**DATA 608**



**FINAL PROJECT REPORT**

**Subject: Hotel Reviews: NLP & Sentiment Analysis**

**Group 5**

**Team members:**

**Deyvis Dalman (30166177)**

**Devanshi Patel (30186470)**

**Jannatul Naeema (30198919)**

**Utsav Agarwal (30198220)**

**INDEX**

[INTRODUCTION 2](#_Toc1442070119)

[DATASET 3](#_Toc1952344451)

[LEARNING OBJECTIVES 4](#_Toc325930394)

[METHODOLOGIES 5](#_Toc905230150)

[DATA WRANGLING AND PREPROCESSING 5](#_Toc389723355)

[WORDCLOUD 8](#_Toc192683014)

[BAG OF WORDS 9](#_Toc1956591153)

[TF-IDF (Term Frequency - Inverse Document Frequency) 10](#_Toc149996200)

[TEXT CLASSIFICATION 12](#_Toc303980956)

[SENTIMENT ANALYSIS WITH VADER AND BERT 12](#_Toc1277339042)

[WEB APPLICATION DEPLOYMENT ON AMAZON AWS 12](#_Toc2051103789)

[CONCLUSION 12](#_Toc353523453)

[FUTURE STEPS 12](#_Toc360737515)

[REFERENCES 12](#_Toc1230619481)

# **INTRODUCTION**

Reviews are a useful tool in providing necessary feedback to business. Apart from informing business on how well their services are received by customers, they also allow consumers to communicate their opinions to others in an honest manner.

Sentiment analysis is a method of natural language processing (NLP) which can be used to ascertain the sentiment of a review. Natural language processing (NLP) may be used to determine the general feeling of a review, whether it be good, bad, or neutral. An understanding of the patterns present within the reviews provides valuable insight into quality of services as well as areas of improvement for a business. These insights provide leadership and management with tools to create meaningful operational plans for improving profitability of the business and retention of higher number of customers.

In this report we describe how our team utilized various NLP techniques and sentiment analysis on the datasets of hotel reviews from TripAdvisor and Booking.com.

# **DATASET**

**Dataset 1: TripAdvisor [1]**

The dataset the team has chosen features hotel reviews scraped from TripAdvisor for 10 different cities (Dubai, Beijing, Long, New York City, New Delhi, San Francisco, Shanghai, Montreal, Las Vegas, Chicago). Each city contains a varying number of hotels. The text fields featured within the dataset are date, review title and full review as shown in Fig 1. Dataset contains approximately 259,000 reviews. The team has chosen 5 of the 10 cities London, Dubai, Delhi, Shanghai and New York to focus their efforts.



**Fig 1: Dataset**

**Dataset 2: Booking.com[6]**

•Found on Kaggle, features 515k reviews from luxury hotels in Europe.

•Dataset was scraped from booking.com, it features 17 fields such as Review date, Average

score, Reviewer Score etc.

# **LEARNING OBJECTIVES**

Devanshi (Learning Object - Completed:5/5)

* Explored Data wrangling process by converting files of “All type format” to CSVs and separating review text and date using regular expression I targeted to learn automation from reading the file name to file conversion, target to complete in 1 week.
* Learning Natural Language Processing (NLP) techniques for keyword extraction techniques (removing stop words, Stemming, lemmatization removing irrelevant words not significant to hotel) on review of a dataset, target to complete in 1-2 week after above point is completed.
* Once the words have been extracted, I intend to visualize an overall wordcloud for each city consisting of all hotels (Using Python WordCloud package), target to complete in 1 week after above point is completed. This is a demonstration of all the above techniques have learned. [4,5,10]
* Explore bag of words technique and visualize output for cities and find the trend.
* Webapp deployment on Amazon AWS (Amazon Web Services)

Deyvis: (Learning Object - Completed: 2/3)

* Learned Text Classification on each review to classify the topic of each review. Possible classifications may be cleanliness, food, service, or amenities for a hotel.
* Breakdown for each label classification to be done to identify areas of strength and weaknesses for a hotel, planned. Time permitting add time series to classification.
* Both rule-based text classification and machine learning text classification will be explored, if possible, in the time frame of the project.

Jannatul: (Learning Object - Completed:4/4)

* Familiarized myself with different NLP techniques. [2,3,5] Mostly I learned details of the concepts of tokenizer, auto tokenizer and how to perform TF-IDF (Term Frequency - Inverse Document Frequency) in week 1. [7,8]
* In the next week, I familiarized myself with various NLP techniques, including the concepts of tokenizer, auto tokenizer, and performing TF-IDF. Specifically, I delved into the details of these techniques and gained a good understanding of them.
* In the following week, I also learned about Vader library and Bert pretrained models and the concept of a pipeline. [9,10] I was able to implement these models and their pipelines effectively.
* Moving on to the next week, I successfully implemented TF-IDF (Term Frequency - Inverse Document Frequency) on Dataset-1 and generated a list of the most relevant words for five cities (Delhi, Dubai, Shanghai, London and NYC).
* In the following 2 weeks I have learned and implemented Vader (Valence Aware Dictionary and Sentiment Reasoner) sentiment analysis. I used this technique to analyze the reviews of four hotels in Dataset-1 and achieved satisfactory results.
* After studying the Bert (Bidirectional Encoder Representations from Transformers) pre-trained model, I attempted to apply it to hotel reviews from Delhi. However, I encountered several challenges along the way. First, I discovered that the Bert model cannot accept input if the reviews exceed 256 tokens, and it does not work with non-English characters or languages other than English. Although most of the reviews in the dataset were in English, there were a few rows with reviews containing unfamiliar characters such as (? or !). Devanshi and I worked together to address this issue, I had to perform data cleaning again and remove those rows. In addition, I added a 'try except' loop to ignore rows with reviews that exceeded 256 words. Finally, I faced the challenge of requiring a GPU to run Bert, which was a new concept for me. I ended up paying for Collab Pro to use GPU and TPU to run the Bert model on the datasets of four cities (Dubai, Delhi, Shanghai, London), and obtained some useful results.
* However, another challenge we faced was that our dataset did not contain a rating column, so we could not compare the sentiments obtained from Bert and Vader with the original ratings.
* At the project planning stage, I proposed to implement the Bert model on reviews of five cities (Delhi, Dubai, Shanghai, London and NYC). After working for a few weeks when I realized the volume of task, I decided to proceed with four cities of Delhi, Dubai, Shanghai, and London out of five cities.

Utsav: (Learning Object - Completed:3/3)

* In week 1, I learned about Vader library and Bert pretrained models [9,10]
* In week2, I perform sentiment analysis using NLTK library and VADER library specifically on three-scale points (positive/negative/Neutral)
* In rest of the week, I tried to do transfer learning with BERT, but it took long time to run the same on our original TripAdvisor dataset with Jannatul. By the time of final presentation, we were able to finish the model on our original TripAdvisor dataset but could not finish the same on Booking.com dataset
* I also learned about dockers and containers deployment on Amazon AWS during the course and worked with Devanshi on deployment using flask

# **METHODOLOGIES**

## **DATA WRANGLING AND PREPROCESSING (TripAdvisor dataset)**

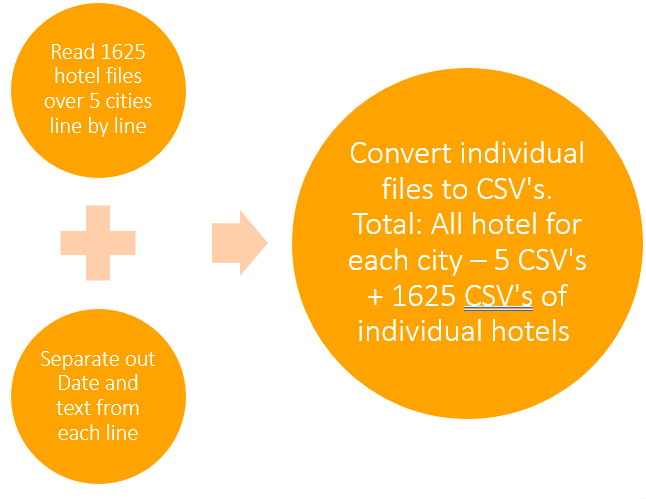
We had to follow a process before we could dive into any technique's exploration in NLP. Thus, making a script to handle more than 1645 files was the most convenient way to handle it in real time. Here are the steps which script followed for data wrangling and preprocessing.

**Step 1:** Give path to the folder of the city, using Python package get list of all files in folders (hotel names list), convert it into a list data structure.

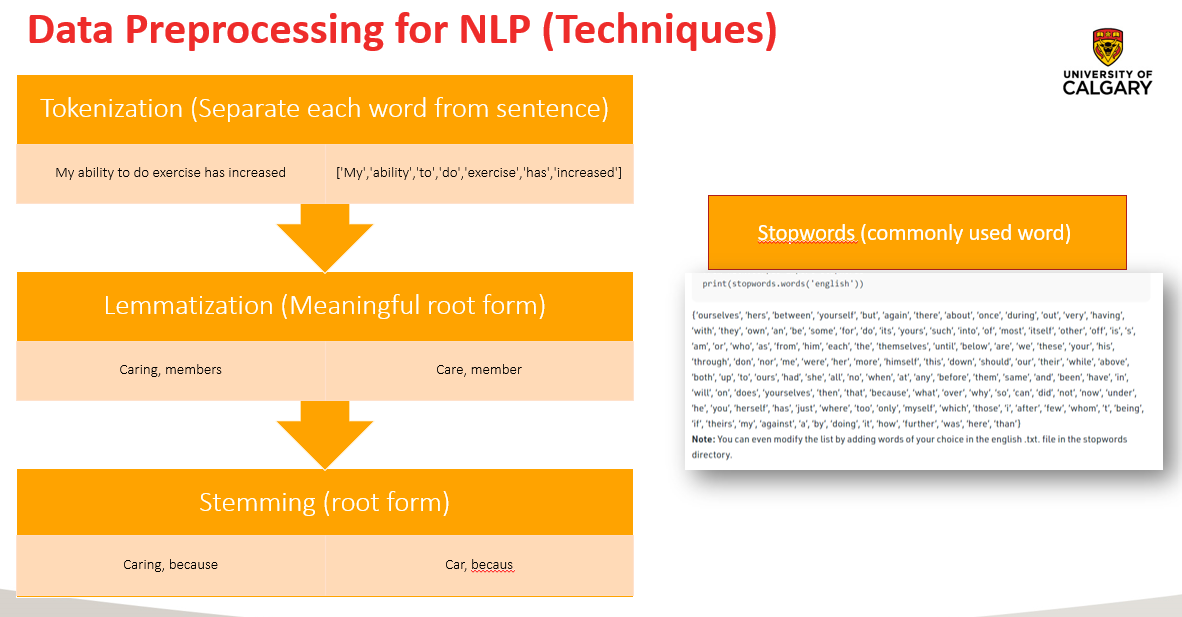
**Step 2:** The list of hotel names obtained is fed further to another function which parses through each file name in the list and in each file, it parses the text line by line where the date and reviews are separated using regular expression. (As shown in Fig 2)

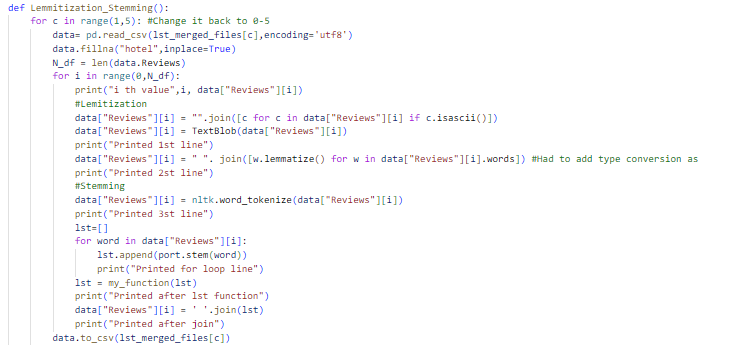
**Step 3:** For each line read it was passed to a function which performed Tokenization. Lemmatization and stemming and removal of stop words. (Example of what each technique does is shown in Fig 3)

**Step 4:** Each line after the NLP preprocessing was added to a pandas DataFrame. Once the whole file is parsed the pandas DataFrame is exported as a “csv” file format. (As shown in Fig 4)



**Fig 2: Separate date and review and removing stopwords**





**Fig 3: Tokenization, Lemmatization, Stemming**



**Fig 4: Final output of preprocessing CSV files**

## 

## **DATA WRANGLING AND PREPROCESSING (Booking.com dataset)[6]**

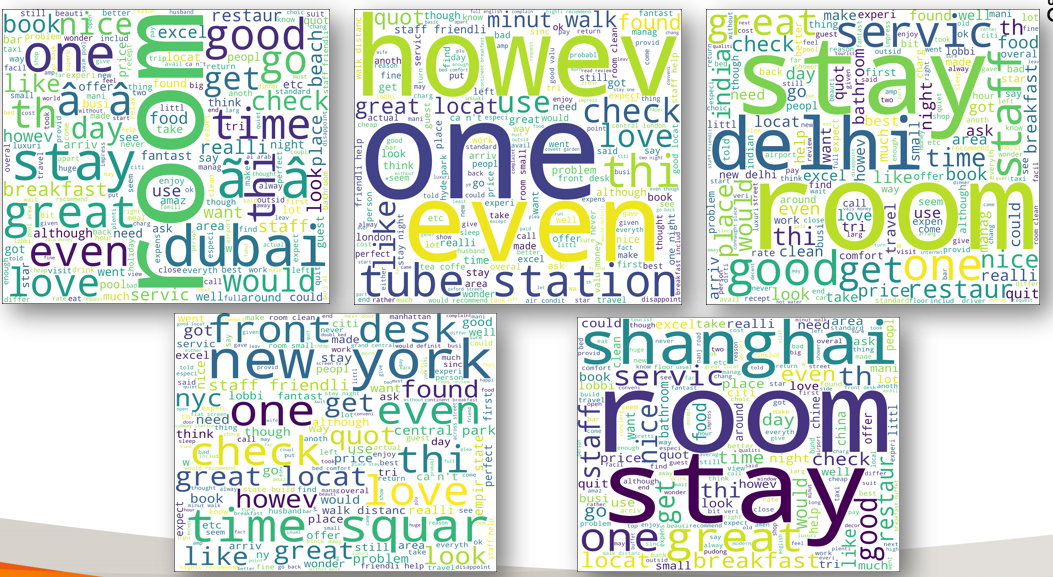
During Sentiment analysis using Vader we realize that Tripadvisor dataset didn’t have individual review score. To do sentiment analysis from Vader, we decided to get another dataset from Kaggle from Booking.com. Dataset was clean with no missing values.

After importing the dataset in pandas DataFrame, we performed Tokenization. Lemmatization and stemming and removal of stopwords to prepare dataset for Natural language processing

## **WORDCLOUD**

* WordCloud is a data visualization technique used for representing text data in which the size of each word indicates its frequency or importance. Significant textual data points can be highlighted using a word cloud. Word clouds are widely used for analyzing data from social network websites.
* Using the visualization technique for each city in our dataset we get following results,



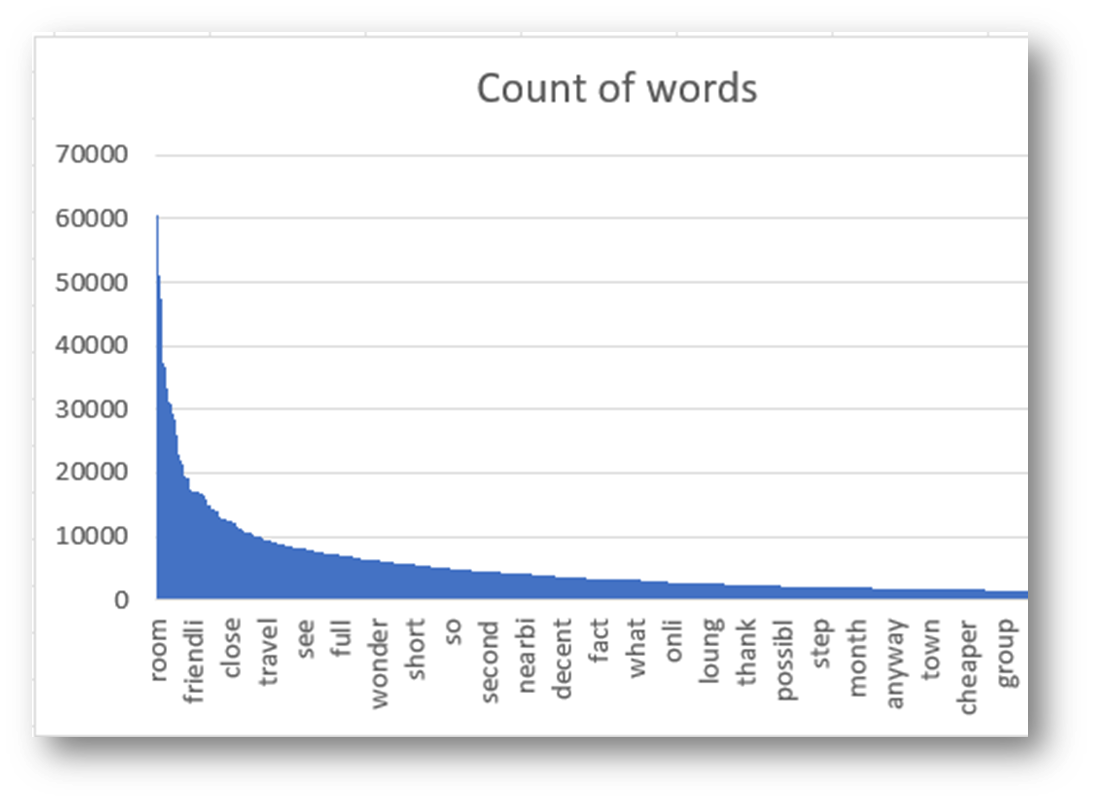


**Fig 5: Top-left(City: Dubai), Top-middle(City) ,Top-Right(City: Delhi), Bottom-left(City: New York), Bottom-right (Shanghai)**

**Insights from wordcloud:** In Dubai we can see that most of the words are nice, great, beach while in shanghai more words about service provided by hotels in room is mentioned by customers such as breakfast, staff, bathroom, time, quiet. Thus, the behavior of the customers in each city is different as the customers in Shanghai want convenient hotels while in Dubai people look for the outdoor view like beach.

## **BAG OF WORDS**

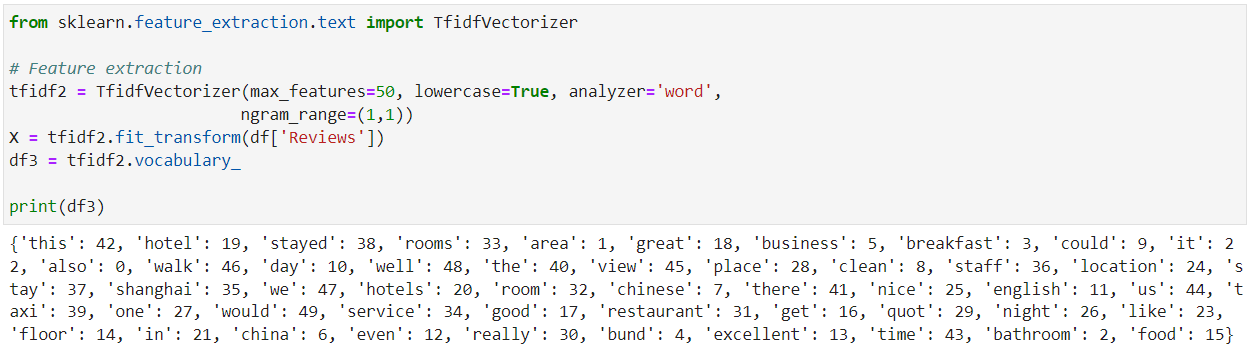
* Bag of Words model is used to preprocess the text by converting it into a bag of words, which keeps a count of the total occurrences of most frequently used words. [11]
* This model can be visualized using a table, which contains the count of words corresponding to the word itself. For better understanding we have converted the table in bar chart. (As shown in Fig 6)



**Fig 6: Bag of words for Dubai**

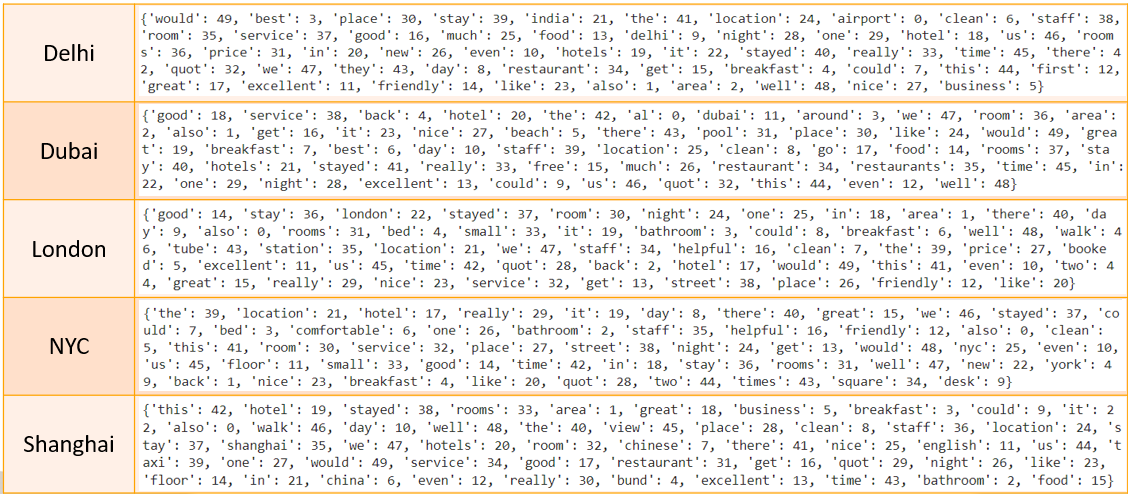
## **TF-IDF (Term Frequency - Inverse Document Frequency)**

* TF-IDF (Term Frequency - Inverse Document Frequency) is a word vectorization method in natural language processing (NLP) to measure how important a term is within a document relative to a collection of documents. [7]
* The first part TF measures the frequency of a term in a document, in our case it is reviews. The higher the frequency of a term appeared in reviews the higher the TF score will be. But the common words in all reviews don't provide much information for identifying relevant reviews. The second part IDF solves this issue by measuring how common or rare a term is across all reviews by giving more weight to terms that appear in fewer reviews. Words appearing in many reviews have a lower IDF score, while those appearing in a few reviews have a higher IDF score. Consequently, words that are unique to a specific review or a small subset of review are given greater importance. By combining these two factors, we calculated the TF-IDF weight for each term in a review. Thus, TF-IDF scores a word by multiplying the word’s Term Frequency (TF) with the Inverse Document Frequency (IDF).



**Fig 7: Sample code for TF-IDF for Shanghai [8]**

* The same procedures have been followed for other cities as well to identify the 50 most relevant words and phrases in the reviews of each city.



**Fig 8: TF-IDF for Hotels Reviews in 5 Cities**

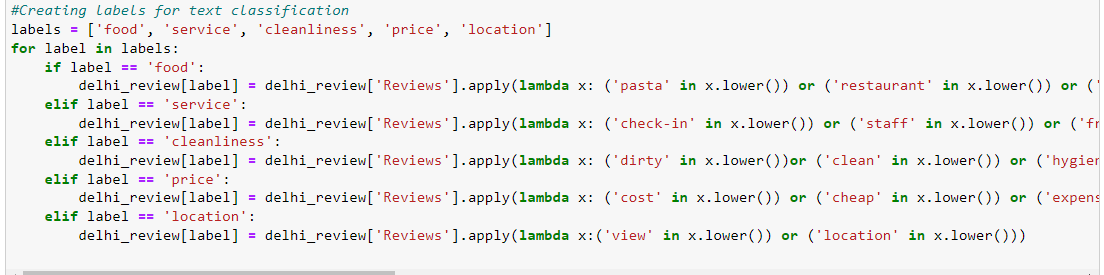
**Insights from TF-IDF:**

Using TF-IDF, we have identified the 50 highest-scoring words for each city, revealing the most important topics for tourists. In Delhi, the most relevant words in reviews are 'would', 'best', 'place', 'stay', 'India', etc. For Dubai, 'good', 'service', 'back', 'hotel', and 'the' are indicative of important areas for tourists, with 'service', 'hotel', 'room', and 'beach' emerging as the most essential topics. In London, the top words include 'good', 'stay', 'London', 'stayed', 'room', and 'night', with other important topics such as 'area', 'bed', and 'bathroom' indicating the priorities of tourists. For NYC, the most scored words are 'the', 'location', 'stayed', 'room', and 'night', with 'bed', 'staff', 'friendly', and 'helpful' indicating important amenities for guests. Finally, in Shanghai, 'this', 'hotel', 'stayed', 'rooms', and 'area' are the most important words according to TF-IDF scores, with 'business', 'breakfast', 'walk', 'day', and 'view' being other relevant words for the city.

To summarize, our analysis highlights the varying importance of amenities across different cities. For Delhi, location, cleanliness, staff, service, and food are the most crucial amenities, while Dubai reviews emphasize service, area, beach, and pool. In Shanghai, business, breakfast, walk, and view are highly valued amenities, whereas London prioritizes area, bed, bathroom, and breakfast. Finally, for NYC, bed quality, comfort, helpfulness, and friendliness are the most essential amenities.

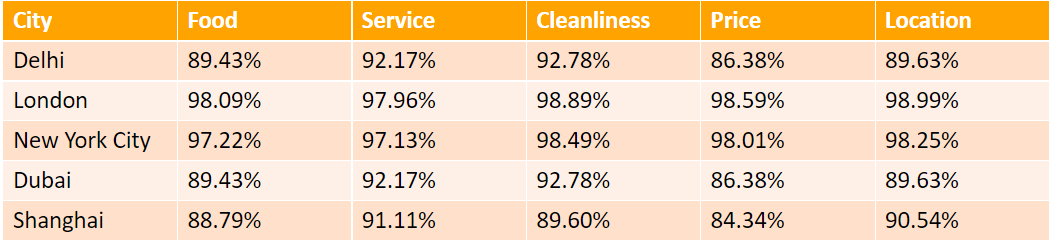
## **TEXT CLASSIFICATION**

Text classification for each city was carried out on the reviews, sorting each review into labels based on subject matter mentioned in a review. Labels were chosen via key words you’d expect to play an important role in a hotel review. The labels chosen were food, service, cleanliness, price and location. Next words to allow labeling for each category were chosen using the top 500 words by frequency count of the reviews.

**Figure 9: Label creation for multinomial logistic regression**

Multinomial logistic regression was chosen to classify each review into the chosen labels. These were the classification accuracy results produced after running the multinomial logistic regression models.

Table 1: Multinomial logistic regression results



Models such as random forest and XGboost were considered for classification. Ultimately multinomial logistic regression was chosen due to its ease of use involving multiple classification. The best performing model was London and the location label, while the worst was Shanghai and the price label. A trend which is visible in the modeling accuracy seems to show countries where English is the primary spoken language perform better in classification for reviews.

**SENTIMENT ANALYSIS WITH VADER AND BERT**

**Vader (Valence Aware Dictionary and Sentiment Reasoner) sentiment analysis**

A picture containing icon

Description automatically generated

VADER sentiment analysis is a lexicon and rule-based sentiment analysis tool that is widely used to sentiments expressed in social media. It is fully open sourced under the MIT License.[12] This model returns a sentiment score in the range -1 to 1, from most negative to most positive. Then the VADER model was run on the entire dataset for Tripadvisor dataset.

During this process team realized that Tripadvisor dataset doesn’t have individual review score. There for team downloaded Booking.com dataset from Kaggle which has 515k reviews with individual review scores. Team then performed Vader to get Negative, Neutral, Positive and compound score for each review. Team then trained a Gaussian Naïve Bayes classifier based on these scores from Vader analysis and Individual Review score to classify review into Negative, Positive or Neutral sentiment. This classifier achieved 82% accuracy on test set.

This trained model was deployed using Webapp on Amazon AWS.

**Insights from Vader Analysis:**

Vader analysis combine with Gaussian Naïve Bayes classifier can be used to predict sentiments from reviews with high accuracy.

**Bert (Bidirectional Encoder Representations from Transformers) Sentiment Analysis from HuggingFace**

Bert is a pre-trained deep learning model which is developed by Google. This model is used for natural language processing (NLP) tasks, e.g. sentiment analysis. Hugging Face provides the platform for Bert and the Hugging Face transformers library contains lots of pre-trained BERT models.[9]

**Implementation of Vader and Bert on Dataset 1:**

Vader: In order to implement the VADER sentiment analysis, the transformer and TensorFlow were installed, followed by importing the pipeline and downloading the VADER sentiment model. Then the VADER model was run on the entire dataset 1 to obtain the sentiments expressed in the dataset.

Bert: In a Google Collab notebook, the "cardiffnlp/twitter-roberta-base-sentiment" model was imported after importing AutoTokenizer. [10] Next, a dictionary was created to store sentiments. Finally, a try-except loop was implemented within a ‘for loop’ to ignore longer reviews.

The outcomes of both the Vader and Bert models on Dataset 1 are displayed in Tables 2-5 for four cities: Delhi, Dubai, Shanghai, and London.

Table 2: Vader and Bert Sentiment Scores on Hotel Reviews of Delhi



Table 3: Vader and Bert Sentiment Scores on Hotel Reviews of Dubai



Table 4: Vader and Bert Sentiment Scores on Hotel Reviews of London

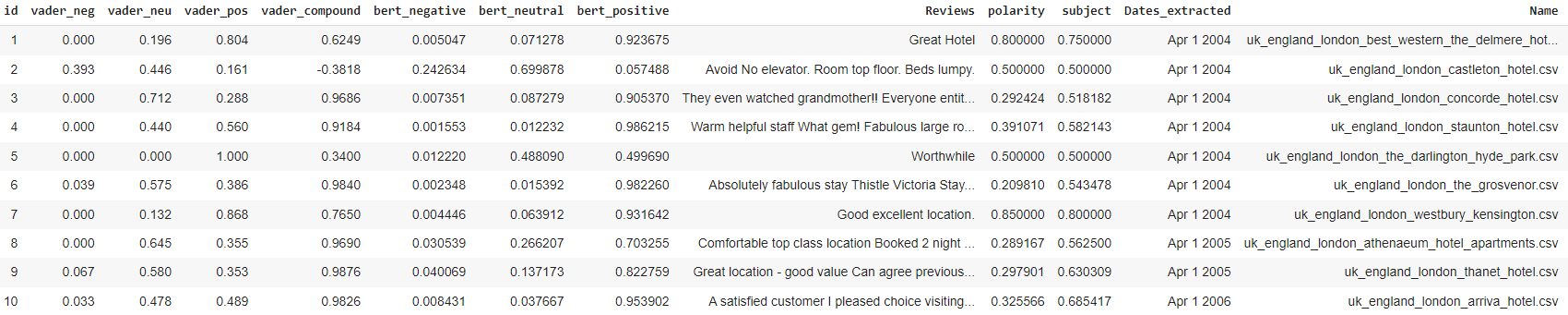
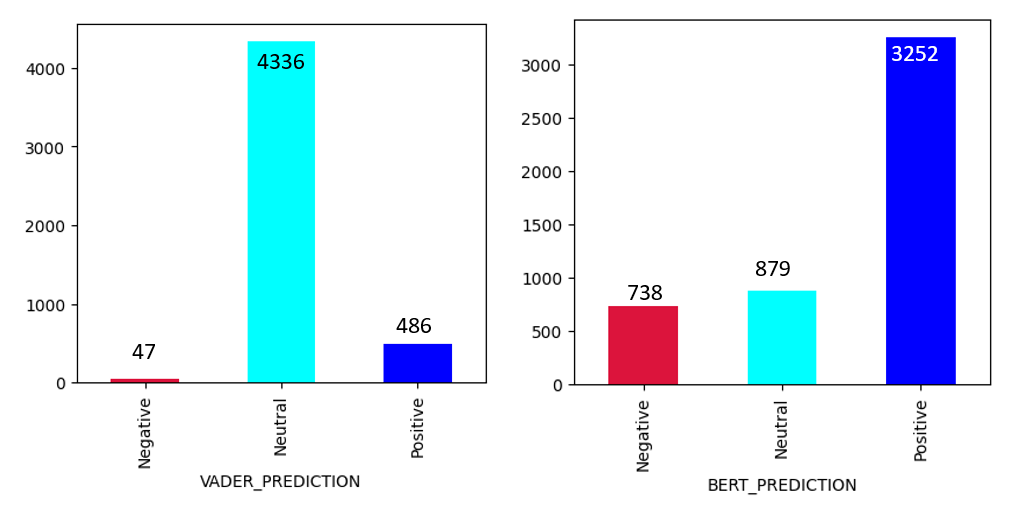


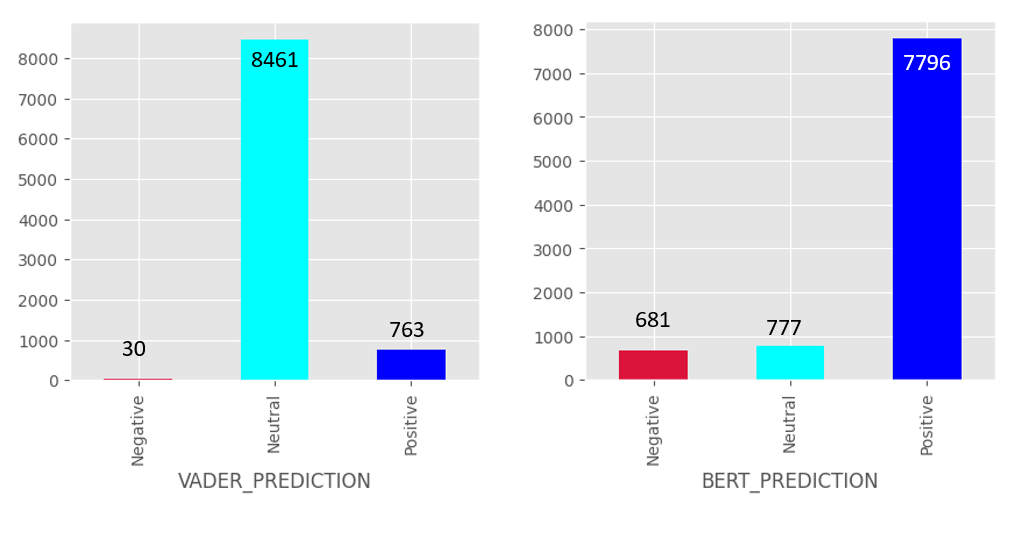
Table 5: Vader and Bert Sentiment Scores on Hotel Reviews of Shanghai



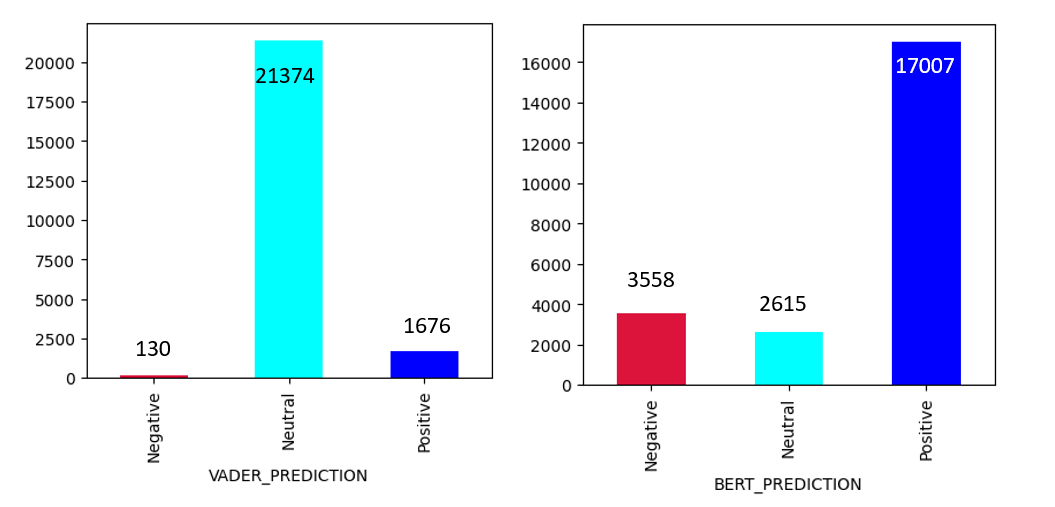
Then bar graphs were plotted to observe the comparison of sentiments of the reviews captured using two methods, this procedure was repeated for four cities (Delhi, Dubai, Shanghai, and London).



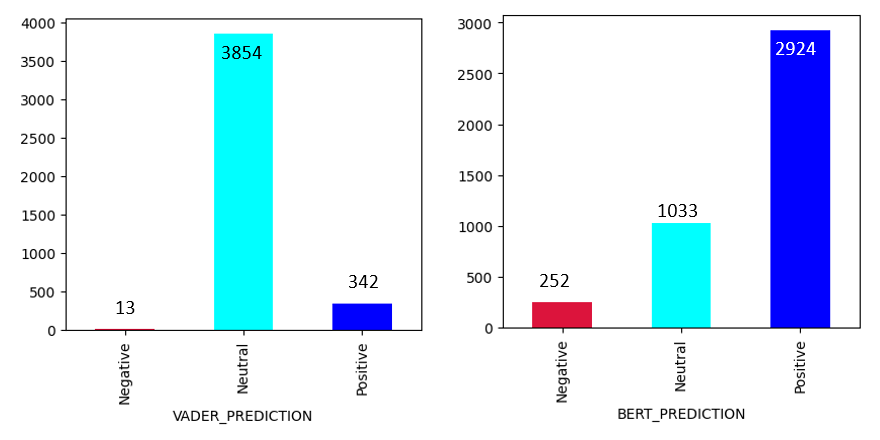
**Fig 10: Sentiment Analysis on Hotel Reviews in Delhi**



**Fig 11: Sentiment Analysis on Hotel Reviews in Dubai**



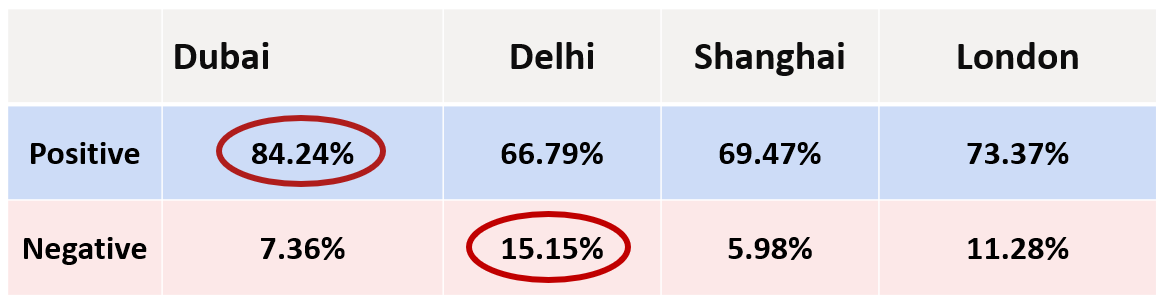
**Fig 12: Sentiment Analysis on Hotel Reviews in London**



**Fig 13: Sentiment Analysis on Hotel Reviews in Shanghai**

**Insights from Vader and Bert Model on Dataset 1:**

From fig 10-13, it is evident that the Bert model outperforms the Vader model in capturing sentiment. The Vader model tends to label a significant portion of sentiments as neutral, while the Bert model is more effective in distinguishing between different types of sentiments. As a result, when applied to the same dataset for each city, the Bert model detects more positive or negative sentiment compared to the Vader model. This suggests that the Bert model is a more reliable tool for sentiment analysis in this context.



**Fig 14: Positive and Negative Sentiments at Different Cities using Bert Model**

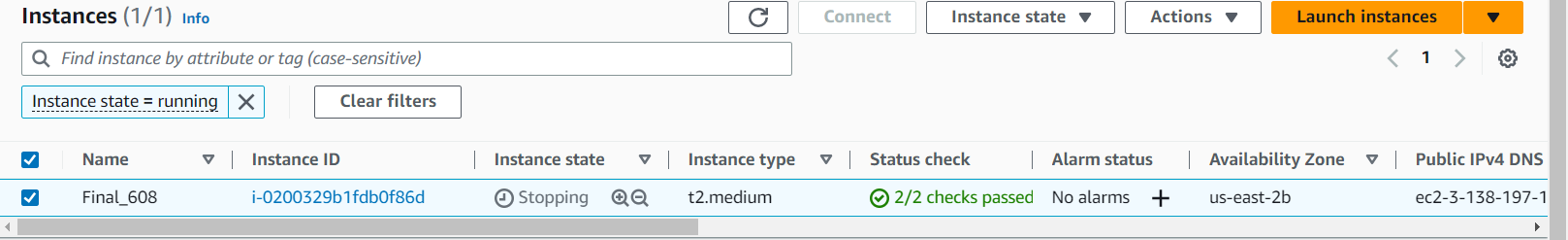
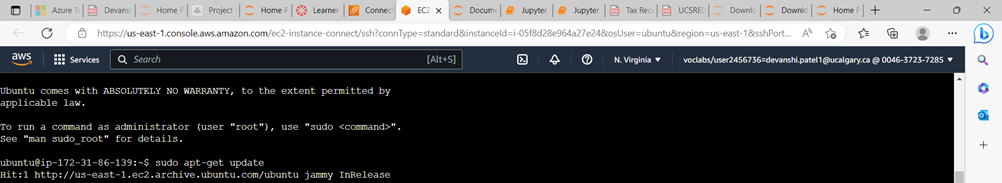
**Insights from Fig 14:**

As it was already established that Bert model performed better, it was used to analyze and summarize the percentage of positive and negative sentiments in hotel reviews from four different cities: Delhi, Dubai, Shanghai, and London. The results of this analysis were presented in figure 12. The findings of figure 12 showed that Dubai had the highest percentage of positive sentiments (84.24%) in their hotel reviews, indicating that guests had a highly positive experience during their stay. In contrast, Delhi had the least percentage of positive sentiments, implying that guests were less satisfied with their stay. Moreover, Delhi had the highest percentage of negative sentiments (15.15%) in their hotel reviews, suggesting that guests were dissatisfied with their stay and London following closely behind. The finding that London had the second-highest percentage of negative sentiments was quite surprising and could indicate potential issues with hotels in the city.

**Insights of using Vader on Dataset 1 and 2:**

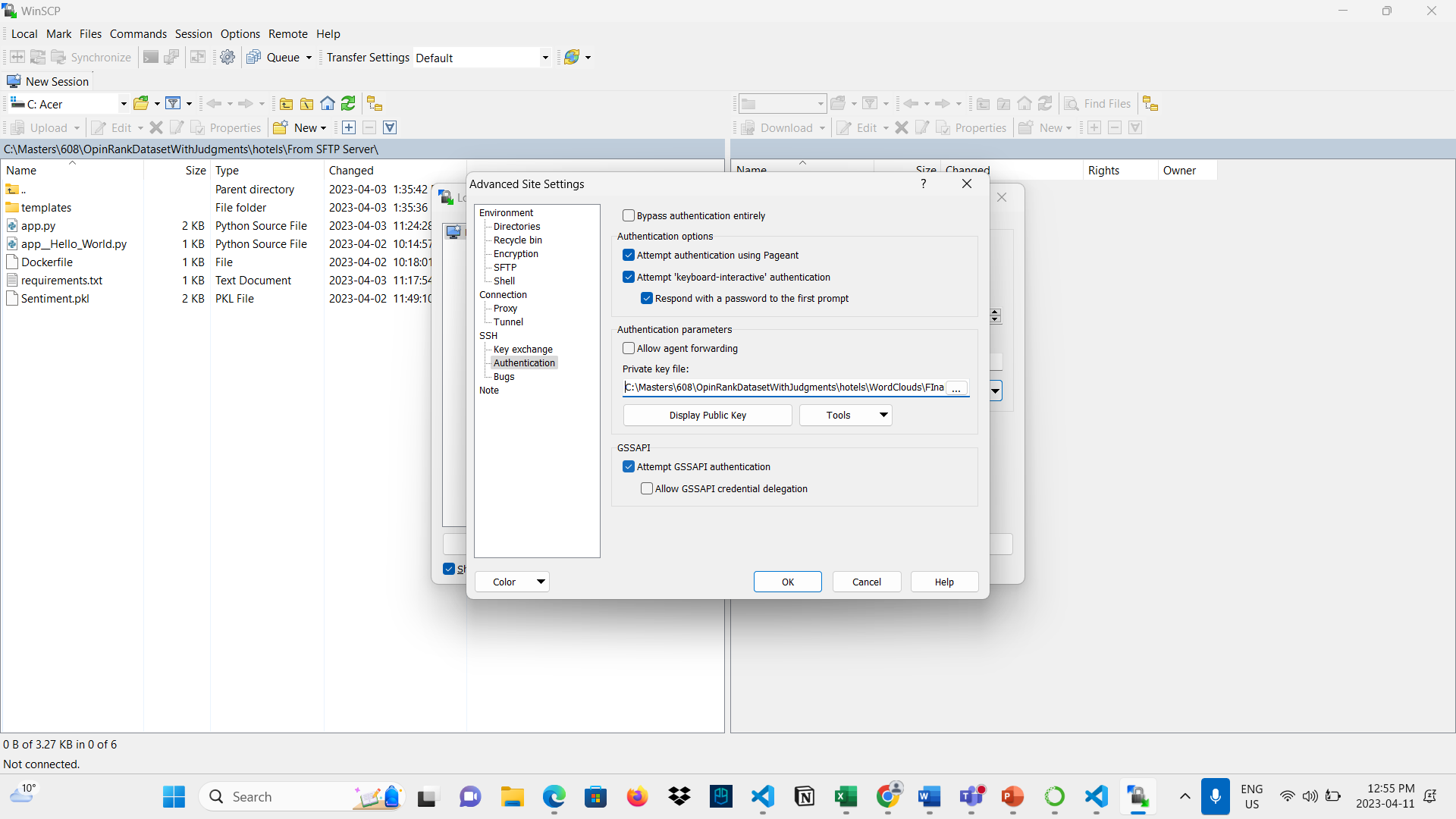
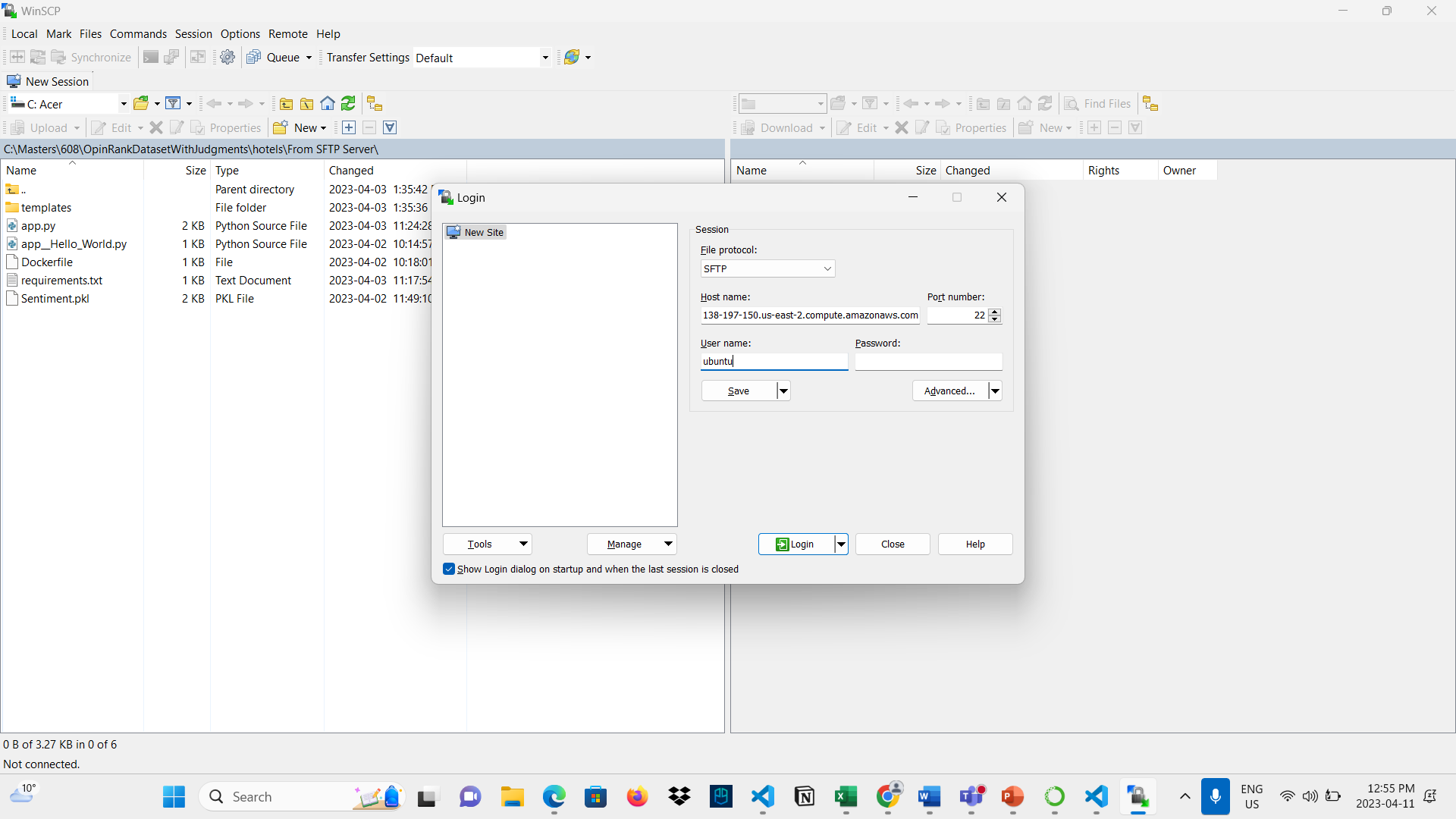
Based on the sentiment scores generated using Vader analysis team wanted to predict average scores for various hotels on both datasets. Using Gradient Boost Regressor team achieved an RMSE (Root Mean Square Error) of 0.53 on booking.com dataset and 0.6 on TripAdvisor Dubai dataset.

## **WEB APPLICATION DEPLOYMENT ON AMAZON AWS**

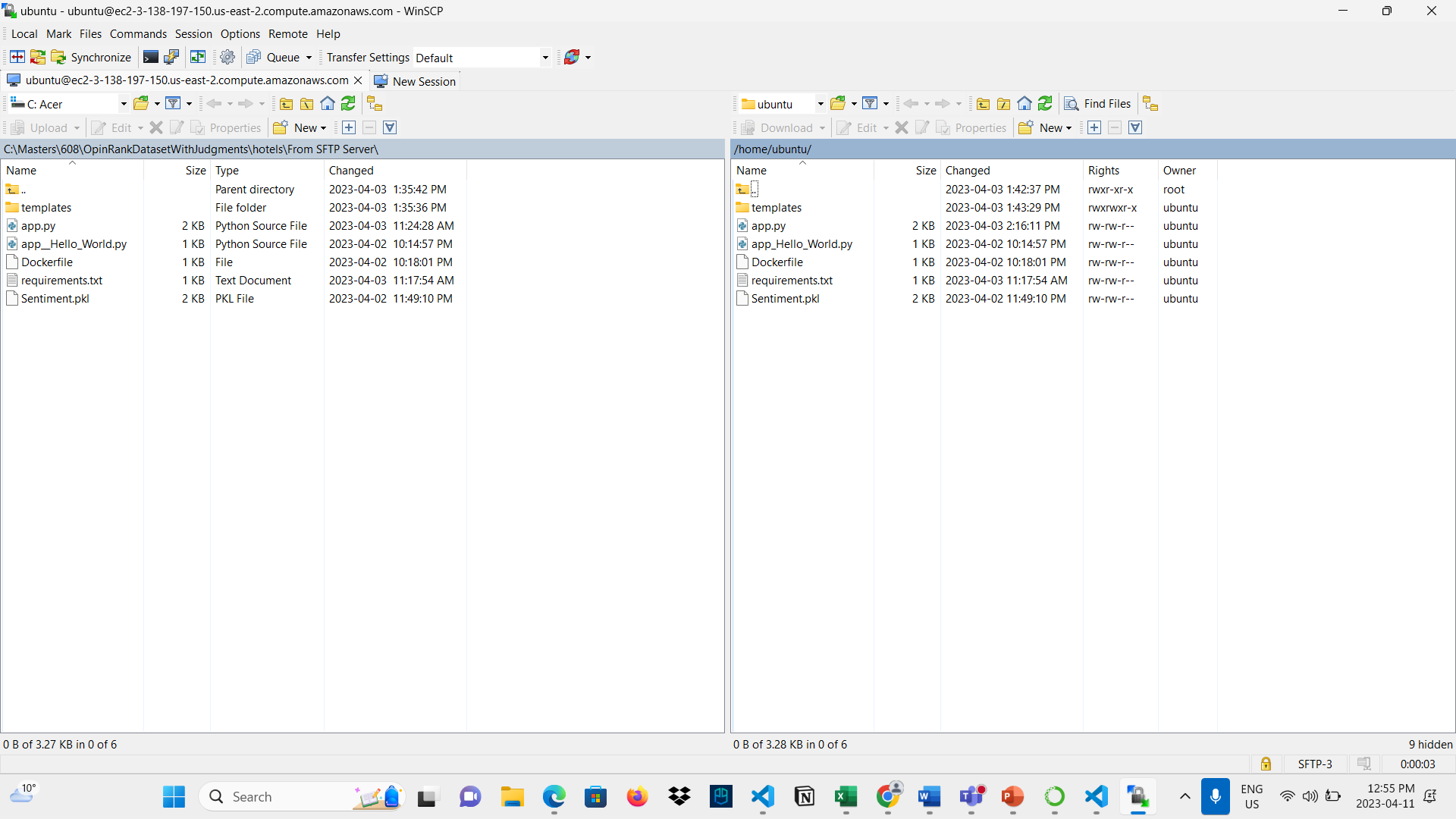
1. Login to AWS Console, Create an EC2 instance.
2. Connect to the EC2 instance Connect of ubuntu. 
3. Execute the “sudo apt-get update” for fetching all updates of Ubuntu.
4. Installing all the environment dependency for running flask application via Python.

* sudo apt-get install python-pip
* pip install flask
* sudo apt install python3-venv
* pip install pandas
* pip install Joblib
* pip install NumPy

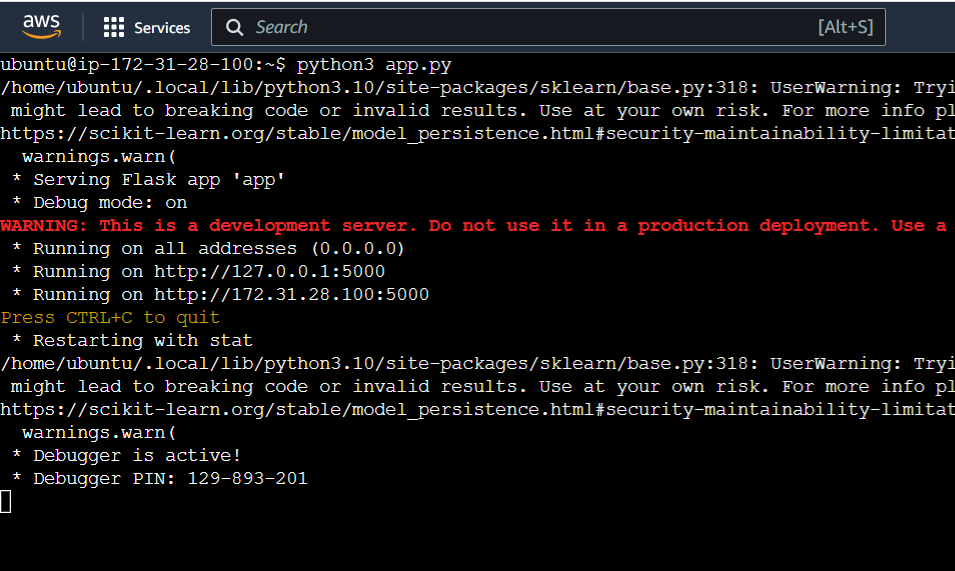
1. Once the dependencies are installed, we transfer the files using WinSCP software vis SFTP protocol and to do that first connect using the host name which is your IPV4 Public name mentioned in the Amazon AWS EC2 instance and Username for EC2 instance is always “ubuntu”,



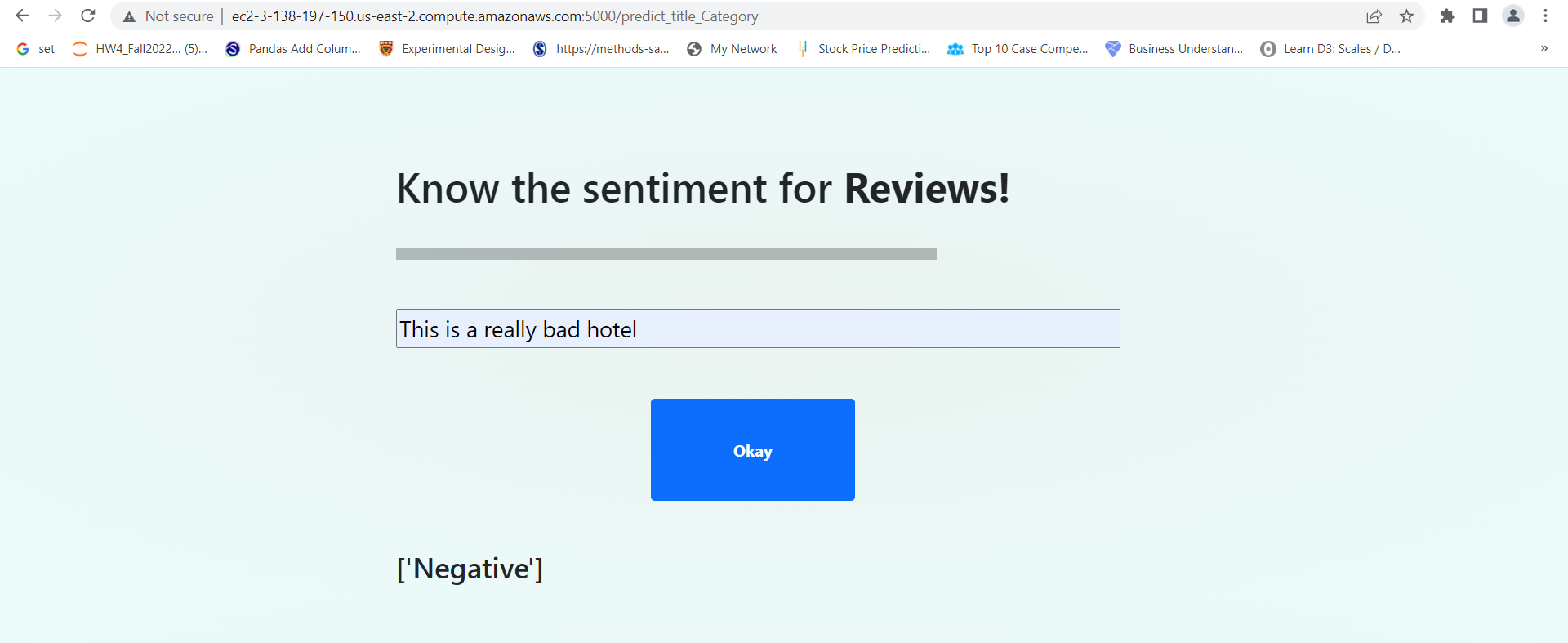
Transferring files to the Amazon EC2 instance in ubuntu OS.



1. Once all the files are transferred connect to SSH shell of the instance and run app.py it will start running the web application on the URL: Public IPv4:5000 port number.



1. After accessing the URL,a user gives any review text in the form such as “This is a really bad hotel” and the ML model predicts if it is a positive, negative or neutral review. For the frontend development HTML,CSS,JavaScript are used, and backend is supported by Machine learning model in “.pkl(pickle format)” while the middleware acts as a server which is implemented using Python Flask.



# **CONCLUSION**

Team learned and implemented various NLP techniques:

* 1. Bag of words
  2. TF-IDF
  3. Text classification
  4. Tokenization, Stemming, Lemmatization
  5. Sentiment analysis using Vader and Bert

Team also learned and deployed trained model using flask and Webapp on Amazon AWS

Following insights were generated during course of project:

1. Insights from wordcloud: The behavior of the customers in each city is different as the customers in Shanghai want convenient hotels while in Dubai people look for the outdoor view like beach.
2. TF-IDF/Bag of words: Analysis highlights the varying importance of amenities across different cities. For Delhi, location, cleanliness, staff, service, and food are the most crucial amenities, while Dubai reviews emphasize service, area, beach, and pool.
3. Text Classification also shows that cleanliness is talked about in most of the reviews, English being the primary language spoken seems to play a role in review multinomial logistic regression model accuracy.
4. Vader and Bert Sentiment analysis:
   1. Vader analysis combined with Gaussian Naïve Bayes classifier can be used to predict sentiment with 82% percentage accuracy.
   2. Bert Analysis is better in predicting sentiments scores compared to Vader. But this is a qualitative analysis through comparison of scores generated.
   3. Bert takes a long time to train therefore we cannot run it on our Booking.com dataset to provide a quantitative conclusion using reviewer scores.
5. A trained model from python can be exported and deployed using Webapp on Amazon AWS

# **FUTURE STEPS**

1. Run Bert model on Booking.com Dataset to provide quantitative conclusion on accuracy of the model for sentiment analysis.
2. Add n-gram classification using Python nltk package and time series for text classification to provide better insight into review context and timing, for text classification.

# **REFERENCES**

1. Trip Advisor Review Dataset. Retrieved from: <http://kavita-ganesan.com/entity-ranking-data/#.XQESU9NKgWq> on 5th March 2023.
2. What is natural language processing (NLP). Retrieved from: <https://www.ibm.com/topics/natural-language-processing> on 5th March 2023.
3. Natural Language Processing (NLP), What it is and why it matters. Retrieved from: <https://www.sas.com/en_ca/insights/analytics/what-is-natural-language-processing-nlp.html> on 5th March 2023.
4. Generating Word Cloud in Python. Retrieved from: [https://www.geeksforgeeks.org/generating-word-cloud-Python/](https://www.geeksforgeeks.org/generating-word-cloud-python/) on 5th March 2023.
5. F. Heimerl, S. Lohmann, S. Lange and T. Ertl, "Word Cloud Explorer: Text Analytics Based on Word Clouds," 2014 47th Hawaii International Conference on System Sciences, Waikoloa, HI, USA, 2014, pp. 1833-1842, doi: 10.1109/HICSS.2014.231. Retrieved from: <https://ieeexplore.ieee.org/document/6758829> on 5th March 2023.
6. Booking.com Review Dataset on Kaggle. Retrieved from: <https://www.kaggle.com/datasets/jiashenliu/515k-hotel-reviews-data-in-europe> on 19th March 2023.
7. TF-IDF [Tutorial] in Python. Retrieved from: <https://www.kaggle.com/code/paulrohan2020/tf-idf-tutorial> on 20th March 2023.
8. Sklearn feature extraction text and TfidfVectorizer. Retrieved from: <https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html> on 20th March 2023.
9. Hugging face pretrained models. Retrieved from: <https://huggingface.co/models> on 20th March 2023.
10. Hugging face transformer Bert models. Retrieved from: <https://huggingface.co/docs/transformers/model_doc/bert#transformers.BertModel> on 20th March 2023.
11. Bag of words (BoW) model in NLP. Retrieved from: [ttps://www.geeksforgeeks.org/bag-of-words-bow-model-in-nlp/](https://www.geeksforgeeks.org/generating-word-cloud-python/) on 5th March 2023.
12. Vader Sentiment 3.3.2. Retrieved from: <https://pypi.org/project/vaderSentiment/#:~:text=VADER%20(Valence%20Aware%20Dictionary%20and,on%20texts%20from%20other%20domains>. on 5th March 2023.