Original Article

Machine Learning based Sentiment Analysis Using Django

Dr.R.Yamini[1]

[1]Assistant Professor, Department of Computing Technology, SRMIST, Kattankulathur.

Abstract

Before purchasing a product or service, People typically analyse the information about the product such as cost, warranty, quality etc. Only after getting satisfaction about such things, People try to buy that product based on the quality of service received. Since this process takes time and a chance of being duped by the dealer are higher, Sentiment analysis (SA) is necessary to purchase a product without any hesitation. Sentiment analysis examines reviews and comments of the products, which are in the form of text that requires several processes for providing the desirable information to the People. Moreover, SA is a significant research direction of Natural Language Processing (NLP). In this paper, a novel sentiment analysis model is developed based on the Machine Learning (ML) Algorithm, which provides an accurate sentiment information for the texts having different perspectives. The method of Stop words are used for data pre-processing. By using count vectorizer, the text data is converted into the form of vectors for extracting the desired features. Finally, the type of sentiment whether it is positive, negative or neutral is determined based on the ML classifier namely, Naive Bayes classifier. This model is developed using Django web framework that provides an accurate sentiment classification to the people or the industries who need the sentiment analysis.

Keywords: Sentiment Analysis (SA), Stop Words, Count Vectorizer, Naive Bayes.

Received date: 18 August 2022 **Accepted: 20 September, 2022** **Published: 07 October, 2022**

DOI: 10.47750/pnr.2022.13.04.047

INTRODUCTION

In recent days, Internet is becoming a significant platform for the people to express their thoughts and opinions and to acquire the knowledge about the products or service that they want since a vast range of text data has stored on the Internet. From this kind of text data, the sentiments or opinions are extracted using various technologies such as Natural Language Processing (NLP) and big data that help people to make better decisions [1, 2].

Sentiment analysis is the process that analyses the meaning of the review or comment and shows that whether the comment is positive, negative or neutral. In the case of hidden sentence in the text, NLP is used, which models and extract the reason for classifying the text. For example, in the sentence, “The cost of swallow nest is too much but it is so delicious”, the polarity of the sentiment based on service is negative but that based on food is positive. In such sentence, the SA fails to predict the decision instead of making the exact decision. The major aim of the SA is to determine the attitudes of bipolar comments or reviews on particular targets in a sentence [3].

The conventional approaches for SA fail to provide the exact polarity of the text. Several companies want to know about the sentiments of their products or service in distinct aspects. This encourages the aspect level sentiment analysis, which analyse variety of sentiments in different aspects. In general, The SA is carried out on several ways that include word level, document level, sentence level and aspect level. Since the big companies and industries prefer the aspect level SA, it is widely used nowadays. The applications of SA are Social media monitoring, Stock Market Prediction, Decision Making, Reshaping Business and Control Public Sentiment, Movie Success and Box-office Revenue Electoral Predictions, etc [4,5].

SA is primarily focused on analysing reviews, comments of various persons, and processing them in order to extract any useful information. Different elements influence the SA process and must be appropriately managed in order to obtain the final classification report. A few of these issues are covered as described below [6].

The first problem is Co-reference Resolution. This problem occurs when there is a confusion about the review containing two correlated words. In the sentence, “After watching the

movie, we went out for dinner; it was amazing", a confusion raises that whether the sentence is about the movie or food. This type of problem is most common in aspect-oriented SA. The second problem is Association with a period. In the case of SA, the time of review or opinion collection is critical. At the same time, a single user or group of users may offer a favourable response to a product, and there may be a scenario where they give a negative answer. As a result, the sentiment analyser faces a challenge at some point in the future. This type of problem is most common in comparative SA [7]. The third problem, Sarcasm Handling is the most important problem which needs much more attention. Sarcasm is the use of words that have the opposite meaning than the information they convey. In the sentence, "What an excellent student he is, he gets the marks of zero in all the monthly tests". The positive word "excellent " has a negative connotation in this example. This type of sentence causes impact on SA [8]. The fourth problem is Negations, which means the negative words in a text that completely alter the meaning of the sentence based on the way it appears. For instance, the statements "This is a good article." and "This is not a good article." have opposite meanings, however the results may change if the analysis is done one word at a time. N-gram analysis is the primary method for dealing with these types of issues. The last problem is Spam Detection, which is about whether the comment or review is written by either a genuine person or a fraudulent. Many people who have no expertise of the company's product or service submit a good or negative assessment of the service. It is extremely difficult to determine which reviews are genuine and which are not; this ultimately plays a crucial role in SA [9].

In order to overcome these challenges in SA, a novel SA application is proposed in this work that includes the processes of data pre-processing, feature extraction and sentiment classification. At first, Stop words are used for data pre-processing, which remove the words that are not desired for sentiment classification. After removing the unwanted words, the remaining text is converted into the form of vector using Count Vectorizer. This process extracts the features from the text. Finally, the ML classifier decides that whether the input text is positive, negative or neutral. In this work, Naive Bayes classification algorithm is used as the ML classifier that provides the polarity of the sentiment.

State Of The Art

Alattar.F et al. [10] have proposed a Filtered-LDA (Latent Dirichlet Allocation) framework for interpreting variations in the sentiment collected from Twitter. This framework has utilized many series connected LDA models having multiple sets of hyper parameters for capturing the reasons of the candidate that cause sentiment variations. By using Topic Model, the tweets that discussed old topics are removed. However, this framework fails to support the SA for Arabic tweets as well as other languages. Fang. Y et al. [11] have carried out the multi-strategy SA of the customer reviews on the basis of semantic fuzziness. This method has calculated the strengths and polarities of the Chinese sentiment phrases by using the value of probability rather than a fixed value. Two different multi-strategy SA methods namely, Support Vector Machine (SVM) and Naive Bayes (NB) techniques have been used. However, the areas related to compound sentiment phrases and modifiers of syntactical structure are not fully analysed with these techniques. In addition, the fuzzy evaluation of article is not carried out. T. Gu et al. [12] have proposed a SA model consisting of multichannel paradigm, which integrates Convolutional Neural Network (CNN), Bidirectional Gated Recurrent Unit (BiGRU), and Variational Information Bottleneck (VIB). The multi-grained sentiment features are extracted using multichannel, in which BiGRU is employed for context extraction and then the local features are extracted by CNN. This technique provides relevant sentiment features to address the challenges of SA that are occurred very rarely. However, the fine grained sentiments are not extracted with this approach. Y. Gao et al. [13] have proposed an Aspect-Level SA Approach on the basis of Collaborative Extraction Hierarchical Attention Network (CE-HEAT), which is evaluated by using SemEval competition at the sentence level in aspect-level SA. The CE-HEAT comprises of two hierarchical attention units, among which the former one extracts the sentiment features and the latter one extracts the aspect features. The CE-HEAT model achieves good performance on aspect-level sentiment classification. It is able to learn the potential relationship between the sentiment and aspect more efficiently. Nevertheless, some important considerations such as how to use the auxiliary information like, sentence structure, semantic features and logical relation to enhance the accuracy of sentiment classification based on aspect-level are not considered in this work. H. T. Phan et al. [14] have proposed the Feature Ensemble Model for enhancing the performance of SA of Tweets enclosing Fuzzy Sentiments. In addition, CNN models are used. This model significantly improves the performance in the SA of tweets. This method only has considered the tweets having fuzzy sentiment and not considered the effect of other elements in them such as sarcasm and slang. Y. Wang et al. [15] have proposed a new sentiment concept based Refined Global Word Embeddings (RGWE) for achieving sentiment words embedding and sentiment representation for words by combining two different refined word embedding techniques for achieving a more extensive word representation. RGWE not only integrates various position features but also integrates external and internal sentiment data with the average of Refined-GloVe and Refined-Word2Vec. Nevertheless, the SA for verbs, adjectives and adverbs are not accomplished with this approach.

Even though the researchers propose several SA approaches using variety of techniques and methodology, still very effective SA is not achieved till now. With this concern, a novel approach for SA application using Django web framework is proposed in this paper.

PROPOSED WORK

In the aspect-level SA, machine learning methods have been frequently employed and therefore a new SA application based on ML approach is proposed in this work. This approach includes, Stop Words, Count Vectorizer and ML classification algorithm, Naive Bayes are used. The input text data is given in the form of document or file. After this file or document is uploaded, several processes such as Data Pre-processing, feature extraction and Sentiment classification are required to achieve accurate sentiment classification. People's reviews or comments are mostly in text format, which might be difficult to interpret and absorb at times. Therefore, a proper procedure for the data sets is required to remove undesired, confusing information. Data Pre-processing is the method of converting data into a specific format that a computer understands. Filtering out insignificant data is one of the usual types of pre-processing. For such data Pre-processing, the Stop words are used. After that, the reviews must be represented as numerical values, which are taken into account as input by ML technique. Converting text reviews into numerical values is difficult, and it must be done correctly in order to reach an exact conclusion. Therefore, a count vectorizer is used in this work for vector conversion. Count Vectorizer extracts the desired features (unique words) from the stop words deleted data. It also turns a given input text into the format of vector based on the count or frequency of each word that exists throughout the text. Finally, the ML classification algorithm Naive Bayes classifies the text into positive, negative or neutral. The Fig. 1. shows the flow of processes to be carried on for sentiment classification.

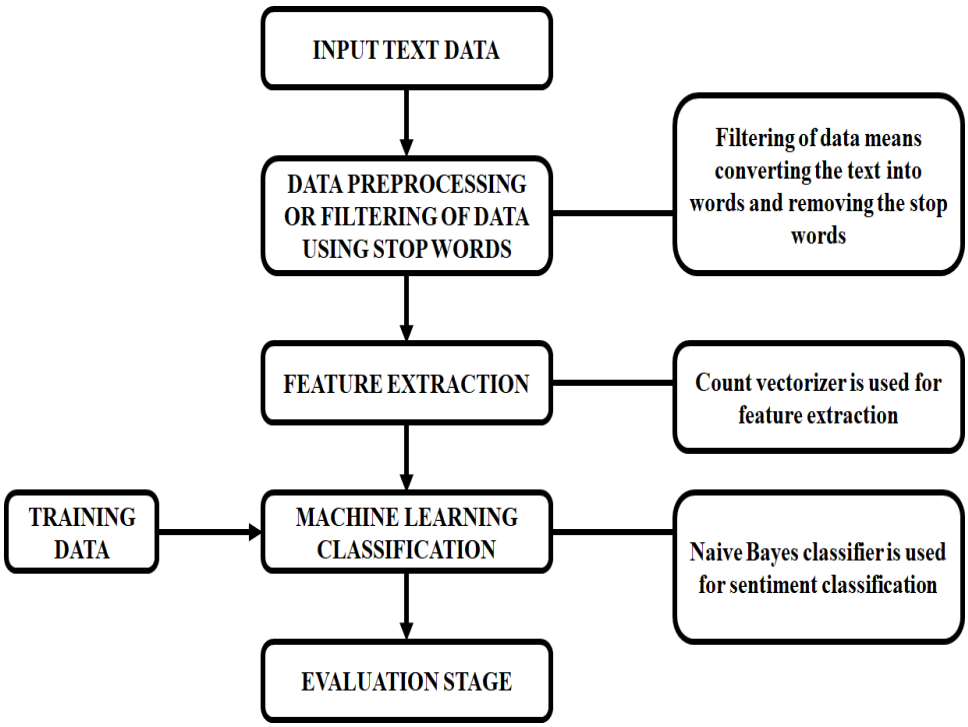

Fig.1. Architecture Diagram of the Proposed Work

A. Data Pre-Processing Using Stop Words

Data Pre-processing is the method of removing insignificant data from the texts. In the proposed SA model, stop words are used for removing the undesired data from the text. A stop word is an often used phrase ("the, "in" " "a," or "an,") to which a search engine is configured to ignore while guiding and recovering entries resulting from the search query. In the sentence, “how to enhance the malware detection approaches”, the major search query is about the malware detection. The search engine tries to find the web pages that contains the terms "how", "to" " enhance ", "the ", "malware", "detection", "approaches,". But the search engine attempts to provide more pages that contains the terms "how", "to" "the "than the pages that enclose the information about several enhancing malware detection approaches. Because the words "how" "to" and "the” are often used words in the language of English. Several stop words are available in English that are unnecessary for analysing the polarity of the sentiment. Some samples of stop words in English are “a”, “the”, “is”, “are” “for”, “an”, “nor”, “but”, “or”, “yet”, “so” etc. In addition to these words, a list of stop words are consolidated and shown in Fig.2. If such unwanted words are ignored, the search engine may concentrate on recovering pages that comprise the important keywords as explained in the above illustration.

['whence', 'here', 'show', 'were', 'why', 'n't', 'the', 'whereupon', 'not', 'more', 'how', 'eight', 'indeed', 'i', 'only', 'via', 'nine', 're', 'themselves', 'almost', 'to', 'already', 'front', 'least', 'becomes', 'thereby', 'doing', 'her', 'together', 'be', 'often', 'then', 'quite', 'less', 'many', 'they', 'ourselves', 'take', 'its', 'yours', 'each', 'would', 'may', 'namely', 'do', 'whose', 'whether', 'side', 'both', 'what', 'between', 'toward', 'our', 'whereby', "m", 'formerly', 'myself', 'had', 'really', 'call', 'keep', "re", 'hereupon', 'can', 'their', 'eleven', "m", 'even', 'around', 'twenty', 'mostly', 'did', 'at', 'an', 'seems', 'serious', 'against', "n't", 'except', 'has', 'five', 'he', 'last', "ve", 'because', 'we', 'himself', 'yet', 'something', 'somehow', "m", 'towards', 'his', 'six', 'anywhere', 'us', "d", 'thru', 'thus', 'which', 'everything', 'become', 'herein', 'one', 'in', 'although', 'sometime', 'give', 'cannot', 'besides', 'across', 'noone', 'ever', 'that', 'over', 'among', 'during', 'however', 'when', 'sometimes', 'still', 'seemed', 'get', "ve", 'him', 'with', 'part', 'beyond', 'everyone', 'same', 'this', 'latterly', 'no', 'regarding', 'elsewhere', 'others', 'moreover', 'else', 'back', 'alone', 'somewhere', 'are', 'will', 'beforehand', 'ten', 'very', 'most', 'three', 'former', "re", 'otherwise', 'several', 'also', 'whatever', 'am', 'becoming', 'beside', "s", 'nothing', 'some', 'since', 'thence', 'anyway', 'out', 'up', 'well', 'it', 'various', 'four', 'top', "s", 'than', 'under', 'might', 'could', 'by', 'too', 'and', 'whom', "ll", 'say', 'therefore', "s", 'other', 'throughout', 'became', 'your', 'put', 'per', "ll", 'fifteen', 'must', 'before', 'whenever', 'anyone', 'without', 'does', 'was', 'where', 'thereafter', "d", 'another', 'yourselves', 'n't', 'see', 'go', 'wherever', 'just', 'seeming', 'hence', 'full', 'whereafter', 'bottom', 'whole', 'own', 'empty', 'due', 'behind', 'while', 'onto', 'wherein', 'off', 'again', 'a', 'two', 'above', 'therein', 'sixty', 'those', 'whereas', 'using', 'latter', 'used', 'my', 'herself', 'hers', 'or', 'neither', 'forty', 'thereupon', 'now', 'after', 'yourself', 'whither', 'rather', 'once', 'from', 'until', 'anything', 'few', 'into', 'such', 'being', 'make', 'mine', 'please', 'along', 'hundred', 'should', 'below', 'third', 'unless', 'upon', 'perhaps', 'ours', 'but', 'never', 'whoever', 'fifty', 'any', 'all', 'nobody', 'there', 'have', 'anyhow', 'of', 'seem', 'down', 'is', 'every', "ll", 'much', 'none', 'further', 'me', 'who', 'nevertheless', 'about', 'everywhere', 'name', 'enough', "d", 'next', 'meanwhile', 'though', 'through', 'on', 'first', 'been', 'hereby', 'if', 'move', 'so', 'either', 'amongst', 'for', 'twelve', 'nor', 'she', 'always', 'these', 'as', "ve", 'amount', "re", 'someone', 'afterwards', 'you', 'nowhere', 'itself', 'done', 'hereafter', 'within', 'made', 'ca', 'them']

Fig. 2. Stop words

The stop words have been used in a wide range of ML approaches for SA that includes the supervised machine learning, Clustering, Information recovery and Word summarization. In supervised machine learning, it eliminates the Stop words from the feature space. In clustering, stop words are removed prior to creating clusters. In Word summarization, the Stop words are never involved in summary scores and they are detached when calculating ROUGE scores. Thus, the stop words are used in variety of situations like mentioned above. Since it has greater potential to remove the confusing words and unnecessary terms, the performance of the proposed SA model is effectively improved.

B. Featuer Extraction Uisng Count Vectorizer

The text data after removing the stop words are to be coverted into a matrix of tokens. For such vectorization, Count Vectorizer is employed to transform the data into format of vector. Count Vectorizer converts the input text data into the vectors depending on the frequency or count of each word that occurs in the whole text. Count vectorizer transforms the text document collection into a token count matrix. This function builds a sparse count matrix. The Count Vectorizer matrix is constructed in the following example. Assume there is a document that contains the following sentences. “This car is beautiful”, “This car is dirty”, and “This car is speedy”. The above sentences form a Count Vectorizer matrix of size 3*6 since there are three documents and six different features (“This”, “car”, “is”, “beautiful”, “dirty”, “speedy”). The vector matrix of the above example is tabulated in Table. 1.

Table 1: Example of Count Vectorizer matrix

Features / **Sentences**	**1**	**2**	**3**	**4**	**5**	**6**
1	1	1	1	1	0	0
2	1	1	1	0	1	0
3	1	1	1	0	0	1

In the above table, the presence of a feature is indicated by a '1', while the absence is indicated by a '0'. The first four features/words in "Sentence 1" are marked as '1', but "Feature 4" is marked as '0' in "Sentence 2" and "Sentence 3", and "Feature 5" and 'Feature 6" are marked as '1' in "Sentence 2" and "Sentence 3" respectively. Thus, the words in the text data are converted into vector format using Count Vectorizer. In the same procedure, the count vectorizer converts the data into matrix even for a large document.

C. Classification Using Naive Bayes Algorithm

In order to acquire the classification result, the text reviews must be transformed into numerical vectors and then processed using various machine learning algorithms. In our proposed SA model, Naive Bayes classifier is used. The proposed Naive Bayes classification algorithm is based on Bayes theorem, which predicts the sentiment polarity of the text. This classifier is used for predicting the membership probabilities to each class, like the probability that a particular data point make up that class. Another name for this is Maximum a Posteriori (MAP).

For a hypothesis with two occurrences A and B, the MAP is calculated as follows:

$$MAP\ (A) = \max\ P\left((A|B)\right) \quad (1)$$

$$MAP\ (A) = \max\ P\left((B|A)\right) * P(A)) \,/\, P(B) \quad (2)$$

Where a P (A) stands for probability of prior. The likelihood of evidence is denoted by the letter P (B). It's used to make the outcome more consistent. If it is deleted, it has no effect on the outcome. P (A\B) stands for probability of Posterior and P(B\A) stands for probability of Likelihood.

Based on the above equation, the sentiment classification of the given text is obtained as follows:

Assume that the number of documents is denoted as n, which are fit into k categories where k ∈ $\{c_1, c_2, \ldots, c_k\}$ and the output class c ∈ C. Then, the predicted output class is defined as follows:

$$P\ (c|d) = \frac{P\,(d|c)\,P(c)}{P\,(d)} \quad (3)$$

Where d represents the document and c represents the classes.

Based on this probability, the polarity of the expression is classified as negative, positive or neutral. The Naive Bayes Classifier has assumed that all of the features are unrelated. The absence or presence of one trait has no influence on the presence or absence of others. Text classification and spam filtering are two areas where the Naive Bayes Model is widely used.

Thus, the output obtained with this algorithm may predict that the given input text document is either positive or negative. Sometimes, the results may predict that the review is Neutral, which means that there is no opinion.

IMPLEMENTATION

The experimental environment requires a system with 1 GB Processor, 3.5 inches or more screen size and 2 GB RAM. The proposed SA model is implemented on Django, which is a python based web framework.

RESULTS AND DISCUSSION

A. Evaluation Metric

Accuracy is evaluated to verify the efficacy of the proposed SA application. The rate of accuracy is defined as the ratio of the number of results that are truly predicted to the total number of results predicted by the proposed SA application. The higher percentage of accuracy indicates that the proposed SA application is good in sentiment analysis. The formula for calculating the accuracy is given below.

$$Accuracy = \frac{n_{correct}}{n_{total}} \quad (4)$$

Where, $n_{correct}$ indicates the number of correct results predicted and n_{total} indicates the total number of results predicted by the application.

B. Predicted Results

In the SA application, the home page is first viewed in the proposed model. The Fig. 3 shows the home page that contains the tag line "*We're in the Business of Helping You Start Your Business*" under "*The Best Business Information "*. In addition, a Get started button is also exist in the home page. By using this, the user is able to move to get more ideas about SA.

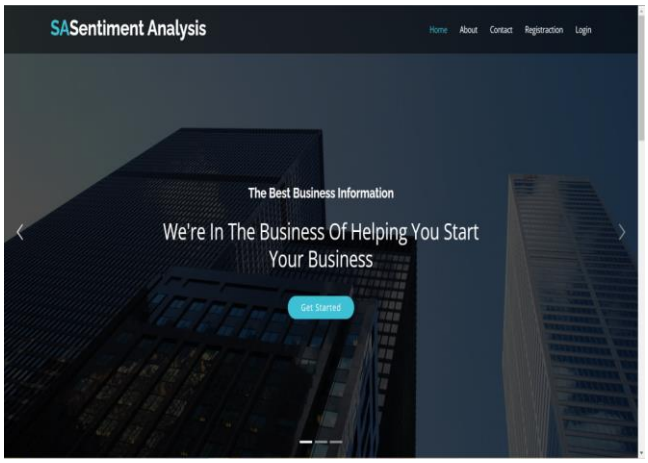

Fig. 3. Home Page

The Fig. 4 shows the 'about' page. In this page, the information about the SA is described such as the definition of SA and the characteristics of the SA.

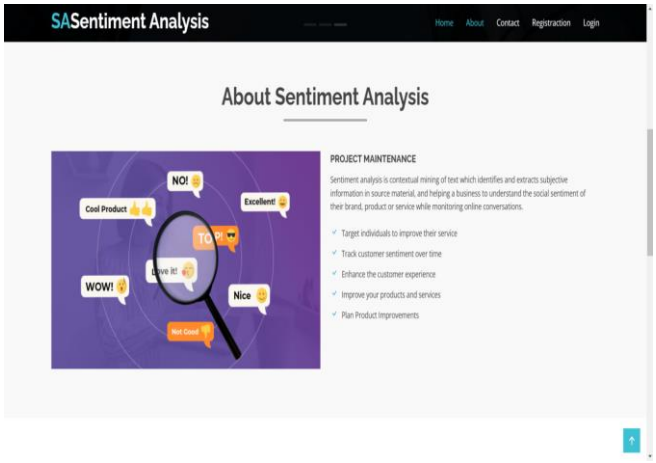

Fig. 4. About page

The Fig. 5 shows the 'Contact' page, which shows the location, contact information like, mail Id and phone number of the company or admin who use it. If the users have any queries related to the application, they are able to contact or clarify by sending the message. For this purpose, in the 'contact' page, the name, Mail ID, Subject and the Message box is given to describe the queries or doubts related to the application.

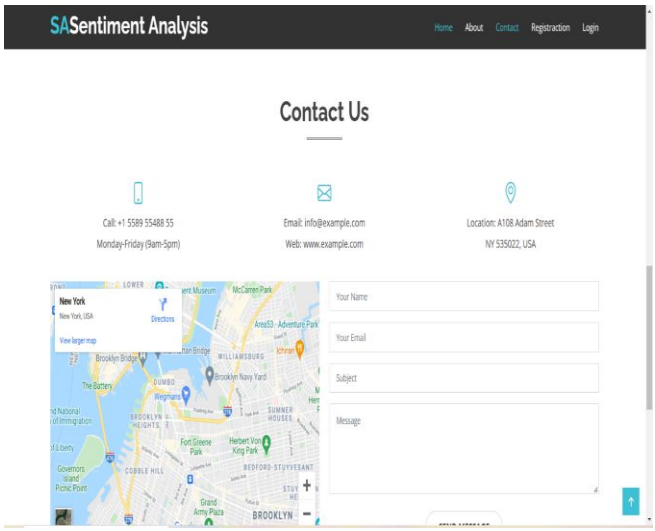

Fig. 5. Contact page

The Fig.6 shows the 'Registration' page, which asks to register the user when using this application for the first time. Once registered, the user has to login for visiting it next time. For registration, the details such as User name, Password and Password confirmation are to be filled and by clicking the Sign up button, the registration has done successfully.

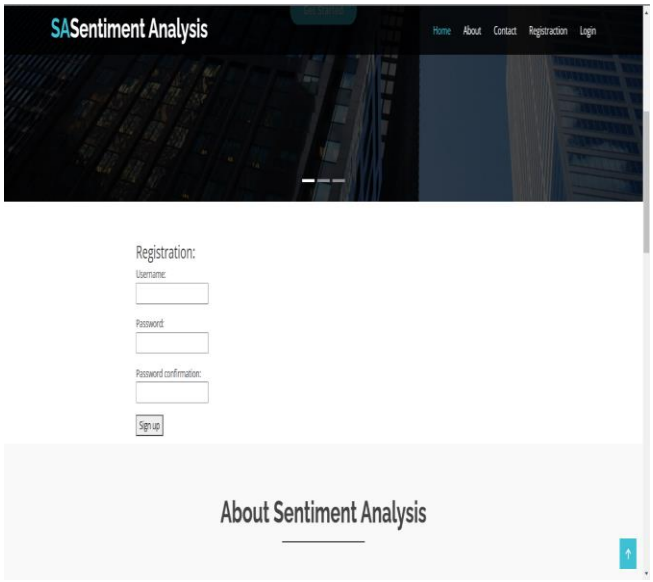

Fig. 6. Registration Page

The next page is 'Login' which is shown in Fig. 7. After registration, the registered user gets a chance of visiting or using this SA application at any time anywhere by using login option that requires the truthful User name and Password given at the time of registration. If both Password and User name are incorrect or mismatch, then user is not able to login. In this case, another option is provided to overcome this issue i.e., Forgot Password. If the user click it, he is able to change the password and use it for the further use.

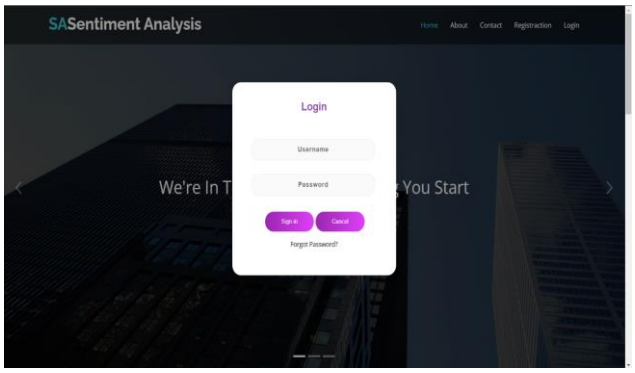

Fig. 7. Login Page

The Fig. 8 shows the 'Upload' page, where the file containing trained data is uploaded. In this uploaded file, all variety of reviews are available that are taken as trained data. The user has to click the check box for accepting the terms and conditions while uploading the training file.

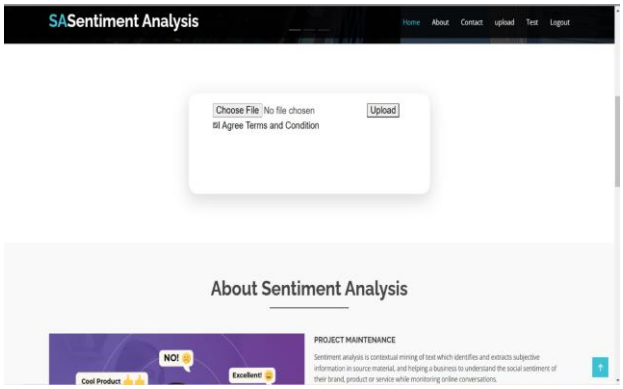

Fig. 8. Upload Page

Finally, the page for predicting or testing the review is shown in Fig. 9. In this page, a comment box with "Enter text here…" is exist, in which the users have to type their own reviews about any product or service. In this experiment, the review "I really like the new design of your website" is typed and then the button, 'PREDICT' is clicked. Thus, the final prediction of the comment is predicted as 'Positive'.

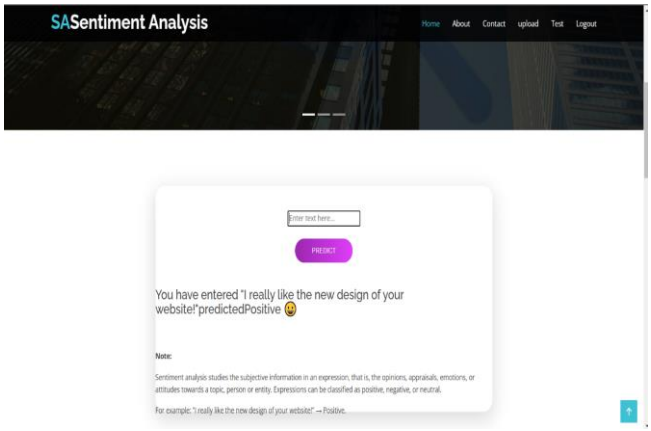

Fig. 9. Final Prediction Page

CONCLUSION

In this work, the Sentiment analysis application by using Django wen framework is developed. For increasing the classification accuracy of the reviews, a supervised classification approach is used in this model. This innovative sentiment analysis algorithm is intended to deliver precise sentiment information for words in various circumstances. Stop words are used for data pre-processing. The text data is turned into a vector for feature extraction by using count vectorizer. Finally, the Naive Bayes classifier, which is used in this work, which determines the type of sentiment, whether positive, negative, or neutral.

The additional research on this work is to be conducted in the following aspects: A limited amount of labelled data is sometimes available on any topic, but a significant volume of data is unlabeled. In that instance, a semi-supervised approach could be used, in which unlabeled data is turned into labelled data before being analysed further. Symbols such as emojies are used in many evaluations and comments to assist convey the sentiment, however these graphics require the use of special tools to analyse.

REFERENCES

L. Yang, Y. Li, J. Wang, and R. S. Sherratt, "Sentiment Analysis for E-Commerce Product Reviews in Chinese Based on Sentiment Lexicon and Deep Learning", in IEEE Access, vol. 08, pp. 23522-23530, 2020.

Cambria, and Erik, "Affective Computing and Sentiment Analysis", IEEE Intelligent Systems, vol. 31, no. 02, pp. 102–107, 2016.

Zhiying Ren, Guangping Zeng, Liu Chen, Qingchuan Zhang, Chunguang Zhang, and Dingqi Pan, "A Lexicon-Enhanced Attention Network for Aspect-Level Sentiment Analysis", IEEE Access, Vol: 08, DOI: 10.1109/ACCESS.2020.2995211, pp: 93464 – 93471, 2020.

W. Medhat, A. Hassan, and H. Korashy, "Sentiment analysis algorithms and applications: A survey", Ain Shams engineering journal, vol. 05, No: 04, pp. 1093-1113, 2014.

B. Zhang, X. Li, X. Xu, K. C. Leung, Z. Chen, and Y. Ye, "Knowledge Guided Capsule Attention Network for Aspect-Based Sentiment Analysis", in IEEE/ACM Transactions on Audio, Speech, and Language Processing, vol. 28, pp. 2538-2551, Doi: 10.1109/TASLP.2020.3017093, 2020.

S. M. Mohammad, "Challenges in sentiment analysis. In A practical guide to sentiment analysis", Springer Cham, pp. 61-83, 2017.

F. A. Pozzi, E. Fersini, E. Messina, and B. Liu, "Challenges of sentiment analysis in social networks: an overview", Sentiment analysis in social networks, pp. 01-11, 2017.

D. H Farias, and P. Rosso, "Irony, sarcasm, and sentiment analysis In Sentiment Analysis in Social Networks", Morgan Kaufmann, pp. 113-128, 2017.

Y. Gao, M. Gong, Y. Xie, and A. K. Qin, "An Attention-Based Unsupervised Adversarial Model for Movie Review Spam Detection", in IEEE Transactions on Multimedia, vol. 23, pp. 784-796, Doi: 10.1109/TMM.2020.2990085, 2021.

F. Alattar, and K. Shaalan, "Using Artificial Intelligence to Understand What Causes Sentiment Changes on Social Media", in IEEE Access, vol. 09, pp. 61756-61767, Doi: 10.1109/ACCESS.2021.3073657, 2021.

Y. Fang, H. Tan, and J. Zhang, "Multi-Strategy Sentiment Analysis of Consumer Reviews Based on Semantic Fuzziness", in IEEE Access, vol. 06, pp. 20625-20631, Doi: 10.1109/ACCESS.2018.2820025, 2018.

T. Gu, G. Xu, and J. Luo, "Sentiment Analysis via Deep Multichannel Neural Networks with Variational Information Bottleneck", in IEEE Access, vol. 08, pp. 121014-121021,
Doi: 10.1109/ACCESS.2020.3006569, 2020.

Y. Gao, J. Liu, P. Li, and D. Zhou, "CE-HEAT: An Aspect-Level Sentiment Classification Approach with Collaborative Extraction Hierarchical Attention Network", in IEEE Access, vol. 07, pp. 168548-168556, Doi: 10.1109/ACCESS.2019.2954590, 2019.

H. T. Phan, V. C. Tran, N. T. Nguyen, and D. Hwang, "Improving the Performance of Sentiment Analysis of Tweets Containing Fuzzy Sentiment Using the Feature Ensemble Model", in IEEE Access, vol. 08, pp. 14630-14641, Doi: 10.1109/ACCESS.2019.2963702, 2020.

Y. Wang, G. Huang, J. Li, H. Li, Y. Zhou, and H. Jiang, "Refined Global Word Embeddings Based on Sentiment Concept for Sentiment Analysis", in IEEE Access, vol. 09, pp. 37075-37085, Doi: 10.1109/ACCESS.2021.3062654, 2021.