FAKE REVIEW DETECTION AND ANALYSIS USING ML AND DL

ST. TERESA’S COLLEGE (AUTONOMOUS)

AFFILIATED TO MAHATMA GANDHI UNIVERSITY

THE FEAR OF THE LORD IS THE BEGINNING OF WISDOM

ST. TERESA’S COLLEGE
(AUTONOMOUS)
ESTD 1925

PROJECT REPORT

In partial fulfilment of the requirements for the award of the degree of

BCA (CLOUD TECHNOLOGY & INFORMATION SECURITY MANAGEMENT)

By
Nandana P S – SB21BCA026
&
Silla Ann Regi – SB21BCA036

III DC BCA (CLOUD TECHNOLOGY & INFORMATION SECURITY MANAGEMENT)

Under the guidance of
Ms Archana Menon P

DEPARTMENT OF BCA (CLOUD TECHNOLOGY & INFORMATION SECURITY MANAGEMENT)

MARCH 2024

FAKE REVIEW DETECTION AND ANALYSIS USING ML AND DL

ST. TERESA'S COLLEGE (AUTONOMOUS)

AFFILIATED TO MAHATMA GANDHI UNIVERSITY

PROJECT REPORT

In partial fulfilment of the requirements for the award of the degree of

BCA (CLOUD TECHNOLOGY & INFORMATION SECURITY MANAGEMENT)

By

Nandana P S – SB21BCA026

&

Silla Ann Regi – SB21BCA036

III DC BCA (CLOUD TECHNOLOGY & INFORMATION SECURITY MANAGEMENT)

Under the guidance of

Ms Archana Menon P

DEPARTMENT OF BCA (CLOUD TECHNOLOGY & INFORMATION SECURITY MANAGEMENT)

MARCH 2024

DECLARATION

We, undersigned, hereby declare that the project report, **“FAKE REVIEW DETECTION AND ANALYSIS USING ML AND DL”**, submitted for partial fulfillment of the requirements for the award of degree of BCA (Cloud Technology & Information Security Management) at St. Teresa’s College (Autonomous), Ernakulam (Affiliated to Mahatma Gandhi University), Kerala, is a bonafide work done by us under the supervision of Ms Archana Menon P. This submission represents our ideas in our own words and where ideas or words of others have not been included. We have adequately and accurately cited and referenced the original sources. We also declare that we have adhered to the ethics of academic honesty and integrity and have not misrepresented or fabricated any data or idea or fact or source in our submission. We understand that any violation of the above will be a cause for disciplinary action by the institute and/or the University and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been obtained. This report has not been previously formed the basis for the award of any degree, diploma or similar title of any other University.

Ernakulam

March 2024

Nandana P S - SB21BCA026

Silla Ann Regi – SB21BCA036

ST. TERESA'S COLLEGE (AUTONOMOUS), ERNAKULAM

BCA (CLOUD TECHNOLOGY & INFORMATION SECURITY MANAGEMENT)

DEPARTMENT OF BCA (CLOUD TECHNOLOGY & INFORMATION SECURITY MANAGEMENT)

THE FEAR OF THE LORD IS THE BEGINNING OF WISDOM

ST. TERESA'S COLLEGE
(AUTONOMOUS)
ESTD 1925

CERTIFICATE

This is to certify that the report entitled **"FAKE REVIEW DETECTION AND ANALYSIS USING ML AND DL"**, submitted by Nandana P S and Silla Ann Regi to the Mahatma Gandhi University in partial fulfillment of the requirements for the award of the Degree of BCA (Cloud Technology and Information Security Management) is a bonafide record of the project work carried out by them under our guidance and supervision. This report in any form has not been submitted to any other University or Institute for any purpose.

ARCHANA MENON P	ARCHANA MENON P	
Internal Supervisor	**Head of the department**	**External Supervisor**

ST. TERESA'S COLLEGE
DEPT. OF COMPUTERS
ERNAKULAM

ACKNOWLEDGEMENT

First and foremost, we thank God Almighty for his blessings. We take this opportunity to express our gratitude to all those who helped us in completing this project successfully. I wish to express our sincere gratitude to the **Manager Rev. Dr. Sr. Vinitha CSST** and the Principal **Dr. Alphonsa Vijaya Joseph** for providing all the facilities.

We express our sincere gratitude towards the Head of the department and our project guide **Ms. Archana Menon P** for her proper guidance and support throughout the project work.

We are indebted to our beloved teachers whose cooperation and suggestion throughout the project which helped us a lot. We thank all our friends and classmates for their support.

We convey our hearty thanks to our parents for the moral support, suggestion and encouragement.

ABSTRACT

With the exponential growth of e-commerce and review-based platforms, the need for reliable methods to detect and mitigate fake reviews has become increasingly urgent especially when people rely on reviews to rate and believe in a products' authenticity and quality. Nowadays not only humans, but trained, dedicated bots are also there to do the fake reviewing. This project investigates the possibility of applying Machine Learning algorithms in fake review detection and analyzes the result. So, the first step is to collect the fake and genuine review text data. There are many fake review datasets available online in sources and repositories like kaggle.com and github.com. The dataset having review text and other details like product category, product type, etc will be passed to an extensive preprocessing type to extract more features from the data and clean the data by removing unwanted samples and features. Natural Language Processing based methods will be used to extract important features like review length. Then the data will be used to train Machine Learning models like KNN (K-Nearest Neighbor), Decision Tree, DNN, Random Forest and XGboost Classifier. The trained models will be evaluated using the performance evaluation metrics like Accuracy, Precision, Recall, F1-score, and confusion matrix. The final model will be chosen based on these metrics and the final model will be tested by inputting a review text. The language of development is python and the development environment is Google Colab. Python libraries like numpy, pandas, and sci-kit learn will be utilized.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATIONS

Sl. No.	ABBREVIATION	FULL FORM
1.	DL	Deep Learning
2.	ML	Machine Learning
3.	CNN	Convolutional Neural Network
4.	RNN	Recurrent Neural Network
5.	LSTM	Long short-term memory
6.	NLP	Natural Language Processing
7.	SVM	Support Vector Machines
8.	LDA	Latent Dirichlet Allocation
9.	NMF	Non negative Matrix Factorization
10.	NER	Named Entity Recognition
11.	BERT	Bidirectional Encoder Representations from Transformers
12.	DNN	Deep Neural Network
13.	GRU	Gated Recurrent Units
14.	GPT	Generative Pre-trained Transformer

Chapter 1
INTRODUCTION

In today's digital age, the prevalence of online reviews has transformed the way consumers make purchasing decisions. From choosing a restaurant for dinner to selecting a hotel for vacation, individuals increasingly rely on the feedback and ratings provided by fellow consumers to guide their choices. However, amidst the wealth of genuine feedback lies a growing concern – the proliferation of fake reviews. These deceptive entries, often crafted to artificially inflate or deflate the reputation of a product, service, or business, pose a significant challenge to the integrity of online review platforms. The detection of fake reviews has emerged as a critical area of research and development, leveraging the power of machine learning (ML) models to sift through vast troves of data and identify suspicious patterns indicative of fraudulent activity.

This session delves into the multifaceted landscape of fake review detection using ML models, exploring its history, the industries affected, the underlying technologies, and the inherent challenges faced by practitioners in this domain. At its core, the detection of fake reviews represents a delicate balance between preserving the authenticity of online feedback and safeguarding consumers against manipulation and deception. As online review platforms continue to serve as indispensable tools for consumer decision-making, the need for effective detection mechanisms has never been more pressing. By harnessing the capabilities of ML models, researchers and practitioners aim to mitigate the adverse impact of fake reviews while fostering trust and transparency within online ecosystems.

Through a comprehensive examination of the historical evolution, industry implications, technological advancements, and persistent challenges surrounding fake review detection, this session endeavors to provide a holistic understanding of this critical domain. By shedding light on the methodologies, strategies, and innovations driving the detection of fake reviews using ML models, participants will gain valuable insights into the complexities and nuances inherent in combating fraudulent activity within online review platforms.

1.1 History of Fake Review Detection

The history of fake review detection is intertwined with the rapid evolution of online commerce and the emergence of digital platforms as primary channels for consumer engagement. As the internet became increasingly ingrained in daily life, so too did the reliance on online reviews as a trusted source of information and guidance. However, with the growing importance of reviews came the inevitable rise of fraudulent practices aimed at manipulating perceptions and influencing consumer behavior.

In the nascent stages of online commerce, the detection of fake reviews was primarily a manual endeavor, relying on human moderators to sift through feedback and identify suspicious entries. These early efforts, while earnest, were often labor-intensive and prone to oversight, as the volume of reviews continued to escalate. As e-commerce platforms gained traction and expanded their reach, the need for more scalable and efficient detection mechanisms became apparent. The advent of machine learning (ML) marked a significant turning point in the evolution of fake review detection, offering a data-driven approach to analyzing and classifying reviews at scale. Leveraging algorithms capable of learning patterns and discerning anomalies within vast datasets, ML models emerged as powerful tools in the fight against fraudulent activity.

Early ML-based approaches focused on extracting features such as sentiment, language usage, and reviewer behavior to distinguish between genuine and fake reviews. With advancements in natural language processing (NLP) techniques, ML models gained the ability to delve deeper into the semantic and contextual nuances of textual data, enabling more nuanced and accurate detection of fake reviews. Sentiment analysis algorithms were refined to detect subtle shifts in tone and sentiment indicative of deceptive intent, while topic modeling techniques allowed for the identification of anomalous topics or keywords associated with fake reviews. As the arms race between fraudsters and detection systems escalated, ML models evolved to incorporate more sophisticated algorithms capable of adapting to evolving tactics and strategies employed by perpetrators of fake reviews. Ensemble learning methods, such as random forests and gradient boosting, were employed to combine the predictive power of multiple models and enhance overall detection accuracy.

Deep learning architectures, including recurrent neural networks (RNNs) and convolutional neural networks (CNNs), offered new avenues for capturing complex linguistic patterns and contextual information. The history of fake review detection is characterized by a constant cycle of innovation and adaptation, as detection systems continue to evolve in response to emerging threats and challenges.

While ML models have significantly advanced the state-of-the-art in fake review detection, the battle against fraudulent activity remains ongoing. As we navigate the intricacies of this dynamic landscape, it is essential to remain vigilant, continuously refining and enhancing detection mechanisms to preserve the integrity of online review platforms and empower consumers to make informed choices.

1.2 Industries Affected by Fake Reviews

The pervasive influence of online reviews extends across a diverse array of industries, shaping consumer perceptions, purchasing decisions, and brand reputations. From e-commerce giants to local businesses, the impact of fake reviews reverberates throughout the digital landscape, posing significant challenges for both consumers and businesses alike.

1. **E-commerce Platforms:**

   E-commerce platforms, such as Amazon, eBay, and Alibaba, serve as hubs for online transactions, offering a vast array of products and services to consumers worldwide. With millions of products listed and countless reviews submitted daily, these platforms represent fertile ground for the proliferation of fake reviews. Whether it's a newly launched product seeking to gain traction or a competitor aiming to undermine a rival's reputation, the incentive to manipulate reviews for financial gain is ever-present. Fake reviews not only distort product ratings but also erode trust in the platform's integrity, potentially driving away customers and tarnishing its reputation.

2. **Hospitality Industry:**

   The hospitality industry, encompassing hotels, restaurants, and travel accommodations, relies heavily on online reviews to attract patrons and differentiate themselves in a crowded marketplace. Positive reviews can serve as powerful endorsements, influencing travelers' decisions and driving bookings. Conversely, negative reviews can have a detrimental impact, dissuading potential guests and undermining the establishment's reputation. In this competitive landscape, the temptation to inflate ratings through fake reviews or disparage competitors through negative feedback is pervasive, leading to an environment rife with deceptive practices and fraudulent activity.

3. **Healthcare Sector:**

   In the healthcare sector, online reviews play a pivotal role in shaping patients' perceptions of medical providers, clinics, and healthcare facilities. From choosing a primary care physician to selecting a specialist for a specific procedure, patients increasingly turn to online feedback to inform their healthcare decisions. However, the integrity of these reviews is often called into question, as fake entries can skew perceptions and mislead patients. Whether it's a clinic seeking to boost its reputation or a disgruntled individual aiming to tarnish a practitioner's standing, the prevalence of fake reviews poses ethical and practical challenges within the healthcare ecosystem.

4. **Technology Products and Services:**

   The technology sector, encompassing gadgets, software, and digital services, is not immune to the scourge of fake reviews. Whether it's a smartphone, a gaming console, or a productivity app, consumers rely on online reviews to gauge the quality, performance, and reliability of tech products and services. However, amidst the sea of feedback lies a minefield of deceptive entries, ranging from sponsored endorsements to malicious attacks aimed at undermining competitors. As technology continues to permeate every aspect of modern life, the need for trustworthy and reliable reviews becomes increasingly critical, making the detection and mitigation of fake reviews a pressing concern for tech companies and consumers alike.

1.3 Technologies Utilized in Fake Review Detection

The detection of fake reviews represents a multifaceted challenge that requires a sophisticated blend of technologies spanning machine learning, natural language processing, and data analytics. As fraudulent tactics evolve and become increasingly sophisticated, the arsenal of tools and techniques used in fake review detection must adapt accordingly. This section explores the diverse array of technologies leveraged in the ongoing battle against deceptive practices within online review platforms.

Machine Learning (ML): At the heart of fake review detection lies the power of machine learning algorithms, capable of learning patterns and discerning anomalies within vast datasets of reviews. Supervised learning techniques, such as support vector machines (SVM), logistic regression, and random forests, are commonly employed to classify reviews as genuine or fake based on a multitude of features. These features may include linguistic characteristics, sentiment analysis scores, reviewer behavior, and temporal patterns. By training ML models on labeled datasets of authentic and fake reviews, these algorithms can learn to distinguish between genuine feedback and fraudulent entries, enabling automated detection at scale.

Natural Language Processing (NLP): Natural language processing techniques play a pivotal role in extracting semantic meaning and sentiment from textual data, enabling deeper analysis of reviews and uncovering subtle cues indicative of fraudulent activity.

Sentiment analysis algorithms are employed to assess the overall tone and sentiment expressed within reviews, identifying discrepancies or inconsistencies that may signal deceptive intent. Additionally, topic modeling techniques, such as latent Dirichlet allocation (LDA) and non-negative matrix factorization (NMF), aid in uncovering anomalous topics or keywords associated with fake reviews.

Named entity recognition (NER) algorithms further enhance detection by identifying entities, such as product names or company identifiers, that may be manipulated or misrepresented in fraudulent entries.

Deep Learning: Deep learning architectures, including recurrent neural networks (RNNs), convolutional neural networks (CNNs), and transformer models like BERT (Bidirectional Encoder Representations from Transformers), have shown remarkable promise in fake review detection by capturing intricate linguistic patterns and contextual information.

RNNs, with their ability to model sequential data, are well-suited for tasks such as detecting temporal patterns and identifying anomalies in review sequences.

CNNs excel at extracting hierarchical features from text, enabling them to discern subtle linguistic nuances and distinguish between genuine and fake reviews.

Transformer models, renowned for their ability to capture long-range dependencies and contextual relationships, offer unparalleled performance in semantic understanding and sentiment analysis, further enhancing the accuracy and robustness of fake review detection systems.

Ensemble Learning and Model Fusion: Ensemble learning techniques, such as bagging, boosting, and stacking, are employed to combine the predictive power of multiple base models, mitigating the risk of overfitting and enhancing overall detection accuracy. By aggregating predictions from diverse models trained on different subsets of data or using distinct algorithms, ensemble methods can effectively capture complementary patterns and improve the robustness of fake review detection systems.

Model fusion approaches further enhance performance by integrating predictions from multiple modalities, such as textual features, metadata, and reviewer behavior, into a unified framework. By leveraging the strengths of each individual model or feature representation, model fusion techniques enable more comprehensive and nuanced detection of fake reviews, enhancing the reliability and efficacy of detection systems.

Chapter 2
LITERATURE SURVEY

The literature survey has been conducted on different papers such as technical papers and review papers in the domain published in leading publications, journals and conferences. Keywords such as fake review detection, machine learning, deep learning etc are used to filter out.

[1] identified 3 types of spam Type1- Untruthful reviews(fake reviews) Type2-Review on brands (targeting on different brands and are not product specific) Type3- Non-reviews(contains advertisements and irrelevant links). Amazon reviews were detected as duplicate and non-duplicate reviews Spam reviews of type2 and type3 are detected based on traditional classification learning as these two can be detected manually. Their research found that there are a large number of duplicate reviews on the same product or the same review on different products. These are written by a group of users to create impact and thus must contain Type1 reviews. These were detected by identifying (i) Duplicates from different users on the same product.(ii) Duplicates from the same user id on different products (iii) Duplicates from different users on different products by calculating similarity scores of two reviews. They used logistic regression to find the probability of each review to be spam. Three features that are the content of the review, the reviewer who wrote the review and the product being reviewed are used for training data for the regression model. Spammers groups influence a majority of customers because of their size thus detection of these groups is equally important.

Focusing on the issue [6] worked on detection of Spammers Groups instead of spotting fake reviews alone. A group of reviewers who work collaboratively to write fake reviews take control of the sentiment of a particular product influencing customer because of their size. The proposed method focuses on detection of these groups along with fake reviews. Candidate groups are identified using frequent itemset mining. These candidate groups are ranked based on their likelihoods for being spam called as GS Rank.GS Rank is based on Group content similarity, Group member content similarity, Group early time frame and labeling given to each group. Spamming behavior indicator is based on both group and individual behavior indicators. For each feature belonging to a group or individual Statistical validation is done. Behavioral distribution of each group is used to identify the spammers groups. The proposed technique does not use from the traditional supervised learning approach for spam detection because of the inherent nature of problem which makes the classic supervised learning approach less effective. Experimental results show that the proposed method outperforms various detection techniques like supervised classification, regression, and learning to rank algorithms but it is a very time consuming task.

[7] proposed a method for the simultaneous detection of fake reviews and fakers .Factor Graph method incorporates features of reviews and reviewers which were divided into categories (i) review related (ii) reviewer related (iii) features between reviewers and reviews (iv) review group features based on the classification the local factor(probability of fakeness).Based on the these features three factors are defined namely local feature factor(find the probability of a reviewer or review to be fake) , group domain factor(using review group rating to measure the reliability of all the rating given to a particular product) and cross domain factor(relationship between reviewer and his product). All the review information is transformed into the Review graph method. This Review Graph model containing all the features is used to create a Model. Model learning and inference of the model help to detect the fake reviews and reviewers in a united framework. Experimental results show that method outperforms all of the other baseline methods significantly with respect to both efficiency and accuracy.

In the same year [4] proposed a method for automatic review spam detection of fake reviews. In addition to some previous criteria like link in the reviews, all capital reviews, product and companies comparison one new criteria was introduced by taking in account the rating given to the product by the user. The sentiment of text is compared against the rating provided by the customers. The numerical value of sentiment of text is produced by using existing sentiment analysis tools like PhpInsight and Alchemy and is compared with the rating provided by the customer for consistency. PhpInsight is a Bayesian classifier that classifies the words of a dictionary as positive, negative and neutral. Alchemy provides natural processing tools like tagging, topic categorisation, and language detection. In short it uses machine learning techniques to analyze the content of the review. The numerical value produced by sentiment analysis the tool was compared with ratings of the review provided by the user and inconsistency between the two indicates a possibility of review spam.

[3] addressed the issue by focussing on the lexical and the syntactic features. Lexical features are used to analyze each word in the review. POS is used as a lexical feature. Type-Token ratio is calculated which is the ratio of number of words of type adjective, noun, verb and adverb to the total number of words in the review. This is used to calculate lexical diversity. The purpose of this is to calculate the review complexity. Syntactic

development for language learners is provided by syntactic features. Syntactic complexity is calculated by the number of clauses and one main clause with some subservient clause joined together known as T-unit. Researchers used 16 new lexical and 25 new syntactic features. Features value is calculated one by one using koRpus (R package) and used as training data for classification of further reviews. Apart from all the reviews posted Negative reviews have a huge impact on the customers thus it is important to detect authentic and manipulative negative reviews.

Analyses of authentic and manipulative reviews by [3] showed review readability, review genre and review writing style can be used to distinguish between the two kinds of reviews. These types of reviews are written with different purposes which can be reflected in their content. Manipulative review writers use sophisticated language for the review so that most of the customers can follow them. Too simplistic reviews attract less people. Readability can be used to differentiate between the fake and true reviews. Authentic reviews are based on real events while the manipulative therefore are considered to be imaginative and authentic are considered to be informative out of the genre of text (conversational, text-oriented, informative and imaginative). Four genres of text are conversational, text-oriented Manipulative reviews containing different distributions of adjectives, nouns, verbs and adverbs than authentic reviews. Third, the writing style of both types of review writers is different. Manipulative reviews contain more affective cues (mainly negative affective cues), perceptual words and they also use more future tense to recommend to other customers [6]

Chapter 3
EXISTING SYSTEM

Existing fake review detection systems encompass a variety of approaches.

3.1 Traditional Heuristic-based Approaches

In the early stages of combating fake reviews, traditional heuristic-based approaches were the primary means of detection. These methods relied on predefined rules, heuristics, and manual moderation techniques to identify suspicious patterns indicative of fraudulent activity within online reviews. While these approaches served as foundational frameworks for fake review detection, they often lacked scalability, adaptability, and accuracy compared to more advanced machine learning-based techniques.

Rule-based Systems: Rule-based systems were among the earliest forms of fake review detection mechanisms, employing predefined rules and criteria to flag reviews exhibiting suspicious characteristics. These rules typically targeted specific attributes such as review length, frequency of keywords, and temporal patterns, aiming to identify deviations from expected norms. While rule-based systems offered simplicity and transparency in their operation, they were often prone to false positives and struggled to adapt to evolving tactics employed by perpetrators of fake reviews.

Keyword Analysis: Keyword analysis techniques focused on identifying specific keywords or phrases commonly associated with fake reviews, such as "amazing," "terrible," or "discount." By analyzing the frequency and distribution of these keywords within reviews, detection systems could flag entries exhibiting an overabundance of positive or negative language as potentially fraudulent. However, keyword-based approaches were limited in their effectiveness, as they failed to capture the nuanced linguistic patterns and contextual cues characteristic of deceptive reviews.

Manual Moderation: Manual moderation involved human reviewers manually inspecting individual reviews to assess their authenticity and credibility. While manual moderation offered the advantage of human judgment and intuition, it was highly labor-intensive, time-consuming, and susceptible to bias. Moreover, manual moderation was impractical for platforms with large volumes of reviews, making it unfeasible as a scalable solution for fake review detection. Despite their limitations, traditional heuristic-based approaches played a foundational role in shaping the evolution of fake review detection methodologies. These early techniques laid the groundwork for more sophisticated machine learning-based models, providing valuable insights into the characteristics and patterns of fake reviews. While modern detection systems have largely supplanted traditional heuristic-based approaches, the principles underlying these methods continue to inform the development of more advanced detection algorithms.

In conclusion, traditional heuristic-based approaches represented the first line of defense against fake reviews, leveraging predefined rules, keyword analysis, and manual moderation techniques to identify suspicious patterns within online feedback. While these methods offered simplicity and transparency, they were often limited in scalability, adaptability, and accuracy compared to more advanced machine learning-based techniques. Nonetheless, the insights gained from traditional approaches have been instrumental in shaping the development of more sophisticated detection systems capable of effectively combating fraudulent activity within online review platforms.

3.2 Natural Language Processing (NLP) Techniques

Natural Language Processing (NLP) Techniques: Natural Language Processing (NLP) techniques have emerged as a pivotal tool in the detection of fake reviews, enabling automated analysis of textual data to uncover semantic meaning, sentiment polarity, and linguistic patterns indicative of deceptive behavior. In this section, we explore the diverse array of NLP techniques utilized in fake review detection, highlighting their role in enhancing the accuracy and robustness of detection systems.

3.2.1 Sentiment Analysis

Sentiment analysis, also known as opinion mining, is a fundamental NLP technique employed in fake review detection to assess the overall sentiment expressed within textual data. By analyzing the polarity of words, phrases, and sentences, sentiment analysis algorithms classify reviews as positive, negative, or neutral, enabling the identification of deceptive sentiment patterns indicative of fake reviews. For example, excessively positive language in a review may raise suspicion, especially if it contrasts with the overall sentiment of other reviews for the same product or service. Sentiment analysis techniques leverage lexicons, machine learning models, and linguistic rules to infer sentiment polarity accurately, enabling detection systems to flag reviews exhibiting anomalous sentiment distributions.

3.2.2 Named Entity Recognition (NER)

Named Entity Recognition (NER) is another critical NLP technique utilized in fake review detection to identify entities such as product names, brand mentions, and company identifiers within textual data. By extracting named entities from reviews, detection systems can analyze the context in which products or services are mentioned, enabling the identification of suspicious patterns or inconsistencies. For example, discrepancies between the named entity mentioned in the review and the product or service being reviewed may indicate deceptive behavior. NER algorithms employ machine learning models, pattern matching techniques, and linguistic rules to accurately identify named entities within textual data, facilitating more nuanced analysis and detection of fake reviews.

3.2.3 Topic Modeling

Topic modeling techniques, such as Latent Dirichlet Allocation (LDA) and Non-negative Matrix Factorization (NMF), are employed in fake review detection to uncover latent topics or themes present within textual data. By clustering reviews based on shared topics or keywords, topic modeling algorithms enable detection systems to identify anomalous topics or clusters associated with fake reviews. For example, reviews discussing unrelated topics or diverging from the dominant themes may raise suspicion, indicating potential manipulation or fraudulent behavior. Topic modeling techniques leverage probabilistic graphical models and matrix factorization algorithms to decompose textual data into latent topics, facilitating more granular analysis and detection of deceptive reviews.

3.2.4 Contextual Analysis

Contextual analysis techniques focus on understanding the context in which reviews are written, enabling detection systems to capture subtle linguistic nuances, sarcasm, and cultural references indicative of deceptive behavior. Contextual analysis encompasses a range of NLP techniques, including syntactic parsing, semantic role labeling, and discourse analysis, aimed at unraveling the complex interplay of linguistic elements within textual data. By analyzing the context surrounding reviews, detection systems can discern genuine feedback from deceptive entries, enhancing detection accuracy and reliability. Contextual analysis techniques leverage deep learning models, neural network architectures, and pre-trained language representations to capture contextual information effectively, enabling detection systems to adapt to diverse linguistic contexts and detect sophisticated forms of deception.

3.3 Deep Learning Architectures

Deep learning architectures have revolutionized fake review detection by enabling the extraction of intricate linguistic patterns, contextual information, and semantic meaning from textual data. In this section, we delve into the diverse array of deep learning models utilized in fake review detection, highlighting their capabilities, advantages, and applications in combating fraudulent activity within online review platforms.

3.3.1 Recurrent Neural Networks (RNNs)

Recurrent Neural Networks (RNNs) are a class of deep learning architectures commonly employed in fake review detection for their ability to model sequential data and capture temporal dependencies within textual sequences. In fake review detection, RNNs are utilized to analyze the sequential structure of reviews, enabling detection systems to uncover patterns of deceptive behavior over time. For example, RNNs can identify subtle shifts in sentiment or linguistic style within reviews, indicating potential manipulation or fraudulent activity. Despite their effectiveness in capturing sequential dependencies, traditional RNNs suffer from limitations such as vanishing gradients and difficulty in modeling long-range dependencies, leading to challenges in detecting subtle linguistic nuances and contextual information.

3.3.2 Long Short-Term Memory (LSTM) Networks

Long Short-Term Memory (LSTM) networks are a variant of RNNs designed to address the vanishing gradient problem and capture long-range dependencies within sequential data. In fake review detection, LSTM networks offer improved performance and robustness compared to traditional RNNs, enabling more accurate modeling of linguistic patterns and contextual information. LSTM networks excel at capturing subtle shifts in sentiment, semantic meaning, and syntactic structure within reviews, enabling detection systems to discern genuine feedback from deceptive entries. By leveraging memory cells and gating mechanisms, LSTM networks can effectively retain important information over extended sequences, facilitating more nuanced analysis and detection of fraudulent activity.

3.3.3 Gated Recurrent Units (GRUs)

Gated Recurrent Units (GRUs) are another variant of RNNs that address the vanishing gradient problem and improve the modeling of long-range dependencies within sequential data. GRUs offer similar capabilities to LSTM networks but with a simpler architecture and fewer parameters, making them more computationally efficient and easier to train. In fake review detection, GRUs are utilized to analyze the sequential structure of reviews and capture temporal dependencies, enabling detection systems to identify suspicious patterns indicative of deceptive behavior. Despite their simpler architecture, GRUs have been shown to achieve comparable performance to LSTM networks in various NLP tasks, making them a popular choice for fake review detection systems.

3.3.4 Convolutional Neural Networks (CNNs)

Convolutional Neural Networks (CNNs) are a class of deep learning architectures primarily used in image processing tasks but have also been adapted for text analysis tasks in fake review detection. In fake review detection, CNNs are applied to analyze the local structure and syntactic features of textual data, enabling detection systems to capture patterns of deceptive behavior within reviews. CNNs employ convolutional layers and pooling operations to extract hierarchical features from text, enabling them to discern subtle linguistic nuances and contextual cues indicative of fake reviews. By leveraging pre-trained word embeddings and attention mechanisms, CNNs can effectively capture important information and patterns within textual data, facilitating more accurate and robust detection of fraudulent activity.

3.3.5 Transformer Models

Transformer models, such as BERT (Bidirectional Encoder Representations from Transformers) and GPT (Generative Pre-trained Transformer), represent the state-of-the-art in fake review detection by capturing complex linguistic patterns, contextual information, and semantic meaning from textual data. Transformer models leverage self-attention mechanisms and multi-layer architectures to capture long-range dependencies and contextual relationships within reviews, enabling detection systems to discern subtle linguistic cues and detect sophisticated forms of deception. By pre-training on large corpora of text data and fine-tuning on specific fake review detection tasks, transformer models can achieve unparalleled performance and robustness, making them the preferred choice for many fake review detection systems.

3.4 Spammer Group Detection Algorithm

Malicious sellers often team up with fake reviewers (spammers) to manipulate online reviews across different products. This creates an unfair advantage for their products and harms competition. Existing detection methods struggle to identify these coordinated attacks because they don't consider attacks spanning multiple products (cross-product attacks), don't analyze all available data, including network structure, reviewer attributes, and the relationship between them. They have proposed a collaborative training-based algorithm to detect spammer groups in online review networks.

The working goes like:

1. Targeted Network: We focus on a specific set of products potentially targeted by spammers and build a smaller network around them (induced sub-network). This allows us to identify coordinated attacks across these products.
2. Collaborative Feature Learning: We leverage a collaborative training method to analyze all the data available in the network. This includes the connections between users (structure), user attributes (reviewer information), and how these features influence each other (structure-attribute correlation).
3. Spammer Group Detection: We use a clustering method (DBSCAN) to identify groups of users with similar behavior, potentially indicating spammers. We then refine these groups by removing unlikely spammers and ranking the remaining ones based on their suspicious activity.

Benefits include detecting cross-product attacks by spammer groups, analyzing all available data (structure, attributes, and their relationship) for better accuracy, achieving superior performance compared to existing methods based on real-world data.This approach offers a more comprehensive and effective way to identify and combat coordinated manipulation of online reviews by spammer groups.

The paper acknowledges a few limitations of the SGDCTH method for detecting spammer groups in online reviews:

1. Manual Thresholds for DBSCAN Clustering: The DBSCAN clustering method used by SGDCTH relies on two manually defined thresholds to identify groups of potential spammers. This introduces subjectivity and requires human intervention to determine the appropriate values. Ideally, the algorithm would learn these thresholds automatically to avoid human bias and adapt to different datasets.
2. Limited Efficiency for Large Networks: While SGDCTH reduces time complexity by focusing on a targeted product set, the collaborative training method used to learn node representations might become computationally expensive for very large or dense networks. The paper suggests exploring more efficient methods for learning node features in such scenarios.
3. Lack of Simulated Attack Patterns: The evaluation of SGDCTH relies on real-world datasets. However, the paper acknowledges that simulating the ever-evolving attack patterns of spammer groups could be beneficial. Including such simulated data in the evaluation process could provide a more robust assessment of the method's effectiveness against real-world threats.

Chapter 4
PROPOSED SYSTEM

The proposed system aims to address the pervasive issue of fake reviews within online platforms by leveraging machine learning models, specifically XGBoost, to effectively detect and mitigate fraudulent activity. Building upon a comprehensive dataset of Amazon product reviews, which has been preprocessed and cleaned to ensure data quality and consistency, the proposed system harnesses the power of advanced ML techniques to discern genuine feedback from deceptive entries.

1. Data Collection and Preprocessing.

The foundation of the proposed system lies in the meticulous collection and preprocessing of Amazon product review data. By curating a diverse dataset comprising both genuine and fake reviews, we ensure the robustness and generalizability of the detection model. The collected data undergoes rigorous preprocessing steps, including text normalization, tokenization, and removal of stopwords and irrelevant characters, to standardize the format and ensure uniformity across reviews.

2. Feature Engineering.

Feature engineering plays a crucial role in enhancing the predictive power of machine learning models. In the proposed system, a comprehensive set of features is extracted from the preprocessed review data to capture various aspects of linguistic, semantic, and contextual information. These features may include word frequency counts, sentiment scores, syntactic patterns, and reviewer metadata, providing rich inputs for the detection model to learn discriminative patterns between genuine and fake reviews.

3. Machine Learning Models.

The proposed system leverages two primary machine learning models for fake review detection: XGBoost and Deep Neural Networks (DNNs). XGBoost, a powerful gradient boosting algorithm, has been identified through empirical evaluation as the top-performing model for fake review detection tasks. Its ability to handle imbalanced datasets, capture complex nonlinear relationships, and provide interpretable insights makes it an ideal choice for our detection framework. Additionally, DNNs, with their capacity to learn hierarchical representations and capture intricate linguistic patterns, serve as complementary models to further enhance detection accuracy and robustness.

4. Model Training and Evaluation.

The machine learning models are trained on the preprocessed dataset using a carefully designed training pipeline. During the training phase, the models learn to distinguish between genuine and fake reviews by optimizing objective functions such as log loss or area under the receiver operating characteristic curve. Cross-validation techniques, such as k-fold cross-validation, are employed to assess model performance and mitigate overfitting. The trained models are then evaluated on a separate validation set to assess their generalization ability and effectiveness in detecting fraudulent activity.

5. Performance Evaluation Metrics.

To quantify the performance of the proposed system, a set of comprehensive evaluation metrics is employed. These metrics include precision, recall, F1-score and accuracy, providing a holistic assessment of detection performance across different performance criteria. By evaluating the models on multiple metrics, we gain a nuanced understanding of their strengths, limitations, and areas for improvement, enabling iterative refinement and optimization of the detection framework.

Chapter 5
SYSTEM REQUIREMENTS

The system requirement is not that much for this project as the training has been carried out in Google Colab free version. So, no specific hardware is required. The design and development of the whole system of image processing and deep learning have been carried out in Google Colab Cloud platform.

5.1 Hardware requirement

Basic system with intel i3 or above processor.

5.2 Software requirement

IDE used for ML development and training - Google Colab.

Language used for ML development and training - Python 3.7 - 3.11

In addition to this various python libraries like Tensorflow for deep learning are also used.

Chapter 6
SYSTEM DESIGN AND ARCHITECTURE

The system design shows the data collection and the long process after that which is the data analysis, feature engineering and preprocessing steps. The models like XGBoost, K-Nearest Neighbour, Deep Neural Networks, Decision Tree, Random Forest are used as the models here.

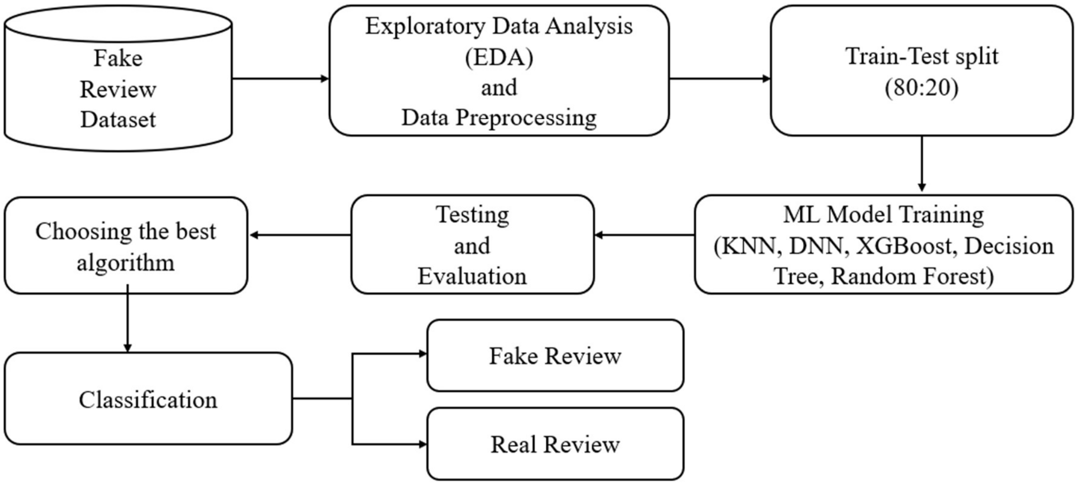

Fake Review Dataset
Exploratory Data Analysis (EDA) and Data Preprocessing
Train-Test split (80:20)
ML Model Training (KNN, DNN, XGBoost, Decision Tree, Random Forest)
Testing and Evaluation
Choosing the best algorithm
Classification
Fake Review
Real Review

Figure 6.1 : Proposed System Flow Diagram

Data preprocessing is a crucial step in preparing the fake review dataset for machine learning model training. It involves cleaning and transforming the raw textual data into a format suitable for feature extraction and subsequent analysis.

The following technical steps are typically involved in data preprocessing:

1. **Text Normalization**
   - Convert text to lowercase: This ensures consistency in the representation of words regardless of their capitalization.
   - Remove punctuation marks: Punctuation marks such as commas, periods, and exclamation marks are removed as they do not contribute to the semantic meaning of the text.

- Handle special characters: Special characters such as emojis, HTML tags, and non-alphanumeric symbols are handled appropriately, either by removing them or replacing them with appropriate representations.

2. **Tokenization**
   - Split text into tokens: The text is split into individual tokens, typically words or subwords, using whitespace or other delimiters.
   - Handle contractions: Contractions such as "don't" are split into their constituent parts ("do" and "n't") to ensure proper tokenization.

3. **Stopwords Removal**
   - Remove common stopwords: Stopwords such as "and," "the," "is," etc., which occur frequently in the English language but carry little semantic meaning, are removed from the text.

4. **Stemming and Lemmatization**
   - Stemming: Reduce words to their root form by removing suffixes. For example, "running" becomes "run."
   - Lemmatization: Convert words to their base or dictionary form. For example, "better" becomes "good."

5. **Spell Checking and Noise Removal**
   - Correct spelling errors: Spelling errors are identified and corrected using algorithms such as Levenshtein distance or spell-check dictionaries.
   - Remove noise: Non-standard characters, gibberish, or irrelevant information are removed to improve the quality and readability of the text.

6. **Normalization and Standardization**
   - Normalize numerical values: If the dataset contains numerical features, they may be normalized to a standard scale (e.g., between 0 and 1) to prevent biases in model training.
   - Standardize text encoding: Ensure consistent text encoding (e.g., UTF-8) across the dataset to avoid encoding-related errors during processing.

6.1 XGBoost Classifier

Ever since its introduction in 2014, XGBoost has been lauded as the holy grail of machine learning hackathons and competitions. From predicting ad click-through rates to classifying high energy physics events, XGBoost has proved its mettle in terms of performance – and speed. The beauty of this powerful algorithm lies in its scalability, which drives fast learning through parallel and distributed computing and offers efficient memory usage.

XGBoost is an ensemble learning method. Sometimes, it may not be sufficient to rely upon the results of just one machine learning model. Ensemble learning offers a systematic solution to combine the predictive power of multiple learners. The resultant is a single model which gives the aggregated output from several models. The models that form the ensemble, also known as base learners, could be either from the same learning algorithm or different learning algorithms. Bagging and boosting are two widely used ensemble learners. Though these two techniques can be used with several statistical models, the most predominant usage has been with decision trees.

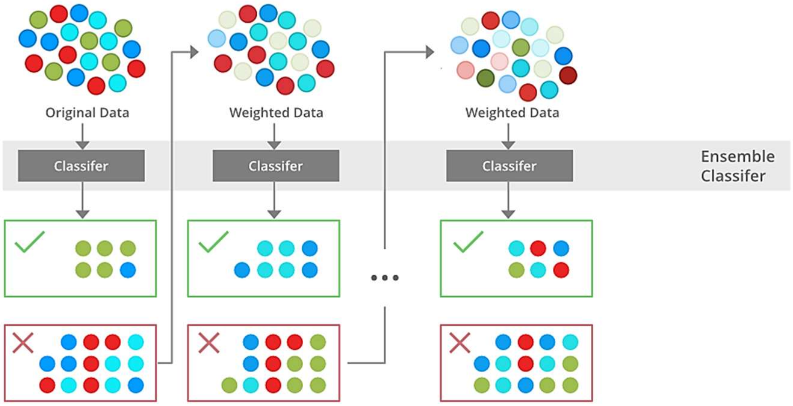


Figure 6.2 : XGBoost Classifier

Unique features of XGBoost

- Regularization: XGBoost has an option to penalize complex models through both L1 and L2 regularization. Regularization helps in preventing overfitting
- Handling sparse data: Missing values or data processing steps like one-hot encoding make data sparse. XGBoost incorporates a sparsity-aware split finding algorithm to handle different types of sparsity patterns in the data
- Weighted quantile sketch: Most existing tree based algorithms can find the split points when the data points are of equal weights (using quantile sketch algorithm). However, they

are not equipped to handle weighted data. XGBoost has a distributed weighted quantile sketch algorithm to effectively handle weighted data

- Block structure for parallel learning: For faster computing, XGBoost can make use of multiple cores on the CPU. This is possible because of a block structure in its system design. Data is sorted and stored in in-memory units called blocks. Unlike other algorithms, this enables the data layout to be reused by subsequent iterations, instead of computing it again. This feature also serves useful for steps like split finding and column sub-sampling
- Cache awareness: In XGBoost, non-contiguous memory access is required to get the gradient statistics by row index. Hence, XGBoost has been designed to make optimal use of hardware. This is done by allocating internal buffers in each thread, where the gradient statistics can be stored
- Out-of-core computing: This feature optimizes the available disk space and maximizes its usage when handling huge datasets that do not fit into memory

6.2 Deep Neural Network (DNN)

A deep neural network (DNN) is an artificial neural network (ANN) with multiple layers between the input and output layers. The DNN finds the correct mathematical manipulation to turn the input into the output, whether it be a linear relationship or a non-linear relationship. The network moves through the layers calculating the probability of each output. For example, a DNN that is trained to recognize dog breeds will go over the given image and calculate the probability that the dog in the image is a certain breed.

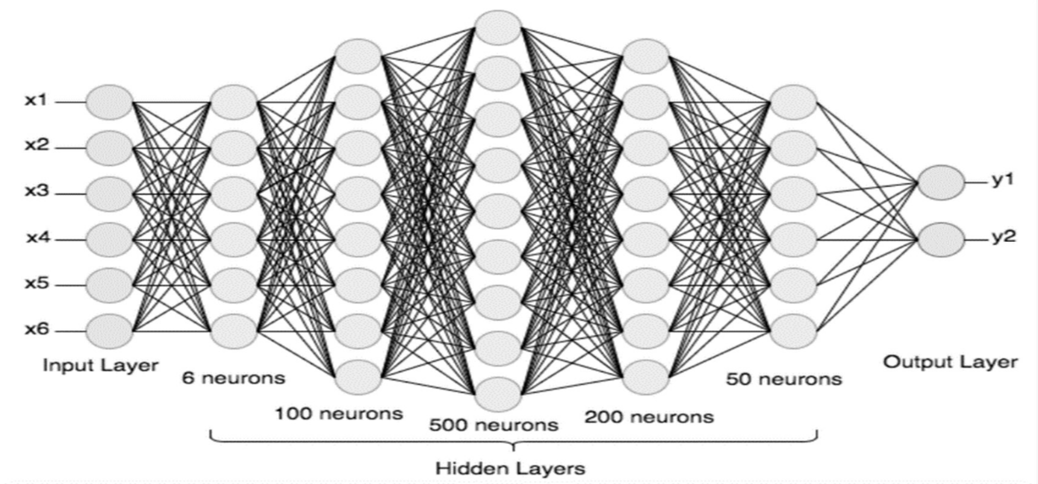


Figure 6.3 : DNN

DNNs are typically feedforward networks in which data flows from the input layer to the output layer without looping back. At first, the DNN creates a map of virtual neurons and assigns random numerical values, or “weights”, to connections between them. The weights and inputs are multiplied and return an output between 0 and 1. If the network didn’t accurately recognize a particular pattern, an algorithm would adjust the weights. That way the algorithm can make certain parameters more influential, until it determines the correct mathematical manipulation to fully process the data.

6.3 K-Nearest Neighbors (KNN) Algorithm.

The K-Nearest Neighbors (KNN) algorithm is a versatile and intuitive classification and regression technique used in machine learning for pattern recognition and predictive modeling tasks. It belongs to the family of instance-based or lazy learning algorithms, where the model does not explicitly learn a mapping from input features to outputs during training, but rather memorizes the training dataset and makes predictions based on the similarity between new data points and existing training instances. In this comprehensive discussion, we explore the technical intricacies of the KNN algorithm, including its principles, implementation, advantages, limitations, and practical considerations.

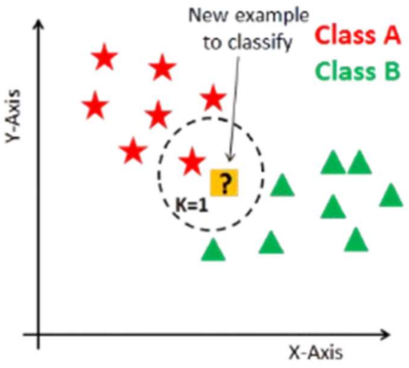


Figure 6.4 : KNN

6.3.1. Principles of KNN

The fundamental principle behind the KNN algorithm is based on the assumption that similar instances in the feature space are likely to belong to the same class or exhibit similar behaviors. Given a new, unlabeled data point, KNN identifies its K nearest neighbors in the training dataset based on a chosen distance metric (e.g., Euclidean distance, Manhattan distance, cosine similarity). The majority class or average of the target values among these neighbors is then assigned to the new data point as its predicted class label or regression value.

6.3.2. Technical Details

a. Training Phase: During the training phase of KNN, the algorithm simply stores the feature vectors and corresponding class labels (for classification) or target values (for regression) of the training dataset. No explicit model is built or parameters are learned during this phase, making KNN computationally inexpensive and memory-efficient.

b. Prediction Phase: In the prediction phase, when presented with a new data point, KNN calculates the distances between the new point and all training instances using the chosen distance metric. It then selects the K nearest neighbors based on these distances and aggregates their class labels (for classification) or target values (for regression) to make predictions for the new data point.

c. Choice of K: The choice of the parameter K, representing the number of neighbors considered, is a critical decision in KNN. A small value of K may lead to overfitting and increased sensitivity to noise, while a large value of K may result in underfitting and decreased model flexibility. The optimal value of K is often determined through cross-validation or grid search techniques.

d. Distance Metrics: Various distance metrics can be used in KNN to quantify the similarity between data points. Common distance metrics include Euclidean distance, Manhattan distance, Minkowski distance, and cosine similarity. The choice of distance metric depends on the nature of the data and the underlying problem domain.

6.4 Decision Tree

A decision tree algorithm is a machine learning method used for both classification and regression tasks. It works by creating a tree-like model of decisions and their possible consequences.

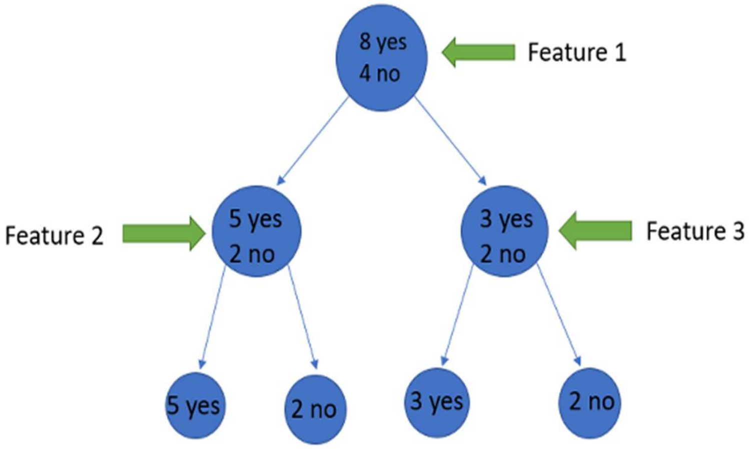


Figure 6.5 : Decision Tree

6.4.1 Structure

The tree consists of nodes (decision points) and branches. There are two main types of nodes:

- Internal nodes (decision nodes): These contain conditions or questions that split the data into subsets based on the answer (Yes/No or multiple choices).
- Leaf nodes (terminal nodes): These represent the final outcome or prediction for a particular data point.

6.4.2 Training

The algorithm learns by iteratively splitting the data based on the most useful feature (attribute) at each node. The goal is to create a tree that accurately separates the data into its different categories (classification) or predicts a continuous value (regression).

6.4.3 Prediction

Once trained, the decision tree can be used to make predictions on new data. A new data point travels down the tree, answering the questions at each node until it reaches a leaf node, which provides the predicted outcome.

6.4.4 Advantages of decision trees

- Easy to interpret: The tree structure offers a clear visualization of the decision-making process.
- No need for feature scaling: Decision trees work well with both categorical and numerical features without requiring them to be on the same scale.
- Can handle missing data: The algorithm can handle missing values in the data by choosing the most appropriate branch based on the available information.

Decision trees are a fundamental machine learning algorithm with a wide range of applications, from spam filtering and fraud detection to medical diagnosis and customer churn prediction.

6.5 Random Forest

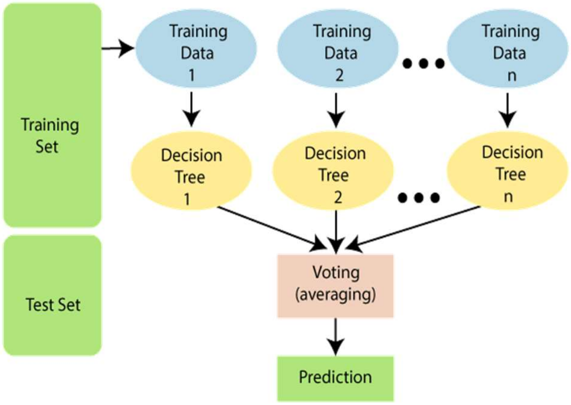


Figure 6.6 : Random Forest

Random forest is a powerful machine learning algorithm that falls under the category of ensemble learning. Unlike a single decision tree, it combines multiple decision trees to create a more robust and accurate model.

6.5.1 Working of Random Forest

1. Random Sampling with Replacement (Bootstrapping):

- The algorithm creates multiple random subsets of data (with replacement) from the original dataset.
- This means a data point can be included in a single subset multiple times.
- These subsets are called bootstrap samples.

2. Building Decision Trees:

- For each bootstrap sample, a decision tree is constructed.

- At each node of the tree, a random subset of features (instead of considering all features) is chosen as candidates for splitting the data.
- The best split among these features is selected based on an impurity measure (e.g., Gini index for classification or variance for regression). This randomness in feature selection helps prevent overfitting.

3. Prediction:

When a new data point arrives:

- It's passed through all the decision trees in the forest.
- Each tree makes a prediction based on its learned model.
- For classification problems, the final prediction is the most voted class from all the trees.
- For regression problems, the final prediction is the average of the predictions from all the trees.

6.5.2 Advantages

- High Accuracy: By combining multiple trees, random forest reduces the variance of individual trees, leading to more accurate predictions.
- Robust to Overfitting: Introducing randomness in feature selection and using bootstrap samples help prevent the model from memorizing the training data and generalizing poorly to unseen data.
- Can handle both Classification and Regression problems.
- Provides Feature Importance: It can tell you which features were most influential in making predictions, offering insights into the data.

Overall, random forest is a versatile and powerful machine learning algorithm known for its accuracy and robustness. It's widely used in various domains like finance, healthcare, and marketing for tasks like fraud detection, customer churn prediction, and image recognition.

Chapter 7
MODULE DESCRIPTION

The project aimed at developing a robust fake review detection system is divided into three fundamental modules, each serving a crucial role in the overall process. This comprehensive approach ensures that the system is meticulously designed, trained, and evaluated to effectively combat fraudulent activities within online review platforms, particularly focusing on the Amazon fake review dataset.

7.1. MODULE 1 - DATA COLLECTION AND PREPARATION MODULE

The Data Collection and Preparation Module serves as the cornerstone of the fake review detection system, encompassing critical steps from the acquisition of the Amazon fake review dataset to its meticulous preprocessing and feature extraction. This module lays the groundwork for subsequent machine learning model development and evaluation, ensuring that the dataset is well-curated, standardized, and optimized for effective training.

7.1.1. Data Collection

The data collection process begins with the acquisition of the Amazon fake review dataset from reliable sources, ensuring its authenticity and relevance to the task at hand. Various sources may be utilized, including publicly available datasets, research repositories, or proprietary data sources. Careful consideration is given to the diversity and representativeness of the dataset, encompassing reviews from a wide range of product categories, time periods, and demographics to capture the full spectrum of fake review characteristics.

7.1.2. Data Analysis

Once the dataset is acquired, it undergoes comprehensive analysis to gain insights into its distribution, structure, and content. Exploratory data analysis techniques are employed to visualize key statistics, such as review lengths, ratings distribution, and class balance between genuine and fake reviews. This analysis helps identify potential biases, anomalies, or data quality issues that may impact subsequent preprocessing and model training steps.

7.1.3. Data Preprocessing

Data preprocessing is a crucial step in ensuring the quality and consistency of the dataset for effective model training. Several preprocessing techniques are applied to clean and standardize the textual data, including:

- **Text Normalization**: Standardizing text by converting uppercase letters to lowercase, removing punctuation marks, and handling special characters to ensure uniformity across reviews.
- **Tokenization**: Segmenting text into individual tokens or words to facilitate subsequent feature extraction and analysis.
- **Stopwords Removal**: Filtering out common stopwords (e.g., "and," "the," "is") that do not contribute to the overall meaning or sentiment of the reviews.
- **Stemming and Lemmatization**: Reducing words to their root forms (e.g., "running" to "run") to improve consistency and reduce feature dimensionality.

7.1.4. Feature Extraction

Feature extraction plays a crucial role in capturing informative signals from the textual data, enabling the machine learning models to discern patterns and relationships indicative of fake reviews. A diverse set of features is extracted from the preprocessed dataset, including:

- **Bag-of-Words (BoW):** Representing reviews as vectors of word frequencies or occurrences to capture the distribution of terms across documents.
- **TF-IDF (Term Frequency-Inverse Document Frequency):** Weighting words based on their frequency in the document and inverse frequency across the entire corpus to prioritize informative terms.
- **Word Embeddings:** Transforming words into dense, low-dimensional vectors that capture semantic meaning and contextual relationships.
- **Syntactic and Semantic Features:** Extracting linguistic features such as part-of-speech tags, named entities, sentiment scores, and syntactic structures to capture higher-level linguistic patterns.

These extracted features serve as input to the machine learning models in the subsequent training phase, enabling them to learn discriminative patterns between genuine and fake reviews effectively.

By meticulously executing each step of the Data Collection and Preparation Module, the fake review detection system ensures that the dataset is well-curated, standardized, and optimized for subsequent model development and evaluation. This structured approach lays the foundation for building accurate, reliable, and scalable machine learning models capable of effectively identifying and mitigating fraudulent activities within online review platforms.

7.2. MODULE 2 - MODEL DEVELOPMENT AND TRAINING

The machine learning module focuses on the development and training of machine learning models, including XGBoost, Deep Neural Networks (DNNs), K-Nearest Neighbour (KNN), Decision Tree and Random Forest for fake review detection. This module involves the following steps:

7.2.1 Model Development

Designing and implementing machine learning models tailored to the characteristics of the preprocessed dataset and the detection task at hand. All the architectures are developed to leverage their respective strengths in capturing complex patterns and relationships within textual data.

7.2.2 Training

The developed models are trained using the preprocessed dataset to learn discriminative patterns between genuine and fake reviews. Training involves optimizing model parameters and objective functions to maximize detection accuracy and generalization ability.

7.3. MODULE 3 - TESTING AND EVALUATION MODULE

The testing and evaluation module focuses on rigorously evaluating the trained models based on metrics such as accuracy, precision, recall, and F1-score to assess their effectiveness in detecting fake reviews. This module encompasses the following steps:

7.3.1 Model Testing

The trained models are subjected to testing using a separate test dataset to evaluate their performance and generalization ability. The test dataset contains unseen samples to assess how well the models can generalize to new data.

7.3.2 Evaluation Metrics

Performance metrics such as accuracy, precision, recall, and F1-score are computed to quantify the effectiveness of the trained models in detecting fake reviews. These metrics provide valuable insights into the models' strengths, weaknesses, and areas for improvement.

7.3.3 Iterative Refinement

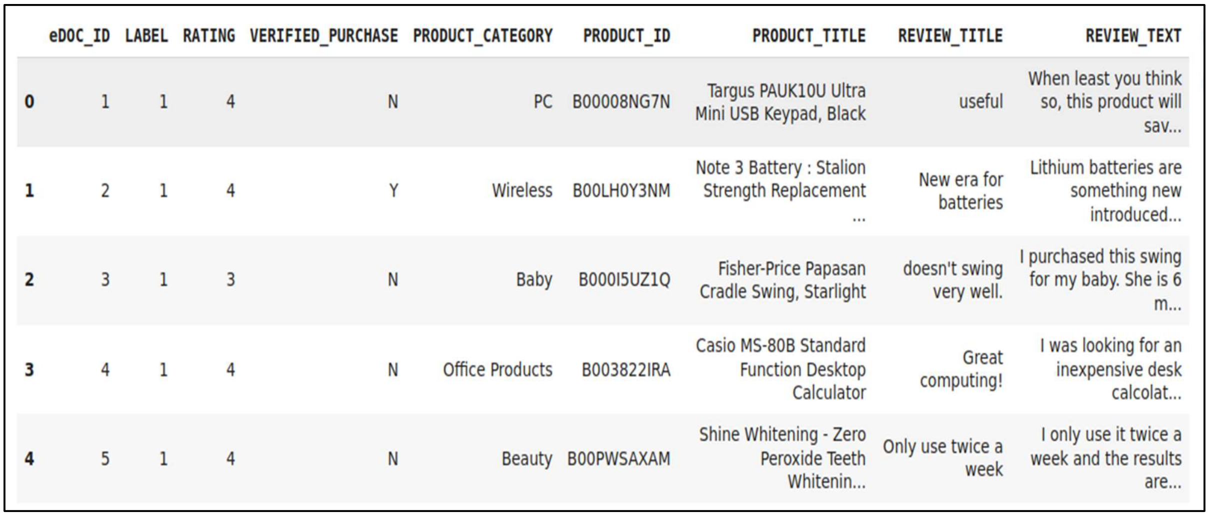
Based on the evaluation results, iterative refinement and optimization of the detection framework are performed to enhance detection accuracy and robustness. This may involve fine-tuning model parameters, adjusting feature representations, or exploring alternative architectures to improve overall performance.

7.3.4 Analysis and Comparison

The trained models are compared based on their accuracy, precision, recall and F1-score values. The algorithm which showed the highest accuracy compared to the other algorithms will be chosen for further testing.

Chapter 8
IMPLEMENTATION

8.1 Data Collection and Preprocessing



	eDOC_ID	LABEL	RATING	VERIFIED_PURCHASE	PRODUCT_CATEGORY	PRODUCT_ID	PRODUCT_TITLE	REVIEW_TITLE	REVIEW_TEXT
0	1	1	4	N	PC	B00008NG7N	Targus PAUK10U Ultra Mini USB Keypad, Black	useful	When least you think so, this product will sav...
1	2	1	4	Y	Wireless	B00LH0Y3NM	Note 3 Battery : Stalion Strength Replacement ...	New era for batteries	Lithium batteries are something new introduced...
2	3	1	3	N	Baby	B000I5UZ1Q	Fisher-Price Papasan Cradle Swing, Starlight	doesn't swing very well.	I purchased this swing for my baby. She is 6 m...
3	4	1	4	N	Office Products	B003822IRA	Casio MS-80B Standard Function Desktop Calculator	Great computing!	I was looking for an inexpensive desk calcolat...
4	5	1	4	N	Beauty	B00PWSAXAM	Shine Whitening - Zero Peroxide Teeth Whitenin...	Only use twice a week	I only use it twice a week and the results are...

Figure 8.1: The Amazon Product review dataset

Basic details

- Total number of samples = 21000
- format = .txt
- Original reviews = 10500
- fake reviews = 10500
- Number of columns = 9
- Column names = 'eDOC_ID', 'LABEL', 'RATING', 'VERIFIED_PURCHASE', 'PRODUCT_CATEGORY', 'PRODUCT_ID', 'PRODUCT_TITLE', 'REVIEW_TITLE', 'REVIEW_TEXT'.
- PRODUCT_CATEGORY column: 30 Unique categories.
- Categories: PC', 'Wireless', 'Baby', 'Office Products', 'Beauty', 'Health & Personal Care', 'Toys', 'Kitchen', 'Furniture', 'Electronics', 'Camera', 'Sports', and so on.
- 700 samples in each category (700 x 30 = 21000 total data)
- 700 = 350 in fake review and 350 in original reviews.

8.1.1. Feature Extraction in Fake Review Dataset

Feature extraction is a crucial step in preparing the fake review dataset for machine learning model training. It involves extracting informative attributes from the textual data to represent key aspects of the reviews. In this section, we delve into the extraction of specific features, including text length, number of sentences, Flesch–Kincaid grade, stop word count, upper case count, and presence of emojis, highlighting their significance in fake review detection.

1. **Text Length**

   The text length feature represents the number of characters or words in the review text. Longer reviews may indicate more detailed feedback or potentially fabricated content aimed at deceiving readers. By capturing the length of the review text, the model can discern patterns associated with genuine versus fake reviews, with longer texts potentially raising suspicion.

2. **Number of Sentences**

   The number of sentences feature counts the total number of sentences in the review text. Reviews with a higher number of sentences may contain more nuanced opinions or detailed descriptions, potentially indicating genuine feedback. Conversely, fake reviews may exhibit shorter, fragmented sentences or repetitive content. Analyzing the sentence structure provides insights into the complexity and coherence of the reviews, aiding in fake review detection.

3. **Flesch–Kincaid Grade**

   The Flesch–Kincaid grade is a readability metric that estimates the grade level required to comprehend the text. Higher-grade levels suggest more complex language usage, potentially indicative of genuine reviews with detailed analyses or technical terminology. Conversely, lower-grade levels may indicate simpler language typical of fake reviews aimed at mass appeal. By assessing the readability of the review text, the model can identify linguistic patterns consistent with genuine or fabricated content.

4. **Stop Word Count**

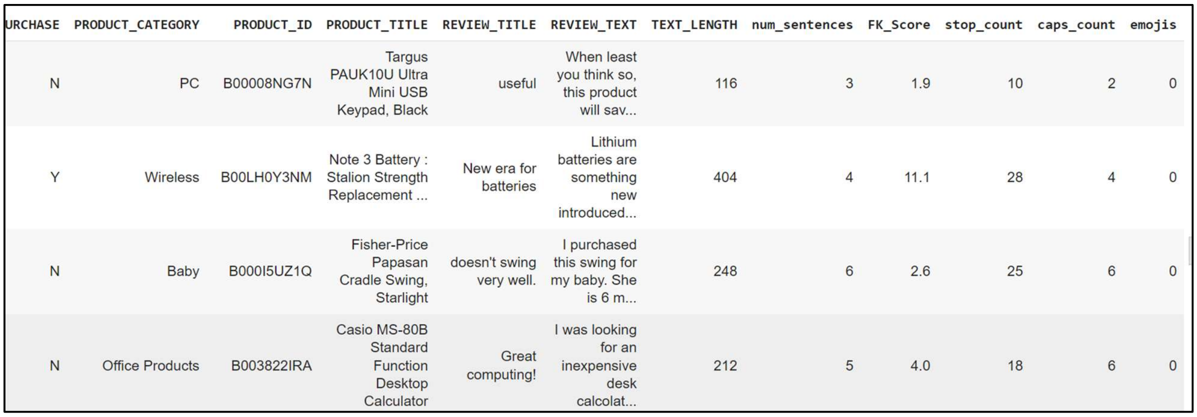
   Stop words are common words such as "and," "the," "is," etc., that carry little semantic meaning. The stop word count feature measures the frequency of stop words in the review text. Reviews with a high stop word count may exhibit less informative content, potentially indicating boilerplate or generic language characteristic of fake reviews. Analyzing stop word usage helps distinguish between genuine reviews with meaningful content and fake reviews with superficial or repetitive language.

5. **Upper Case Count**

The upper case count feature calculates the number of uppercase letters in the review text. Excessive use of uppercase letters may signify emphasis, strong sentiment, or promotional language commonly found in fake reviews. Genuine reviews typically use uppercase letters sparingly for proper nouns or emphasis, while fake reviews may employ them excessively to grab attention or convey enthusiasm falsely.

6. **Presence of Emojis**

Emojis are graphical symbols used to convey emotions, attitudes, or sentiments in textual communication. The presence of emojis in the review text indicates the expression of emotions or opinions by the reviewer. Genuine reviews may use emojis naturally to express satisfaction, disappointment, or humor, while fake reviews may use them strategically to manipulate reader perceptions. Analyzing emoji usage provides insights into the emotional tone and authenticity of the reviews, aiding in distinguishing between genuine and fake feedback.



URCHASE	PRODUCT_CATEGORY	PRODUCT_ID	PRODUCT_TITLE	REVIEW_TITLE	REVIEW_TEXT	TEXT_LENGTH	num_sentences	FK_Score	stop_count	caps_count	emojis
N	PC	B00008NG7N	Targus PAUK10U Ultra Mini USB Keypad, Black	useful	When least you think so, this product will sav...	116	3	1.9	10	2	0
Y	Wireless	B00LH0Y3NM	Note 3 Battery : Stalion Strength Replacement ...	New era for batteries	Lithium batteries are something new introduced...	404	4	11.1	28	4	0
N	Baby	B000I5UZ1Q	Fisher-Price Papasan Cradle Swing, Starlight	doesn't swing very well.	I purchased this swing for my baby. She is 6 m...	248	6	2.6	25	6	0
N	Office Products	B003822IRA	Casio MS-80B Standard Function Desktop Calculator	Great computing!	I was looking for an inexpensive desk calcolat...	212	5	4.0	18	6	0

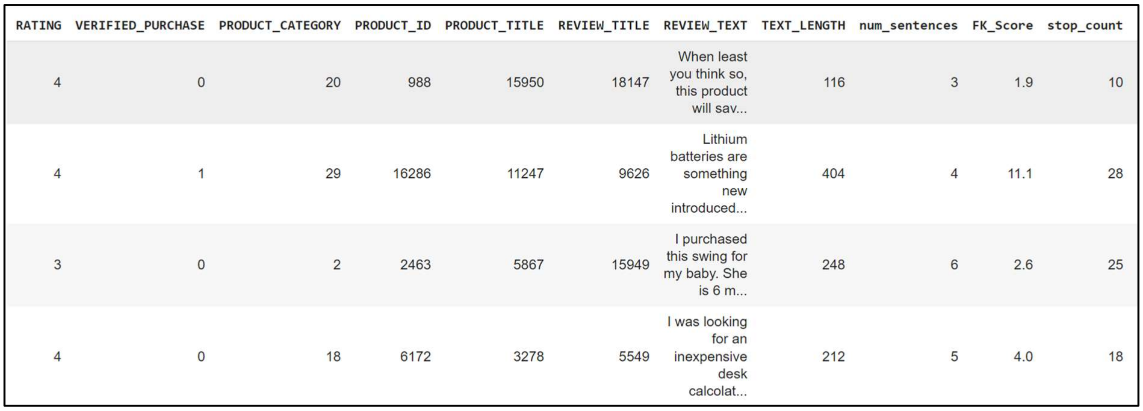
Figure 8.2: Dataset after Feature extraction

8.1.2. Label Encoding and Vectorization

1. Label Encoding

Label encoding is a technique used in machine learning to convert categorical variables (data with labels or names) into numerical representations. This is necessary because most machine learning algorithms can only work with numerical data. Yes and No in the

VERIFIED_PURCHASE column is replaced by 1 and 0. Also, PRODUCT_CATEGORY, PRODUCT_ID, PRODUCT_TITLE and REVIEW_TITLE are to be converted. Sk-learn label encoder is used. It will assign numbers from 0 to n in alphabetical order in each column entries.

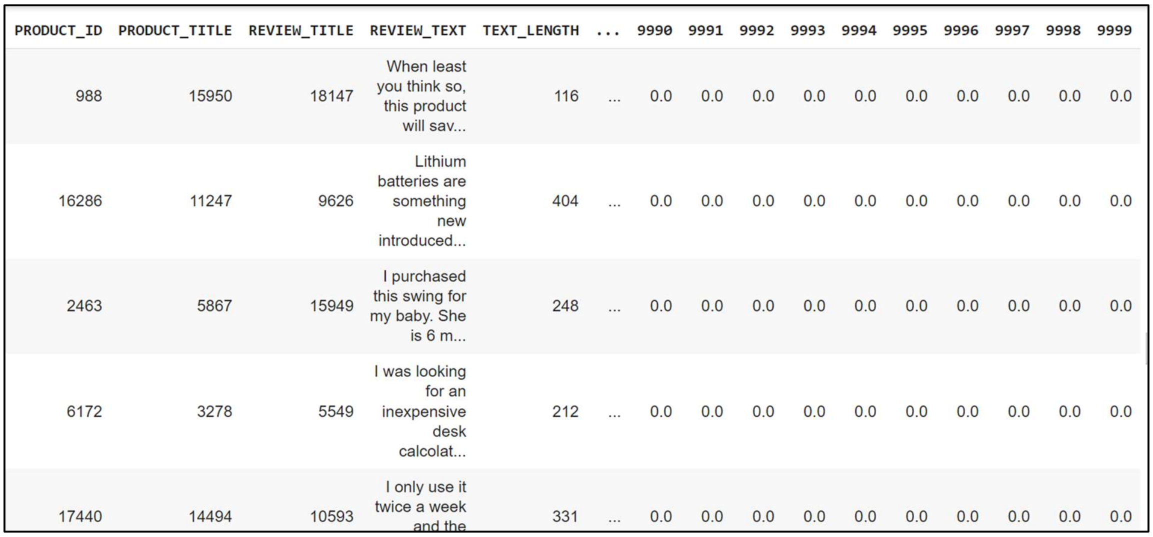


RATING	VERIFIED_PURCHASE	PRODUCT_CATEGORY	PRODUCT_ID	PRODUCT_TITLE	REVIEW_TITLE	REVIEW_TEXT	TEXT_LENGTH	num_sentences	FK_Score	stop_count
4	0	20	988	15950	18147	When least you think so, this product will sav...	116	3	1.9	10
4	1	29	16286	11247	9626	Lithium batteries are something new introduced...	404	4	11.1	28
3	0	2	2463	5867	15949	I purchased this swing for my baby. She is 6 m...	248	6	2.6	25
4	0	18	6172	3278	5549	I was looking for an inexpensive desk calcolat...	212	5	4.0	18

Figure 8.3: Dataset after Label encoding

2. Vectorisation

Vectorisation is used to convert into numbers or vectors when the data is a long text column with information. There is a vectorizer which takes the long text, assigns a number for every word; 1 is assigned if the word is present in the text, else 0 is assigned. We have chosen a vectorizer dictionary of size 10000, so the text will be converted into a vector size 10000. So the total number of features becomes 10013.



PRODUCT_ID	PRODUCT_TITLE	REVIEW_TITLE	REVIEW_TEXT	TEXT_LENGTH	...	9990	9991	9992	9993	9994	9995	9996	9997	9998	9999
988	15950	18147	When least you think so, this product will sav...	116	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
16286	11247	9626	Lithium batteries are something new introduced...	404	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2463	5867	15949	I purchased this swing for my baby. She is 6 m...	248	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
6172	3278	5549	I was looking for an inexpensive desk calcolat...	212	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
17440	14494	10593	I only use it twice a week and the	331	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

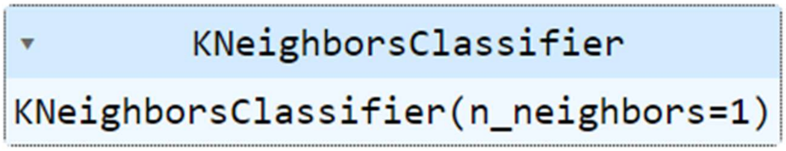
Figure 8.4: Dataset after Vectorisation

8.2 Model Development and Training

Once all the preprocessing and feature extraction is done, we split the final data into two. The train-test split is done in the ratio 80:20 that is 80% of the total data for training and 20% for testing. Training data includes 16800 data and testing data includes 4200. The models trained are KNN, DNN, XGBoost, Decision Tree and Random Forest.

8.2.1 KNN

K value is taken as 1.

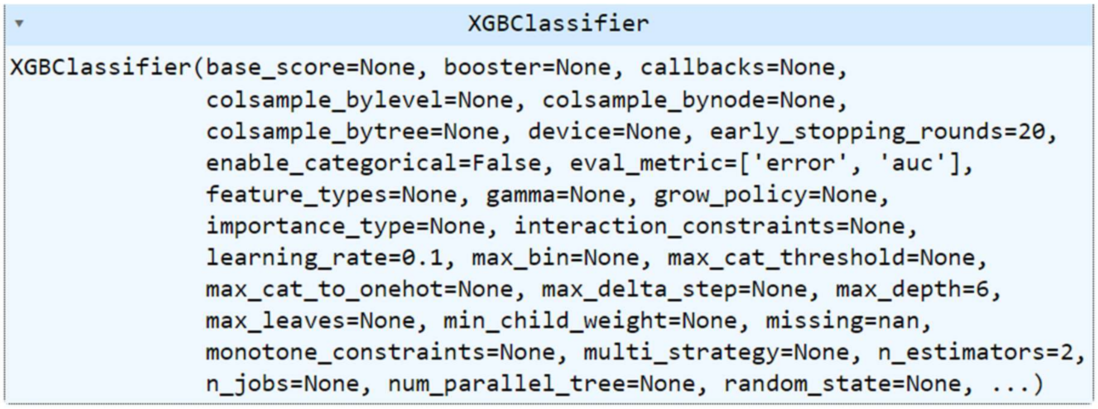


```
KNeighborsClassifier(n_neighbors=1)
```

Figure 8.5: KNN

8.2.2 XGBoost

Number of trees used is 125.

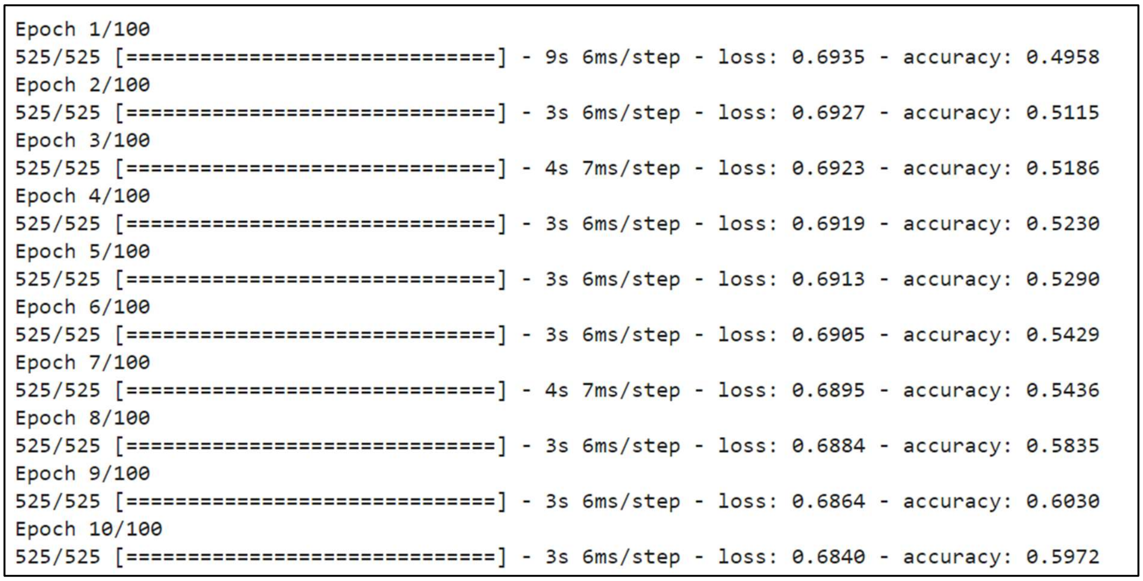


```
XGBClassifier(base_score=None, booster=None, callbacks=None,
              colsample_bylevel=None, colsample_bynode=None,
              colsample_bytree=None, device=None, early_stopping_rounds=20,
              enable_categorical=False, eval_metric=['error', 'auc'],
              feature_types=None, gamma=None, grow_policy=None,
              importance_type=None, interaction_constraints=None,
              learning_rate=0.1, max_bin=None, max_cat_threshold=None,
              max_cat_to_onehot=None, max_delta_step=None, max_depth=6,
              max_leaves=None, min_child_weight=None, missing=nan,
              monotone_constraints=None, multi_strategy=None, n_estimators=2,
              n_jobs=None, num_parallel_tree=None, random_state=None, ...)
```

Figure 8.6: XGBoost

8.2.3 DNN

Number of epochs used is 100. Batch size is assigned to be 32.

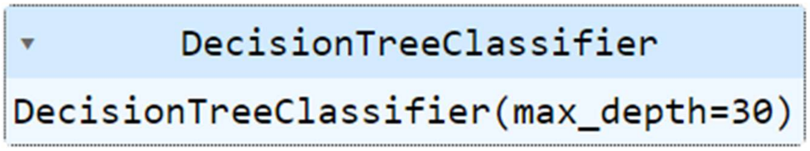


```
Epoch 1/100
525/525 [==============================] - 9s 6ms/step - loss: 0.6935 - accuracy: 0.4958
Epoch 2/100
525/525 [==============================] - 3s 6ms/step - loss: 0.6927 - accuracy: 0.5115
Epoch 3/100
525/525 [==============================] - 4s 7ms/step - loss: 0.6923 - accuracy: 0.5186
Epoch 4/100
525/525 [==============================] - 3s 6ms/step - loss: 0.6919 - accuracy: 0.5230
Epoch 5/100
525/525 [==============================] - 3s 6ms/step - loss: 0.6913 - accuracy: 0.5290
Epoch 6/100
525/525 [==============================] - 3s 6ms/step - loss: 0.6905 - accuracy: 0.5429
Epoch 7/100
525/525 [==============================] - 4s 7ms/step - loss: 0.6895 - accuracy: 0.5436
Epoch 8/100
525/525 [==============================] - 3s 6ms/step - loss: 0.6884 - accuracy: 0.5835
Epoch 9/100
525/525 [==============================] - 3s 6ms/step - loss: 0.6864 - accuracy: 0.6030
Epoch 10/100
525/525 [==============================] - 3s 6ms/step - loss: 0.6840 - accuracy: 0.5972
```

Figure 8.7: DNN

8.2.4 Decision Tree

Depth of the tree is 30.

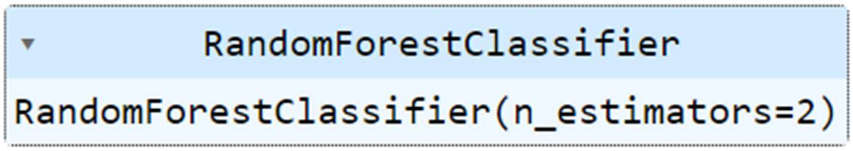


```
▾ DecisionTreeClassifier
DecisionTreeClassifier(max_depth=30)
```

Figure 8.8: Decision Tree

8.2.5 Random Forest

Number of trees used is 2.

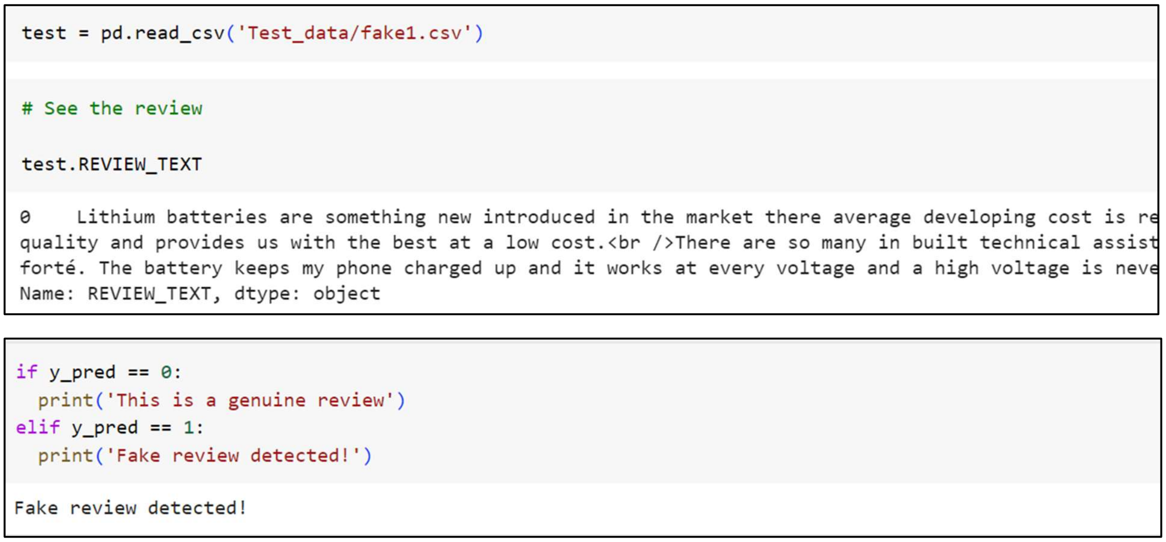


```
▾ RandomForestClassifier
RandomForestClassifier(n_estimators=2)
```

Figure 8.9: Random Forest

8.3 Testing.

8.3.1 Testing using Fake Review

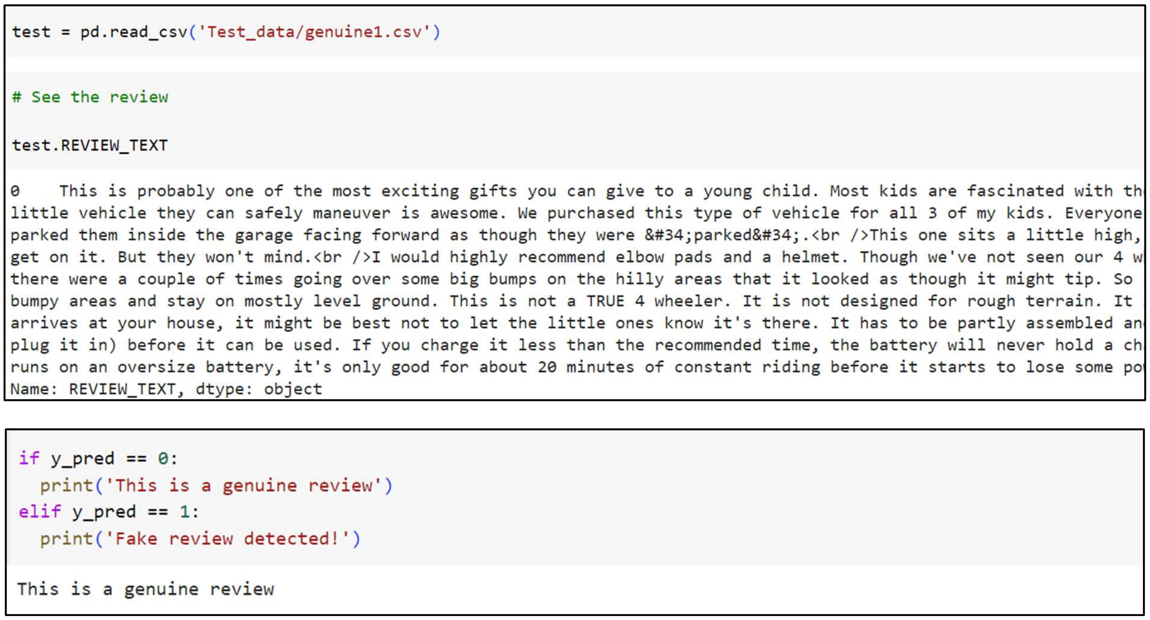


```
test = pd.read_csv('Test_data/fake1.csv')

# See the review

test.REVIEW_TEXT

0    Lithium batteries are something new introduced in the market there average developing cost is re
quality and provides us with the best at a low cost.<br />There are so many in built technical assist
forté. The battery keeps my phone charged up and it works at every voltage and a high voltage is neve
Name: REVIEW_TEXT, dtype: object
```

```
if y_pred == 0:
  print('This is a genuine review')
elif y_pred == 1:
  print('Fake review detected!')

Fake review detected!
```

Fig 8.10:Test Result for Fake Review

8.3.2 Testing using Genuine Review



```
test = pd.read_csv('Test_data/genuine1.csv')

# See the review

test.REVIEW_TEXT

0    This is probably one of the most exciting gifts you can give to a young child. Most kids are fascinated with th
little vehicle they can safely maneuver is awesome. We purchased this type of vehicle for all 3 of my kids. Everyone
parked them inside the garage facing forward as though they were &#34;parked&#34;.<br />This one sits a little high,
get on it. But they won't mind.<br />I would highly recommend elbow pads and a helmet. Though we've not seen our 4 w
there were a couple of times going over some big bumps on the hilly areas that it looked as though it might tip. So
bumpy areas and stay on mostly level ground. This is not a TRUE 4 wheeler. It is not designed for rough terrain. It
arrives at your house, it might be best not to let the little ones know it's there. It has to be partly assembled an
plug it in) before it can be used. If you charge it less than the recommended time, the battery will never hold a ch
runs on an oversize battery, it's only good for about 20 minutes of constant riding before it starts to lose some po
Name: REVIEW_TEXT, dtype: object
```

```
if y_pred == 0:
  print('This is a genuine review')
elif y_pred == 1:
  print('Fake review detected!')

This is a genuine review
```

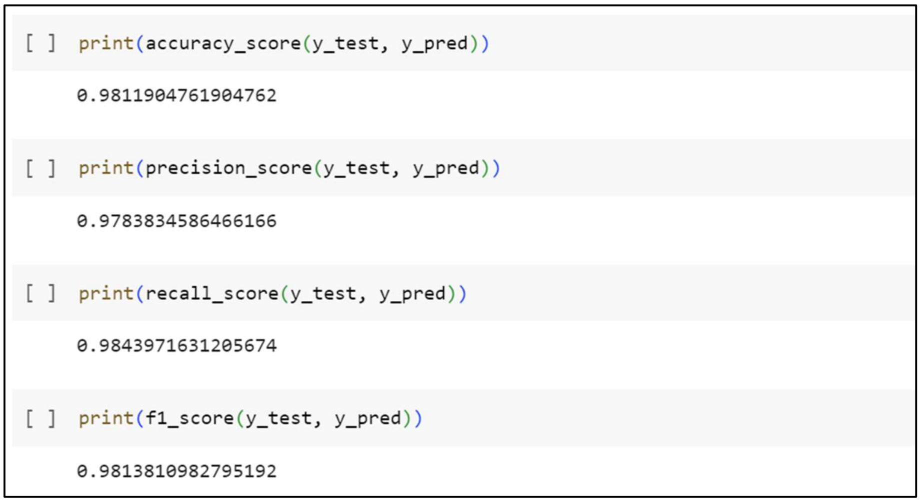
Fig 8.11:Test Result for Genuine Review

Chapter 9

RESULT AND ANALYSIS

9.1 Results

1. KNN Result:
    - Accuracy = 98.11%
    - Precision = 97.83%
    - Recall = 98.43%
    - F1 score = 98.13%



```
[ ]  print(accuracy_score(y_test, y_pred))

     0.9811904761904762

[ ]  print(precision_score(y_test, y_pred))

     0.9783834586466166

[ ]  print(recall_score(y_test, y_pred))

     0.9843971631205674

[ ]  print(f1_score(y_test, y_pred))

     0.9813810982795192
```

Figure 9.1: KNN Results

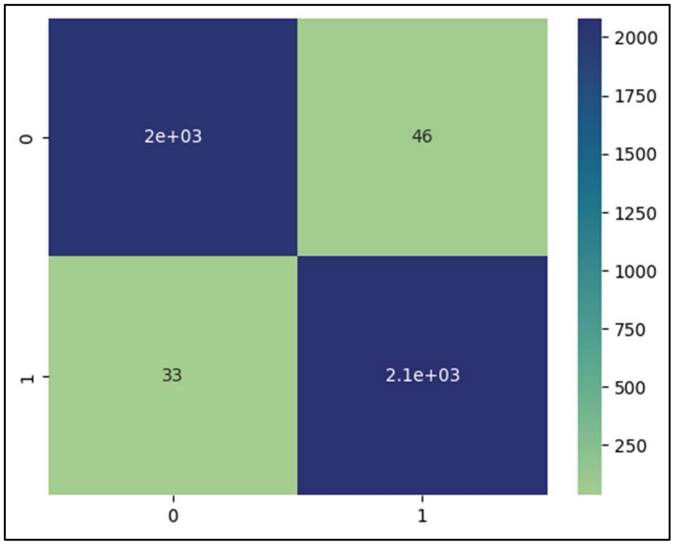
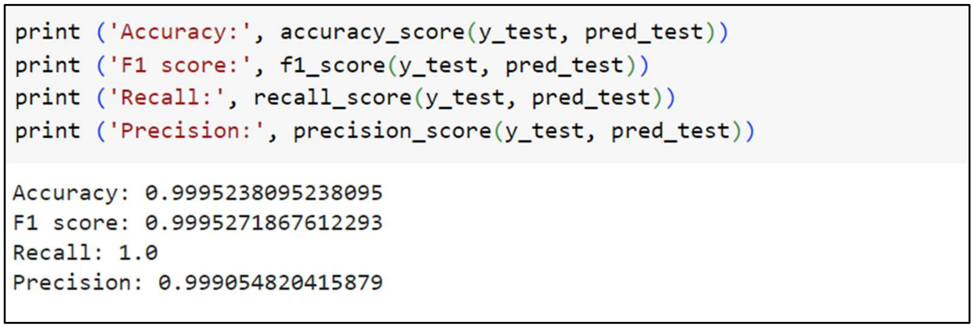


Figure 9.2: KNN Confusion matrix

2. XGBoost Result:
    - Training Accuracy = 100%,
    - Test accuracy = 99.95%.
    - Precision = 99.90%
    - Recall = 100%
    - F1 score = 99.95%



```
print ('Accuracy:', accuracy_score(y_test, pred_test))
print ('F1 score:', f1_score(y_test, pred_test))
print ('Recall:', recall_score(y_test, pred_test))
print ('Precision:', precision_score(y_test, pred_test))
```

```
Accuracy: 0.9995238095238095
F1 score: 0.9995271867612293
Recall: 1.0
Precision: 0.999054820415879
```

Figure 9.3: XGBoost Results

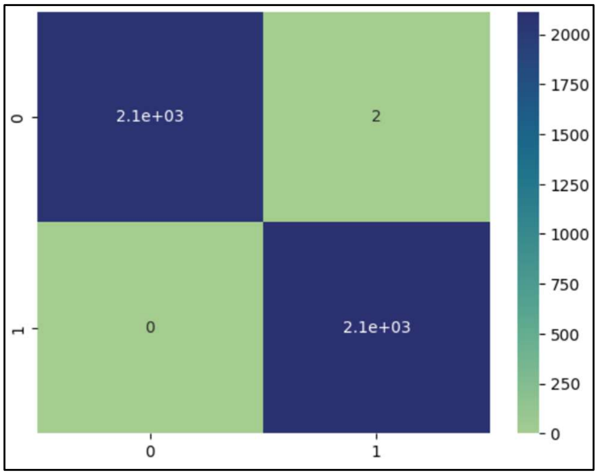
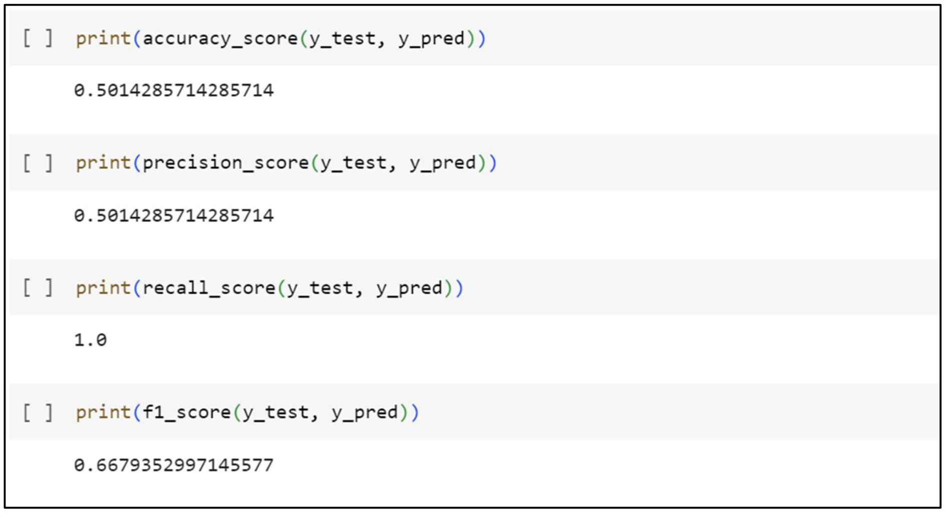


Figure 9.4: XGBoost Confusion matrix

3. DNN Result:
    - Accuracy achieved = 50.35%
    - Precision = 50.14%
    - Recall = 100%
    - F1 score = 66.79%



```
[ ] print(accuracy_score(y_test, y_pred))

    0.5014285714285714

[ ] print(precision_score(y_test, y_pred))

    0.5014285714285714

[ ] print(recall_score(y_test, y_pred))

    1.0

[ ] print(f1_score(y_test, y_pred))

    0.6679352997145577
```

Figure 9.5: DNN Results

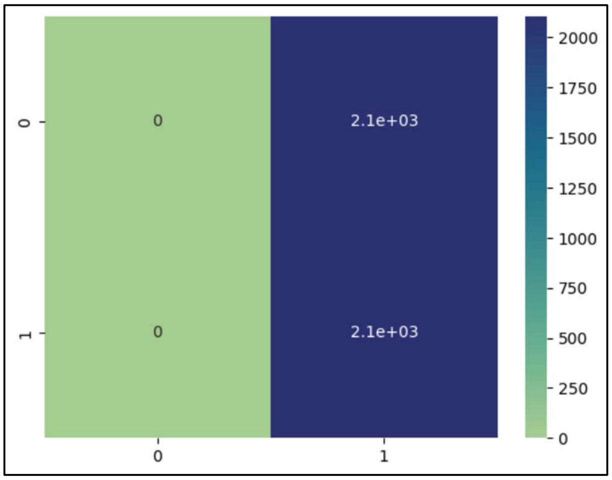
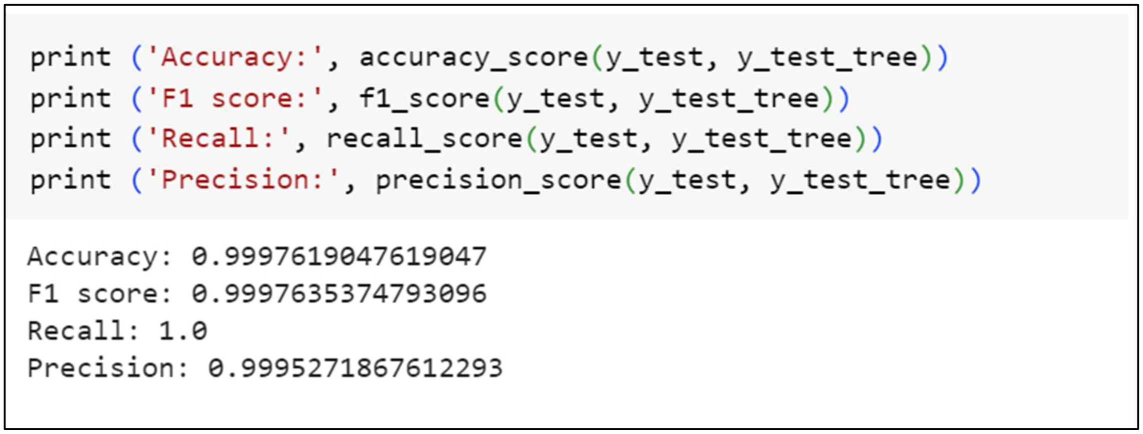


Figure 9.6: DNN Confusion matrix

4. Decision tree Result:
   - Training Accuracy = 100%,
   - Test accuracy = 99.97%.
   - Precision = 99.95%
   - Recall = 100%
   - F1 score = 99.97%



```
print ('Accuracy:', accuracy_score(y_test, y_test_tree))
print ('F1 score:', f1_score(y_test, y_test_tree))
print ('Recall:', recall_score(y_test, y_test_tree))
print ('Precision:', precision_score(y_test, y_test_tree))
```

```
Accuracy: 0.9997619047619047
F1 score: 0.9997635374793096
Recall: 1.0
Precision: 0.9995271867612293
```

Figure 9.7: Decision Tree Results

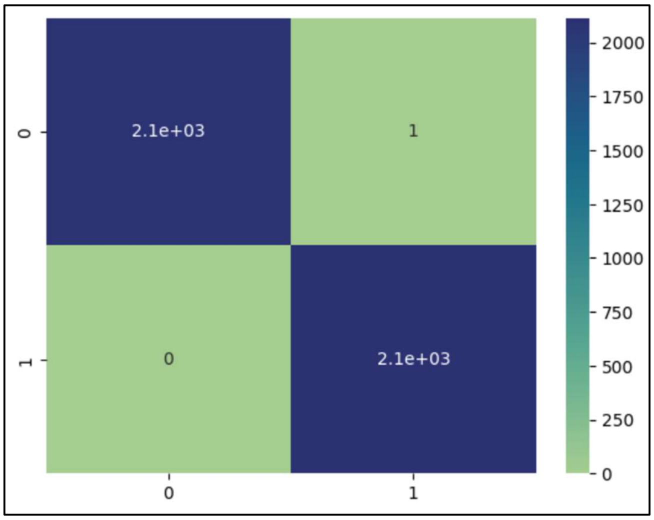
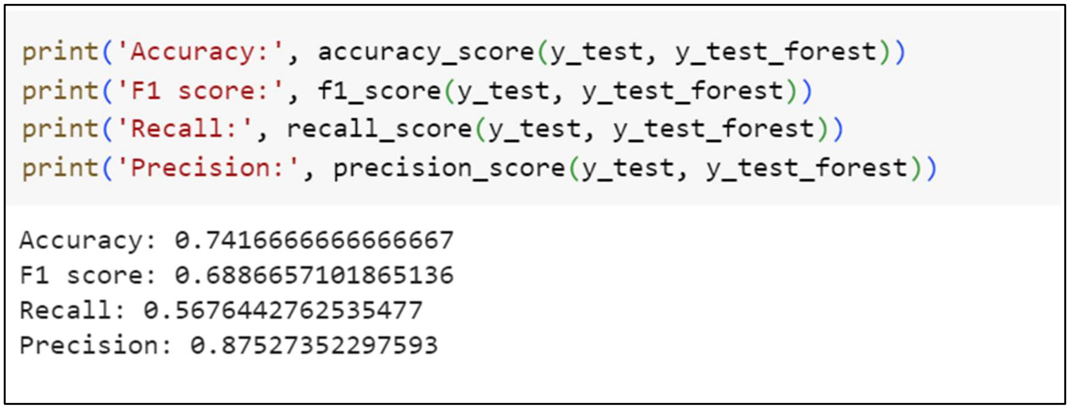


Figure 9.8: Decision Tree Confusion matrix

5. Random Forest Result:

- Training Accuracy = 90.37%,
- Test accuracy = 74.16%.
- Precision = 87.52%
- Recall = 56.76%
- F1 score = 68.86%



```
print('Accuracy:', accuracy_score(y_test, y_test_forest))
print('F1 score:', f1_score(y_test, y_test_forest))
print('Recall:', recall_score(y_test, y_test_forest))
print('Precision:', precision_score(y_test, y_test_forest))
```

```
Accuracy: 0.7416666666666667
F1 score: 0.6886657101865136
Recall: 0.5676442762535477
Precision: 0.87527352297593
```

Figure 9.9: Random Forest Results

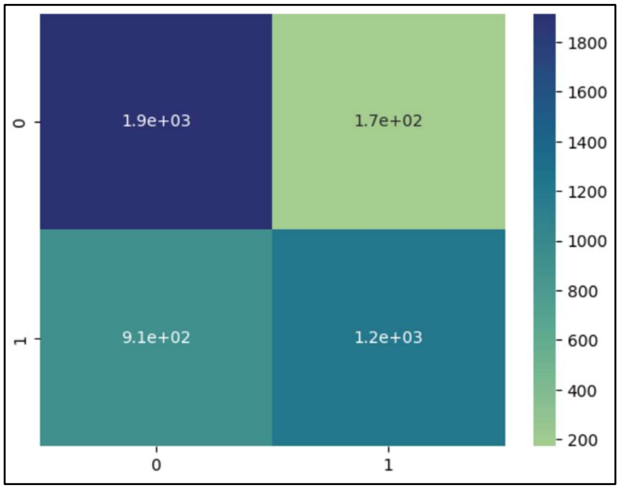
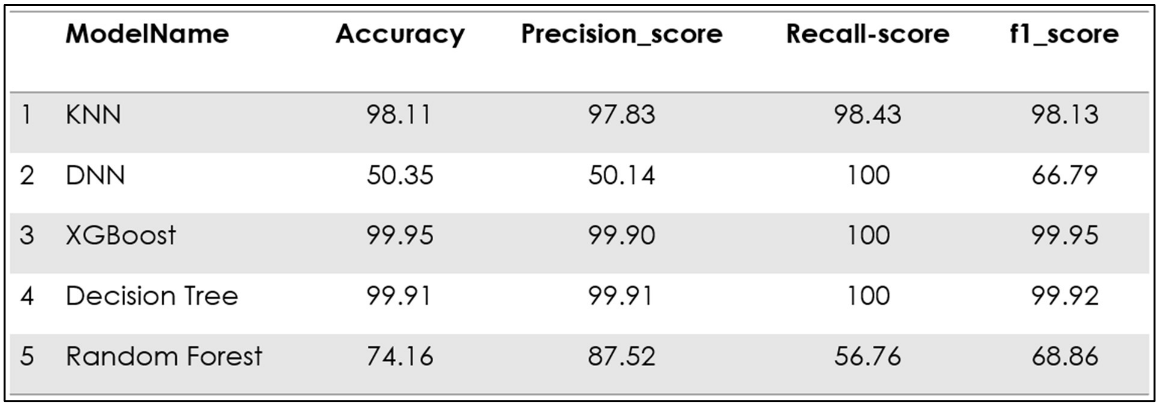


Fig 9.10: Random Forest Confusion matrix

9.2 Analysis

Our evaluation showed that XGBoost achieved the highest accuracy compared to other models. Therefore, we chose XGBoost as the best algorithm among the five algorithms.



	ModelName	Accuracy	Precision_score	Recall-score	f1_score
1	KNN	98.11	97.83	98.43	98.13
2	DNN	50.35	50.14	100	66.79
3	XGBoost	99.95	99.90	100	99.95
4	Decision Tree	99.91	99.91	100	99.92
5	Random Forest	74.16	87.52	56.76	68.86

Fig 9.11: Analysis table

Chapter 10

CONCLUSION

The development of a robust fake review detection system is imperative in today's digital landscape to uphold the integrity of online review platforms, safeguard consumer trust, and mitigate the adverse impact of deceptive practices on businesses and consumers alike. Throughout this report, we have explored the multifaceted landscape of fake review detection, delving into the historical evolution, industries affected, technologies utilized, and persistent challenges faced by practitioners in this domain. The integration of ML models, particularly ensemble learning techniques like XGBoost, has emerged as a promising approach to combating fraudulent activity, leveraging the collective predictive power of multiple base models to enhance detection accuracy and reliability. Our analysis indicates that XGBoost, with its ability to handle imbalanced datasets and capture complex nonlinear relationships, outperforms traditional machine learning algorithms in fake review detection tasks. By leveraging gradient boosting and ensemble learning principles, XGBoost effectively learns from diverse feature representations and adapts to evolving patterns of fraudulent behavior, thereby improving detection efficacy and reducing false positives. Moving forward, there is a pressing need for further research and development in the field of fake review detection to address emerging challenges and harness the full potential of ML models.

REFERENCES

[1] Nitin Jindal et al., "Fake Reviews Detection: A Survey," in IEEE Access, vol. 9, pp. 65771-65802, 2021, doi: 10.1109/ACCESS.2021.3075573.

[2] V. P. Sumathi, S. M. Pudhiyavan, M. Saran and V. N. Kumar, "Fake Review Detection Of E-Commerce Electronic Products Using Machine Learning Techniques," 2021 International Conference on Advancements in Electrical, Electronics, Communication, Computing and Automation (ICAECA), Coimbatore, India, 2021, pp. 1-5, doi: 10.1109/ICAECA52838.2021.9675684.

[3] Dewang et al. , "A Survey on Fake Review Detection using Machine Learning Techniques," 2018 4th International Conference on Computing Communication and Automation (ICCCA), Greater Noida, India, 2018, pp. 1-6, doi: 10.1109/CCAA.2018.8777594.

[4] R. Agarwal and D. K. Sharma, "Detecting Fake Reviews using Machine learning techniques: a survey," 2022 2nd International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE), Greater Noida, India, 2022, pp. 1750-1756, doi: 10.1109/ICACITE53722.2022.9823633.

[5] B. V. Santhosh Krishna, S. Sharma, K. Devika, Y. Sahana, K. N. Sharanya and C. Indraja, "Review of Fake Product Review Detection Techniques," 2022 Second International Conference on Artificial Intelligence and Smart Energy (ICAIS), Coimbatore, India, 2022, pp. 771-776, doi: 10.1109/ICAIS53314.2022.9742735.

[6] D. Singh, M. Memoria and R. Kumar, "Deep Learning Based Model for Fake Review Detection," 2023 International Conference on Advancement in Computation & Computer Technologies (InCACCT), Gharuan, India, 2023, pp. 92-95, doi: 10.1109/InCACCT57535.2023.10141826.

[7] J. Bhopale, R. Bhise, A. Mane and K. Talele, "A Review-and-Reviewer based approach for Fake Review Detection," *2021 Fourth International Conference on Electrical, Computer and Communication Technologies (ICECCT)*, Erode, India, 2021, pp. 1-6, doi: 10.1109/ICECCT52121.2021.9616697.

[8] P. M. Kumar, S. S. Harrsha, K. Abhiram, M. Kavitha and M. Kalyani, "Role of Machine Learning in Fake Review Detection," 2022 6th International Conference on Electronics, Communication and Aerospace Technology, Coimbatore, India, 2022, pp. 1026-1030, doi: 10.1109/ICECA55336.2022.10009174.

[9] M. M. Hassan Sohan, M. M. Khan, I. Nanda and R. Dey, "Fake Product Review Detection Using Machine Learning," 2022 IEEE World AI IoT Congress (AIIoT), Seattle, WA, USA, 2022, pp. 527-532, doi: 10.1109/AIIoT54504.2022.9817271.

[10] Arjun Mukherjee, Vivekanand Venkataraman, Bing Liu, & Natalie S. Glance (2013). Fake Review Detection : Classification and Analysis of Real and Pseudo Reviews.

[11] Zhang, Qi, et al. "Detecting fake reviewers in heterogeneous networks of buyers and sellers: a collaborative training-based spammer group algorithm." Cybersecurity 6.1 (2023): 26.

APPENDIX

SOURCE CODE FOR XGBOOST TRAINING AND TESTING

TRAINING:

```
import time
from xgboost import XGBClassifier

#GET THE XGBOOST MODEL
model = XGBClassifier (
        tree_method = "gpu_hist",
        eval_metric=["error", "auc"], n_estimators=10, max_depth=6,
        learning_rate=0.1, early_stopping_rounds=20
)

# TRANSFORM THE Y DATA INTO XGBOOST FORMAT
from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()
Y_train = le.fit_transform(y_train)
Y_test = le.fit_transform(y_test)

# DEFINE THE EVAL SET AND METRIC
eval_set = [(X_train, y_train), (X_test, y_test)]
eval_metric = ["auc","error"]

 # FIT THE MODEL
model.fit(X_train, Y_train, eval_set=eval_set, verbose=False)
```

```
from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score
from sklearn.metrics import classification_report

# FINAL MODEL ASSESSMENT
pred_test = model.predict(X_test)
pred_train = model.predict(X_train)
print('Train Accuracy: ', accuracy_score(Y_train, pred_train))
print('Test Accuraccy: ', accuracy_score(Y_test, pred_test))

print ('Accuracy:', accuracy_score(y_test, pred_test))
print ('F1 score:', f1_score(y_test, pred_test))
print ('Recall:', recall_score(y_test, pred_test))
print ('Precision:', precision_score(y_test, pred_test))

# MAKE PREDICTIONS
y_pred = model.predict(X_test)

# PRINT THE CONFUSION MATRIX
cf_matrix = confusion_matrix(y_test, y_pred)
sns.heatmap(cf_matrix, annot=True, cmap="crest")
```

TESTING:

```
# READ THE TEST DATA FROM TEST_DATA FOLDER
test = pd.read_csv('Test_data/fake1.csv')

#SEE THE REVIEW
test.REVIEW_TEXT
```

```
test = test.drop(['Unnamed: 0', 'REVIEW_TEXT'], axis = 1)


# LOAD THE MODEL
# LOAD THE TRAINED SAVED XGBOOST MODEL
XGB_model = pickle.load(open("XGB_model.pickle", "rb"))


# MAKE PREDICTION
y_pred = XGB_model.predict(test)


if y_pred == 0:
  print('This is a genuine review')
elif y_pred == 1:
  print('Fake review detected!')
```