ISSN 2277-2685
IJESR/June-2024/ Vol-14/Issue-2s/345-355
T. Ravindar *et. al.,* / International Journal of Engineering & Science Research

IMPROVING HOTEL REVIEW ANALYSIS WITH DEEP LEARNING: A FOCUS ON SUPERVISED AND SEMI SUPERVISED MODELS

T. Ravindar[1], T. Vivek[2], K.Shiva Subramanyam[2], Md. Usaid Afzal[2]
[1]Assist. Professor, [2]UG Scholar, [1,2]Department of Computer Science & Engineering (AI & ML),
[1,2]Kommuri Pratap Reddy Institute of Technology, Ghatkesar, Hyderabad, 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: NLP, SVM, Tokenization, Hotel review analysis.

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]

2. LITERATURE SURVEY

Yuanyuan Wu et. al [6] 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 [7] 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 [8] 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 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 [9] 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 [10] 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 1 shows the proposed detailed system model.

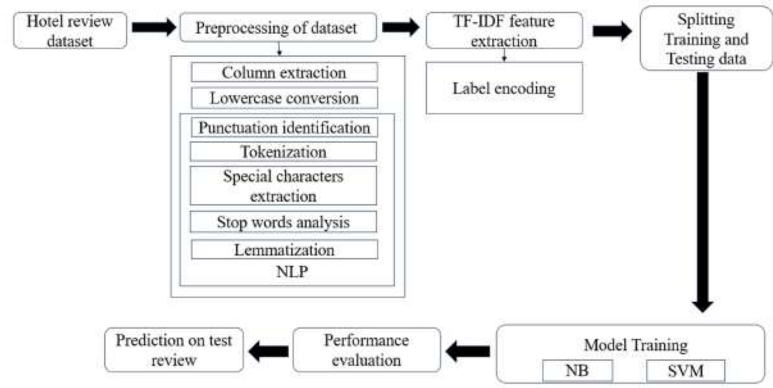


Fig. 1: Block diagram of proposed system.

The detailed operation illustrated in step wise as follows:

3.2 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 $tf * idf$.

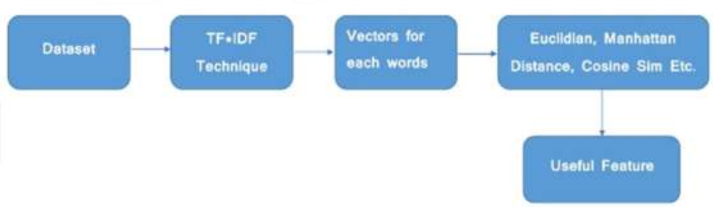


Fig. 2: TF-IDF block diagram.

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.

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

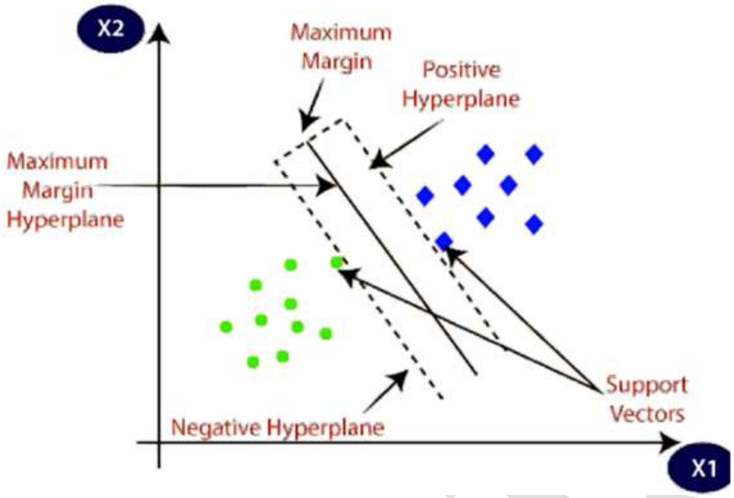
The TF-IDF now is at the right measure to evaluate how important a word is to a document in a collection or corpus. Here are many different variations of TF-IDF but for now let us concentrate on this basic version.

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

3.3 Support Vector Machine Model

Support Vector Machine or SVM is one of the most popular Supervised Learning algorithms, which is used for Classification as well as Regression problems. However, primarily, it is used for Classification problems in Machine Learning. The goal of the SVM algorithm is to create the best line or decision boundary that can segregate n-dimensional space into classes so that we can easily put the new data point in the correct category in the future. This best decision boundary is called a hyperplane.

SVM chooses the extreme points/vectors that help in creating the hyperplane. These extreme cases are called as support vectors, and hence algorithm is termed as Support Vector Machine. Consider the below diagram in which there are two different categories that are classified using a decision boundary or hyperplane:



Machine learning involves predicting and classifying data and to do so we employ various machine learning algorithms according to the dataset. SVM or Support Vector Machine is a linear model for classification and regression problems. It can solve linear and non-linear problems and work well for many practical problems. The idea of SVM is simple: The algorithm creates a line or a hyperplane which separates the data into classes. In machine learning, the radial basis function kernel, or RBF kernel, is a popular kernel function used in various kernelized learning algorithms. In particular, it is commonly used in support vector machine classification. As a simple example, for a classification task with only two features (like the image above), you can think of a hyperplane as a line that linearly separates and classifies a set of data.

- Intuitively, the further from the hyperplane our data points lie, the more confident we are that they have been correctly classified. We therefore want our data points to be as far away from the hyperplane as possible, while still being on the correct side of it.
- So, when new testing data is added, whatever side of the hyperplane it lands will decide the class that we assign to it.

4. RESULTS AND DISCUSSION

Figure 3 represents a portion of the original dataset. It shows some examples of the reviews and their associated labels before any preprocessing or feature extraction.

```
Review,Label
"we stay at hilton for 4 nights last march it was a pleasant stay we got a large room with 2 double beds an
"this is a stunning hotel in an excellent location in the greatest of us cities the entrance and lobby of t
"staying at this hotel was one of the high points of a last minute , budget valentines weekend trip for my
"went to chicago for a week in may , decided to be good to ourselves and stay in the hilton , we were not d
"we stayed here from nov 30 to dec 2 and had a wonderful time the hotel is just beautiful and the service w
"we travel to chicago regularly and have always wanted to stay at the hilton chicago we booked a priceline
"i stayed here for a conference and got the conference rate of 149 i was worried that i would n't get a nic
"we had a great experience at this hotel the hotel is huge ! the rooms were very clean , well appointed , a
"we had our hotel reservations at another hotel set and after we were reading all of the negative reviews w
"i stayed at this hotel over the weekend of the chicago bears fan convention \( feb 27 march 1 \) the hotel
"my stay was quick but awesome after a long day , seeing a substantial line at check in was a bit of a groa
"thirty years ago , we had a tiny room and indifferent service this time , the service was superb and frien
"stayed at the chicago hilton for three nights and from the minute we walked through the door i was very im
"we loved the hotel when i see other posts about it being shabby i ca n't for the life of me figure out wha
"i took the wife and kids to chicago for one last fling before school started back up when i checked in , t
"in the windy city , this is a very good place amazing rooms and service everything is a walking distance a
"i only stayed out with my boyfriend for one night , however enjoyed my stay the staff was friendly , the r
"we booked our stay through priceline com and got a great deal we were only staying one night and we had ou
"great place , great room , great location even though there was a big meeting going on \( rainbow girls in
"my family and i have just had a two week holiday in chicago , and we stayed for a week at the hilton tower
"dispite what other are saying , this was one , if not the best hotel stay in chicago i have had i travel t
"the james hotel is located close to everything in chicago , the miracle mile , pizza house uno , rush stre
"i found this wonderful hotel ! the location is awesome , just minutes away from all the shopping , restaur
"room i stayed in the 16thz floor 2 double bed room the room has a decent size equipment is state of the ar
"my stay at the the james was perfect my room was thoughtfully designed the lighting , the storage space ,
"we needed an extra night in chicago after a gratus stay at the penninsula and won a skyauction bid for the
```

Figure 3. Sample dataset.

Figure 4 represents the same dataset but after applying preprocessing steps like lowercasing, punctuation removal, stop word removal, and lemmatization. The reviews are transformed into a cleaner format suitable for further analysis.

```
stay hilton night last march pleasant stay got large room double bed bathroom tv ok crt flat screen coin
cierge friendly need room cleaned arrived ordered pizza room service pizza ok also main hall beautiful
breakfast charged dollar kinda expensive internet access wifi charged dollar day pro low rate price hu
ge room close attraction loop close metro station con expensive breakfast internet access charged tip l
eaving building always use michigan av exit great view ==== 1
stunning hotel excellent location greatest u city entrance lobby hotel indicates class bedroom large co
mfortable customer service second none located south michigan directly across road grant millenium p
ark also free shuttle water tower heart magnificent mile blue fan buddy guy legend club situated imme
diately behind hotel highly recommended ==== 1
staying hotel one high point last minute budget valentine weekend trip husband got great rate priceline
close subway red line got ohare hotel le hour u able check hour early room great clean good closet spa
ce fantastic bedding one expensive drink irish bar bartender advice restaurant club check made worth
price concierge helpful overall service terrific room great staying treat would stay time especially price
 ==== 1
went chicago week may decided good stay hilton disapointed perhaps becuse quite convention going lo
t people staying night got upgraged exceutive level double bed bathroom bed pillow die comfy ant end
day seemed walked mile staff helpful lot guest seemed ignore staff especially chamber maid seemed th
ink way perhaps people felt people rude unhelpful block away harrison cafe called orange make place
breakfast cafe staff suberb expect min wait sat sun morning ==== 1
stayed nov dec wonderful time hotel beautiful service excellent check maid staff bartender kitty oshea
room king bed comfortable nice feather pillow request type problem feather large flat screen tv nice ba
th product crabtree evelyn included shampoo conditioner mouthwash bady lotion plenty coffee provide
d drank snack kitty oshea crowd fun live irish folk music night good selection beer good shepard pie ch
eese dip crisp location worked well u walked art museum buddy guy blue club right across street shopp
ing breeze using shuttle cab ride museum science industry quite far though cost way excellent weekend
 getaway ==== 1
travel chicago regularly always wanted stay hilton chicago booked priceline room weekend half marath
on place busy staff certainly found time individual guest room nice expected pool area also nice downto
wn hotel location great taking bear game thursday night sox game friiday night would certainly stay wo
uld recomment others great quality even regular room price ==== 1
```

Figure 4. Preprocessed dataset.

Figure 5 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.

```
      able absolutely  access accommodating accommodation account ... white    wrong year yes
yet young
0    0.000000     0.0 8.230158     0.0      0.000000     0.0 ...  0.0 0.000000  0.0 0.0 0.0   0.0
1    0.000000     0.0 0.000000     0.0      0.000000     0.0 ...  0.0 0.000000  0.0 0.0 0.0   0.0
2    3.677279     0.0 0.000000     0.0      0.000000     0.0 ...  0.0 0.000000  0.0 0.0 0.0   0.0
3    0.000000     0.0 0.000000     0.0      0.000000     0.0 ...  0.0 0.000000  0.0 0.0 0.0   0.0
4    0.000000     0.0 0.000000     0.0      0.000000     0.0 ...  0.0 0.000000  0.0 0.0 0.0   0.0
...       ...     ...      ...     ...           ...     ... ...  ...      ...  ... ... ...   ...
1595 0.000000     0.0 0.000000     0.0      4.370426     0.0 ...  0.0 0.000000  0.0 0.0 0.0   0.0
1596 0.000000     0.0 0.000000     0.0      0.000000     0.0 ...  0.0 0.000000  0.0 0.0 0.0   0.0
1597 3.677279     0.0 0.000000     0.0      0.000000     0.0 ...  0.0 4.370426  0.0 0.0 0.0   0.0
1598 0.000000     0.0 0.000000     0.0      0.000000     0.0 ...  0.0 0.000000  0.0 0.0 0.0   0.0
1599 0.000000     0.0 0.000000     0.0      0.000000     0.0 ...  0.0 0.000000  0.0 0.0 0.0   0.0

[1600 rows x 1000 columns]

Total Reviews found in dataset : 1600
Total records used to train machine learning algorithms : 1280
Total records used to test machine learning algorithms  : 320
```

Figure 5. TF-IDF output features.

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. 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).

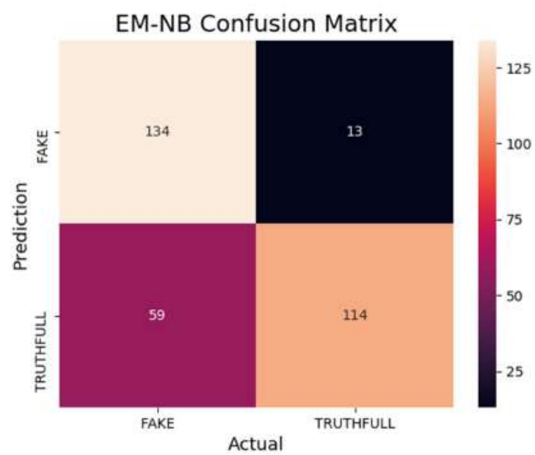


Figure 6. EM-NB confusion matrix

Like Figure 6, Figure 7 represents the confusion matrix for the SVM model. It shows well the SVM model predicted each class and where the model made errors.

ISSN 2277-2685
IJESR/June-2024/ Vol-14/Issue-2s/345-355
T. Ravindar *et. al.,* / International Journal of Engineering & Science Research

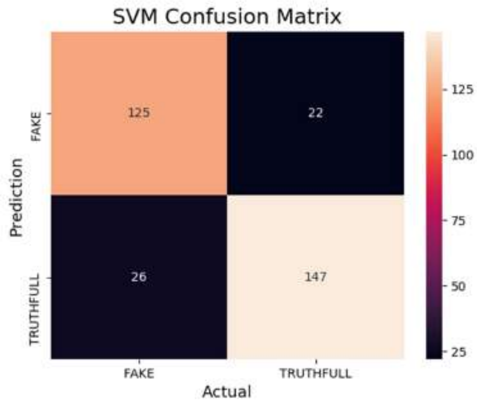


Figure 7. SVM confusion matrix.

Figure 8 shows the prediction results from test data. Here, for the applied test input tweet, the review predicted as truthful, fake, neutal.

Figure 8. Prediction from test data.

Figure 9 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.

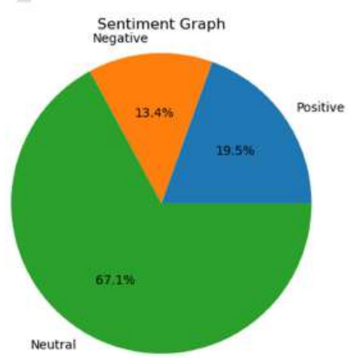


Figure 9 Sentiment graph.

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.

To gauge the performance of these classifiers, a diverse range of metrics, including accuracy, precision, recall, F1-score, ROC curves, and potentially cross-validation, were employed. This comprehensive evaluation enabled an informed selection of the model that excelled in terms of the chosen evaluation criteria. The chosen model can now be subjected to further fine-tuning and, if deemed appropriate, deployed for real-world applications, offering valuable insights and predictions regarding hotel reviews, thereby enhancing decision-making and customer experiences in the hotel industry.

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] 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.

[7] 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.

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

[9] 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

[10] 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.