SENTIMENT ANALYSIS OF HOTEL REVIEW

TEAM MEMBER 1:
NAME: MOGALAPU JAYA SRIKAR
REG NUM: 20BCE2041

TEAM MEMBER 2:
NAME: AYUSHMAN KHUNTIA
REG NUM: 20BCE2615

TEAM MEMBER 3:
NAME: KUNAL SINGH
REG NUM: 20BCE0449

1. INTRODUCTION

1.1. OVERVIEW

In the era of digital communication and online platforms, hotel reviews have become a crucial source of information for potential travellers. These reviews, posted on various websites and social media platforms, offer valuable insights into the experiences and opinions of previous guests. However, analysing a large volume of reviews manually is a time-consuming task. This is where sentiment analysis comes into play. Sentiment analysis, also known as opinion mining, is a computational technique used to determine the sentiment or attitude expressed in a piece of text. It involves automatically classifying text as positive, negative, or neutral, based on the underlying emotions conveyed by the author. By applying sentiment analysis to hotel reviews, we can gain a deeper understanding of customers' sentiments and opinions towards different aspects of their stay, such as customer service, cleanliness, amenities, and overall satisfaction.

The primary goal of sentiment analysis in the context of hotel reviews is to extract meaningful insights from the vast amount of unstructured text data. These insights can help hotel management and staff in several ways. Firstly, they can identify areas that need improvement, enabling them to enhance the overall guest experience. By analysing sentiment patterns, they can pinpoint recurring issues, such as rude staff, outdated facilities, or poor cleanliness, and take appropriate actions to address them. Moreover, sentiment analysis can assist in reputation management for hotels. By monitoring and analysing sentiments expressed in reviews, hoteliers can gauge their brand reputation, identify influencers, and respond promptly to negative feedback. Positive sentiments can also be leveraged as testimonials and marketing material to attract potential customers.

1.2 PURPOSE

In this study, we aim to conduct sentiment analysis on a dataset of hotel reviews to gain insights into customers' perceptions and sentiments. By employing advanced natural language processing (NLP) techniques and machine learning algorithms, we will classify reviews as positive, negative, or neutral, and delve into the underlying factors that contribute to these sentiments. The findings of this analysis will be invaluable to hotel owners, managers, and marketers, enabling them to make data-driven decisions and enhance the overall guest experience.In summary, sentiment analysis of hotel reviews provides a powerful tool for extracting valuable insights from large volumes of unstructured text data. By understanding the sentiments and opinions expressed by customers, hoteliers can identify areas for improvement, manage their reputation, and enhance customer satisfaction. In the following sections, we will delve into the methodology, data collection, and analysis techniques used to conduct sentiment analysis on hotel reviews.

2. LITERATURE SURVEY

2.1. EXISTING METHODS OR APPROACHES TO SOLVE THE PROBLEM

Sentiment analysis has been carried out using a variety of techniques and strategies. The most prevalent approaches in the literature include hybrid approaches, lexicon-based approaches, and machine learning methods.

1. A multi task learning approach [14]
2. Multidimensional and multilevel modelling [13]
3. Hybrid model [12]
4. Using Artificial intelligence, Deep Convolutional Neural Networks [8]
5. Knowledge and aspect guided sentiment analysis [1]
6. Using Machine Learning, Deep Learning and Hidden Markov Model (HMM) [5]

Survey papers from journal	Our Paper
Sentiment Analysis and Opinion Mining: A Survey [59] Machine learning and deep learning algorithms are the foundation of all approaches. There isn't a particular strategy like multilevel or multidimensional modelling.	We reviewed publications that make use of diverse machine learning, deep learning, and modelling methodologies.
A Survey on Sentiment Analysis and Opinion Mining in Greek social media [60] The entire survey was conducted using datasets with Greek as the main language.	We looked at studies that included datasets in languages other than English, Spanish, and Arabic.
Sentiment Analysis: A Comparative Study on Different Approaches [61] The publications they consulted merely outlined alternative methodologies, but did not present their conclusions.	We included results for each paper in our survey paper. outcomes like accuracy and F1 score.

Wherever appropriate, important subjects and keywords have been clarified. Formulas contain explanations and are expressed in their mathematical scientific notation. To better comprehend the flow of the approaches used to grasp the feelings, the architecture of the proposed model is also presented using a visual representation. Each sentiment analysis technique is briefly explained. The contributions of each author were then thoroughly discussed. The descriptions of all the evaluation metrics have then been accompanied by their mathematical scientific representations, as mentioned in the selected articles, base papers, and other resources. The paper and the research are then concluded, along with some suggestions for how the subsequent publications might be better. This is followed by a conclusion and future work. The report concludes with references and citations for all the materials consulted and expertise incorporated into it..

2.2 PROPOSED SOLUTION

The proposed solution uses a neural network model to perform sentiment analysis on hotel reviews.

Import the necessary libraries: pandas, numpy, re, nltk, CountVectorizer from scikit-learn, Sequential and Dense from Keras, and pickle.

Download the stopwords corpus from NLTK using nltk.download('stopwords'). Stop words are common words like "the," "is," and "and" that are often removed during text preprocessing as they carry little meaning for sentiment analysis.

Read the training data into a pandas DataFrame using pd.read_csv(). The data should contain two columns which contain the text of the hotel reviews, which indicates the sentiment (positive or negative) of each review.

Preprocess the reviews:

- Create an empty list to store the preprocessed reviews.
- Iterate over each review in the training data.
- Use regular expressions (re.sub()) to remove non-alphabetic characters from the review text, replacing them with spaces.
- Convert the review to lowercase.
- Split the review into individual words.
- Initialize a PorterStemmer from NLTK for word stemming.
- Use list comprehension to iterate over the words, removing stop words using the English stopwords from NLTK, and performing stemming using the PorterStemmer.
- Join the preprocessed words back into a single string.
- Append the preprocessed review to the list.

Initialize a CountVectorizer object. CountVectorizer is used to convert the preprocessed text into a numerical representation that the neural network can understand.

Save the vocabulary of the CountVectorizer into a pickle file. This vocabulary will be useful later for converting new reviews into the same numerical representation.

Define a Sequential model from Keras, which represents a linear stack of neural network layers.

Add a Dense layers to the model

Add a final Dense layer with units set to 1 and the activation function set to 'sigmoid'. This layer outputs a single value between 0 and 1, representing the predicted sentiment probability (0 for negative sentiment, 1 for positive sentiment).

Compile the model using model.compile() with the Adam optimizer, binary cross-entropy loss function, and accuracy as the metric. The binary cross-entropy loss is commonly used for binary classification problems.

Train the model on the training data ,Set the batch size and the number of epochs. This step iteratively adjusts the model weights to minimize the loss and improve accuracy.

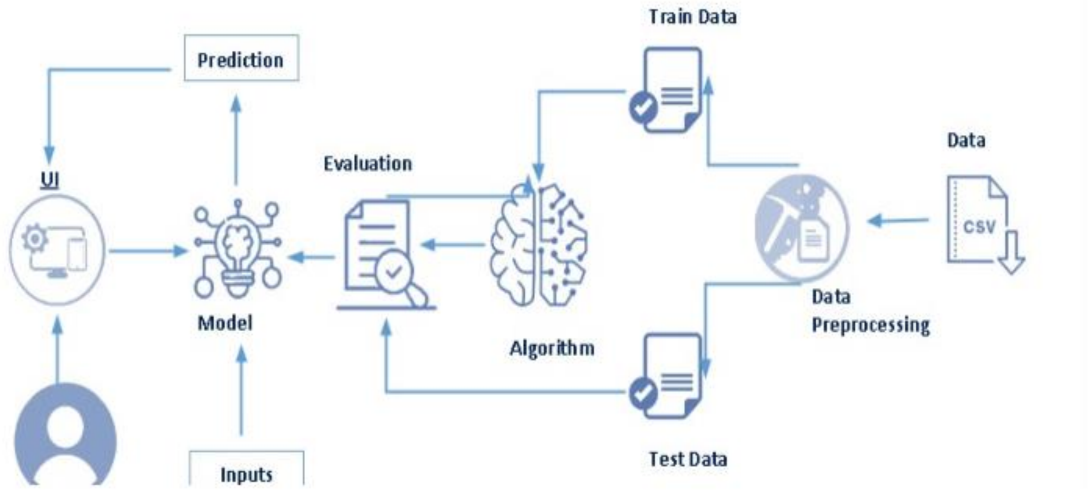
Finally Save the Model.

Integrate the Model with Flask.

Deploy the Model.

3. THEORETICAL ANALYSIS

3.1. BLOCK DIAGRAM



3.2. HARDWARE/SOFTWARE REQUIREMENTS

a) HARDWARE

i-7 Processor with 16 GB RAM

Monitors, Keyboards and Mouse

Laptops

Smartphones or Tablets

b) SOFTWARE

IDE : PYCHARM

IDE : VSCODE

PYTHON

GOOGLE COLLAB

FLASK

MS OFFICE Suite

Browser

System Modelling Tools: StarUML

4. EXPERIMENTAL INVESTIGATIONS

While developing a system for sentiment analysis of hotel reviews, a number of experimental studies were carried out to evaluate and enhance the model's performance.

Data analysis: Conduct exploratory data analysis (EDA) on the training data to learn more about how sentiments are distributed, how long reviews are, and other relevant details. Any data imbalances or irregularities that could affect model performance can be found using this technique.

Tuning the hyperparameters: Experiment with various values for the hidden unit count, activation function, learning rate, batch size, and number of epochs in the neural network model. To find the ideal hyperparameter values that boost the model's performance, use a grid search or random search.

Other than the CountVectorizer used in the corresponding code, other vectorization methods include TF-IDF, word embeddings (such as Word2Vec and GloVe), and transformer-based models (such as BERT). Evaluating the effectiveness of several vectorization techniques to choose the best one for sentiment analysis of hotel reviews.

Using relevant evaluation metrics like accuracy, precision, recall, F1 score, and ROC-AUC, evaluate the model's performance. To obtain more accurate performance estimations, do cross-validation or utilise a different validation set. The confusion matrix as well to comprehend the different kinds of mistakes the model can make (such false positives and false negatives).

Handling Class Imbalance: Use approaches like oversampling or undersampling to fix the problem if the sentiment classes in the training data are unbalanced. This can make the model more adept at handling various sentiment classes.

Fine-tuning Pretrained Models: Instead of starting from scratch when training a model, using BERT or other pretrained models. Because these models may capture contextual information, fine-tuning them on the hotel review dataset may result in improved performance.

Regularisation strategies: Use regularisation strategies to reduce overfitting and enhance generalisation performance, such as L1 or L2 regularisation, dropout, or early stopping.

Error Analysis: Examine the samples that were incorrectly classified to learn more about the model's shortcomings and potential areas for development. Enhancing the training data or using particular data preprocessing techniques to address recurring patterns or difficult scenarios where the model fails.

Test on Unseen Data: Assess the trained model using test data that were not included in the training set, such as actual customer reviews of hotels. In doing so, the performance and generalisation capacities of the model will be more accurately evaluated.

Investigate methods for analysing the model's predictions and identifying the essential characteristics or words that influence positive or negative sentiments. This can reveal important details about the way the model makes decisions and make it easier to spot any biases or problems.

Model Comparison: Compare the performance of the neural network model with other machine learning algorithms such as support vector machines (SVM), random forests, or gradient boosting models. Evaluate their performance using appropriate metrics and determine which approach yields the best results.

Analysis of the training and validation accuracy and loss data during the training process will let us evaluate the model's performance. Any overfitting or underfitting problems can be found by plotting these measures across epochs.

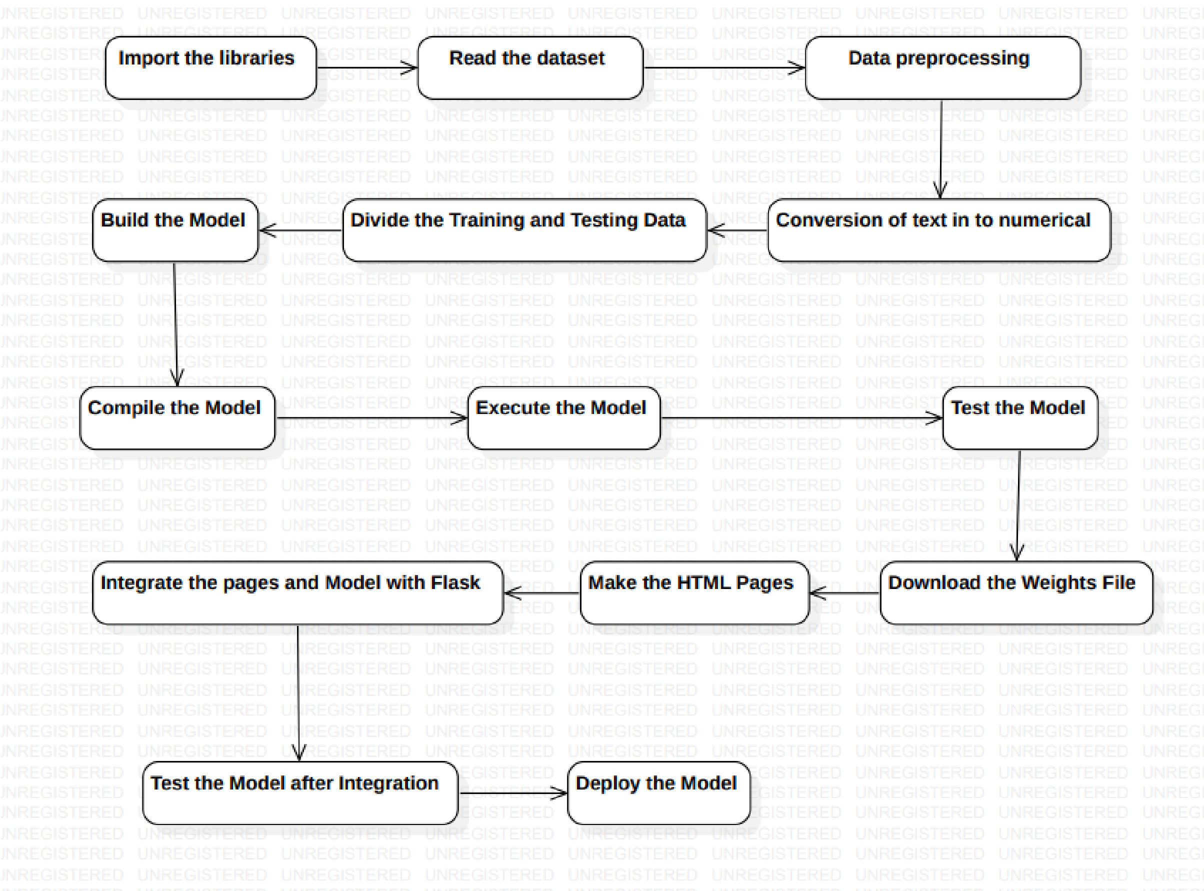
Cross-Validation: Use cross-validation to evaluate how well the model performs on various subsets of the training data. This can give a more reliable estimate of the model's capacity for generalisation.

Plot learning curves to examine the performance of the model as the volume of training data rises. This can assist identify whether the model would benefit from additional training examples or if its performance has reached saturation.

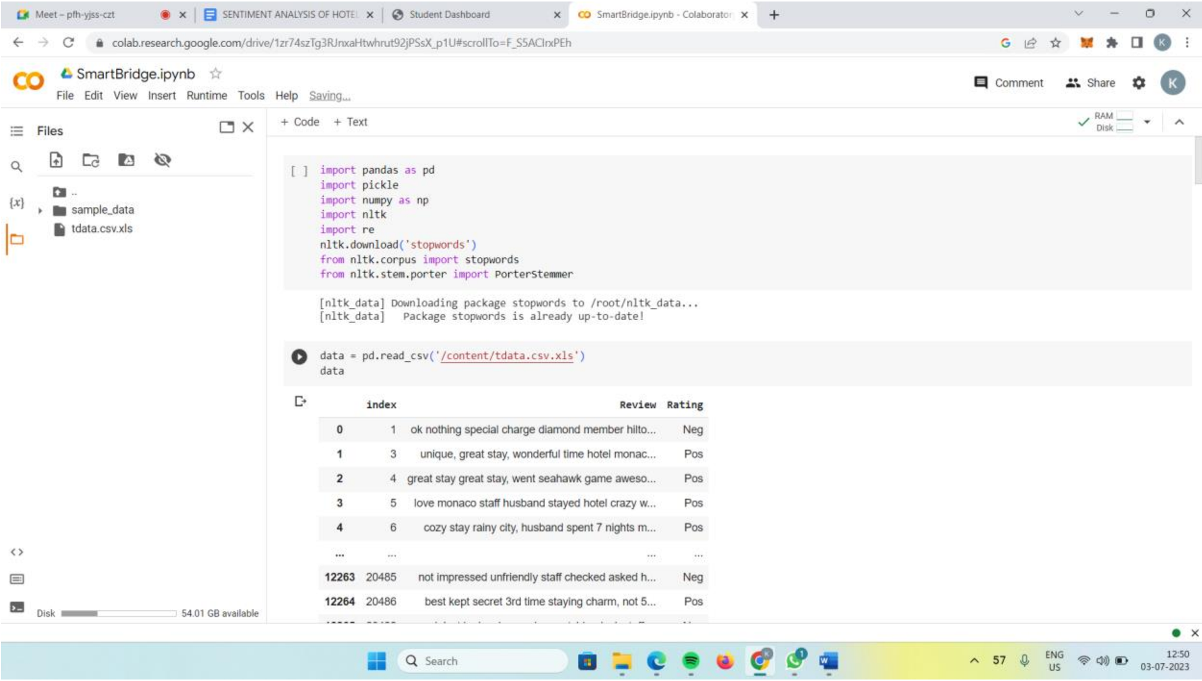
Investigate the significance of various attributes or words in predicting sentiment. Analysing the neural network model's weights can be used.

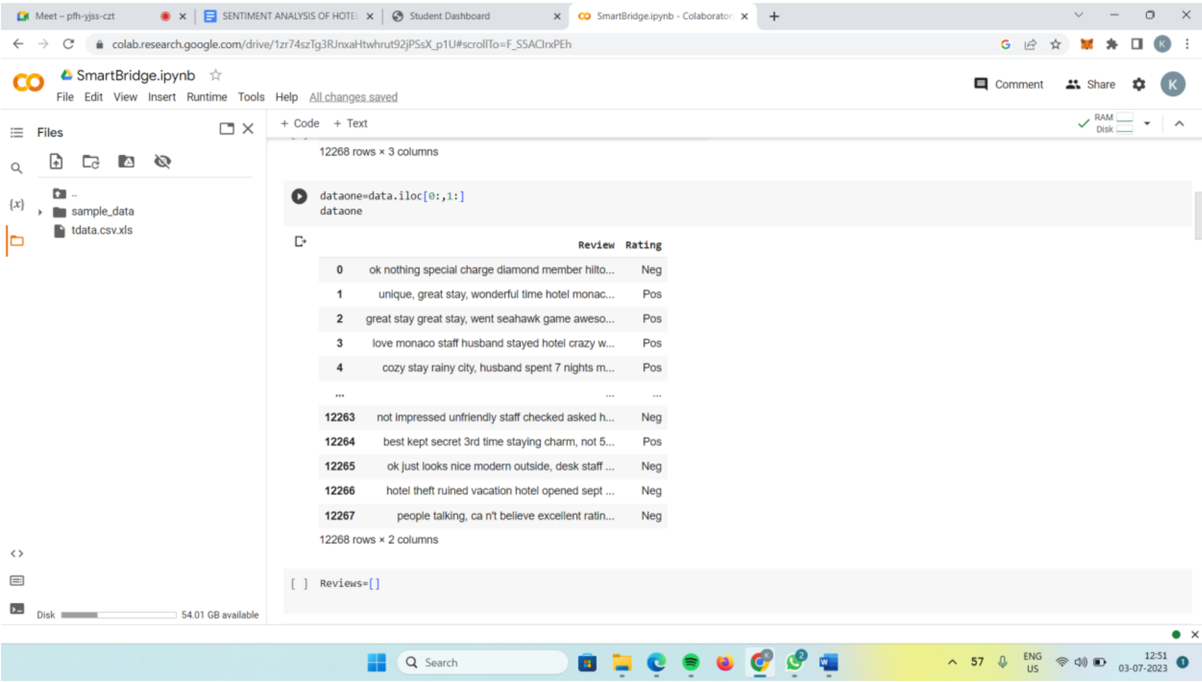
These experimental studies and analyses can offer insightful information on the effectiveness of the sentiment analysis solution and serve as a path for future developments that will increase its accuracy and robustness.

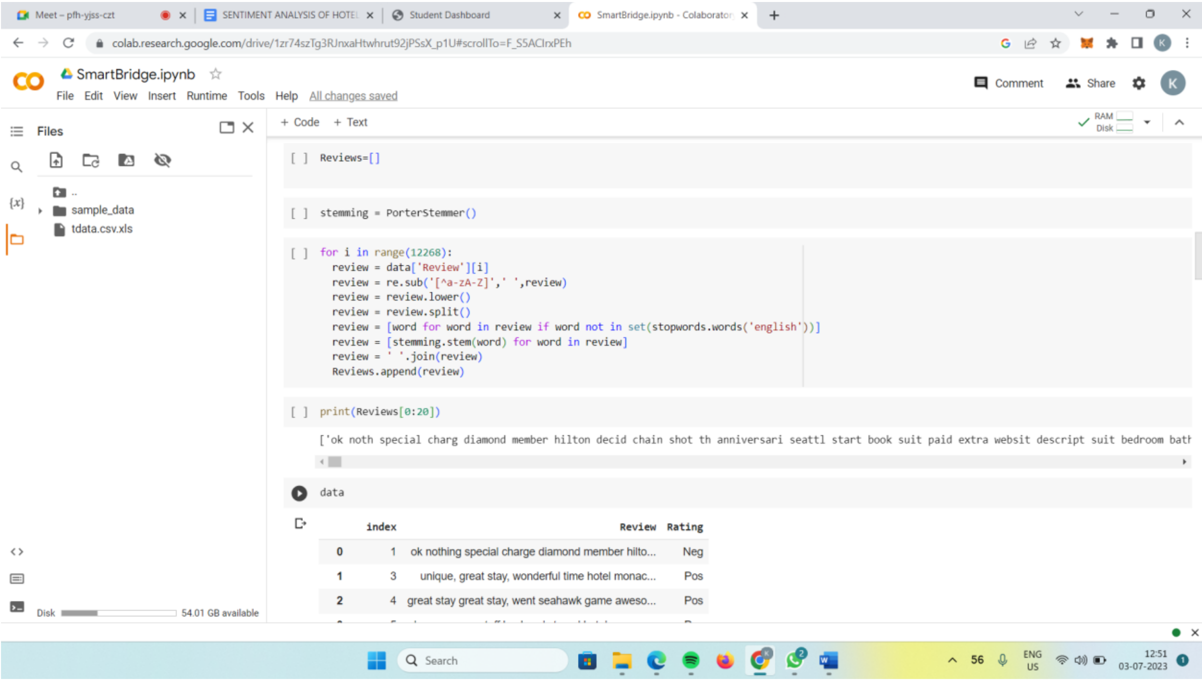
5. FLOWCHART

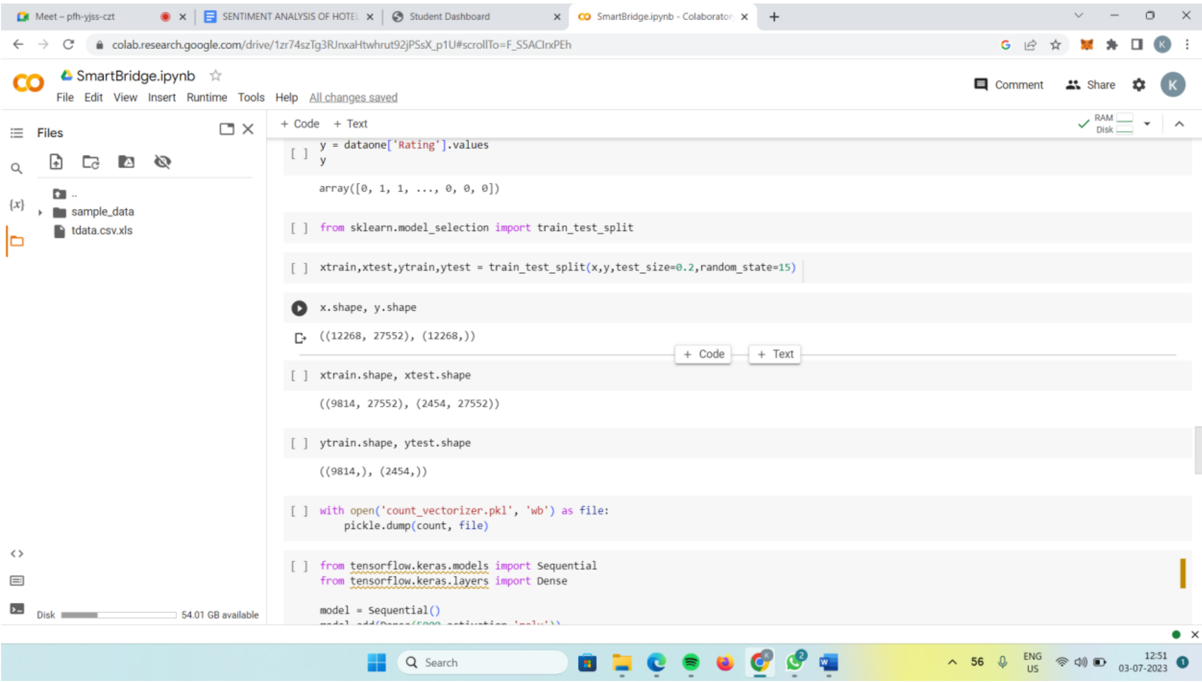


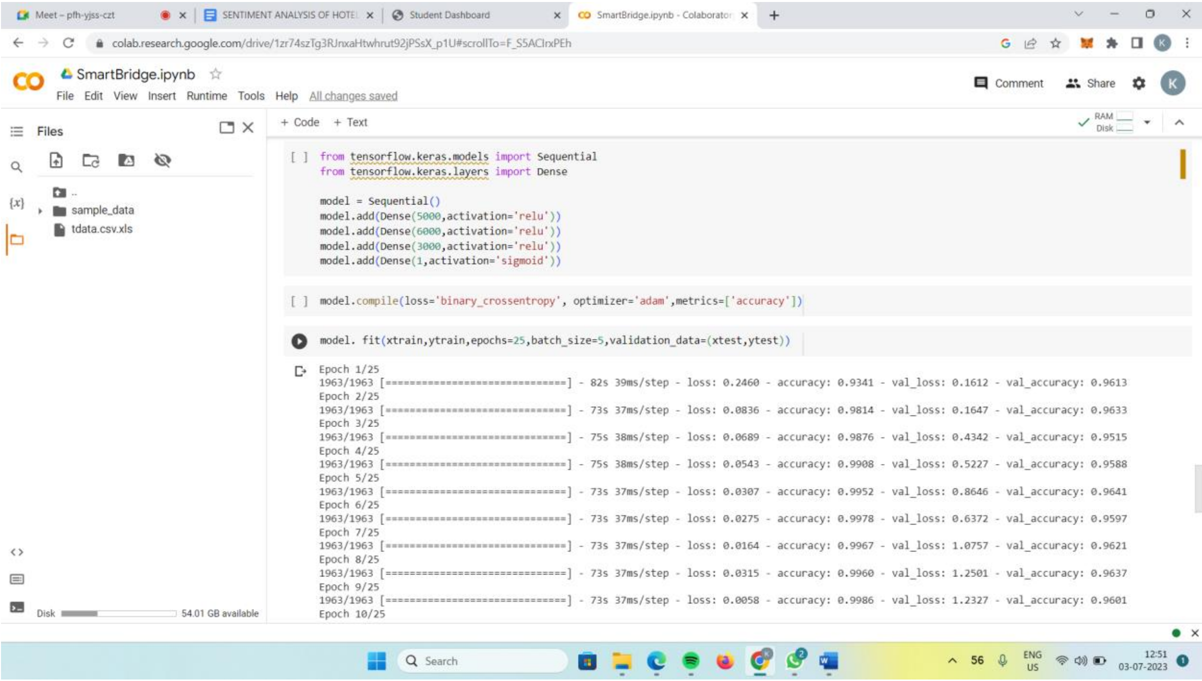
6. RESULT AND SCREENSHOTS

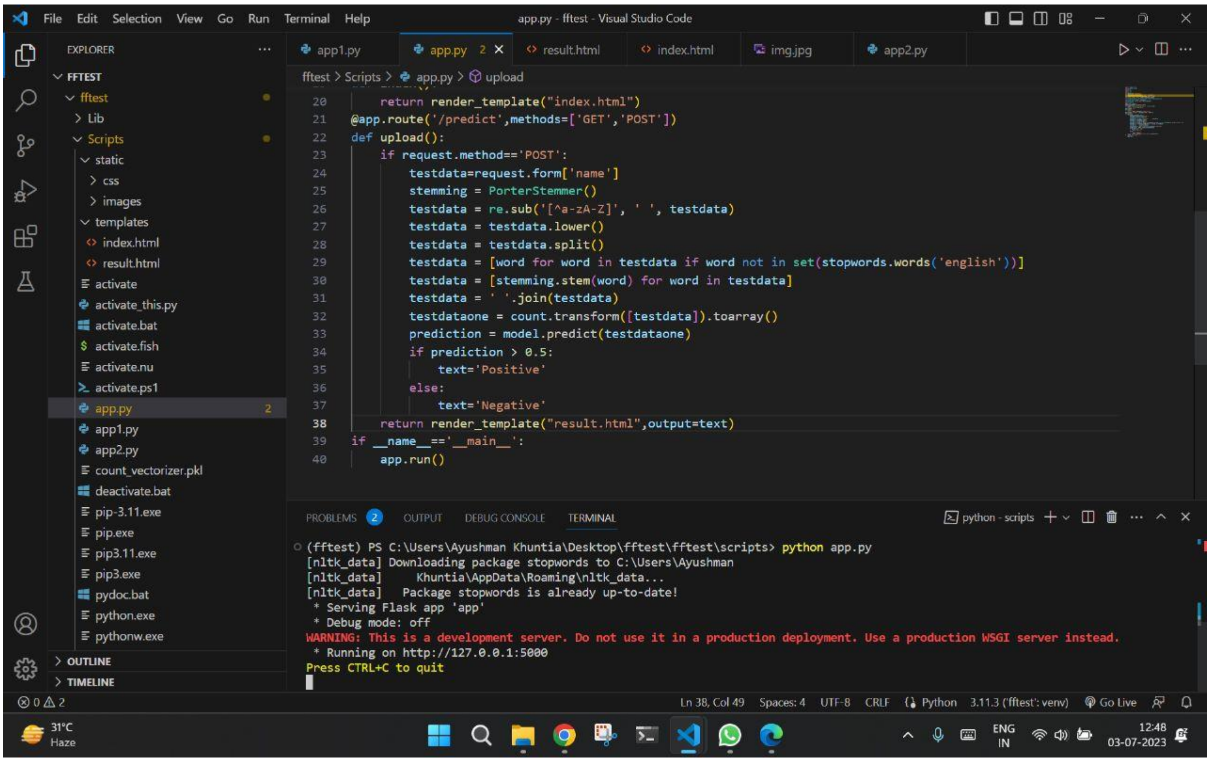


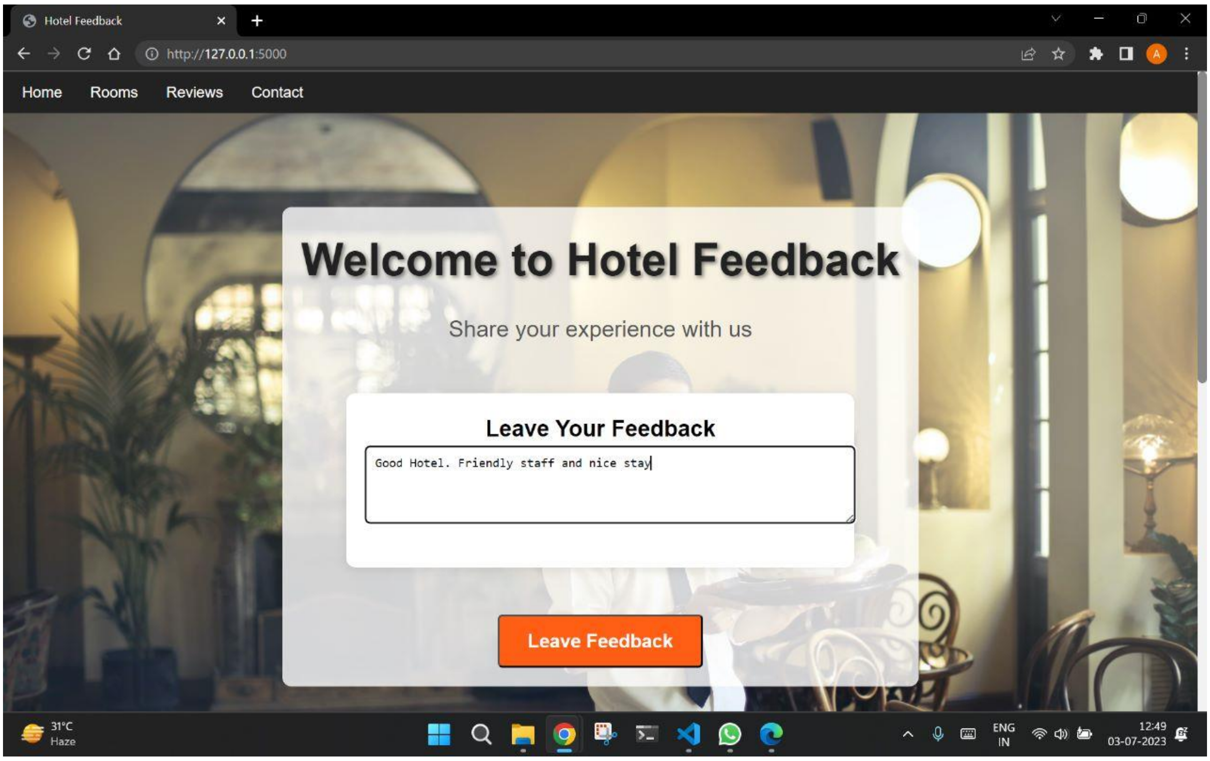


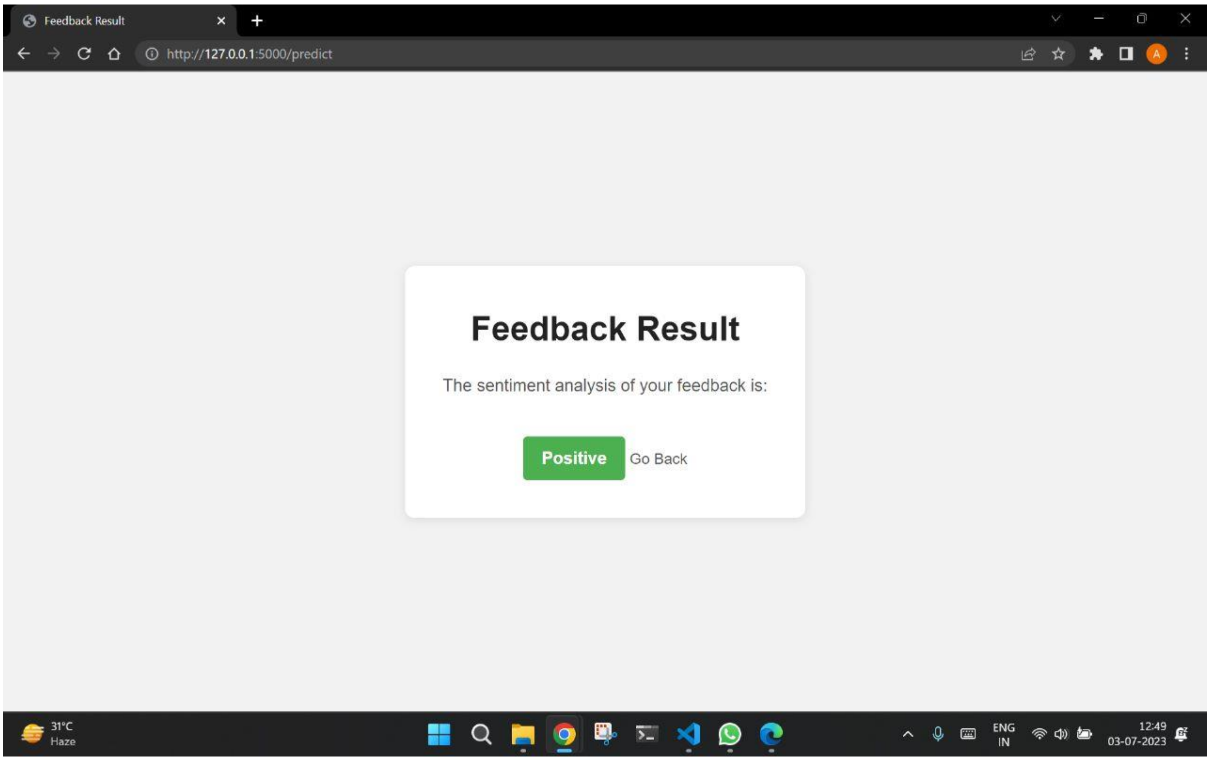

SmartBridge.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Reviews=[]
stemming = PorterStemmer()
for i in range(12268):
    review = data['Review'][i]
    review = re.sub('[^a-zA-Z]',' ',review)
    review = review.lower()
    review = review.split()
    review = [word for word in review if word not in set(stopwords.words('english'))]
    review = [stemming.stem(word) for word in review]
    review = ' '.join(review)
    Reviews.append(review)
print(Reviews[0:20])
['ok noth special charg diamond member hilton decid chain shot th anniversari seattl start book suit paid extra websit descript suit bedroom bath
data
index Review Rating
0 1 ok nothing special charge diamond member hilto... Neg
1 3 unique, great stay, wonderful time hotel monac... Pos
2 4 great stay great stay, went seahawk game aweso... Pos

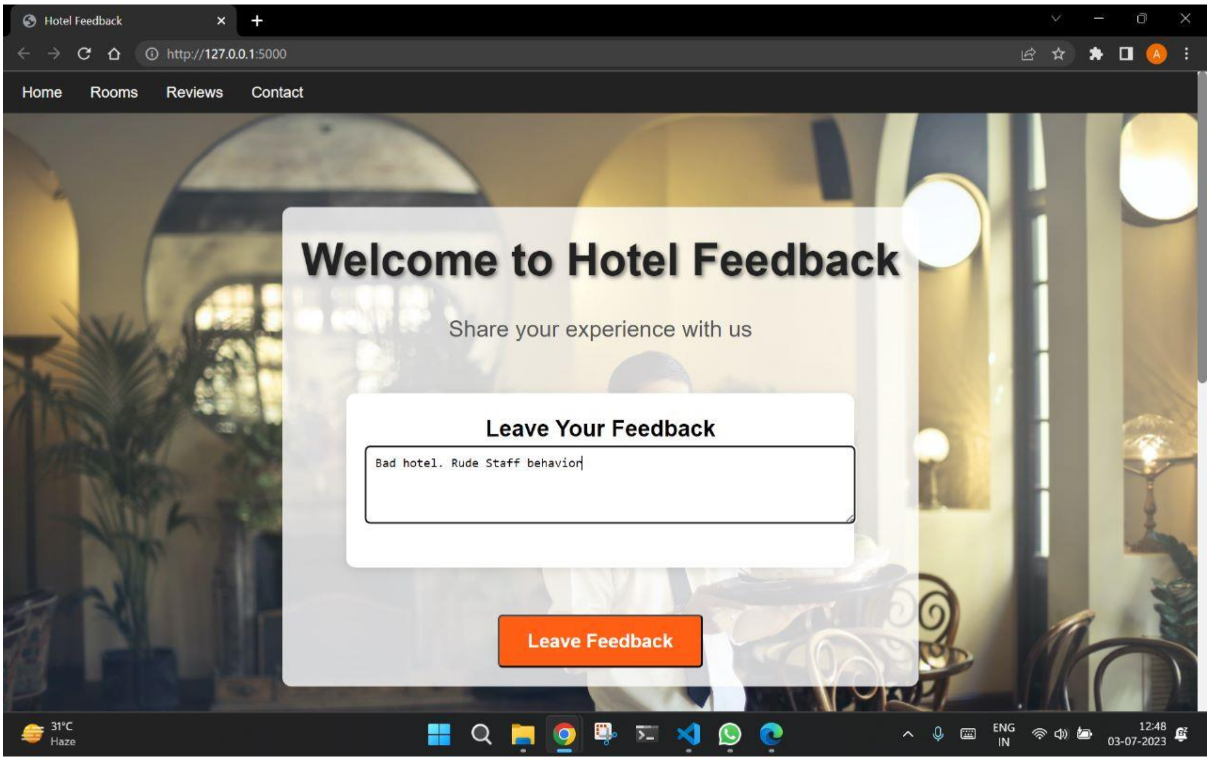

SmartBridge.ipynb
File Edit View Insert Runtime Tools Help All changes saved
y = dataone['Rating'].values
y
array([0, 1, 1, ..., 0, 0, 0])
from sklearn.model_selection import train_test_split
xtrain,xtest,ytrain,ytest = train_test_split(x,y,test_size=0.2,random_state=15)
x.shape, y.shape
((12268, 27552), (12268,))
xtrain.shape, xtest.shape
((9814, 27552), (2454, 27552))
ytrain.shape, ytest.shape
((9814,), (2454,))
with open('count_vectorizer.pkl', 'wb') as file:
    pickle.dump(count, file)
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense
model = Sequential()

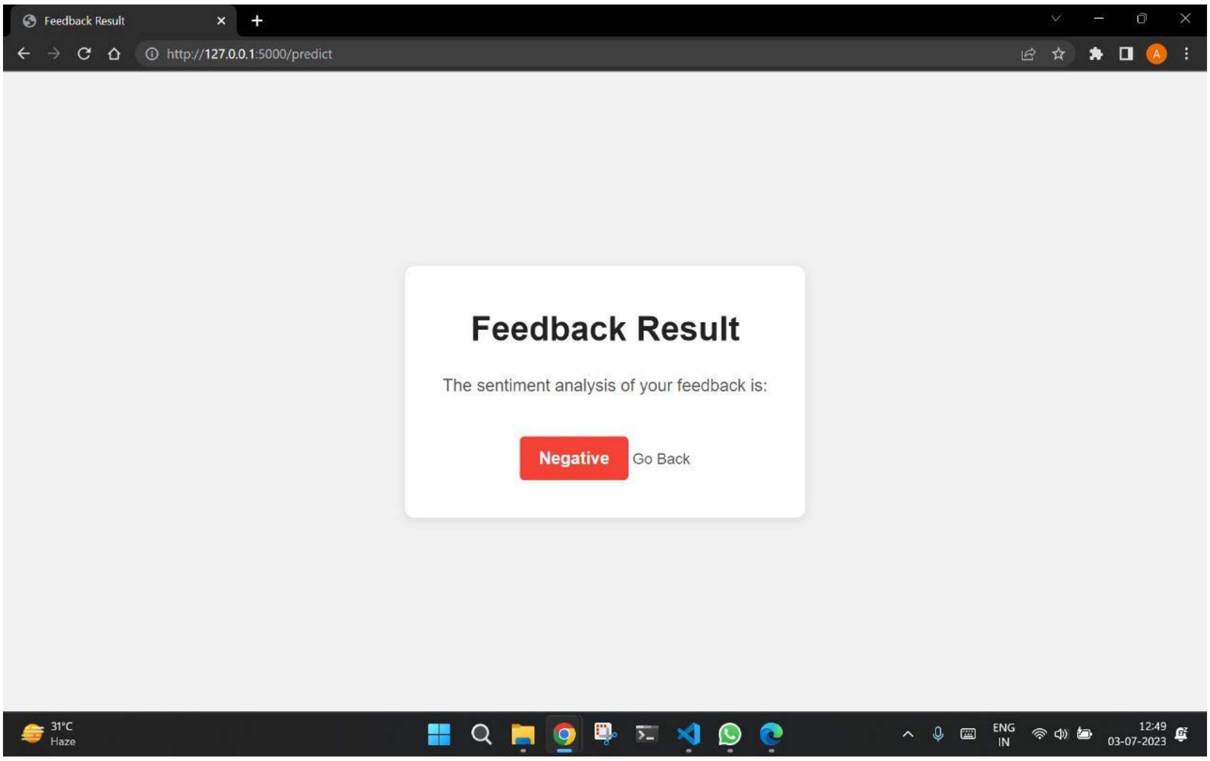

SmartBridge.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Files
sample_data
tdata.csv.xls
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense
model = Sequential()
model.add(Dense(5000,activation='relu'))
model.add(Dense(6000,activation='relu'))
model.add(Dense(3000,activation='relu'))
model.add(Dense(1,activation='sigmoid'))
model.compile(loss='binary_crossentropy', optimizer='adam',metrics=['accuracy'])
model. fit(xtrain,ytrain,epochs=25,batch_size=5,validation_data=(xtest,ytest))
Epoch 1/25
1963/1963 [==============================] - 82s 39ms/step - loss: 0.2460 - accuracy: 0.9341 - val_loss: 0.1612 - val_accuracy: 0.9613
Epoch 2/25
1963/1963 [==============================] - 73s 37ms/step - loss: 0.0836 - accuracy: 0.9814 - val_loss: 0.1647 - val_accuracy: 0.9633
Epoch 3/25
1963/1963 [==============================] - 75s 38ms/step - loss: 0.0689 - accuracy: 0.9876 - val_loss: 0.4342 - val_accuracy: 0.9515
Epoch 4/25
1963/1963 [==============================] - 75s 38ms/step - loss: 0.0543 - accuracy: 0.9908 - val_loss: 0.5227 - val_accuracy: 0.9588
Epoch 5/25
1963/1963 [==============================] - 73s 37ms/step - loss: 0.0307 - accuracy: 0.9952 - val_loss: 0.8646 - val_accuracy: 0.9641
Epoch 6/25
1963/1963 [==============================] - 73s 37ms/step - loss: 0.0275 - accuracy: 0.9978 - val_loss: 0.6372 - val_accuracy: 0.9597
Epoch 7/25
1963/1963 [==============================] - 73s 37ms/step - loss: 0.0164 - accuracy: 0.9967 - val_loss: 1.0757 - val_accuracy: 0.9621
Epoch 8/25
1963/1963 [==============================] - 73s 37ms/step - loss: 0.0315 - accuracy: 0.9960 - val_loss: 1.2501 - val_accuracy: 0.9637
Epoch 9/25
1963/1963 [==============================] - 73s 37ms/step - loss: 0.0058 - accuracy: 0.9986 - val_loss: 1.2327 - val_accuracy: 0.9601
Epoch 10/25

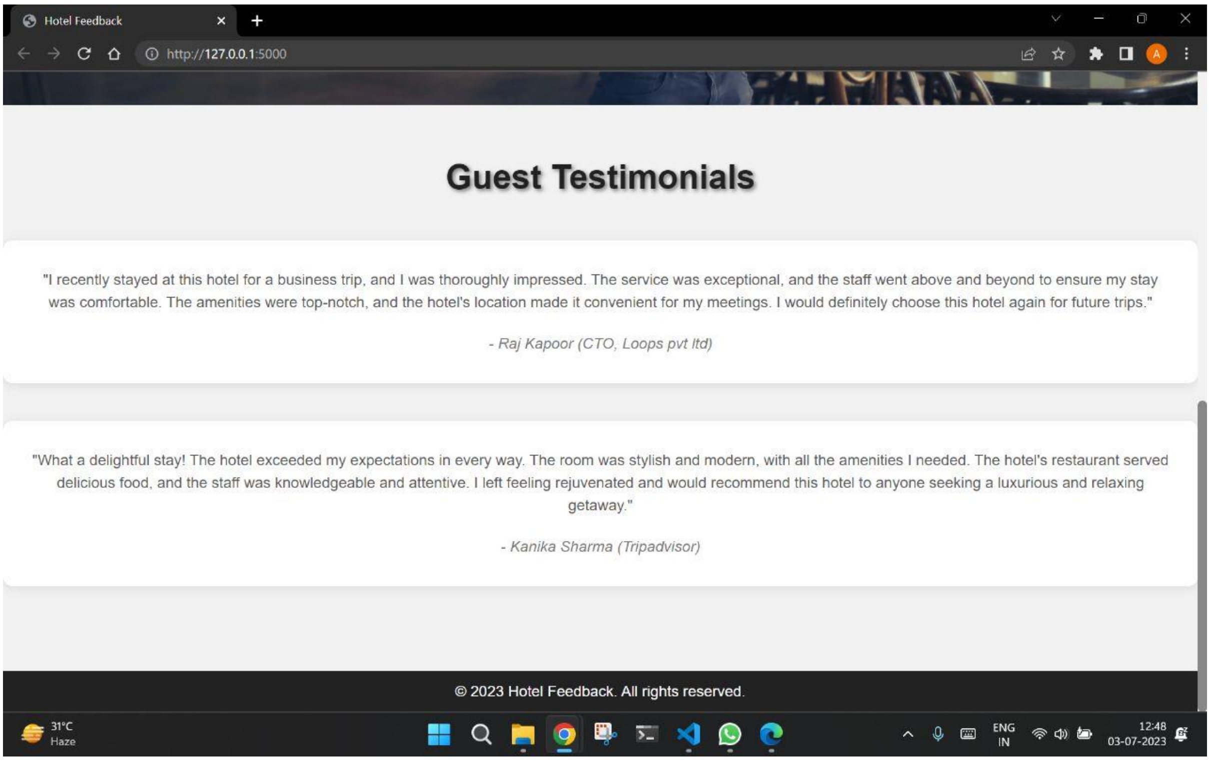

app.py - fftest - Visual Studio Code
app1.py app.py result.html index.html img.jpg app2.py
fftest > Scripts > app.py > upload
return render_template("index.html")
@app.route('/predict',methods=['GET','POST'])
def upload():
if request.method=='POST':
testdata=request.form['name']
stemming = PorterStemmer()
testdata = re.sub('[^a-zA-Z]', ' ', testdata)
testdata = testdata.lower()
testdata = testdata.split()
testdata = [word for word in testdata if word not in set(stopwords.words('english'))]
testdata = [stemming.stem(word) for word in testdata]
testdata = ' '.join(testdata)
testdataone = count.transform([testdata]).toarray()
prediction = model.predict(testdataone)
if prediction > 0.5:
text='Positive'
else:
text='Negative'
return render_template("result.html",output=text)
if __name__=='__main__':
app.run()
(fftest) PS C:\Users\Ayushman Khuntia\Desktop\fftest\fftest\scripts> python app.py
[nltk_data] Downloading package stopwords to C:\Users\Ayushman
[nltk_data]   Khuntia\AppData\Roaming\nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
* Serving Flask app 'app'
* Debug mode: off
WARNING: This is a development server. Do not use it in a production deployment. Use a production WSGI server instead.
* Running on http://127.0.0.1:5000
Press CTRL+C to quit


Hotel Feedback
http://127.0.0.1:5000
Home
Rooms
Reviews
Contact
Welcome to Hotel Feedback
Share your experience with us
Leave Your Feedback
Good Hotel. Friendly staff and nice stay
Leave Feedback
31°C
Haze
ENG
IN
12:49
03-07-2023


Feedback Result
http://127.0.0.1:5000/predict
Feedback Result
The sentiment analysis of your feedback is:
Positive
Go Back
31°C
Haze
ENG
IN
12:49
03-07-2023


Hotel Feedback
http://127.0.0.1:5000
Home
Rooms
Reviews
Contact
Welcome to Hotel Feedback
Share your experience with us
Leave Your Feedback
Bad hotel. Rude Staff behavior
Leave Feedback
31°C
Haze
ENG
IN
12:48
03-07-2023


Feedback Result
http://127.0.0.1:5000/predict
Feedback Result
The sentiment analysis of your feedback is:
Negative
Go Back
31°C
Haze
ENG
IN
12:49
03-07-2023



7. ADVANTAGES & DISADVANTAGES

Advantages of Sentiment Analysis of Hotel Reviews using Deep Learning and Natural Language Processing:

1. **Accuracy**: Deep learning models, such as Recurrent Neural Networks (RNNs) and Convolutional Neural Networks (CNNs), coupled with advanced Natural Language Processing (NLP) techniques, have shown significant improvements in sentiment analysis accuracy. They can capture complex linguistic patterns and contextual information, leading to more precise sentiment classification.

2. **Handling Unstructured Data**: Hotel reviews often contain unstructured text data with varying lengths, sentence structures, and language nuances. Deep learning models excel at handling such unstructured data and can effectively capture the semantics and sentiments expressed in the reviews.

3. **Feature Learning**: Deep learning models can automatically learn relevant features from raw text data without relying on handcrafted feature engineering. This eliminates the need for manual feature extraction, making the sentiment analysis process more efficient.

4. **Contextual Understanding**: Deep learning models can capture the contextual understanding of the text, including sarcasm, irony, and sentiment shift within a sentence or review. This enables more accurate sentiment analysis in situations where the sentiment may not be explicitly stated.

5. **Generalisation**: Deep learning models can generalise well to new and unseen hotel reviews. Once trained on a large dataset, they can adapt to different review styles, languages, and domains, making them versatile for sentiment analysis tasks.

Disadvantages of Sentiment Analysis of Hotel Reviews using Deep Learning and Natural Language Processing:

1. **Data Requirements**: Deep learning models for sentiment analysis typically require a large amount of labelled training data to perform optimally. Collecting and labelling a significant volume of hotel review data can be time-consuming and expensive.

2. **Computational Resources**: Training deep learning models for sentiment analysis requires significant computational resources, including high-performance GPUs or TPUs. Running such models on resource-constrained devices or systems can be challenging.

3. **Interpretability:** Deep learning models are often referred to as "black boxes" because they lack interpretability. Understanding how and why a particular sentiment classification was made can be difficult. This lack of interpretability may limit the ability to explain and validate the sentiment analysis results.

4. **Overfitting:** Deep learning models, especially when dealing with limited training data, are prone to overfitting. Overfitting occurs when the model memorises the training data instead of learning generalizable patterns, leading to poor performance on unseen data.

5. **Domain Specificity:** Deep learning models trained on a specific domain (e.g., hotel reviews) may not generalise well to other domains without additional fine-tuning or retraining. Adapting the models to different domains or industries may require additional labelled data and model adjustments.

8. APPLICATIONS

Applications of Sentiment Analysis of Hotel Reviews using Deep Learning and Natural Language Processing:

1. **Customer Experience Improvement:** Sentiment analysis can help hotel owners and managers identify areas of improvement in their establishments. By analysing sentiments expressed in hotel reviews, they can identify recurring issues and address them promptly. This includes aspects such as customer service, cleanliness, amenities, and overall guest satisfaction.

2. **Reputation Management:** Sentiment analysis enables hoteliers to monitor and manage their online reputation effectively. By analysing sentiments in hotel reviews, they can identify trends, influencers, and potential issues that might impact their brand reputation. Promptly addressing negative feedback and leveraging positive sentiments can help maintain a positive online image.

3. **Competitor Analysis:** Sentiment analysis can be used to analyse the sentiments expressed in hotel reviews of competitors. This provides insights into their strengths and weaknesses, allowing hoteliers to benchmark their own performance and make strategic decisions to stay competitive in the market.

4. **Pricing and Revenue Management**: Sentiment analysis can help hotels understand the correlation between sentiment and pricing. By analysing sentiments expressed in reviews alongside pricing information, hotels can identify the price points at which guests express the highest levels of satisfaction. This information can guide pricing strategies and revenue management decisions.

5. **Marketing and Advertising:** Positive sentiments expressed in hotel reviews can be leveraged as testimonials and marketing material. Hoteliers can identify positive sentiment trends and specific

aspects that guests appreciate the most, enabling them to create targeted marketing campaigns and highlight their unique selling points.

6. **Customer Segmentation:** Sentiment analysis can assist in segmenting customers based on their sentiments and preferences. By grouping customers with similar sentiments and preferences together, hotels can offer personalised experiences and tailor their services to meet specific customer needs.

7. **Predictive Analytics:** By analysing historical sentiment data, deep learning models can be trained to make predictions about future customer sentiments. This can help hotels anticipate customer needs and preferences, enabling them to proactively enhance their services and offerings.

8. **Feedback Analysis:** Sentiment analysis can automate the process of analysing customer feedback. By categorising and summarising sentiments expressed in reviews, hoteliers can gain a comprehensive understanding of customer opinions, identify emerging trends, and take proactive actions to address concerns or capitalise on opportunities.

9. **CONCLUSION**

In conclusion, sentiment analysis of hotel reviews using deep learning and natural language processing techniques offers significant advantages in understanding and extracting insights from large volumes of unstructured text data. By applying advanced machine learning models and NLP algorithms, hotel owners, managers, and marketers can gain valuable insights into customer sentiments and opinions. Through sentiment analysis, hotels can identify areas for improvement and enhance the overall guest experience. By analysing sentiments expressed in reviews, they can pinpoint recurring issues, such as poor customer service, cleanliness concerns, or outdated facilities, and take proactive actions to address them. This helps in enhancing customer satisfaction and loyalty.

Moreover, sentiment analysis aids in reputation management for hotels. By monitoring sentiments expressed in reviews, hoteliers can gauge their brand reputation, identify influencers, and promptly respond to negative feedback. Positive sentiments can also be leveraged as testimonials and marketing material to attract potential customers. The combination of deep learning and NLP techniques in sentiment analysis allows for accurate sentiment classification and contextual understanding. Deep learning models can capture complex linguistic patterns and contextual information, enabling them to analyse sentiments even in situations where the sentiment may not be explicitly stated, such as sarcasm or irony.

However, it is important to consider the challenges associated with sentiment analysis using deep learning and NLP, such as the need for large labelled datasets, computational resources, and the lack of interpretability in deep learning models. Overfitting and domain specificity are also factors to be addressed when implementing sentiment analysis in the hotel industry. Despite these challenges, sentiment analysis using deep learning and NLP provides a powerful tool for extracting meaningful insights from hotel reviews. By understanding customer sentiments and opinions, hotels can make data-driven decisions, enhance their services, and maintain a positive brand image in a highly competitive industry. Overall, sentiment analysis using deep learning and NLP has the potential to revolutionise the way hotels analyse and utilise guest feedback. It enables hotels to stay ahead of the curve, adapt to customer preferences, and continuously improve their offerings to deliver exceptional guest experiences.

10. **FUTURE SCOPE**

The future scope of hotel reviews using deep learning and natural language processing (NLP) is vast and holds tremendous potential for the hotel industry. Here are some key areas where advancements and innovations can be expected:

1. **Multilingual Sentiment Analysis**: As the global tourism industry continues to grow, there is a need for sentiment analysis models that can handle multiple languages. Advancements in deep learning and NLP can facilitate the development of multilingual sentiment analysis models capable of processing and analysing hotel reviews in various languages.

2. **Integration with Voice and Multimedia Data**: With the rise of voice assistants and multimedia content, the future scope of hotel review analysis includes integrating sentiment analysis with voice data and analysing sentiment in images or videos. This can enable a more comprehensive understanding of customer sentiments and preferences.

3. **Predictive Analytics and Personalization:** By leveraging historical sentiment data, deep learning models can be trained to make predictions about future customer sentiments. This can enable hotels to personalise their services, anticipate customer needs, and tailor their offerings to individual preferences.

11. **REFERENCES**

1. Z. Ke, J. Sheng, Z. Li, W. Silamu and Q. Guo, "Knowledge-Guided Sentiment Analysis Via Learning From Natural Language Explanations," in IEEE Access, vol. 9, pp. 3570-3578, 2021, doi: 10.1109/ACCESS.2020.3048088.

2. S. Al-Azani and E. -S. M. El-Alfy, "Early and Late Fusion of Emojis and Text to Enhance Opinion Mining," in IEEE Access, vol. 9, pp. 121031-121045, 2021, doi: 10.1109/ACCESS.2021.3108502.

3. P. K. Singh and S. Paul, "Deep Learning Approach for Negation Handling in Sentiment Analysis," in IEEE Access, vol. 9, pp. 102579-102592, 2021, doi: 10.1109/ACCESS.2021.3095412.

4. A. Boumhidi, A. Benlahbib and E. H. Nfaoui, "Cross-Platform Reputation Generation System Based on Aspect-Based Sentiment Analysis," in IEEE Access, vol. 10, pp. 2515-2531, 2022, doi: 10.1109/ACCESS.2021.3139956.

5. Isidoros Perikos, Spyridon Kardakis, Ioannis Hatzilygeroudis, Sentiment analysis using novel and interpretable architectures of Hidden Markov Models, Knowledge-Based Systems, Volume 229,2021,107332,ISSN 0950-7051,https://doi.org/10.1016/j.knosys.2021.107332.

6. Sandra Jardim, Carlos Mora,Customer reviews sentiment-based analysis and clustering for market-oriented tourism services and products development or positioning,Procedia Computer Science,Volume 196,2022,Pages 199-206,ISSN 1877-0509,https://doi.org/10.1016/j.procs.2021.12.006.

7. 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. 8, pp. 23522-23530, 2020, doi: 10.1109/ACCESS.2020.2969854.

8. F. Alattar and K. Shaalan, "Using Artificial Intelligence to Understand What Causes Sentiment Changes on Social Media," in IEEE Access, vol. 9, pp. 61756-61767, 2021, doi: 10.1109/ACCESS.2021.3073657.

9. Z. Jianqiang, G. Xiaolin and Z. Xuejun, "Deep Convolution Neural Networks for Twitter Sentiment Analysis," in IEEE Access, vol. 6, pp. 23253-23260, 2018, doi: 10.1109/ACCESS.2017.2776930.

10. I. Gupta and N. Joshi, "Feature-Based Twitter Sentiment Analysis With Improved Negation Handling," in IEEE Transactions on Computational Social Systems, vol. 8, no. 4, pp. 917-927, Aug. 2021, doi: 10.1109/TCSS.2021.3069413.

11. Chuchu Liu, Fan Fang, Xu Lin, Tie Cai, Xu Tan, Jianguo Liu, Xin Lu, Improving sentiment analysis accuracy with emoji embedding,Journal of Safety Science and Resilience,Volume 2, Issue 4,2021,Pages 246-252,ISSN 2666-4496,https://doi.org/10.1016/j.jnlssr.2021.10.003.

12. Y. Feng and Y. Cheng, "Short Text Sentiment Analysis Based on Multi-Channel CNN With Multi-Head Attention Mechanism," in IEEE Access, vol. 9, pp. 19854-19863, 2021, doi: 10.1109/ACCESS.2021.3054521.

13. B. Wang, D. Shan, A. Fan, L. Liu and J. Gao, "A Sentiment Classification Method of Web Social Media Based on Multidimensional and Multilevel Modeling," in IEEE Transactions on Industrial Informatics, vol. 18, no. 2, pp. 1240-1249, Feb. 2022, doi: 10.1109/TII.2021.3085663.

14. F. M. Plaza-Del-Arco, M. D. Molina-González, L. A. Ureña-López and M. T. Martín-Valdivia, "A Multi-Task Learning Approach to Hate Speech Detection Leveraging Sentiment Analysis," in IEEE Access, vol. 9, pp. 112478-112489, 2021, doi: 10.1109/ACCESS.2021.3103697.

15. Md. Shahriare Satu, Md. Imran Khan, Mufti Mahmud, Shahadat Uddin,Matthew A. Summers, Julian M.W. Quinn, Mohammad Ali Moni,TClustVID: A novel machine learning classification model to investigate topics and sentiment in COVID-19 tweets,Knowledge-Based Systems,Volume 226,2021,107126,ISSN 0950-7051

16. E. Martínez-Cámara, Y. Almeida-Cruz, M. C. D. Galiano, S. Estévez-Velarde, M. Á. G. Cumbreras, M. G. Vega, and Y. Gutiérrez, ‘‘Overview of TASS 2018: Opinions, health, and emotions,’’ in Proc. TASS 2018: Workshop Semantic Anal. at SEPLN, TASS@SEPLN 2018, vol. 2172, 2018, pp. 13–27.

17. M. C. Díaz-Galiano, M. G. Vega, and E. Casasola, ‘‘Overview of TASS 2019: One further for the global Spanish sentiment analysis corpus,’’ in Proc. IberLEF@ SEPLN, 2019, pp. 550–560. 18

18. F. M. Plaza-del-Arco, C. Strapparava, L. A. Ureña-López, and M. T. Martín-Valdivia, ‘‘EmoEvent: A multilingual emotion corpus based on different events,’’ in Proc. 12th Lang. Resour. Eval. Conf., Marseille, France, May 2020, pp. 1492–1498

19. Marseille, France, May 2020, pp. 1492–1498. [46] V. Basile, C. Bosco, E. Fersini, D. Nozza, V. Patti, F. M. Rangel Pardo, P. Rosso, and M. Sanguinetti, ‘‘SemEval-2019 task 5: Multilingual detection of hate speech against immigrants and women in Twitter,’’ in Proc. 13th Int. Workshop Semantic Eval., Minneapolis, MN, USA, 2019, pp. 54–63, doi: 10.18653/v1/S19-2007

20. github: SophonPlus/ChineseNlpCorpus/blob/master/datasets/weibo_senti_100k/intro.ipynb

21. 21. Jinhuakst (2018), *ChineseNLPCorpus*, GitHub.

22. 22. Imtiaz, N., Middleton, J., Chakraborty, J., Robson, N., Bai, G., Murphy-Hill, E. (2019), *Investigating the Effects of Gender Bias on GitHub*, IEEE

23. 23. Kanbayashi., Y., Sakaguchi, K., Hongo, F., Ishikawa, T., Tabuchi, Y., Osamu, U., Takayama, K., Taguchi, T. (2021), *Predictors for development of denosumab-induced hypoalcaemia in cancer patients with bone metastases determined by ordered logistic regression analysis*, Springer Nature.

24. 24. Rish, I. (2001), *An Empirical study of the naïve Bayes Classifier,* T.J. Watson Research Center.

25. 25. Rao, H., Shi, X., Rodrigue, A. K., Feng, J., Xia, Y., Elhoseny, M., Yuan, X., Gu, L. (2019), *Feature Selection based on Artificial bee colony and gradient boosting decision tree*, Elsevier B.V.

26. 25. Yadav, A. Vishwakarma, D. K. (2019), *Sentiment analysis using deep learning architectures: a Review*, Springer Nature.

27. 26. Devlin J, Chang M W, Lee K, et al. (2018), *Bert: Pre-training of deep bidirectional transformers for language understanding.*

28. 27. He, Y., Sun, S., Niu, F., Li, F. (2017) ,*A deep learning model of emotional semantic enhancement for Weibo sentiment analysis*, Chin. J. Compute., vol. 40, no. 4, pp. 773–790, 2017.

29. 28. Z. Yang, D. Yang, C. Dyer, X. He, A. Smola, and E. Howy (2021), *Hierarchical attention networks for document classification,'' in Proc. Conf. North Amer. Chapter Assoc. Comput. Linguistics*, pp. 1480–1489. 19862 VOLUME 9

30. 29.Y. Feng, Y. Cheng (2021), *Short Text Sentiment Analysis Based on Multi-Channel CNN With Multi-Head Attention Mechanism.*

31. 30. A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin (2017), *Attention is all you need*, in Proc. Adv. Neural Inf. Process. Syst. , pp. 5998–6008.

32. 31. F. Chollet. (2015), *Keras[CP/OL].*, GitHub.

33. 32. D. P. Kingma and J. Ba (2014), *Adam: A method for stochastic optimization*, arXiv:1412.6980.

34. 33. A. Bermingham and A. F. Smeaton (2010), *Classifying sentiment in microblogs: Is brevity an advantage?*, Proc. 19th ACM Int. Conf. Inf. Knowl. Manage., pp. 1833–1836.

35. 34. Y. Cao, R. Xu, and T. Chen (2015), *Combining convolutional neural network and support vector machine for sentiment classification, Proc. Chin. Nat. Conf. Social Media Process.*, pp. 144–155.

36. 35. I. Hendrickx, S. N. Kim, Z. Kozareva, P. Nakov, D. Ó. Séaghdha, S. Padó, M. Pennacchiotti, L. Romano, and S. Szpakowicz (2009), *SemEval-2010 task 8: Multi-way classification of semantic relations between pairs of nominals,* Proc. Workshop Semantic Eval., Recent Achievements Future Directions (DEW), pp. 33–38.

37. 36. M. Pontiki, D. Galanis, J. Pavlopoulos, H. Papageorgiou, I. Androutsopoulos, and S. Manandhar (2014), *SemEval-2014 task 4: Aspect based sentiment analysis*, Proc. 8th Int. Workshop Semantic Eval. (SemEval). Dublin, Ireland: Association Computational Linguistics, pp. 27–35.

38. 37. M. Nabil, M. Aly, and A. Atiya (2015), *ASTD: Arabic sentiment tweets dataset*, Proc. Conf. Empirical Methods Natural Lang. Process., pp. 2515–2519.

39. 38. N. A. Abdulla, N. A. Ahmed, M. A. Shehab, and M. Al-Ayyoub (2013), *Arabic sentiment analysis: lexicon-based and corpus-based*, Proc. IEEE Jordan Conf. Appl. Electr. Eng. Comput. Technol. (AEECT), pp. 1–6.

40. 39. A. Mourad and K. Darwish (2013), *Subjectivity and sentiment analysis of modern standard Arabic and Arabic microblogs*, Proc. 4th Workshop Comput. Approaches Subjectivity, Sentiment Social Media Anal., pp. 55–64.

41. 40. S. Rosenthal, N. Farra, and P. Nakov (2017), *SemEval-2017 task 4: Sentiment analysis in Twitter*, Proc. 11th Int. Workshop Semantic Eval. (SemEval), pp. 502–518.

42. Mohammad, S. M., Salameh, M., & Kiritchenko, S. (2016). How translation alters sentiment. Journal of Artificial Intelligence Research, 55, 95-130.

43. Gupta, I., & Joshi, N. (2021). Feature-based twitter sentiment analysis with improved negation handling. IEEE Transactions on Computational Social Systems, 8(4), 917-927.

44. Lamsal, R. (2020). Coronavirus (covid-19) tweets dataset. *IEEE Dataport*, *10*.

45. Ding, Y. (2013). A quantitative analysis of words with implied negation in semantics. Theory and Practice in Language Studies, 3(7), 1200.

46. Farooq, U., Mansour, H., Nongaillard, A., & Qadir, M. A. (2016). Negation handling in sentiment analysis at sentence level.

47. Pang, B., & Lee, L. (2004). A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. arXiv preprint cs/0409058.

48. Abdel-Hafez, A., Xu, Y., & Tjondronegoro, D. (2012). Product Reputation Model: An Opinion Mining Based Approach. In *Sdad@ ecml/pkdd* (pp. 16-27).

49. Farooq, U., Nongaillard, A., Ouzrout, Y., & Qadir, M. A. (2016). A feature-based reputation model for product evaluation. *International Journal of Information Technology & Decision Making*, *15*(06), 1521-1553.

50. Yan, Z., Jing, X., & Pedrycz, W. (2017). Fusing and mining opinions for reputation generation. *Information Fusion*, *36*, 172-184.

51. Benlahbib, A., & Nfaoui, E. H. (2020). A hybrid approach for generating reputation based on opinions fusion and sentiment analysis. *Journal of Organizational Computing and Electronic Commerce*, *30*(1), 9-27.

52. Elmurngi, E. I., & Gherbi, A. (2020). Building sentiment analysis model and compute reputation scores in e-commerce environment using machine learning techniques. *International Journal of Organizational and Collective Intelligence (IJOCI)*, *10*(1), 32-62.

53. Benlahbib, A. (2020). Aggregating customer review attributes for online reputation generation. *IEEE Access*, *8*, 96550-96564.

54. Gupta, A., Priyani, S., & Balakrishnan, R. (2020). Customized reputation generation of entities using sentiment analysis. *World Journal of Engineering*, *18*(4), 596-605.

55. Boumhidi, A., & Nfaoui, E. (2021). Leveraging lexicon-based and sentiment analysis techniques for online reputation generation. *Int. J. Intell. Eng. Syst.*, *14*(6), 274-289.

56. Pontiki, M., Galanis, D., Papageorgiou, H., Androutsopoulos, I., Manandhar, S., AL-Smadi, M., ... & Eryiğit, G. (2016). Semeval-2016 task 5: Aspect based sentiment analysis. In *ProWorkshop on Semantic Evaluation (SemEval-2016)* (pp. 19-30). Association for Computational Linguistics.

57. Xue, W., & Li, T. (2018). Aspect based sentiment analysis with gated convolutional networks. *arXiv preprint arXiv:1805.07043*.

58. LeCun, Y., & Bengio, Y. (1998). In MA Arbib, editor. *Convolutional Networks for Images, Speech, and Time Series*, 255-258.

59. Dey, L., Chakraborty, S., Biswas, A., Bose, B., & Tiwari, S. (2016). Sentiment analysis of review datasets using naive bayes and k-nn classifier. *arXiv preprint arXiv:1610.09982*.

60. Vinodhini, G., & Chandrasekaran, R. M. (2012). Sentiment analysis and opinion mining: a survey. *International Journal*, *2*(6), 282-292.

61. Alexandridis, G., Varlamis, I., Korovesis, K., Caridakis, G., & Tsantilas, P. (2021). A survey on sentiment analysis and opinion mining in greek social media. *Information*, *12*(8), 331.

12. APPENDIX (CODE)

```
import pandas as pd
import numpy as np
import nltk
import re
nltk. download('stopwords')
from nltk.corpus import stopwords
from nltk.stem.porter import PorterStemmer
data = pd.read_csv('')
data
dataone=data.iloc[0:,1:]
dataone
Reviews=[]
stemming = PorterStemmer()
for i in range(12268):
  review = data['Review'][i]
  review = re.sub('[^a-zA-Z]',' ',review)
  review = review.lower()
  review = review.split()
  review = [word for word in review if word not in set(stopwords.words('english'))]
  review = [stemming.stem(word) for word in review]
  review = ' '.join(review)
  Reviews.append(review)
print(Reviews[0:20])
Data
```

```
Dataone
dataone['Rating']=dataone['Rating'].replace({'Pos':1,'Neg':0})
print(dataone)
Dataone
from sklearn.feature_extraction.text import CountVectorizer
count = CountVectorizer()
x = count.fit_transform(Reviews).toarray()
x
y = dataone['Rating'].values
y
from sklearn.model_selection import train_test_split
xtrain,xtest,ytrain,ytest = train_test_split(x,y,test_size=0.2,random_state=15)
x.shape, y.shape
xtrain.shape, xtest.shape
ytrain.shape, ytest.shape
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense
model = Sequential()
model.add(Dense(2000,activation='relu'))
model.add(Dense(3000,activation='relu'))
model.add(Dense(1500,activation='relu'))
model.add(Dense(1,activation='sigmoid'))
model.compile(loss='binary_crossentropy', optimizer='adam',metrics=['accuracy'])
model. fit(xtrain,ytrain,epochs=10,batch_size=5,validation_data=(xtest,ytest))
testdata= ''
testdata = re.sub('[^a-zA-Z]',' ',testdata)
testdata = testdata.lower()
testdata = testdata.split()
testdata = [word for word in testdata if word not in set(stopwords.words('english'))]
testdata = [stemming.stem(word) for word in testdata]
testdata = ' '.join(testdata)
testdata = count.transform([testdata]).toarray()
prediction = model.predict(testdata)
print(testdata)
print(prediction)
if prediction>0.5:
  print('Positive')
else:
  print('Negative')
```

Main.py

```
import numpy as np
import pandas as pd
import os
import nltk
import re
nltk.download('stopwords')
from tensorflow.keras.models import load_model
from tensorflow.keras.preprocessing import image
```

```python
from flask import Flask,render_template,request
from sklearn.feature_extraction.text import CountVectorizer
from nltk.corpus import stopwords
from nltk.stem.porter import PorterStemmer
import pickle
app=Flask(__name__)
model=load_model("weightsone.h5")
with open('count_vectorizer.pkl', 'rb') as file:
    count = pickle.load(file)
@app.route('/')
def index():
    return render_template("index.html")
@app.route('/predict',methods=['GET','POST'])
def upload():
    if request.method=='POST':
        testdata=request.form['name']
        stemming = PorterStemmer()
        testdata = re.sub('[^a-zA-Z]', ' ', testdata)
        testdata = testdata.lower()
        testdata = testdata.split()
        testdata = [word for word in testdata if word not in set(stopwords.words('english'))]
        testdata = [stemming.stem(word) for word in testdata]
        testdata = ' '.join(testdata)
        testdataone = count.transform([testdata]).toarray()
        prediction = model.predict(testdataone)
        if prediction > 0.5:
            text='Positive'
        else:
            text='Negative'
    return render_template("result.html",output=text)
if __name__=='__main__':
    app.run()
```

Index.html

```html
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Hotel Feedback</title>
  <style>
    /* Reset default browser styles */
    body, h1, h2, p, ul, li {
      margin: 0;
      padding: 0;
    }

    /* Global styles */
    body {
      font-family: Arial, sans-serif;
      line-height: 1.5;
      background-color: #f2f2f2;
    }
```

```css
header {
  background-color: #222;
  color: #fff;
  padding: 10px;
  display: flex;
  justify-content: space-between;
  align-items: center;
  box-shadow: 0px 2px 4px rgba(0, 0, 0, 0.2);
}

header ul {
  list-style: none;
  display: flex;
}

header ul li {
  margin-right: 10px;
}

header ul li a {
  color: #fff;
  text-decoration: none;
  padding: 10px;
  border-radius: 5px;
  transition: background-color 0.3s ease;
}

header ul li a:hover {
  background-color: #333;
}

nav ul {
  list-style: none;
}

nav ul li {
  display: inline-block;
  margin-right: 10px;
}

nav ul li a {
  color: #fff;
  text-decoration: none;
  padding: 10px;
  transition: all 0.3s ease;
}

nav ul li a:hover {
  background-color: #333;
  border-radius: 5px;
}

.hero {
  text-align: center;
  padding: 100px 0;
```

```css
  background-image: url('/static/images/img.jpg');
  background-size: cover;
  background-position: center;
}

.hero-content {
  background-color: rgba(255, 255, 255, 0.8);
  padding: 20px;
  border-radius: 10px;
  display: inline-block;
}

.hero h1 {
  font-size: 48px;
  margin-bottom: 20px;
  color: #222;
  text-shadow: 2px 2px 4px rgba(0, 0, 0, 0.5);
}

.hero p {
  font-size: 24px;
  margin-bottom: 30px;
  color: #555;
}

.cta-button {
  display: inline-block;
  padding: 15px 30px;
  background-color: #ff5e14;
  color: #fff;
  text-decoration: none;
  font-size: 20px;
  font-weight: bold;
  border-radius: 5px;
  transition: all 0.3s ease;
}

.cta-button:hover {
  background-color: #e54311;
}

.testimonial {
  padding: 50px 0;
  text-align: center;
}

.testimonial h2 {
  font-size: 36px;
  margin-bottom: 40px;
  color: #222;
  text-shadow: 2px 2px 4px rgba(0, 0, 0, 0.5);
}

.testimonial-card {
  background-color: #fff;
```

```css
  padding: 30px;
  border-radius: 10px;
  margin-bottom: 40px;
  box-shadow: 0px 2px 10px rgba(0, 0, 0, 0.1);
}

.testimonial-card p {
  margin-bottom: 20px;
  color: #555;
}

.testimonial-card cite {
  font-style: italic;
  color: #777;
}

footer {
  background-color: #222;
  color: #fff;
  text-align: center;
  padding: 10px;
}

.feedback-form {
  text-align: center;
  margin: 50px auto;
  max-width: 500px;
  padding: 20px;
  background-color: #fff;
  border-radius: 10px;
  box-shadow: 0px 2px 10px rgba(0, 0, 0, 0.1);
}

.feedback-form textarea {
  width: 100%;
  padding: 10px;
  margin-bottom: 20px;
  border: 1px solid #ccc;
  border-radius: 5px;
  resize: vertical;
}

.feedback-form button {
  padding: 10px 20px;
  background-color: #ff5e14;
  color: #fff;
  font-size: 18px;
  border: none;
  border-radius: 5px;
  cursor: pointer;
  transition: all 0.3s ease;
}

.feedback-form button:hover {
  background-color: #e54311;
```

```
    }
    ::-webkit-scrollbar {
      width: 10px;
    }

    ::-webkit-scrollbar-track {
      background-color: #f2f2f2;
    }

    ::-webkit-scrollbar-thumb {
      background-color: #888;
      border-radius: 5px;
    }

    ::-webkit-scrollbar-thumb:hover {
      background-color: #555;
    }
  </style>
</head>
<body>
  <header>
    <nav>
      <ul>
        <li><a href="#">Home</a></li>
        <li><a href="#">Rooms</a></li>
        <li><a href="#">Reviews</a></li>
        <li><a href="#">Contact</a></li>
      </ul>
    </nav>
  </header>

  <section class="hero">
    <div class="hero-content">
      <h1>Welcome to Hotel Feedback</h1>
      <p>Share your experience with us</p>
      <section class="feedback-form">
    <h2>Leave Your Feedback</h2>
    <form method="POST" action="/predict">

     <label for="review"></label>
        <textarea id="review" name="name" rows="4" cols="50"></textarea><br>


  </section>
      <input type="submit" class="cta-button" value="Leave Feedback">
    </form>
    </div>
  </section>

  <section class="testimonial">
    <h2>Guest Testimonials</h2>
    <div class="testimonial-container">
      <div class="testimonial-card">
        <p>"I recently stayed at this hotel for a business trip, and I was thoroughly impressed. The service
was exceptional, and the staff went above and beyond to ensure my stay was comfortable. The amenities
```

```
were top-notch, and the hotel's location made it convenient for my meetings. I would definitely choose this hotel again for future trips."</p>
        <cite>- Raj Kapoor (CTO, Loops pvt ltd)</cite>
      </div>
      <div class="testimonial-card">
        <p>"What a delightful stay! The hotel exceeded my expectations in every way. The room was stylish and modern, with all the amenities I needed. The hotel's restaurant served delicious food, and the staff was knowledgeable and attentive. I left feeling rejuvenated and would recommend this hotel to anyone seeking a luxurious and relaxing getaway."</p>
        <cite>- Kanika Sharma (Tripadvisor)</cite>
      </div>
    </div>
  </section>



  <footer>
    <p>&copy; 2023 Hotel Feedback. All rights reserved.</p>
  </footer>
</body>
</html>
```

Result.html

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Feedback Result</title>
  <style>
    body, h1, p {
      margin: 0;
      padding: 0;
    }
    body {
      font-family: Arial, sans-serif;
      line-height: 1.5;
      background-color: #f2f2f2;
      display: flex;
      justify-content: center;
      align-items: center;
      height: 100vh;
    }

    .result-container {
      background-color: #fff;
      padding: 40px;
      border-radius: 10px;
      box-shadow: 0px 2px 10px rgba(0, 0, 0, 0.1);
      text-align: center;
    }

    .result-container h1 {
```

```
      font-size: 36px;
      margin-bottom: 20px;
      color: #222;
    }

    .result-container p {
      font-size: 18px;
      color: #555;
      margin-bottom: 40px;
    }

    .sentiment-label {
      display: inline-block;
      padding: 10px 20px;
      font-size: 18px;
      font-weight: bold;
      border-radius: 5px;
    }

    .Positive {
      background-color: #4caf50;
      color: #fff;
    }

    .Negative {
      background-color: #f44336;
      color: #fff;
    }

    .neutral {
      background-color: #2196f3;
      color: #fff;
    }

    .go-back-link {
      color: #555;
      text-decoration: none;
      font-size: 16px;
      transition: all 0.3s ease;
    }

    .go-back-link:hover {
      color: #222;
    }
  </style>
</head>
<body>
  <div class="result-container">
    <h1>Feedback Result</h1>
    <p>The sentiment analysis of your feedback is:</p>
    <span class="sentiment-label {{ output }}"> {{ output }} </span>
    <a href="/" class="go-back-link">Go Back</a>
  </div>
</body>
</html>
```