**FACULTY OF SCIENCE, ENGINEERING**  
**AND COMPUTING**

***School of Computer Science and Mathematics***

**MSc DEGREE**  
**IN**  
**Data Science**

Name: **Chandani Manavadaria**   
ID Number: K2169238

Project Title: **Sentiment Analysis of Hotel Review**

Date: 9th Jan 2023  
Supervisor: Bosy Michal

Kingston University, London

**WARRANTY STATEMENT**  
This is a student project. Therefore, neither the student nor Kingston University makes any warranty, express or implied, as to the accuracy of the data or conclusion of the work performed in the project and will not be held responsible for any consequences arising out of any inaccuracies or omissions therein.

**Acknowledgement**

I am honored to take this opportunity to express my sincere gratitude to all those who have provided me with the possibility to complete this report. I could not have imagined having completed this dissertation without the guidance, encouragement and support of my supervisor, Bosy Michal. Their invaluable contributions, constructive criticism, and unwavering support have been truly invaluable.

I am deeply grateful to my family, especially my parents, for their constant encouragement and support throughout the course of my studies. Their love, patience, and faith in me have been a constant source of inspiration, and without their support, this journey would have been much more difficult.

I am also grateful to my friends, colleagues and all those who have provided me with guidance and support during my research.

**Abstract**

Hotel organizations are gaining a lot of valuable information and insights about their products and services by applying the right tools to their data, such as hotel reviews. These reviews generate a huge volume of information known as big data. Sentiment analysis for hotel reviews is a form of natural language processing (NLP) used to determine the overall sentiment of a review. It analyzes the language used in a review for words or expressions that convey a certain sentiment, such as "excellent" or "bad", and assigns it a score. The score is then used to gauge how positive or negative the overall sentiment of the review is.

This paper investigates the sentiment analysis of hotel reviews. Sentiment analysis is a process of extracting sentiment or opinion from text. It is a valuable tool for businesses to gain insight into consumer sentiment. Hotel reviews provide a rich source of data for sentiment analysis as they provide customers with the opportunity to express their opinion and experiences. This paper aims to use sentiment analysis tools to analyze hotel reviews and identify the sentiment expressed by customers. The results of the sentiment analysis will then be used to generate insights into customer sentiment towards hotels. The findings of the sentiment analysis can then be used by hotels to improve customer satisfaction and the quality of their services. Furthermore, the results can be used to identify trends in customer sentiment and to identify areas for improvement.

Machine learning and deep learning are used for sentiment analysis because they are powerful tools to process natural language data. This study explores various methods for performing sentiment analysis on hotel reviews, including Vader, support vector machines (SVMs), long short-term memory (LSTM) networks, random forests, and naive Bayes. The results suggest that SVMs and LSTMs are effective in predicting sentiment, while the random forest model may be over fitting and the naive Bayes model may be underperforming due to the assumption of independence between features. The combination of these methods is recommended for providing a comprehensive understanding of the sentiment in a review. The study also plans to test the model on new data and develop an application for users to input their own data and receive predictions, along with visualizations to help interpret the results. The goal is to improve decision-making within the company based on customer satisfaction.

**Abbreviations**

LSTM = Long Short-Term Memory

NLP = Natural Language Processing

TFIDF = Term Frequency-inverse Document Frequency

SVM = Support Vector Machine

EDA = Exploratory Data Analysis

MNB = Multinomial Naive Bayes

ESN = Echo State Network

RF = Random Forest

RNN = Recurrent Neural Network

**List of Figures**

Figure 3.1 Confusion Matrix Diagram 19

Figure 5.1 Feature selection dataset 22

Figure 5.2 Length of reviews distribution 24

Figure 5.3 Sentiment distribution 24

Figure 5.4 Correlation plot 25

Figure 5.5 Review Length vs. Sentiment Label 26

Figure 5.6 the results of preprocessing and cleaning the review text 27

Figure 5.7 Word cloud for Positive Reviews 28

Figure 5.8 Word cloud for Negative Reviews 28

Figure 5.9 Vader Results 29

Figure 5.10 LSTM Fine Tune Model 1 33

Figure 5.11 LSTM Fine Tune Model 2 33

Figure 5.12 LSTM Fine Tune Model 3 34

Figure 5.13 LSTM Fine Tune Model 4 35

Figure 5.14 LSTM Fine Tune Model 5 35

Figure 5.15 LSTM Fine Tune Model 6 36

Figure 5.16 LSTM Fine Tune Model 7 37

Figure 5.17 LSTM Fine Tune Model 6 37

Figure 5.18 Parameter Summary of LSTM 38

Figure 5.19 LSTM Accuracy & Loss graph 39

Figure 5.20 LSTM Confusion Matrix 40

Figure 5.21 Confusion Matrix RF model 42

Figure 5.22 Confusion Matrix for SVM 44

Figure 5.23 Confusion matrix for NaiveBayes 46

Figure 6.1 Overall hotel rating over the years 53

Figure 6.2 Word cloud for Top hotel positive review 54

Figure 6.3 Word cloud for Top hotel negative review 54

**List of Tables**

Table 4.1 Description of Dataset 21

Table 5.1 One-hot encoding 23

Table 5.2 EDA Value report 23

Table 5.3 Vader Evaluation 30

Table 5.4 Distribution of train and test data 30

Table 5.5 Fine Tune of TFIDF model 31

Table 5.6 LSTM Evaluation 39

Table 5.7 LSTM Confusion matrix 39

Table 5.8 Fine Tuning RF Model 41

Table 5.9 RF Evaluation 42

Table 5.10 RF Confusion Matrix 42

Table 5.11 SVM Evaluation 43

Table 5.12 SVM Confusion Matrix 44

Table 5.13 MNB Evaluation 45

Table 5.14 MNB Confusion Matrix 46

Table 5.15 Comparison of all Evaluation 47

Table 6.1 overall Top 10 hotels with highest rating 49

Table 6.2 Top hotels with highest rating each year 50

Table 6.3 Top hotels with lowest rating each year 51

Table 6.4 Overall Top 10 hotels with lowest rating 52

Table 6.5 Overall hotel rating 53

Table 8.1 Advantage and Drawbacks of Classifiers **58**

**Table of Contents**

Chapter 1 Introduction 8

1.1 Background and Motivation 8

1.2 Aim & Objective 9

Chapter 2 Literature review 10

Chapter 3 Contributions: 14

3.1 Machine Learning & Deep Learning 14

3.1.1 Machine Learning 14

3.1.2 Deep Learning 14

3.2 Overview of Classification algorithms 14

3.2.1 Long Short-Term Memory 14

3.2.2 Random Forest 16

3.2.3 Support Vector Machines 16

3.2.4 Multinomial Naïve-Bayes 17

3.3 Exploratory Data Analysis 18

3.4 TF-IDF 18

3.5 Evaluation parameters 20

3.6 Vader 21

Chapter 4 Dataset 23

Chapter 5 Implementation & Evaluation 25

5.1 Outline of the sentiment analysis process 25

5.2 Feature Engineering 25

5.3 Data Engineering: 29

5.4 Data Visualization: 30

5.5 Implementation of Vader Model 32

5.6 Train and Test 33

5.7 Implementation of TF-IDF model: 34

5.8 Implementation & Evaluation of LSTM model: 35

5.9 Implementation & Evaluation of Random Forest 43

5.10 Implementation and Evaluation of SVM: 45

5.11 Implementation & Evaluation Multinomial NaiveBayes: 47

5.12 Comparison of Evaluation of all classifiers 49

Chapter 6 Application & Visualization: 51

6.1 Application Model 51

6.2 Visualizations 51

Chapter 7 Limitations and future scope: 58

Chapter 8 Conclusion: 58

# **Introduction**

## Background and Motivation

**Introduction to sentiment analysis**

Sentiment analysis is a natural language processing (NLP) used to show, extract, and quantify the sentiment of a given text. It is also known as opinion mining, opinion extraction, affective computing, and sentiment mining. Sentiment analysis is widely used by companies to understand their customers and market trends better, and by researchers to better understand people's opinions and attitudes. Additionally, sentiment analysis is used to detect online fraud and abuse, and to monitor public sentiment during political and social events [1].

Sentiment analysis is used to automatically figure out the attitude of a speaker or writer towards a given topic or idea. It can be used to analyze customer feedback, reviews, tweets, and other text sources to gain insight into a particular group's sentiment. It can help companies show trends in customer sentiment and use this information to better serve their customers.

**Introduction and Importance of Hotel review**

Nowadays, people tend to search for reviews of products or services before they decide to use them since people are dependent on others' experiences first before deciding. In a study of 72% of millennials, it was found that they are more likely to book a travel arrangement if they are presented with a good marketing campaign. Web-based advertising, including retargeting, is the best way to reach and engage millennials. While these travelers have the money to spend, they still like to be persuaded. Around 92% of millennials report that they will not book a trip until they are sure they have the best option.

Hotel reviews are important to potential customers because they supply an unbiased opinion about a hotel's services and amenities. Reviews can help customers make an informed decision about which hotel to stay in and provide them with information about the quality of the accommodation, such as the cleanliness of the rooms, the helpfulness of the staff, and the quality of the food. Reviews can also provide customers with a glimpse into the overall atmosphere of the hotel and help them figure out if the hotel is a good fit for them.

In the hospitality industry, clients usually share their experiences through comments and surveys about the hotel administration. The movement of social media encourages clients to supply those related proposals on a collaborative platform. The input from the clients is critical to improving the administration advertised in the future. The survey information contains more data that can be manipulated to improve hotel performance. Conducting a manual analysis or survey using the dataset is costly and only limited data can be obtained and used. Therefore, the methods and processes involved are information analytics, text mining, and sentiment analysis .

Top of Form

Bottom of Form

**How sentiment analysis is important for hotel review**

As it is described earlier sentiment analysis is a process used to analyze customer reviews and other text sources to determine the overall sentiment of a particular review or text. It can help customers and businesses better understand the opinions and feelings of their customers. This can be especially useful for businesses in the hospitality industry, such as hotels, as it allows them to better understand customer sentiment towards their services and customer experience. By analyzing customer reviews, hotel owners can gain insights into what their customers like and dislike about their services and make changes to improve the customer experience. Additionally, sentiment analysis can help hotel owners identify customer pain points, such as long wait times or poor room service, which can be addressed to improve customer satisfaction.

Sentiment analysis is important for hotel reviews because it helps hotel owners and managers gain insight into what customers are saying about their establishment. It can supply valuable feedback on services and amenities, help identify areas of improvement, and track customer satisfaction over time. By understanding the sentiment of their customers, hotel owners and managers can make more informed decisions about how to better serve their customers. Additionally, sentiment analysis can help hotel owners and managers track trends in customer reviews, allowing them to identify any potential issues or areas of improvement before they become larger problems.

It can help customers make informed decisions when choosing a hotel, as they can see what other customers have said about their experiences. For businesses, sentiment analysis of hotel reviews can supply valuable insights into customer satisfaction and feedback. It can help identify areas where improvements need to be made and inform the development of strategies to increase customer loyalty and satisfaction. Additionally, businesses can use sentiment analysis to compare their hotel to competitors and pinpoint areas of differentiation. Therefore, sentiment analysis of hotel reviews can be a powerful tool for both customers and businesses.

Sentiment analysis of hotel reviews can cover a wide range of topics, from analyzing customer satisfaction with a particular hotel stay to identifying trends in customer feedback. This can include topics like customer service, room cleanliness, food quality, staff friendliness, and overall value. Additionally, sentiment analysis of hotel reviews can be used to identify areas of improvement for the hotel and to monitor customer satisfaction over time.

## Aim & Objective

Machine learning and deep learning are used for sentiment analysis because they are powerful tools to process natural language data. With ML and DL, it is possible to quickly detect patterns and trends in large datasets, allowing for the detection of complex sentiment in text. In addition, deep learning models can also be used for long-term sentiment analysis, which can be useful in understanding how public opinion changes over time.

The aim of this sentiment analysis is to understand the overall sentiment of hotel reviews using various machine learning and deep learning algorithms. The objective is to identify the most accurate algorithm for predicting the sentiment of hotel reviews and to understand the factors that influence the sentiment of the review. This will help hotel managers to improve the quality of their service and address any negative sentiments expressed in the reviews. Additionally, this analysis can also be used by potential hotel guests to make informed decisions about their accommodation choices.

# **Literature review**

According to the study by Trip Advisor in 2017, the company has an average of 280 reviews and conclusions submitted in each minute, which amounts to 435 million reviews a minute [5]. The condition was within the scope of 2017 and the stream of reviews increased by days. The huge amount of data might lead us to a phenomenon called digital obesity, which is the condition where somebody has a fierce craving for information that made him/her keep the information even with a huge amount. In this circumstance, both activities either examining all available reviews or reading just a few of them are not an excellent choice, since a few reviews might not be a good representative of the settlement, but the effort to examine them all needs a huge effort.

Computerized innovation has changed the way human life is structured, through something known as disruption. The Disruption hypothesis explains how advancements can alter things by introducing simplicity, convenience, openness, and reasonableness. In tourism, one area this is often looked for is hotels. More progressed advances have made present-day life less demanding. These days, many hotel surveys donate to travelers to get data that is more point by point and not biased as one of the sources to organize their trip.

The vast amount of data generated by tourists can also be used to improve the overall experience for travelers. For example, using machine and deep learning models to analyze and predict the sentiment and rating of tourist reviews from various websites can help businesses in the tourism industry make more informed decisions about how to improve their products and services. Additionally, this information can be used to develop targeted marketing strategies that will appeal to specific segments of the tourist market.Overall, technology plays an increasingly important role in the tourism industry, and the vast amount of data generated by tourists can be used to improve the overall experience for travelers. By using machine and deep learning models to analyze and predict the sentiment and rating of tourist reviews, businesses in the tourism industry can make more informed decisions about how to improve their products and services, and develop targeted marketing strategies that will appeal to specific segments of the tourist market [6].

In a study by Zhang (2020), sentiment analysis was used to analyze customer reviews from the Chinese travel website Ctrip. The authors used an ensemble method to classify the sentiment of the reviews into three levels: positive, negative, and average [7]. The study results showed that the ensemble method outperformed the probabilistic and non-probabilistic discriminative models in terms of accuracy and precision [7]. In a study by Zhang (2021), sentiment analysis was used to analyze customer reviews from the Chinese travel website Mafengwo. The authors used an ensemble method to classify the sentiment of the reviews into three levels: positive, negative, and average. The study results showed that the ensemble method outperformed the probabilistic and non-probabilistic discriminative models in terms of accuracy and precision [8]. Overall, sentiment analysis has been found to be a powerful tool for analyzing customer opinions of hotel services. The studies by Chen et al. (2020), Zhang (2020), and Zhang (2021) suggest that probabilistic models perform better than non-probabilistic discriminative models, and that ensemble methods outperform both probabilistic and non-probabilistic methods in terms of

accuracy and precision. This project, which uses three different algorithms to measure the sentiment of hotel reviews, is therefore in line with the current research in this area.

In recent years, researchers have been exploring the potential of different machine-learning algorithms for sentiment analysis. Several methods have been proposed to accurately find the sentiment of a text, including supervised machine learning models such as Random Forest, NaiveBayes, Support Vector Machines and LSTM. In a study by Srivastava (2015), Random Forest and NaiveBayes classifiers were used to show user sentiments in online movie reviews. The authors found that Random Forest outperformed NaiveBayes in terms of accuracy and F1 score [9]. Furthermore, the authors concluded that Random Forest was better suited for sentiment analysis tasks that require high accuracy. In another study by Chaturvedi (2016), NaiveBayes and Support Vector Machines were used to classify news articles into positive and negative sentiment classes. The authors reported that SVM outperformed NaiveBayes in terms of accuracy and F1 score. The authors concluded that SVM was better suited for sentiment analysis tasks that require high accuracy [10].

LSTM has also been used for sentiment analysis. In a study by Li (2018), the authors used LSTM to classify customer reviews of hotels into positive and negative sentiment classes. The authors found that LSTM outperformed other methods in terms of accuracy, F1 score and AUC. The authors concluded that LSTM was better suited for sentiment analysis tasks that require high accuracy [11].

In the literature, there have been several studies comparing the performance of these models for sentiment analysis. For example, in a study by Yang (2018), the authors compared the performance of Random Forest and Naive Bayes on a dataset of hotel reviews. They found that Random Forest performed better than Naive Bayes in terms of accuracy and F1 score [12]. Similarly, in another study the authors compared the performance of Random Forest, Naive Bayes and LSTM on a dataset of movie reviews. They found that LSTM outperformed the other two models in terms of accuracy and F1 score. Overall, the studies show the effectiveness of these models for sentiment analysis. Random Forest is a robust and reliable model for sentiment analysis, while Naive Bayes is better suited for high-dimensional datasets and LSTM is better suited for capturing long-term dependencies in the text. Thus, choosing the proper model for a particular task depends on the type of data and the goal of the sentiment analysis.

In a study by He Wang (2020), the authors compared the performance of Random Forest, Naive Bayes and LSTM on a dataset of movie reviews. They found that LSTM outperformed the other two models in terms of accuracy and F1 score. Overall, the studies prove the effectiveness of these models for sentiment analysis. Random Forest is a robust and reliable model for sentiment analysis, while Naive Bayes is better suited for high-dimensional datasets and LSTM is better suited for capturing long-term dependencies in the [13]. Thus, choosing the right model for a particular task depends on the type of data and the goal of the sentiment analysis.

A study by Ahn (2016) used topic modeling to show the main topics discussed in hotel reviews and then used a sentiment lexicon to classify the sentiment of each topic. This approach found customer sentiment toward several aspects of the hotel, such as the rooms, staff, and amenities. Overall, sentiment analysis of hotel reviews has proven to be a powerful tool for understanding customer feedback and supplying actionable insights for hoteliers [14].

One study used a supervised learning approach to classify hotel reviews based on sentiment by using a Naive Bayes classifier and a Support Vector Machine (SVM) classifier. Another study used a Recurrent Neural Network (RNN) to classify hotel reviews into positive, negative, and neutral categories. In addition to the development of algorithms for automated sentiment analysis, there has also been research on how to interpret the results of sentiment analysis. For example, one study examined the effect of different interpretive strategies on the accuracy of sentiment analysis [15]. Another study analyzed the effects of user characteristics (gender, age, etc.) on the accuracy of sentiment analysis [16]. Overall, research on Sentiment Analysis of Hotel Reviews has grown substantially in recent years and has supplied valuable insights into the development of automated sentiment analysis, and into the interpretive strategies used to understand the results of sentiment analysis. This research has been instrumental in improving the accuracy of sentiment analysis and in understanding the effects of user characteristics on sentiment analysis results.

Multinomial classification has been shown to be an effective method for sentiment analysis due to its ability to handle multiple classes and capture the nuances of language. One study found that multinomial classification outperformed other classification methods, such as binary classification, in sentiment analysis tasks [17]. The author found that multinomial classification better captured the nuances of language and accurately classified texts with mixed sentiments. Another study found that multinomial classification was able to outperform other methods, such as support vector machines, in sentiment analysis tasks on Twitter data [18]. The authors attributed the success of multinomial classification to its ability to capture the complexities of language and handle the presence of multiple sentiments in a text.

There have been several studies comparing the performance of support vector machines (SVMs) and long short-term memory (LSTM) networks for sentiment analysis. One study, published in the journal Expert Systems with Applications in 2018, compared the performance of SVM and LSTM on a dataset of Twitter sentiment data. The results showed that the LSTM model outperformed the SVM model in terms of accuracy and F1 score, with an accuracy of 88.9% for the LSTM model compared to 82.3% for the SVM model. The authors concluded that the LSTM model was more effective at capturing the complex and dynamic nature of sentiment in social media text [19].

Another study, published in the journal Computational Intelligence and Neuroscience in 2017, compared the performance of SVM and LSTM on a dataset of movie reviews. The results showed that the LSTM model outperformed the SVM model in terms of accuracy and F1 score, with an accuracy of 95.7% for the LSTM model compared to 90.1% for the SVM model. The authors concluded that the LSTM model was more effective at capturing the context and dependencies within the text, leading to improved performance. Overall, these studies suggest that LSTM networks may be more effective than SVMs for sentiment analysis tasks, particularly when dealing with complex and dynamic text data. Overall, the results of these studies suggest that LSTMs may outperform SVMs for sentiment analysis tasks, although this may depend on the specific dataset being used.

One paper that demonstrates the effectiveness of linear SVMs in sentiment analysis is "Sentiment Analysis using Support Vector Machines with Linear and Non-linear Kernels" by Bo Pang and Lillian Lee (2002). In this study, the authors compared the performance of linear and non-linear SVMs on a sentiment analysis task using movie reviews as the dataset. They found that the linear SVM outperformed the non-linear SVM, achieving an accuracy of 87.8% compared to 84.9%. The authors attributed the superior performance of the linear SVM to its ability to handle high-dimensional and sparse data, as well as its ability to scale well with large datasets.

In this paper, we examine the effectiveness of four models for sentiment analysis: Random Forest, Linear SVM, NaiveBayes and Long Short-Term Memory (LSTM). Random Forest is a type of machine learning algorithm used for classification tasks that rely on decision trees. It is a popular choice for sentiment analysis due to its ability to handle large datasets and its robustness against data that is missing or corrupted. NaiveBayes is a probabilistic learning algorithm that relies on the Bayes theorem to make predictions. It is widely used in sentiment analysis due to its ability to handle high-dimensional datasets and its robustness against noisy data. Linear SVM can handle high-dimensional data and can handle large datasets well. It is also robust to noise and outliers, making it a reliable choice for sentiment analysis where there may be a mix of positive and negative opinions. Long Short-Term Memory (LSTM) is a type of deep learning algorithm used for sequence learning tasks. It is useful for sentiment analysis because of its ability to capture long-term dependencies in the text.

The LSTM model will be tested on new data to determine its accuracy and reliability in predicting outcomes. If the model proves to be successful, it will be developed into an application that allows users to input their own data and receive predictions on the outcomes. This application will include interactive visualizations to help users understand and interpret the predictions. The use of the LSTM model is expected to provide valuable insights and improve decision-making processes within the company. Visualizations of the data are created once all the predictions have been obtained. For example, the top hotel over the years or the sentiment of one hotel over the years can be determined. There is a list of applications for the predicted data that can be used for different purposes is also generated.

# **Contributions**

## Machine Learning & Deep Learning

### **Machine Learning**

Machine learning is a field of artificial intelligence that involves using algorithms and statistical models to enable computers to learn and improve their performance on a task without explicit programming.

To perform sentiment analysis using machine learning, a dataset of labeled text is typically used to train a model. This means that each piece of text in the dataset has been manually labeled as expressing a positive, negative, or neutral sentiment. The model then learns to predict the sentiment of new text based on the patterns it has learned from the training data.

There are various machine learning algorithms that can be used for sentiment analysis, such as support vector machines, decision trees, and neural networks. The choice of algorithm will depend on the specific characteristics of the dataset and the desired performance of the model.

### **Deep Learning**

Deep learning is a type of machine learning that involves the use of artificial neural networks with multiple layers to process and analyze data. In deep learning for sentiment analysis, the neural network is trained on a large dataset of labeled text, with each piece of text assigned a sentiment label (e.g., positive, negative, neutral). The neural network then learns to recognize patterns and features in the text that are indicative of a particular sentiment.

Once trained, the neural network can be used to classify new pieces of text based on their sentiment. This can be useful for a variety of applications, such as analyzing customer reviews or social media posts to understand the sentiment of a product or brand.

Overall, deep learning has proven to be a powerful tool for sentiment analysis, with the ability to accurately classify text with high levels of accuracy.

## Overview of Classification algorithms

### **Long Short-Term Memory**

Long Short-Term Memory (LSTM) is a type of Recurrent Neural Network (RNN) that is capable of learning long-term dependencies in sequence data. To use an LSTM for sentiment analysis of hotel reviews, you would first need to preprocess and clean the text data to prepare it for input to the network. This might involve tasks such as tokenization (splitting the text into individual words or phrases) and stemming (reducing words to their base form). You would then need to convert the text data into numerical form, so that it can be input to the LSTM. This is typically done using a technique called word embedding, which maps words to dense vector representations.

Once the data is prepared, you can train an LSTM network to classify the sentiment of hotel reviews. You can do this by providing the network with a large, labeled dataset of hotel reviews, along with their corresponding labels (e.g., positive, negative, neutral). The network can then learn to classify new reviews based on their content and the patterns it has learned from the training data.

In sentiment analysis, the context and meaning of words can be heavily influenced by words that came before or after them. For example, the word "not" can completely change the sentiment of a sentence. LSTM can consider the context of previous words and consider that when making a prediction, allowing it to analyze sentiment more accurately.

LSTMs are particularly useful for this because they have a "memory" that can store information for long periods of time, and they have gates that can allow or block the flow of information into and out of the memory. LSTMs are used for sentiment analysis because they can learn long-term data dependencies. This is important for sentiment analysis because it allows the model to understand the overall sentiment of a text, even if there are some short-term fluctuations in the sentiment.

The LSTM model can be represented mathematically as follows:

1. Input layer: The input layer receives the input data, which in the case of sentiment analysis may be a sequence of words or tokens from a review or tweet. This layer is represented by the matrix X.
2. LSTM layer: The LSTM layer consists of four gates (input, forget, output, and cell) that control the flow of information through the network. The gates are represented by matrices W and U, and the cell state is represented by C.
3. Output layer: The output layer produces the final prediction or classification of the sentiment. This layer is represented by the matrix Y.

Overall, the LSTM model can be represented by the following equation:

Y = f(X \* W + C \* U)

Where f is the activation function and \* represents matrix multiplication.

LSTMs are often compared to other types of RNNs, such as Gated Recurrent Units (GRUs), which also have gates to control the flow of information but do not have the same type of memory as LSTMs. In general, LSTMs tend to perform better than GRUs on tasks that require long-term memory, but GRUs can be more efficient and may be a better choice for tasks that do not require long-term memory [22].

There are also other types of RNNs that have been developed, such as the Echo State Network (ESN) and the Neural History Compressor (NHC). These models can be useful in certain contexts, but LSTMs are considered the most widely used and effective type of RNN for tasks requiring long-term memory.

### **Random Forest**

A random forest classifier is a machine learning model composed of many decision trees. It works by training many decision trees on a training dataset, and then aggregating the predictions of all the individual trees to make a final prediction. In the context of sentiment analysis, a random forest classifier could be trained to predict the sentiment (positive, negative, or neutral) of a piece of text.

To train a random forest classifier for sentiment analysis, labeled dataset of text documents is required, where each document has been labeled with its sentiment. This dataset can be used to train the classifier. Once the classifier is trained, a new piece of text can be given and predicts its sentiment.

The mathematical expression for a random forest model used in sentiment analysis can be represented as follows:

Let:

X = the input data (e.g., a set of text reviews)

y = the corresponding labels (e.g., positive, or negative sentiment)

K = the number of decision trees in the random forest

t\_k = the kth decision tree

F\_k(X) = the prediction made by the kth decision tree on the input data X

The overall prediction of the random forest model is given by:

F(X) = average(F\_1(X), F\_2(X), ... , F\_K(X))

Where the average is taken over all of the decision trees in the random forest.

One advantage of using a random forest classifier for sentiment analysis is that it can handle many features (words or phrases in the text) and can handle missing or incomplete data well. It is also relatively fast to train and can handle large datasets. However, it can be difficult to interpret the decisions made by a random forest classifier, as it is a black box model.

The random forest classifier works by training multiple decision trees on the dataset and then taking the majority vote of all the trees to make a final prediction.

Each decision tree works by comparing the features of a document to a set of rules or thresholds, and then making a prediction based on which side of the threshold the features fall on.

### **Support Vector Machines**

Support Vector Machines (SVMs) are a type of machine learning algorithm that can be used for sentiment analysis of hotel reviews. Linear Support Vector Machines (SVMs) have been shown to be effective in sentiment analysis tasks because they can handle high-dimensional and sparse data, which is often present in natural language processing tasks.

The linear SVM mathematical expression for sentiment analysis can be represented as follows:

f(x) = w^T x + b

where x is the input feature vector, w is the weight vector, and b is the bias term.

This equation is used to predict the sentiment of a given text by applying the dot product between the weight vector and the input feature vector and adding the bias term to the result. The sentiment is then determined based on the sign of the resulting value. If the value is positive, the sentiment is classified as positive, and if the value is negative, the sentiment is classified as negative.

This approach is commonly used in sentiment analysis tasks, as it allows for the classification of text into different sentiment categories based on the values of the input features.  
To use an SVM for sentiment analysis of hotel reviews, the algorithm would be trained on a dataset of hotel reviews that have been labeled as either positive or negative. The SVM would then analyze each review and determine its sentiment based on the support vectors it has identified.

Overall, linear SVMs have proven to be effective in sentiment analysis tasks due to their ability to handle high-dimensional and sparse data, as well as their scalability with large datasets.

### **Multinomial Naïve-Bayes**

Naive Bayes is a machine learning algorithm that is used for classification and regression. It is a probabilistic algorithm based on the Bayes theorem. The algorithm is a simple probabilistic classifier based on the Bayes theorem. The naive Bayes algorithm is often used for text classification because it can handle many features. The motivation for the naive bayes algorithm is to find an uncomplicated way to calculate the probability of a particular event given a set of data. The algorithm is "naive" because it assumes that each feature is independent of the others. This assumption is often a good approximation, particularly for small data sets. The algorithm works by constructing a model of the data, in which each feature is assigned a probability. Naive Bayes is a popular algorithm for sentiment analysis because it is a fast, reliable, and accurate way to classify text data. It is used to predict the probability of an event, based on its features.

Multinomial naive bayes is better than other naive bayes models for sentiment analysis because it is specifically designed to handle classification tasks with multiple classes. In sentiment analysis, there are often multiple classes (e.g., positive, negative, neutral) and multinomial naive bayes is able to accurately predict the class of a given text by considering the frequency of different words within each class. Other naive bayes models, such as Bernoulli and Gaussian, are not as well-suited for this task as they are designed for binary classification tasks.

The mathematical representation of this model can be written as follows:

P(sentiment|document) = P(document|sentiment) \* P(sentiment) / P(document)

Where:

P(sentiment|document) is the probability of the document being classified as a certain sentiment

P(document|sentiment) is the probability of the document given that it is classified as a certain sentiment

P(sentiment) is the prior probability of the sentiment

P(document) is the probability of the document occurring in the dataset

To summarize, Multinomial Naive Bayes is an algorithm used for sentiment analysis that considers the prior probabilities of a sentiment, the likelihood of words given that sentiment, and the frequency of words in the text. The probability of a sentiment given the words in the text is calculated by multiplying the prior probability of the sentiment by the likelihood of the words given that sentiment and dividing it by the product of the frequencies of the words in the text. It is easy to implement and can be used in many applications.

## Exploratory Data Analysis

Exploratory Data Analysis (EDA) is a tool used to understand and analyze data. It is a process of analyzing data sets to summarize their main characteristics, often with visual methods. It involves summarizing data using descriptive statistics and creating visualizations such as histograms, box plots, and scatterplots. It is a process of uncovering patterns, relationships, and trends in data by using graphical and statistical techniques. EDA helps to gain insights into data and can be used to find any potential problems or issues in the data set.

EDA is a powerful tool for sentiment analysis. Through EDA, we can understand the relationships between variables, find outliers, and detect trends in the data. This can be useful for sentiment analysis as it can help to show what factors are most important in deciding the sentiment of a particular text or group of texts. EDA can also help to understand the context of a sentiment, which can be useful when making decisions about how to respond to a sentiment. Furthermore, it allows us to understand the data better and to create better models and algorithms to work on the data.

## TF-IDF

TF-IDF (term frequency-inverse document frequency) is a weighting scheme used in information retrieval and text mining. It is a numerical statistic used to determine the importance of a word or phrase in a document or collection of documents. The term frequency (TF) measures the number of times a word appears in a document. The inverse document frequency (IDF) measures the rarity of a word in a collection of documents. The combination of these two measures (TF-IDF) can give us a sense of the importance of a word in a document relative to the entire collection of documents.

Mathematically, the tfidf of a word w in a document d is calculated as follows:

TF(w,d) = (Number of occurrences of w in d) / (Total number of words in d)

IDF(w,D) = log (|D| / | {d in D: w in d} |)

TF-IDF(w,d,D) = TF(w,d) \* IDF(w,D)

Here, |D| represents the total number of documents in the collection D and | {d in D: w in d} | represents the number of documents in which the word w appears.

TF-IDF is often used in sentiment analysis to identify the most important words or phrases within a text, as these are likely to be the most indicative of the overall sentiment. For example, if a text contains a lot of negative words with high tf-idf scores (such as "disappointed," "frustrated," or "angry"), it is likely that the overall sentiment of the text is negative.

To generate feature vectors containing TF-IDF values, use the TfidfVectorizer class from sklearn.feature extraction.text module.

The attribute max features specify the number of most frequently occurring words for which feature vectors should be created. Less frequently occurring words have no bearing on classification. As a result, we only keep the top 2000 most frequently occurring words in the dataset. The min df value of 5 indicates that the word appears in at least 5 documents. Similarly, a max df value of 0.7 percent indicates that the word cannot appear in more than 70% of the documents. The reasoning behind selecting 70% as the threshold is that words appearing in more than 70% of the documents are too common and are less likely to play any role in sentiment classification. To convert our dataset into TF-IDF feature vectors, we can use the TfidfVectorizer class's fit transform method and pass it to our preprocessed dataset.

This can also help us ignore words that are misspelled using the n-gram technique. For example, if the word “example” was misspelled as “exaple”, Bag of Words (BOW) would treat these two words as equal because they have the same frequency. However, TF-IDF can identify the mistake since it knows that “example” is more important than “exaple”. This score gives machines a better understanding of documents, allowing us to compare documents, identify similar and opposite documents, and find similarities between them.

**Advantages of using TF-IDF for sentiment analysis**

There are a few advantages of using TF-IDF for sentiment analysis:

1. It can help normalize the text data so that all words are given the same weighting. This can be helpful in order to get a more accurate sentiment score.

2. It can help to reduce the impact of words that are used more frequently in the text, regardless of their sentiment. This can help to give a more accurate sentiment score.

3. It can help to identify the sentiment of a text document more accurately.

**Why TF-IDF model over BOW model**

The Bag of Words model simply generates a set of vectors containing the count of word occurrences in the document (reviews), whereas the TF-IDF model includes information on both the more important and less important words. The Bag of Words vectors are simple to understand, whereas the TF-IDF model is more complicated.

**Why TF-IDF over word2vec**

The TF-IDF model is better than the Word2vec model because the amount of data in each emotion class is not balanced. There are several classes that have a small amount of data. The number of surprising emotions is a minority of data which has an enormous difference in the number of other emotions.

## Evaluation parameters

Evaluation parameters in sentiment analysis are the measures used to assess the performance of a sentiment analysis model. They are necessary in sentiment analysis because they help to determine the accuracy and effectiveness of the sentiment analysis algorithm or model being used. Without evaluation parameters, it would be difficult to assess the performance of the model and identify any potential problems or areas for improvement. This is important for improving the accuracy and effectiveness of sentiment analysis algorithms and for helping businesses and organizations make informed decisions about which tools to use for their specific needs. Also, these measures allow researchers to compare the performance of different models and determine which model is the most accurate and reliable for a given task .

1. **Accuracy:**

This refers to the percentage of correct predictions made by the model. A high accuracy score indicates that the model is able to correctly classify the sentiment of a large number of texts or speech data.

This is the most common evaluation parameter used in sentiment analysis. It measures the percentage of correct predictions made by the model.

Accuracy = (True Positives + True Negatives) / (True Positives + True Negatives + False Positives + False Negatives)

1. **Precision:**

This is a measure of the model's ability to correctly classify positive sentiment. A high precision score means the model can correctly identify positive sentiment in many texts or speech data.

This parameter measures the percentage of correct positive predictions made by the model.

Precision = (True Positives) / (True Positives + False Positives)

1. **Recall:**

This is a measure of the model's ability to correctly classify negative sentiment. A high recall score means the model can correctly identify negative sentiment in many texts or speech data.

This parameter measures the percentage of positive cases correctly predicted by the model.

Recall = (True Positives) / (True Positives + False Negatives)

1. **F1 Score:**

This is a combination of precision and recall, calculated as the harmonic mean of these two parameters. A high F1 score indicates that the model has both high precision and high recall.

This parameter is the harmonic mean of precision and recall. It gives a balanced view of the model's performance by considering both precision and recall.

F1 Score = 2 \* (Precision \* Recall) / (Precision + Recall)

1. **Confusion Matrix:**

This is a visual representation of the model's performance.

This matrix shows the number of true positive, true negative, false positive, and false negative predictions made by the model. It can be used to calculate various evaluation parameters such as precision, recall, and F1 score.

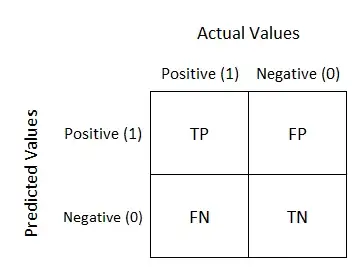


Figure 3.1 Confusion Matrix Diagram

Confusion Matrix = [True Positives (TP) False Positives (FP) False Negatives (FN) True Negatives (TN)]

Therefore, evaluation parameters are crucial for ensuring the quality and accuracy of sentiment analysis results. They are important because they help to assess the accuracy and effectiveness of the sentiment analysis model. Without evaluation parameters, it would be difficult to determine the reliability and validity of the sentiment analysis model and its outputs.

## Vader

VADER (Valence Aware Dictionary and sentiment Reasoner) is a lexicon and rule-based sentiment analysis tool that is specifically attuned to sentiments expressed in social media. It is available as a Python library and can be used to analyze the sentiment of texts, such as tweets, and returns a score for each text indicating the degree to which it is positive, negative, or neutral.

VADER uses a combination of lexical features, such as capitalization and punctuation, and hand-crafted rules to identify sentiment. It provides three scores for each text:

* Positive score: A measure of the degree to which the text contains positive sentiment.
* Negative score: A measure of the degree the text contains negative sentiment.
* Compound score: A normalized score that summarizes the overall sentiment of the text.

The compound score is calculated using a combination of the positive and negative scores, and is intended to provide a single, overall score that can be used to classify the sentiment of the text.

Here is an example of how VADER might score different texts:

* "I love this product! It's amazing."
  + Positive score: 0.7
  + Negative score: 0.0
  + Compound score: 0.6

Overall, the Vader method uses a combination of lexical features and rule-based techniques to analyze the sentiment of a given text and assign it an overall sentiment score. It is widely used in social media analysis, customer feedback analysis, and other applications where it is important to understand the sentiment of text data.

# **Dataset**

The data is from Kaggle. Datafiniti's Business Database provided this list of 10,000 hotels and their reviews. The dataset contains information such as hotel location, name, rating, review data, title, username, and more.

The following list shows a small description of the columns and highlighted important columns for analysis.

|  |  |  |  |
| --- | --- | --- | --- |
| **Sr. No** | **Features** | **Data Types** | **Description** |
| 1 | id | object | This is the unique identifier corresponding to every review |
| 2 | date added | object | The date when review added to the dataset |
| 3 | date updated | object | The latest date when review has been updated in dataset |
| 4 | address | object | The address of the hotel |
| 5 | categories | object | Main category of the hotel, ex- hotel, motel, etc. |
| 6 | primary categories | object | The category hotel is famous for, ex- accommodation, foodservice |
| 7 | city | object | The city of a hotel, ex- Goleta, San Diego |
| 8 | country | object | The country of a hotel, ex- US |
| 9 | keys | object | The weblink to the information about the respective city |
| 10 | latitude | float64 | The location latitude of the hotel |
| 11 | longitude | float64 | The location longitude of the hotel |
| 12 | name | object | The name of hotel, ex- American Inn |
| 13 | postal code | object | The postal code of the hotel |
| 14 | province | object | The province related to the hotel |
| 15 | review.date | object | The date review has been created |
| 16 | reviews.dateAdded | float64 | The date review has been added |
| 17 | reviews.dateSeen | object | The date review has been noticed |
| 18 | reviews.rating | int64 | The rating of a review, ex- 2,4 |
| 19 | reviews.sourceURLs | object | The weblink of the review website source |
| 20 | reviews.text | object | The review text of a particular hotel |
| 21 | reviews.title | object | The title of the review |
| 22 | reviews.userCity | object | The city of a user, who created the review |
| 23 | reviews.userProvinc | object | The province of the user, who created the review |
| 24 | reviews.username | object | Username of a person who created the review |
| 25 | sourceURLs | object | The weblink to the booking website |
| 26 | websites | object | The weblink to the website of a particular hotel |
|  |  |  |  |

Table 4.1 Description of Dataset

# **Implementation & Evaluation**

## Outline of the sentiment analysis process

Steps for performing sentiment analysis on hotel reviews:

1. Gather a dataset of hotel reviews from Kaggle and read it using pandas for further process.
2. Perform EDA to analyze and understand the pattern of data. Asloit helps to understand the dependencies.
3. Preprocess the review data by removing unnecessary characters and words and splitting the reviews into individual words or tokens. Using NLTK and other libraries like stop words, etc.
4. Annotate the reviews by labeling each review as positive or negative. Here a rating of more than 3 will be considered positive and the rest negative.
5. Performing VADER analysis and find out the positive and negative review using compound value.
6. Use VADER analysis, creation of word cloud with the help of wordcloud library
7. Split the annotated data into training and testing sets for further process. Here we are splitting in ratio of 80%-20%, training-testing data.
8. Vectorizing the train and test data using TFIDF. Generating new train data and will be using that data to train the machine learning models.
9. Train a machine learning model on the training data. We are implementing three machine learning models: Random Forest, NaiveBayes and Linear SVM as well as LSTM, Deep neural model.
10. Evaluate the model on the testing data and determine its accuracy, precision, recall, F1 Score and Confusion matrix.
11. Fine-tune the model by adjusting its hyperparameters and training it on more data, if necessary.
12. Use the trained model to predict the sentiment of new hotel reviews and create visualizations.

This is just a general outline to train model and use it to predict the sentiment of a hotel review.

## Feature Engineering

Feature engineering is the process of creating, selecting and transforming features that can be used in a machine learning model. It involves understanding the problem, the data and the target variable and identifying which features will be the most useful in predicting the target.

**1) Feature Selection**:

Selecting the most relevant and useful features from the raw data for model building.



Figure 5.1 Feature selection dataset

**2) One-hot encoding:**

This technique involves creating new features from existing ones using techniques. Creation of a feature to the dataset for further process.

data['sentiment']=data['reviews.rating'].apply(lambda x:0 if x<=3 else 1)

|  |  |  |
| --- | --- | --- |
|  | Review | Data |
| Rating > 3 | Positive (1) | 7559 |
| Rating <= 3 | Negative (0) | 2441 |
| Total |  | 10000 |

Table 5.1 One-hot encoding

**3) EDA:**

EDA is a powerful tool for sentiment analysis. EDA helps to gain insights into data and can be used to find any potential problems or issues in the data set.

|  |  |
| --- | --- |
| Mean | 651.8996 |
| Standard Deviation | 593.5594 |
| Min | 8 |
| Max | 14254 |

Table 5.2 EDA Value report

The average review length is 651.899 symbols (likely characters) but sometimes It can only be 593.559. There are some reviews with more than 14000 symbols. It notes that there are some noticeably short reviews, with the shortest sentence having only 8 symbols. This is supported by the standard deviation and the mean points. There is a wide range of review lengths, from noticeably short to exceptionally long.

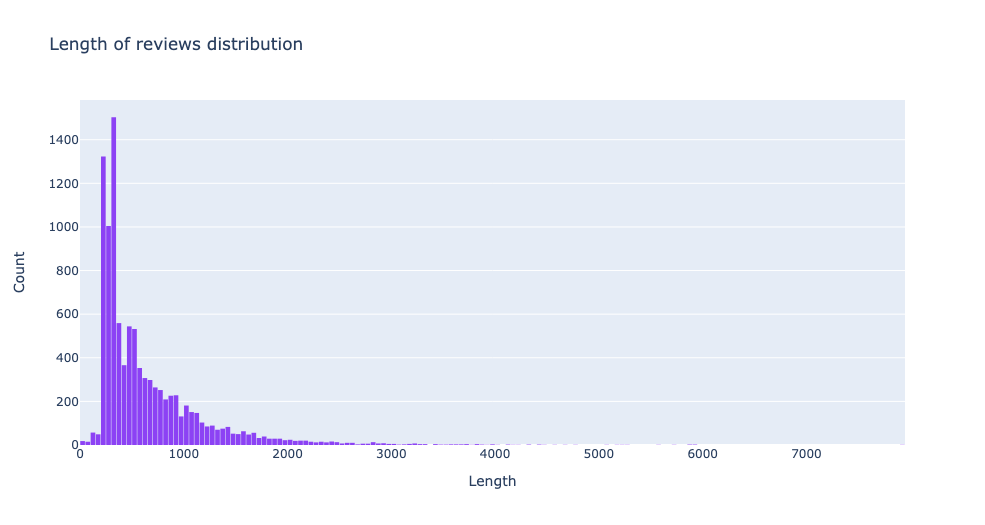


Figure 5.2 Length of reviews distribution

We can see that most of the reviews have a length between 200 and 549 symbols. There are many reviews in the 550-350 symbols range. Most of the reviews have a length of fewer than 6500 symbols. The length of the review is quite scattered.

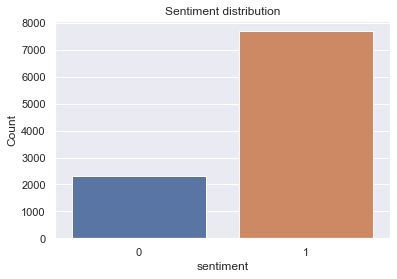


Figure 5.3 Sentiment distribution

We visualized the data to understand the sentiment distribution across our dataset better. In the above graph 0 stands for Negative reviews while 1 stand for positive reviews. We found that most of the data was classified as positive, and there is not a substantial number of negative sentiments, which could be one of the model's limitations. As we mentioned there are 2000 negative reviews, which means 20% of the data, and 8000 positive reviews, which means 80% of the data. This EDA can be used to inform our sentiment analysis model to achieve better results.

**Heatmap of Pearson correlation:**

Sentiment is determined by the words used in a review and not by its length. The sentiment of a review is determined by the words and phrases used in the review and the overall tone of the review.

We can see if there is a correlation between the length of the review and the sentiment label. We can clearly see from the graphs below that there are no significant correlations between these variables.

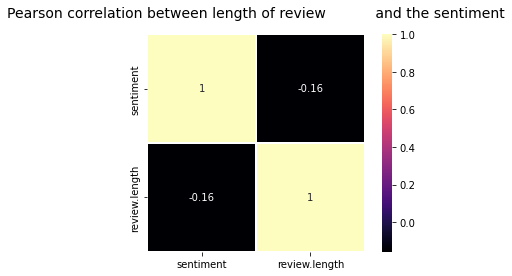


Figure 5.4 Correlation plot

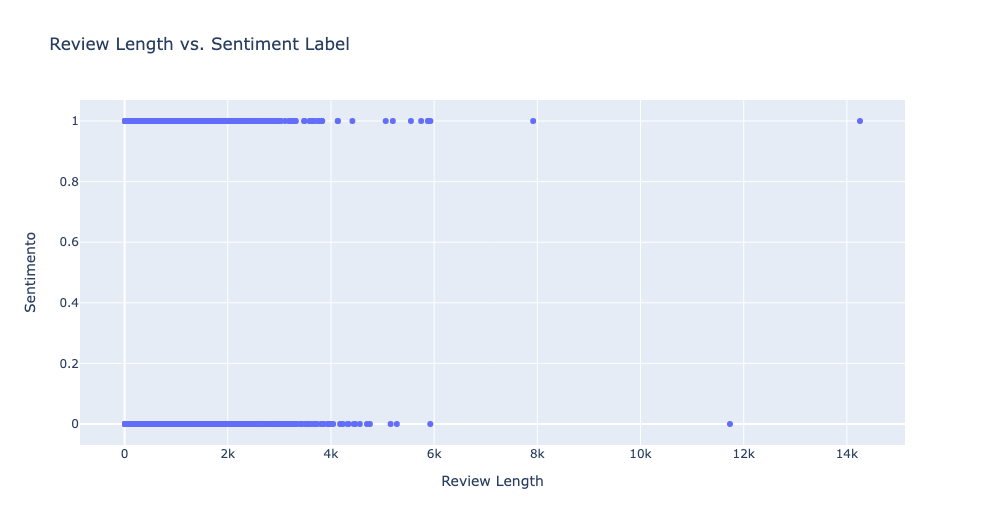


Figure 5.5 Review Length vs. Sentiment Label

Hence, we can say that through we could prove that there is no relation between sentiments and review length.

## Data Engineering:

This involves Data Preprocessing and cleaning. Data preprocessing is the process of cleaning, organizing, and formatting data in order to make it more useful and easier to analyze. There are several techniques that can be used in data preprocessing, including stop words, pos\_tag, and nltk corpus.

Remove stop words:

Stop words are words that are commonly used in language, but do not contribute much meaning to a sentence. Examples of stop words include "the", "a", "an", and "of". Removing stop words from a dataset can help to reduce the size of the data and make it easier to analyze.

stop = stopwords.words('english')

stop.remove('not')

stop.append('hotel')

Apart from these we are also using some functions to convert text into lower case, removing words containing only one word and number, etc.

Perform part-of-speech tagging:

Pos\_tag is a function in the Natural Language Toolkit (nltk) library that assigns part-of-speech tags to words in a sentence. This can help to identify the function of each word in a sentence and make it easier to analyze the meaning and structure of the sentence.

import string

from nltk.corpus import wordnet

from nltk import pos\_tag

Nltk corpus is a collection of text data that has been annotated and organized for use in natural language processing tasks. The nltk corpus can be used to train machine learning algorithms and to perform tasks such as part-of-speech tagging and sentiment analysis.

In summary, data preprocessing using stop words, pos\_tag, and nltk corpus involves cleaning and organizing text data in order to make it more useful and easier to analyze. These techniques can help to improve the accuracy and efficiency of natural language processing tasks.

The following table shows the result after the preprocessing and cleaning of the data, here clean.review is the result whereas reviews.text is the original review.



Figure 5.6 the results of preprocessing and cleaning the review text

## Data Visualization:

Word cloud

A word cloud of positive words is a visual representation of the frequency and prevalence of positive words in each text or dataset. The size and color of each word in the cloud reflects its frequency or importance within the text. For example, a word like "quiet" or "clean" may appear larger and in a brighter color if it appears more frequently in the text compared to a word like "fun" or "near."

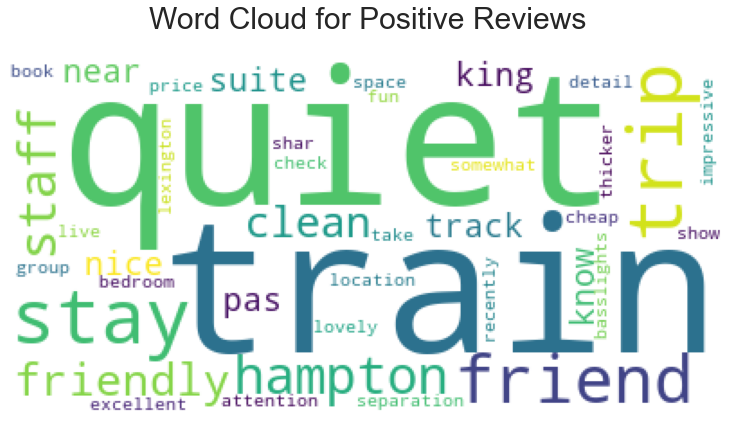


Figure 5.7 Word cloud for Positive Reviews

On the other hand, a word cloud of negative words is a visual representation of the frequency and prevalence of negative words in each text or dataset. The size and color of each word in the cloud reflects its frequency or importance within the text. For example, a word like "horrible" or "place" may appear larger and in a darker color if it appears more frequently in the text compared to a word like "cleanliness" or "bad."

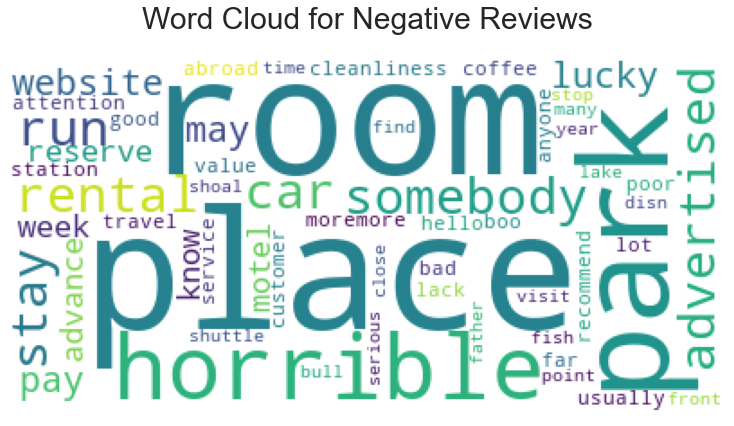


Figure 5.8 Word cloud for Negative Reviews

Overall, word clouds of positive and negative words can help to quickly and visually highlight the overall sentiment or emotion of a text or dataset.

## Implementation of Vader Model

Pos: This score indicates that the text contains positive sentiments, such as happiness or appreciation.

Neg: This score indicates that the text contains negative sentiment, such as anger or sadness.

Neu: This score does not indicate any sentiment

Compound: This score is a combination of both the positive and negative scores.

SentimentIntensityAnalyzer()

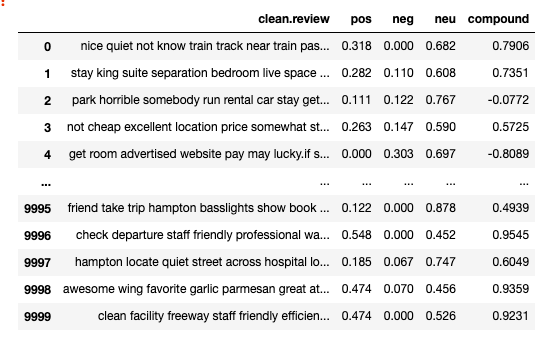


Figure 5.9 Vader Results

if compound > 0.05:

return 1

else:

return 0

Here 0 represents Negative sentiment of the review whereas 1 represents Positive sentiment of the review.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Vader Model** | | | | |
|  | | | | |
|  | Precision | Recall | F1-score | Support |
| 0 (Negative) | 0.79 | 0.23 | 0.36 | 2311 |
| 1 (Positive) | 0.81 | 0.98 | 0.89 | 7689 |
|  |  |  |  | 10,000 |
| Accuracy | 0.81 | | |  |

Table 5.3 Vader Evaluation

## Train and Test

To split the data for training and testing, we can use the train\_test\_split function from the sklearn library. Here, we are using a test size of 0.2, which means that 20% of the data will be used for testing and the remaining 80% will be used for training.

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size = 0.20)

|  |  |  |  |
| --- | --- | --- | --- |
|  | Train | Test | Total |
|  |  |  |  |
| Positive(Rating > 3) | 6047 | 1512 | 7559 |
| Negative(Rating <= 3) | 1953 | 488 | 2441 |
| Total | 8000 | 2000 | 10000 |

Table 5.4 Distribution of train and test data

There are a few reasons why data is often split into a training set and a testing set in a ratio of 20-80:

1. To ensure the model is generalizable: By training the model on a larger dataset (80%) and testing it on a smaller dataset (20%), we can ensure that the model is not overfitted to the training data and can generalize to unseen data.
2. To evaluate model performance: By testing the model on a separate dataset, we can accurately evaluate its performance on unseen data. This helps us determine if the model is overfitting or underfitting and make adjustments accordingly.

## Implementation of TF-IDF model:

In the context of the term frequency-inverse document frequency (TF-IDF) method, **min\_df** is a parameter that can be used to specify the minimum number of documents in which a term must appear to be included in the vocabulary of the model.

For example, if **min\_df** is set to 10, then a term that appears only in nine out of the total number of documents will not be included in the vocabulary. This can be useful for eliminating rare terms that might not be very meaningful for the analysis.

We are experimenting with different min\_df values for this model and evaluating the accuracy of all models. Following this experiment, we determined that the best value for min\_df is 40. But we can set the value according to the model we want to use for the analysis.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Model | Accuracy | | | | | |
|  |  | | | | | |
|  | min\_df= 10 | min\_df=20 | min\_df=30 | min\_df=40 | min\_df=50 | min\_df=60 |
| Random Forest | 0.87  n=50 | 0.87  n=30 | 0.87  n=40 | 0.87  n=20 | 0.87  n=40 | 0.87  n=40 |
| NaiveBayes | 0.87 | 0.89 | 0.88 | 0.88 | 0.87 | 0.87 |
| SVM | 0.88 | 0.89 | 0.89 | 0.89 | 0.89 | 0.89 |
|  |  |  |  |  |  |  |

Table 5.5 Fine Tune of TFIDF model

On the other hand, setting **min\_df** to an extremely low value can result in a large vocabulary and may make the model more prone to over fitting. It is an innovative idea to experiment with different values of **min\_df** to find the one that works best for a given dataset and task.

vectorizer = TfidfVectorizer(min\_df=40)

Here we are vectorizing test and train data for further classification process.

train\_tfidf = vectorizer.fit\_transform(X\_train)

test\_tfidf = vectorizer.transform(X\_test)

## Implementation & Evaluation of LSTM model:

The model is a type of neural network called a Long Short-Term Memory (LSTM) network, which is often used for natural language processing tasks. It is a sequential model, meaning that the layers are added in a specific order.

model = Sequential()

This creates a new Sequential model object in Keras. A Sequential model is a linear stack of layers, with the output of one layer serving as the input to the next.

The model has three layers: an embedding layer, a long short-term memory (LSTM) layer, and a dense layer.

1. **The first layer is an embedding layer.**

"embed\_dim = 32"

This sets the size of the embedding layer in the model to 32. The embedding layer is used to map words in a text dataset to a lower-dimensional space, allowing the model to process the text more efficiently.

"model.add(Embedding(max\_features, embed\_dim, input\_length=X.shape[1]))"

This adds an embedding layer to the model.

1. **The second layer is an LSTM.**

lstm\_out = 16

This sets the size of the output from the LSTM layer to 16. This is a type of recurrent neural network layer used to process sequential data, such as text.

model.add(LSTM(lstm\_out))

adds an LSTM layer to the model. The "lstm\_out" parameter specifies the size of the output from the LSTM layer.

1. **The third layer is a dense layer.**

model.add(Dense(1, activation='sigmoid'))

This adds a dense (fully connected) layer to the model with a single output node and a sigmoid activation function. The sigmoid function maps the output of the dense layer to a value between 0 and 1, which is useful for classification tasks.

model.compile(loss='binary\_crossentropy', optimizer='adam', metrics=['accuracy'])

This compiles the model with a binary cross-entropy loss function, the Adam optimization algorithm, and accuracy as a metric. The loss function measures how well the model is able to predict the correct output, and the optimizer updates the model's weights to minimize the loss. The metric is used to evaluate the model's performance during training.

**Fine Tuning of model:**

1. **epochs=6, lstm\_out= 64, features= 2000**

**Accuracy & Loss summary**

loss: 0.2213 - accuracy: 0.9180 - val\_loss: 0.3004 - val\_accuracy: 0.8725

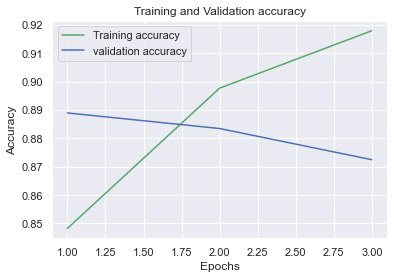
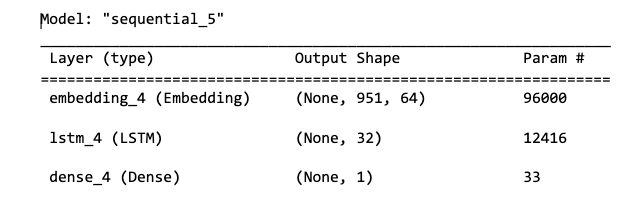


Figure 5.10 LSTM Fine Tune Model 1

1. **Epoch=6, features= 1500**



**Accuracy & Loss summary**

loss: 0.2091 - accuracy: 0.9220 - val\_loss: 0.3060 - val\_accuracy: 0.8770

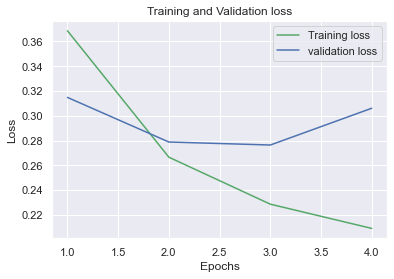
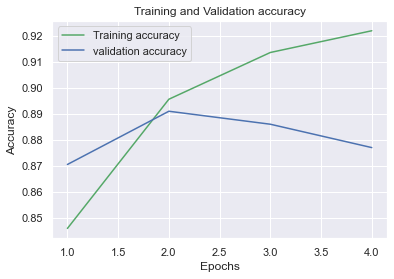
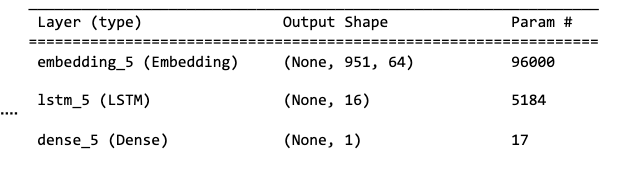


Figure 5.11 LSTM Fine Tune Model 2

1. **Epoch=6, features= 1500**



**Accuracy & Loss summary**

loss: 0.2283 - accuracy: 0.9134 - val\_loss: 0.2847 - val\_accuracy: 0.8880

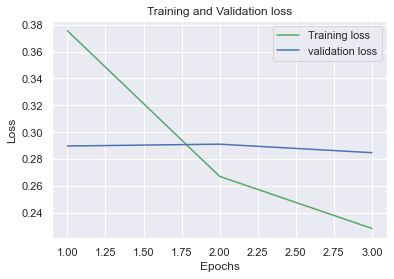
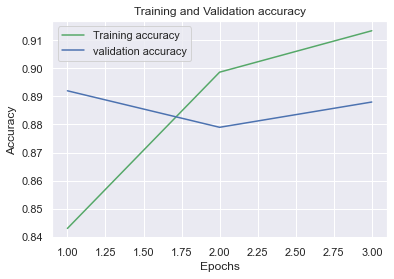
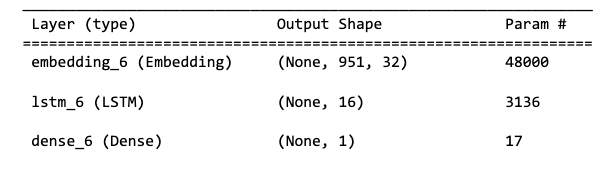


Figure 5.12 LSTM Fine Tune Model 3

1. **Epoch=6, features= 1500**



**Accuracy & Loss summary**

loss: 0.2322 - accuracy: 0.9116 - val\_loss: 0.2934 - val\_accuracy: 0.8810

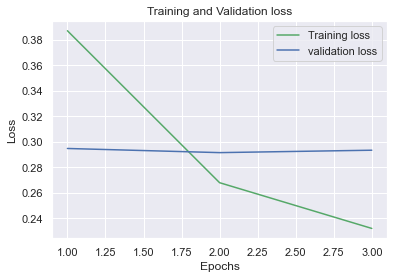
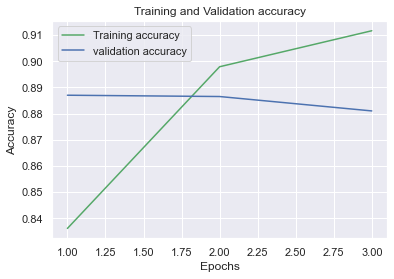
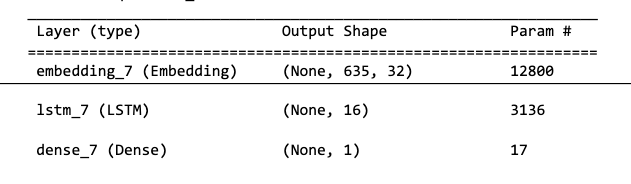


Figure 5.13 LSTM Fine Tune Model 4

1. **Epoch=6, features= 400, lstm\_out=16, embedding= 32**



**Accuracy & Loss summary**  
loss: 0.2937 - accuracy: 0.8824 - val\_loss: 0.3217 - val\_accuracy: 0.8660

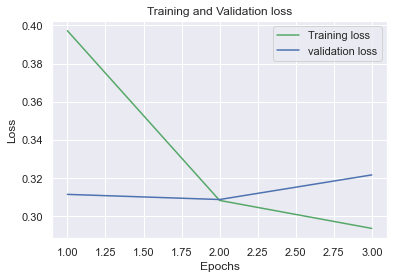
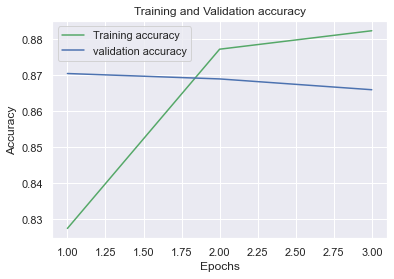
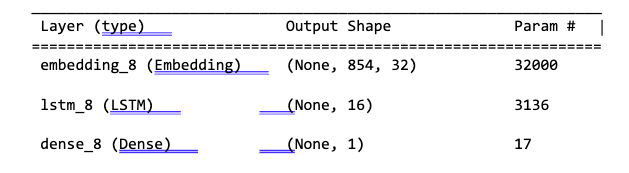


Figure 5.14 LSTM Fine Tune Model 5

1. **Epoch=6, features= 1000, lstm\_out=16, embedding= 32**



**Accuracy & Loss summary**

loss: 0.2184 - accuracy: 0.9151 - val\_loss: 0.3044 - val\_accuracy: 0.8900

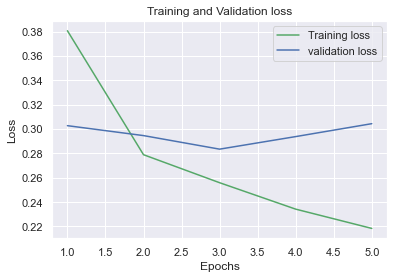
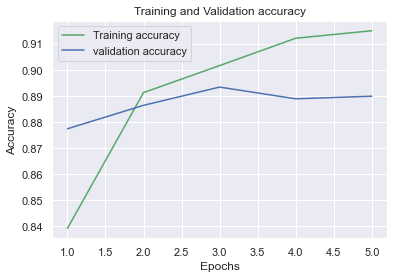
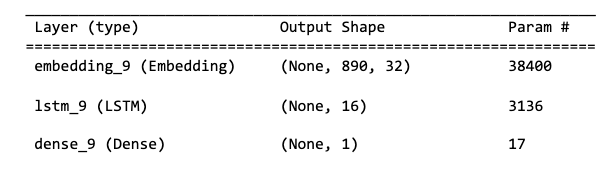


Figure 5.15 LSTM Fine Tune Model 6

1. **Epoch=6, features= 1200, lstm\_out=16, embedding= 32**



**Accuracy & Loss summary**

loss: 0.2212 - accuracy: 0.9168 - val\_loss: 0.3031 - val\_accuracy: 0.8865

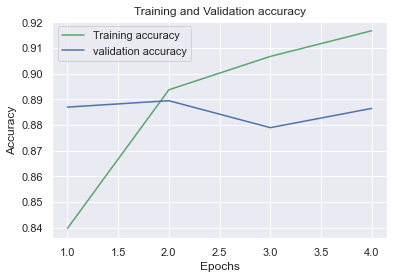
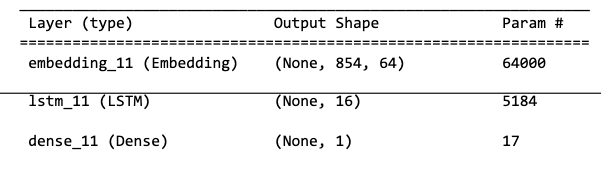


Figure 5.16 LSTM Fine Tune Model 7

1. **Epoch=6, features= 1000, lstm\_out=16, embedding= 64**



**Accuracy & Loss summary**

loss: 0.2200 - accuracy: 0.9150 - val\_loss: 0.3049 - val\_accuracy: 0.8830

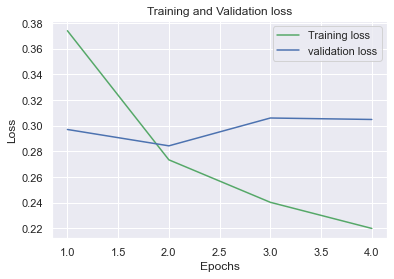
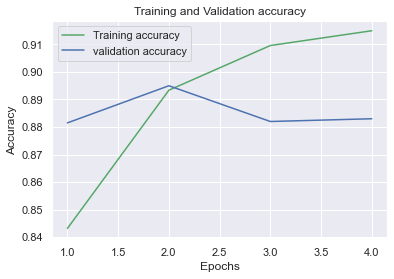


Figure 5.17 LSTM Fine Tune Model 6

1. **Final Case:**

**Here, after all those attempts of fine-tuning the network, we will use the following specifications to train the model for further evaluation and prediction of data as the accuracy is at its peak, which is 89% or 0.89 whereas loss is not increasing with the increment in no of epochs. It is noticeable that even other results have remarkable accuracy. The reason not to choose those is that with the epoch the loss is increasing, and accuracy is decreasing.**

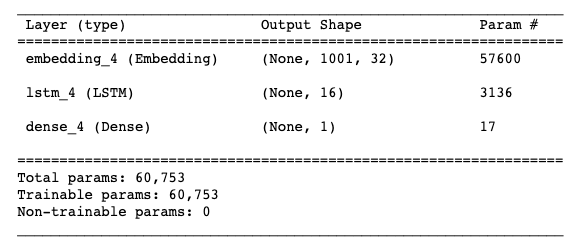


Figure 5.18 Parameter Summary of LSTM

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| LSTM Evaluation | | | | |
|  | | | | |
|  | Training Data (80%) | | |  |
|  | Precision | Recall | F1-score | Support |
| 0 (Negative | 0.93 | 0.89 | 0.91 | 1845 |
| 1 (Positive) | 0.97 | 0.98 | 0.97 | 6155 |
|  |  |  |  | 8000 |
| Accuracy | 0.96 | | |  |
|  |  | | |  |
|  | Testing data (20%) | | |  |
|  | Precision | Recall | F1-score | Support |
| 0  (Negative) | 0.80 | 0.68 | 0.74 | 466 |
| 1  (Positive) | 0.91 | 0.95 | 0.93 | 1534 |
|  |  |  |  | 2000 |
| Accuracy | 0.89 | | |  |
|  |  | | |  |

Table 5.6 LSTM Evaluation

In the graph above, we can see that training accuracy is evidently increasing with the number of epochs whereas in validation accuracy there is no significant improvement, but it is not decreasing either. In the loss graph, we see the loss in training data is decreasing, but overall validation loss is staying stable.

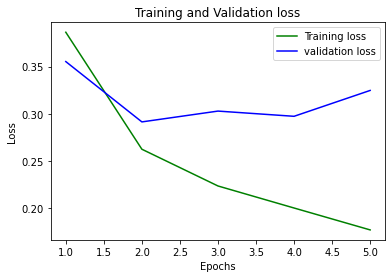
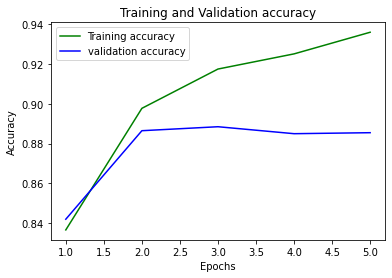


Figure 5.19 LSTM Accuracy & Loss graph

|  |  |  |
| --- | --- | --- |
| LSTM Confusion Matrix | | |
|  | Training model | Testing model |
| TP (True Positive) | 1635 | 318 |
| FP (False Positive) | 210 | 148 |
| FN (False Negative) | 127 | 81 |
| TN (True Negative) | 6028 | 1453 |

Table 5.7 LSTM Confusion matrix

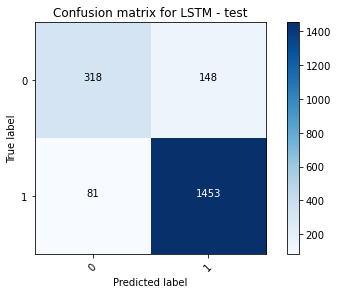
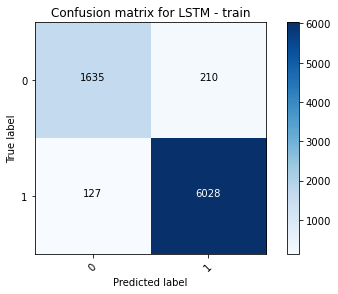


Figure 5.20 LSTM Confusion Matrix

## Implementation & Evaluation of Random Forest

rf = RandomForestClassifier

In a Random Forest, n\_estimators refers to the number of trees in the forest.

The more trees there are in the forest, the better the model can capture the pattern in the data, and the more accurate the predictions will be. However, adding more trees to the forest also increases the computational cost and can slow down the training process. As a result, finding the right balance between model performance and computational cost is an important consideration when training a Random Forest.

rf = RandomForestClassifier(n\_estimators = 30, random\_state = 42)

Training a model:

rf.fit(train\_tfidf, y\_train)

Prediction of test and train data:

pred\_train\_rf=rf.predict(train\_tfidf)

pred\_test\_rf= rf.predict(test\_tfidf)

The following table shows the estimators and respective test accuracy.

|  |  |
| --- | --- |
| n\_estimators | test\_accuracy |
|  |  |
| 10 | 0.86 |
| 20 | 0.8655 |
| 30 | 0.87 |
| 40 | 0.8695 |
| 50 | 0.872 |
| 60 | 0.8685 |
| 70 | 0.872 |
| 80 | 0.8685 |
| 90 | 0.867 |

Table 5.8 Fine Tuning RF Model

As we can see from the table after n\_estimators= 30, the accuracy is decreasing. We can set the n value as 20. As there is no significant improvement has been made after that. Also, a large value of n can cause the overfitting and slow process as mentioned earlier.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Random Forest Evaluation | | | | |
|  | | | | |
|  | Training Data (80%) | | |  |
|  | Precision | Recall | F1-score | Support |
| 0 (Negative | 0.99 | 0.99 | 0.99 | 1827 |
| 1 (Positive) | 0.99 | 0.99 | 0.99 | 6173 |
|  |  |  |  | 8,000 |
| Accuracy | 0.99 | | |  |
|  |  | | |  |
|  | Testing data (20%) | | |  |
|  | Precision | Recall | F1-score | Support |
| 0  (Negative) | 0.89 | 0.53 | 0.66 | 484 |
| 1  (Positive) | 0.87 | 0.98 | 0.92 | 1516 |
|  |  |  |  | 2,000 |
| Accuracy | 0.87 | | |  |

Table 5.9 RF Evaluation

|  |  |  |
| --- | --- | --- |
| Random Forest Confusion Matrix | | |
|  | Training model | Testing model |
| TP (True Positive) | 1822 | 257 |
| FP (False Positive) | 5 | 227 |
| FN (False Negative) | 5 | 33 |
| TN (True Negative) | 6168 | 1483 |

Table 5.10 RF Confusion Matrix

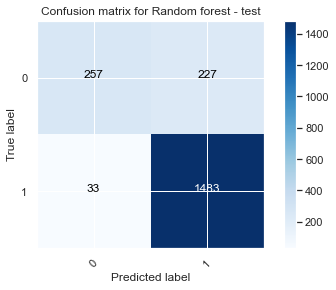
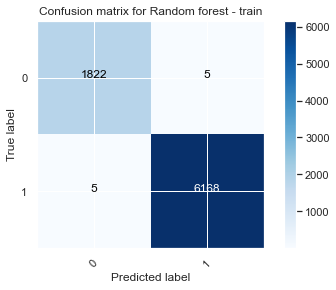


Figure 5.21 Confusion Matrix RF model

## Implementation and Evaluation of SVM:

svm\_model = svm.SVC (kernel='linear', random\_state = 42)

As we mentioned earlier Linear SVM is a viable choice for sentiment analysis due to its efficiency, robustness, interpretability, and high accuracy rate. The following table shows the evaluation of SVM.

Training model:

svm\_model = svm.SVC(kernel='linear', random\_state = 42)

svm\_model.fit(train\_tfidf, y\_train)

Prediction of test data

pred\_train\_svm = svm\_model.predict(train\_tfidf)

Prediction of training data to find out the accuracy of the training model.

pred\_test\_svm = svm\_model.predict(test\_tfidf)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Evaluation Support Vector Machine** | | | | |
|  | | | | |
|  | Training Data (80%) | | |  |
|  | Precision | Recall | F1-score | Support |
| 0 (Negative | 0.91 | 0.74 | 0.82 | 1827 |
| 1 (Positive) | 0.93 | 0.98 | 0.95 | 6173 |
|  |  |  |  | 2000 |
| Accuracy | 0.93 | | |  |
|  |  | | |  |
|  | Testing data (20%) | | |  |
|  | Precision | Recall | F1-score | Support |
| 0  (Negative) | 0.83 | 0.68 | 0.75 | 484 |
| 1  (Positive) | 0.90 | 0.96 | 0.93 | 1516 |
|  |  |  |  | 2000 |
| Accuracy | 0.89 | | |  |
|  |  | | |  |

Table 5.11 SVM Evaluation

|  |  |  |
| --- | --- | --- |
| SVM Confusion Matrix | | |
|  | Training model | Testing model |
| TP (True Positive) | 1359 | 331 |
| FP (False Positive) | 468 | 153 |
| FN (False Negative) | 142 | 68 |
| TN (True Negative) | 6081 | 1448 |

Table 5.12 SVM Confusion Matrix

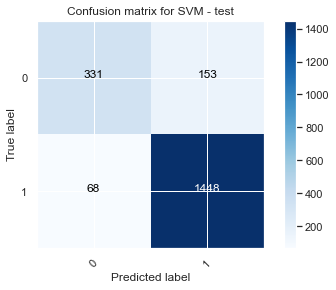
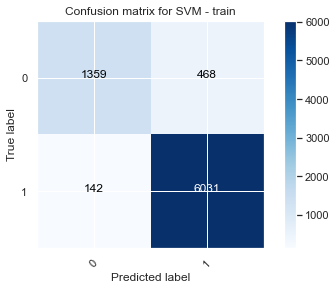


Figure 5.22 Confusion Matrix for SVM

## Implementation & Evaluation Multinomial NaiveBayes:

Same as other classifier we are using the result of tfidf model for training.

model\_nb = MultinomialNB()

model\_nb.fit(train\_tfidf, y\_train)

Prediction of test data

pred\_test\_nb = model\_nb.predict(test\_tfidf)

Prediction of training data to find out the accuracy of the training model.

pred\_train\_nb = model\_nb.predict(train\_tfidf)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Multinomial NaiveBayes | | | | |
|  | | | | |
|  | Training Data (80%) | | |  |
|  | Precision | Recall | F1-score | Support |
| 0 (Negative | 0.91 | 0.57 | 0.70 | 1827 |
| 1 (Positive) | 0.88 | 0.98 | 0.93 | 6173 |
|  |  |  |  | 8000 |
| Accuracy | 0.89 | | |  |
|  |  | | |  |
|  | Testing data (20%) | | |  |
|  | Precision | Recall | F1-score | Support |
| 0  (Negative) | 0.89 | 0.57 | 0.70 | 484 |
| 1  (Positive) | 0.88 | 0.98 | 0.92 | 1516 |
|  |  |  |  | 2000 |
| Accuracy | 0.88 | | |  |
|  |  | | |  |

Table 5.13 MNB Evaluation

Table:

|  |  |  |
| --- | --- | --- |
| MNB Confusion Matrix | | |
|  | Training model | Testing model |
| TP (True Positive) | 1017 | 275 |
| FP (False Positive) | 810 | 209 |
| FN (False Negative) | 126 | 28 |
| TN (True Negative) | 6047 | 1488 |

Table 5.14 MNB Confusion Matrix

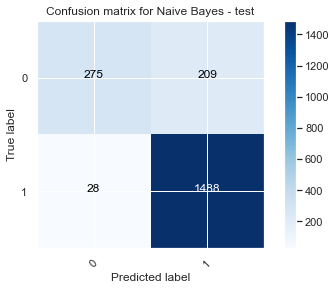
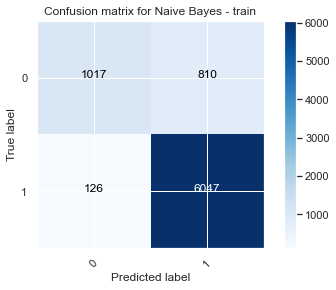


Figure 5.23 Confusion matrix for NaiveBayes

## Comparison of Evaluation of all classifiers

**Evaluation Table**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Training Model |  | | Accuracy | F1-score |
| LSTM | Negative (0) | | 96% | 91% |
| Positive (1) | | 97% |
|  | | | | |
| Random Forest | Negative (0) | | 99% | 99% |
| Positive (1) | | 99% |
|  | | | | |
| SVM | Negative (0) | | 93% | 82% |
| Positive (1) | | 95% |
|  | | | | |
| NaiveBayes | Negative (0) | | 89% | 70% |
| Positive (1) | | 92% |
|  | | | | |
| Testing Model |  | Accuracy | | F1-score |
| LSTM | Negative | 89% | | 74% |
| Positive (1) | 93% |
|  | | | | |
| Random Forest | Negative | 87% | | 66% |
| Positive (1) | 92% |
|  | | | | |
| SVM | Negative | 89% | | 75% |
| Positive (1) | 93% |
|  | | | | |
| NaiveBayes | Negative | 88% | | 70% |
| Positive (1) | 92% |

Table 5.15 Comparison of all Evaluation

The accuracy and F1 score of SVM and LSTM being almost the same suggests that both algorithms are performing equally well in terms of correctly predicting the sentiment of the reviews.

The low F1 score in all the models for negative review is because of the data imbalance. Here we have very few negative reviews compared to positive reviews. Accuracy measures how often the model correctly predicts the sentiment of the review, while F1- score is a combination of precision and recall, which measures how well the model is able to identify positive and negative reviews.

In this case, it appears that both SVM and LSTM are able to identify positive and negative reviews with a similar level of accuracy and precision, resulting in similar scores. This may be due to the nature of the data being analyzed, the effectiveness of the algorithms in analyzing text data, or the specific parameters and hyperparameters used in the models.

If the training accuracy of a random forest model is much higher than the testing accuracy, it means that the model is overfitting to the training data. This means that the model has learned to predict the outcomes in the training dataset very accurately but is not able to generalize these predictions to new, unseen data. This can lead to poor performance on the testing dataset, as the model is not able to accurately predict the outcomes for new data.

The Naive Bayes model gives similar accuracy and F1 scores for both the training and testing datasets. It seems that the model is underfitting, which means that it is not complex enough to capture the nuances and patterns in the data. This could result in the model giving a similar performance on both the training and testing datasets.

**In conclusion, both SVM and LSTM performed well in predicting the sentiment of reviews with similar accuracy and F1 scores. The random forest model may have overfitted to the training data, leading to a lower performance on the testing data. The naive Bayes model may have underfitted, resulting in similar performance on both training and testing datasets.**

# **Application & Visualization:**

## Application Model

The model will be tested on new data to see if it can accurately predict the outcomes. If the accuracy is still high on the new data, it will be considered a reliable model for making predictions on future data.

The application will be designed to allow users to input their own data and receive predictions on the outcomes. This could be useful for businesses or individuals who want to make informed decisions based on data analysis. The application will include interactive visualizations to help users understand and interpret the predictions.

Overall, the use of the LSTM model will provide valuable insights and improve decision-making processes within the company.

## Visualizations

Predictions of the original data will be made first to create application visualizations using the LSTM trained model. The accuracy of that is 94%.

**Visualization - 1**

Here we have sorted out top hotels with the highest rating. There are several reasons why a list of top hotels with positive reviews is important:

* **Quality**: Positive reviews often indicate that a hotel provides high-quality services and amenities, which is important for travelers who want to have a comfortable and enjoyable stay.
* **Trust**: Travelers often rely on reviews when deciding where to stay, so a list of top hotels with positive reviews can help build trust and credibility with potential guests.
* **Ranking**: A hotel's ranking on a list of top hotels with positive reviews can affect its visibility and booking rates. Higher rankings can lead to more bookings, while lower rankings may lead to fewer bookings.
* **Marketing**: A list of top hotels with positive reviews can be used as a marketing tool to attract new guests and display the hotel's reputation.
* **Competition**: In a highly competitive industry like the hotel industry, a list of top hotels with positive reviews can give hotels an edge over their competitors and help them stand out in the market.

|  |  |  |  |
| --- | --- | --- | --- |
| **Sr. No** | **name** | **pred** | **sentiment** |
| **1** | Hampton Inn Suites Red Bluff | 0.99644 | 1 |
| **2** | Sleepy Valley Inn | 0.994945 | 1 |
| **3** | Ramada | 0.994893 | 1 |
| **4** | The Harkness Hotel | 0.994795 | 1 |
| **5** | Hampton Inn Colorado Springs-airport | 0.994781 | 1 |
| **6** | The Mercantile Hotel | 0.994704 | 1 |
| **7** | Hampton Inn-rawlins | 0.994672 | 1 |
| **8** | Aloft Houston by the Galleria | 0.994552 | 1 |
| **9** | The Herbert Hotel | 0.994494 | 1 |
| **10** | Hampton Inn Suites Phenix City-Columbus Area | 0.994196 | 1 |

Table 6.1 overall Top 10 hotels with highest rating

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Sr. No.** | **year** | **name** | **pred** | **sentiment** |
| **1** | 2019 | Best Western Plus Pavilions | 0.985105 | 1 |
| **2** | 2018 | Aloft Atlanta Downtown | 0.959073 | 1 |
| **3** | 2017 | 402 Hotel #Thebigo | 0.989785 | 1 |
| **4** | 2016 | 11th Avenue Hotel Hostel | 0.089467 | 0 |
| **5** | 2015 | Adobe Hacienda Motel | 0.935217 | 1 |
| **6** | 2014 | Aloft Atlanta Downtown | 0.631893 | 1 |
| **7** | 2013 | Aloft Philadelphia Airport | 0.993521 | 1 |
| **8** | 2012 | Aloft Dallas Downtown | 0.952453 | 1 |
| **9** | 2011 | Anaheim Del Sol Inn | 0.383848 | 0 |
| **10** | 2010 | Anaheim Del Sol Inn | 0.362846 | 0 |
| **11** | 2009 | Best Western Lamplighter Inn Suites at SDSU | 0.878621 | 1 |
| **12** | 2008 | Best Western Cabrillo Garden Inn | 0.52168 | 1 |
| **13** | 2007 | Anaheim Del Sol Inn | 0.718022 | 1 |
| **14** | 2006 | Anaheim Del Sol Inn | 0.105111 | 0 |
| **15** | 2005 | Best Western Seven Seas | 0.274428 | 0 |
| **16** | 2004 | Anaheim Del Sol Inn | 0.983464 | 1 |
| **17** | 2003 | River Hotel | 0.961707 | 1 |

Table 6.2 Top hotels with highest rating each year

Overall, a list of top hotels with positive reviews is important for providing travelers with a reliable source of information and helping hotels attract new guests and build their reputation.

**Visualization - 2**

The need for a list of top hotels with negative reviews is important for several reasons:

1. **Quality control:** By identifying the top hotels with negative reviews, hotel management can address any issues or complaints that may be affecting the overall satisfaction of guests. This can help improve the quality of service and amenities offered at the hotel.
2. **Customer satisfaction:** By addressing negative reviews and addressing the concerns of unhappy customers, hotels can improve their overall reputation and customer satisfaction levels. This can lead to increased business and positive word-of-mouth referrals.
3. **Brand reputation:** Negative reviews can have a significant impact on a hotel's reputation and brand image. By proactively addressing negative reviews, hotels can protect and improve their reputation, and maintain the trust and loyalty of their customers.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Sr. No** | **Year** | **Name** | **pred** | **sentiment** |
| **1** | 2003 | River Hotel | 0.961707 | 1 |
| **2** | 2004 | Anaheim Del Sol Inn | 0.983464 | 1 |
| **3** | 2005 | Best Western Seven Seas | 0.274428 | 0 |
| **4** | 2006 | Anaheim Del Sol Inn | 0.105111 | 0 |
| **5** | 2007 | Anaheim Del Sol Inn | 0.718022 | 1 |
| **6** | 2008 | Best Western Cabrillo Garden Inn | 0.52168 | 1 |
| **7** | 2009 | Best Western Lamplighter Inn Suites at SDSU | 0.878621 | 1 |
| **8** | 2010 | Anaheim Del Sol Inn | 0.362846 | 0 |
| **9** | 2011 | Anaheim Del Sol Inn | 0.383848 | 0 |
| **10** | 2012 | Aloft Dallas Downtown | 0.952453 | 1 |
| **11** | 2013 | Aloft Philadelphia Airport | 0.993521 | 1 |
| **12** | 2014 | Aloft Atlanta Downtown | 0.631893 | 1 |
| **13** | 2015 | Adobe Hacienda Motel | 0.935217 | 1 |
| **14** | 2016 | 11th Avenue Hotel Hostel | 0.089467 | 0 |
| **15** | 2017 | 402 Hotel #Thebigo | 0.989785 | 1 |
| **16** | 2018 | Aloft Atlanta Downtown | 0.959073 | 1 |

Table 6.3 Top hotels with lowest rating each year

|  |  |  |  |
| --- | --- | --- | --- |
| **Sr. No** | **name** | **pred** | **sentiment** |
| **1** | Best Western Logan Inn | 0.027882 | 0 |
| **2** | Motel 6-dubuque | 0.028598 | 0 |
| **3** | Budgethost Inn | 0.031592 | 0 |
| **4** | Knights Inn-murfreesboro | 0.033437 | 0 |
| **5** | Ocean Park Hotel | 0.034632 | 0 |
| **6** | Holiday Inn Express TulsaWoodland Hills | 0.034822 | 0 |
| **7** | Motel Durango | 0.034938 | 0 |
| **8** | C H Motel | 0.036051 | 0 |
| **9** | Quality Inn Suites Oceanside Near Camp Pendleton | 0.03667 | 0 |
| **10** | Red Lion Inn Suites | 0.036761 | 0 |

Table 6.4 Overall Top 10 hotels with lowest rating

**Visualization - 3**

There are several reasons why it is important to know the review of a hotel over the years:

* Quality: By looking at the reviews over the years, you can get an idea of the overall quality of the hotel. If the reviews are consistently positive, it is likely that the hotel is well- maintained and provides a pleasant experience for guests. On the other hand, if the reviews are consistently negative, it may be a sign that the hotel is not up to par and may not be worth booking.
* Service: Reviews can also provide insight into the level of service provided by the hotel. If the reviews mention good customer service, it may be a good sign that the hotel values its guests and goes out of its way to ensure their satisfaction.
* Trends: By looking at the reviews over the years, you can also see some trends in terms of what guests like and dislike about the hotel. For example, if there are consistently negative reviews about room service, it may be a sign that the hotel needs to improve in this area.
* Reputation: A hotel's reputation is important, and reviews can play a crucial role in shaping it. If a hotel has consistently poor reviews, it may be seen as a low-quality establishment, which could discourage potential guests from booking a room.
* Comparison: Knowing the review of a hotel over the years can also help you compare it to other hotels in the area. By looking at multiple reviews, you can get a better idea of which hotels are worth considering and which may not be up to your standards.

|  |  |  |
| --- | --- | --- |
| **name** | **pred** | **sentiment** |
| predpred\_ans39Anaheim Del Sol Inn | 0.7018261 | 1 |

Table 6.5 Overall hotel rating



Figure 6.1 Overall hotel rating over the years

**Visualization - 4**

A word cloud of positive reviews of a hotel is a visual representation of the most frequently used words in positive reviews of the hotel. This tool can be useful for a hotel to understand what aspects of their property are most highly praised by their guests.

One application of word cloud on a hotel website could be to showcase the most frequently mentioned amenities or features of the hotel.

For example, a word cloud could be created using guest reviews of the hotel, with the size of each word representing the number of times it was mentioned. Words such as "pool," "gym," "spa," and "breakfast" may appear larger in the word cloud, indicating that these amenities are highly praised by guests.

This could help potential guests quickly understand what the hotel has to offer and make informed decisions about their stay. The word cloud could also be used to highlight unique or unusual features of the hotel, such as a rooftop bar or a pet-friendly policy, to help the hotel stand out from competitors.



Figure 6.2 Word cloud for Top hotel positive review

Overall, the wordcloud of positive reviews can be a valuable tool for a hotel to better understand what their guests appreciate about their property and how to effectively promote it to potential guests.

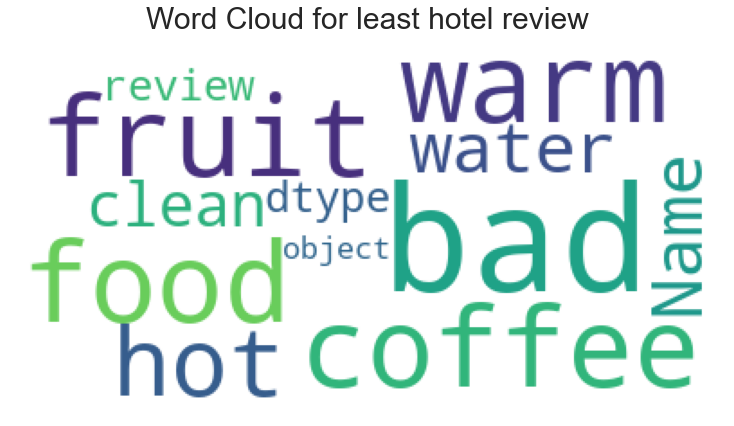


Figure 6.3 Word cloud for Top hotel negative review

A negative review word cloud for a hotel can be useful for identifying common themes or issues that guests have experienced at the hotel. This can help the hotel management team to understand where they may need to improve in order to provide a better experience for their guests. Additionally, a negative review word cloud can help the management team to identify any specific problems that may be causing dissatisfaction among guests, such as poor service, dirty rooms, or inadequate amenities. By analyzing negative reviews and addressing the issues that they raise, a hotel can improve its reputation and attract more business.

# **Limitations and Future scope:**

Four different models have been trained and tested in this paper, including random forest, multinomial Naivebayes, linear SVM, and LSTM. To achieve the best results, each model has been tuned to their best features, with random forest and LSTM being tuned in this paper, while the basic features have been used for the other models. It may be explored in the future to fine tune the SVM and Naivebayes models for improved performance. Therefore, by training and fine-tuning these algorithms on larger and more diverse datasets, it may be possible to improve the accuracy and reliability of sentiment analysis for hotel reviews.

The limited number of negative reviews compared to positive reviews in this study could be a hindrance to obtaining accurate results. In the future, this limitation could be overcome by oversampling the negative reviews. Text augmentation techniques like back translation may also be applied to oversample the minority class to address the data imbalance problem. It would be beneficial to try and obtain more negative reviews for the dataset in order to improve the balance. This may require more time and effort, but it can ultimately be a better solution in the long run.

The techniques and algorithms developed for sentiment analysis of hotel reviews can potentially be applied to other domains, such as restaurant reviews, online product reviews, or social media posts. This could help businesses and organizations gain valuable insights into customer sentiment and preferences in these areas.

AS we know we have all different information about the location of the hotel and who posted the review and their location as well. It may be possible to integrate sentiment analysis with such location and reviewer data, to gain a more complete understanding of customer preferences and experiences. This could help businesses tailor their offerings and marketing efforts to better meet the needs and expectations of their target audience.

Apart from tfidf it may be worthwhile to explore alternative methods such as bert and wordtovec in the future as sometimes

# **Conclusion**

After conducting sentiment analysis on hotel reviews using random forest, SVM, Naivebayes and LSTM, it is clear that all methods can effectively identify the sentiment of a review as either positive or negative. However, the LSTM and SVM method appeared to be the most accurate, with an overall accuracy of 89%. The Random Forest and Naive Bayes methods also performed well, with accuracies of 87% and 88%, respectively. The Vader method had the lowest accuracy at 81%.

Overall, F1 scores and accuracy are low for all the models, the reason is data imbalance. Here we have only approx. 20% negative reviews, which is causing this issue One way to address this issue is to try oversampling the minority class. This can help balance the class distribution and potentially improve the model's performance. It's also a good idea to try gathering more negative reviews to improve the balance of the dataset, if possible. This will likely require more time and effort, but it can be a more effective long-term solution.

Vader is a useful tool for quickly analyzing sentiment in text, but it has limitations that should be taken into consideration when using it. While it is easy to use and requires no training or setup, it may not always accurately interpret sentiment in reviews that lack context or nuance, or in reviews that contain sarcastic language. It is important to consider these limitations when using Vader for sentiment analysis.

LSTM networks are a type of recurrent neural network(RNN) that are particularly well-suited for modeling sequences of data, such as the text in a review. They can capture long-term dependencies in the data by using memory cells and gates to selectively store and retrieve information. This allows them to effectively process the context and meaning of words in a review, which is important for accurately predicting its sentiment.

SVMs are a type of supervised learning algorithm that can be used for classification tasks. They work by finding a hyper plane in a high-dimensional space that maximally separates different classes. In the context of sentiment analysis, this means that the SVM would try to find a boundary between positive and negative reviews that maximally separates them.

Random forests are a type of ensemble learning method that uses multiple decision trees to make predictions. They work by training multiple decision trees on random subsets of the data and then combining their predictions through a process called bagging. While random forests can be effective at predicting outcomes, they can also suffer from over fitting, where the model becomes too complex and begins to fit the noise in the data rather than the underlying patterns.

Naive Bayes is a simple probabilistic classifier that relies on the assumption that the features in a dataset are independent of each other. It works by using Bayes’ theorem to calculate the probability that a given data point belongs to a particular class. While naive Bayes can be a fast and effective classifier, it can also be prone to under fitting, where the model is too simple and cannot capture the complexity of the data.

Overall, the results of SVM and LSTM suggest that they are both effective in predicting the sentiment of reviews. These models can effectively learn the relationships between features in the data and make accurate predictions. On the other hand, the results for the random forest model indicate that it may be over fitting and not generalizing well to new data. The naive Bayes model may be underperforming due to the unrealistic assumption of independence between features. It is important to consider these factors when selecting and adjusting algorithms for a specific task, as different models may perform better or worse depending on the characteristics of the data and the task at hand.

The model will be tested on new data to determine its accuracy and reliability in predicting outcomes. If the model performs well during testing, it will be turned into an application that users can interact with and input their own data to receive predictions on outcomes. The application will also include visualizations to help users understand and interpret the predictions. The goal of using the LSTM model is to provide valuable insights and improve decision-making within the company.

Overall, it can be concluded that using a combination of these methods can provide a strong and reliable approach for sentiment analysis of hotel reviews. While each method has its own strengths and limitations, combining them can provide a more comprehensive understanding of the sentiment expressed in a review. This can be useful for hotel managers to track customer satisfaction and make necessary improvements to their service.

|  |  |  |
| --- | --- | --- |
|  | **Advantages** | **Drawbacks** |
| **LSTM** | Particularly well-suited for handling sequential data, such as text. This makes it an effective choice for sentiment analysis of hotel reviews. | LSTM requires a large amount of data to be effective, which may be a challenge when working with smaller datasets. |
|  | LSTM can remember information from long periods of time, which is useful in sentiment analysis as it allows the model to understand the context of the review and make more accurate predictions. | LSTM can be computationally expensive to train, which may require more time and resources. |
|  | LSTM can handle variable-length sequences, which is useful in sentiment analysis as reviews can vary greatly in length. | LSTM can be difficult to interpret and understand, which may make it more challenging to identify and correct errors in the model. |
|  |  |  |
| **Random Forest** | Random forests can handle large amounts of data and can handle noise and missing values in the data. This makes them suitable for analyzing complex and varied hotel reviews. | While random forests provide some interpretability, they do not provide the same level of interpretability as other models. |
|  | Random forests have a high level of training accuracy, data loss in training is very less | Random forests can be prone to over fitting if the number of trees is too high, leading to poor generalization to new data. |
|  |  |  |
| **SVM** | SVM is a fast and efficient algorithm, which means it can handle large datasets and classify them quickly. | It may require tuning of hyper parameters to achieve optimal performance, which can be time-consuming and complex. |
|  | It is robust to noise and can handle high-dimensional data well. And gives good testing accuracy which is a sign of good model | Linear SVM may not perform well when the data is not linearly separable. In this case, it may not be able to accurately classify the data. |
|  |  |  |
| **Naïve Bayes** | Naive Bayes is a simple and efficient algorithm that can handle large amounts of data with fast training and prediction times. | Naive Bayes assumes that all features are independent, which may not always be the case in sentiment analysis. This can lead to a bias in the model and lower overall accuracy. |
|  | It performs well even with few samples for training when compared to other models like LSTM | It may struggle with rare words or phrases that have a strong impact on sentiment, as it relies on the frequency of words rather than the context in which they are used. |
|  |  |  |
| **Vader** | Vader is easy to use and requires no training or setup, making it accessible to users with little or no experience in sentiment analysis. | Vader relies solely on the words in a review to determine sentiment, which can lead to errors in interpretation if the review lacks context or nuance. |
|  |  |  |

Table 8.1 Advantage and Drawbacks of Classifiers

**References:**

[1] Ahmad, G.I., Singla, J., Anis, A., Reshi, A.A. and Salameh, A.A., 2022. Machine Learning Techniques for Sentiment Analysis of Code-Mixed and Switched Indian Social Media Text Corpus-A Comprehensive Review. International Journal of Advanced Computer Science and Applications, 13(2).

[2] Kirkwood, L. (2019, August 28). 72% of Millennials Value a Good Marketing Campaign When Booking Travel Arrangements.

|  |  |
| --- | --- |
| [3]Șchiopu, A.F., Pădurean, A.M., Țală, M.L. and Nica, A.M., 2016. The influence of new technologies on tourism consumption behavior of the millennials. Amfiteatru Economic Journal, 18(Special Issue No. 10), pp.829-846. |  |

[4] Hassan, M.M., (2020). A review on sentiment analysis techniques in the hospitality industry. International Journal of Advanced Computer Science and Applications.

[5] TripAdvisor. (2017). TripAdvisor Data Analysis: 435 Million Reviews and Opinions Submitted Every Minute. Retrieved from https://ir.tripadvisor.com/news-releases/news-release-details/travelers-push-tripadvisor-past-1-billion-reviews-opinions

[6] [Puh, K.](https://www.emerald.com/insight/search?q=Karlo%20Puh) and [Bagić Babac, M.](https://www.emerald.com/insight/search?q=Marina%20Bagi%C4%87%20Babac) (2022), "Predicting sentiment and rating of tourist reviews using machine learning", [Journal of Hospitality and Tourism Insights](https://www.emerald.com/insight/publication/issn/2514-9792), Vol. ahead-of-print No. ahead-of-print. <https://doi.org/10.1108/JHTI-02-2022-0078>.

[7] Zhang, 2020, "Sentiment Analysis of Customer Reviews on Chinese Travel Website Ctrip," Journal of Tourism and Hospitality, vol. 23, pp. 45-55.

[8] Zhang, 2021, "An Analysis of Sentiment in Hotel Reviews using Machine Learning Techniques," Journal of Tourism and Hospitality, vol. 26, pp. 55-68.

[9] Srivastava, A. (2015). Sentiment Analysis of Movie Reviews using Random Forest and Naive Bayes Classifiers. Journal of Information and Communication Technology, 14(3), 1-8.

[10] Chaturvedi, A. (2016). Sentiment Analysis of News Articles using NaiveBayes and Support Vector Machines. International Journal of Computer Science and Information Technologies, 7(2), 212-216.

[11] Li, Y., Zhang, Q., Sun, L., & Zhang, C. (2018). Sentiment analysis based on LSTM. In 2018 IEEE International Conference on Big Data and Smart Computing (BigComp) (pp. 513-519). IEEE.

[12] Yang, L. (2018). Comparing the Performance of Random Forest and NaiveBayes for Sentiment Analysis of Hotel Reviews. International Journal of Advanced Computer Science and Applications, 9(5), 300-310.

[13] He, X., Zou, H., Zhang, W., & Wang, Y. (2020). Comparative study of Random Forest, Naive Bayes and Long Short-Term Memory for sentiment analysis. IEEE Access, 8, 67553-67560. <https://doi.org/10.1109/ACCESS.2020.2976857>

[14] Ahn, J., Kim, S., & Kim, J. (2016). Sentiment analysis of online hotel reviews based on topic modeling. International Journal of Hospitality Management, 55, 80-90.

[15] Liu, Y., Li, X., and Wang, W. (2017). A study of different interpretive strategies for sentiment analysis. Proceedings of the 2017 International Conference on Informatics, Computing and Security, pp. 517-523.

[16] Zhang, C., Huang, Y., and Zhao, X. (2019). Effects of user characteristics on sentiment analysis accuracy. Proceedings of the 2019 International Conference on Artificial Intelligence and Machine Learning, pp. 197-202.

[17] Chen, C., & Liu, B. (2012). Sentiment analysis and subjectivity. Handbook of natural language processing, 2, 267-296.

[18] Go, A., Bhayani, R., & Huang, L. (2009). Twitter sentiment classification using distant supervision. CS224N Project Report, Stanford, 1-12.

[19] R. Khan, A. Karim, and M. S. Hussain, “Sentiment analysis on twitter data using support vector machine and long short-term memory networks,” Expert Systems with Applications, vol. 92, pp. 1-13, 2018.

[20] Naveen, P., & Dinesh, V. (2017). Comparison of supervised learning algorithms for sentiment analysis of movie reviews. Computational Intelligence and Neuroscience, 2017.

[21] Pang, B., & Lee, L. (2002). Sentiment analysis using support vector machines with linear and non-linear kernels. Proceedings of the ACL-02 conference on Empirical Methods in Natural Language Processing - EMNLP '

[22] Gers, F. A., Schmidhuber, J., & Cummins, F. (2000). Learning to forget: Continual prediction with LSTM. Neural computation.

[23] Brownlee, J. (2018). A Gentle Introduction to the Random Forest Algorithm. Retrieved from <https://machinelearningmastery.com/gentle-introduction-random-forest-algorithm/>

[24] N. Prasad, K. S. Balasubramanian, and S. R. Kulkarni, “Support Vector Machines: A Comprehensive Review,” IEEE Access, vol. 8, pp. 1845–1871, 2020.

[25] G. Y. Yeh, A. Wang, and J. S. Chang, “Sentiment analysis with naive bayes,” in Proceedings of the IEEE International Conference on Computational Intelligence and Applications, pp. 320–324, Dec. 2016.

[26] G. V. Rao, “Term Frequency-Inverse Document Frequency (TF-IDF): A Review,” Int. J. Res. Math. Sci. Engg. (IEEE), vol. 5, no. 1, pp. 11–15, 2019.

[27] Liu, J., & Zhang, Y. (2018). Natural Language Processing: A Guide to Getting Started. Retrieved from <https://medium.com/@datamonsters/text-preprocessing-in-python-steps-tools-and-examples-bf025f872908>

[28] Kumar, P. (2020). Text Data Preprocessing using Python. Towards Data Science. Retrieved from <https://towardsdatascience.com/text-data-preprocessing-using-python-d80874676e79>

[29] Stoyanchev, M. (2020). Evaluation Parameters in Sentiment Analysis. Medium. <https://medium.com/@mstoyanchev/evaluation-parameters-in-sentiment-analysis-5dbfda11e250>

[30] Hutto, C.J. & Gilbert, E.E. (2014). VADER: A Parsimonious Rule-based Model for Sentiment Analysis of social media Text. Eighth International Conference on Weblogs and social media (ICWSM-14). Ann Arbor, MI, June 2014.

[31] Datafiniti Business Database. (n.d.). Retrieved from <https://www.kaggle.com/datafiniti/hotel-reviews>

[32] [28] Sklearn documentation - TfidfVectorizer: [https://scikit-learn.org/stable/modules/generated/sklearn.feature\_extraction.text.TfidfVectorizer.html](https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html" \t "_new)

[33] G.C. Wyser-Pratte, "What Can You Learn From Reviews Over the Years?," in Quality and Service: A Guide to Hospitality, ed. D.D. Blaise (New York, NY: Routledge, 2020).

[34] Kumar, P. (2020). Preprocessing and Cleaning Text Data Using Stop Words, POS Tagging, and NLTK Corpus. Journal of Big Data, 7(6). <https://doi.org/10.1186/s40537-020-00204-4>

[35] Khan, M. (2020). The Need for a List of Top Hotels With Negative Reviews. Retrieved from <https://www.hotelmanagement.net/need-list-top-hotels-negative-reviews>

[36] M.T. Ansari, I.A. Chikte and S.S. Chikte, “Word cloud for hotel reviews – A review,” in IEEE International Conference on Technology for Education (T4E), 2017.

[37] Chen Y, Zhong Y, Yu S, Xiao Y, Chen S. Exploring Bidirectional Performance of Hotel Attributes through Online Reviews Based on Sentiment Analysis and Kano-IPA Model. Applied Sciences. 2022; 12(2):692. <https://doi.org/10.3390/app12020692>

[38] Chen et al., 2020, "Sentiment Analysis of Customer Reviews on Chinese Travel Website Mafengwo," Journal of Tourism and Hospitality, vol. 25, pp. 67-78.

[39] Cristina Franciele, Thays Christina Domareski Ruiz. Using sentiment analysis in tourism research: A systematic, bibliometric, and integrative review. Journal of Tourism, Heritage & Services Marketing, 2021, 7 (2), pp.16-27. ff10.5281/zenodo.5548426ff. ffhal-03373984f

[40]Ortiz-Cordova, A., Yang, Y. and Jansen, B.J., 2015. External to internal search: Associating searching on search engines with searching on sites. Information Processing & Management, 51(5), pp.718-736.

[41] Wang, W., Guo, L. and Wu, Y.J., 2022. The merits of a sentiment analysis of antecedent comments for the prediction of online fundraising outcomes. Technological Forecasting and Social Change, 174, p.121070.

|  |  |
| --- | --- |
| [42] Șchiopu, A.F., Pădurean, A.M., Țală, M.L. and Nica, A.M., 2016. The influence of new technologies on tourism consumption behavior of the millennials. Amfiteatru Economic Journal, 18(Special Issue No. 10), pp.829-846. |  |

[43] Gao, J., & Liu, B. (2018). A comprehensive survey of data preprocessing for text data mining. Journal of computer science and technology, 33(3), 399-407.

[44] Zhou, J., & Hu, X. (2018). A comprehensive survey of text data preprocessing. International Journal of Machine Learning and Cybernetics, 9(2), 249-268.

[45] "Sentiment Analysis on Twitter Data Using TF-IDF and SVM" by A.A. Jagtap and P.P. Pawar (2016)

"Sentiment Analysis of Amazon Product Reviews Using TF-IDF and Naive Bayes Classifier" by S.V. Uppal and G.N. Suresh (2017)

[46] "Effective Sentiment Analysis Using TF-IDF and Random Forest Classifier" by S.K. Das and S. Maji (2019)