International Journal For Advanced Research In Science & Technology
A peer reviewed international journal
www.ijarst.in
ISSN: 2457-0362

Integrated NLP with Machine Learning for Improved Hotel Review Analysis

M. Mahipal Reddy[1], Tatipamula Sanjay Kumar[2], Chakali Harikrishna[2], Mallam Sowmya[2], Perum Kovuri Shiva Shankar[2]

[1]Assistant Professor, [2]UG Scholar, [1,2]Department of Computer Science and Engineering (Data Science)

[1,2]Malla Reddy Engineering College and Management Sciences, Kistapur, Medchal, 501401, Telangana

ABSTRACT

Online reviews have great impact on today's business and commerce. Decision making for purchase of online products mostly depends on reviews given by the users. Hence, opportunistic individuals or groups try to manipulate product reviews for their own interests. In recent years, online reviews have become the most important resource of customers' opinions. These reviews are used increasingly by individuals and organizations to make purchase and business decisions. Unfortunately, driven by the desire for profit or publicity, fraudsters have produced deceptive (spam) reviews. The fraudsters' activities mislead potential customers and organizations reshaping their businesses and prevent opinion-mining techniques from reaching accurate conclusions. Fake reviews can be created in two main ways. First, in a (a) human-generated way by paying human content creators to write authentic-appearing but not real reviews of products — in this case, the review author never saw said products but still writes about them. Second, in a (b) computer-generated way by using text-generation algorithms to automate the fake review creation. Traditionally, human-generated fake reviews have been traded like commodities in a "market of fakes" – one can simply order reviews online in each quantity, and human writers would carry out the work. However, the technological progress in text generation – natural language processing (NLP) and machine learning (ML) to be more specific – has incentivized the automation of fake reviews, as with generative language models, fake reviews could be generated at scale and a fraction of the cost compared to human-generated fake reviews. This work introduces some semi-supervised and supervised text mining models to detect fake online reviews as well as compares the efficiency of both techniques on dataset containing hotel reviews.

Keywords: Recommendation system, Sentiment analysis, Hotel review analysis, Natural language processing, Machine learning.

1. INTRODUCTION

In this era of the internet, customers can post their reviews or opinions on several websites. These reviews are helpful for the organizations and for future consumers, who get an idea about products or services before making a selection. In recent years, it has been observed that the number of customer reviews has increased significantly [1]. Customer reviews affect the decision of potential buyers. In other words, when customers see reviews on social media, they determine whether to buy the product or reverse their purchasing decisions. Therefore, consumer reviews offer an invaluable service for individuals [2]. Positive reviews bring big financial gains, while negative reviews often exert a negative financial effect. Consequently, with customers becoming increasingly influential to the marketplace, there is a growing trend towards relying on customers' opinions to reshape businesses by enhancing products, services, and marketing [3]. For example, when several customers who purchased a specific model of Acer laptop posted reviews complaining about the low display quality, the manufacturer was

inspired to produce a higher-resolution version of the laptop.Fake reviews can be created in two main ways. First, in a (a) human-generated way by paying human content creators to write authentic-appearing but not real reviews of products — in this case, the review author never saw said products but still writes about them. Second, in a (b) computer-generated way by using text-generation algorithms to automate the fake review creation [4]. Traditionally, human-generated fake reviews have been traded like commodities in a "market of fakes" – one can simply order reviews online in a given quantity, and human writers would carry out the work. However, the technological progress in text generation – natural language processing (NLP) and machine learning (ML) to be more specific – has incentivized the automation of fake reviews, as with generative language models, fake reviews could be generated at scale and a fraction of the cost compared to human-generated fake reviews.Analyzing hotel reviews can provide valuable insights for both the hospitality industry and researchers in various fields. Here are some research motivations and potential areas of study related to hotel review analysis, such as improving customer satisfaction, topic modeling and trend analysis.Understanding the key drivers of customer satisfaction in hotels by analyzing reviews can help hotels and businesses in the service industry enhance their offerings and meet customer expectations.Develop advanced sentiment analysis techniques to not only identify positive and negative sentiments but also detect underlying emotions in hotel reviews. This can be valuable for understanding the emotional aspects of customer experiences. Use topic modeling techniques to identify the most discussed topics in hotel reviews over time [5]. This can reveal evolving customer preferences and emerging trends in the hospitality industry. Compare reviews of different hotels within the same region or category to identify what sets successful hotels apart from the rest. This can help hotels benchmark their performance.

2. LITERATURE SURVEY

Yuanyuan Wu et. al [11] proposes an antecedent–consequence–intervention conceptual framework to develop an initial research agenda for investigating fake reviews. Based on a review of the extant literature on this issue, they identify 20 future research questions and suggest 18 propositions. Notably, research on fake reviews is often limited by lack of high-quality datasets. To alleviate this problem, they comprehensively compile and summarize the existing fake reviews-related public datasets. They conclude by presenting the theoretical and practical implications of the current research.

Liu et. al [12] proposed a method for the detection of fake reviews based on review records associated with products. They first analyse the characteristics of review data using a crawled Amazon China dataset, which shows that the patterns of review records for products are similar in normal situations. In the proposed method, they first extract the review records of products to a temporal feature vector and then develop an isolation forest algorithm to detect outlier reviews by focusing on the differences between the patterns of product reviews to identify outlier reviews. They will verify the effectiveness of our method and compare it to some existing temporal outlier detection methods using the crawled Amazon China dataset. They will also study the impact caused by the parameter selection of the review records. Our work provides a new perspective of outlier review detection and our experiment demonstrates the effectiveness of our proposed method.

Mohawesh et. al [13] presented an extensive survey of the most notable works to date on machine learning-based fake review detection. Firstly, they have reviewed the feature extraction approaches used by many researchers. Then, they detailed the existing datasets with their construction methods. Then, they outlined some traditional machine learning models and neural network models applied for fake review detection with summary tables. Traditional statistical machine learning enhances text classification model performance by improving the feature extraction and classifier design. In contrast, deep learning improves performance by enhancing the presentation learning method, algorithm's structure and additional knowledge. They also provided a comparative analysis of some neural network

International Journal For Advanced Research In Science & Technology
A peer reviewed international journal
www.ijarst.in
ISSN: 2457-0362

model-based deep learning and transformers that have not been used in fake review detection. The outcomes showed that RoBERTa achieved the highest accuracy on both datasets. Further, recall, precision, and F1 score proved the efficacy of using RoBERTa in detecting fake reviews. Finally, they summarised the current gaps in this research area and the possible future direction to get robust outcomes in this domain.

Ahmed et. al [14] proposed a fake news detection model that use n-gram analysis and machine learning techniques. They investigate and compare two different features extraction techniques and six different machine classification techniques. Experimental evaluation yields the best performance using Term Frequency-Inverted Document Frequency (TF-IDF) as feature extraction technique, and Linear Support Vector Machine (LSVM) as a classifier, with an accuracy of 92%.

Atefeh Heydari et. al [15] proposed a robust review spam detection system. A detailed literature survey has shown potential of the timing element when applied to this domain and lead to the development of review spam detection approach based on time series analysis methods. Based on the consideration that the capture of burst patterns in reviewing process can improve the detection accuracy, in this experiment, they propose a review spam detection approach which investigates bogusness of reviews fallen.

3. PROPOSED SYSTEM

3.1 Overview

In this project, we make some classification approaches for detecting fake online reviews, some of which are semi-supervised, and others are supervised. For semi-supervised learning, we use Expectation-maximization algorithm. Statistical Naive Bayes classifier and Support Vector Machines (SVM) are used as classifiers in our research work to improve the performance of classification. We have mainly focused on the content of the review-based approaches. As feature we have used word frequency count, sentiment polarity and length of review. Figure 4.1 shows the proposed detailed system model.

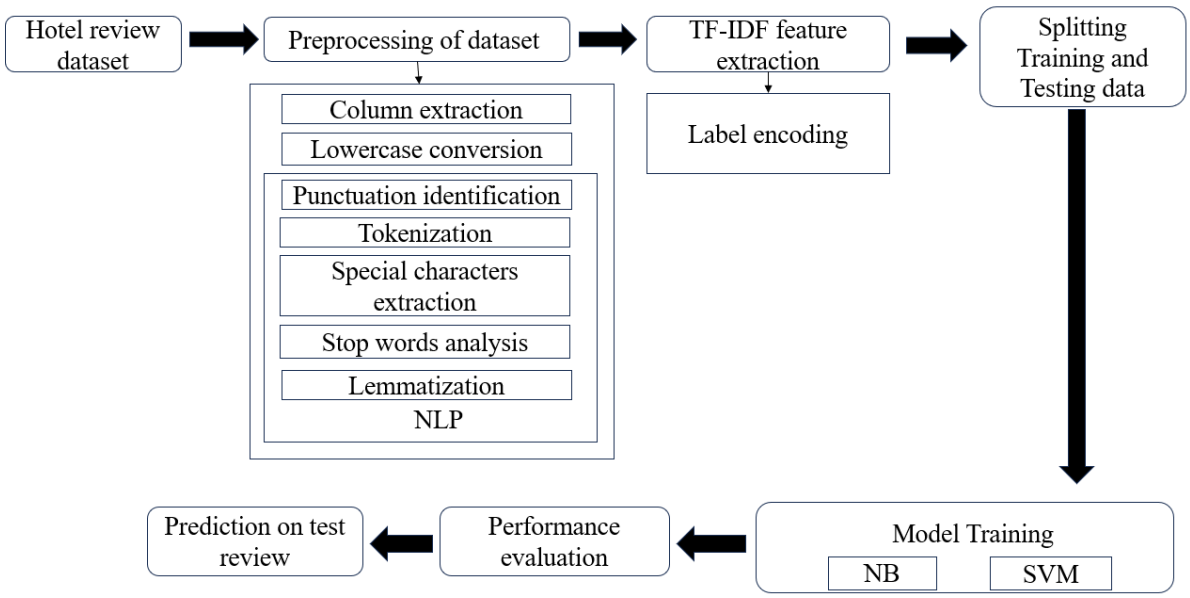

Fig. 1: Block diagram of proposed system.

The detailed operation illustrated in step wise as follows:

Step 1: Data Preprocessing: Load the hotel review dataset, which typically contains text reviews and corresponding labels (e.g., positive/negative sentiment). Perform data cleaning, including removing special characters, punctuation, and irrelevant information. Tokenize the text by splitting it into words or phrases. Convert text to lowercase to ensure consistency. Remove stop words (common words like "the," "is," "and") to reduce noise. Apply stemming or lemmatization to reduce words to their root form.

Step 2: NLP Processing: Conduct sentiment analysis or any other relevant NLP tasks on the preprocessed text data. Extract features that capture the sentiment or other meaningful information from the text reviews. Also create additional features like the length of the review or the frequency of certain words.

Step 3: TF-IDF Feature Extraction: Convert the preprocessed text into numerical vectors using TF-IDF (Term Frequency-Inverse Document Frequency) transformation. TF-IDF assigns weights to words based on their frequency in a document relative to their frequency across all documents. This results in a sparse matrix where each row represents a review, and each column represents a unique word or term.

Step 4: Existing EM-NB Classifier: This classifier combines the Naive Bayes algorithm with the Expectation-Maximization algorithm to handle missing data. Train the EM-NB classifier on the TF-IDF transformed training data. Evaluate the classifier's performance using metrics like accuracy, precision, recall, F1-score, and confusion matrix on the test data.

Step 5: Proposed SVM Classifier: Propose a Support Vector Machine (SVM) classifier as an alternative to EM-NB. SVM is known for its effectiveness in text classification tasks. Train the SVM classifier on the same TF-IDF transformed training data. Tune hyperparameters like the kernel function and regularization parameter for optimal performance.

Step 6: Hotel Review Prediction: Use SVM classifiers to predict hotel review sentiment or any other relevant labels on the test data.

Step 7: Performance Estimation: Compare the performance of the EM-NB and SVM classifiers using various evaluation metrics, including accuracy, precision, recall, F1-score, and ROC curves.

3.2 Data Preprocessing

Data pre-processing is a process of preparing the raw data and making it suitable for a machine learning model. It is the first and crucial step while creating a machine learning model. When creating a machine learning project, it is not always a case that we come across the clean and formatted data. And while doing any operation with data, it is mandatory to clean it and put in a formatted way. So, for this, we use data pre-processing task.

Why do we need Data Pre-processing?

A real-world data generally contains noises, missing values, and maybe in an unusable format which cannot be directly used for machine learning models. Data pre-processing is required tasks for cleaning the data and making it suitable for a machine learning model which also increases the accuracy and efficiency of a machine learning model.

Step 1: Load the Hotel Review Dataset: In this step, load the dataset into your data analysis environment. This dataset typically includes text reviews (the input) and corresponding labels (the output), which can be sentiments like "positive" or "negative." The goal is to train a model to predict these labels based on the text.

Step 2: Data Cleaning: Data cleaning involves several sub-steps: Removing Special Characters and Punctuation: Special characters (e.g., @, $, %) and punctuation (e.g., !, ?, .) are often irrelevant to sentiment analysis and can be removed to focus on the actual text content. Handling Irrelevant

Information: Sometimes, there might be metadata or other information in the text data that's not relevant to the analysis. You should remove such information to concentrate on the review text itself.

Step 3: Tokenization:

- Tokenization is the process of splitting the text into smaller units, such as words or phrases (tokens). This step is crucial because it breaks down the text into manageable pieces for further analysis.
- For example, the sentence "I love this hotel" would be tokenized into ["I", "love", "this", "hotel"].

Step 4: Convert Text to Lowercase:

- Converting all text to lowercase ensures consistency in your text data. It prevents the model from treating "good" and "Good" as two different words, which could lead to incorrect feature extraction and modeling.
- For example, "Good" and "good" should be treated as the same word.

Step 5: Remove Stop Words:

- Stop words are common words in a language that often don't carry significant meaning and can be safely removed to reduce noise in the data. Examples include "the," "is," "and," "in," etc.
- Removing stop words can help improve the efficiency of the model and reduce the dimensionality of the data without losing much valuable information.

Step 6: Apply Stemming or Lemmatization:

- Stemming and lemmatization are techniques used to reduce words to their root form, which helps in standardizing words and improving feature extraction.
- Stemming: It involves removing suffixes from words to obtain the word stem. For example, "jumping" becomes "jump," "flies" becomes "fli," etc. Stemming is more aggressive but may result in non-words.
- Lemmatization: It is a more advanced technique that reduces words to their base or dictionary form (lemma). For example, "better" becomes "good," "running" becomes "run," etc. Lemmatization is more accurate but computationally expensive.
- The choice between stemming and lemmatization depends on your specific NLP task and dataset.

4.3 Dataset Splitting

In machine learning data pre-processing, we divide our dataset into a training set and test set. This is one of the crucial steps of data pre-processing as by doing this, we can enhance the performance of our machine learning model. Suppose if we have given training to our machine learning model by a dataset and we test it by a completely different dataset. Then, it will create difficulties for our model to understand the correlations between the models. If we train our model very well and its training accuracy is also very high, but we provide a new dataset to it, then it will decrease the performance. So, we always try to make a machine learning model which performs well with the training set and with the test dataset.

Training Set: A subset of dataset to train the machine learning model, and we already know the output.

International Journal For Advanced Research In Science & Technology
A peer reviewed international journal
www.ijarst.in
ISSN: 2457-0362

Test set: A subset of dataset to test the machine learning model, and by using the test set, model predicts the output.

4.4 TF-IDF Feature Extraction

TF-IDF which stands for Term Frequency – Inverse Document Frequency. It is one of the most important techniques used for information retrieval to represent how important a specific word or phrase is to a given document. Let's take an example, we have a string or Bag of Words (BOW) and we have to extract information from it, then we can use this approach.

Figure 4.2 shows the TF-IDF feature extraction block diagram. The tf-idf value increases in proportion to the number of times a word appears in the document but is often offset by the frequency of the word in the corpus, which helps to adjust with respect to the fact that some words appear more frequently in general. TF-IDF use two statistical methods, first is Term Frequency and the other is Inverse Document Frequency. Term frequency refers to the total number of times a given term t appears in the document doc against (per) the total number of all words in the document and the inverse document frequency measure of how much information the word provides. It measures the weight of a given word in the entire document. IDF show how common or rare a given word is across all documents. TF-IDF can be computed as $tf * idf$.

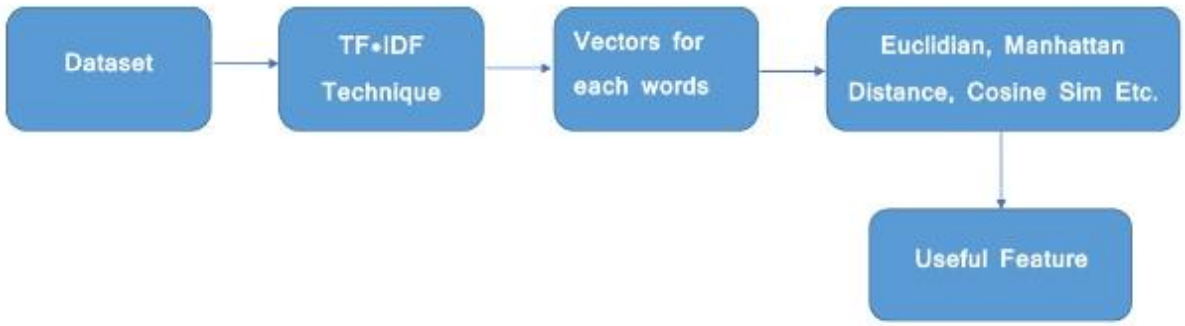

Fig. 2: TF-IDF block diagram.

TF-IDF do not convert directly raw data into useful features. Firstly, it converts raw strings or dataset into vectors and each word has its own vector. Then we'll use a particular technique for retrieving the feature like Cosine Similarity which works on vectors, etc.

Terminology

t — term (word)

d — document (set of words)

N — count of corpus

corpus — the total document set

Step 1: Term Frequency (TF): Suppose we have a set of English text documents and wish to rank which document is most relevant to the query, "Data Science is awesome!" A simple way to start out is by eliminating documents that do not contain all three words "Data" is", "Science", and "awesome", but this still leaves many documents. To further distinguish them, we might count the number of times each term occurs in each document; the number of times a term occurs in a document is called its term frequency. The weight of a term that occurs in a document is simply proportional to the term frequency.

$$tf(t,d) = count\ of\ t\ in\ d\ /\ number\ of\ words\ in\ d$$

Step 2: Document Frequency: This measures the importance of document in whole set of corpora, this is very similar to TF. The only difference is that TF is frequency counter for a term t in document d, whereas DF is the count of occurrences of term t in the document set N. In other words, DF is the number of documents in which the word is present. We consider one occurrence if the term consists in the document at least once, we do not need to know the number of times the term is present.

$$df(t) = occurrence\ of\ t\ in\ documents$$

Step 3: Inverse Document Frequency (IDF): While computing TF, all terms are considered equally important. However, it is known that certain terms, such as "is", "of", and "that", may appear a lot of times but have little importance. Thus, we need to weigh down the frequent terms while scale up the rare ones, by computing IDF, an inverse document frequency factor is incorporated which diminishes the weight of terms that occur very frequently in the document set and increases the weight of terms that occur rarely. The IDF is the inverse of the document frequency which measures the informativeness of term t. When we calculate IDF, it will be very low for the most occurring words such as stop words (because stop words such as "is" is present in almost all of the documents, and N/df will give a very low value to that word). This finally gives what we want, a relative weightage.

$$idf(t) = N/df$$

Now there are few other problems with the IDF, in case of a large corpus, say 100,000,000, the IDF value explodes, to avoid the effect we take the log of idf . During the query time, when a word which is not in vocab occurs, the df will be 0. As we cannot divide by 0, we smoothen the value by adding 1 to the denominator.

$$idf(t) = log(N/(df + 1))$$

The TF-IDF now is at the right measure to evaluate how important a word is to a document in a collection or corpus. Here are many different variations of TF-IDF but for now let us concentrate on this basic version.

$$tf - idf(t,d) = tf(t,d) * log(N/(df + 1))$$

4. RESULTS

4.1 Implementation

The implementation of this work focused on creating a graphical user interface (GUI) application using the Tkinter library. The application is designed for hotel review analysis, including preprocessing text data, running machine learning algorithms (EM-Naive Bayes and SVM), and predicting sentiment and authenticity of reviews. Here's a breakdown of the code:

- Import Libraries:
    - The code starts by importing the necessary libraries for building the GUI and performing various data processing and machine learning tasks. These libraries include Tkinter, Matplotlib, NumPy, Pandas, NLTK, Scikit-Learn, and more.
- GUI Initialization:
    - The main window of the GUI is created using Tk() and given the title "Integrated NLP with Machine Learning for Improved Hotel Review Analysis." The window size is set to 1300x1200 pixels.
- Data Processing Functions:

International Journal For Advanced Research In Science & Technology
A peer reviewed international journal
www.ijarst.in
ISSN: 2457-0362

- Several functions are defined for data processing and machine learning tasks. These functions include:
  - cleanPost(doc): Preprocesses a text document by removing special characters, punctuation, stop words, and lemmatizing words.
  - uploadDataset(): Allows the user to upload a dataset containing hotel reviews and labels.
  - preprocess(): Preprocesses the uploaded dataset by cleaning text, extracting features using TF-IDF, and splitting it into training and test sets.
  - TFIDFfeatureEng(): Performs TF-IDF feature extraction on the preprocessed text data.
  - gaussianKernelGramMatrixFull(X1, X2, sigma=0.1): Computes a Gaussian kernel Gram matrix for SVM.
  - EMNaiveBayes(): Runs the EM-Naive Bayes algorithm on the preprocessed data, calculates accuracy, and displays classification reports and confusion matrices.
  - runSVM(): Runs the SVM algorithm on the preprocessed data, calculates accuracy, and displays classification reports and confusion matrices.
  - predict(): Allows the user to upload a test review dataset, predicts sentiment and authenticity, and displays results.

Figure 3. Preprocessed dataset.

International Journal For Advanced Research In Science & Technology
A peer reviewed international journal
www.ijarst.in
ISSN: 2457-0362

Figure 3 represents the output of the TF-IDF (Term Frequency-Inverse Document Frequency) feature extraction process. It shows the TF-IDF vectors that were created from the preprocessed reviews. Each row corresponds to a review, and each column corresponds to a unique term in the corpus. The values in the matrix represent the TF-IDF weights of the terms in the reviews.

Table 1 presents the classification report for the EM-Naive Bayes model. The classification report provides various metrics for each class (0 and 1) and overall averages. The metrics include accuracy, precision, recall, and F1-score. These metrics evaluate the performance of the model in predicting each class and overall. It also includes the support, which is the number of samples in each class.

Table 1. EM-Naive Bayes Classification Report

Class	Accuracy	Precision	Recall	F1-Score	Support
0	-	0.69	0.91	0.79	147
1	-	0.90	0.66	0.76	173
Macro Avg	-	0.80	0.79	0.77	320
Weighted Avg	0.78	0.80	0.78	0.77	320

Like Table 1, this table 2 presents the classification report for the SVM model. It provides the same set of metrics for each class (0 and 1) and overall averages, which shows superior performance than the conventional EM-NB.

Table .2. SVM Classification Report

Class	Accuracy	Precision	Recall	F1-Score	Support
0	-	0.83	0.85	0.84	147
1	-	0.87	0.85	0.86	173
Macro Avg	-	0.85	0.85	0.85	320
Weighted Avg	0.85	0.85	0.85	0.85	320

A confusion matrix visually represents the performance of a classification model. This figure 10.4 show the confusion matrix specific to the EM-NB model. It includes the true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN) for both classes (0 and 1).

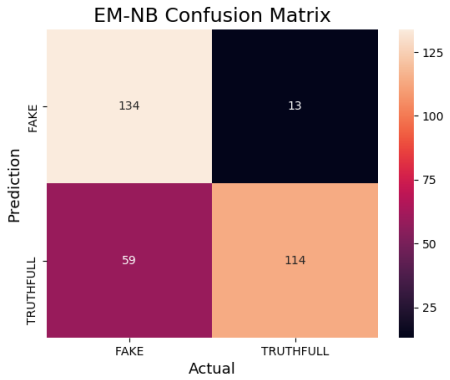

Figure 4. EM-NB confusion matrix

International Journal For Advanced Research In Science & Technology
A peer reviewed international journal
www.ijarst.in
ISSN: 2457-0362

Like Figure 4 Figure 5 represents the confusion matrix for the SVM model. It shows well the SVM model predicted each class and where the model made errors.

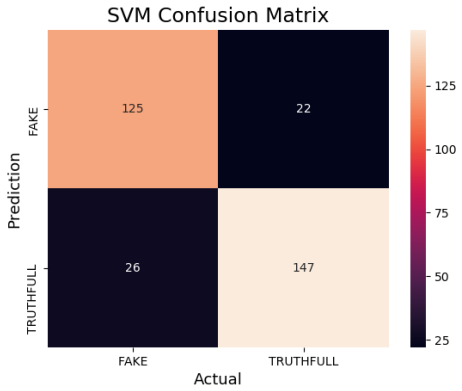

Figure 5. SVM confusion matrix.

Figure 6 shows the prediction results from test data. Here, for the applied test input tweet, the review predicted as truthful, fake, neutal.

Figure 6. Prediction from test data.

Figure 7 shows the sentiment graph. Here, the 13.4% of reviews are predicted as negative, 19.5% of reviews are predicted as positive, and 67.1% of reviews are predicted as neutral.

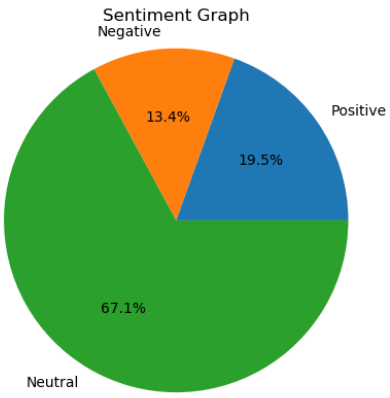

Figure 7. Sentiment graph.

5. Conclusion

This work focused on hotel review data analysis and prediction, the dataset was meticulously preprocessed by addressing text-related challenges such as special character removal, tokenization, lowercase conversion, stop word elimination, and stemming or lemmatization. Natural Language Processing (NLP) techniques were employed to extract meaningful features, including sentiment, and other relevant characteristics from the textual reviews. Furthermore, the TF-IDF technique was applied to transform the preprocessed text into numerical vectors, facilitating the representation of reviews as feature-rich sparse matrices. Two distinct classification approaches were considered: firstly, an existing EM-NB classifier was employed. EM-NB is notable for its ability to effectively handle missing data by incorporating the Expectation-Maximization algorithm into the traditional Naive Bayes framework. This classifier was trained using the TF-IDF transformed training data and subsequently evaluated on the test data utilizing an array of performance metrics, thereby enabling a comprehensive assessment of its predictive capabilities. Secondly, SVM classifier was proposed as an alternative to EM-NB. SVMs are renowned for their proficiency in text classification tasks. The SVM classifier was trained on the same TF-IDF transformed dataset, and hyperparameters, such as the choice of kernel function and regularization parameter, were carefully tuned to optimize its performance. Both the EM-NB and SVM classifiers were then utilized to predict hotel review sentiments or other pertinent labels on the test data. The resulting predictions were meticulously stored for subsequent analysis.

REFERENCES

[1] Ameur, Asma, Sana Hamdi, and Sadok Ben Yahia. "Arabic Aspect Category Detection for Hotel Reviews based on Data Augmentation and Classifier Chains." *Proceedings of the 38th ACM/SIGAPP Symposium on Applied Computing*. 2023.

[2] Kusumaningrum, Retno, et al. "Deep learning-based application for multilevel sentiment analysis of Indonesian hotel reviews." Heliyon (2023).

[3] Li, Chunhong, et al. "Let photos speak: the effect of user-generated visual content on hotel review helpfulness." *Journal of Hospitality & Tourism Research* 47.4 (2023): 665-690.

[4] Madhoushi, Zohreh, Abdul Razak Hamdan, and Suhaila Zainudin. "Semi-Supervised Model for Aspect Sentiment Detection." *Information* 14.5 (2023): 293.

[5] Almasri, Miada, Norah Al-Malki, and Reem Alotaibi. "A semi supervised approach to Arabic aspect category detection using Bert and teacher-student model." *PeerJ Computer Science* 9 (2023): e1425.

[6] Alawadh, Husam M., et al. "Semantic Features-Based Discourse Analysis Using Deceptive and Real Text Reviews." *Information* 14.1 (2023): 34.

[7] Lu, Junwen, et al. "BSTC: A Fake Review Detection Model Based on a Pre-Trained Language Model and Convolutional Neural Network." *Electronics* 12.10 (2023): 2165.

[8] Sajid, Tanveer, et al. "Analysis and Challenges in Detecting the Fake Reviews of Products using Naïve Bayes and Random Forest Techniques." (2023).

[9] Shekhawat, Sayar Singh, Harish Sharma, and Sandeep Kumar. "Memetic spider monkey optimization for spam review detection problem." *Big Data* 11.2 (2023): 137-149.

[10] Peng, Jing, Yue Wang, and Yuan Meng. "Detecting E-Commerce Water Army through Graph Modeling on User Multiple Collusive Relationships: A Case Study of China's Hotel Industry." *Journal of Theoretical and Applied Electronic Commerce Research* 18.1 (2023): 105-129.

[11] Yuanyuan Wu, Eric W.T. Ngai, Pengkun Wu, Chong Wu, Fake online reviews: Literature review, synthesis, and directions for future research, Decision Support Systems, Volume 132, 2020, 113280, ISSN 0167-9236, https://doi.org/10.1016/j.dss.2020.113280.

[12] W. Liu, J. He, S. Han, F. Cai, Z. Yang and N. Zhu, "A Method for the Detection of Fake Reviews Based on Temporal Features of Reviews and Comments," in IEEE Engineering Management Review, vol. 47, no. 4, pp. 67-79, 1 Fourthquarter,Dec. 2019, doi: 10.1109/EMR.2019.2928964.

[13] R. Mohawesh. "Fake Reviews Detection: A Survey," in IEEE Access, vol. 9, pp. 65771-65802, 2021, doi: 10.1109/ACCESS.2021.3075573.

[14] Ahmed, H., Traore, I., Saad, S. (2017). Detection of Online Fake News Using N-Gram Analysis and Machine Learning Techniques. In: Traore, I., Woungang, I., Awad, A. (eds) Intelligent, Secure, and Dependable Systems in Distributed and Cloud Environments. ISDDC 2017. Lecture Notes in Computer Science (), vol 10618. Springer, Cham. https://doi.org/10.1007/978-3-319-69155-8_9

[15] Atefeh Heydari, Mohammadali Tavakoli, Naomie Salim, Detection of fake opinions using time series, Expert Systems with Applications, Volume 58, 2016, Pages 83-92, ISSN 0957-4174, https://doi.org/10.1016/j.eswa.2016.03.020.