THESIS PAPER

Fake Review Detection Using Machine Learning and Deep Learning Algorithms

Prepared by

Name: Ankan Podder

Student ID: 2018-2-50-048

Name: Fayeja Farhana

Student ID: 2018-2-50-024

Name: Farzana Afroze

Student ID: 2018-2-50-007

Supervised by

Dr. Mohammad Arifuzzaman

Associate Professor

This Thesis Paper is Submitted in Partial Fulfillment of the Requirements for the Degree of Bachelor of Science in Information and Communications Engineering

DEPARTMENT OF ELECTRONICS & COMMUNICATIONS ENGINEERING

EAST WEST UNIVERSITY

APPROVAL

The thesis paper titled “Fake Review Detection Using Machine Learning and Deep Learning Algorithms” submitted by Ankan Podder (Student ID: 2018-2-50-048), Fayeja Farhana (Student ID: 2018-2-50-024) and Farzana Afroze (Student ID: 2018-2-50-007) to the Department of Electronics and Communications Engineering, East West University, Dhaka, Bangladesh has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of Bachelor of Science in Information and Communications Engineering and approved as to its style and contents.

Approved By

(Supervisor)

Dr. Mohammad Arifuzzaman

Associate Professor

ECE Department

East West University

Dhaka, Bangladesh

DECLERATION

We declare that our work has not been previously submitted and approved for the award of a degree by this or any other University. As per of my knowledge and belief, this paper contains no material previously published or written by another person except where due reference is made in the paper itself. We hereby, declare that the work presented in this thesis paper is the outcome of the investigation performed by us under the supervision of Dr. Mohammad Arifuzzaman, Associate Professor, Department of Electronics & Communications Engineering, East West University, Dhaka, Bangladesh.

Countersigned

(Supervisor)

Dr. Mohammad Arifuzzaman

Signature

Ankan Podder

Student ID: 2018-2-50-048

Signature

Fayeja Farhana

Student ID: 2018-2-50-024

Signature

Farzana Afroze

Student ID: 2018-2-50-007

DEDICATION

This paper is dedicated

To

Our beloved parents and honorable teachers

ACKNOWLEDGEMENT

We would like to extend our gratitude to our supervisor, Dr. Mohammad Arifuzzaman for his inspiring words and sincere guidance to complete this work in an efficient way. His constant support gave us the assurance we needed to work on this paper. Our completion of this paper could not have been accomplished without the support of our honorable supervisor. We are truly honored and thankful to have him as a supervisor in the journey of our thesis. We are also grateful to our honorable faculty members and office stuffs who rendered their help during the period of our thesis work.

ABSTRACT

With the rapid growth of e-commerce platform, the Fake Review Detection has become a popular and significant topic for both the businesses and research area in recent years. For making a best decision, online reviews play a very important role in today's e-commerce. There are millions of reviews available regarding various product and services in different social sites and marketing websites. Customers write their opinion based on their experience and these reviews become the source of information for the other consumers because depend on those reviews, people take their decision. But sometimes, it becomes so tough to find out the genuine reviews as there is no restriction for written a review in any online platform. Anyone can write reviews according to them which raise the number of fake reviews and can give wrong information and mislead a customer. These fake or genuine reviews play a significant role for the reputation and revenue of an organization. Usually, positive reviews attract more customers and gain high profit whereas negative reviews badly affect the reputation of an organization. This situation makes us interested to distinguish the fake and genuine review. In this study, we discuss some Machine Learning algorithms (Naïve Bayes, Random Forrest, K-Nearest Neighbor, Logistic Regression, Support Vector Machine (Linear), Decision Tree) and Deep Learning algorithms (Dense Layer Architecture, LSTM, BiLSTM) and using those algorithms we try to identify whether the review is credible or not. We applied confusion matrix and analyzed the experimental results. Then, we show some trade-off between Machine Learning and Deep Learning algorithms. Also, we discuss about some challenges we faced while doing this thesis and we discuss some of our future plan regarding this thesis.

Table of Contents

APPROVAL i

DECLERATION........ ii

DEDICATION........ iii

ACKNOWLEDGEMENT........ iv

ABSTRACT........ v

List of Figures........ ix

List of Tables xii

CHAPTER ONE: Introduction 1

1.1 Introduction........ 1

1.2 Background 2

1.3 Problem Statement........ 3

1.4 Motivation........ 4

1.5 Thesis Organizations........ 6

CHAPTER TWO: Literature Review........ 7

CHAPTER THREE: **Introduction to Fake Review Detection**........ 12

3.1 Fake Review Detection........ 12

3.1.1 Understanding Review........ 12

3.1.2 Understanding Fake Review 12

3.1.3 Impact of Sentiment Analysis in Fake Review Detection 13

3.1.4 Importance of Fake Review Detection 13

3.2 Antecedent and Consequent of Fake Reviews........ 14

3.2.1 Antecedent of Fake Review 14

3.2.2 Consequent of Fake Review 14

3.3 Source of Fake Reviews........ 15

3.4 Methods of Fake Review Detection 15

3.4.1 Feature Engineering Extraction 16

3.4.2 Sentiment Analysis Approaches 17

3.5. Challenges in Fake Review Detection 21

CHAPTER FOUR: Introduction to Machine Learning Algorithms 22

4.1 Multinomial Naive Bayes 23

4.2 Bernoulli Naive Bayes 24

4.3 Random Forrest 25

4.4 K-Nearest Neighbors 26

4.5 Logistic Regression 27

4.6 Decision Tree 29

4.7 Support Vector Machine 29

4.8 Evaluation Parameters 30

CHAPTER FIVE: Introduction to Deep Learning Algorithms 32

5.1 Simple Dense layer 32

5.2 Recurrent Neural Network (RNN) 33

5.3 Long Short-Term Memory (LSTM) 34

5.4 Bidirectional Long Short-Term Memory (BiLSTM) 37

CHAPTER SIX: Balanced and Imbalanced Dataset 38

CHAPTER SEVEN: Research Methodology 41

7.1 Data Pre-Processing 48

7.1.1 Clean Text 48

7.1.2 Remove Stopwords 49

7.1.3 Lemmatization 50

7.2 Feature Extraction 51

7.2.1 Bag of Words 51

7.2.2 TF-IDF 52

7.3 Splitting Training and Testing data.. 54

CHAPTER EIGHT: Result and Analysis.. 55

8.1 Applying Confusion Matrix for Machine Learning Algorithms.. 62

8.2 Applying Deep Learning.. 67

CHAPTER NINE: Machine Learning vs Deep Learning 73

9.1 Advantages of Machine Learning 73

9.2 Challenges of Machine Learning 74

9.3 Advantages of Deep Learning 74

9.4 Challenges of Deep Learning... 75

9.5 Final Comparison of ML and DL 76

CHAPTER TEN: Conclusion 78

10.1 Research Challenges 78

10.2 Future Work.. 78

10.3 Conclusion 79

REFERENCES.. 80

List of Figures

Figure 3.1: Sentiment Analysis Approaches ... 17
Figure 3.2: Working of a Lexical Technique ... 18
Figure 3.3: Steps Involved in Machine Learning Approaches ... 19

Figure 4.1: Random Forest Classifier ... 25
Figure 4.2: k-Nearest Neighbor algorithm ... 27
Figure 4.3: Logistic regression ... 28
Figure 4.4: Support Vector Machine ... 30

Figure 5.1: Recurrent Neural Network ... 34
Figure 5.2: BiLSTM ... 37

Figure 6.1: Balanced Vs Imbalanced Dataset ... 38
Figure 6.2: Confusion Matrix of Imbalanced Dataset ... 39
Figure 6.3: Confusion Matrix of Balanced Dataset ... 40

Figure 7.1: Hotel Dataset ... 41
Figure 7.2: Distribution of reviews of hotel dataset ... 42
Figure 7.3: Distribution of hotels ... 42
Figure 7.4: represents the distribution of most used words in overall dataset: ... 43
Figure 7.5: represents the distribution of most used words in the non-fake reviews: ... 43
Figure 7.6: Distribution of most used words in the fake reviews ... 43
Figure 7.7: Restaurant Dataset ... 44
Figure 7.8: Rating distribution ... 45
Figure 7.9: Useful count distribution ... 45
Figure 7.10: RestaurantID distribution ... 46
Figure 7.11: Most used words in overall dataset ... 46
Figure 7.12: Most used words in the non-fake reviews ... 47
Figure 7.13: Most used words in the fake reviews ... 47
Figure 7.14: Proposed Model for Fake Review Detection ... 48

Figure 7.15: Deceptive Opinion Spam Hotel Dataset.. 49
Figure 7.16: The Yelp Restaurant Review Dataset.. 49
Figure 7.17: Deceptive Opinion Spam Hotel Dataset.. 49
Figure 7.18: The Yelp Restaurant Review Dataset.. 50
Figure 7.19: Deceptive Opinion Spam Hotel Dataset.. 50
Figure 7.20: The Yelp Restaurant Review Dataset.. 50
Figure 7.21: Deceptive Opinion Spam Hotel Dataset.. 50
Figure 7.22: The Yelp Restaurant Review Dataset.. 50

Figure 8.1: Plotting of review length vs word frequency 61
Figure 8.2: Plotting of review length vs word density.. 61
Figure 8.3: Confusion matrix of Multinomial Naive Bayes 62
Figure 8.4: Confusion matrix of Bernoulli Naive Bayes 62
Figure 8.5: Confusion matrix of Random Forrest.. 63
Figure 8.6: Confusion matrix of Logistic Regression.. 63
Figure 8.7: Confusion matrix of k-Nearest Neighbors 63
Figure 8.8: Confusion matrix of Decision Tree 64
Figure 8.9: Confusion matrix of Support Vector Machine 64
Figure 8.10: Confusion matrix of Multinomial Naive Bayes 64
Figure 8.11: Confusion matrix of Bernoulli Naive Bayes 65
Figure 8.12: Confusion matrix of Random Forrest.. 65
Figure 8.13: Confusion matrix of Logistic Regression.. 65
Figure 8.14: Confusion matrix of k-Nearest Neighbors 66
Figure 8.15: Confusion matrix of Decision Tree 66
Figure 8.16: Confusion matrix of Support Vector Machine 66
Figure 8.17: Dense Layer Architecture for Hotel Dataset 68
Figure 8.18: RNN-LSTM for Hotel Dataset.. 69
Figure 8.19: RNN-LSTM for Hotel Dataset.. 69
Figure 8.20: Dense Layer Architecture for Restaurant Dataset.. 70
Figure 8.21: RNN-LSTM for Restaurant Dataset.. 71
Figure 8.22: RNN-BiLSTM for Restaurant Dataset.. 72

Figure 9.1: Overfitting and Underfitting... 76

List of Tables

Table 2.1 Summary of previous studies on fake review detection 9

Table 7.1 Summery of the datasets 51
Table 7.2 Bag of Words 52
Table 7.3 Frequency list of words 52
Table 7.4 Term Frequency 53
Table 7.5 IDF 53
Table 7.6: TF-IDF values 53

Table 8.1: Hotel Dataset 55
Table 8.2: Restaurant Dataset 58
Table 8.3: Comparison of Hotel and Restaurant Datasets Using Machine Learning Algorithms 60

Table 9.1: Comparison of ML and DL algorithms 77

CHAPTER ONE

Introduction

1.1 Introduction

Reviews are statements which express suggestion, opinion or experience of someone about any market product **[1].** In this era of the internet, online reviews getting much popularity. People are very much interested in to share their experience of using various product or visiting any place. They post their review or their opinion on several online websites. Those reviews are very much helpful for various organizations or consumers who are interested in buying the similar products or getting the services. Depends on those reviews they get an idea and can take a decision before making a selection. In recent years, customers reviews have been increased rapidly and these reviews significantly affects the buyer's decision. In other words, when customers see reviews on social media, they determine whether to buy the product or reverse their purchasing decisions. Therefore, consumer reviews offer an invaluable service for individuals **[2].**

Reviews can be both positive or negative. Positive reviews always bring a huge financial gain whereas negative reviews badly affect the marketplace or any organization. Both these positive or negative reviews badly affect the customers decision because when they want to take a decision for buying a particular product or getting service from any organizations, they always check the customers feedback regarding those products or services and depending on those feedbacks, they decide whether they should go for this one or not. Therefore, customer's reviews or their feedbacks are becoming a very useful source of information for taking any decision for the customers. For example, before visiting a place, people always search for the hotel or cottage near the place. They visit different hotel and cottage's websites and check the customers reviews or opinions regarding the hotel management or services. Depending on the positive or negative reviews they have got, they take decision for booking a room to a particular hotel. Thus, historical reviews became very credible sources of information to most people in several online services **[3].**

Since, people are allowed to express or share their positive or negative opinion in different social sites, the sites containing the consumer reviews contribute a lot to the marketplaces. People can openly post their feedback or critiques of any company at any time without any kinds of obligations or limitations. This lack of restriction leads many such company to engage themselves with various types of unfair means. Sometimes, a lot of company use different social sites to unfairly promote their goods or services and sometimes they unfairly critique their competitor which misleads the consumers. In order to promote the company and make a higher profit, the owner sometimes hire some people to post negative feedback about their rival's products. Since, reviews are considered forms of sharing authentic feedback about positive or negative services, any attempt to manipulate those reviews by writing misleading or inauthentic content is considered as deceptive action and such reviews are labelled as fake **[3].** Accordingly, a person who posts fake reviews is called a spammer **[4].** Such types of fake reviews lead us to think about the credibility of the available feedbacks or reviews on different social sites. We, the consumer becomes confused to distinguish the fake and non-fake reviews and we doubt that may be all the posted feedbacks are fake or may be these all are authentic. In such situation, we badly need to determine the authenticity of the consumer's opinion or reviews and for that the 'Detection of Fake Reviews' become and still in the state of effective and most demanding research area.

So far, many studies and investigations have been made for detecting the fake and genuine reviews and also the challenges of its. Initially fake review detection was introduced by Jinal et al. **[5].** There are various ways to identify fake reviews. Machine learning and Deep learning techniques are two of them. The main task of fake review detection is to classify the fake and non-fake reviews.

In this study, we have tried to detect the fake and genuine reviews by applying some Machine Learning and Deep Learning Algorithms. Here, we tried to show some comparison between these two algorithms for computing the accuracy for our dataset. Machine Leaning and Deep Learning techniques provide a big contribution to the detection of the fake review which will help the researcher who are interested in the field of fake review detection, can choose the best Machine Learning and Deep Learning method. Also, this paper will help the reader to easily understand the field of fake review detection and the importance of this field.

1.2 Background

The era of social media and dramatic improvement on Internet changed the life style of people for making product purchase online. The product purchase is relied on reviews or comments of customers who earlier done online purchase of products. The large volume of reviews and some other information regarding products available on the social media websites make challenging for the public for better decision or sometimes leads wrong decision on purchasing products. Also, large volume of products comments with sellers and manufactures can confuse customers for right decision of product purchasing. Thus, the problem arises from reviews or comments, star ratings posted on social media and websites **[6]**. Because of that, sometimes people getting confused about the authenticity of those comments or reviews and they cannot take decision based on those reviews.

Fake reviews decrease informativeness, information quality, and the effective use of online product reviews. Fake reviews also damage the credibility of reviews, and negatively affect review helpfulness. In addition, fake reviews seriously affect the development of online product reviews and stakeholder's commitment to the reduction of information asymmetry between merchants and customers **[7]**.

Fake review detection task is one of the challenging classification tasks in the field of knowledge discovery. Multiple angles of capturing deception in reviews data have been focused by researchers for a decade. Focus of our research work is to investigate the techniques and classification model to identify individual fake reviews by analyzing different perspective of review data.

In case of customer, there are not enough information available to accept and believe the online reviews as genuine or fake. On the other hand, because of the business competition or business politics, the opponent sellers and manufacturers also post fake reviews for degrading the reputation of opponent product and brand. Considering these issues, we choose to research in this area. Our proposed work is carried with restaurants reviews and we try to detect the fake and genuine reviews by applying some machine leaning and deep learning algorithms.

1.3 Problem Statement

Fake reviews are inconsistent with real evaluations of products or services. Thus, fake reviews are false, bogus, and deceptive reviews which are posted by different types of people like consumers and online merchants. Fake reviews either belong to positive or negative polarity. The reviews which contain praising statement about any of the product, these are included into the positive polarity and containing loathing statements about products are fall in negative polarity.

All of the reviews have a significant effect on product perception. For building and maintaining good reputation, reviews play crucial role in the e-commerce platform. Besides that, these are effective for making a decision for purchasing any products for the end users. These fake reviews not only mislead customers for taking decision regarding purchasing product but also many businesses containing good quality products also suffer because of these fake reviews. Increasing need of detecting fake reviews makes the fake review detection a befitting topic for the researcher nowadays.

A lot of researches have been going on in this field to find out a proper solution for detecting the fake reviews. In our case, we have tried to model our research by applying two techniques: one is Machine Learning Algorithm and another one is Deep Learning Algorithm on two different datasets. We can summarize our ultimate goal of this research work on the field of Fake Review Detection is as follows:

i. Selecting two datasets relevant to this research topic
ii. Pre-processing the targeted data before applying different techniques
iii. Analyzing datasets using Machine Learning Algorithms
iv. Analyzing datasets using Deep Learning Algorithms
v. Presenting comparison between the results after applying these two techniques
vi. Analyzing the results and give a conclusion from our findings of this research

1.4 Motivation

Apart from traditional markets and superstores, buyers are drawn to online marketplaces because they escape excessive traffic, long wait times, public gatherings, and the recent Covid-19 lockdown. With the rapid technological development, customers are increasingly relying on product reviews for information before purchasing any product. Fake reviews, on the other hand, make online reviews less valuable by giving an inaccurate impression of product quality. As a result, fake review detection is badly needed for recent days. Unfortunately, automatic detection has only had little success in this difficult task so far.

In the United States, more than 80% of consumers indicate they use online reviews before purchasing a product (Smith and Anderson, 2016). Although the oldest members of Generation Z are only now reaching adulthood, their purchasing power is considerable. To be relevant, marketers must understand this generation's particular needs and spending habits [by Coral Ouellette on January 7, 2022]. This generation, born after 1998, is expected to have $44 billion in purchasing power. Currently, 93% of parents claim their Gen Z child has an impact on household expenses. In just a few years, this generation will dominant for 40% of all consumer purchases [by Coral Ouellette on January 7, 2022]. 95% of this group have a smartphone, which they use for roughly 10 hours per day or more. As a result, they are 2X as likely than millennials to shop on mobile devices. However, using social platforms, 85% of them use social media to learn about new products [by Coral Ouellette on January 7, 2022].

Keep these online shopping statistics in mind, we can understand the future of the online shopping. As reviews are one of the most influential factors in consumers' purchasing decisions, fraudulent actors may be tempted to hire or use automated methods to generate fake reviews in order to boost the attractiveness of their products and services or to harm competitor’s reputations. We can predict that, in near future, we will experience a large number of people will go for the online shopping instead of the traditional shops which increase a huge number of reviews in different online platform. As people are more likely to dependent on those reviews for getting the idea about the quality of products and making their purchasing decision based on those reviews, there is a chance to increase the number of fake reviews as well. Therefore, the fake review detection still in the state of effective and most demanding research area and getting an online platform for detecting fake reviews is a very necessary tool for recent days and for the future as well.

Taking the necessity of distinguishing the fake and genuine review into consideration, we choose to work on this field. For this research, we have used two datasets: one is from hotel reviews and another one is restaurant reviews. We used both the machine learning and deep learning algorithms on these two datasets. The motivation behind using these two algorithms on our targeted data is that, we can find out the most accurate algorithms for detecting the fake reviews and we can get a comparison after applying these two techniques. This research findings will help us building an automated fake review detection platform in future.

1.5 Thesis Organizations

Our whole thesis work consists of total ten chapters. In our first chapter we have included introduction, background, problem statement, motivation and thesis organization of our paper. In the second chapter we have presented the literature review where we have briefly discussed similar works have been before regarding our work. In the third chapter we have given the basic idea about fake review detection. In our fourth and fifth chapters we have discussed the implemented machine learning and deep learning algorithms. In the sixth chapter we have shown our datasets in terms of balanced and imbalanced form. Then in our seventh chapter we have presented our research methodology. The eighth chapter represents our research findings and analysis based on machine learning and deep learning algorithms. Then we have given the comparison between machine learning and deep learning in chapter nine. Finally, in chapter ten we have given the concluding words regarding our whole thesis work.

CHAPTER TWO

Literature Review

Fake review identification has received a lot of attention in recent years from both corporations and researchers. Fake reviews must be detected in order for reviews to reflect actual user experiences and opinions **[8]**. There are several research work and different detection approaches have already been published. Some of the worked are reflected below:

Ning Wang et al. **[9]** have proposed a suspicion degree determining method based on the three-dimensional time series, Since the existing methods do not fully consider the time burst characteristics of reviews. Besides, combining the suspicion degree feature, review text features, and reviewer's behavior features together, this paper has proposed a more comprehensive fake review identification framework. The experimental results have showed that the proposed method outperforms the most advanced methods.

Joni Salminen et al. **[10]** have addressed the creation and detection of fake reviews. Firstly, they have experimented with two language models and they are ULMFiT and GPT-2, to generate fake product reviews based on an Amazon e-commerce dataset. Using the better model, GPT-2, the have created a dataset for a classification task of fake review detection. They have showed that a machine classifier can accomplish this goal near-perfectly, whereas human raters exhibit significantly lower accuracy and agreement than the tested algorithms. Their model was also effective on detected human generated fake reviews. Their results imply that, while fake review detection is challenging for humans, “machines can fight machines” in the task of detecting fake reviews.

Nizar Alsharif **[11]** has considered the issue of fake opinion identification in e-commerce businesses based on deep learning recurrent neural network long short-term memory (RNN-LSTM). He has performed experiment using a standard Yelp product review dataset. He also has used a linguistic inquiry and word count dictionary to extract additional linguistic features from the review texts, which has helped to distinguish between real and fake reviews. Instances of these features include: the authenticity of the review’s text, the analytical thinking of the reviewer, negative words, positive words, and personal pronouns. The proposed RNN-LSTM model has reported a better performance for the classification of the reviews into either fake or real categories, achieving results of 98% in terms of accuracy and F1-score.

Hanif Khan et al. **[12]** have proposed a supervised learning-based technique for the detection of fake reviews from the online textual content. The study employs machine learning classifiers for bifurcating fake and genuine reviews. Experimental results are evaluated against different evaluation measures and the performance of the proposed system is compared with baseline works.

Rodrigo Barbado et al. **[13]** have proposes a feature framework for detecting fake reviews that has been evaluated in the consumer electronics domain. The contributions are fourfold: (i) Construction of a dataset for classifying fake reviews in the consumer electronics domain in four different cities based on scraping techniques; (ii) definition of a feature framework for fake review detection; (iii) development of a fake review classification method based on the proposed framework and (iv) evaluation and analysis of the results for each of the cities under study. They have reached an 82% F-Score on the classification task and the Ada Boost classifier has been proven to be the best one by statistical means according to the Friedman test.

Arjun Mukherjee, Vivek Venkataraman and Bing Liu et al. **[14]**, for real review data, they have used filtered (fake) and unfiltered (non-fake) reviews from Yelp.com (which are closest to ground truth labels) to perform a comprehensive set of classification experiments also employing only n-gram features. They have found that fake review detection on Yelp's real-life data only gives 67.8% accuracy, but this accuracy still indicates that n-gram features are indeed useful. After that, they have proposed a novel and principled method to discover the precise difference between the two types of review data using the information theoretic measure KL-divergence and its asymmetric property. This reveals some very interesting psycholinguistic phenomena about forced and natural fake reviewers. To improve classification on the real Yelp review data, they have proposed an additional set of behavioral features about reviewers and their reviews for learning, which dramatically have improved the classification result on real-life opinion spam data.

Petr Hajek et al. **[15]**, have propose two neural network models that integrate traditional bag-of-words as well as the word context and consumer emotions. Specifically, their models learn document-level representation by using three sets of features: (1) n-grams, (2) word embeddings and (3) various lexicon-based emotion indicators. Such a high-dimensional feature representation is used to classify fake reviews into four domains. To demonstrate the effectiveness of the presented detection systems, they have compared their classification performance with several state-of-the-art methods for fake review detection.

Jindal and Liu **[16]** presented the first study aimed at detecting fake product reviews based on the similarity of review and product features. More precisely, spammers' tendency to duplicate their product reviews was used.

To detect spammers who can adapt their behavior, Wang et al. **[17]** proposed a heterogeneous review graph that captures the relationships among reviews, reviewers and reviewed shops. Thus, the trustiness of reviewers, honesty of reviews and reliability of shops could be calculated without considering the review content. Inspired by this approach, Liu et al. proposed a probabilistic graph classifier in which the multimodal embedded representation of nodes is obtained using a bidirectional NN with an attention mechanism.

Table 2.1 Summary of previous studies on fake review detection

Study	Content-based features	Classifier	Dataset	Performance
[18]	Positive/negative words, brand name, similarity of review and product features, numeric and capital words	LR	Amazon	AUC = 0.78
[19]	Unigrams and bigrams, review length, first-person pronouns, similarity with other reviews, ratio of question sentences, ratio of the capital letters, subjective/ objective words, positive/negative words	NB, Co-training	Epinions	F-score = 0.6
[20]	User rating, app rating	DT, LCGM	App Store	
[21]	Unigrams and bigrams	SVM	Hotels	Acc = 0.86
[22]	Frequency of characters, words and punctuation marks	SVM	Hotels	F-score = 0.84
[23]	Unigrams and bigrams	SVM	Yelp	Acc = 0.86

[24]	Unigrams, positive/negative words, spatial words, first-person pronouns	SAGE	Hotels and doctors	Acc = 0.65
[25]	Unigrams and bigrams	SVM	Restaurants	Acc = 0.85
[26]	Review length, content similarity among user’s (product’s) reviews	SSL	Yelp	AUC = 0.79
[27]	Product word embeddings, bigrams and trigrams	Bagging	Hotels, restaurants and doctors	F-score = 0.77
[28]	Sentence weights, POS, first-person pronouns	CNN, SWNN	Hotels, restaurants and doctors	Acc = 0.84
[29]	CBOW word embeddings	CNN, GRNN	Hotels, restaurants and doctors	Acc = 0.84
[30]	Unigrams	k-NN, NB, DT, SVM	Movies	Acc = 0.82
[31]	Bigrams, LIWC, POS	k-NN, RF	Hotels	Acc = 0.77
[32]	CBOW word embeddings	SSL	Yelp	AUC = 0.83
[33]	Unigrams, bigrams, trigrams and four-grams	SVM	Hotels	Acc = 0.90
[34]	First and last sentence, middle context	LSTM ensemble	Hotels, restaurants and doctors	Acc = 0.83
[35]	Positive/negative words, bigrams, LDA	AdaBoost	Yelp	F-score = 0.81
[36]	Skip-Gram word embeddings, review length, capitalized words, numerals, POS, positive/negative words	BERT	Hotels, Yelp	Acc = 0.89
[37]	Positive/negative words, review length, first-person pronouns, multimodal embeddings	LR	Dianping	F-score = 0.81

[38]	Skip-Gram word embeddings, unigrams, bigrams and trigrams	DFFNN	Hotels	Acc = 0.89

Acc accuracy, *AUC* area under *ROC* curve, *BERT* bidirectional encoder representations from transformers, *CNN* convolutional neural network, *DFFNN* deep feed-forward neural network, *DT* decision tree, *FNR* false negative rate, *FPR* false positive rate, *GRNN* general regression neural network, *k-NN* k-nearest neighbor, *LCGM* latent class graphical model, *LDA* latent Dirichlet allocation, *LIWC* linguistic inquiry and word count, *LR* logistic regression, *LSTM* long short-term memory, *NB* Naive Bayes, *POS* part-of-speech tagging, *RF* random forest, *SAGE* sparse additive generative model, *SSL* semi-supervised learning, *SVM* support vector machine, *SWNN* sentence weighted neural network

CHAPTER THREE

Introduction to Fake Review Detection

3.1 Fake Review Detection

In this era of the internet, e-commerce platforms have drastically changed the way of sharing opinion. Sharing opinion is one of the ways to write reviews about products or services. Depending on the available reviews of different online sources, people take their valuable decision. They get an idea about the products before making a selection. Therefore, determining the fake and genuine reviews is essential as it is directly affecting both the organizations and future consumers.

3.1.1 Understanding Review

Basically, online reviews are comments, tweets, posts, opinions which are shared on different online platform like review sites, news sites, e-commerce sites or may be in different social sites. Reviews are considered as individual's personal opinion or experience regarding a product or service **[39].** In recent years, the number of customers reviews have increased significantly which creates an impact on the potential buyer. The way of customer's feedback or experience related to a product or service increasingly influence to the marketplaces as well. There is a growing trend towards relying on customers' opinions to reshape businesses by enhancing products, services, and marketing **[40].**

3.1.2 Understanding Fake Review

Different social media like Facebook, Tweeter or almost every e-commerce website allows consumer to openly share their experience or critiques of any company or product without any obligations or limitations. This lack of restrictions leads some certain companies to unfairly promote their goods or services and sometimes they unfairly critics their competitive companies. Sometimes, a lot of companies hired some person to intentionally write some positive reviews to promote their own products and asked the hired person to write some negative reviews about their rival's products.

Reviews published by people who have not personally encountered the items being reviewed are considered fake reviews **[40].** Accordingly, a person who posts fake reviews is called a spammer **[40].** When the spammer works with other spammers to achieve a specific goal, the spammers are called a group of spammers **[40].** Opinion spamming is an immoral activity of posting fake reviews. The goal of opinion spamming is to misguide the review readers **[1].**

3.1.3 Impact of Sentiment Analysis in Fake Review Detection

Sentiment Analysis (SA) is a powerful technology to detect fake reviews and customer's feelings. Sentiment Analysis (SA) is based on Natural Language Processing (NLP) Technique which is used to extract user's feelings and opinions about any manufactured goods or service provided **[41].** The other name of Sentiment Analysis is Opinion Mining (OM). This is very useful in the decision-making process. One of the main reasons of using Sentiment Analysis (SA) is to extract emotions from opinion, to detect the fake positive and fake negative reviews from opinion of various consumer. Sentiment Analysis is a sort of categorization in which data is divided into many categories. These categories can be binary in nature like positive or negative or they can have many classes like- happy, sad, angry, etc **[42].**

3.1.4 Importance of Fake Review Detection

Because of the rapid growth of the e-commerce business, user's reviews become very important set of data for any organizations. In today's era, for gaining profit for a business organization, user reviews play a very significant role. Because nowadays, a large group of people are highly dependent on checking reviews of another user's before going to make this happen. That is why these reviews become the source of information for another new users. These reviews directly affect company's reputations and profitability. Fake reviews decrease informativeness, information quality, and the effective use of online product reviews **[6].** These also damage the credibility of reviews. For gaining financial profit, a lot of online sellers tend to publish positive reviews for their own company and at the same time they post negative reviews about their competitor's company. These types of wrong information badly affect for gaining profit and customers also misguided. Therefore, detecting the fake review and distinguishing the fake and genuine review become very important issue for both the online sellers and customers.

3.2 Antecedent and Consequent of Fake Reviews

If we want to know why the spammer post fake reviews, opportunity seeking could be the great example in that case. The term 'false information' refers the information about products which includes fake reviews in e-commerce, hoaxes on collaborative platforms and fake news on social media (Pantano, 2020) **[6].** The growing trend in the numbers of published articles related to fake review indicates the issue's fast emergence in research. However, most of this research focuses on detection while antecedents and consequences remain largely unexplored **[6].**

3.2.1 Antecedent of Fake Review

Studies confirm that online product reviews affect consumers' purchase decisions (Heydari et al., 2015), product reputations (Petrescu et al., 2018), sales volumes, and merchants' profits (Dellarocas, 2006) **[6].** Fake reviews are mainly posted by some small owners, small management companies, weak brands, low ratings, and inferior quality. But sometimes. strong brands, high ratings, superior quality, and competitive advantages might also post fake reviews under fierce competition **[6].** Finally, individual consumers may post fake reviews to seek rewards (Thakur et al., 2018); this behavior is broadly rooted in psychological needs that stem from three sources: upset customers, self-appointed brand managers, and social status **[6].** No study has yet explored in detail, why and how do individuals, review platforms, and AI agents post fake reviews. In particular, the underlying psychological mechanisms that cause individuals without external financial incentives to post fake reviews should be scrutinized **[6].** In addition, the motives and the rationales of platforms that post false reviews and the role of AI agents in posting fake reviews remain unclear and call for analysis and studies **[6].**

3.2.2 Consequent of Fake Review

The effects of fake reviews have raised serious concern and various theoretical models are employed to highlight the consequences of fake reviews **[6].** The existence of fake reviews constantly increases the number of online product reviews (Petrescu et al., 2018). Most fake reviews are either positive or negative, whereas few fake reviews are neutral (Luca & Zervas, 2016) **[6].** As a distorted form of online product reviews, fake reviews aggravate the dispersion of review ratings. Fake reviews manipulate consumers' purchase intentions and should directly affect product sales/revenues **[6].** However, the influence degrees of fake reviews on the development of online reviews should be quantified and further studies should be undertaken **[6].**

3.3 Source of Fake Reviews

There are multiple categories of inauthentic or fake reviews. Here are some common sources from where most of the fake reviews come from **[43]**:

- **Global Vendors:** There are some service provider or vendors available who sell both the positive and negative reviews to businesses around the world. Basically, these reviews can be found while trying to find or buy reviews from different online websites.
- **Business Owners and Marketers:** Business owners generate some positive or negative reviews directly or indirectly. Most of the times, they write positive review by themselves for their own benefits and write the negative reviews of their competitors to harm their businesses.
- **Employees:** Generally, current employees write positive reviews for their employers so that they can earn extra benefit whereas the former employees post negative reviews of their former employers in retaliation for being fired or laid off.
- **Friends and Family:** They write positive reviews on behalf of a company or brand they are closely associate with and sometimes they write negative reviews about the competitor of their closed company.
- **Customers:** In order to get refund or other compensations, sometimes customers post negative reviews. They actually lie or exaggerate a bad experience to get discount or some other benefits.

3.4 Methods of Fake Review Detection

The fake reviews detection problem has been tackled since 2007 **[44].** Fake reviews detection has become critical issue for customers to make better decision on products trustworthy as well as the vendors to make their purchase **[45].** The following methods are used for detecting fake reviews:

3.4.1 Feature Engineering Extraction

Feature extraction is one of the major challenges when it comes to detect fake reviews. This basically split into different nodes such as: we have reviewer who posted the review, then we have the actual review and product matching feature and we also can capture some network related information.

3.4.1.1 Review Centric Feature

This approach identifies review as fake review based on the content of reviews written by reviewers. In this method, various features like review content similarity, use of capitals, all capital words, use of numerals, brand name, similarity between products and reviews, repeated use of good and bad words in review **[39],** percentages of pronouns / nouns / adjectives / verbs, lexical validity, lexical diversity, content diversity, active / passive voice, pictures / links etc **[46].**

3.4.1.2 Reviewer Centric Feature

This method depends on the behavior of reviewers **[39].** This approach considers information about users and all reviews that are written by them **[47].** Features used in this method are account age, profile picture, number of written reviews by one reviewer, maximum rating per day **[39],** number of shared/helpful reviews, percentage of positive and negative reviews, ratio of varied purchase, rating deviation, review length etc. **[46].**

3.4.1.3 Product Centric Feature

This method mainly focuses on the product related information. In this method, sales rank of product, price of product etc are considered as features **[39].**

3.4.1.4 Network Centric Feature

This approach includes capturing the IP addresses, GPS information. We can see the timestamp and the pattern of it, if is the hour of the day or week and month. Then we can also see the traffic patterns of sender IP neighborhood density. Whenever this is a spam network, they are usually close to the network and all these reviews during a specific interval of time come from the same IP neighborhood. Also, we can see, from which device the reviews are being posted **[46].**

3.4.2 Sentiment Analysis Approaches

Sentiment analysis has been practiced on a variety of topics. For instance, sentiment analysis studies for movie reviews, product reviews, and news and blogs **[48].** There are some sentiment analysis approaches available for the detection of fake reviews, some of them are discussed in this section.

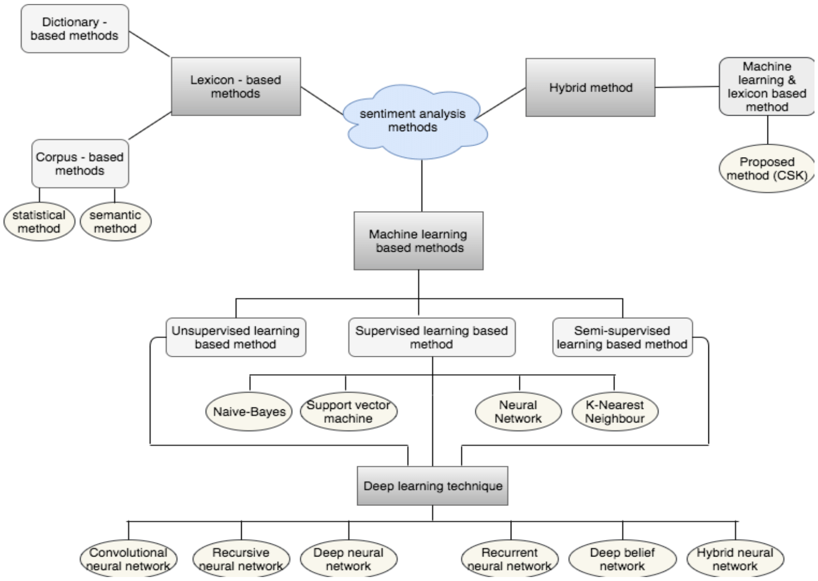


Figure 3.1: Sentiment Analysis Approaches

[Source: https://www.researchgate.net/figure/Techniques-Of-Sentiment-Analysis-31-Machine-Learning-based-TechniqueMachine-Learning_fig1_326200798]

3.4.2.1 Lexicon Based Approaches

The lexicon-based approach uses sentiment dictionary with opinion words and match them with the data for determining polarity. There are three techniques to construct a sentiment lexicon: manual construction, corpus-based methods and dictionary-based methods **[49].** The manual construction is a difficult and time-consuming task. Corpus-based methods can produce opinion words with relatively high accuracy. Finally, in the dictionary-based techniques, the idea is to first collect a small set of opinion words manually with known orientations, and then to grow this set by searching in the WordNet dictionary for their synonyms and antonyms **[49].**

This technique is governed by the use of a dictionary consisting pre-tagged lexicons. The input text is converted to tokens by the Tokenizer **[48].** Every new token encountered is then matched for the lexicon in the dictionary. If there is a positive match, the score is added to the total pool of score for the input text. For instance, if “dramatic” is a positive match in the dictionary then the total score of the text is incremented. Otherwise, the score is decremented or the word is tagged as negative. Though this technique appears to be amateur in nature, its variants have proved to be worthy. Lexical analysis has a limitation: its performance (in terms of time complexity and accuracy) degrades drastically with the exponential growth of the size of dictionary (number of words) **[48].** Figure 3.2 shows the working of a lexical technique.

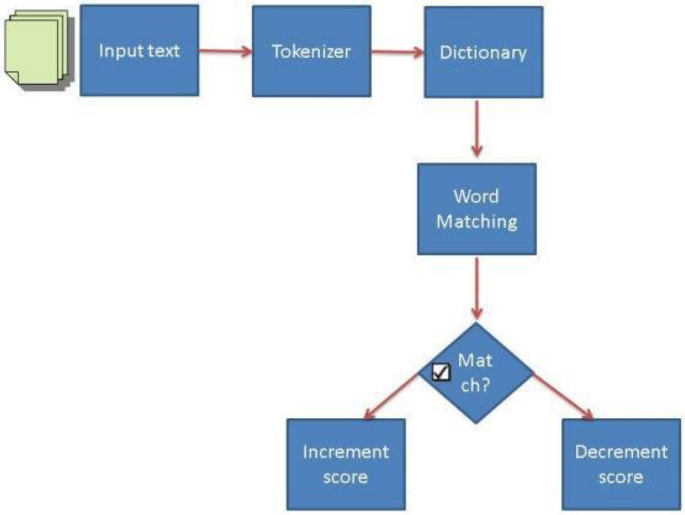


Figure 3.2: Working of a Lexical Technique

[Source: https://www.researchgate.net/figure/Working-of-a-lexical-technique_fig2_285648161]

3.4.2.2 Machine Learning Based Approaches

Machine learning is one of the most prominent techniques gaining interest of researchers due to its adaptability and accuracy. In sentiment analysis, mostly the supervised learning variants of this technique are employed. It comprises of three stages: Data collection, Pre-processing, Training data, Classification and plotting results **[48].** In the training data, a collection of tagged corpora is provided. The Classifier is presented a series of feature vectors from the previous data. A model is created based on the training data set which is employed over the new/unseen text for classification purpose. In machine learning technique, the key to accuracy of a classifier is the selection of appropriate features **[48].** Generally, unigrams (single word phrases), bi-grams (two consecutive phrases), tri-grams (three consecutive phrases) are selected as feature vectors. There are a variety of proposed features namely number of positive words, number of negative words, length of the document, Support Vector Machines (SVM), Naïve Bayes (NB), Neural Network and K-Nearest Neighbour (K-NN) algorithm. Accuracy is reported to vary from 63% to 80% depending upon the combination of various features selected **[48].** The following figure (Figure: 3.3) shows the steps involved in the machine learning approaches:

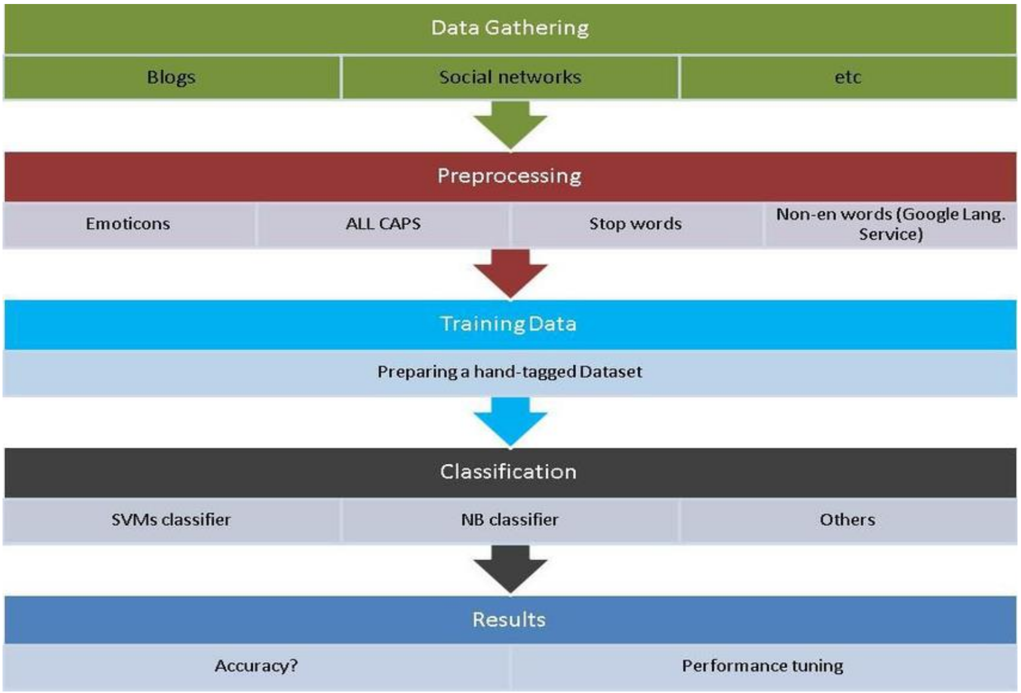


Figure 3.3: Steps Involved in Machine Learning Approaches

[Source: https://www.researchgate.net/figure/Steps-involved-in-the-machine-learning-approach_fig1_285648161]

The machine learning technique faces challenges in: designing a classifier, availability of training data, correct interpretation of an unforeseen phrase. It overcomes the limitation of lexical approach of performance degradation and it works well even when the dictionary size grows exponentially **[48].**

3.4.2.3 Hybrid Approaches

In the hybrid approach, the combination of both the machine learning and the lexicon-based approaches has the potential to improve the sentiment classification performance **[49].** This approach could collectively exhibit the accuracy of a machine learning approach and the speed of lexical approach **[48].** There are some advantages and limitations in using these different approaches depending on the purpose of the analysis. The main advantages of hybrid approaches are the lexicon/learning symbiosis, the detection and measurement of sentiment at the concept level and the lesser sensitivity to changes in topic domain. While the main limitation is that reviews are with a lot of noise (irrelevant words for the subject of the review) are often assigned a neutral score because the method fails to detect any sentiment **[49].**

3.4.2.4 Deep Learning Based Approaches

Deep Learning is a machine learning method based on neural network architectures with multiple layers of processing units, which has been successfully applied to a broad set of problems in the areas of image recognition and natural language processing **[50].** In deep learning, a computer model learns to perform classification tasks directly from images, text, or sound. Deep learning models can achieve state-of-the-art accuracy, sometimes exceeding human-level performance. Models are trained by using a large set of labelled data and neural network architectures that contain many layers. Most deep learning methods use neural network architectures, which is why deep learning models are often referred to as deep neural networks **[51].**

With advances in deep learning over the last decade, numerous deep learning models such as autoencoders, decoders, Unidirectional and Bidirectional Long Short-Term Memory models, Recurrent neural networks, and convolutional neural networks have been used to handle large amounts of data **[52].** These models efficiently aim to discover features or important information in the input dataset intrinsically, and they learn representations and correlations of complexities in the data using heuristic learning to solve diverse NLP problems.

3.5 Challenges in Fake Review Detection

A lot of issues arise, when it comes to detect fake reviews. Sometimes it is difficult to distinguish the fake and authentic reviews because there are some aspects available which influence the fake review detection process. In order to get the accurate information regarding the reviews, we need to carefully manage the detection process. A few of these are discussed in given below:

Sarcasm in Review Text: Sometimes people express sarcasm in review text which may be genuine or fake information related to products. Classification of these reviews into fake and genuine reviews is a difficult task.

- **Implicit Sentiments and Contextual Information in Re- view Text:** Identifying implicit sentiments and contextual information in reviews creates a problem in classifying fake reviews as this combines both behavioral analysis and text analysis.
- **Seller Reviewer Collusion:** It is found that sometimes sellers fix prizes with the websites promoting their product to enhance the value of the product through fake reviews and sharing the profit with the websites. These reviews are written by experienced people and it is very difficult to detect these reviews.
- **Domain Dependence of Classifiers:** The classifiers trained to detect fake reviews in one domain may not give the same results when used in others domain. Cross domain deception opinion detection is one of the active areas which need to be explored.
- **Getting Labelled Training Dataset:** Although there has been a lot of research related to getting labelled dataset for fake review detection, it is still found that getting the right kind of data set is still a major problem and effective methods need to be still explored.

CHAPTER FOUR

Introduction to Machine Learning Algorithms

A machine learning algorithm is a computational process that uses input data to achieve a desired task without being literally programmed (i.e., “hard coded”) to produce a particular outcome. These algorithms are "soft programmed" in the sense that they automatically adjust or adapt their design as a result of repetition (i.e., experience) to get better and better at doing the target objective. Training is the adaptation process, which involves providing samples of input data together with intended consequences. The algorithm then optimizes itself so that it can not only provide the desired result when given the training inputs, but can also generalize to create the desired result when given new, previously unknown data. The "learning" aspect of machine learning is this training. The training does not have to be limited to a single adaption over a set period of time. A good algorithm, like people, can practice "lifelong" learning as it processes fresh data.

A computer algorithm can evolve in a variety of ways in response to training. The input data can be chosen and weighted to produce the best results. Iterative optimization can be used to alter the algorithm's variable numerical parameters. It can arrange a network of possible computational pathways for the best outcomes. It can take the supplied data to generate probability distributions and use them to forecast outcomes.

Machine learning's objective is to mimic how humans (and other sentient organisms) learn to process sensory (input) data in order to achieve a goal. This objective could be a pattern recognition challenge in which the learner must discriminate between apples and oranges. Although each apple and orange are distinct, we can typically distinguish one from the other. Rather than hard-coding a computer with a plethora of accurate representations of apples and oranges, it can be programmed to learn to identify them through repeated exposure to real apples and oranges. Each training example of input data (color, shape, odor, etc.) is coupled with its known categorization label in this supervised learning example (apple or orange). When the items to be categorized contain many changeable attributes within their own classes but yet have fundamental qualities that define them, it allows the learner to cope with similarities and differences. Most significantly, a competent learner should be able to recognize an unfamiliar apple or orange.

The so-called unsupervised algorithm is a second type of machine learning. It's possible that the goal is to toss a dart at a bull's-eye. In the mechanism that regulates the course of the dart, the device (or human) has a variety of degrees of freedom. The student practices throwing the dart rather than attempting to program the kinematics in advance. The kinematic degrees of freedom are changed for each trial such that the dart gets closer and closer to the target. This is unsupervised since the training does not link a specific kinematic input configuration to a certain output. From the training data, the algorithm finds its own path. The trained dart thrower should be able to alter the target.

Semi-supervised learning is a third type of machine learning in which some data is labeled while others are not. In this case, the labeled part can be used to help the unlabeled part learn faster. This scenario is more akin to how humans gain their skills than most natural systems **[53]**.

We have used only supervised machine learning algorithms regarding fake review detection. The supervised machine learning algorithms that we have used for fake review detection are as follows:

1. Multinomial Naive Bayes
2. Bernoulli Naive Bayes
3. Random Forrest
4. K-Nearest Neighbors
5. Logistic Regression
6. Decision Tree
7. Support Vector Machine

4.1 Multinomial Naive Bayes

There are many of software or programs for analyzing numerical data, but only a few for analyzing texts. Multinomial Naive Bayes is one of the most widely used supervised learning classifications for categorical text data analysis.

The Multinomial Naive Bayes algorithm is a Bayesian learning approach popular in Natural Language Processing (NLP) (NLP). Using the Bayes theorem, the program estimates the tag of a text, such as an email or a newspaper piece. It calculates each tag's likelihood for a given sample and outputs the tag with the greatest chance.

Bayes theorem, formulated by Thomas Bayes, calculates the probability of an event occurring based on the prior knowledge of conditions related to an event. It is based on the following formula:

$P(A|B) = P(A) * P(B|A)/P(B)$ .... (4.1)

Where we are calculating the probability of class A when predictor B is already provided.

P(B) = prior probability of B

P(A) = prior probability of class A

P(B|A) = occurrence of predictor B given class A probability

This formula helps in calculating the probability of the tags in the text **[54].**

4.2 Bernoulli Naive Bayes

Bernoulli The Naive Bayes family includes Naive Bayes. It only accepts binary data. The most basic example is determining whether or not a word appears in a document for each value. That is a rudimentary model. Bernoulli may get better results in circumstances where counting the frequency of words is less critical. To put it another way, we must count every value binary term occurrence feature, such as whether a word appears in a document or not. Rather than finding the frequency of a word in the document, these attributes are utilised. $P(X=1)=p$ or $P(X=0)=1-p$ are the mutually exclusive outcomes of the Bernoulli distribution in layman's words. We can have numerous features in the BernoulliNB theorem, but each one is supposed to be unique.

$P(x_i \mid y) = P(i \mid y)\, x_i + (1 - P(i \mid y))(1 - x_i)$ .... (4.2)

According to the decision rule formula, x needs to be binary. Think about the formula in the case where xi=1 and the case where xi=0. So, i is the event where xi=1 or the event where xi=0**[55]**.

4.3 Random Forrest

A forest is a grouping of trees, while a random forest is a grouping of categorization trees. The creation of a tree in which the members of the class variable dwell on the leaf nodes and the entities of other dependent variables reside on the intermediate nodes is known as a classification tree. Class variables, also known as decision variables or predictor variables, can include yes/no decisions to forecast disease or loan acceptance, spam/no spam in emails, good/bad/moderate for product attributes, 0-9 for handwritten digits in pattern recognition, and so on. Random forests produce numerous classification trees, and to add a new classification tree to the forest, add it to each of the individual classification trees.

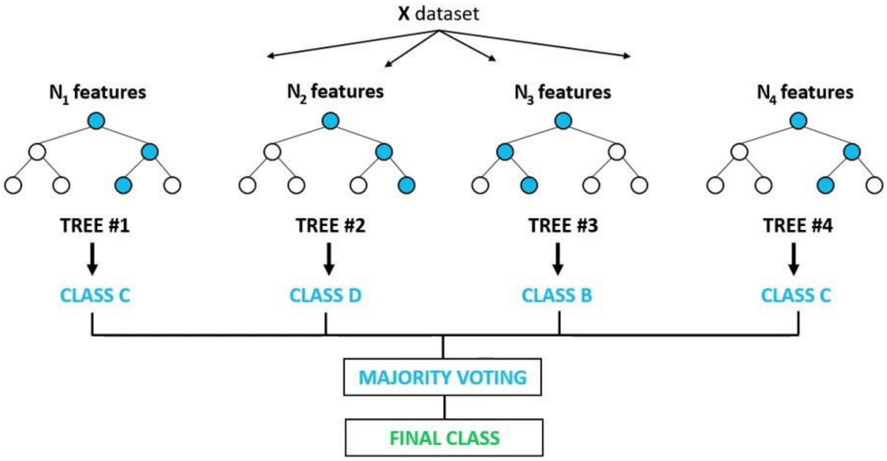


Figure 4.1: Random Forest Classifier

[Source: https://medium.com/analytics-vidhya/random-forest-classifier-and-its-hyperparameters-8467bec755f6 Retrieved: 24th May’2022, 12:36AM]

The Random Forest's Construction:

1. If the training set contains N cases, pick all of them at random as a distinct collection of data from the original data set.
2. Choose Tt attributes at random from the training data set for T number of attributes so that the optimal selection of t variables is chosen to split each node. The value of t should remain constant as the tree grows.

3. Without pruning, each tree will develop to its maximum potential.

The random forest's error rate is determined by the following two factors:

I. The error rate will rise only if and only if the correlation between any two trees in the forest rises.
II. The error rate determines the tree's strength; the lower the error rate, the stronger the tree, and the forest as a whole.

Random Forest Properties:

- It is often regarded as the most accurate algorithm.
- It is extremely efficient on large data sets, even when there are hundreds of thousands of input variables, and there is no need for data pruning.
- It is particularly efficient when it comes to feature subset selection and missing data imputation.
- During the forest development phase, the random forest algorithm generates an internal unbiased estimate of the generalization error.
- The created forest will be suitable for future data addition **[56].**

4.4 K-Nearest Neighbors

K-Nearest Neighbor is one of the simplest Machine Learning algorithms based on Supervised Learning technique. When there is little or no prior knowledge about the distribution of the data, K-nearest-neighbor (kNN) classification is one of the most fundamental and straightforward classification methods. It should be one of the first choices for a classification study. The requirement to perform discriminant analysis when trustworthy parametric estimates of probability densities are unknown or impossible to calculate led to the development of K-nearest-neighbor classification **[57]**.

KNN works by calculating the distances between a query and all of the instances in the data, picking the K closest examples to the query, and then voting for the most frequent label (in the case of classification) or averaging the labels (in the case of regression) **[58].**

Suppose there are two categories, i.e., Category A and Category B, and we have a new data point x1, so this data point will lie in which of these categories. To solve this type of problem, we need a K-NN algorithm. With the help of K-NN, we can easily identify the category or class of a particular dataset. Consider the below diagram:

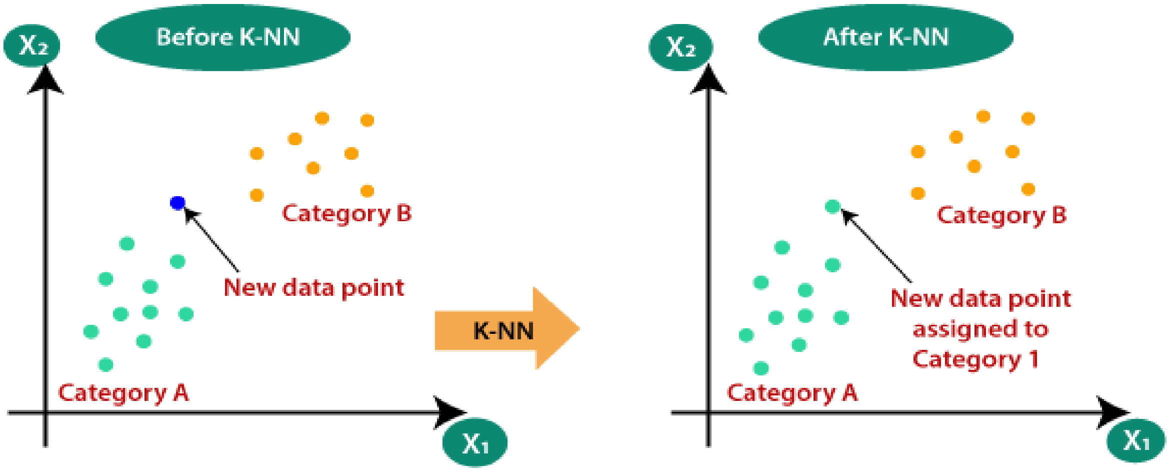


Figure 4.2: k-Nearest Neighbor algorithm

[Source: https://www.javatpoint.com/k-nearest-neighbor-algorithm-for-machine-learning]

The K-NN working can be explained on the basis of the below algorithm:

Step-1: Select the number K of the neighbors

Step-2: Calculate the Euclidean distance of **K number of neighbors**

Step-3: Take the K nearest neighbors as per the calculated Euclidean distance.

Step-4: Among these k neighbors, count the number of the data points in each category.

Step-5: Assign the new data points to that category for which the number of the neighbor is maximum.

Step-6: Our model is ready **[59]**

4.5 Logistic Regression

The classification technique logistic regression is used to assign observations to a discrete set of classes. Email spam or not spam, online transactions fraud or not fraud, and tumor malignant or benign are some examples of classification issues. The logistic sigmoid function translates the output of logistic regression into a probability value. It is based on the concept of probability and is a predictive analytic method.

A Logistic Regression model is similar to a Linear Regression model, except that it uses a more sophisticated cost function, which is known as the 'Sigmoid function' or the 'logistic function' instead of a linear function.

The hypothesis of logistic regression tends it to limit the cost function between 0 and 1. Therefore linear functions fail to represent it as it can have a value greater than 1 or less than 0 which is not possible as per the hypothesis of logistic regression.

$0 \leq h_\theta(x) \leq 1$ .... (4.3)

When using linear regression, we used a formula of the hypothesis i.e.

$h\Theta(x) = \beta_0 + \beta_1 X$ .... (4.4)

For logistic regression we are going to modify it a little bit i.e.

$\sigma(Z) = \sigma\ (\beta_0 + \beta_1 X)$ .... (4.5)

We have expected that our hypothesis will give values between 0 and 1.

$Z = \beta_0 + \beta_1 X$ .... (4.6)

$h\Theta(x) = \text{sigmoid}(Z)$ .... (4.7)

i.e. $h\Theta(x) = 1/(1 + e\hat{\ }{-}(\beta_0 + \beta_1 X))$ .... (4.8)

This is the basic concept of Logistic Regression **[60]**.

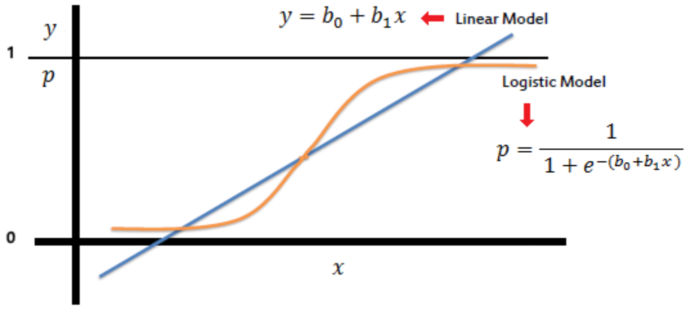


Figure 4.3: Logistic regression

[Source: https://www.saedsayad.com/logistic_regression.htm]

4.6 Decision Tree

The most powerful and widely used tool for categorization and prediction is the decision tree. Each internal node symbolizes a test on an attribute, each branch represents a test outcome, and each leaf node (terminal node) stores a class label.

By separating the source set into subgroups based on attribute value tests, a tree can be "trained." Recursive partitioning is the process of repeating this method on each derived subset. When all of the subsets at a node have the same value of the target variable, or when splitting no longer adds value to the predictions, the recursion is complete. No domain expertise or programming skills are required to build a decision tree classifier.

Instances are classified using decision trees by sorting them down the tree from the root to a leaf node, which provides the classification. As indicated in the above diagram, an instance is classified by starting at the root node of the tree, checking the attribute specified by this node, and then progressing along the tree branch according to the attribute value. This procedure is then repeated for the new node's subtree **[61]**

4.7 Support Vector Machine

Support-vector machines (SVM, also known as support-vector networks) are supervised learning models that examine data for classification and regression analysis in machine learning.

An SVM training algorithm produces a model that assigns new examples to one of two categories, making it a non-probabilistic binary linear classifier, given a series of training examples, each marked as belonging to one of two categories (although methods such as Platt scaling exist to use SVM in a probabilistic classification setting). SVMs can perform non-linear classification as well as linear classification by applying the kernel trick, which involves implicitly translating their inputs into high-dimensional feature spaces **[62]**.

Each data item is plotted as a point in n-dimensional space (where n is the number of features you have), with the value of each feature being the value of a certain coordinate in the SVM algorithm. Then we accomplish classification by locating the hyper-plane that clearly distinguishes the two classes (look at the below snapshot).

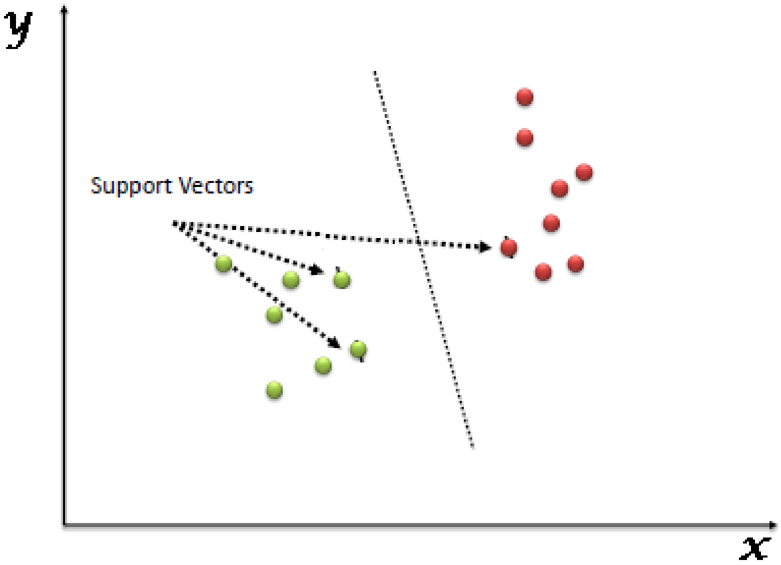


Figure 4.4: Support Vector Machine

[Source: https://www.analyticsvidhya.com/blog/2017/09/understaing-support-vector-machine-example-code/ Retrieved: 25th-January'2022, 9.29PM]

Support Vectors are simply the coordinates of individual observation. The SVM classifier is a frontier that best segregates the two classes (hyper-plane/ line) **[63]**.

These are the algorithms that we have used. The result of these algorithms is presented in the Chapter Seven.

4.8 Evaluation Parameters

To understand classifier model's performance, we need to be familiar with some evaluation parameters. A confusion matrix is a table that is used to describe the performance of a classifier algorithm by evaluating the accuracy of it. The elements of confusion matrix are:

True Positive (TP): Which results when classifier model correctly predicts the positive class.

True Negative (TN): Which results when classifier model correctly predicts the negative class.

False Positive (FP): Which results when classifier model incorrectly predicts the positive class.

False Negative (FN): Which results when classifier model incorrectly predicts the negative class.

Table 3.1: Confusion Matrix

	Predicted Values		
Actual Values		Predicted Positive (1)	Predicted Negative (0)
	Actual Positive (1)	True Positive (TP)	False Negative (FN)
	Actual Negative (0)	False Positive (FP)	True Negative (TN)

Based on the data of confusion matrix, precision, recall, F-measure, and accuracy are the evaluation measures used for evaluating performance of classifier **[64]**.

Precision: Precision is the ratio of correctly predicted positive results to the total predicted positive results. It measures the exactness of the classifier result **[64]**.

$$Precision = \frac{TP}{TP+FP} \text{ .... (4.9)}$$

Recall: Recall measures how accurately classifier model identifies and returns True Positives data. It also refers as True Positives rate. A higher recall is essential for a better classifier model.

$$Recall = \frac{TP}{TP+FN} \text{ .... (4.10)}$$

F-measure: F-measure also refers as F-1 score, is the harmonic mean of the precision and recall. It is required to optimize the system towards either precision or recall which have a more influence on final result **[64]**.

$$F\text{-}measure = \frac{2* Precision * Recall}{Precision+ Recall} \text{ .... (4.11)}$$

Accuracy: It is the most intuitive performance measure. It can be calculated as the ratio of correctly classified reviews to total number of reviews **[64]**.

$$Accuracy = \frac{TP+TN}{TP+FP+TN+FN} \text{ .... (4.12)}$$

CHAPTER FIVE

Introduction to Deep Learning Algorithms

Deep learning (also known as deep structured learning) is a type of machine learning technology that uses artificial neural networks to learn representations. There are three types of learning: supervised, semi-supervised, and unsupervised.

Deep-learning architectures such as deep neural networks, deep belief networks, deep reinforcement learning, recurrent neural networks, and convolutional neural networks have been used in fields such as computer vision, speech recognition, natural language processing, machine translation, bioinformatics, drug design, medical image analysis, climate science, material inspection, and board game programs, producing results that are comparable to, and in some cases superior to, traditional methods.

In deep learning, the word "deep" refers to the employment of numerous layers in the network. A linear perceptron cannot be a universal classifier, but a network with a nonpolynomial activation function and one hidden layer of unlimited width can, according to early research. Deep learning is a recent variant that involves an unbounded number of layers of bounded size, allowing for practical application and optimization while maintaining theoretical universality under mild conditions **[65]**.

We have used simple dense layer and recurrent neural network for our deep learning section

5.1 Simple Dense layer

A dense layer in a neural network is one that is deeply coupled to its preceding layer, meaning its neurons are connected to every neuron in the previous layer. The most popular layer in artificial neural network networks is this one.

In a model, the dense layer's neuron receives input from every neuron in the preceding layer, and the dense layer's neurons conduct matrix-vector multiplication. The row vector of the output from the preceding layers is identical to the column vector of the dense layer in matrix vector multiplication. In matrix-vector multiplication, the row vector must have the same number of columns as the column vector.

The general formula for a matrix-vector product is:

$$A\mathbf{x} = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n \\ \vdots \\ a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n \end{bmatrix}. \qquad \text{... (5.1)}$$

Where A is a (M x N) matrix and x is a (1 ???? N) matrix. Backpropagation can update the values under the matrix, which are the trained parameters of the preceding layers. The most widely used algorithm for training feedforward neural networks is backpropagation. In a neural network, backpropagation computes the gradient of the loss function with respect to the network's weights for a single input or output. We can deduct from the preceding understanding that the thick layer's output will be an N-dimensional vector. We can see that the vectors' dimensions are shrinking. So, a dense layer is employed to change the dimension of the vectors by utilizing each neuron.

As previously stated, every neuron in the preceding layers sends its output to every neuron in the dense layer. So, if the preceding layer produces a (M x N) matrix by aggregating the results of each neuron, this output passes through the dense layer, which should have a count of N neurons **[66]**.

5.2 Recurrent Neural Network (RNN)

RNNs are a sort of Neural Network in which the output from the previous step is used as input in the next stage. Traditional neural networks have inputs and outputs that are independent of one another, however in some circumstances, such as when predicting the next word of a phrase, the prior words are necessary, and so the previous words must be remembered. As a result, RNN was created, which used a Hidden Layer to overcome the problem. The Hidden state, which remembers certain information about a sequence, is the most essential aspect of RNN.

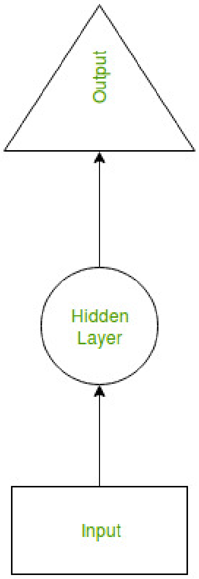


Figure 5.1: Recurrent Neural Network

[Source: https://www.geeksforgeeks.org/introduction-to-recurrent-neural-network/ Retrieved: 24th-May'2022, 2.18PM]

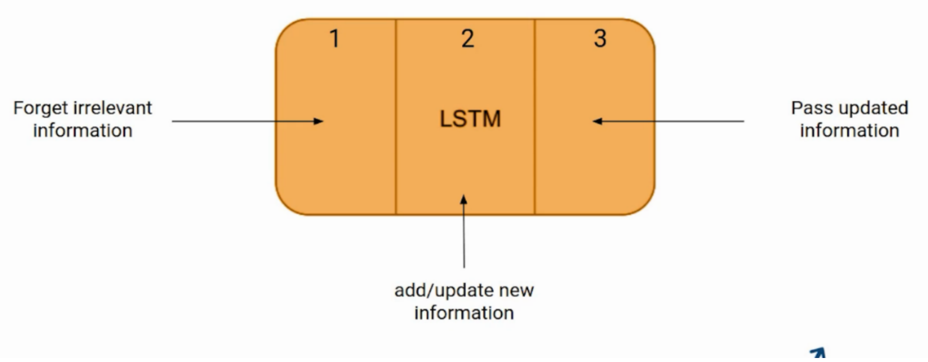
RNNs have a "memory" that stores all information about the calculations. It employs the same settings for each input since it produces the same outcome by performing the same task on all inputs or hidden layers. Unlike other neural networks, this decreases the complexity of parameters **[67]**.

5.3 Long Short-Term Memory (LSTM)

The Long Short Term Memory Network (LSTMN) is a type of sophisticated RNN (sequential network) that permits information to be retained. It can deal with the vanishing gradient problem that RNN has. For persistent memory, a recurrent neural network, also known as RNN, is used.

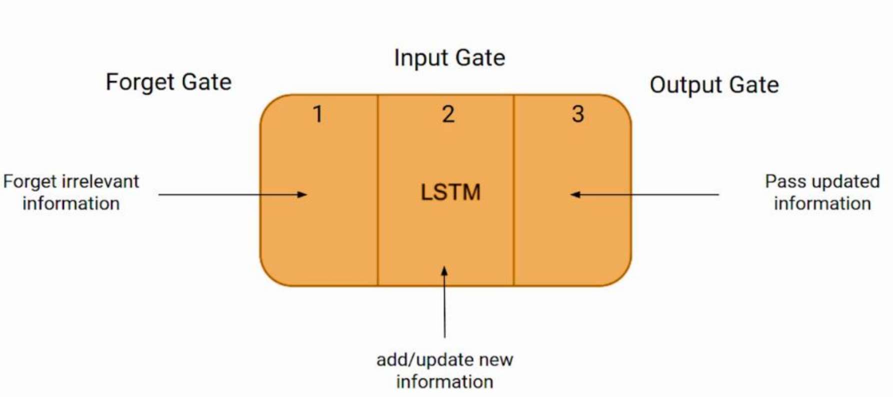
Let's assume you recall the previous scene when viewing a video or you know what happened in the prior chapter while reading a book. RNNs work in a similar way, remembering previous information and applying it to the current input. Because of the shrinking gradient, RNNs are unable to recall long-term dependencies. Long-term dependency problems are explicitly avoided with LSTMs.

At a high-level LSTM works very much like an RNN cell. Here is the internal functioning of the LSTM network. The LSTM consists of three parts, as shown in the image below and each part performs an individual function.



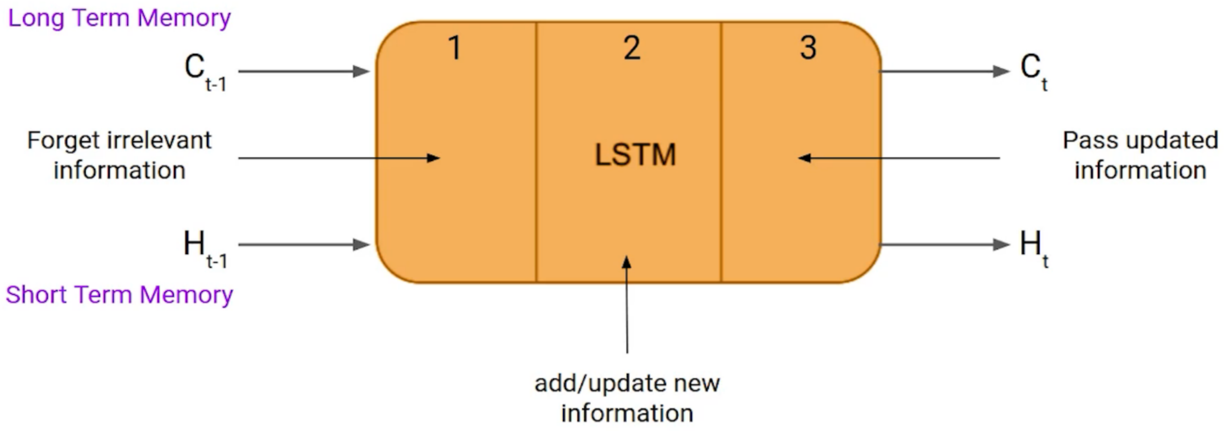
The first part chooses whether the information coming from the previous timestamp is to be remembered or is irrelevant and can be forgotten. In the second part, the cell tries to learn new information from the input to this cell. At last, in the third part, the cell passes the updated information from the current timestamp to the next timestamp.

These three parts of an LSTM cell are known as gates. The first part is called **Forget gate, the** second part is known as **the Input gate** and the last one is **the Output gate**.

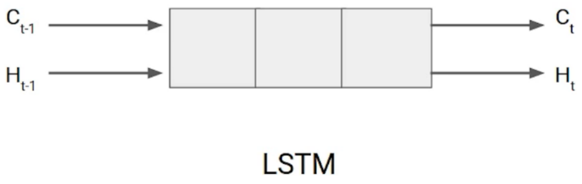


Just like a simple RNN, an LSTM also has a hidden state where H(t-1) represents the hidden state of the previous timestamp and Ht is the hidden state of the current timestamp. In addition to that LSTM also have a cell state represented by C(t-1) and C(t) for previous and current timestamp respectively.

Here the hidden state is known as Short term memory and the cell state is known as Long term memory. Refer to the following image.



It is interesting to note that the cell state carries the information along with all the timestamps.



Bob is a nice person. Dan on the other hand is evil.

Let’s take an example to understand how LSTM works. Here we have two sentences separated by a full stop. The first sentence is “Bob is a nice person” and the second sentence is

"Dan, on the Other hand, is evil". It is very clear, in the first sentence we are talking about Bob and as soon as we encounter the full stop(.) we started talking about Dan.

As we move from the first sentence to the second sentence, our network should realize that we are no more talking about Bob. Now our subject is Dan. Here, the Forget gate of the network allows it to forget about it. Let's understand the roles played by these gates in LSTM architecture **[68]**.

5.4 Bidirectional Long Short-Term Memory (BiLSTM)

A Bidirectional LSTM, also known as a BiLSTM, is a sequence processing model that consists of two LSTMs, one of which takes input in one direction and the other in the other. BiLSTMs effectively increase the quantity of data available to the network, providing the algorithm with more context **[69]**.

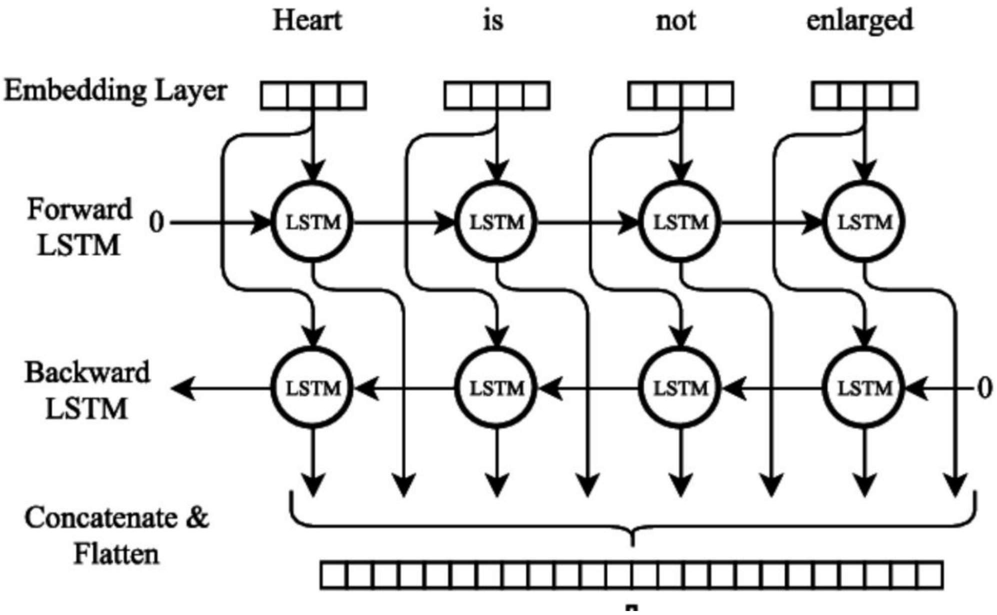


Figure 5.2: BiLSTM

[Source: https://paperswithcode.com/method/bilstm#:~:text=A%20Bidirectional%20LSTM%2C%20or%20biLSTM,other%20in%20a%20backwards%20direction. Retrieved: 25th May, 2022, 1.25 PM]

CHAPTER SIX

Balanced vs Imbalanced Dataset

The definition of balanced dataset is the same approximation of data in each class of the dataset. For example, if there are two classes in a dataset, the balance dataset would have approximately 50% data for each of the class. Balanced dataset provides optimum output of the model.

Imbalance dataset means the distribution of data points across the classes are uneven, biased, or skewed. Imbalanced data is quite common in real-life as datasets are always in some sort of imbalanced form. The problem arises when data in one section class dominates other section of class data. Often time, it causes machine learning models to be more biased towards domination section of class. Therefore, the accuracy measurement of models using the imbalanced dataset may not be effective as it hardly classifies any data from minority classes. The balanced and imbalanced datasets are shown below in Figure 6.1.

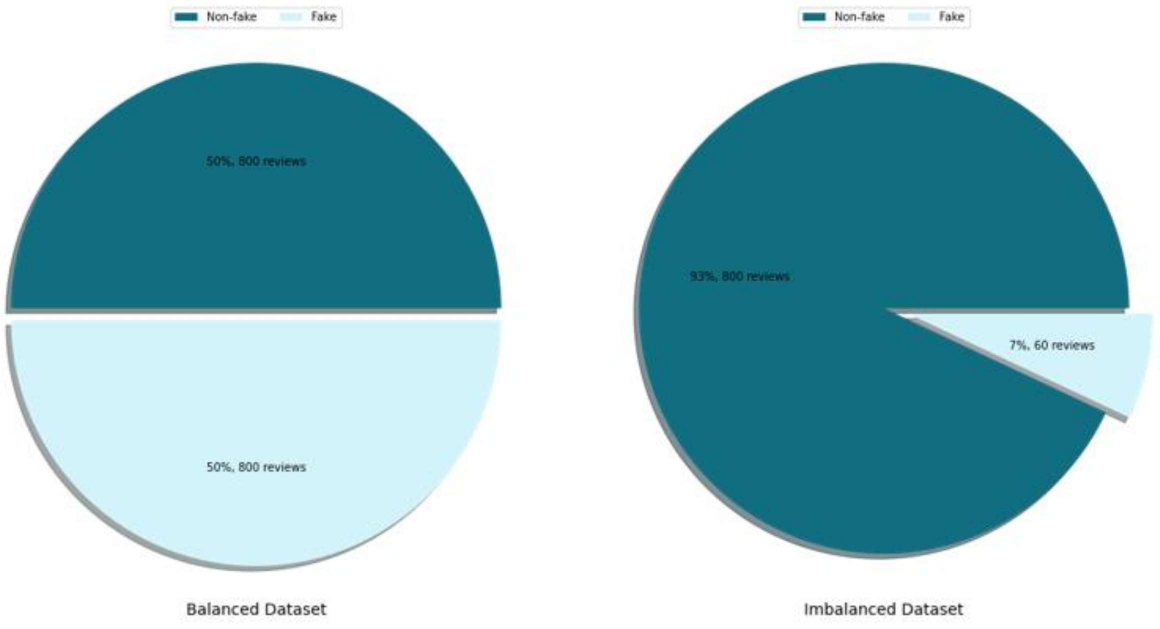
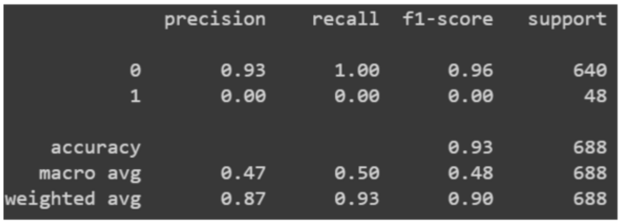


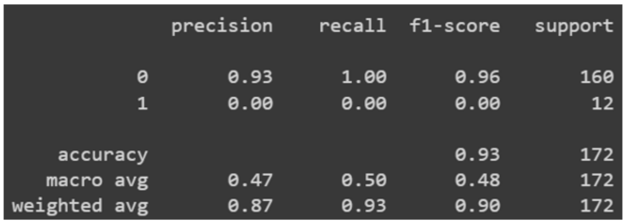
Figure 6.1: Balanced Vs Imbalanced Dataset

Here, we apply Multinomial Naive Bayes classifier on both datasets. For balanced dataset, we have achieved test accuracy of 84.69% against train accuracy of 96.17%. For imbalanced dataset, we have achieved test accuracy of 93.02% against train accuracy of 93.02%. If we compare both dataset, imbalanced dataset looks ideal as it proves better test accuracy than balanced dataset. But the accuracy measurement of imbalanced dataset is not effective. To understand this, we need to look at the confusion matrix of both datasets.



```
              precision    recall  f1-score   support

           0       0.93      1.00      0.96       640
           1       0.00      0.00      0.00        48

    accuracy                           0.93       688
   macro avg       0.47      0.50      0.48       688
weighted avg       0.87      0.93      0.90       688
```

Confusion Matrix of Training Data



```
              precision    recall  f1-score   support

           0       0.93      1.00      0.96       160
           1       0.00      0.00      0.00        12

    accuracy                           0.93       172
   macro avg       0.47      0.50      0.48       172
weighted avg       0.87      0.93      0.90       172
```

Confusion Matrix of Testing Data

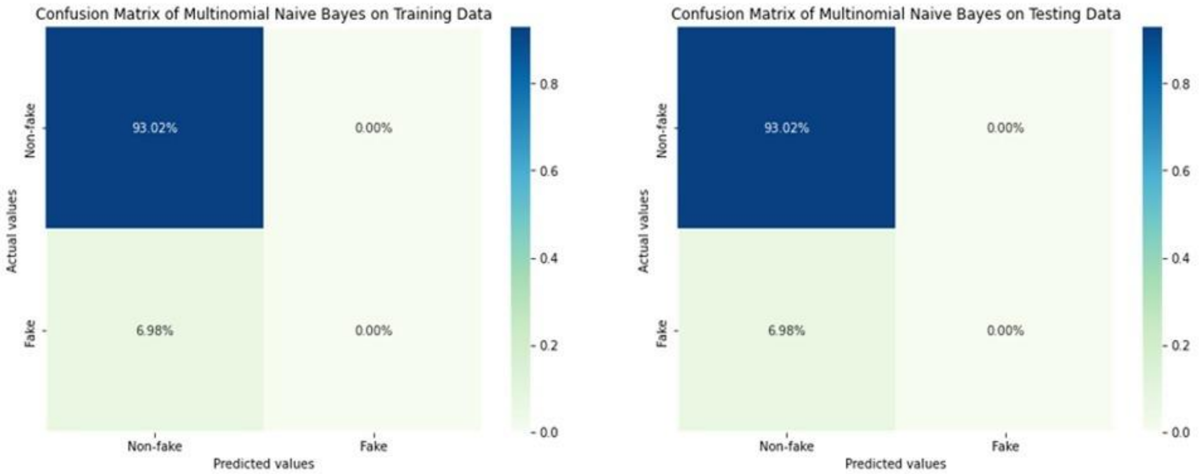
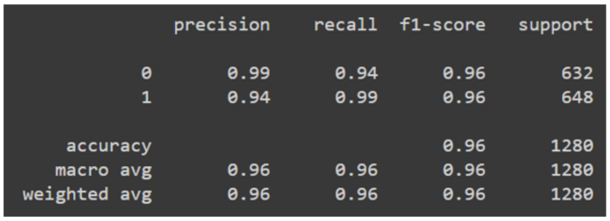


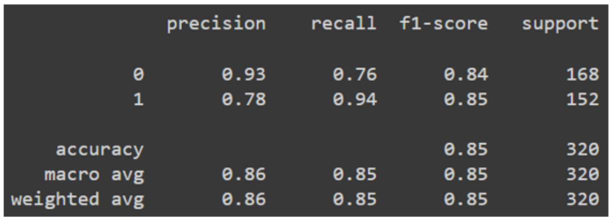
Figure 6.2: Confusion Matrix of Imbalanced Dataset

In figure 6.2, we notice that the training data of non-fake(1) and fake(0) are 640 and 48 reviews, in which the Multinomial Naive Bayes classifier model has learned all the non-fake reviews as precision, recall and f-1 score are quite high. But it has failed to learn a single fake review out of 48 reviews. We can see that in the figure 6.2 as precision, recall and f-1 score are 0. As a result, in testing data, we can see that the model has failed to predict a single fake review out of 12 reviews. The model only manages to predict the non-fake reviews and based on that it has given a test accuracy of 93% which is not accurate.



```
              precision    recall  f1-score   support

           0       0.99      0.94      0.96       632
           1       0.94      0.99      0.96       648

    accuracy                           0.96      1280
   macro avg       0.96      0.96      0.96      1280
weighted avg       0.96      0.96      0.96      1280
```

Confusion Matrix of Training Data



```
              precision    recall  f1-score   support

           0       0.93      0.76      0.84       168
           1       0.78      0.94      0.85       152

    accuracy                           0.85       320
   macro avg       0.86      0.85      0.85       320
weighted avg       0.86      0.85      0.85       320
```

Confusion Matrix of Testing Data

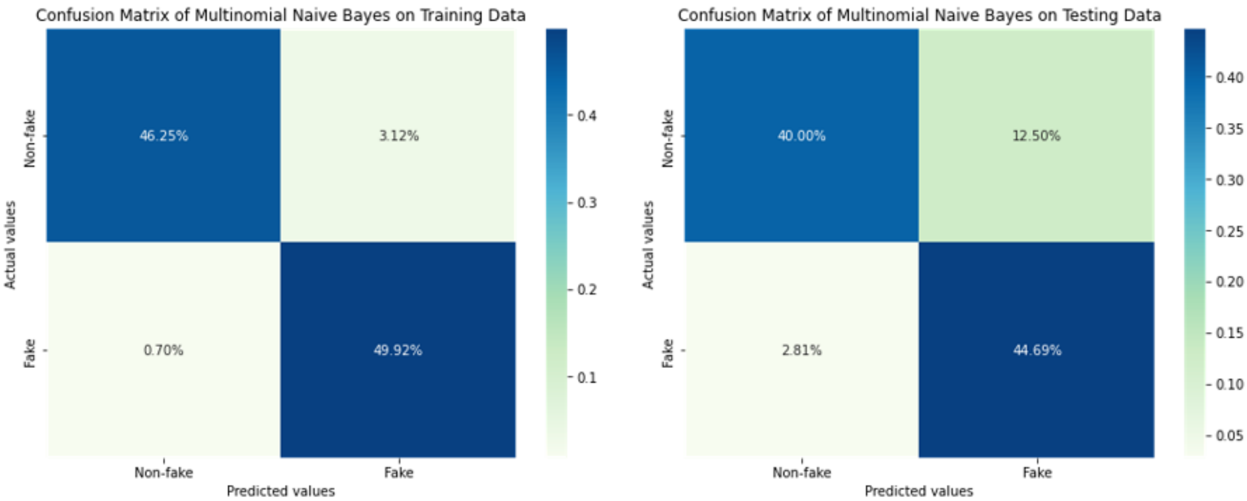


Figure 6.3: Confusion Matrix of Balanced Dataset

In figure 6.3, we notice that the Multinomial Naive Bayes classifier model has learned and predicted both training and test data with high precision, recall and f-1 score in both non-fake and reviews. The test accuracy of this balanced dataset is more accurate and precise than imbalanced dataset. In this thesis paper, we have chosen balanced dataset to have precise accuracy for our research work.

CHAPTER SEVEN

Research Methodology

The aim of the paper is to figure out the fake and non-fake reviews. For this, we will use two datasets. One is "gold-standard" Deceptive Opinion Spam Hotel Dataset developed by (Ott et al. 2011) **[21]** and another is "The Yelp Restaurant Review Dataset" from (Mukherjee et al. 2013) **[23].**

First, we will discuss about Deceptive Opinion Spam Hotel dataset developed by (Ott et al. 2011) **[21].** The dataset consists of 1600 hotel reviews from 20 Chicago hotels, in which 800 reviews are fake and 800 reviews are non-fake which is shown in Figure 7.1.

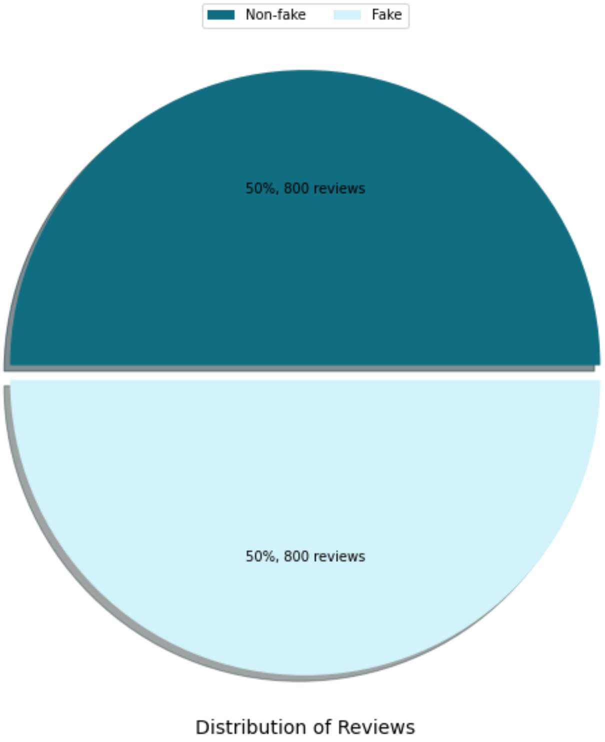


Figure 7.1: Hotel Dataset

The non-fake hotel reviews are labelled as '0' and the fake hotel reviews are labelled as '1'. In this dataset, for non-fake reviews, 400 are written with a negative sentiment polarity and another 400 are written with a positive sentiment polarity. Similarly, dataset includes same format for fake reviews. The dataset has no missing values. In Figure 7.2, the deceptive or non-fake reviews were generated from Amazon Mechanical Turk (AMT). ATM is crowdsourcing website that hires people to perform discontinuous on-demand tasks that computers are currently unable to do. The rest of the reviews are taken from various online web sources like Expedia, Hotels.com, TripAdvisor, Yelp etc.

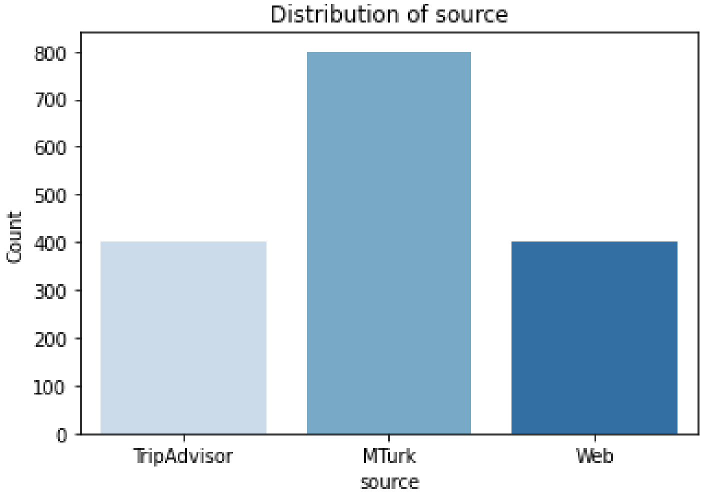


Figure 7.2: Distribution of reviews of hotel dataset

The dataset consists of 80 reviews for each of the 20 most popular Chicago hotels which is shown in Figure 7.3:

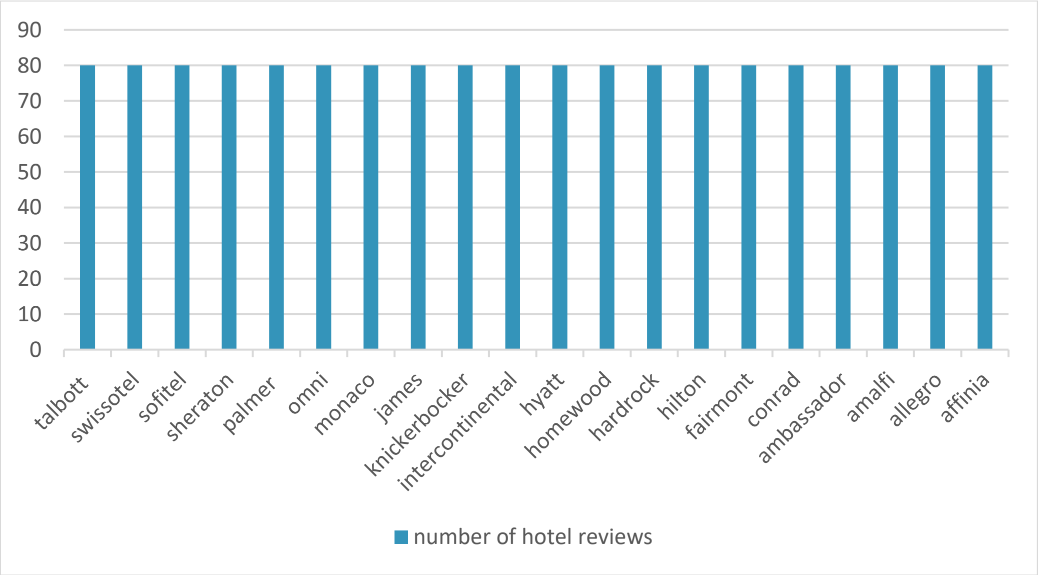


Figure 7.3: Distribution of hotels

Now we will use world cloud, a data visualization technique in our dataset **[21]**, that helps to represent a text data in which the size of individual words indicates its frequency or importance. We will show three stages of text data: overall reviews, non-fake reviews, and fake reviews.

Figure 7.4 represents the distribution of most used words in overall dataset:

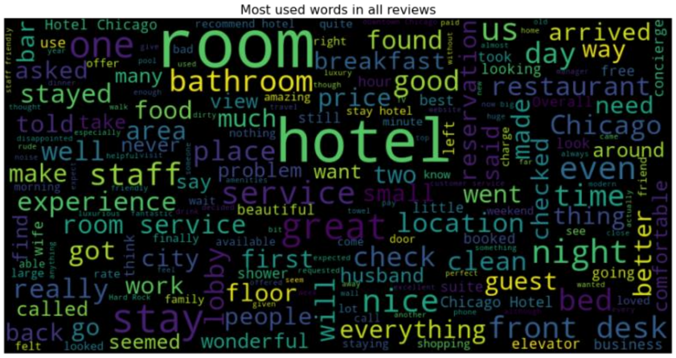


Figure 7.4: Distribution of most used words

Figure 7.5 represents the distribution of most used words in the non-fake reviews:

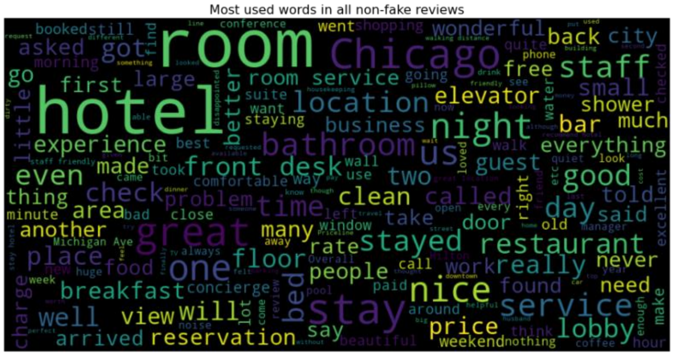


Figure 7.5: Distribution of most used words in the non-fake reviews

Figure 7.6 represents the distribution of most used words in the fake reviews:

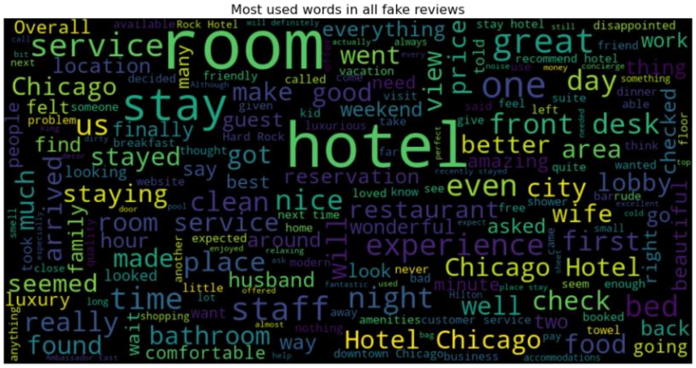


Figure 7.6: Distribution of most used words in the fake reviews

Now the second dataset that we use in this paper is “The Yelp Restaurant Reviews Dataset” from the paper “What Yelp Fake Review Filter Might Be Doing” **[23]**. Yelp is a popular online website that helps users locate local businesses and provides reliable information through ratings and reviews. The dataset consists of 16878 real-life customer reviews from Yelp in reviewContent column. Reviews obtained from the restaurant page are labelled and wherein we get all Y reviews from the filtered section(fake) and N reviews from the regular page(non-fake) in flagged column. The dataset has no missing values. The dataset includes information such as date, reviewID, reviewerID, rating, restaurantID, and more. Each review in the dataset **[23]** consists of 9 parts:

<date> <reviewID> <reviewerID> <rating> <usefulCount> <coolCount> <funnyCount> <flagged> <restaurantID>

Within those reviews, 8577 reviews are labelled as non-fake (N) and 8301 reviews are labelled as fake (Y). The pie chat displays non-fake reviews as 50.82% and fake reviews as 49.18%. In Figure 7.7 the Review data is color-categorized by non-fake (dark cyan) and fake (light grayish cyan).

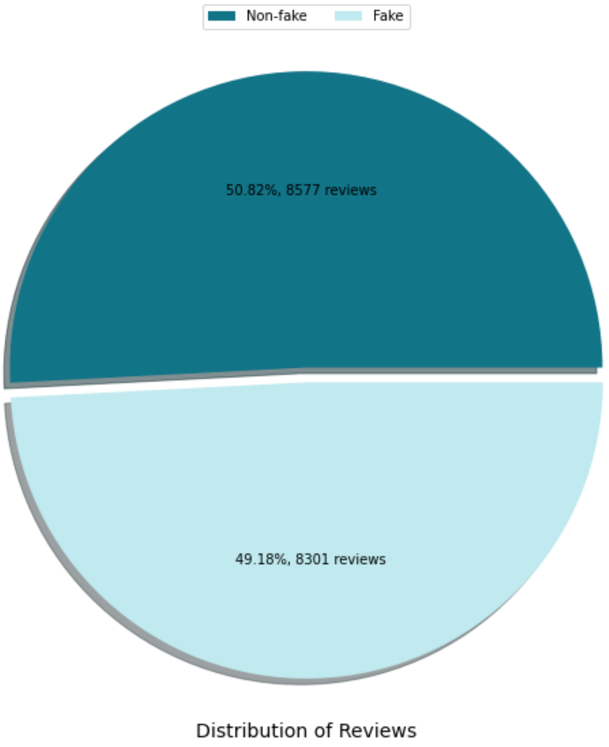


Figure 7.7: Restaurant Dataset

The reviews also include ratings as labelled in the dataset **[23]**. There are 8443 reviews that are labelled as 5 rating. 4836 reviews are labelled as 4 ratings. Figure 7.8 represents the rating distribution:

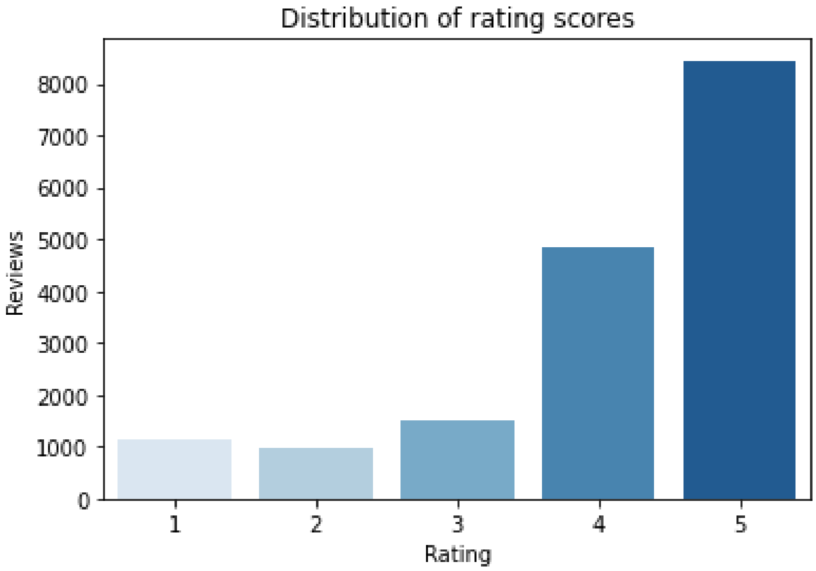


Figure 7.8: Rating distribution

The dataset **[23]** contains useful counts, that means how many users find the review useful. For example, there are 1969 review texts that are liked once as useful. 956 review texts that are liked twice as useful. There are 27 review texts that got 10 likes as useful by users and 1 review text got 84 likes as useful by users. Figure 7.9 represents the useful count distribution:

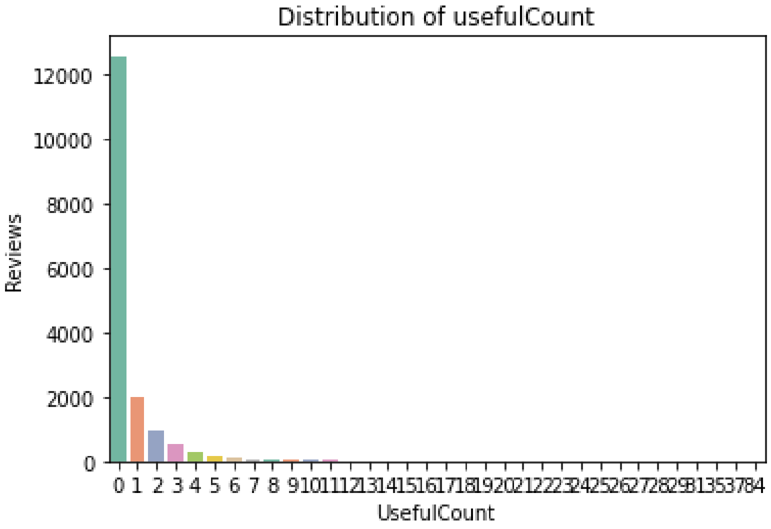


Figure 7.9: Useful count distribution

The dataset **[23]** contains restaurantID, that means how many users have reviewed a particular restaurant. For example, a restaurant labelled as “HOJqzz1WvOmeR9oESJ4d9A” got 2302 reviews from customers which is the highest. In this dataset, the restaurant names are labelled as random text. Figure 7.10 represents the restaurantID distribution:

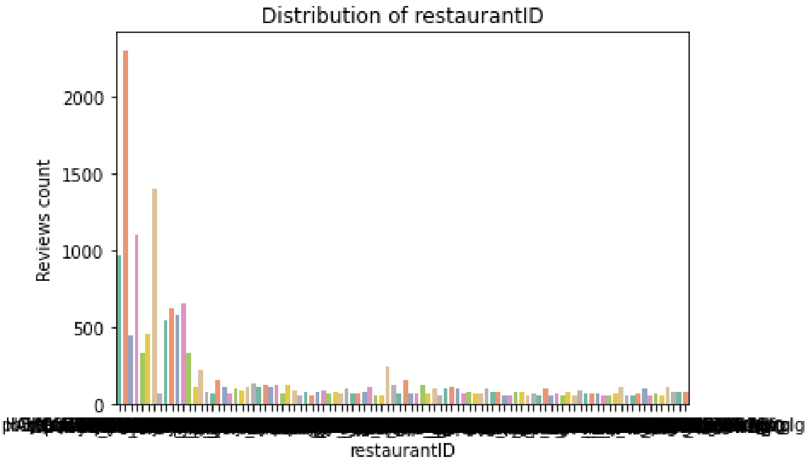


Figure 7.10: RestaurantID distribution

Now we will use the world cloud same as previously to visualize the review text data from our dataset **[23].**

Figure 7.11 represents the distribution of most used words in overall dataset:

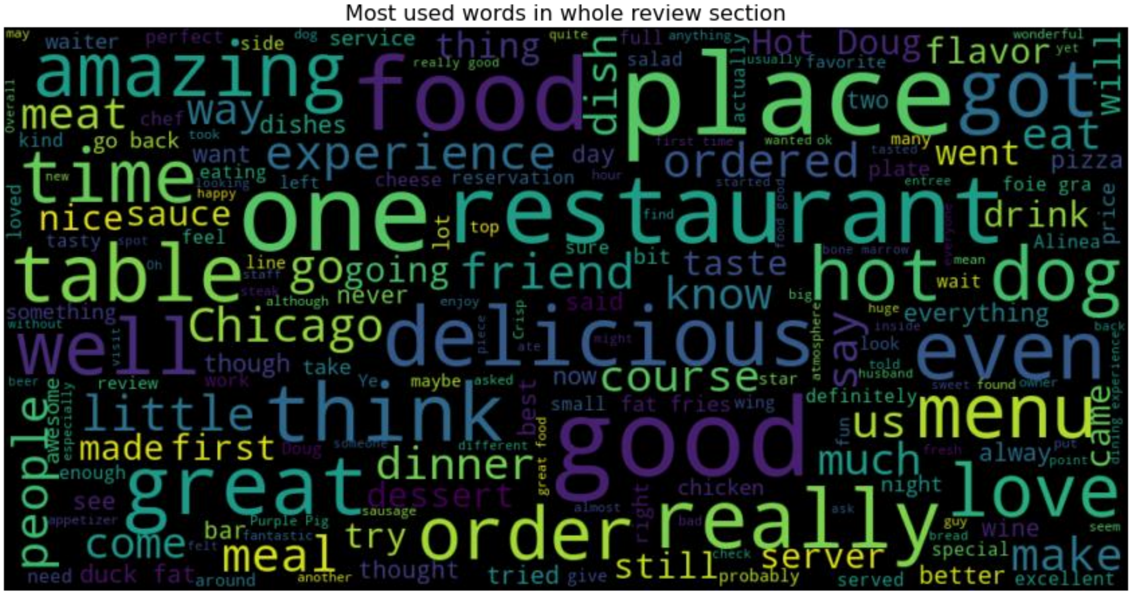


Figure 7.11: Most used words in overall dataset

Figure 7.12 represents the distribution of most used words in the non-fake reviews:

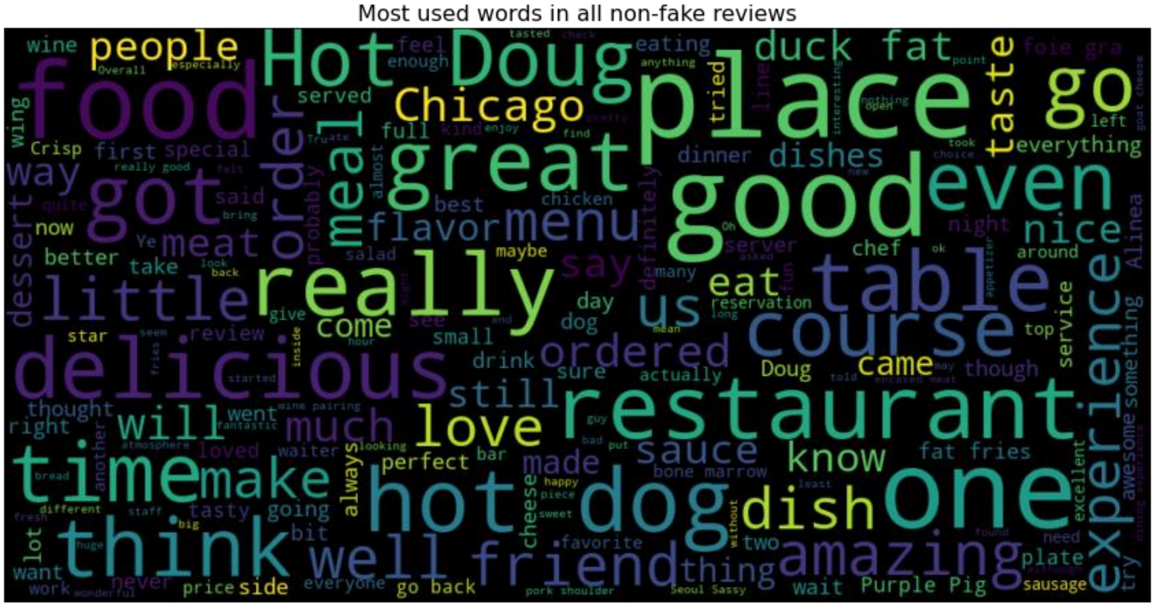


Figure 7.12: Most used words in the non-fake reviews

Figure 7.13 represents the distribution of most used words in the fake reviews:

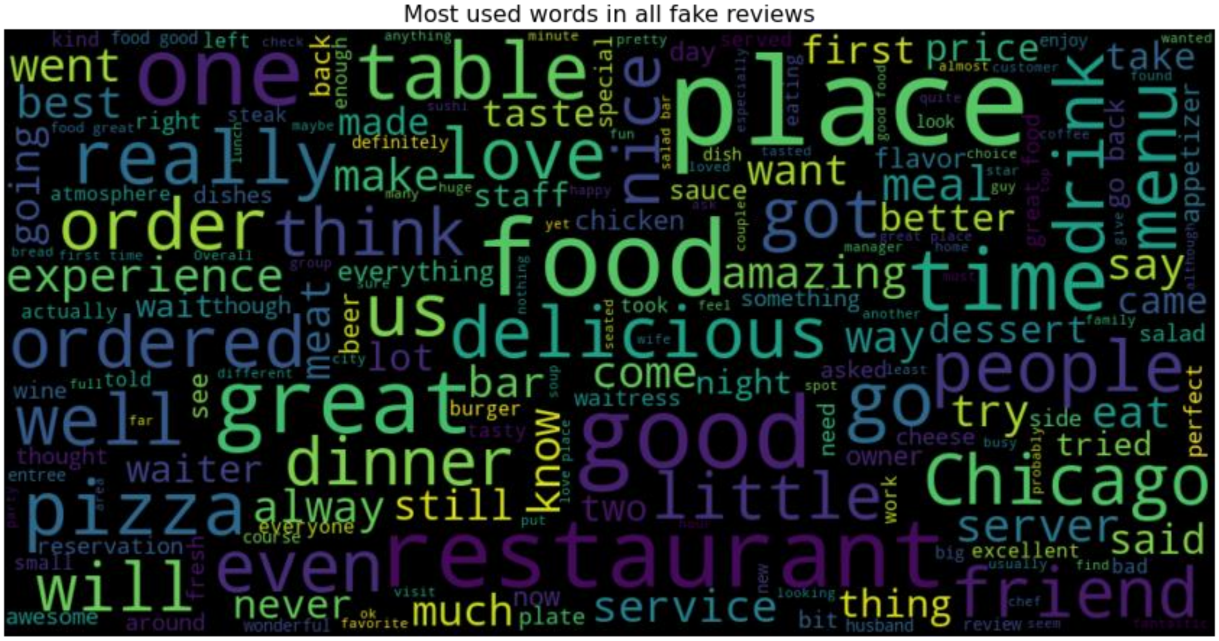


Figure 7.13: Most used words in the fake reviews

Figure 7.14 represents the proposed Model for Fake Review Detection

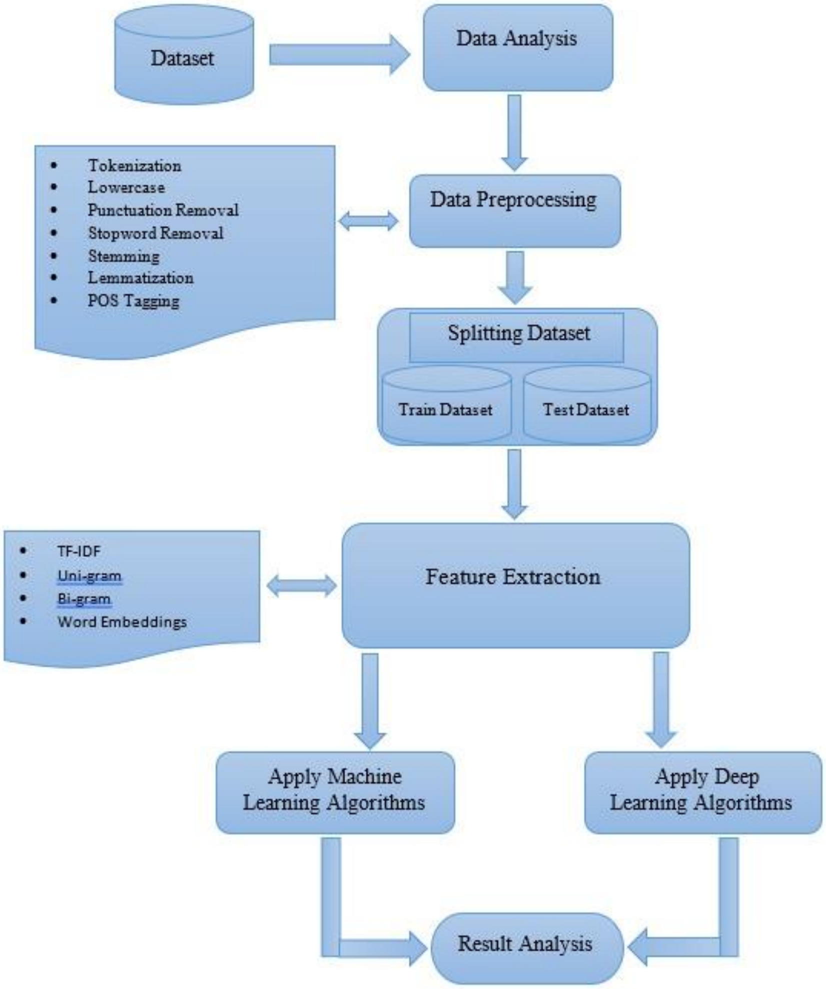


Figure 7.14: Proposed Model for Fake Review Detection

7.1 Data Pre-Processing

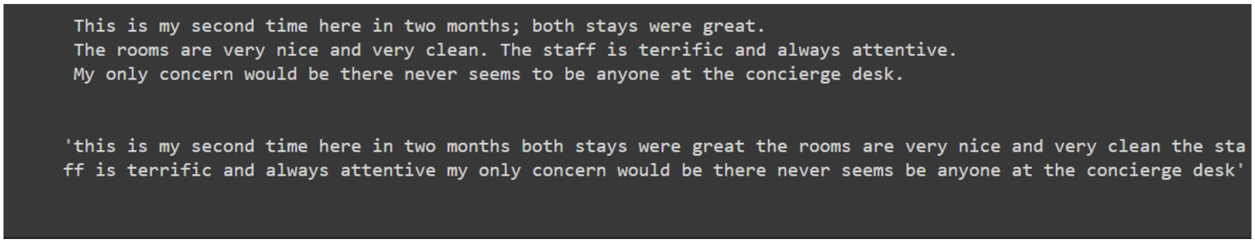
As we are working with two datasets **[21]**, **[23]**, the data pre-processing part will same for both the dataset. We will keep the review text column and flagged column and drop all other columns.

7.1.1 Clean Text

The first step of data pre-processing is to transform the raw review text data into cleaned text data. Here, we will,

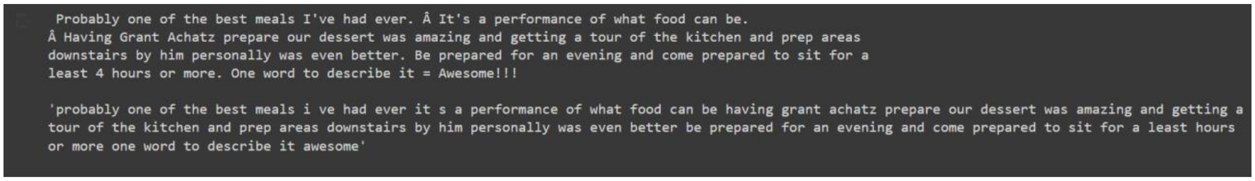
- Tokenize all the review texts
- Lowercase all the review texts
- Remove any characters that is not in a-z OR A-Z alphabet, that includes generic text, numbers, symbols etc.
- Remove whitespaces
- Remove tabs

After cleaning the raw texts of both Deceptive Opinion Spam Hotel Dataset **[21]** and The Yelp Restaurant Review Dataset **[23]**, the result will look like Figure 7.15 & Figure 7.16.



```
 This is my second time here in two months; both stays were great.
 The rooms are very nice and very clean. The staff is terrific and always attentive.
 My only concern would be there never seems to be anyone at the concierge desk.


'this is my second time here in two months both stays were great the rooms are very nice and very clean the sta
ff is terrific and always attentive my only concern would be there never seems be anyone at the concierge desk'
```

Figure 7.15: Deceptive Opinion Spam Hotel Dataset

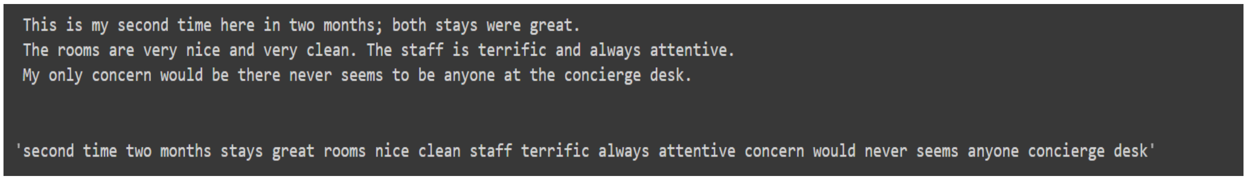


```
 Probably one of the best meals I've had ever. Â It's a performance of what food can be.
Â Having Grant Achatz prepare our dessert was amazing and getting a tour of the kitchen and prep areas
downstairs by him personally was even better. Be prepared for an evening and come prepared to sit for a
least 4 hours or more. One word to describe it = Awesome!!!

'probably one of the best meals i ve had ever it s a performance of what food can be having grant achatz prepare our dessert was amazing and getting a
tour of the kitchen and prep areas downstairs by him personally was even better be prepared for an evening and come prepared to sit for a least hours
or more one word to describe it awesome'
```

Figure 7.16: The Yelp Restaurant Review Dataset

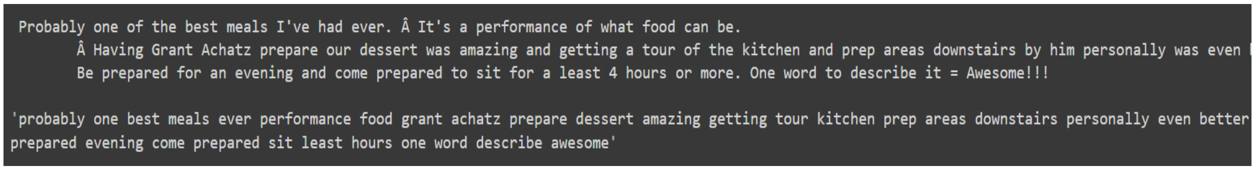
7.1.2 Remove Stopwords

Stopwords are a set of words that are too frequent in a language that don't change the overall meaning of the sentence. By using nltk.download('stopwords') command we download stopwords from NLTK library and remove these stopwords from our review texts which is shown in Figure 7.17 & Figure 7.18.



```
 This is my second time here in two months; both stays were great.
 The rooms are very nice and very clean. The staff is terrific and always attentive.
 My only concern would be there never seems to be anyone at the concierge desk.


'second time two months stays great rooms nice clean staff terrific always attentive concern would never seems anyone concierge desk'
```

Figure 7.17: Deceptive Opinion Spam Hotel Dataset

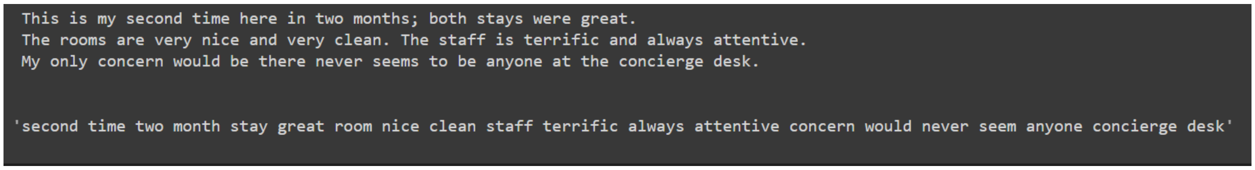


```
 Probably one of the best meals I've had ever. Â It's a performance of what food can be.
        Â Having Grant Achatz prepare our dessert was amazing and getting a tour of the kitchen and prep areas downstairs by him personally was even
        Be prepared for an evening and come prepared to sit for a least 4 hours or more. One word to describe it = Awesome!!!

'probably one best meals ever performance food grant achatz prepare dessert amazing getting tour kitchen prep areas downstairs personally even better
prepared evening come prepared sit least hours one word describe awesome'
```

Figure 7.18: The Yelp Restaurant Review Dataset

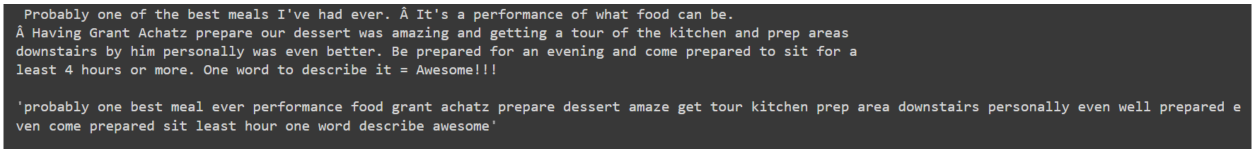
7.1.3 Lemmatization

Both goal of both stemming and lemmatization is to reduce inflectional forms and sometimes derivationally related forms of a word to a common base form **[70]**. For our datasets we use lemmatization as it removes inflectional endings properly and to return the base or dictionary form of a word with the use of a vocabulary and morphological analysis of words which is shown in Figure 7.19 & Figure 7.20.



```
 This is my second time here in two months; both stays were great.
 The rooms are very nice and very clean. The staff is terrific and always attentive.
 My only concern would be there never seems to be anyone at the concierge desk.


'second time two month stay great room nice clean staff terrific always attentive concern would never seem anyone concierge desk'
```

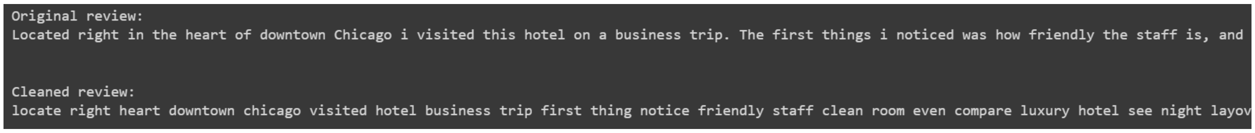
Figure 7.19: Deceptive Opinion Spam Hotel Dataset



```
 Probably one of the best meals I've had ever. Â It's a performance of what food can be.
Â Having Grant Achatz prepare our dessert was amazing and getting a tour of the kitchen and prep areas
downstairs by him personally was even better. Be prepared for an evening and come prepared to sit for a
least 4 hours or more. One word to describe it = Awesome!!!

'probably one best meal ever performance food grant achatz prepare dessert amaze get tour kitchen prep area downstairs personally even well prepared e
ven come prepared sit least hour one word describe awesome'
```

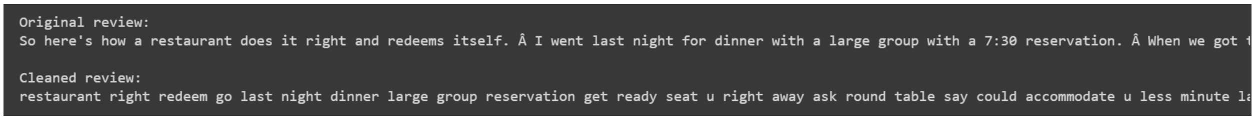
Figure 7.20: The Yelp Restaurant Review Dataset

After completing the whole data pre-processing functions, the text reviews will look like Figure 7.21 & Figure 7.22.



```
Original review:
Located right in the heart of downtown Chicago i visited this hotel on a business trip. The first things i noticed was how friendly the staff is, and


Cleaned review:
locate right heart downtown chicago visited hotel business trip first thing notice friendly staff clean room even compare luxury hotel see night layov
```

Figure 7.21: Deceptive Opinion Spam Hotel Dataset

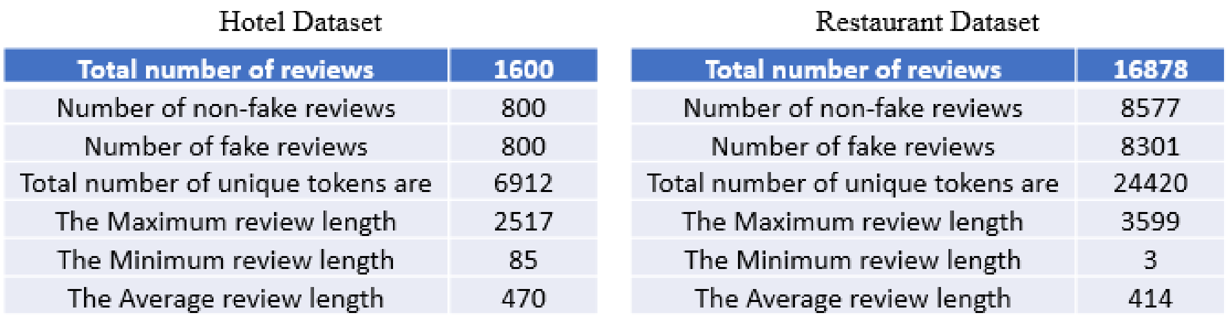


```
 Original review:
 So here's how a restaurant does it right and redeems itself. Â I went last night for dinner with a large group with a 7:30 reservation. Â When we got t

 Cleaned review:
 restaurant right redeem go last night dinner large group reservation get ready seat u right away ask round table say could accommodate u less minute la
```

Figure 7.22: The Yelp Restaurant Review Dataset

Here is the summery of the datasets **[21]**, **[23]** after data pre-processing:

Table 7.1 Summery of the datasets



Hotel Dataset

Total number of reviews	1600
Number of non-fake reviews	800
Number of fake reviews	800
Total number of unique tokens are	6912
The Maximum review length	2517
The Minimum review length	85
The Average review length	470

Restaurant Dataset

Total number of reviews	16878
Number of non-fake reviews	8577
Number of fake reviews	8301
Total number of unique tokens are	24420
The Maximum review length	3599
The Minimum review length	3
The Average review length	414

7.2 Feature Extraction

Natural language processing (NLP) is a branch of artificial intelligence which concerned with giving computers the ability to understand human languages in text or voice form by transforming it into numerical format vectors. One of the popular texts vectorizing technique is Word Embedding as it doesn't create a spare matrix of vectorized sentences as a result computation cost is low and it retain most linguistic information present in the sentence. Here we will discuss about two popular frequency-based Word Embedding forms:

- Bag of Words
- TF-IDF Model

7.2.1 Bag of Words

Bag of Words (BoW) is an algorithm, which is a way of extracting features from text or document and calculating the number of times words appear in the text or document without following any grammatical rule or words order. It is a matrix of tokens in which if the word appears in the text or document, it will give one value to that word otherwise it will give zero. For example, here are two sentences:

- everyone is playing football
- i am not playing football

Table 7.2 Bag of Words

Bag of Words							
Words	everyone	is	playing	football	i	am	not
Sentence 1	1	1	1	1	0	0	0
Sentence 2	0	0	1	1	1	1	1

Now, in the table, using Bag of Words algorithm, we can see all the words that appear in both sentences labelled as one and words that does not appear in both sentences labelled as zero.

7.2.2 TF-IDF

TF-IDF stands for Term Frequency - Inverse Document Frequency, is a numerical statistic that reflects how important a word is to a document in a collection or corpus. It measures relevance of the word rather than the frequency of the word.

Term Frequency, TF = $\frac{Number\ of\ occurnce\ of\ word\ in\ a\ document}{Total\ number\ of\ words\ in\ a\ document}$ ………………………………(7.1)

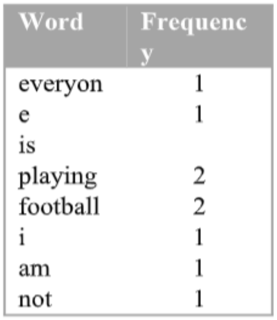
Inverse Document Frequency, IDF = log $(\frac{Total\ number\ of\ documents}{Number\ of\ documents\ containing\ the\ word})$ ……..……(7.2)

TF-IDF of a word is calculated by the multiplication of Term Frequency score and Inverse Document Frequency score.

For example, here are two sentences:

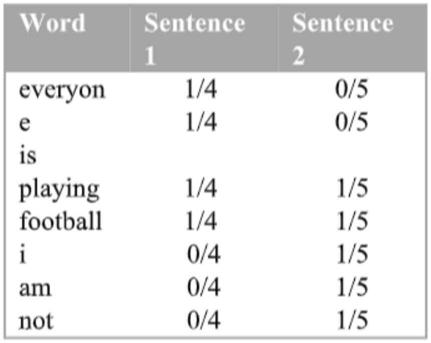
- everyone is playing football
- i am not playing football

Table 7.3 Frequency list of words



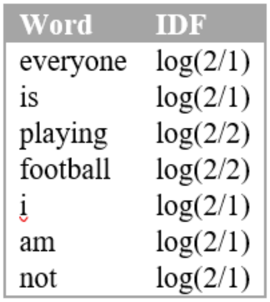
Word	Frequenc y
everyon	1
e	1
is	
playing	2
football	2
i	1
am	1
not	1

Table 7.4 Term Frequency



Word	Sentence 1	Sentence 2
everyone	1/4	0/5
is	1/4	0/5
playing	1/4	1/5
football	1/4	1/5
i	0/4	1/5
am	0/4	1/5
not	0/4	1/5

Table 7.5 IDF



Word	IDF
everyone	log(2/1)
is	log(2/1)
playing	log(2/2)
football	log(2/2)
i	log(2/1)
am	log(2/1)
not	log(2/1)

Table 7.6: TF-IDF values

TF-IDF							
Words	everyone	is	playing	football	i	am	not
Sentence 1	1/4 * log(2/1) = 0.075	1/4 * log(2/1) = 0.075	1/4 * log(2/2) = 0.075	1/4 * log(2/2) = 0.075	0/4 * log(2/1) = 0	0/4 * log(2/1) = 0	0/4 * log(2/1) = 0
Sentence 2	0/5 * log(2/1) = 0	0/5 * log(2/1) = 0	1/5 * log(2/2) = 0	1/5 * log(2/2) = 0	1/5 * log(2/1) = 0	1/5 * log(2/1) = 0	1/5 * log(2/1) = 0

If we compare TF-IDF model with Bag of Words we can see, Bag of Words contains only zeros & ones. It gives all words have the same importance and doesn't preserve any semantic information. On the other hand, TF-IDF values a word based on its importance in the whole document or corpus. For feature extraction we have applied TF-IDF in our research work.

7.3 Splitting Training and Testing data

For the evaluations of our datasets, we divide it into two sets of data: training set and testing set. For training the machine learning model, we use 80% of the comprises data and for testing, we use 20% of the comprises data.

So, The Deceptive Opinion Spam dataset consists of 1280 review data for training and 320 review data for testing.

The Yelp Restaurant Review Dataset consists of 13502 review data for training and 3376 review data for testing.

CHAPTER EIGHT

Result and Analysis

For our research work, we have chosen two datasets containing reviews which are labelled as fake and non-fake. In this paper, we study the correlation between the real-life reviews and the flagged of the reviews given as fake or non-fake. In one dataset, we have 1600 hotel reviews in which 50% are labelled as non-fake and 50% are fake. In another dataset, we have 16878 real-life customer reviews on restaurant from Yelp in which 50.82% are labelled as non-fake and 49.18% are fake.

In this analysis, we have implemented both machine learning and deep learning algorithms. The machine learning algorithm classifiers are Multinomial Naive Bayes, Bernoulli Naive Bayes, Random Forrest, k-Nearest Neighbors, Logistic Regression, Support Vector Machine (Linear) and Decision Tree to detect fake reviews using supervised learning. For feature extraction we applied TF-IDF as it gives best results for both our dataset. For deep learning algorithms, we implemented RNN-Long Short-Term Memory (LSTM) to detect fake reviews using supervised learning. Our Obtained experiment Results by Using Machine Learning Classifiers are given below:

Table 8.1: Hotel Dataset

ML Classifier	Feature Matrix	Train Accuracy (%)	Test Accuracy (%)	Train Precision	Test Precision	Train Recall	Test Recall	Train F1-score	Test F1-score
Multinomial Naive Bayes	**POS + Uni-gram**	**96.17%**	**84.69%**	**0.96**	**0.86**	**0.96**	**0.85**	**0.96**	**0.85**
	POS + Bi-gram	**100%**	**74.06%**	**1.00**	**0.80**	**1.00**	**0.74**	**1.00**	**0.73**
Bernoulli Naive Bayes	**POS + Uni-gram**	**95.78%**	**82.81%**	**0.96**	**0.85**	**0.96**	**0.83**	**0.96**	**0.83**
	POS + Bi-gram	**99.92%**	**65.63%**	**1.00**	**0.78**	**1.00**	**0.66**	**1.00**	**0.62**
Random Forrest	**POS + Uni-gram**	**100%**	**86.25%**	**1.00**	**0.86**	**1.00**	**0.86**	**1.00**	**0.86**
	POS + Bi-gram	**100%**	**78.44%**	**1.00**	**0.78**	**1.00**	**0.78**	**1.00**	**0.78**

Logistic Regression	**POS + Uni-gram**	**97.34%**	**86.56%**	**0.97**	**0.87**	**0.97**	**0.87**	**0.97**	**0.87**
	POS + Bi-gram	**100%**	**85.00%**	**1.00**	**0.85**	**1.00**	**0.85**	**1.00**	**0.85**
SVM	**POS + Uni-gram**	**98.75%**	**88.45%**	**0.99**	**0.89**	**0.99**	**0.88**	**0.99**	**0.88**
	POS + Bi-gram	**100%**	**84.06%**	**1.00**	**0.85**	**1.00**	**0.84**	**1.00**	**0.84**
K-Nearest Neighbors	**POS + Uni-gram**	**84.85%**	**80.00%**	**0.85**	**0.81**	**0.85**	**0.80**	**0.85**	**0.80**
	POS + Bi-gram	**83.20%**	**76.56%**	**0.84**	**0.78**	**0.83**	**0.77**	**0.83**	**0.76**
Decision Tree	**POS + Uni-gram**	**100%**	**70.31%**	**1.00**	**0.71**	**1.00**	**0.70**	**1.00**	**0.70**
	POS + Bi-gram	**100%**	**68.44%**	**1.00**	**0.69**	**1.00**	**0.68**	**1.00**	**0.68**

Here, Deceptive Opinion Spam Hotel Dataset **[21],** we applied data pre-processing by as applying tokenization, lowercase, stopwords remove, stemming and lemmatization. For feature extraction we applied TF-IDF. We have applied the machine learning algorithms using both unigram and bigram feature. Unigram is a set of continuous words from a given text in which the occurrence of each word is independent of its previous word. In bigram, the occurrence of each word is dependent on its previous word. For example, the sentence “everyone is playing football”:

In unigram: “everyone”, “is”, “playing”, “football”.

In bigram: “everyone is”, “is playing”, “playing football”

By applying the data pre-processing model, we have achieved test accuracy rate of 84.69% with Multinomial Naïve bayes classifier using unigram feature. The training accuracy is 96.17%. The overall precision, recall and F-1 score are also quite high in both training and testing dataset. As the frequency of words increases due to bigram implementation, the train accuracy improves as 100% but we can see a drop of test accuracy which is 74.06%. In bigram implementation, the test f-1 score is also decreases as 0.73 compare with unigram Multinomial Naïve bayes classifier which is 0.85.

In Bernoulli Naïve Bayes classifier using unigram feature, we have achieved the test accuracy of 85.31% against training accuracy of 96.56%. In bigram, the increase in the frequency of words results this significant improvement in train accuracy of 100% but a large drop in test accuracy which is 65.63%.

In Random Forrest classifier using unigram feature, the test accuracy is 86.25% against train accuracy of 100%. Here, we set the number of trees in the forest (n_estimators)=200, the maximum depth of the tree (max_depth)=0.8, the number of jobs to run in parallel (n_jobs) = -1 which means using all processors. The train precision, recall and f-1 score are 1.00 against test precision, recall and f-1 score of 0.86. In bigram implementation, the test accuracy is 78.44% against train accuracy of 100%. And test precision, recall and f-1 score are 0.78.

In Logistic Regression, in both unigram and bigram feature, the test accuracy is 86.56% and 85.00% against train accuracy of 98.75% and 100% respectively. The reason Logistic Regression has a quite high test accuracy because it is a binary classification and will work best on binary labels. The precision, recall and f-1 score are also quite high in Logistic Regression.

In k-Nearest Neighbors using unigram feature, we achieve test accuracy of 80.00% against train accuracy of 84.85%. We change n_neighbors = 10, which is the number of neighbors to use to achieve this test accuracy. In bigram, we see a decrease in test accuracy which is 76.56% and precision, recall and f-1 score also decrease.

The Decision Tree classifier shows quite poor test accuracy in both unigram and bigram features. The precision, recall and f-1 similar.

In Support Vector Machine (linear) classifier using unigram feature, we have achieved the highest test accuracy of 88.45% against train accuracy of 98.75%. The test precision, recall and f-1 score are also the highest that we have achieved so far which is 0.88 and this indicates how well the classifier has managed to classify each level. In bigram implementation, we can see a drop of test accuracy which is 84.06%. The test precision, recall and f-1 score are still quite high.

Table 8.2: Restaurant Dataset

ML Classifier	Feature Matrix	Train Accuracy (%)	Test Accuracy (%)	Train Precision	Test Precision	Train Recall	Test Recall	Train F1-score	Test F1-score
Multinomial Naive Bayes	**POS + Uni-gram**	**88.24%**	**83.30%**	**0.89**	**0.84**	**0.88**	**0.83**	**0.88**	**0.83**
	POS + Bi-gram	**97.96%**	**80.00%**	**0.98**	**0.81**	**0.98**	**0.80**	**0.98**	**0.80**
Bernoulli Naive Bayes	**POS + Uni-gram**	**74.60%**	**71.92%**	**0.77**	**0.74**	**0.75**	**0.72**	**0.74**	**0.71**
	POS + Bi-gram	**83.54%**	**68.36%**	**0.88**	**0.78**	**0.84**	**0.68**	**0.83**	**0.65**
Random Forrest	**POS + Uni-gram**	**99.92%**	**81.00%**	**1.00**	**0.81**	**1.00**	**0.81**	**1.00**	**0.81**
	POS + Bi-gram	**99.79%**	**80.42%**	**1.00**	**0.81**	**1.00**	**0.80**	**1.00**	**0.80**
Logistic Regression	**POS + Uni-gram**	**88.80%**	**84.54%**	**0.89**	**0.85**	**0.89**	**0.85**	**0.89**	**0.85**
	POS + Bi-gram	**96.56%**	**82.11%**	**0.97**	**0.82**	**0.97**	**0.82**	**0.97**	**0.82**
SVM	**POS + Uni-gram**	**91.66%**	**85.25%**	**0.92**	**0.85**	**0.92**	**0.85**	**0.92**	**0.85**
	POS + Bi-gram	**98.96%**	**83.10%**	**0.99**	**0.83**	**0.99**	**0.83**	**0.99**	**0.83**
K-Nearest Neighbors	**POS + Uni-gram**	**84.57%**	**82.58%**	**0.85**	**0.83**	**0.85**	**0.83**	**0.85**	**0.83**
	POS + Bi-gram	**59.60%**	**50.15%**	**0.63**	**0.50**	**0.60**	**0.50**	**0.57**	**0.46**
Decision Tree	**POS + Uni-gram**	**99.93%**	**75.15%**	**1.00**	**0.75**	**1.00**	**0.75**	**1.00**	**0.75**
	POS + Bi-gram	**99.82%**	**73.90%**	**1.00**	**0.74**	**1.00**	**0.74**	**1.00**	**0.74**

Here, In the Yelp Restaurant Dataset **[23]**, again we applied the same model of data pre-processing as previous dataset and for feature extraction we applied TF-IDF. We applied the same machine learning algorithms using both unigram and bigram feature.

By doing these, we have achieved test accuracy rate of 83.30% with Multinomial Naïve bayes classifier using unigram feature. The training accuracy is 88.24%. The overall test precision, recall and f-1 score are 0.83. We can see a slight drop of test accuracy in the bigram implementation of Multinomial Naïve bayes classifier which is 80.00%. The overall test precision, recall and f-1 score are also dropped.

In Bernoulli Naïve Bayes classifier using unigram feature, we have achieved the test accuracy of 71.92% against training accuracy of 74.60%. In bigram, the increase in the frequency of words results an improvement in train accuracy of 83.54% but decreases in test accuracy which is 68.36%.

We applied the same parameters of n_estimators, max_depth and n_jobs as previous in Random Forrest classifier using unigram feature and achieve test accuracy is 81.00% against train accuracy 99.92%. The train precision, recall and f-1 score are 1.00 against test precision, recall and f-1 score of 0.81. In bigram implementation, we see a slight decreases in the test accuracy of 80.42% against train accuracy of 99.79%. The test precision, recall and f-1 score are almost same as unigram.

In Logistic Regression, the test accuracy is 84.54% against train accuracy of 88.80% in unigram. The test precision, recall and f-1 score are 0.85. In bigram, we can see a small drop in test accuracy as previous.

In both k-NN and Decision tree, we achieve the test accuracy is 82.58% and 75.15% against train accuracy of 84.57% and 99.93% respectively using unigram feature. In bigram k-NN, we can see a large drop in test accuracy which is 50.15% against train accuracy of 59.60% The train precision, recall and f-1 score are 0.63, 0.60, 0.57 against test precision, recall and f-1 score of 0.50, 0.50 and 0.47. Of all the classifiers, this is the lowest accuracy that we have achieve. In bigram Decision Tree the test accuracy is 73.90% against train accuracy of 100%.

Again, for this dataset also, we have achieved the highest test accuracy of 85.25% against train accuracy of 91.66% in Support Vector Machine (Linear) classifier using unigram feature. The test precision, recall and f-1 score are also quite high. In bigram implementation, we can see a drop of test accuracy which is 83.10%. The test precision, recall and f-1 score are 0.83.

From this experiment on both datasets, if we compare these classifiers based on our findings, we can come up to this conclusion that Support Vector Machine (Linear) classifier using unigram feature represents the suitable model for our fake review detection of all machine learning algorithms.

Table 8.3: Comparison of Hotel and Restaurant Datasets Using Machine Learning Algorithms

	Hotel Dataset					**Restaurant Dataset**				
ML Classifier	Train Accuracy (%)	Test Accuracy (%)	Test Precision	Test Recall	Test F1-score	Train Accuracy (%)	Test Accuracy (%)	Test Precision	Test Recall	Test F1-score
Multinomial Naive Bayes	96.17%	84.69%	0.86	0.85	0.85	88.24%	83.30%	0.84	0.83	0.83
Bernoulli Naive Bayes	96.56%	85.31%	0.86	0.85	0.85	74.60%	71.92%	0.74	0.72	0.71
Random Forrest	100%	86.25%	0.86	0.86	0.86	99.92%	81.00%	0.81	0.81	0.81
Logistic Regression	97.34%	86.56%	0.87	0.87	0.87	88.80%	84.54%	0.85	0.85	0.85
Support Vector Machine (Linear)	**98.75%**	**88.45%**	**0.89**	**0.88**	**0.88**	**91.66%**	**85.25%**	**0.85**	**0.85**	**0.85**
K-Nearest Neighbors	98.75%	88.45%	0.85	0.85	0.85	84.57%	82.58%	0.83	0.83	0.83
Decision Tree	100%	70.31%	0.71	0.70	0.70	99.93%	75.15%	0.75	0.75	0.75

This table represents the results obtain using machine learning algorithms unigram features on both datasets. Here, we can see in both cases, Support Vector Machine (Linear) performs better than other machine learning algorithms.

If we look at the datasets, we can see hotel dataset is significantly smaller than restaurant dataset. Despite being a small dataset, the test accuracy of hotel dataset is similar to restaurant dataset and in some cases it performs better. To understand why, we need to look at the review length and word distribution of the datasets.

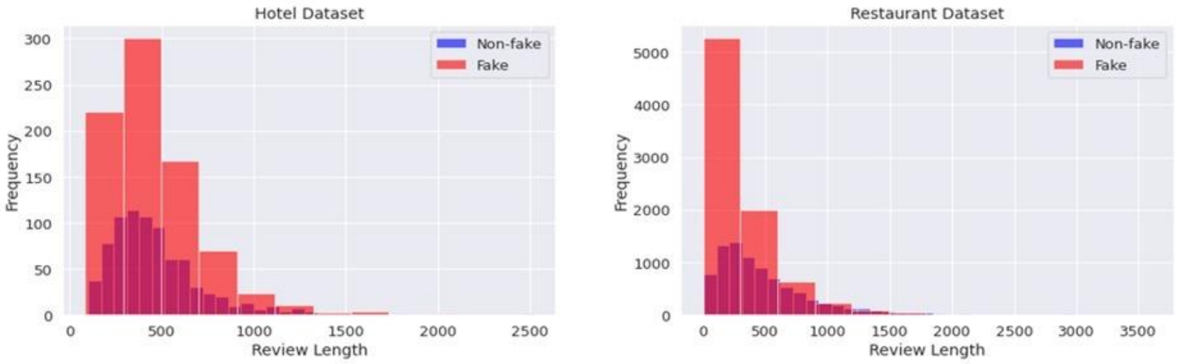


Figure 8.1: Plotting of review length vs word frequency

In Figure 8.1, we can see the review length and word frequency of two datasets. The review length represents the length of the reviews and frequency represents how frequent a word appears in a document. Despite the difference between the number of reviews, both datasets appear quite similar as the review length and word frequency of fake reviews are much higher than non-fake reviews. This is one of the reasons, the hotel dataset performs quite similar like restaurant dataset.

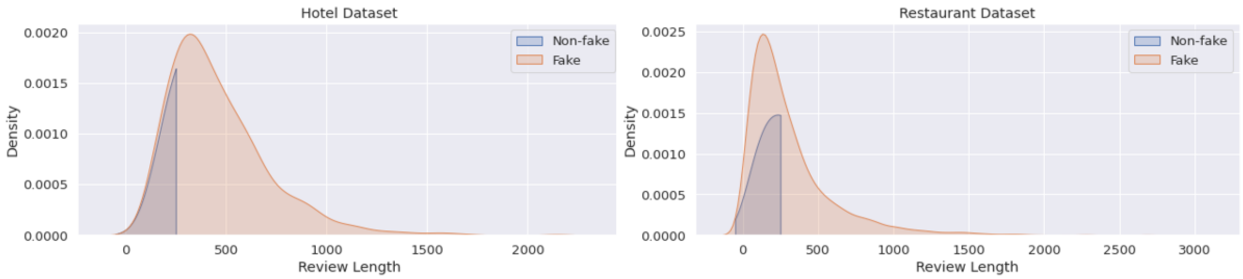


Figure 8.2: Plotting of review length vs word density

Also in Figure 8.2, we can see the review length vs word density of two datasets which are quite similar with each other. This plot shows that in both datasets, fake reviews have more words per review than non-fake reviews. In hotel dataset, non-fake reviews seem to be concentrated around 100 to 150 words and fake reviews seem to be concentrated around 300 to 400 words. In restaurant dataset, both non-fake reviews and fake reviews seem to be concentrated around 200 to 400 words.

In Figure 8.3 and Figure 8.2, we can see both hotel and restaurant datasets appear quite similar, as the number of fake reviews is higher than non-fake reviews. This can explain, despite being a small dataset, the hotel dataset performs quite similar like restaurant dataset and in some cases performs better.

8.1 Applying Confusion Matrix for Machine Learning Algorithms

Here, we have applied the confusion matrix on our machine learning classifier models (unigram) such as Naive Bayes, Bernoulli Naive Bayes, Random Forrest, k-Nearest Neighbors, Logistic Regression, Support Vector Machine (Linear) and Decision Tree to observe the performance of the models.

For Hotel Dataset, the confusion matrix is shown from Figure 8.3to Figure 8.9:

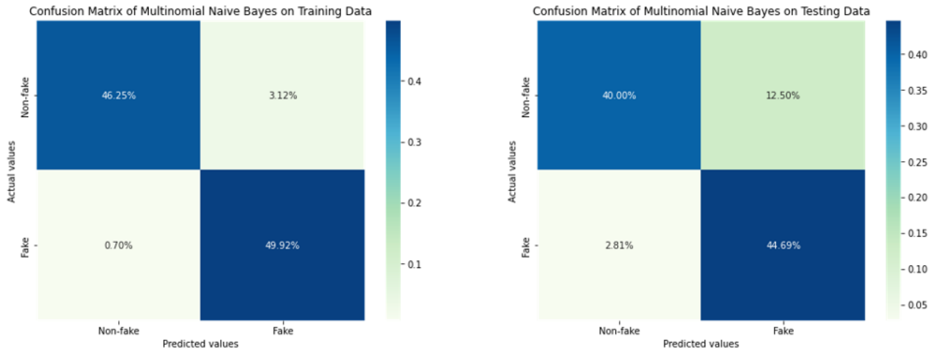


Figure 8.3: Confusion matrix of Multinomial Naive Bayes

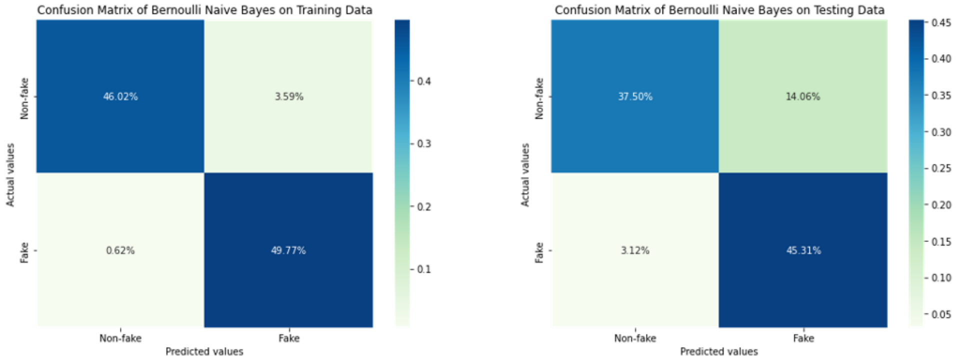


Figure 8.4: Confusion matrix of Bernoulli Naive Bayes

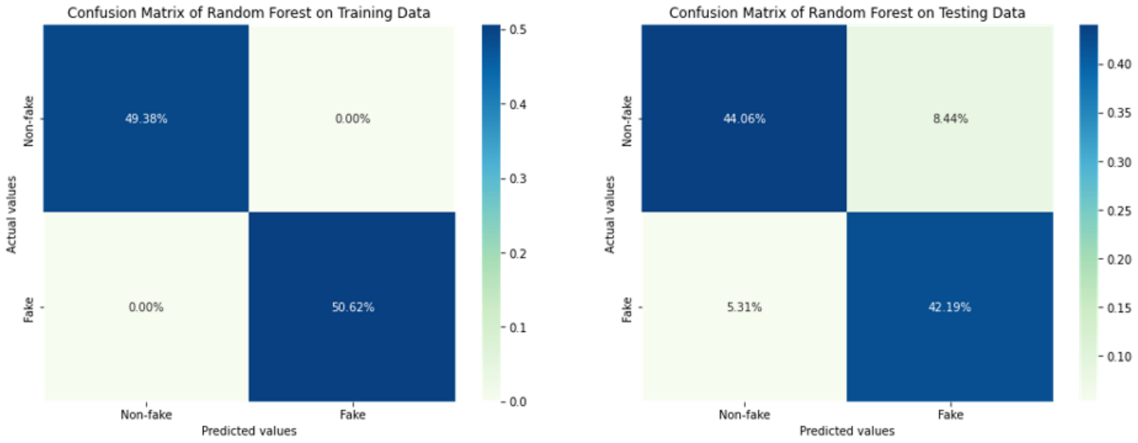


Figure 8.5: Confusion matrix of Random Forrest

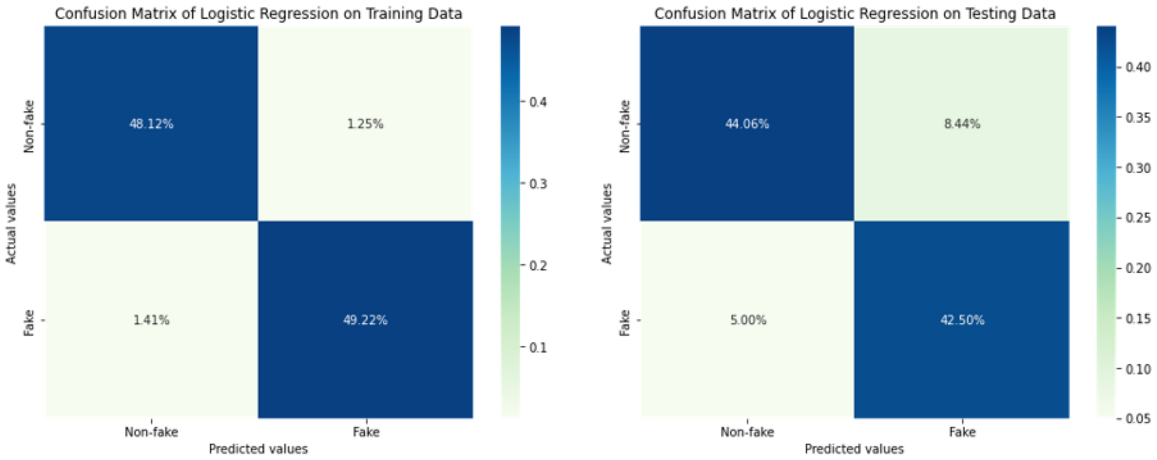


Figure 8.6: Confusion matrix of Logistic Regression

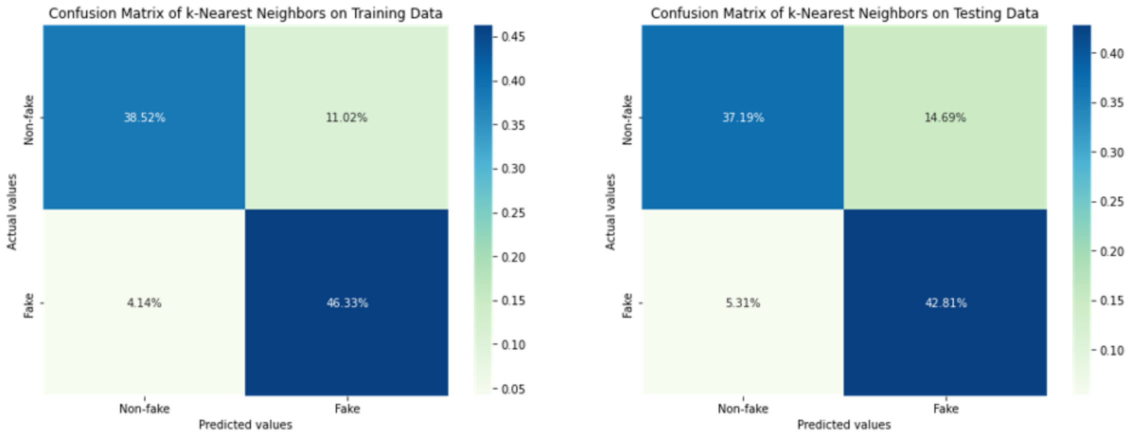


Figure 8.7: Confusion matrix of k-Nearest Neighbors

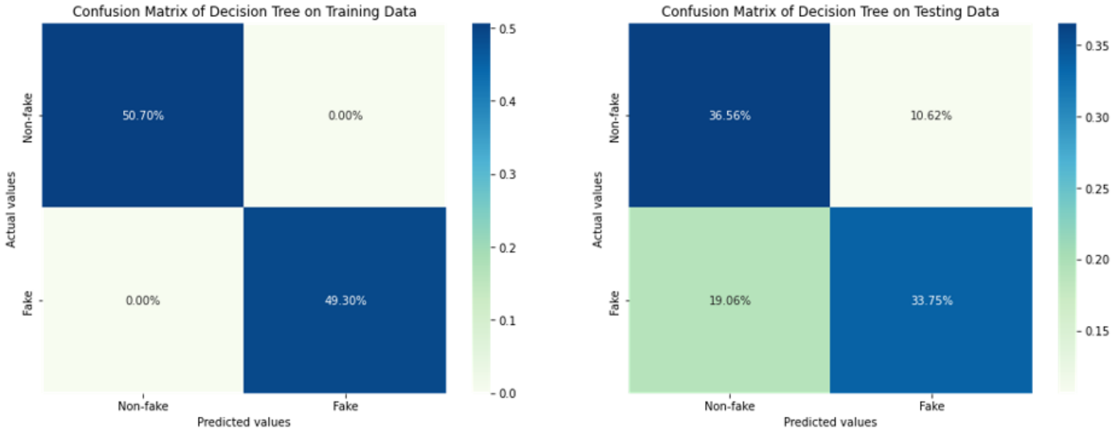


Figure 8.8: Confusion matrix of Decision Tree

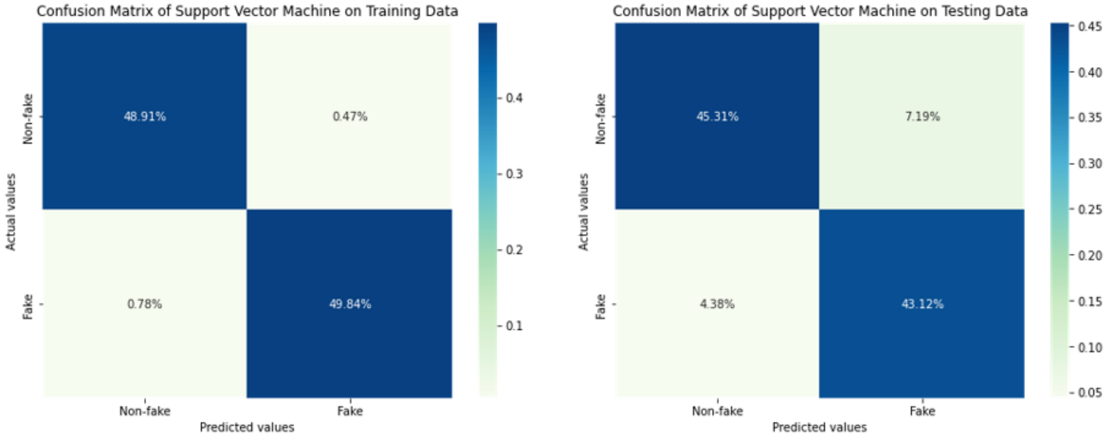


Figure 8.9: Confusion matrix of Support Vector Machine

For Restaurant Dataset, the confusion matrix is shown from Figure 8.10 to Figure 8.16:

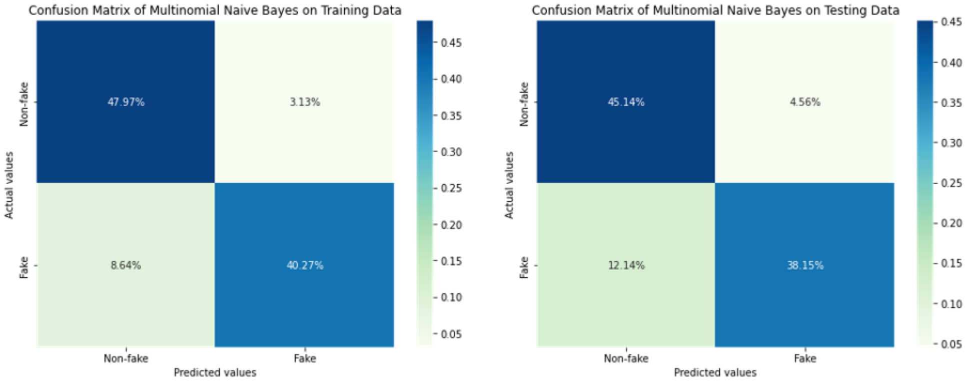


Figure 8.10: Confusion matrix of Multinomial Naive Bayes

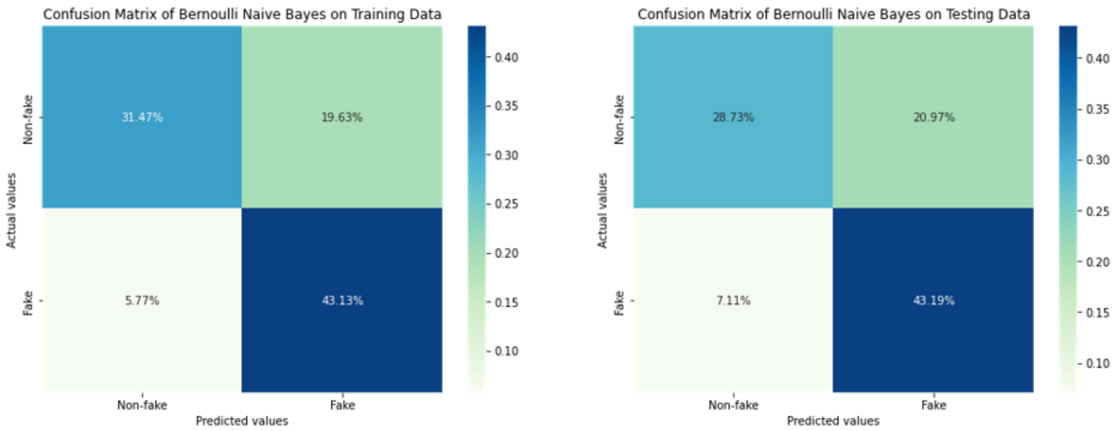


Figure 8.11: Confusion matrix of Bernoulli Naive Bayes

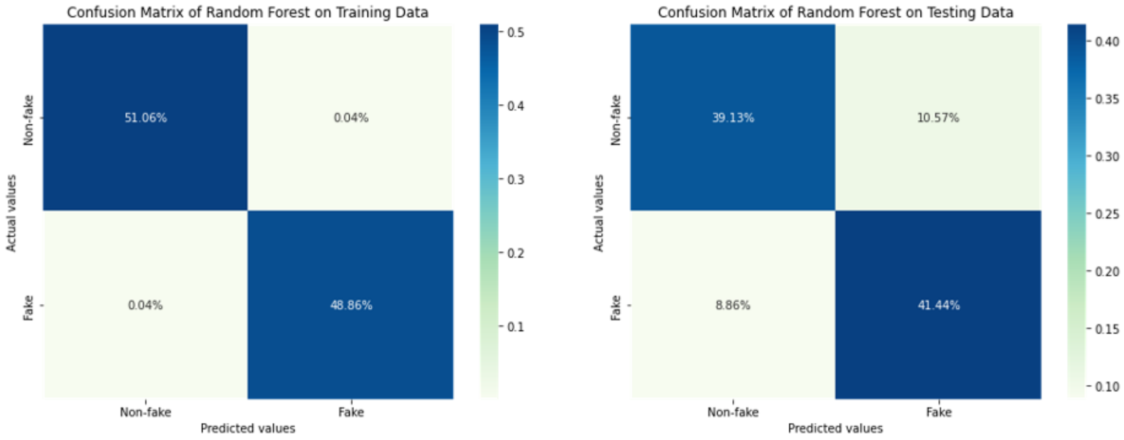


Figure 8.12: Confusion matrix of Random Forrest

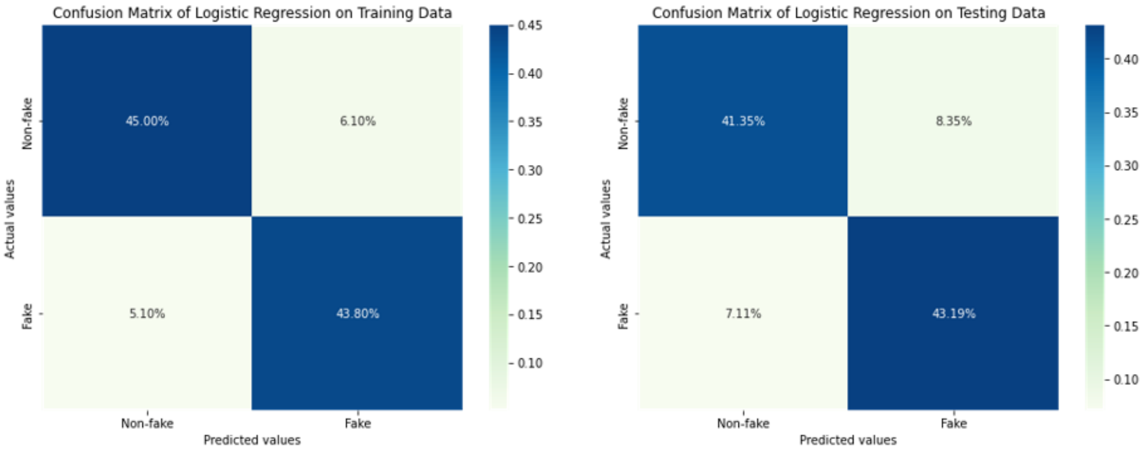


Figure 8.13: Confusion matrix of Logistic Regression

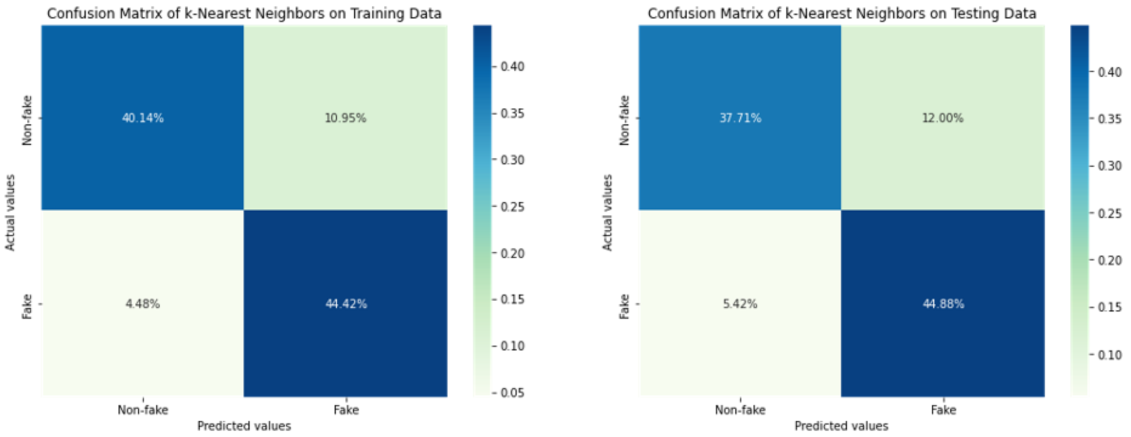


Figure 8.14: Confusion matrix of k-Nearest Neighbors

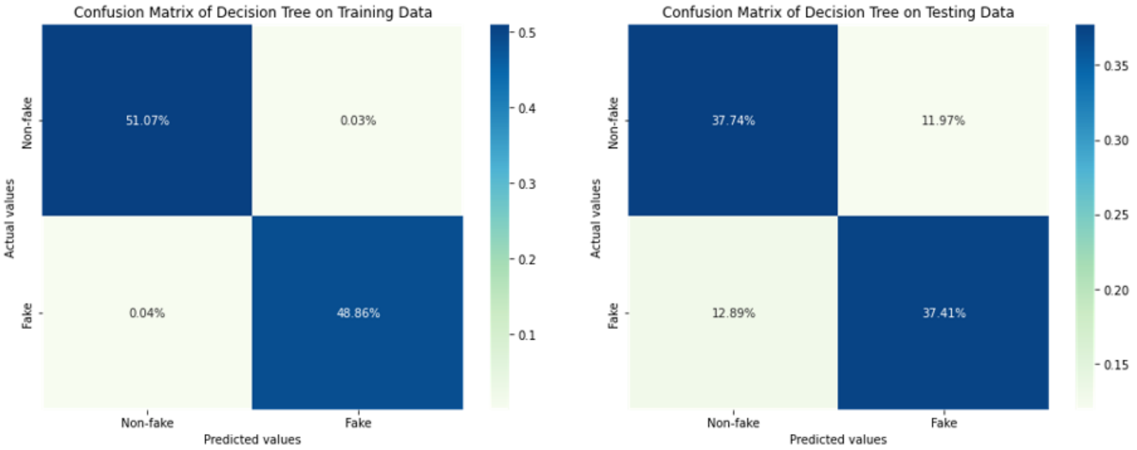


Figure 8.15: Confusion matrix of Decision Tree

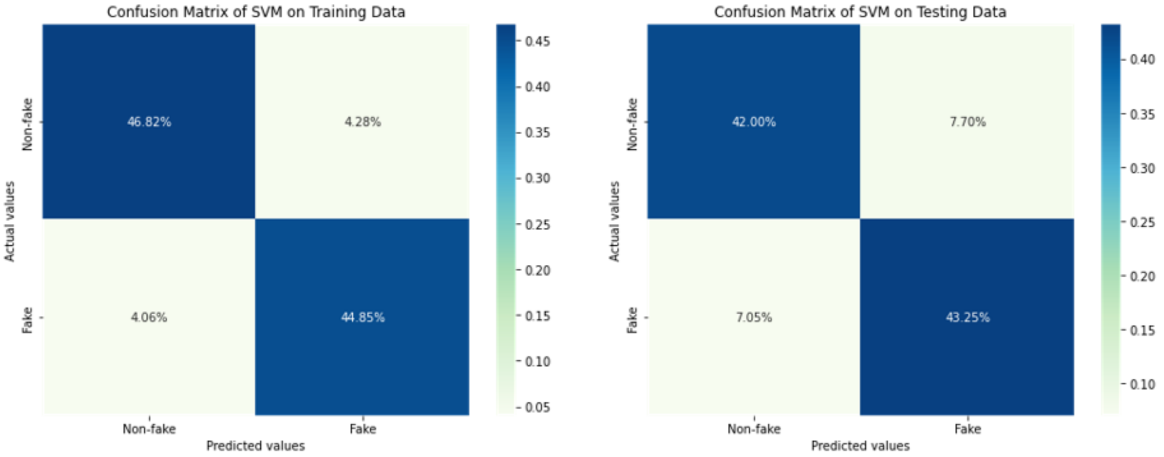


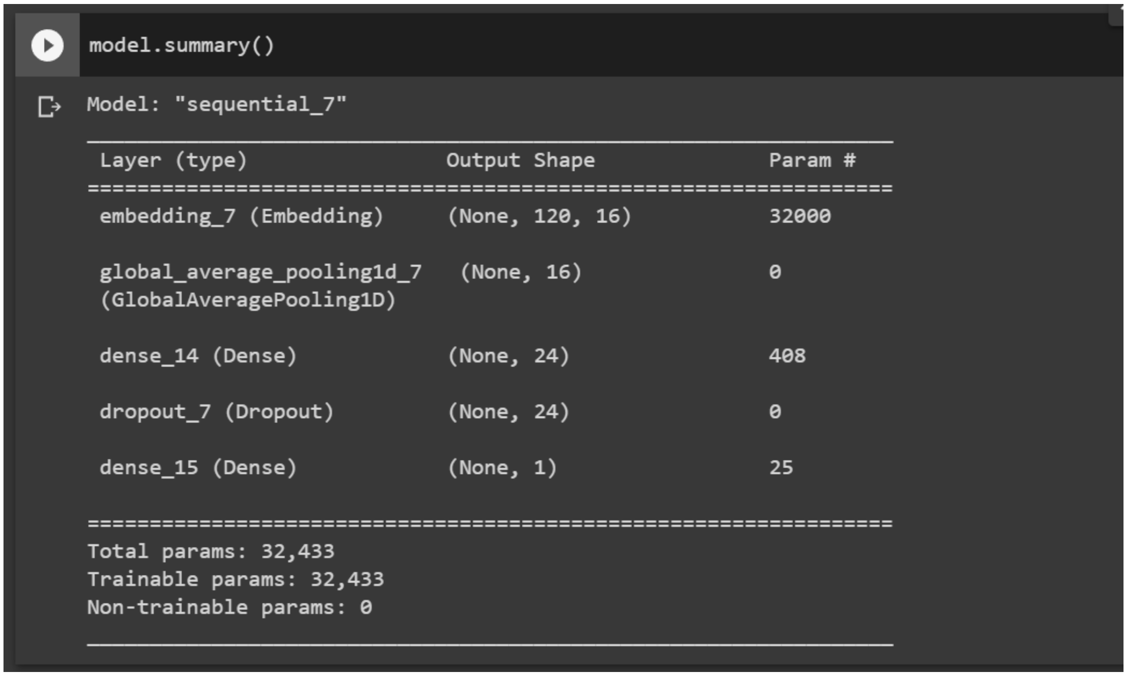
Figure 8.16: Confusion matrix of Support Vector Machine

The confusion matrix shows that the classifiers have correctly predicted a good number of fake and non-fake reviews with minor errors for both datasets. Based on these figures, we can see that we have created acceptable machine learning models.

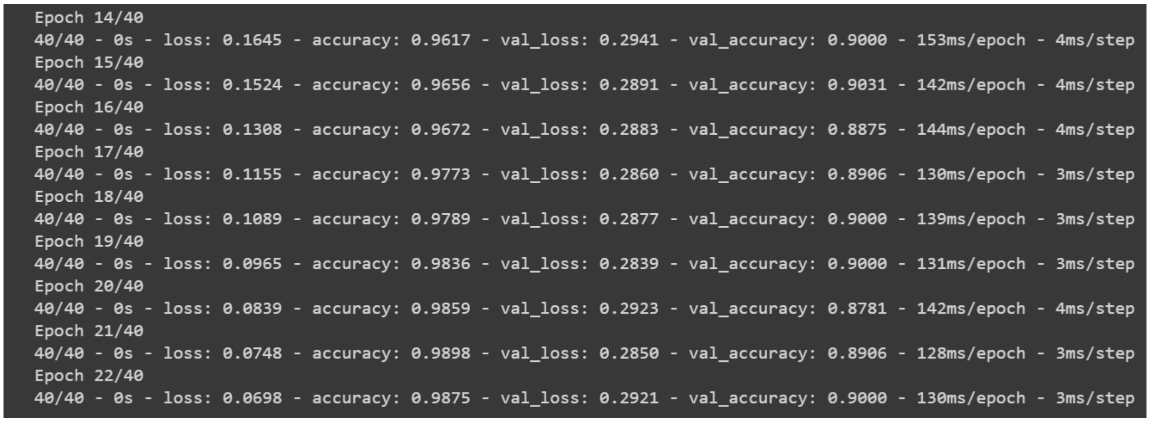
8.2 Applying Deep Learning

We applied three deep learning algorithms for our datasets: Dense Layer Architecture, RNN-LSTM or Long Short-Term Memory and RNN-BiLSTM or Bidirectional Long Short-Term Memory

For Dense Layer Architecture on Hotel Dataset, we have selected the vocab_size is 2000. The embedding dimension is 16, that means we are converting every single token of words into 16 dimension and the input length is 120. To avoid vanishing gradient problem, we use relu based activation function in deep learning to have better accuracy. For deep learning model, we use early stop method which stops the training of the model once it stops improving on a hold on the validation dataset. We applied the same process for LSTM and BiLSTM algorithms and results are below from Figure 8.17 to Figure 8.19:



```
model.summary()

Model: "sequential_7"
_________________________________________________________________
 Layer (type)                Output Shape              Param #   
=================================================================
 embedding_7 (Embedding)     (None, 120, 16)           32000     
                                                                 
 global_average_pooling1d_7   (None, 16)               0         
 (GlobalAveragePooling1D)                                        
                                                                 
 dense_14 (Dense)            (None, 24)                408       
                                                                 
 dropout_7 (Dropout)         (None, 24)                0         
                                                                 
 dense_15 (Dense)            (None, 1)                 25        
                                                                 
=================================================================
Total params: 32,433
Trainable params: 32,433
Non-trainable params: 0
_________________________________________________________________
```



```
Epoch 14/40
40/40 - 0s - loss: 0.1645 - accuracy: 0.9617 - val_loss: 0.2941 - val_accuracy: 0.9000 - 153ms/epoch - 4ms/step
Epoch 15/40
40/40 - 0s - loss: 0.1524 - accuracy: 0.9656 - val_loss: 0.2891 - val_accuracy: 0.9031 - 142ms/epoch - 4ms/step
Epoch 16/40
40/40 - 0s - loss: 0.1308 - accuracy: 0.9672 - val_loss: 0.2883 - val_accuracy: 0.8875 - 144ms/epoch - 4ms/step
Epoch 17/40
40/40 - 0s - loss: 0.1155 - accuracy: 0.9773 - val_loss: 0.2860 - val_accuracy: 0.8906 - 130ms/epoch - 3ms/step
Epoch 18/40
40/40 - 0s - loss: 0.1089 - accuracy: 0.9789 - val_loss: 0.2877 - val_accuracy: 0.9000 - 139ms/epoch - 3ms/step
Epoch 19/40
40/40 - 0s - loss: 0.0965 - accuracy: 0.9836 - val_loss: 0.2839 - val_accuracy: 0.9000 - 131ms/epoch - 3ms/step
Epoch 20/40
40/40 - 0s - loss: 0.0839 - accuracy: 0.9859 - val_loss: 0.2923 - val_accuracy: 0.8781 - 142ms/epoch - 4ms/step
Epoch 21/40
40/40 - 0s - loss: 0.0748 - accuracy: 0.9898 - val_loss: 0.2850 - val_accuracy: 0.8906 - 128ms/epoch - 3ms/step
Epoch 22/40
40/40 - 0s - loss: 0.0698 - accuracy: 0.9875 - val_loss: 0.2921 - val_accuracy: 0.9000 - 130ms/epoch - 3ms/step
```

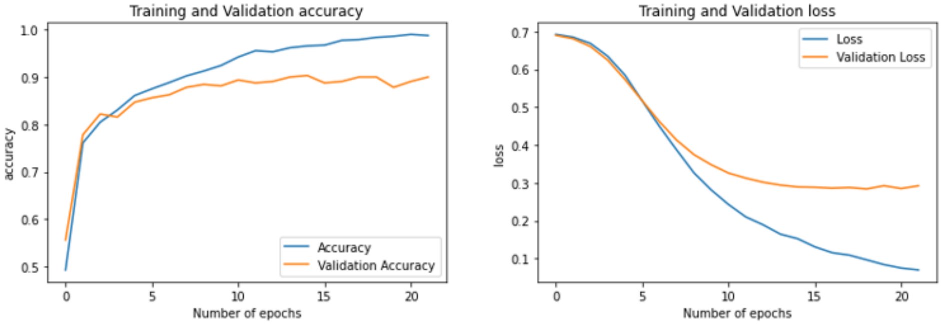
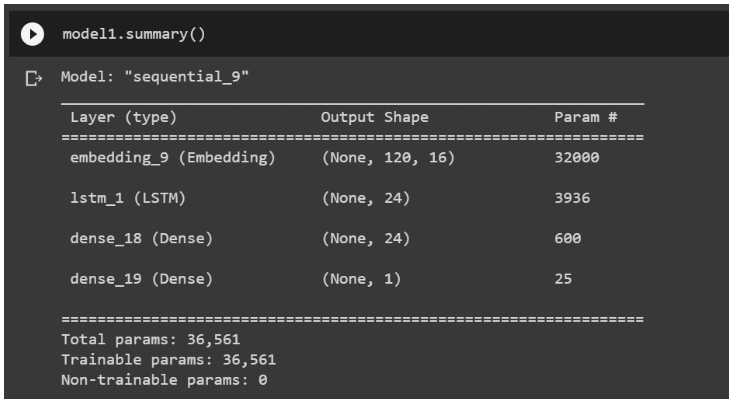
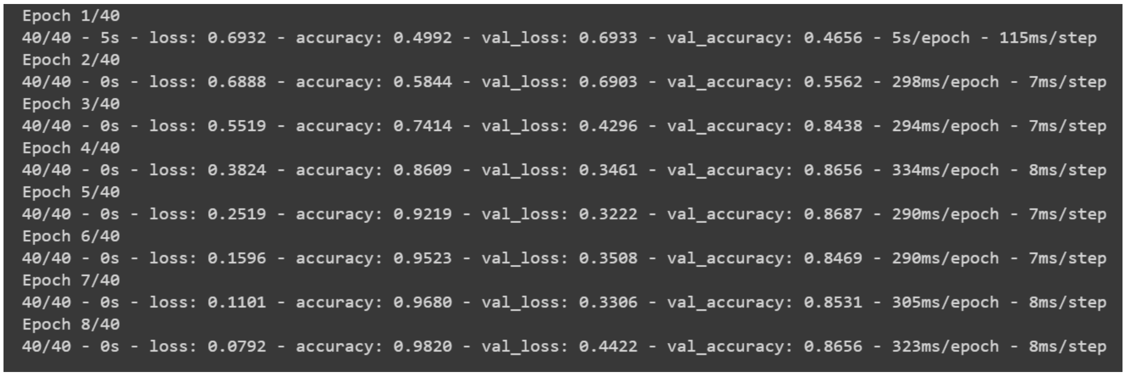


Figure 8.17: Dense Layer Architecture for Hotel Dataset



```
model1.summary()

Model: "sequential_9"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 embedding_9 (Embedding)     (None, 120, 16)           32000

 lstm_1 (LSTM)               (None, 24)                3936

 dense_18 (Dense)            (None, 24)                600

 dense_19 (Dense)            (None, 1)                 25

=================================================================
Total params: 36,561
Trainable params: 36,561
Non-trainable params: 0
```



```
Epoch 1/40
40/40 - 5s - loss: 0.6932 - accuracy: 0.4992 - val_loss: 0.6933 - val_accuracy: 0.4656 - 5s/epoch - 115ms/step
Epoch 2/40
40/40 - 0s - loss: 0.6888 - accuracy: 0.5844 - val_loss: 0.6903 - val_accuracy: 0.5562 - 298ms/epoch - 7ms/step
Epoch 3/40
40/40 - 0s - loss: 0.5519 - accuracy: 0.7414 - val_loss: 0.4296 - val_accuracy: 0.8438 - 294ms/epoch - 7ms/step
Epoch 4/40
40/40 - 0s - loss: 0.3824 - accuracy: 0.8609 - val_loss: 0.3461 - val_accuracy: 0.8656 - 334ms/epoch - 8ms/step
Epoch 5/40
40/40 - 0s - loss: 0.2519 - accuracy: 0.9219 - val_loss: 0.3222 - val_accuracy: 0.8687 - 290ms/epoch - 7ms/step
Epoch 6/40
40/40 - 0s - loss: 0.1596 - accuracy: 0.9523 - val_loss: 0.3508 - val_accuracy: 0.8469 - 290ms/epoch - 7ms/step
Epoch 7/40
40/40 - 0s - loss: 0.1101 - accuracy: 0.9680 - val_loss: 0.3306 - val_accuracy: 0.8531 - 305ms/epoch - 8ms/step
Epoch 8/40
40/40 - 0s - loss: 0.0792 - accuracy: 0.9820 - val_loss: 0.4422 - val_accuracy: 0.8656 - 323ms/epoch - 8ms/step
```

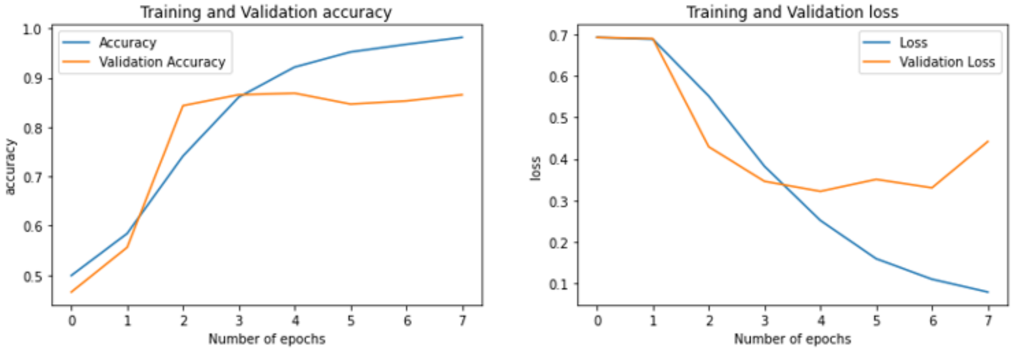
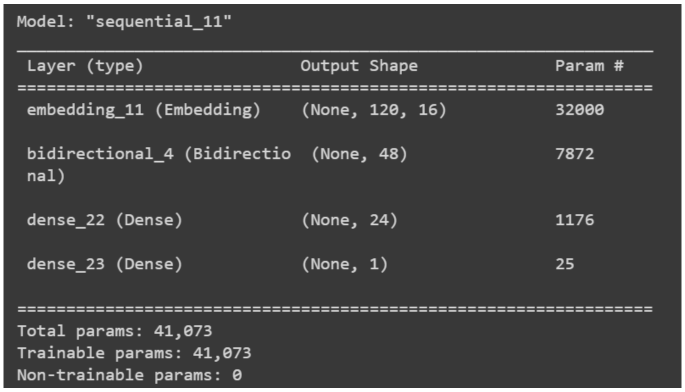
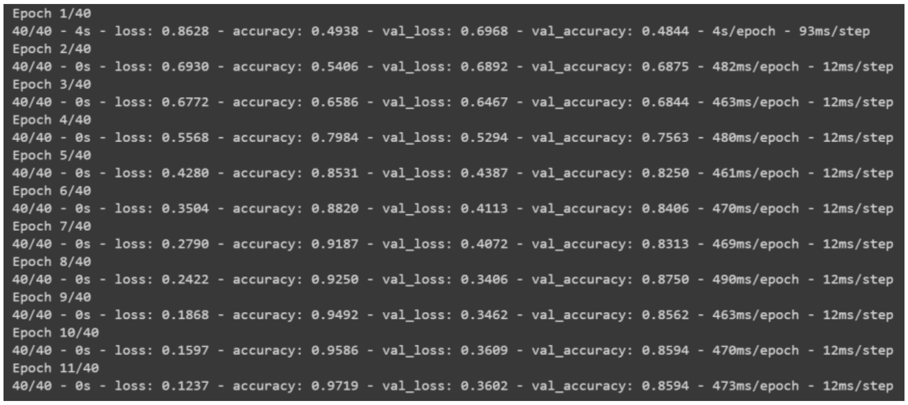


Figure 8.18: RNN-LSTM for Hotel Dataset



```
Model: "sequential_11"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 embedding_11 (Embedding)    (None, 120, 16)           32000

 bidirectional_4 (Bidirectio  (None, 48)               7872
 nal)

 dense_22 (Dense)            (None, 24)                1176

 dense_23 (Dense)            (None, 1)                 25

=================================================================
Total params: 41,073
Trainable params: 41,073
Non-trainable params: 0
```



```
Epoch 1/40
40/40 - 4s - loss: 0.8628 - accuracy: 0.4938 - val_loss: 0.6968 - val_accuracy: 0.4844 - 4s/epoch - 93ms/step
Epoch 2/40
40/40 - 0s - loss: 0.6930 - accuracy: 0.5406 - val_loss: 0.6892 - val_accuracy: 0.6875 - 482ms/epoch - 12ms/step
Epoch 3/40
40/40 - 0s - loss: 0.6772 - accuracy: 0.6586 - val_loss: 0.6467 - val_accuracy: 0.6844 - 463ms/epoch - 12ms/step
Epoch 4/40
40/40 - 0s - loss: 0.5568 - accuracy: 0.7984 - val_loss: 0.5294 - val_accuracy: 0.7563 - 480ms/epoch - 12ms/step
Epoch 5/40
40/40 - 0s - loss: 0.4280 - accuracy: 0.8531 - val_loss: 0.4387 - val_accuracy: 0.8250 - 461ms/epoch - 12ms/step
Epoch 6/40
40/40 - 0s - loss: 0.3504 - accuracy: 0.8820 - val_loss: 0.4113 - val_accuracy: 0.8406 - 470ms/epoch - 12ms/step
Epoch 7/40
40/40 - 0s - loss: 0.2790 - accuracy: 0.9187 - val_loss: 0.4072 - val_accuracy: 0.8313 - 469ms/epoch - 12ms/step
Epoch 8/40
40/40 - 0s - loss: 0.2422 - accuracy: 0.9250 - val_loss: 0.3406 - val_accuracy: 0.8750 - 490ms/epoch - 12ms/step
Epoch 9/40
40/40 - 0s - loss: 0.1868 - accuracy: 0.9492 - val_loss: 0.3462 - val_accuracy: 0.8562 - 463ms/epoch - 12ms/step
Epoch 10/40
40/40 - 0s - loss: 0.1597 - accuracy: 0.9586 - val_loss: 0.3609 - val_accuracy: 0.8594 - 470ms/epoch - 12ms/step
Epoch 11/40
40/40 - 0s - loss: 0.1237 - accuracy: 0.9719 - val_loss: 0.3602 - val_accuracy: 0.8594 - 473ms/epoch - 12ms/step
```

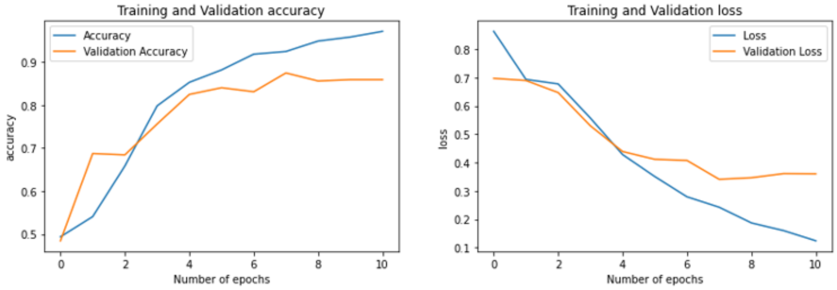
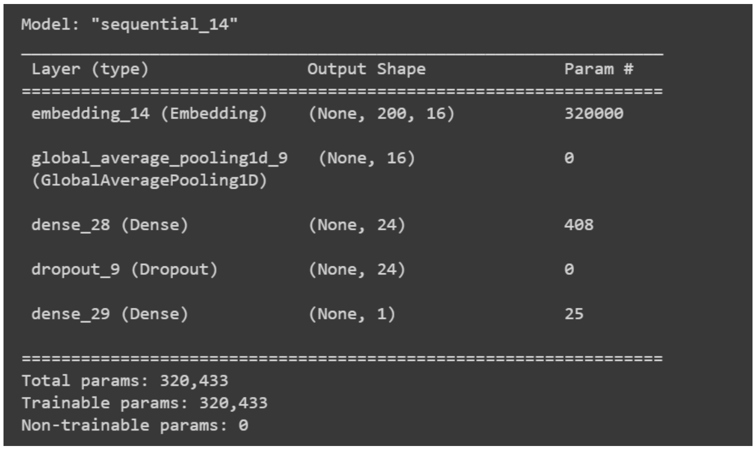


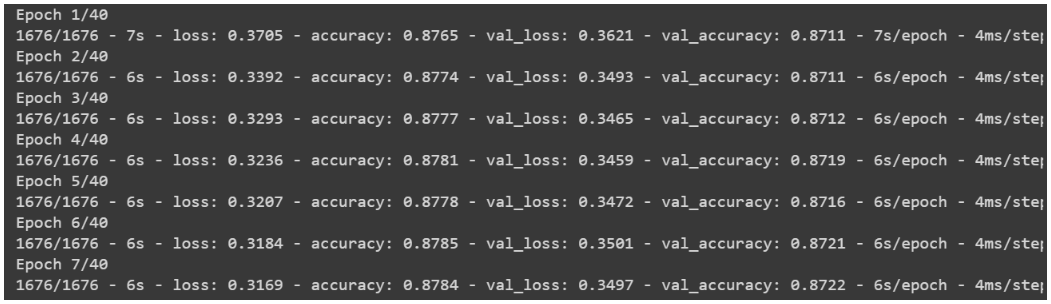
Figure 8.19: RNN-LSTM for Hotel Dataset

By applying Dense Layer Architecture, we have achieved the highest test accuracy of 90% for Hotel Dataset. The test accuracy of LSTM and Bi LSTM is 86.56% and 85.94% respectively.

For Restaurant Dataset, we have selected the vocab_size is 20000. The embedding dimension is 16, and the input length is 200. For Restaurant Dataset, we have achieved the highest test accuracy of 87.22%. The test accuracy of LSTM and Bi LSTM is 87.05% and 86.88% respectively. These are shown from Figure 8.22 to Figure 8.24:



```
Model: "sequential_14"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 embedding_14 (Embedding)    (None, 200, 16)           320000

 global_average_pooling1d_9   (None, 16)               0
 (GlobalAveragePooling1D)

 dense_28 (Dense)            (None, 24)                408

 dropout_9 (Dropout)         (None, 24)                0

 dense_29 (Dense)            (None, 1)                 25

=================================================================
Total params: 320,433
Trainable params: 320,433
Non-trainable params: 0
```



```
Epoch 1/40
1676/1676 - 7s - loss: 0.3705 - accuracy: 0.8765 - val_loss: 0.3621 - val_accuracy: 0.8711 - 7s/epoch - 4ms/ste
Epoch 2/40
1676/1676 - 6s - loss: 0.3392 - accuracy: 0.8774 - val_loss: 0.3493 - val_accuracy: 0.8711 - 6s/epoch - 4ms/ste
Epoch 3/40
1676/1676 - 6s - loss: 0.3293 - accuracy: 0.8777 - val_loss: 0.3465 - val_accuracy: 0.8712 - 6s/epoch - 4ms/ste
Epoch 4/40
1676/1676 - 6s - loss: 0.3236 - accuracy: 0.8781 - val_loss: 0.3459 - val_accuracy: 0.8719 - 6s/epoch - 4ms/ste
Epoch 5/40
1676/1676 - 6s - loss: 0.3207 - accuracy: 0.8778 - val_loss: 0.3472 - val_accuracy: 0.8716 - 6s/epoch - 4ms/ste
Epoch 6/40
1676/1676 - 6s - loss: 0.3184 - accuracy: 0.8785 - val_loss: 0.3501 - val_accuracy: 0.8721 - 6s/epoch - 4ms/ste
Epoch 7/40
1676/1676 - 6s - loss: 0.3169 - accuracy: 0.8784 - val_loss: 0.3497 - val_accuracy: 0.8722 - 6s/epoch - 4ms/ste
```

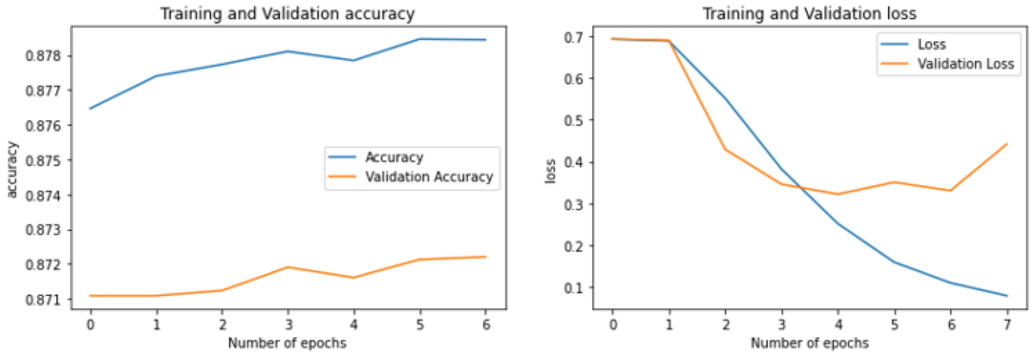
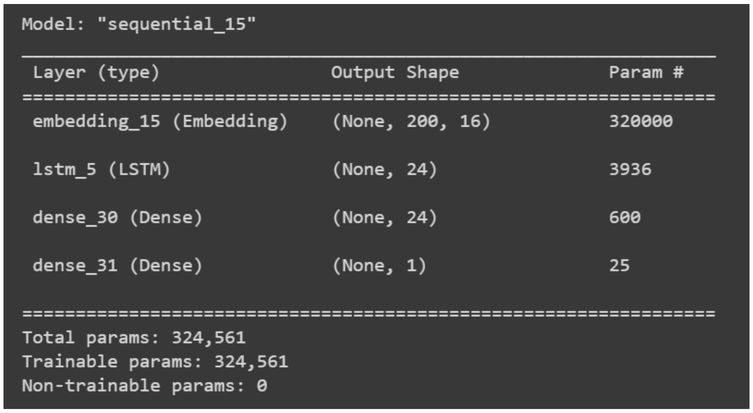
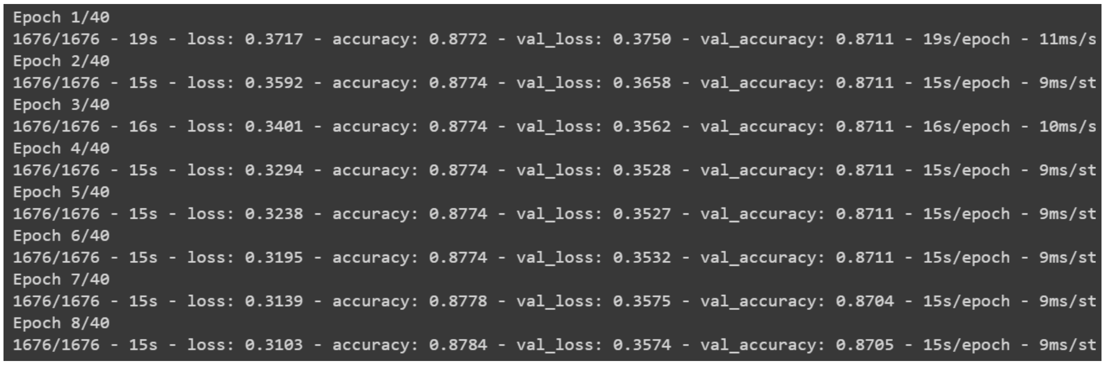


Figure 8.20: Dense Layer Architecture for Restaurant Dataset



```
Model: "sequential_15"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 embedding_15 (Embedding)    (None, 200, 16)           320000

 lstm_5 (LSTM)               (None, 24)                3936

 dense_30 (Dense)            (None, 24)                600

 dense_31 (Dense)            (None, 1)                 25

=================================================================
Total params: 324,561
Trainable params: 324,561
Non-trainable params: 0
```



```
Epoch 1/40
1676/1676 - 19s - loss: 0.3717 - accuracy: 0.8772 - val_loss: 0.3750 - val_accuracy: 0.8711 - 19s/epoch - 11ms/s
Epoch 2/40
1676/1676 - 15s - loss: 0.3592 - accuracy: 0.8774 - val_loss: 0.3658 - val_accuracy: 0.8711 - 15s/epoch - 9ms/st
Epoch 3/40
1676/1676 - 16s - loss: 0.3401 - accuracy: 0.8774 - val_loss: 0.3562 - val_accuracy: 0.8711 - 16s/epoch - 10ms/s
Epoch 4/40
1676/1676 - 15s - loss: 0.3294 - accuracy: 0.8774 - val_loss: 0.3528 - val_accuracy: 0.8711 - 15s/epoch - 9ms/st
Epoch 5/40
1676/1676 - 15s - loss: 0.3238 - accuracy: 0.8774 - val_loss: 0.3527 - val_accuracy: 0.8711 - 15s/epoch - 9ms/st
Epoch 6/40
1676/1676 - 15s - loss: 0.3195 - accuracy: 0.8774 - val_loss: 0.3532 - val_accuracy: 0.8711 - 15s/epoch - 9ms/st
Epoch 7/40
1676/1676 - 15s - loss: 0.3139 - accuracy: 0.8778 - val_loss: 0.3575 - val_accuracy: 0.8704 - 15s/epoch - 9ms/st
Epoch 8/40
1676/1676 - 15s - loss: 0.3103 - accuracy: 0.8784 - val_loss: 0.3574 - val_accuracy: 0.8705 - 15s/epoch - 9ms/st
```

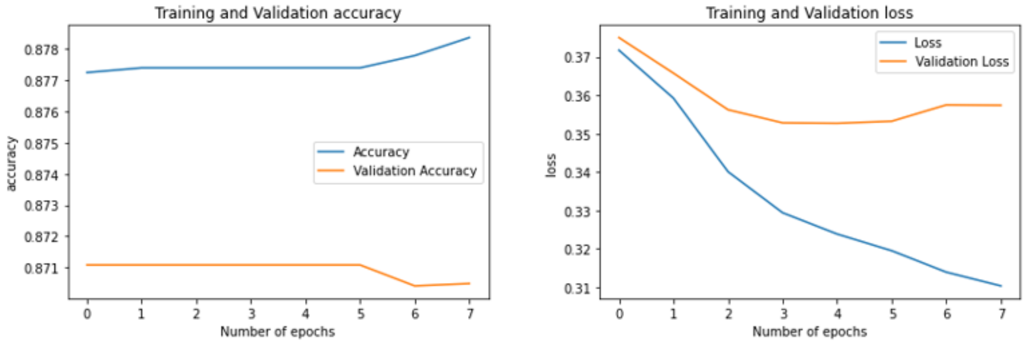
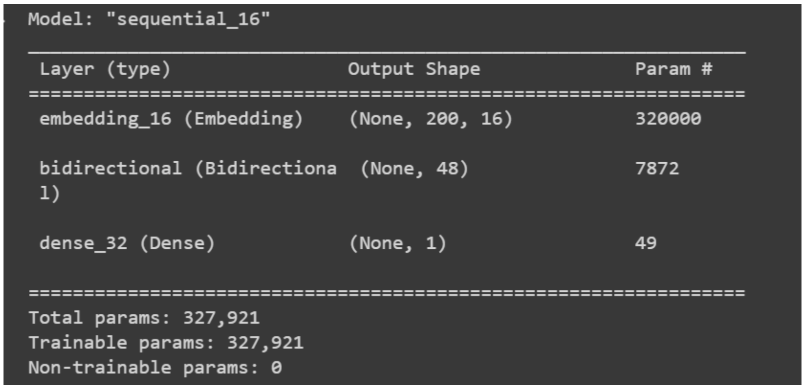
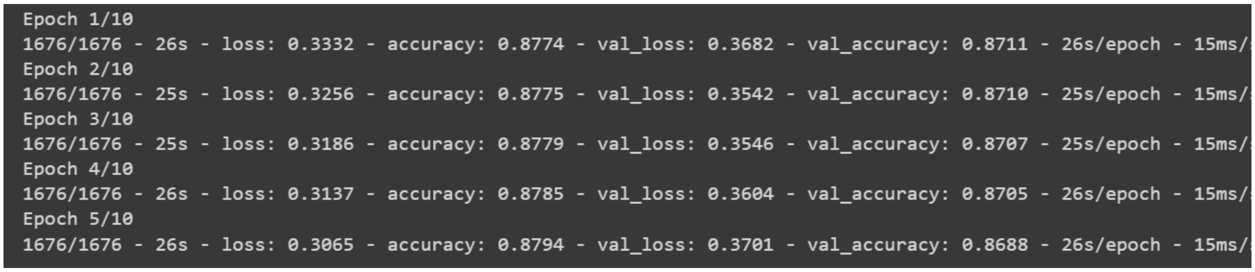


Figure 8.21: RNN-LSTM for Restaurant Dataset



```
Model: "sequential_16"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 embedding_16 (Embedding)    (None, 200, 16)           320000

 bidirectional (Bidirectiona  (None, 48)               7872
 l)

 dense_32 (Dense)            (None, 1)                 49

=================================================================
Total params: 327,921
Trainable params: 327,921
Non-trainable params: 0
```



```
Epoch 1/10
1676/1676 - 26s - loss: 0.3332 - accuracy: 0.8774 - val_loss: 0.3682 - val_accuracy: 0.8711 - 26s/epoch - 15ms/
Epoch 2/10
1676/1676 - 25s - loss: 0.3256 - accuracy: 0.8775 - val_loss: 0.3542 - val_accuracy: 0.8710 - 25s/epoch - 15ms/
Epoch 3/10
1676/1676 - 25s - loss: 0.3186 - accuracy: 0.8779 - val_loss: 0.3546 - val_accuracy: 0.8707 - 25s/epoch - 15ms/
Epoch 4/10
1676/1676 - 26s - loss: 0.3137 - accuracy: 0.8785 - val_loss: 0.3604 - val_accuracy: 0.8705 - 26s/epoch - 15ms/
Epoch 5/10
1676/1676 - 26s - loss: 0.3065 - accuracy: 0.8794 - val_loss: 0.3701 - val_accuracy: 0.8688 - 26s/epoch - 15ms/
```

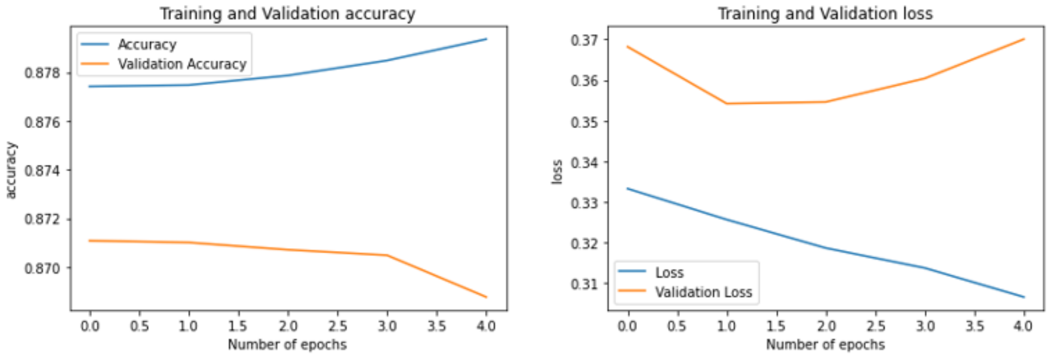


Figure 8.22: RNN-BiLSTM for Restaurant Dataset

CHAPTER NINE

Machine Learning vs Deep Learning

Machine learning is a branch of artificial intelligence and computer science which focuses on the use of data as input and algorithms to imitate the way humans learn and gradually becomes more accurate at predicting the outcomes **[71]**.

Deep learning is a subset of machine learning that is essentially a three-layer neural network. These neural networks aim to imitate the activity of the human brain by allowing it to "learn" from enormous amounts of data, albeit they are far from perfect. While a single layer neural network may produce approximate predictions, additional hidden layers can help to improve and tune for accuracy.

The advantages and challenges of machine learning and deep learning are given below:

9.1 Advantages of Machine Learning

- One of most important advantages of machine learning is it able to discover specific trends and patterns by analyzing large amount of data. Often, these patterns would not be apparent to human eyes. In most cases, a well build machine learning model can accurately identify those patterns and trends. For example, e-commerce companies like Amazon, Alibaba, eBay uses machine learning techniques to understand the browsing behaviors and purchasing histories of its user and predicts which products and deals that the users may find useful and push those products and deals.
- Machine learning models are capable of continuous accuracy improvement with the increasement of refined data as it trains more data. Internet companies like Amazon, Walmart collects huge volume of refined data every day and by training those data it improves the accuracy of recommended products and deals to its customers.
- An important feature of machine learning is the ability of automation on various decision-making tasks as it automatically performs many time-consuming and repetitive tasks to help to improve the model without human intervention. A common example is anti-virus software, which learn to filters new threats as they recognize with the help of machine learning techniques.

- Now these days, machine learning is being used by every industry possible. From state defense to small retail shops, it is being used as it helps to generate profit by reducing cost, automation, less human resources, and the ability to analyze patterns from previous data. It is being used in real-life applications such as image and speech recognition, fake review detection, self-driving cars, etc. **[72][73]**

9.2 Challenges of Machine Learning

- Data acquisition is one of the most important parts of machine learning, which can also be problematic. Data acquisition means collecting the data from a relevant source through surveys, real-life physical conditions, etc. before it can be used. In this process, there is a chance, it may contain imbalanced data, inaccurate data, or data full of errors. This can lead to poor accuracy in model building. Also, often time, to collect data, an organization have to pay for it. All this makes data acquisition a massive disadvantage.
- It is always important to remember that we need to provide well-cleaned data and apply feature engineering before we train it in the machine learning model. A dataset full of errors or imbalanced can causes incorrect results.
- There are different kinds of machine learning algorithms, and we need to identify which algorithm works best for the dataset. This is a manual process and also a disadvantage.
- When we process large volume of data in machine leaning model, the time complexity of that model increases so that we can have consideration amount of accuracy. It also needs massive amount of computing resources to process that dataset **[72][73]**.

9.3 Advantages of Deep Learning

- One of the import parts of artificial intelligence is the process of extracting features from raw data for better results. In deep learning, it can extract features by itself. A deep learning algorithm can analyze the data for features that are important and combine those data to learn faster.
- According to research from Gartner, up to 80% of organizations data are unstructured as it contains different formats of data such as texts, pictures, numbers, etc. **[74]**. Traditional

machine learning algorithms cannot analyze those data. Using deep learning algorithms, we can train those unstructured data and able to find patterns that are relevant.

- In deep learning, we don’t need well-labelled data as the model can learn form the dataset without any guidance. For example, a well-build deep learning algorithm can detect physical abnormality of the human body at earlier stages. Machine learning algorithms are not good in this kind of learning.
- A well-trained deep learning model can perform many time-consuming and repetitive tasks in a short period of time without human intervention.

9.4 Challenges of Deep Learning

- Deep learning requires a huge volume of data in order to perform better than other techniques as it trains the model itself. To learn about the data and try to solve the problem, the algorithm requires huge parameters to tune **[75]**.
- It is extremely costly to build a huge deep learning model as it requires expensive GPUs and computer equipment.
- One of the major challenges of deep learning is we don’t know how it arrives at a particular solution. Like human brain, the process of neural network is embedded in the thousands of simulated neurons arrange into interconnected layers **[75]**.
- Overfitting is a concept in deep learning, which occurs when a statistical model performs well in training data but doesn’t perform accurately in validation data. Just like overfitting, underfitting occurs when model hasn’t trained enough, or the inputs are not significant to determine relation between input and output **[76]**. This happens when we use large number of parameters and epochs. If the accuracy of some consecutive epochs does not vary, this is an indication of overfitting. By reducing parameter, layers and epochs we can overcome this problem.

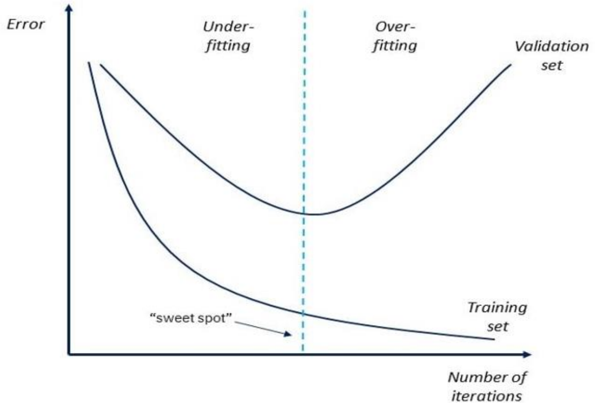


Figure 9.1: Overfitting and Underfitting

[Source: https://www.ibm.com/cloud/learn/overfitting]

9.5 Final Comparison of ML and DL

For our thesis paper, we have applied both machine learning and deep learning algorithms for fake review detection. With machine learning algorithms, we have achieved quite good test accuracies particularly with Support Vector Machine (Linear) and Logistic Regression with accuracies of 88.45% and 86.56% for Hotel Dataset and 85.25% and 84.54% for Restaurant Dataset respectively with higher precision, recall and f-1 score. But the implementation of deep learning on these datasets gives us better test accuracy then machine learning particularly Dense Layer Architecture with Validation Accuracy of 90% and 87.22% for Hotel and Restaurant Dataset respectively. LSTM and BiLSTM algorithms also perform better than most of the machine learning algorithms. So, if compare these results based on our finding on these particular datasets, we can come up to this conclusion that Dense Layer Architecture as well as deep learning algorithms represents the suitable method for fake review detection.

Table 9.1: Comparison of ML and DL algorithms

Model	Algorithms	Test Accuracy (%)	
		Hotel Dataset	Restaurant Dataset
Machine Learning	Multinomial Naive Bayes	84.69%	83.30%
	Bernoulli Naive Bayes	85.31%	71.92%
	Random Forrest	86.25%	81.00%
	Logistic Regression	86.56%	84.54%
	Support Vector Machine (linear)	88.45%	85.25%
	K-Nearest Neighbors	88.45%	82.58%
	Decision Tree	70.31%	75.15%
Deep Learning	Dense Layer Architecture	90%	87.22%
	RNN-LSTM	86.56%	87.05%
	RNN-BiLSTM	85.94%	86.88%

CHAPTER TEN

Conclusion

10.1 Research Challenges

There are a lot of challenges occur for detecting the fake reviews. From our point of view, dataset plays a major role for getting the better accuracy. Because in our case, we have applied the algorithms on two datasets we have collected. The first dataset (Restaurant Reviews) contains total 16,878 reviews and the second one (Hotel Reviews) contains 1,600 reviews only. After applying the machine learning and deep learning algorithms on both datasets, we have got overall a better accuracy for the second one. Besides that, there are some other challenges occur while detecting the fake reviews such as: when there is only one review available for a particular product it is difficult to detect whether the review was genuine or fake. Sometimes, it is hard to distinguish the fake or genuine review from the ratings. In some cases, sometimes fake reviews are written intentionally exactly following the way of the genuine reviews which makes a confusion about the behavior of that review. Also, before starting the training and testing, the classification and the preprocessing parts put a major impact on detecting the fake reviews.

10.2 Future Work

For improving the performance of the techniques which we have used, we will continue our research work in future, and we planned to propose some algorithms for detecting the fake reviews. We are also interested in applying BERT algorithm on our datasets to see how it performs as it is a latest natural language processing algorithm developed by Google and published in 2018. Moreover, we want to extend this work by performing similar analysis on a completely different dataset such as Twitter and Facebook. By classifying fake review from social media platforms, we hope to get one step closer towards building an automated fake review detection platform. We also hope this study provides a baseline for the future tests and broadens scope of the solutions dealing with fake review detection. The social media data will ensure that the variations in the language are taken care of. We would like to further dig deep and evaluate the effects of such review propagation and come up with simple techniques for faster prediction.

10.3 Conclusion

In this era of the digitalization, rapid development of the internet makes us involve with different online platforms. Nowadays, people are more interested in buying something from online rather than visiting a shopping center and whatever decision they are going to take is dependent on the feedback, someone has given on the website. That is why, online reviews play a crucial role in most of the people's life. Therefore, it is more important to find out the credible content and for that distinguish the genuine or fake review has become the vivid and ongoing research area. This Thesis paper represents machine learning algorithms such as Multinomial Naive Bayes, Bernoulli Naive Bayes, Random Forrest, k-Nearest Neighbors, Logistic Regression, Support Vector Machine (Linear), Decision Tree and deep leaning algorithms such as Dense Layer Architecture, LSTM and BiLSTM and we have showed the comparison between them to identify the fake reviews. The machine learning algorithms such as Logistic Regression, Support Vector Machine (Linear) provides better accuracy of all machine learning algorithms for both datasets. The Logistic Regression provides test accuracy of 86.56% and 84.54% for Hotel and Restaurant dataset and Support Vector Machine provides test accuracy of 88.45% and 85.25% for Hotel and Restaurant dataset. We have applied the Confusion Matrix for each of these algorithms to observe how well we build the model. In this case, we have achieved high precise, recall and f-1 score. When we applied Deep learning algorithms, it provides better accuracy than machine learning algorithms for both datasets particularly Dense layer Architecture which provides accuracy of 90% for Hotel Dataset and 87.22% for Restaurant Dataset. Here, both machine learning and deep learning algorithms perform well but deep learning algorithm particularly Dense Layer Architecture represents the suitable method for fake review detection for our dataset.

REFERENCES

1. https://www.jetir.org/papers/JETIR2104042.pdf Retrieved: 6th - February’2022, 10.27 PM
2. R. Mohawesh et al., "Fake Reviews Detection: A Survey," in IEEE Access, vol. 9, pp. 65771-65802, 2021, doi: 10.1109/ACCESS.2021.3075573
3. E. Elmurngi and A. Gherbi, Detecting Fake Reviews through Sentiment Analysis Using Machine Learning Techniques. IARIA/DATA ANALYTICS, 2017.
4. Wang, Z, T Hou, D Song, Z Li and T Kong, “Detecting review spammer groups via bipartite graph projection”, The Computer Journal, 59(6), pp. 861–874, 2015.
5. A. Molla, Y. Biadgie, and K.-A. Sohn, “Detecting Negative Deceptive Opinion from Tweets.” in International Conference on Mobile and Wireless Technology. Singapore: Springer, 2017.
6. Maheswari, s & S.S, Dhenakaran. (2021). Detection of Fake and Genuine Reviews with Hybridization of Fuzzy and Neural Networks Techniques.
7. SAHUT, J. M., LAROCHE, M., & BRAUNE, E. Antecedents and consequences of fake reviews: A marketing approach Short title (VSI): Fake Reviews.
8. http://www2.cs.uh.edu/~arjun/tr/UIC-CS-TR-yelp-spam.pdf Retrived: 4th January,2022, 3.09 PM
9. Wang, N., Yang, J., Kong, X., & Gao, Y. (2022). A fake review identification framework considering the suspicion degree of reviews with time burst characteristics. *Expert Systems With Applications, 190,* 116207. https://doi.org/10.1016/j.eswa.2021.116207
10. Salminen, J., Kandpal, C., Kamel, A., Jung, S., & Jansen, B. (2022). Creating and detecting fake reviews of online products. *Journal Of Retailing And Consumer Services, 64,* 102771. https://doi.org/10.1016/j.jretconser.2021.102771
11. Alsharif, N. (2022). Fake opinion detection in an e-commerce business based on a long-short memory algorithm. *Soft Computing*. https://doi.org/10.1007/s00500-022-06806-5
12. Khan H., Asghar M.U., Asghar M.Z., Srivastava G., Maddikunta P.K.R., Gadekallu T.R. (2021) Fake Review Classification Using Supervised Machine Learning. In: Del Bimbo A. et al. (eds) Pattern Recognition. ICPR International Workshops and Challenges. ICPR 2021. Lecture Notes in Computer Science, vol 12664. Springer, Cham. https://doi.org/10.1007/978-3-030-68799-1_19

13. Barbado, R., Araque, O., & Iglesias, C. (2019). A framework for fake review detection in online consumer electronics retailers. *Information Processing & Management*, *56*(4), 1234-1244. https://doi.org/10.1016/j.ipm.2019.03.002
14. Mukherjee, A., Venkataraman, V., Liu, B., & Glance, N. (2013). *Fake Review Detection: Classification and Analysis of Real and Pseudo Reviews* [Ebook] (p. 11). uh.edu. Retrieved 17 March 2022, from http://www2.cs.uh.edu/~arjun/tr/UIC-CS-TR-yelp-spam.pdf.
15. Hajek, P., Barushka, A., & Munk, M. (2020). Fake consumer review detection using deep neural networks integrating word embeddings and emotion mining. Neural Computing And Applications, 32(23), 17259-17274. doi: 10.1007/s00521-020-04757-2
16. Jindal, N., & Liu, B. (2007). Analyzing and Detecting Review Spam. *Seventh IEEE International Conference On Data Mining (ICDM 2007)*. doi: 10.1109/icdm.2007.68
17. Wang, G., Li, C., Wang, W., Zhang, Y., Shen, D., & Zhang, X. et al. (2018). Joint Embedding of Words and Labels for Text Classification. *Proceedings Of The 56Th Annual Meeting Of The Association For Computational Linguistics (Volume 1: Long Papers)*. doi: 10.18653/v1/p18-1216
18. Liu, Y., Pang, B., & Wang, X. (2019). Opinion spam detection by incorporating multimodal embedded representation into a probabilistic review graph. Neurocomputing, 366, 276-283. doi: 10.1016/j.neucom.2019.08.013
19. Li F, Huang M, Yang Y, Zhu X (2011) Learning to identify review spam. In: International joint conference on artificial intelligence (IJCAI 2011), pp 2488–2493
20. Chandy R, Gu H (2012) Identifying spam in the iOS app store. In: Proceedings of the 2nd joint WICOW/AIRWeb workshop on web quality, ACM, pp 56–59. https://doi.org/10.1145/2184305. 2184317
21. Ott M, Cardie C, Hancock JT (2013) Negative deceptive opinion spam. In: 2013 conference of the North American chapter of the association for computational linguistics: human language technologies, ACL, pp 497–501
22. Shojaee S, Murad MAA, Azman AB, Sharef NM, Nadali S (2013) Detecting deceptive reviews using lexical and syntactic features. In: 13th international conference on intelligent systems design and applications, IEEE, pp 53–58. https://doi.org/10.1109/ isda.2013.6920707

23. Mukherjee A, Venkataraman V, Liu B, Glance N (2013) What yelp fake review filter might be doing?. In: 7th international AAAI conference on weblogs and social media, AAAI, pp 409–418
24. Li J, Ott M, Cardie C, Hovy E (2014) Towards a general rule for identifying deceptive opinion spam. In: Proceedings of the 52nd annual meeting of the association for computational linguistics, ACL, vol 1, pp 1566–1576. https://doi.org/10.3115/v1/p14-1147
25. Li H, Chen Z, Mukherjee A, Liu B, Shao J (2015) Analyzing and detecting opinion spam on a large-scale dataset via temporal and spatial patterns. In: 9th international AAAI conference on web and social media (ICWSM 2015), AAAI, pp 634–637
26. Rayana S, Akoglu L (2015) Collective opinion spam detection: bridging review networks and metadata. In: 21th ACM SIGKDD international conference on knowledge discovery and data mining, ACM, pp 985–994. https://doi.org/10.1145/2783258. 2783370
27. Sun C, Du Q, Tian G (2016) Exploiting product related review features for fake review detection. Math Probl Eng 2016:1–7. https://doi.org/10.1155/2016/4935792
28. Li L, Qin B, Ren W, Liu T (2017) Document representation and feature combination for deceptive spam review detection. Neurocomputing 254:33–41. https://doi.org/10.1016/j.neucom.2016. 10.080
29. Ren Y, Ji D (2017) Neural networks for deceptive opinion spam detection: an empirical study. Inf Sci 385:213–224. https://doi. org/10.1016/j.ins.2017.01.015
30. Elmurngi E, Gherbi A (2017) An empirical study on detecting fake reviews using machine learning techniques. In: 7th international conference on innovative computing technology (INTECH), IEEE, pp 107–114. https://doi.org/10.1109/intech. 2017.8102442
31. Rout JK, Dash AK, Ray NK (2018) A framework for fake review detection: issues and challenges. In: 2018 international conference on information technology (ICIT), IEEE, pp 7–10. https:// doi.org/10.1109/icit.2018.00014
32. Yilmaz CM, Durahim AO (2018) SPR2EP: a semi-supervised spam review detection framework. In: 2018 IEEE/ACM international conference on advances in social networks analysis and mining (ASONAM), IEEE, pp 306–313. https://doi.org/10.1109/ asonam.2018.8508314
33. Ahmed H, Traore I, Saad S (2018) Detecting opinion spams and fake news using text classification. Secur Priv 1(1):e9. https://doi. org/10.1002/spy2.9

34. Zeng ZY, Lin JJ, Chen MS, Chen MH, Lan YQ, Liu JL (2019) A review structure based ensemble model for deceptive review spam. Information 10(7):243. https://doi.org/10.3390/ info10070243
35. Barbado R, Araque O, Iglesias CA (2019) A framework for fake review detection in online consumer electronics retailers. Inf Process Manag 56(4):1234–1244. https://doi.org/10.1016/j. indmarman.2019.08.003
36. Kennedy S, Walsh N, Sloka K, McCarren A, Foster J (2019) Fact or factitious? Contextualized opinion spam detection. In: Proceedings of the 57th annual meeting of the association for computational linguistics: student research workshop, ACL, pp 344–350. https://doi.org/10.18653/v1/p19-2048
37. Liu Y, Pang B, Wang X (2019) Opinion spam detection by incorporating multimodal embedded representation into a probabilistic review graph. Neurocomputing 366:276–283. https://doi. org/10.1016/j.neucom.2019.08.013
38. Barushka A, Hajek P (2019) Review spam detection using word embeddings and deep neural networks. In: MacIntyre J, Maglogiannis I, Iliadis L, Pimenidis E (eds) Artificial intelligence applications and innovations. AIAI 2019, vol 559. IFIP advances in information and communication technology. Springer, Cham, pp 340–350. https://doi.org/10.1007/978-3-030-19823-7_28
39. Patel, Nidhi A., and Rakesh Patel. "A survey on fake review detection using machine learning techniques." *2018 4th International Conference on Computing Communication and Automation (ICCCA)*. IEEE, 2018.
40. Mohawesh, R., Xu, S., Tran, S. N., Ollington, R., Springer, M., Jararweh, Y., & Maqsood, S. (2021). Fake reviews detection: A survey. *IEEE Access*, *9*, 65771-65802.
41. Kaur, G., & Malik, K. (2021). A comprehensive overview of sentiment analysis and fake review detection. *Mobile Radio Communications and 5G Networks*, 293-304.
42. https://data-flair.training/blogs/data-science-r-sentiment-analysis-project/ Retrieved: 6th - February’2022, 11.00 PM
43. https://www.reviewtrackers.com/blog/fake-reviews/ Retrieved: 12th - February’2022, 8.00 PM

44. N. Jindal and B. Liu, “Opinion spam and analysis,” in Proceedings of the 2008 International Conference on Web Search and Data Mining, ser. WSDM ’08, 2008, pp. 219–230.
45. S. Banerjee and A.Y.K. Chua. 2014. “Applauses in hotel reviews: Genuine or deceptive ?”, 2014 Science and Information Conference (2014), pp. 938–942,2014.
46. https://youtu.be/ooRAKphG8ic Retrieved: 12th - February’2022, 10.05 PM
47. A. Rastogi, M. Mehrotra, “Opinion spam Detection in Online Reviews”, Journal of information and Knowledge Management, vol. 16, no. 04, pp. 1-38, 2017.
48. Thakkar, H., & Patel, D. (2015). Approaches for sentiment analysis on twitter: A state-of-art study. *arXiv preprint arXiv:1512.01043*.
49. Alessia, D., Ferri, F., Grifoni, P., & Guzzo, T. (2015). Approaches, tools and applications for sentiment analysis implementation. *International Journal of Computer Applications*, *125*(3).
50. Garcia-Rodriguez, J., Angelopoulou, A., Tomás, D., & Lewis, A. (2019). Complex methods applied to data analysis, processing, and visualisation. *Complexity*, *2019*.
51. https://www.mathworks.com/discovery/deep-learning.html Retrieved: 20th - February’2022, 09.50 PM
52. M. S. A. Mr. Srikanth Tammina, "Sentiment Analysis on Customer Reviews using Convolutional Neural Network".
53. El Naqa, I., & Murphy, M. (2015). What Is Machine Learning?. *Machine Learning In Radiation Oncology*, 3-11. doi: 10.1007/978-3-319-18305-3_1
54. https://www.upgrad.com/blog/multinomial-naive-bayes-explained/ Retrieved: 23rd-January’2022, 4.00PM
55. https://medium.com/@nansha3120/bernoulli-naive-bayes-and-its-implementation-cca33ccb8d2e Retrieved: 23rd-January’2022, 4.30PM
56. J. K. Jaiswal and R. Samikannu, "Application of Random Forest Algorithm on Feature Subset Selection and Classification and Regression," 2017 World Congress on Computing and Communication Technologies (WCCCT), 2017, pp. 65-68, doi: 10.1109/WCCCT.2016.25.
57. Peterson, L. E. (2009). K-nearest neighbor. *Scholarpedia*, *4*(2), 1883.

58. https://towardsdatascience.com/machine-learning-basics-with-the-k-nearest-neighbors-algorithm-6a6e71d01761 Retrieved: 30th-January’2022, 9.37PM
59. https://www.javatpoint.com/k-nearest-neighbor-algorithm-for-machine-learning Retrived: 26th May, 2022
60. https://towardsdatascience.com/introduction-to-logistic-regression-66248243c148 Retrieved: 30th-January’2022, 9.57PM
61. https://www.geeksforgeeks.org/decision-tree/ Retrieved: 30th-January’2022, 10.18PM
62. https://en.wikipedia.org/wiki/Support-vector_machine Retrieved: 25th-January’2022, 9.10PM
63. https://www.analyticsvidhya.com/blog/2017/09/understaing-support-vector-machine-example-code/ Retrieved: 25th-January’2022, 9.29PM
64. Tripathy, Abinash. Sentiment Analysis Using Machine Learning Techniques. Diss. 2017.
65. https://en.wikipedia.org/wiki/Deep_learning Retrieved: 24th-May’2022, 1.45PM
66. [https://analyticsindiamag.com/a-complete-understanding-of-dense-layers-in-neural-networks/] Retrieved: 24th May,2022, 2.08 AM
67. https://www.geeksforgeeks.org/introduction-to-recurrent-neural-network/ Retrieved: 24th-May’2022, 2.18PM
68. https://www.analyticsvidhya.com/blog/2021/03/introduction-to-long-short-term-memory-lstm/ Retrieved: 24th-May’2022, 2.39PM
69. https://paperswithcode.com/method/bilstm#:~:text=A%20Bidirectional%20LSTM%2C%20or%20biLSTM,other%20in%20a%20backwards%20direction.] Retrieved: 25th May, 2022, 1.25 PM
70. https://nlp.stanford.edu/IR-book/html/htmledition/stemming-and-lemmatization-1.html Retrieved: 25th May, 2022, 1.49 PM
71. https://www.ibm.com/cloud/learn/machine-learning Retrieved: 25th May, 2022, 2.12 PM
72. http://ivyproschool.com/blog/advantages-and-disadvantages-of-machine-learning-in-2020/ Retrieved: 25th May, 2022, 2.25 PM
73. https://data-flair.training/blogs/advantages-and-disadvantages-of-machine-learning/ Retrieved: 25th May, 2022, 2.39 PM

74. https://www.forbes.com/sites/forbestechcouncil/2017/06/05/the-big-unstructured-data-problem/ Retrieved: 25th May, 2022, 2.47 PM
75. https://indatalabs.com/blog/deep-learning-strengths-challenges Retrieved: 25th May, 2022, 10.41 PM
76. https://www.ibm.com/cloud/learn/overfitting Retrieved: 25th May, 2022, 10.57 PM
77. Bhowmik, N. R., Arifuzzaman, M., & Mondal, M. R. H. (2022). Sentiment analysis on Bangla text using extended lexicon dictionary and deep learning algorithms. Array, 100123.
78. Bhowmik, N. R., Arifuzzaman, M., Mondal, M. R. H., & Islam, M. S. (2021). Bangla text sentiment analysis using supervised machine learning with extended lexicon dictionary. Natural Language Processing Research, 1(3-4), 34-45
79. Toma, T., Hassan, S., & Arifuzzaman, M. (2021, July). An Analysis of Supervised Machine Learning Algorithms for Spam Email Detection. In 2021 International Conference on Automation, Control and Mechatronics for Industry 4.0 (ACMI) (pp. 1-5). IEEE.
80. Hasan, M. R., Maliha, M., & Arifuzzaman, M. (2019, July). Sentiment Analysis with NLP on Twitter Data. In 2019 International Conference on Computer, Communication, Chemical, Materials and Electronic Engineering (IC4ME2) (pp. 1-4). IEEE.
81. T. T. Chhowa, M. A. Rahman, A. K. Paul and R. Ahmmed, "A Narrative Analysis on Deep Learning in IoT based Medical Big Data Analysis with Future Perspectives," 2019 International Conference on Electrical, Computer and Communication Engineering (ECCE), Cox'sBazar, Bangladesh, 2019, pp. 1-6, doi: 10.1109/ECACE.2019.8679200.