding the k nearest neighbors [26]. These methods make the algorithm more effective and efficient for categorizing the huge amount of text data.

Chapter 4. System Design and Specifications

In this section, discussion on various modules been used in the system. We used a set of libraries and packages in order to find the strongest positive, negative and neutral comment in the system. Initially using scrapy helps in data collection in the system. The extracted data are then pushed into our system, in order to gain advantage over the data to benefit the manufacturers. Machine learning algorithms used to determine the performance of our extracted data using the jupyter notebook, which performs running multiple functions using python libraries. We used the python version of 3.7.1 for analysing the data and load list of libraries in the model. Jupyter notebook is been used by many researchers to perform list of predictions on data. It has the advantage of downloading various formats of files using auxiliary tools such as html, py etc.

For this circumstance, "notebook" or "scratch pad records" mean files that contain both code and rich substance segments, for instance, figures, joins, conditions, ... Because of the mix of code and substance parts, these records are the ideal spot to join an assessment depiction, and its results, similarly as, they can be executed play out the data examination in authentic time. For now, you ought to understand that "Jupyter" is a free shortened form meaning Julia, Python, and R. These programming vernaculars were the chief objective lingos of the Jupyter application, anyway nowadays, the diary development also supports various tongues.

As a server-client application, the Jupyter Notebook App empowers you to adjust and run your scratch pads through a web program. The application can be executed on a PC without Internet access, or it might be presented on a remote server, where you can get to it through the Internet. Its two standard fragments are the parts and a dashboard. A piece is a program that runs and introspects the customer's code. The Jupyter Notebook App has a section for Python code, yet there are moreover bits available for other programming languages. The dashboard of the application not simply exhibits to you the scratch cushion reports that you have made and can restore anyway can similarly be used to manage the bits: you can which ones are running and shut them down if fundamental.

Chapter 5. Implementation

5.1 Scrapy:

In python, to implement web-crawling we use an open source platform called as Scrapy. We use web crawler to extract data from websites, in order to perform sentiment analysis. It initially started in the year 1999, by fetching the zip codes from websites. Many organizations implement scrapy to scrape data from web. Companies like PriceWiki, Taralabs etc [1]. We will use scrapy to scrape the reviews from the amazon website in python programming language. By using web scraping, we go fetch title of the product, stars of that product and finally the review comment from the user [1]. There are certain rules and regulations to make this scrapy legal [2]. We ensured those terms and conditions to make our scrapy legal. Reasons for using scrapy is to do sentiment analysis for the product reviews. This leads to understanding for the organization in public sentiment towards their product. By using these reviews, an organization can improve their products or reduce the failure over the next product [29]. Let us discuss the different stages of scraping amazon reviews from website [1]. There are lots of website, we have to find the structure pattern of each html page in order to fetch the desired information in the system.

(i) Understanding HTML structure of amazon web page

Initially, we have to find the pattern of the website in order to fetch the data from the website. Comments, title and stars have a particular classes, ids and other patters in a particular manner. We have to deep study those patters before writing our scrapper.

(ii) Scrapy parser implementation

Next step is we have to write our python code to target the web page and fetch the data's. Once the web page link is updated, scrapy parser will target the web page and scrap those data.

(iii) Fetching and Storage

Finally, the collected information are dumped into csv or json file. Those final data are stored which will be useful for us to scrap the strongest positive, strongest negative and neutral comments from the reviewers.

Installing scrapy in our system by initiating the command "pip install scrapy". Next step is creating space for our project. Once the project is created, it automatically creates one folder

ad one file. Folder which contains the code for scraping and scrapy.cfg is a configuration file which deploys our project on the server. In the HTML page of the amazon web page, the division id cm_cr-review_list, which holds both the star ratings and review comments. Inside this div, we can see tow class attributes namely review text and review-rating, these two attributes consist of stars and comments of reviewers. Now we are going to use these patterns to fetch the data from the websites. In our program "amazon_reviews.py", we are using a list to store the multiple websites for our scrapy to scrape the data from web page [1]. Then it is passed to the function to append with the list of page numbers on the website. Now once we run the spider file, we have to review our output file reviews.csv to view all our web data. "scrapy runspider" This command will run our "amazon_reviews.py" python file which performs scraping.

5.2 Clean Text:

Text is the type of data. It is a sequence to give a meaning. Dealing with language modelling or natural language processing, cleaning data is the whole important process. Indivial text do not have lot of context, characters like "a","b", etc can be rearranged to form a new word.

NLTK:

Sentiment analysis helps the business holder to understand the experience of the customers over a particular product by analyzing their emotions shown towards the product. Businesses can receive feedback from the customers either they are happy or sad with the product to gain deep knowledge about the product. In this project, we going to perform sentiment analysis over a product review using machine learning techniques. To perform machine learning techniques and textual analysis, python is the best programming language as it has an open-source and it is effective in working with huge amount of data sets and performing mathematical calculations. Natural Language Toolkit (NLTK), is one of the best packages in python for performing sentiment analysis.

NLTK is an open-source natural language processing for performing sentiment analysis and it is available in python. NLTK has the capability of performing stemming, tokenization, and other computational linguistics. NLTK can be installed in python by issuing the command

“pip install nltk”, This command will download and install NLTK packages. NLTK comes with the inbuilt analyzer module of sentiment which makes our work easy during the findings of sentiment towards the reviews from the customer [4]. We can create our own classification model using the naïve bayes. Naïve bayes is a probabilistic model based on bayes theorem that will build a classification model from training data [15]. This particular classifier helps us to find the positive, negative and neutral by the supervised learning algorithm. We feed the small amount of data to train the model into our system.

Thus, the output we are getting from the system either will be positive, negative or neutral from the comments, the sample data which we have consist of an array of sentences with their class types like positive or negative or neutral to train the classifier. After the tokenization we have all the words by tokenizing by breaking the first sentence [4]. Then later we try to find the accuracy of the tokenized words in the system. In our model, we imported the stopwords, wordnet, and Lancaster stemmer to perform the operations. Stopwords module consists of all the words such as the, an etc. While wordnet helps us to do the stemming as well as lemmatization [27]. This kind of wordnet is better than stemming because this will give the common meaning for the different words. Which will be effective processing in the system. Which has a faster response as well as gaining the output and fetching the accurate values will be easier in the system. Wordnet holds a huge lexical database with English. All the noun, verb, adjectives etc are stored in the combination of synonyms. Thus wordnet helps us being a useful tool for computational linguistics and natural language processing. Wordnet groups up the words all give the meaning of that group in common which makes our system faster in providing the results to the system. Lancasterstemmer performs the word stemming and lemmatization. Which will benefit in the performance of the system. Thus the system will be more efficient for the system.

Beautiful Soup:

Beautiful Soup is one of the python libraries, we are using this library to fetch the sentence from the documents. The purpose of using this library is to minimize the programmer's hours of work. We are using the command of "Beautiful Soup ("our sentence",'lxml'), get_text()". In this command, Beautiful Soup fetches our sentence in the data set using the command

get_text(). This get_text() helps us to fetch the part of sentence from the entire data set in the faster way. Particular tags are fetched and stored in the variable. Initially we are cleaning our text by using the decontracted. Which helps to frame the words, for instance "won't" leads to "will not". In this way we can use these sorts of words to have the accurate polarity and subjectivity in the words.

Word Net Lemmatizer:

Grouping of words in order to define it as a single term process is the Lemmatization. Lemmatization is little different from stemming. We are using this lemmatization in order to perform morphological analysis of the words. Which is one step better at stemming. Examples are, better leads to good. "Word_tokenize" is used to break the sentence into one by one word, in order to perform the lemmatization. Thus, we ensure all the words in the lemmatization gets filtered and updated in the data set [5]. We are using pos_tag (parts of speech) in order to map to the correct character in the lemmatization. "nltk.pos_tag(['feet'])" This tag creates a output of "[('feet', 'NNS')]" . Which will be useful for fetching and matching the right character in the desired output.

Word2Vec:

Word embedding is used for natural language processing, while every word is mapped to real number vectors. It is used to capture the meaning of word in documents. Word2vec is used to learn word embeddings by using a two-layer neural network. Input will be a text document and output will be set of vectors. It implements mathematical operations used to detect the similarities. Thus, the trained words will be useful to fetch similar words close to each other. For example, men, human and women will be clustered in one region and red, yellow etc combines in one cluster.

Gensim:

Gensim will allow us to build word embeddings by trained word2vec models on our dataset. Importing this package in python by issuing a command “pip install genism”. Thus, the genism module will be downloaded and installed and can be used for execution in our model.

we can handle huge amount of data collected using online algorithms and data streaming. Gensim has streamed parallelized implementation of faster text. It is open source developed.

5.3 Finding Polarity and Subjectivity:

One of the python libraries used to do some Natural Language Processing tasks. Operations such as sentiment analysis, translation, extraction etc can be performed. Polarity results are in float value from -1 to 1. Where 1 is positive, -1 is negative and 0 is neutral. Subjectivity is termed as personal opinion, emotion or judgement from the review. It is also float value ranges between -1 to +1. Initially we created two list namely polarity and subjectivity in order to store the values of all possible sentences score [28]. Then we are creating a for loop for fetching one by one sentences. Inside out for loop, we are creating a variable “analysis” and defining the TextBlob(each sentence”). Now we are going to perform sentiment analysis in the process. So now we are going to find the polarity and subjectivity score and append it to our list. To issue that, the command is “sentiment.polarity” which will find the polarity score of that particular sentence and append the value to the polarity list. After findings of polarity and subjectivity score, we are going to store to the pandas dataframe, in order to do further visualization. We are going to create a column known as rating for defining the comments are positive, negative or neutral. It will helpful to view how many positive, negative and neutral comments. We will use the matplotlib to display the pie chart. “kind=pie” command will be useful for displaying the pie chart. We use the green for displaying positive comments, red to display negative comments and orange for displaying the neutral comments.

5.4 Improving Text Handling:

Pickle:

pickle can be used to serializing and de-serializing python object structures. Serialization refers to converting a python objects into byte streams that can be saved on disk or sent to a network. If we wanted to use that python object again, we can de serialize that pickle file, which is easier and faster process [14].

It will be useful for storing python objects to a database and transmitting data over a transmission control protocol. Pickle will be effective usage on working with machine learning algorithms, where we can save them to be able to make new set of predictions and not rewrite everything or train the model. If we are sending data to other programming languages then pickle is not useful, it can be useful only if it sticks with python programming model. Once the pickle file cannot be unpickled with the different python version used for pickling. The version should be same for both pickling and unpickling [14].

we can use pickle with various python datatypes such as Booleans, integers, floats, complex numbers, strings, tuples, list, sets, dictionaries. All the above datatypes can be pickled.

To import the pickle file, we have to use the command "import pickle". Let us give an example on pickling and depickling a dictionary. Dictionary consist of a key and a value, so we going to pickle the dictionary and save it in a file and later unpickle it and use it in python. Let us consider we have a dictionary called "pip_dict", inside our pip we have list of keys and values. For instance keys as name and value as version. To open a file, we use a function called as "open()". We use this to write our pickle into that file. While the second argument is "wb" which means writing in binary mode. once the file is opened, we use the command "pickle.dump()", which has two arguments, the object which we are going to pickle while the second is the filename in which we going to store the python object. Once we done with writing the file we have to close the file issuing the command "close()".

Let us unpickle the python objects from that file. It is same as we use "open()" but this time we use 'rb' as second argument. Which means read in binary mode. Now we use "pickle.load()", in order to load the pickle file into our program. when it's done, we need to use "close()" to close the file at the end. filename="pip", We are defining the file name in order to fetch.

infile = open(filename,'rb'), We are passing the file name into the new variable to unload it. "pip_new_dict=pickle.load(infile)", Now the pickle file is unloaded and it can be fetched. Finally, we close infile() in order to finish the entire process. We can also check with both the variables like pip_dict==pip_new_dict, it will return True, since we use this process. The process of using pickle is very faster and efficient, while it can be created as a backup which makes our system efficient and providing faster response time.

Regular expression:

We use a regular expression, used to identify patterns in a sentence. We can to replace them with our text by identifying the patterns which we want to replace. If we wanted to do with a particular text, we can identify and fetch those values using the regular expression. For instance, if we fetch some values from the online, use it to extract the brand name and add it to the database.

5.5 Lambda Architecture:

Streaming data helps us to provide a faster response in providing details. Streaming data having the continuous input data which carries useful information can be filtered on finding the accurate results. It has important characters such as providing availability and accessibility helping in analysis and fetching useful information from data [11]. It has the old data and compares with the new data coming out from the streaming leads to accurate output. Spark streaming is considered as processing massive amounts of data.

Processing mega data sets is heavy to maintain real-time updates. We need a architecture to have our data processing quickly as well as accurate, so we use the lambda architecture. Stream processing allowing to process the data real time at faster time as well as detecting conditions in less time. The latency in stream processing is faster than batch processing.

Lambda Architecture:

Processing mega data sets is heavy to maintain real-time updates. We need an architecture to have our data processing quickly as well as accurate, so we use the lambda architecture. Lambda architecture is a fastest data processing architecture that will handle massive amount of data. It takes advantage by performing both batch and stream processing methods [11]. This approach has high throughput, fault tolerant and latency which makes our process effective and quicker. Stream processing allowing to process the data real time at faster time as well as detecting conditions in less time. The latency in stream processing is faster than batch processing.

As technology is increasing day by day, there is massive amounts of data generating on daily basis. Managing massive amounts of data in real-time have cropped up. This gives motivation for creating a processing architecture on producing frequencies of fetching positive, negative and neutral comments. Major requirements of creating such a processing architecture needs to be scalable, fault tolerant and extensible. Such lambda architectures can be benefitted on fetching the reviews which will be useful for the manufacturers on building up on the best product based on the review of the old project [11]. Hence predicting the problem of the product from the reviews is meaningful, such continuous data from lambda architecture allows us to have a meaningful output in order to have better results from the product.

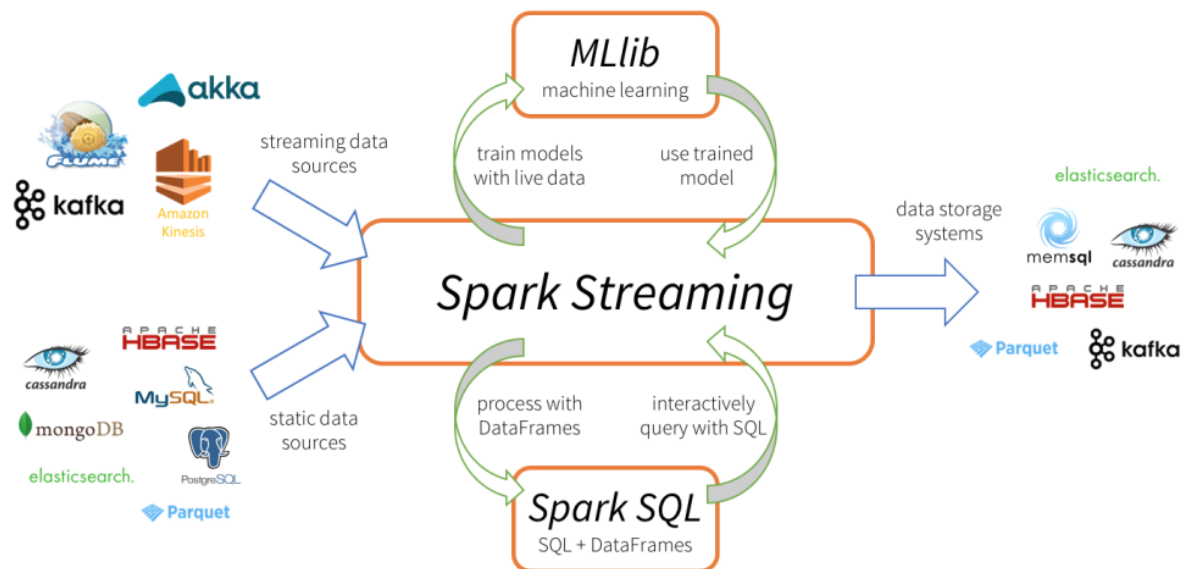
Worst reviews can have the chance of decreasing popularity from buying that product, while good reviews help the consumers to select the product easily and these positive reviews might help the manufactures make good sales on product. For example, tesla motor launch had lot of positive reviews which improved their stocks drastically. such idea from customers helps us to understand their mood on the product.

There are multiple ways to feeding data to the lambda architecture, we preferred to use the spark streaming. Once the data are processed, it is pushed to the databases namely Cassandra. Apache spark helps us to do analysis on spark streaming. Apache spark being a open0source platform that will process massive amount of data for data analysis [11]. Apache spark is programming model that will form the Resilient Distributed Dataset. RDD helps to handle the fault and latency of the processor.

Architecture includes three layers namely batch layer, speed layer and serving layer. The data which is feed into the batch and speed layer simultaneously. Batch layers store the old data which is not changed during the process and keeps on updating the data every second. Now these data are precomputed in certain amount of time. Speed layer reducing the low latency of the overall architecture with the new data. Finally, serving layer helps us displaying the information retrieved from the historical data.

To make this model as working, we ingest amazon reviews into our lambda architecture. The feed data is stored in the text file and stored in the text file is fetched into the textfile stream from the spark streaming. Streaming data is feed into architecture every second in the form of

reviews per second through spark. The model for sentiment is analyzing through NLP tool. Now we define and classify the sentiment into three classes into positive, negative and neutral. Various tools are used to form the architecture in the fastest way.



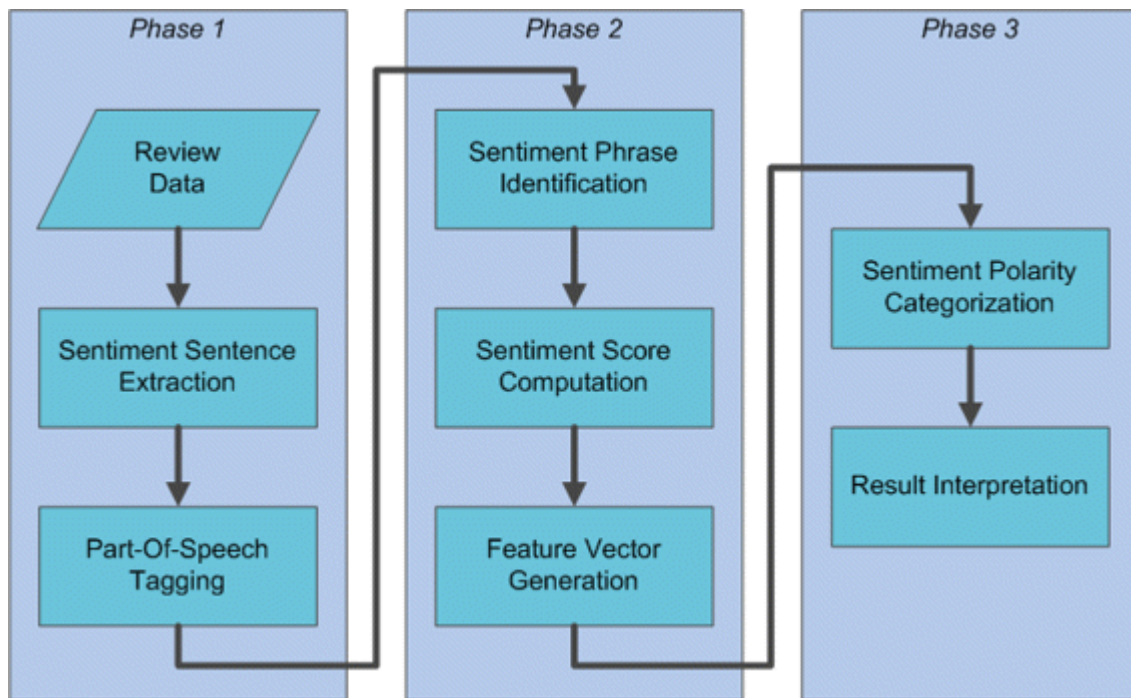
Processing mega data sets is heavy to maintain real-time updates. We need architecture to have our data processing quickly as well as accurate, so we use the lambda architecture. Lambda architecture is a fastest data processing architecture that will handle massive amount of data. It takes advantage by performing both batch and stream processing methods. This approach has high throughput, fault tolerant and latency which makes our process effective and quicker.

Stream processing:

Allowing to process the data real time at faster time as well as detecting conditions in less time.

The latency in stream processing is faster than batch processing.

Flow of Sentiment Analysis:



Big data system with a series of layers known as Lambda architecture. There are three layers namely Batch layer, Serving layer and Speed layer. It is a technique for data processing with huge amounts of data in effective manner. Batch layer manages the master data set, serving layer indexing batch views so that they can be queried in low-latency, speed layer deals with the new data. Precompute the query function deals to be accessed quickly.

Batch layer:

New data is feed into the system. For every second new data is fed into batch and speed layer. It creates the batch view, which has a high latency operation because the function runs on all data. When it finishes the function, new data are collected. The batch views are precomputed by the component known as the batch layer. Batch layer maintain a master data set by storing all the immutable data. Now it creates a batch view from the master data set.

Serving layer:

These batch views are to be loaded and queried effectively. Here the serving layer comes into play by loading the batch views and making them query-able.

Speed layer:

It produces the results on the basis of supporting serving layer to reduce latency.

"Query = λ (Complete data) = λ (live streaming data) * λ (Stored data)" Equation for lambda architecture.

Working:

Our aim is to fetch the strongest positive, negative and neutral comments from each batch. So, we are diving the data set into three pandas namely dfp(positive), dfn(negative) and dfnn(neutral). So, once this command is activated it divides the data into batches and passes into the lambda architecture. In order to create certain batch we are diving the file into several batches equally to fetch the data. So, we are creating a list of csv files in order to activate and fetching the data in streaming. The conditions we are giving for each csv file is separating into the tab value, in order to access these data easily and effectively. Now lets discuss working about the lambda architecture,

PySpark:

It is initiated by the command "pyspark.streaming.StreamingContext". The purpose of Streaming context is to have a connection with spark cluster and to create Dstream with various input values. Once the Dstreams is created and transformed, it can be either started or stopped by "context.start() and context.stop()". While "context.awaitTermination()" is useful for making thread waiting to get terminate by the context by stop().

The next command is "from pyspark.sql import Row". It is a row in a Data frame. we can access the value but mentioning the keyword "row.key" and "row[key]". Thus, the row in our program is useful for fetching the values from the speed layer and store these values in the data store. While row [0] fetches our comment and row [1] fetches the count of the words in the speed layer.

textFileStream:

This textfilestream is useful in creating an input stream that keeps tracking of Hadoop-compatible system for reading the new text files. Now it creates new text files and move those files to the desired directory in order to access those data from the new data files.

.map:

This function fetches the data and makes all the column separated by the tab space. Then we use another map function for data cleaning from the fetched data.

.filter:

Now we use the filter option to choose the desired brands. Since we are fetching lots of brands from the website, we have chosen the desired brand in the data, to obtain the strongest positive, negative and neutral comments in the database.

frequency:

the next step is obtaining the desired column and finding the frequency of each and every data in the database in order to have the strongest comment for improvement in the product for future sales.

Advantages of Lambda Architecture:

smart grid enabling us to deal with huge amount of data to be aggregated and analyzed in such a way that we can fetch the accurate results in the model. traditional smart grid will not allow you to work sufficient storage and faster processing in the system. The present system uses the Hadoop big data in order to do a faster response in the system to have an efficient system. We can use real-time visualization and data mining applications which can be used with the smart system.

We can use Hadoop which is an open-source platform that will allow us to work on the massive amount of data. We use the distributed storage file system in order to handle the streaming live data. A given file is stored is split into blocks and these blocks are stored into the set of data nodes.map-reduce is one of the processing components of the Hadoop. Each consists of a master node and the slave nodes to handle the big data. The slaves will work as it was instructed by the master nodes. MapReduce and HDFS will execute and run on the same node which makes the task to be scheduled on the nodes in the system. The basic idea behind the lambda architecture is to perform the compute arbitrary functions on the distributed datasets in real-time.

5.6 Apache Cassandra:

One of the leading distributed databases for big data management is the apache Cassandra. Which have lot of advantages such as zero downtime, linear scalability and many data center deployed. Widely many web scale companies use the Cassandra for online transaction processing, therefore there's a growing requirement for practical data modelling model that should provide efficient schema design. Apache Cassandra manages huge dataset across variety of multiple data centers. Cassandra use cases such as sensor data, IOT, social networking etc and still many more applications. It has the significant difference between the traditional data model approach and this model.

Traditional data model technique in relational database, It has the defined shapes of database findings and workings. It follows the path such as understanding and arranging data into relations, minimizing data duplicates which is a primary focus of the traditional model in the system. Next, Queries plays a second role in the schema design. Traditional database model are data driven model in which a piece of data will define patterns and uses materialized view to minimize the frequently executed queries in the system [18].

Cassandra data model provides the best read and write performance for the specific set of queries in which our application needs to be performed to do some findings in the system. Cassandra data modelling initiates with application queries. CQl does not support like joins and data aggregation in which the traditional data models had. Efficient Cassandra data modelling design relies over the data nesting or denormalization to perform the complex queries over the single table which have the best performance in the system. It is pretty straight forward that the same values are stored across the different Cassandra tables in order to have the efficient and faster performance in the system which leads to a data duplication in the model. This states that the traditional data modelling approach is totally different with the data modelling approach of Cassandra.

Cassandra solution architect is provided which has the role of providing both the database and application design in the system, by capturing all the application data and defining the application workflow with the database. The solution architect works on the conceptual data model to data model in logical. Now the logical model works efficient and supports the application queries in the workflow and promises the effective performance in the system. Now, physical data types, keys etc are applied over the physical data model which can be instantiated in Cassandra using CQL. This makes our system more effective by capturing all

the necessary data and avoiding the data duplication compared to traditional data model. Thus, it is used by the many web companies to handle the big data [18].

As the data storage we use the apache Cassandra database. Because it is the right option to obtain scalable and high available. It is a open source distributed data store to handle large amounts of data across the servers [7].

we create a function namely "save(time,rdd)". We are passing the timestamp as well as rdd in the data store. Then we create a variable to connect with the spark and Cassandra using the command "spark.createDataFrame", then we pass the rdd inside the data frame by mapping those data. Then we allocate the rows according to the data. we will pass the date, count of the value and finally the processed comment inside the data store. "df.write.format" option will write the data into the data store. Inside the write format command, we will be passing the key space name, table name and finally append option will update the value in the data store. Now let us discuss inside the Cassandra.

Once we are connected to the server, we will initiate the command "sudo service Cassandra start" which will start the Cassandra. This will inform the file system we going to start the Cassandra. And posting this command "cqlsh", will logs in into the Cassandra command prompt. here we can create alter delete etc perform actions on data store. Next step is getting into the key space. If there is no key space, we have to create one, after creating key space, we have to initiate the command "use streaming". This command will allow us to get into that streaming. Now we have to create the table name which is same as we initiated in the program. After creating and running the command of .save(), all the values is stored in the data store. Then we can initiate the query in order to find the strongest comment of either positive, negative and neutral comments in the data store.

RDD:

Resilient Distributed Data set, these elements are responsible for running and operating on multiple nodes in order to perform parallel processing. RDD once created cannot be changed during the process. If they have any sort of failure, it can be easily recovering by its own. We can perform various operations on the RDD. There are two different ways to perform operations on RDD they are Transformation and Action. List of Transformation operations

are Filter, group-by and map performing on RDD's. On the other hand, Action is used to perform computation operation and send the results back to the driver.

count(): Count() function is used to calculate the number of frequencies in the system.

o/p: Number of elements in RDD → 8

collect(): collect() function is used to return all the values in the RDD.

o/p: "Elements in RDD -> ['scala', 'java', 'hadoop', 'spark', 'akka', 'spark vs hadoop', 'pyspark', 'pyspark and spark']"

foreach(f): foreach() helps to return the values that meet the desired requirement mentioned inside the function.

o/p:

cala

java

hadoop

spark

akka

spark vs hadoop

pyspark

pyspark and spark

filter():

This will create the new RDD by fetching only the filtered values in the old RDD.

condition is fetching the word spark existing in the term:

o/p:

Filtered RDD -> ['spark', 'spark vs hadoop', 'pyspark', 'pyspark and spark']

map(): map() is used to find the frequencies of the occurrence in that operation.

o/p:

Key value pair -> [('scala', 1), ('java', 1), ('hadoop', 1), ('spark', 1), ('akka', 1), ('spark vs hadoop', 1), ('pyspark', 1), ('pyspark and spark', 1)]

reduce(): we can perform various binary operations, the elements in the RDD is returned. For instance in the above function we are carrying out the addition function in the system.

o/p:

Adding all the elements -> 15

join(): join() function is used to join two RDD into a single RDD.

o/p:

Join RDD -> [('spark', (1, 2)), ('hadoop', (4, 5))]

5.7 Visualization:

We are starting off by displaying the number of stars presented in the data set. Because each brand having a list of stars. Before going further analyzations, we have to come across the number of stars presented in the data set. For this purpose, we are using the matplotlib in order to display our visualization techniques.

TextBlob:

TextBlob is one of the python libraries for processing and visualising textual data. It providing a various operation like speech tagging, sentiment analysis etc. “pip install TextBlob” is used for installing the TextBlob and we going to import by issuing a command “from textblob import TextBlob”.

Wordcloud:

In order post all our positive reviews in a single image, we have to fetch all the list of words which have more frequencies and post in the image. Therefore, we create a new column which will store all the words in that particular column. We will be using `column.agg()`, it returns multiple results. It is used to pass list of functions to be applied on a series. It is used to fetch all single elements and gives a faster result which will be used to give the desired results faster.

In order to display image with positive words, we will create a new dataframe which holds the positive reviews. Once the dataframe is created is passed to the function and made it to available to the wordcloud in the function.

We will be representing text data as a image in which size indicates its importance and frequency. It acts a data visualization tool. It helps us analysing customer feedback and fetching the most important positive or negative word in the image.

Chapter 6. Testing and Evaluation

Data collection is portrayed as the arrangement of social affair, assessing and separating definite bits of learning for research using standard affirmed techniques. A researcher can survey their theory dependent on accumulated data. All things considered, data gathering is the fundamental and most noteworthy development for research, autonomous of the field of research. The strategy of data social occasion is different for different fields of study, dependent upon the required information.

Pre-getting ready implies the progressions associated with our data before urging it to the count. Data Preprocessing is a strategy that is used to change over the unrefined data into an unblemished enlightening gathering. So, to speak, at whatever point the data is collected from different sources it is assembled in rough association which isn't attainable for the examination. For achieving better results from the associated model in Machine Learning adventures the association of the data must be in a real manner.

For example, consider a model that predicts whether an audit is phony, using the title, surveys, and checked buy as features. We appropriate the data into getting ready and test sets, with a 80-20 split. In the wake of setting up, the model achieves 99% precision on both the readiness set and the test set. We'd expect a lower precision on the test set, so we take a gander at the data and locate that countless the models in the test set are duplicates of models in the planning set (we neglect to scour duplicate segments for a comparable spam email from our data database before separating the data). We've accidentally arranged on a bit of our test data, and along these lines, we're never again decisively assessing how well our model aggregates up to new data.

Scoring is additionally called polarity accuracy, and is the way toward producing esteems dependent on a prepared AI model, given some new input information. The qualities or scores that are made can speak to sentiment of customers over products, yet they may likewise speak to a reasonable class or result. The importance of the score relies upon the sort of information was given, and the kind of model that was made.

In the trial results, the crude information from the module scrapy is gotten through the python library. The panda's library in python is utilized in order to peruse the information such that

will be appropriate for investigation like arrangement or dataframe structure. Pandas has different info and yield arrangements, for example, CSV,HTML,XLS and so forth. From the outset, the required libraries are imported so as to not demonstrate mistake on fundamental capacities.

One of the rich data structures python library is Pandas. We are using the pandas for performing analysis as well as feeding data into the lambda architecture. While importing the dataset we introduce certain conditions like removing the null lines as well as informing the pandas that the delimiter should be ",". The purpose of using pandas is handling the missing data, insertion and deletion of data in data frame and performing some analytics using the word cloud to feed all the words in the comment in the image representation. One of the finest libraries in python for handling data is Pandas [3].

Numpy are representations for numerical data and efficient implementation in a high-level language. Numpy performance can be improved in three ways vectorizing calculations, removing duplicates in memory, and reducing operation count. we can call NumPy array as ndarray, short for N-dimensional array. data pointer holds the memory address in the array. data type description is the kind of elements such as floating-point numbers or integers. shape describes a mesh grid x-, y- and z-coordinates. strides the number of bytes which was skipped in memory to proceed to the next column. flags defined the factors in which rows are stored one after another memory [17].

Numpy memory providing a powerful way of fetching the same memory. we can create a ndarray using the the arrange() which will post the values and reshape() will give the desired shape of the numpy. `x=np.arange(9).reshape(3,3)`, As it was defined earlier, it holds the integers values like from 0 to 9 and gives the shape of 3:3. x holds the numpy array with the following value. We do not need to be created using the slices, by modifying strides, We can enable a pleathora of different ways in the underlying data [17]. we should use the performance measures in order to do the information retrieval. It includes the recall, precision and accuracy. To calculate the precision and recall, we use four cells like true positive, true negative, false positive and false negative. Using these details, the values are calculated.

Recall:

recall is the proportion of real positive cases that are correctly predicted positive.

$\text{recall} = \text{tp} / (\text{tp} + \text{tn})$ correct details is divided by the number of results should have been returned in the system.

Precision:

It is a proportion of predicted positive cases that are correctly real positive.

$\text{precision} = \text{tp} / (\text{tp} + \text{fp})$

It displays the correct results dividing by the number of overall results.

Accuracy:

Accuracy helps us identifying, how accurate our results will be displayed. the accuracy is calculated by $\text{accuracy} = (\text{tp} + \text{tn}) / (\text{tp} + \text{tn} + \text{fp} + \text{fn})$. Training the model with knn classifier, “k” is the nearest neighbors to take vote from. Hence a circle is created as big as to enclose the datapoints on the plane. Good confidence level, that our class value matches with the respective values in the plane. This parameter is little bit crucial on matching with the reviews received from the customers. Thus, boundaries are made with the respective datapoints of each class. Finding the “k” effect value of each class. If the value of the polarity and subjectivity matches then our boundary matches with respective class values.

Training error and validation error are two parameters need to come across. At error rate $k=1$, the value is always zero for training sample. Hence the nearest value in any training point is itself. The prediction will be accurate with k value as 1. It is pretty clear that, at value 1 in k, leads to overfitting with boundaries. The value of k should be used for every prediction.

Training with Naïve Bayes algorithm, there are three classes (“positive”, “negative” and “neutral”). With examples of few reviews in each category, training our Naïve Bayes classifier so that new reviews are automatically categorized.

Training set will consist of set of reviews. A term-document matrix (TDM) comprises of a rundown of word frequencies showing up in a lot of reports. The TDM matrix is a rectangular matrix of set of words and m documents Furthermore, it is sparse, it contains for the most part zeros. The entry(i,j) of the TDM matrix depicts to the recurrence of word "I" in report

"j". Now the TDM frequencies are calculated to find frequency and occurrence of each term. The assumption of Naïve Bayes classifier is that each term is independent to each other.

Chapter 7. Conclusions and Future Work

Developed algorithm will analyse the polarity of the reviews, each portion of text in the reviews. The polarity and subjectivity value represent the strength and weakness of the product. This algorithm combined with SVM classification and Knn to develop approach for the sentiment analysis. Companies can get benefitted by predicting the customer sentiment and use it against improving the product. In traditional ways fetching strongest comments are always been a hard to be explored. Using our model can help in decision making and understand what specific customer concerns on both the good and bad opinions about the product. Final result analysis helps manufacturers yield comprehensive decision support information.

Giant company may get profited from this type of sentimental and text mining process. Majority of fake reviews can ruin the profit and standard of a huge company. Model created will help them to overcome such problems at the initial stage and the data displays on market sentiment for company's benefit very quickly. Massive amount of data processing requires huge amount of data, but having a architecture that processes the data quicker and efficient results will help to take the decision making as fast as possible to improve the standard and respond to the customer problems sooner.

Further work can be done on this research. Predicting positive, negative or neutral was always been a challenging part in the real world. With our system it can be done at the faster rate. Therefore algorithm design can be improved on visualizing the information, which can be useful for manufacturers to dig deep into this issue. A practical website in the form of website portal is desired to be as our major future work. This type of system will help the customers to have the same privilege same as the manufacturers have. Communication between the customer and the manufacturers on dealing with the quality of the product can have a greater impact on the success of the product. The system is expected to possess on:

- a) Verified users can be able to access the website to view the product strength and weakness.
- b) Users can able to forecast on the discussion and make immediate faster results from a massive amount of data.
- c) Users can able to view the sentiment polarity average to make sure their complaints can be valid and addressed by the manufacturers.

References:

- [1] D. Myers and J. W. McGuffee, "Choosing Scrapy," *J. Comput. Sci. Coll.*, vol. 31, no. 1, pp. 83–89, Oct. 2015.
- [2] "Is Web Data Scraping Legal?," *Datahut - Blog*, 30-Oct-2018. .
- [3] W. Mckinney, "pandas: a Foundational Python Library for Data Analysis and Statistics," *Python High Performance Science Computer*, Jan. 2011.
- [4] E. Loper and S. Bird, "NLTK: the Natural Language Toolkit," *CoRR*, vol. cs.CL/0205028, Jul. 2002.
- [5] "Python | Lemmatization with NLTK," *GeeksforGeeks*, 06-Nov-2018. .
- [6] "Customizing matplotlib — Matplotlib 2.0.2 documentation." [Online]. Available: <https://matplotlib.org/users/customizing.html>. [Accessed: 02-Sep-2019].
- [7] "Python | Introduction to Matplotlib," *GeeksforGeeks*, 14-May-2018. .
- [8] B. Pang and L. Lee, "A Sentimental Education: Sentiment Analysis Using Subjectivity Summarization Based on Minimum Cuts," in *Proceedings of the 42Nd Annual Meeting on Association for Computational Linguistics*, Stroudsburg, PA, USA, 2004.
- [9] B. Liu, E. Blasch, Y. Chen, D. Shen, and G. Chen, "Scalable sentiment classification for Big Data analysis using Naïve Bayes Classifier," in *2013 IEEE International Conference on Big Data*, 2013, pp. 99–104.
- [10] C. Troussas, M. Virvou, K. J. Espinosa, K. Llaguno, and J. Caro, "Sentiment analysis of Facebook statuses using Naive Bayes classifier for language learning," in *IISA 2013*, 2013, pp. 1–6.
- [11] M. Gribaudo, M. Iacono, and M. Kiran, "A performance modeling framework for lambda architecture based applications," *Future Generation Computer Systems*, Jul. 2017.
- [12] S. Das, R. K. Behera, M. kumar, and S. K. Rath, "Real-Time Sentiment Analysis of Twitter Streaming data for Stock Prediction," *Procedia Computer Science*, vol. 132, pp. 956–964, Jan. 2018.
- [13] R. Safrin, K. Sharmila, T. S. S. Subangi, and E. A. Vimal, "SENTIMENT ANALYSIS ON ONLINE PRODUCT," 2017.
- [14] "Understanding Python Pickling with example," *GeeksforGeeks*, 08-Jun-2017. .
- [15] S. I.V., "Sentiment Analysis in Python using NLTK," *OSFY - OpensourceForYou*, Dec. 2016.
- [16] "Regular Expressions: A Brief Tutorial." [Online]. Available: <http://misc.yarinareth.net/regex.html>. [Accessed: 02-Sep-2019].
- [17] S. van der Walt, S. C. Colbert, and G. Varoquaux, "The NumPy Array: A Structure for Efficient Numerical Computation," *Computing in Science Engineering*, vol. 13, no. 2, pp. 22–30, Mar. 2011.

- [18] A. Chebotko, A. Kashlev, and S. Lu, "A Big Data Modeling Methodology for Apache Cassandra," in *2015 IEEE International Congress on Big Data*, 2015, pp. 238–245.
- [19] S. Mukherjee and P. Bhattacharyya, "Feature Specific Sentiment Analysis for Product Reviews," in *Computational Linguistics and Intelligent Text Processing*, 2012, pp. 475–487.
- [20] H. Cui, "Comparative Experiments on Sentiment Classification for Online Product Reviews," p. 6.
- [21] W. Zhang, H. Xu, and W. Wan, "Weakness Finder: Find product weakness from Chinese reviews by using aspects based sentiment analysis," *Expert Systems with Applications*, vol. 39, no. 11, pp. 10283–10291, Sep. 2012.
- [22] S.-M. Kim and E. Hovy, "Determining the Sentiment of Opinions," in *Proceedings of the 20th International Conference on Computational Linguistics*, Stroudsburg, PA, USA, 2004.
- [23] W. Zheng and Q. Ye, "Sentiment Classification of Chinese Traveler Reviews by Support Vector Machine Algorithm," in *2009 Third International Symposium on Intelligent Information Technology Application*, 2009, vol. 3, pp. 335–338.
- [24] S. K. Bharti, K. S. Babu, and S. K. Jena, "Parsing-based Sarcasm Sentiment Recognition in Twitter Data," in *Proceedings of the 2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2015*, New York, NY, USA, 2015, pp. 1373–1380.
- [25] A. Khadjeh Nassirtoussi, S. Aghabozorgi, T. Ying Wah, and D. C. L. Ngo, "Text mining of news-headlines for FOREX market prediction: A Multi-layer Dimension Reduction Algorithm with semantics and sentiment," *Expert Systems with Applications*, vol. 42, no. 1, pp. 306–324, Jan. 2015.
- [26] S. Jiang, G. Pang, M. Wu, and L. Kuang, "An improved K-nearest-neighbor algorithm for text categorization," *Expert Systems with Applications*, vol. 39, no. 1, pp. 1503–1509, Jan. 2012.
- [27] M. Kumar and A. Bala, "Analyzing Twitter sentiments through big data," in *2016 3rd International Conference on Computing for Sustainable Global Development (INDIACom)*, 2016, pp. 2628–2631.
- [28] M. Krantzlein and D. C. Lo, "Training on the poles for review sentiment polarity classification," in *2017 IEEE International Conference on Big Data (Big Data)*, 2017, pp. 3934–3937.
- [29] I. E. Alaoui, Y. Gahi, and R. Messoussi, "Full Consideration of Big Data Characteristics in Sentiment Analysis Context," in *2019 IEEE 4th International Conference on Cloud Computing and Big Data Analysis (ICCCBDA)*, 2019, pp. 126–130.
- [30] Z. Wu and Y. Lu, "A study on micro-blog sentiment analysis of public emergencies under the environment of big data," in *2017 29th Chinese Control And Decision Conference (CCDC)*, 2017, pp. 4435–4438.

[31] B. Agarwal, A. Ravikumar, and S. Saha, "A Novel Approach to Big Data Veracity Using Crowdsourcing Techniques and Bayesian Predictors," in *2016 15th IEEE International Conference on Machine Learning and Applications (ICMLA)*, 2016, pp. 1020–1023.