**Aspect based Sentiment Analysis using Logistic Regression, Naive Bayes and LSTM**

**ABSTRACT**

Classifying the reviews into a certain category of interest as positive, negative is known as Sentiment Analysis. So we can analyze these reviews using opinion mining methodology. The objective of this project is to do aspect based sentiment analysis. The reviews are identified, then they are pre-processed and then the polarity is calculated. In this proposed methodology we will perform sentiment classification using Machine Learning approach Logistic Regression, Naive Bayes and the Deep Learning approach Recurrent Neural Network (RNN).The Features are extracted from the labeled information which gives us two explicit models. The entire task relies upon the aspects which must create the most elevated accuracy for the reviews. We will be comparing both the models which is later tested on different reviews and then the accuracy is calculated.

### INDEX

|  |  |  |
| --- | --- | --- |
| **TABLE OF CONTENTS** | | **PAGE NO** |
| **ABSTRACT INDEX**  **LIST OF TABLES**  **LIST OF FIGURES** | | **I II V**  **VI** |
| **CHAPTER** | 1. **INTRODUCTION**    1. Aim       1. Opinion mining 1.2.Problem Statement   1.2.1. Existing System 1.3.Proposed System   1. **LITERATURE SURVEY** 2. **PRESENT WORK**    1. Sentimental Analysis       1. Aspect based opinion mining   3.2.Technical Description  3.3.Input dataset   * + 1. Data Preprocessing        1. Stop word removal        2. Stemming        3. Lemmatization | **1** |
|  | 2 |
|  | 2 |
|  | 3 |
|  | 4 |
|  | 4 |
| **CHAPTER** | **5** |
| **CHAPTER** | **10** |
|  | 11 |
|  | 12 |
|  | 14 |
|  | 14 |
|  | 15 |
|  | 16 |
|  | 16 |
|  | 16 |

|  |  |  |
| --- | --- | --- |
| **CHAPTER** | 3.4.Classification   * + 1. Supervised Learning        1. Naive Bayes classifier        2. Logistic Regression 3.5.Project Description     2. Overview of the Project   3.5.1.1.Recurrent Neural Networks  3.5.1.2.LSTM   * + 1. System requirements        1. Hardware Requirements        2. Software Requirements     2. Technical Requirements        1. Choice of Language        2. Choice of Platform     3. Processing steps  1. **RESULTS AND DISCUSSIONS**    1. Output screens   4.2.Performance metrics for classification algorithms   * + 1. Output analysis of Naive Bayes classifier     2. Output analysis of Logistic Regression     3. Output analysis of LSTM 4.3.Comparison of Results   4.3.1.Analysis of Results | 16  17  18  22  24  25  25  32  34  35  35  35  35  35  37  39  **40**  41  41  42  43  42  43 |

|  |  |  |
| --- | --- | --- |
| **CHAPTER**  **CHAPTER** | 1. **CONCLUSION AND FUTURE SCOPE**    1. Conclusion 5.2.Future Scope 2. **REFERENCES**    1. References   **APPENDICES** | **44**  45  45  **46**  47  **48** |

|  |  |  |
| --- | --- | --- |
| **Table** | **Title** | **Page** |
| 3.1.1.1 | Polarity of Aspects…………...………........................ | 13 |
| 3.3.1.1 | Input dataset of reviews on Movies……………….….... | 14 |
| 4.2.1.2 | Confusion matrix Naive Bayes…………………..……. | 40 |
| 4.2.2.2 | Confusion Matrix Logistic Regression……….……….. | 41 |
| 4.3.4.1 | Analysis of results………….…………………………… | 43 |

|  |  |  |
| --- | --- | --- |
| **Figure** | **Title** | **Page** |
| 3.4.2.1 | Main Methodology of Naive Bayes Classifier………... | 20 |
| 3.4.3.1 | Representation of Logistic Regression Classifier.……. | 22 |
| 3.5.1.1 | Flow Diagram for the Project………………………..… | 24 |
| 3.5.2.1 | Recurrent Neural Networks Classifier | 31 |
| 3.5.3.2.1 | Spyder GUI Welcome page…………………………… | 36 |
| 4.2.1.1 | Output of Naive bayes classifier……………………… | 40 |
| 4.2.2.1 | Output of Logistic Regression classifier……..………. | 41 |
| 4.3.2.1 | Output of LSTM classifier……………………………. | 42 |
| 4.3.3.1 | Comparison of Models……………………………….. | 42 |

**CHAPTER-1 INTRODUCTION**

1. **INTRODUCTION**
   1. **Aim:**

Data mining research has successfully shaped numerous methods, tools, and algorithms for handling huge volume of data to solve real world problems. The key objectives of the data mining process are to effectively handle large scale data, mine actionable rules, patterns and gain insightful knowledge

As the internet and its technology is growing, people got the freedom of expressing their views, interests and opinions about the things they see around or use regularly in the form of reviews and feedback. Presently, a day's loads of individuals are utilizing the web and doing web based shopping and in the long run they will search for good things. Today’s service providers or product providers are more interested in the reviews of their customers because they contain the opinion of the customer and/or, his/her interest about that product or service. Service providers are faced challenging issues in finding behavior or interest of their customer.

Since people have different blogs,IMDB, twitter, forum discussions, data analysts require more attention to get the sentiment from the reviews which were posted by the people in the comments. In order to grow up the business of the data analysts, they require special attention to extract opinion of the people about the particular entity.

### Opinion Mining:

Opinion mining is extracting people’s opinions from web. It mainly analyzes people’s opinions, emotions, appraisals, emotions towards organizations, entity(product), topics, issues etc. It involves in building a system to collect and categorize the opinions about a product. Opinion mining is also called as sentimental analysis.

### Levels of opinion mining:

There are three levels of opinion mining.

* + - * + **Document Level:** In this approaches whole document is considered as a single entity and the analysis approaches in applied on the whole document. The result generated in document level sometimes not appropriate.
        + **Sentence Level:** In the sentence level approaches every sentence is considered as an entity and analysis approaches is applied on individual sentence then their result is summarized to provide the overall result of the document.
* **Aspect Level:** Phrase-level opinion mining is also known as aspect based opinion mining. It performs fine grained analysis and directly looks at the opinion. The goal of this level of analysis is to discover sentiments on aspects of items. Nouns or noun phrases which are explicitly mentioned are called explicit aspects.

The proposed system is based on aspect level sentimental analysis. Phrase level sentiment analysis or aspect-based sentiment analysis is the best solution to mine people's interests from online reviews. Aspects are the attributes of the service or product and we have named service or product as an entity. It is a context dependent since people are using different type of text or data in different types of media. Aspect level sentiment analysis works by finding all the aspect terms and mining those aspect terms to get the opinion.

For example, consider a sentence “I am impressed for dell customer service but it is getting motherboard problems”. In this sentence, aspect terms are ‘customer service’ and ‘motherboard’. Opinion or sentiment expressed towards those aspects is ‘impressed’ and ‘problems’. By using our school English knowledge we can say that customer is satisfied with dell’s customer service but unsatisfied with their motherboard performance.

By using phrase level sentiment analysis, we can solve the problems of data analysts while making important decisions. In order to get the sentiment, we need to make a machine to learn and this can be done by supervised learning and unsupervised learning.

Classification predicts the class labels of categorical data. Class labels are predetermined and it builds the classifier model. For some data we train the classifier then we perform testing for new data. This classifier tries to find out the relationship among attributes and classifies based on the maximum relationship in terms of probability.

Supervised text classification algorithms which were famous are Naive-Bayes, Logistic Regression. The accuracy of classification is different for these algorithms and way of classification is also different. Naive Bayes classification algorithms are the more scalable algorithms to classify documents based on the frequency of words. Naive Bayes classifiers consider more number of features for training the classifier. Naive Bayes algorithm follows Bayes theorem by assuming independence of its features.

Logistic Regression was another classifier in the competition in this field. It is better classifier than Naive Bayes. Logistic Regression provides a unique solution which is binary either yes or no.

In view of the above we have developed a system that is used to classify opinion using aspect level classification to get positive and negative aspects. Then to get the sentiment of the testing data ,Naive Bayes and Logistic Regression are used.

### Problem Statement:

By using document and sentence level opinion mining, users are uninterested of any entity. Whereas in aspect based opinion mining by using polarity users get most interesting aspects. The goal is to find interesting aspects based upon polarity and relative importance. Positive and negative aspects will be found based on the relative importance. Finally Naive Bayes and Logistic Regression and Recurrent Neural Networks (RNN) are used to test the accuracy.

### Existing system:

The author provide a survey on hybrid approach to analyze aspect based sentiment of movie reviews. It covers work on explicit and implicit aspects which is crucial for aspect based sentiment of movie reviews. Previous researches have assumed that review level classification which only determines general sentiment of movie reviews. The approach is done by using Dependency Parsing, Association rule mining. Senti wordnet is used to solve aspect based sentimental analysis. The work is done on Hate crime Twitter sentiment data set and Stanford Twitter sentiment dataset.

For racial aspects from hate crime, the approach successfully classiﬁed 28% as neutral, 53% as hate tweets and 19% as free tweets. Besides, it also classiﬁed 24% as neutral tweets, 58% as hate tweets and 18% as free tweets towards sexual aspects. Finally, for religion aspects, the hybrid approach classiﬁed 32% as neutral, 48% as hate tweets and 20% as free tweets.

### Proposed system:

To develop a system that is used to classify opinion using aspect level classification to get positive and negative aspects. The aspect based opinion mining is done based upon the polarity.

Positive and negative aspects will be extracted. Naive Bayes and Logistic Regression are used to get the sentiment of the test data.

# CHAPTER-2 LITERATURE SURVEY

## LITERATURE SURVEY

### Paper-1: B. Pang, L. Lee, and S. Vaithyanathan, “Thumbs up?: sentiment classification using machine learning techniques,” in Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume

### Association for Computational Linguistics, 2002, pp. 79–86.?

### Pang applied machine learning, methods for classifying the online movie reviews, collected from the Internet Movie Database (IMDb), to positive or negative, by obtaining the list of 14 affective key words (love, wonderful, best, great, superb, still, beautiful,bad,worst, stupid, boring, waste,and !)which are then applied straight forward way to form the base-line for classification accuracy. The results showed that using machine learning methods like SVM, Naive Bayes etc.achieved significant improvement over the baseline.

### Paper-2: Sentiment Analysis of Review Datasets using Naive Bayes’ classifier LopamudraDey, Sanjay Chakraborty.

### In this paper for used machine learning algorithm such as Naive Bayes.A well-known Bayesian network classifier is the Naive Bayes’ classifier is a probabilistic classifier based on the Bayes’ theorem. Positive Sentiment in subjective sentence: “I loved the movie Mary Kom”—This sentence is expressed positive sentiment. Negative sentiment in subjective sentences: “Phata poster nikla hero is a flop movie” defined sentence is expressed negative sentiment .The experimental results show that the classifiers yielded better results for the movie reviews with the Naive Bayes’. we can say Naive Bayes’ classifier can be used successfully to analyze movie reviews. more accuracy for naive bayes.

### Paper-3: Jingbo Zhu, Huizhen Wang, Muhua Zhu, Benjamin K. Tsou, Matthew Ma “Aspect-based opinion polling from customer reviews” IEEE Transactions on affective computing, Vol. 2, No.1, January-March 2011

The goal of opinion polling (customer survey) is to discover customer satisfaction on a particular product, service, or business. This is traditionally done by carefully designing some questions for customers to answer. An aspect-based opinion polling system takes as input a set of textual reviews and some predefined aspects, and identifies the polarity of each aspect from each review to produce an opinion polling. This paper focuses on aspect-based opinion polling from unlabeled free-form textual customer reviews without requiring customers to answer any questions. First, a multi-aspect bootstrapping method is proposed to learn aspect-related terms of each aspect that are used for aspect identification. Second, an aspect-based segmentation model is proposed to segment a multi-aspect sentence into multiple single-aspect units as basic units for opinion polling. Finally, an aspect-based opinion polling algorithm is presented in detail.

### **Paper-4: V. Peddinti and P. Chintalapoodi,V.M.Kiran, in sentiment analysis of movie reviews ', Analyzing Microtext Workshop, AAAI, 2011.**

### Some researches made to identify the public opinion about movies, news etc. from twitter tweets. V.M. Kiran et al  had taken the information from other publicly available databases like IMDB and Blippr. collection from the internet   movie reviews by obtaining the list of best ,great ,superb, beautiful ,worst, bad which are used for classification accuracy. The Internet Movie Database (IMDb)  is used to search the input raw data. IMDb.com is a web based database for information of movie, television shows, etc. From the website, the detailed information containing the values for the following: actors, release date, rank of sale, audiences reviews, etc.of a movie was obtained.

### **Paper-5: Minhoe Hur Seoul National University “Box-office forecasting based on sentiments of movie reviews and Independent subspace method”, Information Sciences,2016?**

A number of studies related to box-office performance prediction have already been    conducted Several factors influencing box-office performance have been identified, including  movie director, actors and plot summary as well as marketing activities used to promote the movies ..Because of insufficient sample size, the findings cannot be generalized to other movie genres. Hur et al. (2016) considered three kinds of features (i.e. motion picture, external and review sentiment) to develop box-office forecasting models using four machine learning techniques.

# **Paper-6:**[Blety Babu Alengadan](https://ieeexplore.ieee.org/author/37086242523); [Shamsuddin S. Khan](https://ieeexplore.ieee.org/author/37086039557)["A proposed system for modifying aspect based opinion mining for ranking of products".Third International Conference on Sensing, Signal Processing and Security (ICSSS)](https://ieeexplore.ieee.org/xpl/mostRecentIssue.jsp?punumber=8063614) 2017.

In this the author includes four important phases: Pre-processing, Enhanced Aspect Identification and Opinion Word Extraction with modified naive bayes model, Aspect Polarity Identification, Products and Aspects Ranking in sequence. The developed system will be beneficial for the consumers and business analysts/entrepreneurs who seek customers’ online opinions for analyzing purchase decisions and market statistics and strategies.

## **Paper-7: “[Text sentiment analysis based on long short-term memory](https://ieeexplore.ieee.org/document/7778967/)”[Dan Li](https://ieeexplore.ieee.org/author/37086097311); [Jiang Qian](https://ieeexplore.ieee.org/author/37085455584) [2016 First IEEE International Conference on Computer Communication and the Internet ICCCI)](https://ieeexplore.ieee.org/xpl/mostRecentIssue.jsp?punumber=7757193) 2016.**

This paper the author promotes a RNN language model based on Long Short Term Memory (LSTM), which can get complete sequence information effectively. Compared with the traditional RNN language model, LSTM is better in analyzing emotion of long sentences. And as a language model, LSTM is applied to achieve multi-classification for text emotional attributes. So though training different emotion models, we can know which emotion the sentence belongs to by using these emotion models. And numerical experiments show that it can produce better accuracy rate and recall rate than the conventional RNN.

### Paper 8: “Aspect-based sentiment analysis of movie reviews “ Tun Thura Thet ,Jin-Cheon Na and Christopher S.G. Khoo.

### This paper discussed about a method for automatic sentiment analysis of movie reviews is proposed, implemented and evaluated. In contrast most studies that focus on determining only sentiment orientation (positive versus negative) . , that analyses sentiment in a given textual unit objective of understanding the polarities of the opinions expressed and the types of emotions toward various aspects of a subject. Sentiments, such as opinions, attitudes, thoughts, judgements and emotions, are private states of individuals which are not open to objective observation or the verification. We conducted experiments with a dataset of 1000 sentences: 500 positive and 500 negative. The movie review sentences were manually collected from the discussion board of a movie review site (www.imdb.com). For the experiments, our own dataset was used because aspect level sentiment labels are required to verify the effectiveness of our aspect-based sentiment analysis approach. Most of the publicly available movie review datasets contain only document level or sentence level sentiment labels. Using machine learning models logistic regression ,naive bayes.

### paper 9: “Sentiment Analysis for Movie Reviews” Ankit Goyal,

### Amey Parulekar.

### Movie reviews are an important way to gauge the performance of a movie.While providing a numerical/stars rating to a movie tells us about the success or failure of a movie quantitatively, a collection of movie reviews is what gives us a deeper qualitative insight on different aspects of the movie A textual movie review tells us about the strong and weak points of the movie and deeper analysis of a movie review can tell us if the movie in general meets the expectations of the reviewer. Using sentiment analysis,we can find the state of mind of the reviewer while providing the review and understand if the person was “happy”, “sad”, “angry” and so on. In Logistic Regression model seemed to have best performance across all feature representations. One can also use a Naive Bayes’ Classifier they also provide good accuracy percentage.

# CHAPTER:3 PRESENT WORK

### 3.1 Sentiment Analysis:

1. **PRESENT WORK**

Sentiment analysis is also called as opinion mining to extract people’s opinion from web. The sentimental analysis have become an integral part of the product marketing and user experience as both businesses and consumers turn to online resources for feedback on products and services.

Opinion mining has been an emerging research field in Computational Linguistics, Text Analysis and Natural Language Processing (NLP) in recent years. It is the computational study of people’s opinions towards entities and their aspects. Entities usually refer to individuals, events, topics, products and organizations. Aspects are attributes or components of entities. In the last few years, social media has become an excellent source to express and share people’s opinion on entities and their aspects. With the availability of vast opinionated web contents in the form of comments, reviews, blogs, tweets, status updates, etc. it is harder for people to analyze all opinions at a time to make good decisions. So, there is a need for effective automated systems to evaluate opinions and generate accurate results. Sentiment Analysis, Emotion Analysis, Subjectivity Detection has also become an active research area in recent years along with opinion mining.

Some people express their opinions in binary scale (i.e. Positive or Negative) and some other expresses their opinions explicitly in terms of ratings (i.e. one to three or five stars). The term “Polarity” in opinion mining refers to the orientation scale. The polarity is predicted on either a binary (positive or negative) or multivariate scale using sentiment polarity classification or polarity classification techniques.

According to Bing Liu, Opinion mining tasks are generally classified into three major levels: document-level, sentence-level, and phrase-level. In Phrase level, classification is performed in fine- grained manner. Here, features or aspects of entities are mainly focused and polarity is calculated for each and every individual aspects. For example, the sentence “The laptop’s sound is good, but the battery life is very short” evaluates two aspects (Speaker Quality, Battery Life) of laptop (entity). The sentiment on laptop’s sound is positive and battery life of laptop is negative. Aspect based opinion mining comes under phrase level opinion mining task.

### 3.1.1 Aspect based opinion mining:

People not only express their opinion on documents and sentences, but also in aspects and entities. Level of information provided in document level or sentence level is not sufficient for making a good decision and therefore looking in-depth into aspects and entities gave a new direction for research called aspect or feature based opinion mining. A drawback in document or sentence level is that they cannot provide complete information of a product, For example, a positive or negative review of a particular product doesn’t mean that the reviewer likes or dislikes all aspects of that product. A person who needs to buy a mobile with excellent camera quality will search only for the reviews about that particular aspect i.e. “picture quality” rather than overall review of that mobile.

Single aspect opinions are reviews where people focus only on one aspect of the product, whereas in multi-aspect, people express differing opinions on more than one aspect simultaneously in the same review and even in same sentence of that product. For example, “This book had a good storyline, but the paper quality is bad”. Here, the reviewer gave a positive mention on the “story” aspect and negative mention on the “paper quality” aspect of the book. Segmenting multi aspect sentence into multiple single aspect sentences is called sentence segmentation and it is a challenging task in aspect based opinion mining.

Two major tasks in aspect based opinion mining are aspect extraction and aspect sentiment classification. Process of identifying the opinion words from the given sentence is called aspect extraction and categorizing the extracted opinion words into one of the polarity scales is called aspect sentiment classification. Study shows that adjectives, adverbs and subjective nouns are considered to be aspect related words in most cases and gives high performance.

Consider this review “I was worried about what the sound quality of ipod would be like, but that seems to be just as good as the actual CD sound. The only area I wish Apple had improved upon would be the screen. The only problem is the battery life. After about 3-4 months, you see your battery draining faster than it should.”

|  |  |  |
| --- | --- | --- |
| Aspect | Positive(+) | Negative(-) |
| Sound | \* |  |
| Battery |  | \* |

**Table3.1.1.1:** Polarity of aspect

### Technical Description:

**3.3.1 Input Dataset:**

Input dataset is a collection of reviews on movies. For example:

##Made after QUARTET was, TRIO continued the quality of the earlier film versions of the short stories by Maugham. Here the three stories are THE VERGER, MR. KNOW-IT-ALL, and SANITORIUM. The first two are comic (THE VERGER is like a prolonged joke, but one with a good pay-off), and the last more serious (as health issues are involved). Again the author introduces the film and the stories.<br /><br />James Hayter, soon to have his signature role as Samuel Pickwick, is the hero in THE VERGER. He holds this small custodial-type job in a church, but the new Vicar (Michael Hordern) is an intellectual snob. When he hears Hayter has no schooling he fires him. Hayter has saved some money, so he tells his wife (Kathleen Harrison) he fancies buying a small news and tobacco shop. He has a good eye, and his store thrives. Soon he has a whole chain of stores. When his grandchild is christened by Hordern, the latter is amazed to see how prosperous his ex-Verger. The payoff is when bank manager Felix Aylmer meets with Hayter about diversifying his investments. I'll leave it to you to hear the unintentional but ironic coda of the meeting.<br /><br />According to Maugham he met a man like Max Kelada (Nigel Patrick) on a cruise. In MR. KNOW-IT-ALL Kelada is a splashy, friendly, and slightly overbearing type from the Middle East who is on a business trip (regarding jewelry) by steamship. His state-room mate is Mr. Grey (the ever quiet and proper Wilfred Hyde-White) who is somewhat, silently disapproving of Max. Max likes to enliven things, and soon is heavily involved in the ship's entertainment. At this point the story actually resembles part of the plot of the non-Maugham story and film CHINA SEAS (1935), as Max makes a bet that he can tell a real piece of jewelry from a fake (after insisting that a piece of jewelry he spotted is real). I won't describe the way Max rises to the occasion.<br /><br />SANITORIUM is the longest segment. Roland Culver plays "Ashenden" (the fictional alter-ego of Maugham - a writer and one time spy as in Hitchcock's THE SECRET AGENT). Here he has to use a sanitorium for a couple of months for his health. He finds a remarkable crew of people, including Jean Simmons as a frail but beautiful young woman, Finlay Currie as an irascible Scotsman, John Laurie as a second irascible Scotsman who is "at war" with Currie, Raymond Huntley as a quiet patient who only shows his internal anger at his situation when his wife shows up, and Michael Rennie as a young man who has a serious life threatening illness. Culver watches as three stories among these characters play out to their conclusions. The last, dealing with Simmons and Rennie, is ironic but deeply moving.<br /><br />It was a dandy follow-up to the earlier QUARTET, and well worth watching.

**3.3.1.1** Input dataset of reviews on movies

Input data set is a collection of reviews.It contains nearly about 50000 reviews.These reviews need to be preprocessed to remove unwanted and noisy data. Preprocessing is the main step in natural language processing. It involves several steps. After preprocessing feature extraction will be done. By using relative importance aspects will be extracted and separated as positive and negative . In training phase we have taken a few aspects. Then based on that we have given another few aspects for testing phase. Then by applying classification algorithms like Naive Bayes Classifier and Logistic Regression an accuracy of classification can be determined.

### 3.2 Data preprocessing:

Data pre-processing is an important step in the data mining process. Data-gathering methods are often loosely controlled, resulting in out-of-range values, impossible data combinations, missing values, etc. Analyzing data that has not been carefully screened for such problems can produce misleading results. Thus, the representation and quality of data is first and foremost before running an analysis.

If there is much irrelevant and redundant information present or noisy and unreliable data, then knowledge discovery during the training phase is more difficult. Data cleaning is the process of detecting and correcting corrupt or inaccurate records from a record set, table, or database and refers to identifying incomplete, incorrect, inaccurate or irrelevant parts of the data and then replacing, modifying, or deleting the dirty or coarse data.

After cleaning, a data set should be consistent with other similar data sets in the system. The inconsistencies detected or removed may have been originally caused by user entry errors, by corruption in transmission or storage, or by different data dictionary definitions of similar entities in different stores. Data cleaning differs from data validation in that validation almost invariably means data is rejected from the system at entry and is performed at the time of entry, rather than on batches of data.

In this project we have taken reviews on movie reviews. These reviews are not cleaned. So preprocessing should be done. The entire project is done in python language with Spyder as IDE by using NLTK package. Natural Language Tool Kit (NLTK) is a comprehensive Python library for natural language processing and text analytics. NLTK is often used for rapid prototyping of text processing programs and can even be used in production applications. In preprocessing the following steps will be followed. Stop word removal, stemming and lemmatizing. Then preprocessed sentences will be produced.

### Stop word removal:

Stop words are common words that generally do not contribute to the meaning of a sentence, at least for the purposes of information retrieval and natural language processing. These are words such as ‘the’ and ‘a’. Most search engines will filter out stop words from search queries and documents in order to save space in their index. NLTK comes with a stop words corpus that contains word lists for many languages.

### Stemming:

Stemming is a technique to remove affixes from a word, ending up with the stem. For example, the stem of cooking is cook, and a good stemming algorithm knows that the “ing” suffix can be removed. Stemming is most commonly used by search engines for indexing words. Instead of storing all forms of a word, a search engine can store only the stems, greatly reducing the size of index while increasing retrieval accuracy. One of the most common stemming algorithms is the Porter stemming algorithm by Martin Porter. It is designed to remove and replace well-known suffixes of English words.

### Lemmatization:

Lemmatization is very similar to stemming, but is more akin to synonym replacement. A lemma is a root word, as opposed to the root stem. So unlike stemming, you are always left with a valid word that means the same thing. However, the word you end up with can be completely different.

### Classification:

A conditional probability is a probability that event X will occur, given the evidence. So, our initial formula looks like this:

P (sentiment | aspect) = P (sentiment) P (aspect |sentiment) / P (sentence)

We can drop the dividing P (line), as it's the same for both classes, and we just want to rank them rather than calculate a precise probability. We can use the independence assumption to let us treat P (aspect | sentiment) as the product of P (token | sentiment) across all the tokens in the sentence. So, we estimate P (token | sentiment) as:

Count (this token in class) + 1 / count (all tokens in class) +count (all tokens)

The extra 1 and count of all tokens is called “add one” or Laplace smoothing and stops a 0 finding its way into the multiplications. If there was not any sentence with an unseen token in, it would score zero.

The classify function starts by calculating the prior probability (the chance of it being one or the other before any tokens are looked at) based on the number of positive and negative examples; in our example, that will always be 0.5, as for each observation (positive/negative aspect update), there are the same amount of data. We then tokenize the incoming document and for each class multiply together the likelihood of each word being seen in that class. We sort the final result and return the highest scoring class.

Our study classifies the polarity of the context/aspect update in a sentence level. Sentence level, in most cases, is more accurate than the phrase level because every status update has its own style in addressing user’s sentiment.

### 3.4.1 Supervised learning:

Supervised learning is the machine learning task of inferring a function from labeled training data. Each example is a pair consisting of an input object and a desired output value. The data are labelled with pre-defined classes. It is like a teacher gives the classes. Test cases are into these classes too. Use training data to infer model. Apply model to test data.

Also, it is used in many situations of natural learning. A set of input and output patterns called a training set is required for this learning mode. Typically, supervised learning rewards accurate classifications or associations and punishes those which yield inaccurate responses.The teacher estimates the negative error gradient direction and reduces the error accordingly.

In many situations, the inputs, outputs and the computed gradient are deterministic, however, the minimization of error proceeds over all its random realizations. As a result, most supervised learning algorithms reduce to stochastic minimization of error in multidimensional weight space.

Supervised learning networks represent the main stream of the development in neural networks. Some examples of well-known pioneering networks include the Perceptron, ADALINE/MADALINE, and various multi layer networks. Two phases are involved in a supervised learning network: retrieving phase and learning phase. Some of the most predominant Supervised Learning techniques in Sentiment Analysis have been SVM, Nave Bayesian Classifiers and other Decision Trees. A Naive Bayes classifier is a simple probabilistic model based on the Bayes rule along with a strong independence assumption. Support Vector Machines has outperformed other classifiers such as Naive Bayes.

### 3.4.1.1 Naive Bayes Classifier:

Bayesian classifiers are based around the Bayes rule, a way of looking at conditional probabilities that allows you to flip the condition around in a convenient way. A conditional probably is a probably that event X will occur, given the evidence Y. That is normally written P(X | Y). The Bayes rule allows us to determine this probability when all we have is the probability of the opposite result and of the two components individually.

P(X | Y) = P(X) P(Y | X) /P(Y).

This restatement can be very helpful when we are trying to estimate the probability of something based on examples of it occurring.

In this case, we are trying to estimate the probability that a document is positive or negative, given its contents. We can restate that, so that is in terms of the probability of that document occurring if it has been predetermined to be positive or negative. This is convenient, because we have examples of positive and negative opinions from our data set above.

The Bayesian Classification represents a supervised learning method as well as a statistical method for classification. Assumes an underlying probabilistic model and it allows us to capture uncertainty about the model in a principled way by determining probabilities of the outcomes. It can solve diagnostic and predictive problems.

Bayesian classification provides practical learning algorithms and prior knowledge and observed data can be combined. Bayesian Classification provides a useful perspective for understanding and evaluating many learning algorithms. It calculates explicit probabilities for hypothesis and it is robust to noise in input data.

The point of view that renders this process a “naive” Bayesian one is that we make a large assumption about how we can calculate the probability of the document occurring; it is equal to the product of the probabilities of each word within its occurrence. This implies that there is no link between one word and another word. Independence assumption it is called.

We can estimate the probability of a word occurring, given a positive or negative sentiment by looking through a series of examples of positive and negative sentiments and counting how often it occurs in each class. This is what makes this supervised learning, the requirement for pre classified examples to train on.

`

|  |
| --- |
| Positive Aspect  Negative Aspect |

Training set

Test set

classifier

Accuracy of classifier

**FIGURE: 3.4.2.1** Main Methodology of Naive-Bayes classifier

Assume independence among attributes Ai when class is given:

* + - * P(A1, A2, …, An |C) = P(A1| Cj) P(A2| Cj)… P(An| Cj)
      * Can estimate P(Ai| Cj) for all Ai and Cj.
      * New point is classified to Cj if P(Cj)  P(Ai| Cj) is maximal.

For continuous attributes:

Discretize the range into bins. One ordinal attribute per bin. Violates independence assumption. Two-way split: (A < v) or (A > v). Choose only one of the two splits as new attribute. Probability density estimation: Assume attribute follows a normal distribution. Use data to estimate parameters of distribution (e.g., mean and standard deviation). Once probability distribution is known, can use it to estimate the conditional probability P (Ai|c)

For discrete attributes:

P(Ai | Ck) = |Aik|/ Nc

where |Aik| is number of instances having attribute Ai and belongs to class C

#### Advantages of the Naive Bayes Classification technique can be summarised as follows:

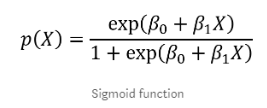
Robust to isolated noise points. Handle missing values by ignoring the instance during probability estimate calculations. Robust to irrelevant attributes. It is fast and space efficient. It is not sensitive to irrelevant features. Independence assumption may not hold for some attributes. Fast to train (single scan). Fast to classify. Not sensitive to irrelevant features. Handles real and discrete data. Handles streaming data well. Use other techniques such as Bayesian Belief Networks (BBN).

#### Disadvantages of the Naive Bayes Classification technique:

Assumes independence of features. It makes a very strong assumption on the shape of your data distribution, i.e. any two features are independent given the output class. Due to this, the result can be (potentially) very bad - hence, a “naive” classifier. This is not as terrible as people generally think, because the NB classifier can be optimal even if the assumption is violated, and its results can be good even in the case of sub-optimality.

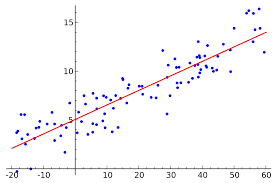
### 3.4.1.1 Logistic Regression:

When it comes to classification, we are determining the probability of an observation to be part of a certain class or not. Therefore, we wish to express the probability with a value between 0 and 1. A probability close to 1 means the observation is very likely to be part of that category. In order to generate values between 0 and 1, we express the probability using this equation:



Logistic regression is the classification counterpart to linear regression. Predictions are mapped to be between 0 and 1 through the [logistic function](https://en.wikipedia.org/wiki/Logistic_function" \t "https://elitedatascience.com/_blank), which means that predictions can be interpreted as class probabilities.

The models themselves are still "linear," so they work well when your classes are [linearly separable](http://www.ece.utep.edu/research/webfuzzy/docs/kk-thesis/kk-thesis-html/node19.html" \t "https://elitedatascience.com/_blank) (i.e. they can be separated by a single decision surface). Logistic regression can also be regularized by penalizing coefficients with a tunable penalty strength.



**FIGURE 3.4.3.1** Representation of logistic regression classifier

#### The advantages of the Logistic Regression technique can be summarized as follows:

* Outputs have a nice probabilistic interpretation, and the algorithm can be regularized to avoid over fitting.
* Logistic models can be updated easily with new data using stochastic gradient descent.
* It is more robust: the independent variables don’t have to be normally distributed, or have equal variance in each group.
* It may handle nonlinear effects
* You can add explicit interaction and power terms
* There is no homogeneity of variance assumption.
* Normally distributed error terms are not assumed.
* It does not require that the independents be an interval.
* It does not require that the independents be unbounded.

**The disadvantages of the Logistic Regression technique can be summarized as follows:**

* + - * Logistic regression tends to underperform when there are multiple or non-linear decision boundaries.
      * They are not flexible enough to naturally capture more complex relationships.
      * Logistic regression attempts to predict outcomes based on a set of independent variables, but logic models are vulnerable to overconfidence.

Logistic regression requires that each data point be independent of all other data points. If observations are related to one another, then the model will tend to overweight the significance of those observations. This is a major disadvantage, because a lot of scientific and social-scientific research relies on research techniques involving multiple observations of the same individuals.

Limited Outcome Variables.

### Project Description:

* + 1. **Overview of the project:**

**Data Collection**

**Data Preprocessing**

**Negative**

**Apply algorithm for analysis**

**Positive**

**Aspect based sentimental analysis**

**Opinion words**

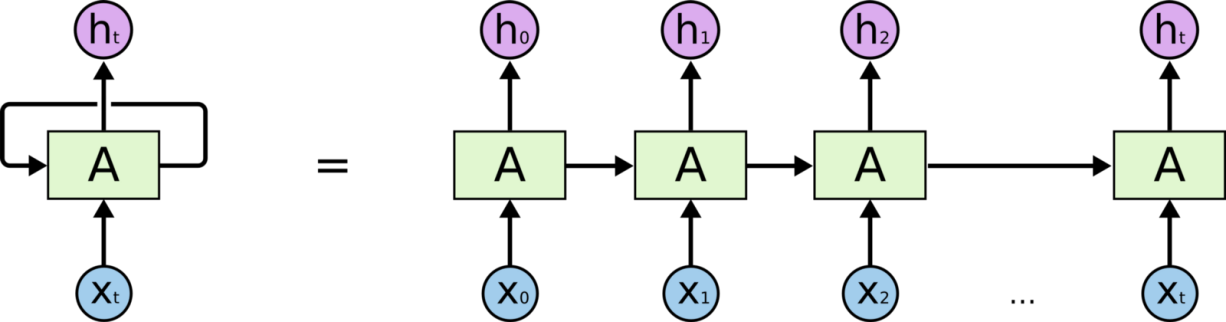
**Analysis of Result**

**FIGURE 3.5.1.1** Flow Diagram for the Project

**3.5.2 Recurrent Neural Networks**

A **recurrent neural network** ( **RNN**) is a class of artificial neural network where connections between nodes form a directed graph along a temporal sequence. This allows it to exhibit temporal dynamic behavior. Unlike feedforward neural networks, RNNs can use their internal state (memory) to process sequences of inputs. Recurrent Neural Networks (RNN) are a powerful and robust type of neural networks and belong to the most promising algorithms out there at the moment because they are the only ones with an internal memory.Because of their internal memory, RNN’s are able to remember important things about the input they received, which enables them to be very precise in predicting what’s coming next. Artificial neural networks are created with interconnected data processing components that are loosely designed to function like the human brain. They are composed of layers of artificial neurons (network nodes) that have the capability to process input and forward output to other nodes in the network. The nodes are connected by edges or weights that influence a signal's strength and the network's ultimate output.

In some cases, artificial neural networks process information in a single direction from input to output. These "feedforward" neural networks include convolutional neural networks that underpin image recognition systems . RNNs, on the other hand, can be layered to process information in two directions. Like feedforward neural networks, RNNs can process data from initial input to final output. Unlike feed forward neural networks, RNNs use feedback loops such as Backpropagation Through Time or BPTT throughout the computational process to loop information back into the network. This connects inputs together and is what enables RNNs to process sequential and temporal data.



**Fig 3.5.2.1.**Recurrent Neural Network Classification

**The advantages of the RNN technique can be summarized as follows:**

1. Regardless of the sequence length, the learned model always has the same input size, because it is specified in terms of transition from one state to another state, rather then specified in terms of variable-length history of states.

2. It is possible to use same transition function f with same parameters at every time step.

**Disadvantages of RNN:**

1. The first disadvantage of RNN is the gradient vanishing and exploding problems. It makes the training of RNN difficult, in two ways:

a. it cannot process very long sequences if using tanh as its activation function,

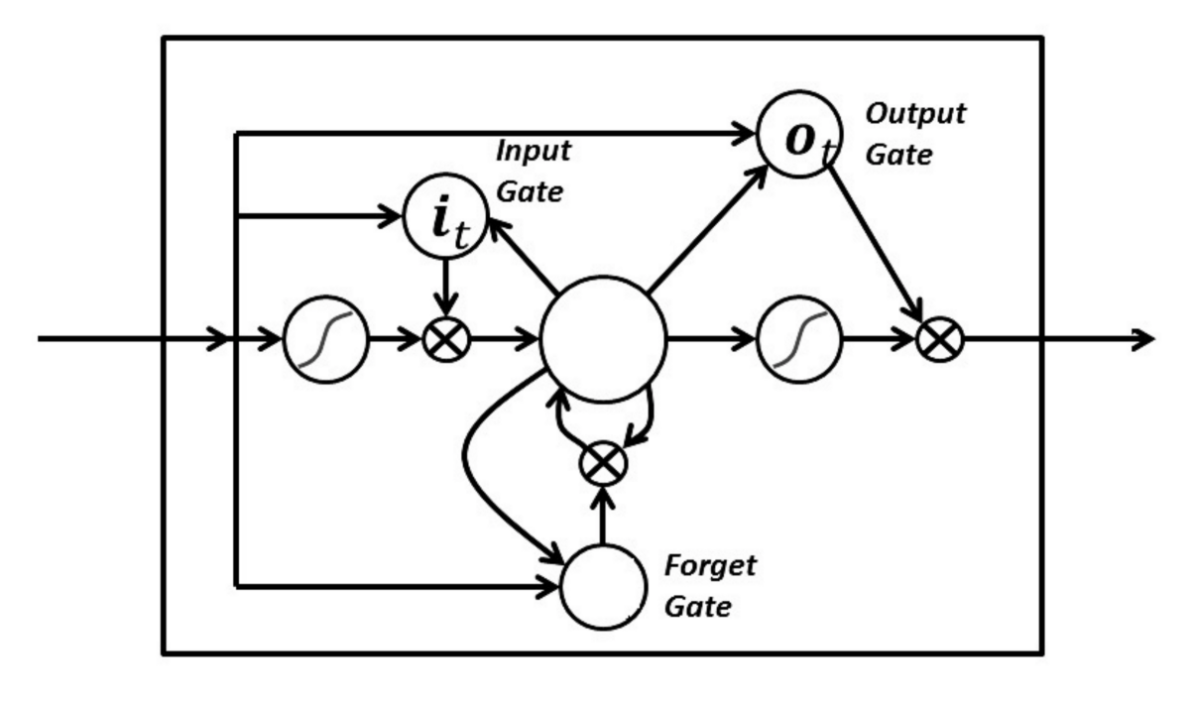
b. it is very unstable if using relu as its activation function.

2. Second, it cannot be stacked into very deep models. This is mostly because of the saturated activation function used in RNN models, making the gradient decay over layers.

**Vanishing Gradients:**W e speak of “Vanishing Gradients“ when the values of a gradient are too small and the model stops learning or takes way too long because of that. This was a major problem in the 1990s and much harder to solve than the exploding gradients. Fortunately, it was solved through the concept of LSTM by Sepp Hochreiter and Juergen Schmidhuber.

**3.5.1.1 .LONG-SHORT TERM MEMORY:**

One drawback to standard RNNs is the vanishing gradient problem, in which performance of the neural network suffers because it can't be trained properly. This happens with deeply layered neural networks, which are used to process complex data. Standard RNNs that use a gradient-based learning method degrade the bigger and more complex they get. Tuning the parameters effectively at the earliest layers becomes too time consuming and computationally expensive. One solution to the problem is called Long Short-Term Memory (LSTM) units, which were invented by computer scientists Sepp Hochreiter and Jurgen Schmidhuber in 1997. Long Short-Term Memory (LSTM) networks are an extension for recurrent neural networks, which basically extends their memory. Therefore it is well suited to learn from important experiences that have very long time lags in between.LSTM’s enable RNN’s to remember their inputs over a long period of time. This is because LSTM’s contain their information in a memory, that is much like the memory of a computer because the LSTM can read, write and delete information from its memory.



In an LSTM you have three gates: input, forget and output gate. These gates determine whether or not to let new input in (input gate), delete the information because it isn’t important (forget gate) or to let it impact the output at the current time step (output gate).

The problematic issues of vanishing gradients is solved through LSTM because it keeps the gradients steep enough and therefore the training relatively short and the accuracy high.

**Advantages of LSTM:**

Neural Networks is a very powerful technique and is used for image recognition and many other applications. One of the limitation is that, there is no memory associated with the model. Which is a problem for sequential data, like text or time series. RNN addresses that issue by including a feedback look which serves as a kind of memory. So the past inputs to the model leave a footprint. LSTM extends that idea and by creating both a short-term and a long-term memory component.

### System Requirements:

* + - 1. **Hardware Requirements:**

|  |  |
| --- | --- |
| * PROCESSOR | : I5 for machine learning models, GPU for deep learning models. |
| * RAM | : At least 8 GB. |
| * HARD DISK | :700 MB for a typical installation of SPYDER |
| * KEY BOARD | : Multimedia Keyboard |

* + - 1. **Software Requirements:**
         * OPERATING SYSTEM : Windows XP or higher
         * FRONT END : SPYDER
         * ONLINE PLATFORMS :GOOGLE COLAB

### Technical Requirements:

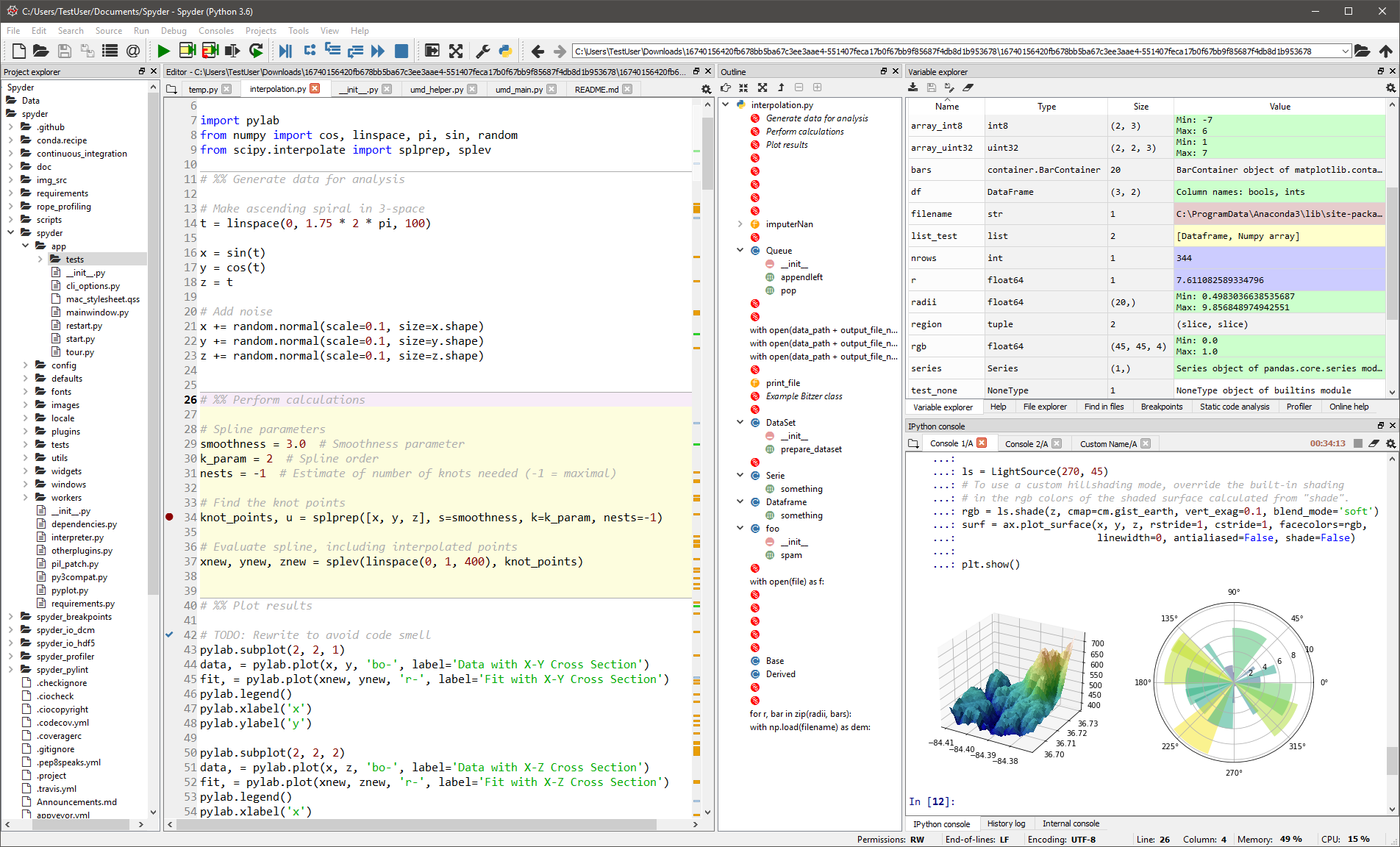
* + - 1. **Choice of language:**

Python is a general-purpose interpreted, interactive, object-oriented, and high-level programming language. It was created by Guido van Rossum during 1985- 1990. Like Perl, Python source code is also available under the GNU General Public License (GPL). Python is designed to be highly readable. It uses English keywords frequently where as other languages use punctuation, and it has fewer syntactical constructions than other languages.

* **Python is Interpreted :** Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.
* **Python is Interactive:** You can actually sit at a Python prompt and interact with the interpreter directly to write your programs.
* **Python is Object-Oriented** : Python supports Object-Oriented style or technique of programming that encapsulates code within objects.
* **Python is a Beginner's Language** : Python is a great language for the beginner-level programmers and supports the development of a wide range of applications from simple text processing to WWW browsers to games.

### Choice of platform:

Spyder is a powerful scientific environment written in Python, for Python, and designed by and for scientists, engineers and data analysts. It features a unique combination of the advanced editing, analysis, debugging, and profiling functionality of a comprehensive development tool with the data exploration, interactive execution, deep inspection, and beautiful visualization capabilities of a scientific package. Furthermore, Spyder offers built-in integration with many popular scientific packages, including [NumPy](https://www.numpy.org/), [SciPy](https://www.scipy.org/),  [Pandas](https://pandas.pydata.org/), [IPython](https://ipython.org/), [QtConsole](https://qtconsole.readthedocs.io/en/stable/) , [Matplotlib](https://matplotlib.org/), [SymPy](https://www.sympy.org/en/index.html), and more . Beyond its many built-in features, Spyder’s abilities can be extended even further via its plugin system and API. Spyder can also be used as a PyQt5 extension library, allowing you to build upon its functionality and embed its components, such as the interactive console, in your own software.



**FIGURE 3.5.3.2.1** Spyder GUI Main page

### 3.5.4 Processing steps:

1. Preparing the data set on Movie Reviews.
2. Preprocessing the data set.
3. Train the data set using classification algorithm.
4. Test the model on test data set.
5. Calculate accuracy using performance metrics.

# CHAPTER-4 RESULTS AND DISCUSSIONS

## RESULTS AND DISCUSSIONS

### Output screens:

### Performance metrics for classification algorithms:

Analysis is generally measured using following parameters.

**True Positives(TP):** These are the correctly predicted positive values which means that the value of actual class is yes and the value of predicted class is also yes.

**True Negatives(TN):** These are the correctly predicted negative values which means that the value of actual class is no and the value of predicted class is also no.

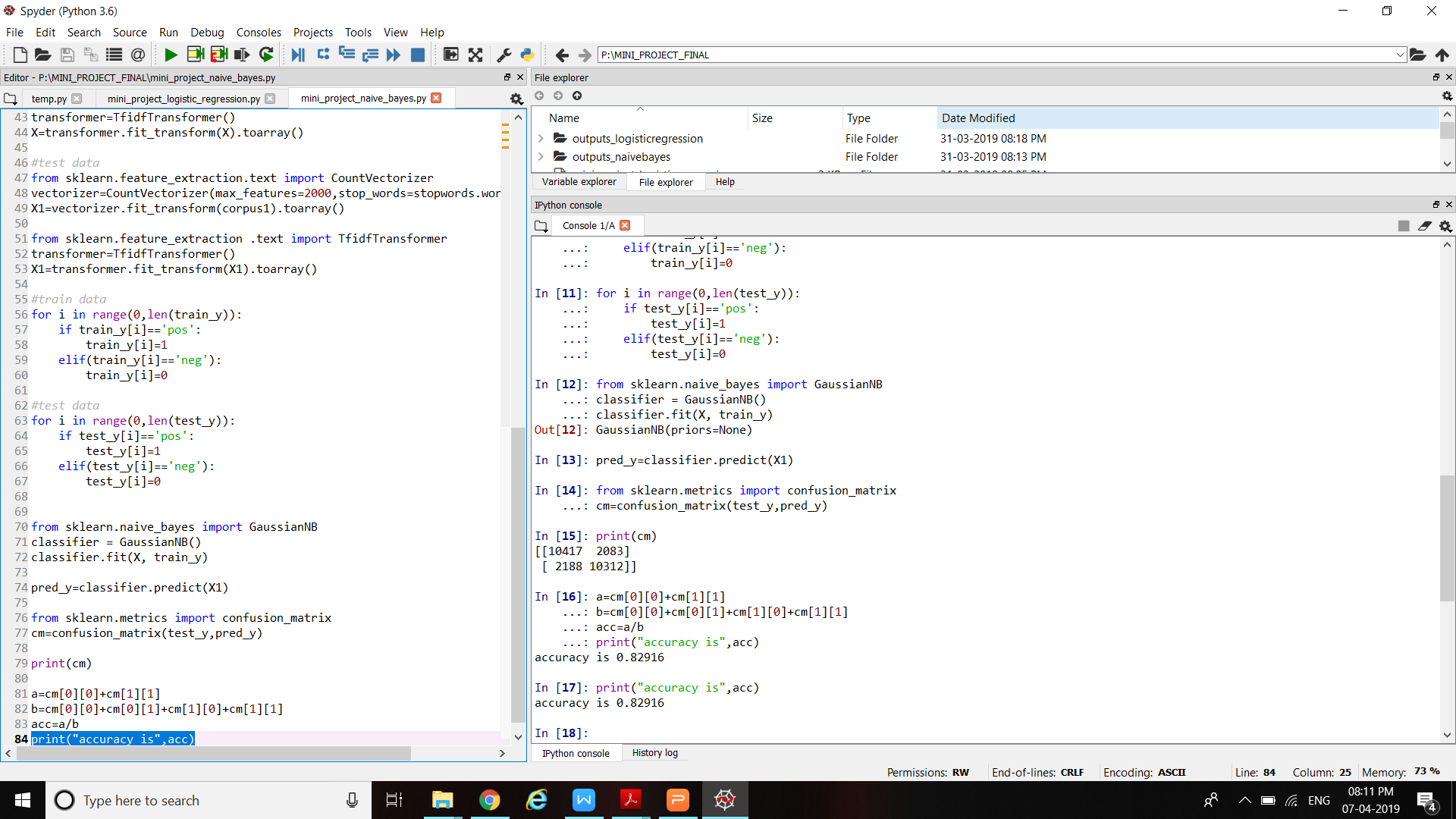
**False Positives(FP):** When actual class is no predicted class is yes.

**False Negatives(FN):** When actual class is yes predicted class is no.

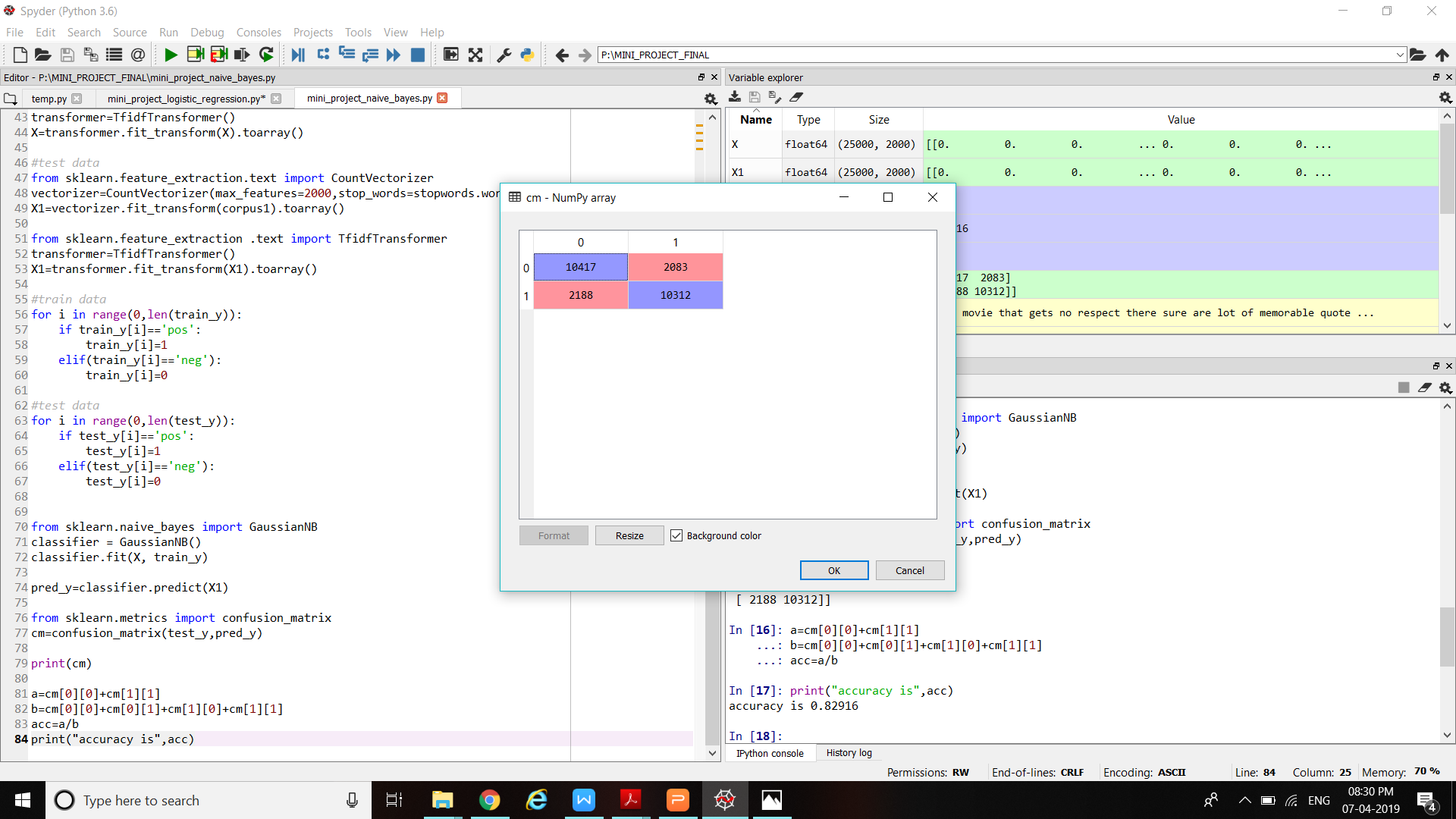
**Accuracy-**Accuracy is the most intuitive performance measure and it is simply a ratio of correctly predicted observation to the total observations.

#### Accuracy= TP+TN/TP+FN+TN+FN

* + 1. **Output analysis of Naive Bayes classifier:**

****

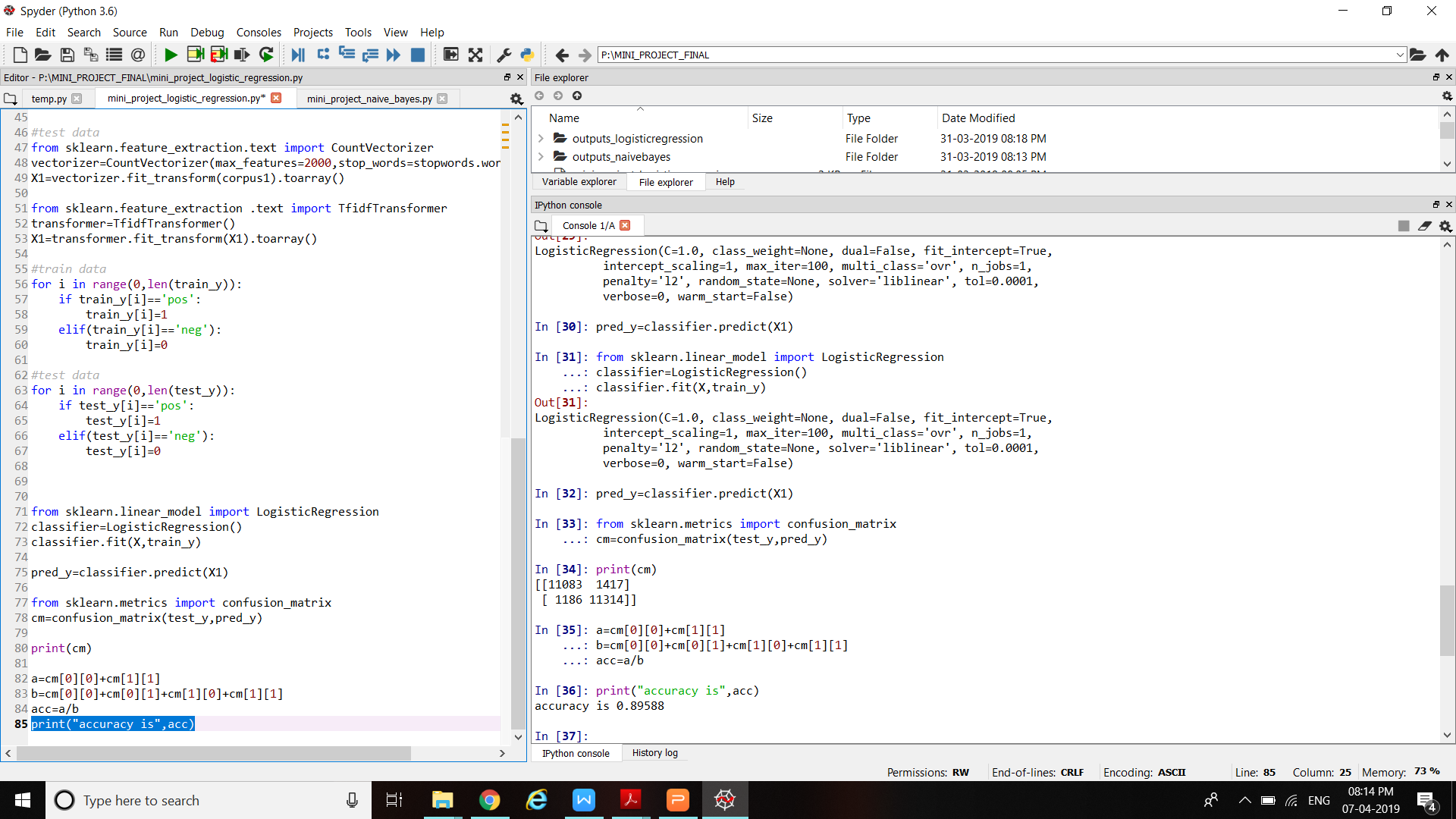
**FIGURE 4.2.1.1** Output of Naive-Bayes classifier



**TABLE 4.2.1.2** Confusion Matrix

### Percentage of Accuracy: 82.91

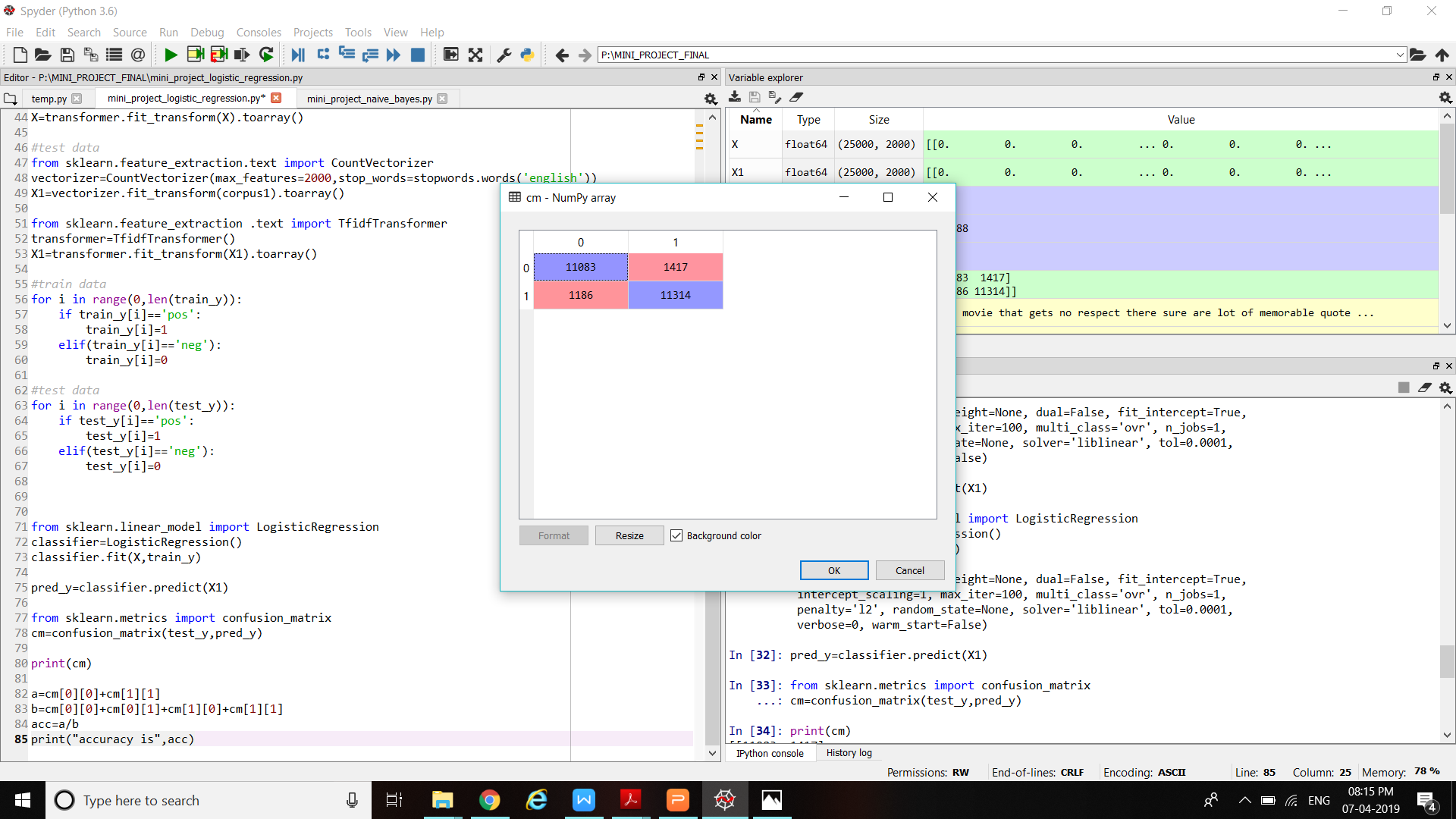
* + 1. **Output analysis of Logistic Regression:**

****

**FIGURE 4.2.2.1** Output of Logistic Regression

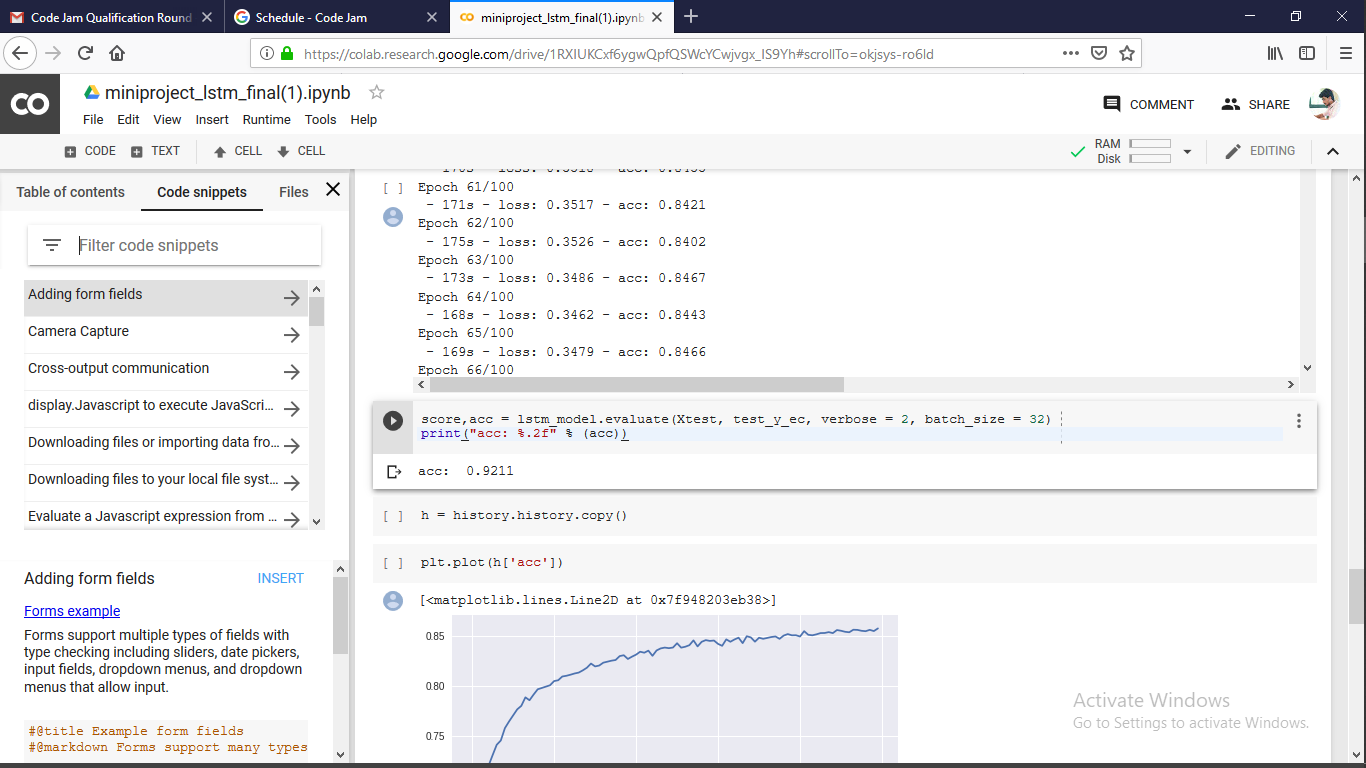
### Percentage of Accuracy:89.58

**TABLE 4.2.2.1** Performance Metrics



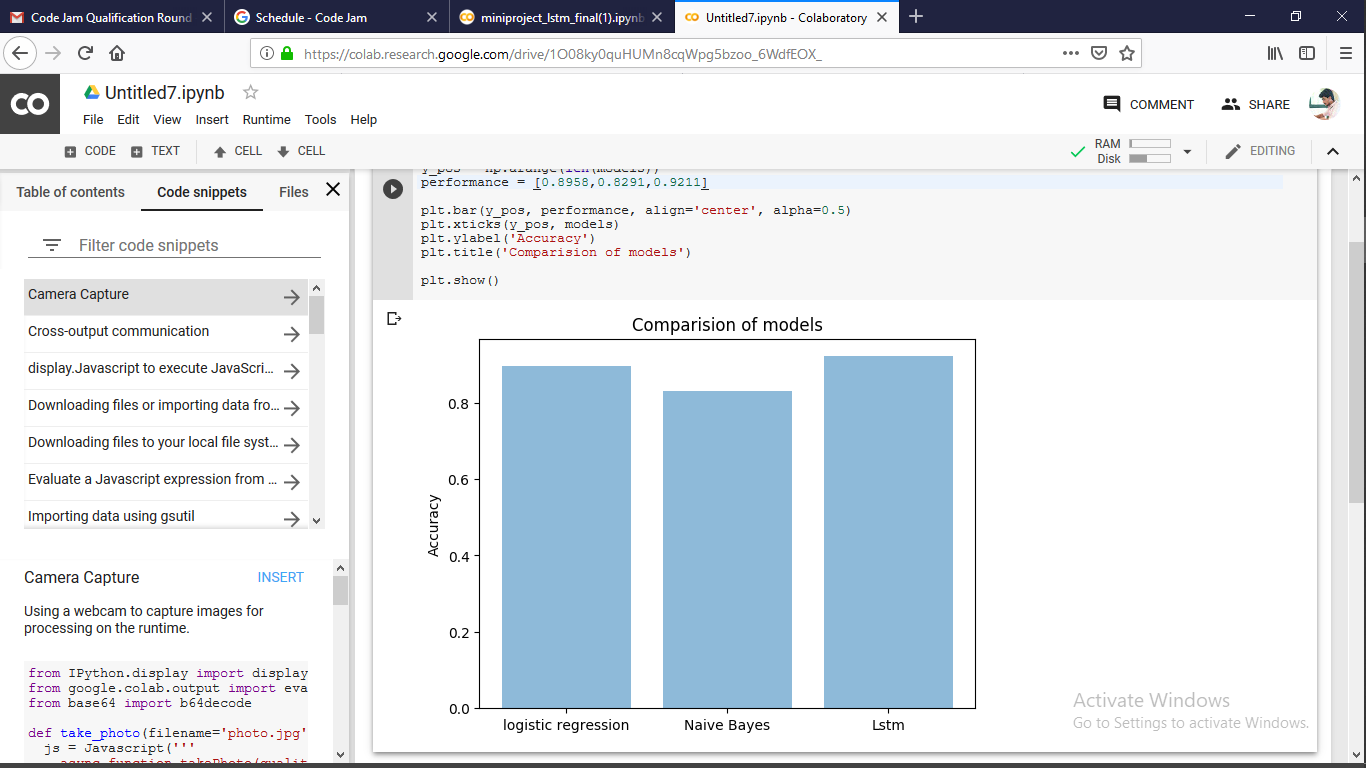
**TABLE 4.2.2.2** Confusion matrix

* + 1. **Output analysis of LSTM:**



**Fig 4.2.3.1**.Output analysis of LSTM

### Comparison of results:

****

**FIGURE 4.3.1.1** Comparison of the models

### Analysis of results:

|  |  |  |  |
| --- | --- | --- | --- |
| Measure | Naive Bayes | Logistic  Regression | LSTM |
| Accuracy | 82.58 | 89.58 | 92.11 |

**TABLE 4.3.4.1** Analysis of results

**CHAPTER-5 CONCLUSION AND FUTURE SCOPE**

## CONCLUSION AND FUTURE WORK

### Conclusion:

The reviews are on the movies. These reviews contained noisy and some unwanted data. Those unwanted data is removed during preprocessing so that training the classifier will be easy. By using the threshold value, classification of the aspects into positive and negative is done. On the extracted aspects Naive-bayes and Logistic Regression and LSTM are applied. These classifiers are applied to find out the correct classification of aspects. Finally experimental results are presented. These experimental results shows that accuracy of LSTM is greater than the accuracy of Logistic Regression and Naive-Bayes algorithm.

### Future Scope:

In future, it will be proposed to work on strong positive and strong negative aspect terms .This will be implemented by using the Fuzzy concepts. By using the Fuzzy logic, the results will be better and accurate when compared to existing methods.

# CHAPTER-6 REFERENCES

## REFERENCES

### 6.1 References:

### B. Pang, L. Lee, and S. Vaithyanathan, “Thumbs up?: sentiment classification using machine learning techniques,” in Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume Association for Computational Linguistics, 2002, pp. 79–86.?

### Sentiment Analysis of Review Datasets using Naive Bayes’ classifier LopamudraDey, Sanjay Chakraborty.

### Jingbo Zhu, Huizhen Wang, Muhua Zhu, Benjamin K. Tsou, Matthew Ma “Aspect-based opinion polling from customer reviews” IEEE Transactions on affective computing, Vol. 2, No.1, January-March 2011

### V. Peddinti and P. Chintalapoodi,V.M.Kiran, in sentiment analysis of movie reviews ', Analyzing Microtext Workshop, AAAI, 2011

### Minhoe Hur Seoul National University “Box-office forecasting based on sentiments of movie reviews and Independent subspace method”, Information Sciences,2016?

# [Blety Babu Alengadan](https://ieeexplore.ieee.org/author/37086242523); [Shamsuddin S. Khan](https://ieeexplore.ieee.org/author/37086039557)["A proposed system for modifying aspect based opinion mining for ranking of products".Third International Conference on Sensing, Signal Processing and Security (ICSSS)](https://ieeexplore.ieee.org/xpl/mostRecentIssue.jsp?punumber=8063614) 2017.

# “[Text sentiment analysis based on long short-term memory](https://ieeexplore.ieee.org/document/7778967/)”[Dan Li](https://ieeexplore.ieee.org/author/37086097311); [Jiang Qian](https://ieeexplore.ieee.org/author/37085455584) [2016 First IEEE International Conference on Computer Communication and the Internet ICCCI)](https://ieeexplore.ieee.org/xpl/mostRecentIssue.jsp?punumber=7757193) 2016.

# Sentiment Analysis of Review Datasets using Naive Bayes’ classifier LopamudraDey, Sanjay Chakraborty.

# “Aspect-based sentiment analysis of movie reviews “ Tun Thura Thet ,Jin-Cheon Na and Christopher S.G. Khoo.

# APPENDICS

### Naive\_bayes.py:

### import pandas as pd

### import numpy as np

### import nltk

### import string

### import matplotlib.pyplot as plt

### import re

### from nltk.corpus import stopwords

### def load\_data(path):

### data = pd.read\_csv(path)

### x = data['reviewText'].tolist()

### y = data['sentiment'].tolist()

### return x, y

### #train and test

### train\_x, train\_y = load\_data('train.csv')

### test\_x, test\_y = load\_data('test.csv')

### #train data

### corpus=[]

### for i in range(0,len(train\_x)):

### review=re.sub(r'\W',' ',str(train\_x[i]))

### review=review.lower()

### review=re.sub(r'\s+[a-z]\s+',' ',review)

### review=re.sub(r'^[a-z]\s+',' ',review)

### review=re.sub(r'\s+',' ',review)

### corpus.append(review)

### #test data

### corpus1=[]

### for i in range(0,len(test\_x)):

### review=re.sub(r'\W',' ',str(train\_x[i]))

### review=review.lower()

### review=re.sub(r'\s+[a-z]\s+',' ',review)

### review=re.sub(r'^[a-z]\s+',' ',review)

### review=re.sub(r'\s+',' ',review)

### corpus1.append(review)

### 

### #train data

### from sklearn.feature\_extraction.text import CountVectorizer

### vectorizer=CountVectorizer(max\_features=2000,stop\_words=stopwords.words('english'))

### X=vectorizer.fit\_transform(corpus).toarray()

### from sklearn.feature\_extraction .text import TfidfTransformer

### transformer=TfidfTransformer()

### X=transformer.fit\_transform(X).toarray()

### #test data

### from sklearn.feature\_extraction.text import CountVectorizer

### vectorizer=CountVectorizer(max\_features=2000,stop\_words=stopwords.words('english'))

### X1=vectorizer.fit\_transform(corpus1).toarray()

### from sklearn.feature\_extraction .text import TfidfTransformer

### transformer=TfidfTransformer()

### X1=transformer.fit\_transform(X1).toarray()

### #train data

### for i in range(0,len(train\_y)):

### if train\_y[i]=='pos':

### train\_y[i]=1

### elif(train\_y[i]=='neg'):

### train\_y[i]=0

### 

### #test data

### for i in range(0,len(test\_y)):

### if test\_y[i]=='pos':

### test\_y[i]=1

### elif(test\_y[i]=='neg'):

### test\_y[i]=0

### 

### 

### from sklearn.naive\_bayes import GaussianNB

### classifier = GaussianNB()

### classifier.fit(X, train\_y)

### 

### pred\_y=classifier.predict(X1)

### from sklearn.metrics import confusion\_matrix

### cm=confusion\_matrix(test\_y,pred\_y)

### print(cm)

### a=cm[0][0]+cm[1][1]

### b=cm[0][0]+cm[0][1]+cm[1][0]+cm[1][1]

### acc=a/b

### print("accuracy is",acc)

### Logistic\_Regression.py

import pandas as pd

import numpy as np

import nltk

import string

import matplotlib.pyplot as plt

import re

from nltk.corpus import stopwords

def load\_data(path):

data = pd.read\_csv(path)

x = data['reviewText'].tolist()

y = data['sentiment'].tolist()

return x, y

#train and test

train\_x, train\_y = load\_data('train.csv')

test\_x, test\_y = load\_data('test.csv')

#train data

corpus=[]

for i in range(0,len(train\_x)):

review=re.sub(r'\W',' ',str(train\_x[i]))

review=review.lower()

review=re.sub(r'\s+[a-z]\s+',' ',review)

review=re.sub(r'^[a-z]\s+',' ',review)

review=re.sub(r'\s+',' ',review)

corpus.append(review)

#test data

corpus1=[]

for i in range(0,len(test\_x)):

review=re.sub(r'\W',' ',str(train\_x[i]))

review=review.lower()

review=re.sub(r'\s+[a-z]\s+',' ',review)

review=re.sub(r'^[a-z]\s+',' ',review)

review=re.sub(r'\s+',' ',review)

corpus1.append(review)

#train data

from sklearn.feature\_extraction.text import CountVectorizer

vectorizer=CountVectorizer(max\_features=2000,stop\_words=stopwords.words('english'))

X=vectorizer.fit\_transform(corpus).toarray()

from sklearn.feature\_extraction .text import TfidfTransformer

transformer=TfidfTransformer()

X=transformer.fit\_transform(X).toarray()

#test data

from sklearn.feature\_extraction.text import CountVectorizer

vectorizer=CountVectorizer(max\_features=2000,stop\_words=stopwords.words('english'))

X1=vectorizer.fit\_transform(corpus1).toarray()

from sklearn.feature\_extraction .text import TfidfTransformer

transformer=TfidfTransformer()

X1=transformer.fit\_transform(X1).toarray()

#train data

for i in range(0,len(train\_y)):

if train\_y[i]=='pos':

train\_y[i]=1

elif(train\_y[i]=='neg'):

train\_y[i]=0

#test data

for i in range(0,len(test\_y)):

if test\_y[i]=='pos':

test\_y[i]=1

elif(test\_y[i]=='neg'):

test\_y[i]=0

from sklearn.linear\_model import LogisticRegression

classifier=LogisticRegression()

classifier.fit(X,train\_y)

pred\_y=classifier.predict(X1)

from sklearn.metrics import confusion\_matrix

cm=confusion\_matrix(test\_y,pred\_y)

print(cm)

a=cm[0][0]+cm[1][1]

b=cm[0][0]+cm[0][1]+cm[1][0]+cm[1][1]

acc=a/b

print("accuracy is",acc)

**LSTM.py**

import pandas as pd

import numpy as np

import nltk

nltk.download("stopwords")

import string

import matplotlib.pyplot as plt

from keras.models import Sequential

from keras.layers import Dense

from keras.layers import Flatten

from keras.layers import Embedding

from keras.layers.convolutional import Conv1D

from keras.layers.convolutional import MaxPooling1D

import time

start = time.time()

def load\_data(path):

data = pd.read\_csv(path)

x = data['reviewText'].tolist()

y = data['sentiment'].tolist()

return x, y

lemmatizer = nltk.WordNetLemmatizer()

stopwords = nltk.corpus.stopwords.words('english')

transtbl = str.maketrans(string.punctuation, ' ' \* len(string.punctuation))

def preprocessing(line):

line = line.replace('<br />', '')

line = line.translate(transtbl)

tokens = []

for t in nltk.word\_tokenize(line):

t = t.lower()

if t not in stopwords:

lemma = lemmatizer.lemmatize(t, 'v')

tokens.append(lemma)

return ' '.join(tokens)

import numpy as np

from keras.models import Model

from keras.layers import Dense, Input, Dropout, LSTM, Activation, Embedding

from keras.preprocessing import sequence

np.random.seed(1)

from google.colab import files

uploaded = files.upload()

for fn in uploaded.keys():

print('User uploaded file "{name}" with length {length} bytes'.format(

name=fn, length=len(uploaded[fn])))

def read\_glove\_vec(glove\_file):

with open(glove\_file, 'r', encoding='utf-8', errors='ignore') as f:

words = set()

word\_to\_vec\_map = {}

for line in f:

line = line.strip().split()

curr\_word = line[0]

words.add(curr\_word)

word\_to\_vec\_map[curr\_word] = np.array(line[1:], dtype=np.float64)

i = 1

words\_to\_index = {}

for w in sorted(words):

words\_to\_index[w] = i

i = i + 1

return words\_to\_index, word\_to\_vec\_map

word\_to\_index, word\_to\_vec\_map = read\_glove\_vec('glove.6B.50d.txt')

from keras.preprocessing import sequence

nltk.download('punkt')

nltk.download('wordnet')

train\_x, train\_y = load\_data('train.csv')

test\_x, test\_y = load\_data('test.csv')

train\_x = [preprocessing(x) for x in train\_x]

test\_x = [preprocessing(x) for x in test\_x]

train\_y\_ec = np.array([0 if x == 'neg' else 1 for x in train\_y])

test\_y\_ec = np.array([0 if x == 'neg' else 1 for x in test\_y])

print(train\_y\_ec)

print(test\_y\_ec)

from keras.preprocessing.text import Tokenizer

tokenizer = Tokenizer()

tokenizer.fit\_on\_texts(train\_x)

train\_x\_ec = tokenizer.texts\_to\_sequences(train\_x)

test\_x\_ec = tokenizer.texts\_to\_sequences(test\_x)

from keras.preprocessing.sequence import pad\_sequences

Xtrain = pad\_sequences(train\_x\_ec, maxlen=100, padding='post')

Xtest = pad\_sequences(test\_x\_ec, maxlen=100, padding='post')

vocab\_size = len(tokenizer.word\_index) + 1

def load\_embedding(filename):

file = open(filename,'r',encoding='utf-8', errors='ignore')

lines = file.readlines()

file.close()

embedding = dict()

for line in lines:

parts = line.split()

embedding[parts[0]] = np.asarray(parts[1:], dtype='float32')

return embedding

def get\_weight\_matrix(embedding, vocab):

vocab\_size = len(vocab) + 1

weight\_matrix = np.zeros((vocab\_size, 50))

for word, i in vocab.items():

vector = embedding.get(word)

if vector is not None:

weight\_matrix[i] = vector

return weight\_matrix

from keras.layers import Embedding, SpatialDropout1D

raw\_embedding = load\_embedding('glove.6B.50d.txt')

embedding\_vectors = get\_weight\_matrix(raw\_embedding, tokenizer.word\_index)

embedding\_layer = Embedding(vocab\_size, 50, weights=[embedding\_vectors], input\_length=100, trainable=False)

lstm\_model = Sequential()

lstm\_model.add(embedding\_layer)

lstm\_model.add(SpatialDropout1D(0.4))

lstm\_model.add(LSTM(128, dropout=0.2, recurrent\_dropout=0.2))

lstm\_model.add(Dense(20, activation='tanh'))

lstm\_model.add(Dense(1, activation='sigmoid'))

lstm\_model.compile(loss='binary\_crossentropy',optimizer='adam',metrics=['accuracy'])

end = time.time()

print("Neural Network model trained in %f seconds" % (end-start))

start = time.time()

history = lstm\_model.fit(Xtrain, train\_y\_ec, batch\_size=32, epochs = 300, verbose = 2)

end = time.time()

print("LSTM model trained in %f seconds" % (end-start))

score,acc = lstm\_model.evaluate(Xtest, test\_y\_ec, verbose = 2, batch\_size = 32)

print("acc: %.2f" % (acc))

h = history.history.copy()

plt.plot(h['acc'])

plt.plot(h['loss'])

sample = ["this movie is worst"]

sample\_ec = tokenizer.texts\_to\_sequences(sample)

sample\_ec\_pad = pad\_sequences(sample\_ec, maxlen=100, padding='post')

sample\_prediction = lstm\_model.predict(sample\_ec\_pad)

result = "neg" if sample\_prediction < 0.5 else "pos"

print ("the prediction result on the sample comment is: " + result)

import matplotlib.pyplot as plt; plt.rcdefaults()

import numpy as np

import matplotlib.pyplot as plt

models = ('logistic regression', 'Naive Bayes', 'Lstm')

y\_pos = np.arange(len(models))

performance = [0.8958,0.8291,0.9211]

plt.bar(y\_pos, performance, align='center', alpha=0.5)

plt.xticks(y\_pos, models)

plt.ylabel('Accuracy')

plt.title('Comparision of models')

plt.show()