VISVESVARAYA TECHNOLOGICAL UNIVERSITY

"Jnana Sangam", Belgavi-590014

A Project Report on

"Detection and Classification of Chat GPT Generated Contents Using Deep Transformer Models"

Submitted to

Visvesvaraya Technological University

In the partial fulfilment of requirements for the award of degree

Bachelor of Engineering

In

COMPUTER SCIENCE AND ENGINEERING

Submitted By

P YASHASH	**4RA20CS065**
POOJA CN	**4RA20CS067**
VARSHA HS	**4RA20CS111**
YASEER HUSSAIN	**4RA20CS115**

Under the Guidance of

Ms. SUSHMA D S

Assistant Professor, Dept. of CSE

Rajeev Institute of Technology, Hassan

RAJEEV INSTITUTE OF TECHNOLOGY

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

HASSAN-573201

2023-2024

RAJEEV INSTITUTE OF TECHNOLOGY, HASSAN

(Approved by AICTE, New Delhi and Affiliated to VTU, Belagavi.)

Plot #1-D, Growth Centre, Industrial Area, B-M Bypass Road,

Hassan – 573201 Ph:(08172)-243180/83/84 Fax:(08172)-243183

Department of Computer Science and Engineering

CERTIFICATE

Certified that the **Project Work** entitled **"**Detection and Classification of Chat GPT Generated Contents Using Deep Transformer Model**"** is carried out by Mr. P YASHASH 4RA20CS065, Ms. POOJA C N 4RA20CS067, Ms. VARSHA H S 4RA20CS111 and Mr. YASEER HUSSAIN 4RA20CS115, a Bonafide students of **RAJEEV INSTITUTE OF TECHNOLOGY**, Hassan in partial fulfilment for the award of **BACHELOR OF ENGINEERING** in **COMPUTER SCIENCE AND ENGINEERING** of the Visvesvaraya Technological University, Belagavi during the year 2023-2024. The Projectreport has been approved as it satisfies the academic requirements in respect of project work prescribed for the said Degree**.**

Ms. Sushma DS	**Dr. SHRISHAIL MATH**	**Dr. MAHESH P K**
Assistant Professor	Head of the Department	Principal
Dept. of Computer Science & Engineering	Computer Science & Engineering	**RIT, HASSAN**
RIT, HASSAN	**RIT, HASSAN**	

Name of the examiners	Signature with date
1.	……………………
2. ………………....	……………………..

DECLARATION

We, P Yashash, Pooja CN, Varsha HS, Yaseer Hussain bearing 4RA20CS065, 4RA20CS067, 4RA20CS111, 4RA20CS115, respectively, students of 8th Sem B.E in **Computer Science and Engineering**, **Rajeev Institute of Technology, Hassan**, hereby declare that the work being presented in the dissertation entitled " Detection and Classification of Chat GPT Generated Contents using Deep Transformer Models" has been carried out by us under the supervision of guide **Sushma D S**, Assistant Professor, Computer Science and Engineering, **Rajeev Institute of Technology, Hassan**, as partial fulfilment of the requirement for the award of the Degree of **Bachelor of Engineering in Computer Science and Engineering at Visvesvaraya Technological University, Belagavi** is an authentic record of our own work carried out by us during the academic year 2023-2024

P Yashash

4RA20CS065

Pooja C.N

4RA20CS067

Varsha H.S

4RA20CS111

Yaseer Hussain

4RA20CS115

Place: HASSAN

Date:

ACKNOWLEDGEMENT

The satisfaction and euphoria that accompany the successful of any task would be incomplete without the mention of the people who made it possible, whose constant guidance and encouragement crowned our efforts with success.

We would like to profoundly thank our **Management of Rajeev Institute of Technology** & our President **Dr. Rachana Rajeev** for providing such a healthy environment.

We would like to express our sincere thanks to our principal **Dr. Mahesh P K**, Rajeev Institute of Technology for his encouragement that motivated us for successful completionof the Project Work.

We wish to express our gratitude to **Dr. Shrishail Math**, Head of the Department ofComputer Science & Engineering for providing a good working environment and for his constant support and encouragement.

It gives us great pleasure to express our gratitude to **Sushma D.S,** Assistant professor, Department of Computer Science & Engineering for her/his expert guidance, initiative and encouragement that led us to complete this Project Work.

We would also like to thank all our staffs of Computer Science and Engineering department who have directly or indirectly helped us in the successful completion of this Project Work and, we would like to thank our parents.

P Yashash
4RA20CS065

Pooja CN
4RA20CS067

Varsha HS
4RA20CS111

Yaseer Hussain
4RA20CS115

ABSTRACT

This project aims to address the pressing issue of Chat GPT Text, which has become increasingly prevalent in today's society. With the internet and social media making news more accessible than ever, the spread of Chat GPT Text can have a significant impact on social, economic, and political environments. In response to this challenge, this project investigates the use of machine learning algorithms to accurately classify news as real or fake. The project utilizes KNN, Decision Tree, and Logistic Regression algorithms to analyse large datasets of news articles and learn the patterns and characteristics of real and Chat GPT Text. The primary objective of this project is to provide users with a tool that can accurately detect Chat GPT Text and help prevent its spread.

This work signifies a significant step toward mitigating the misuse of AI-powered text generation tools by providing a robust framework for detecting and classifying AI generated textual content. By experimenting with a diverse set of machine learning and deep learning models, the study highlights the effectiveness of these models in accurately identifying content generated by ChatGPT, which is crucial for addressing concerns related to the authenticity and integrity of textual content in various applications. The results suggest that this research offers a valuable foundation for further advancements in AI content detection and moderation, ultimately contributing to a more responsible and trustworthy use of AI-generated text.

CONTENTS

Abstract	i
Acknowledgement	ii
Contents	iii
List of Figures	v

Chapter Title	**Page No.**
1. Introduction	1
1.1 Introduction about the project	1
1.2 Motivation	2
1.3 Problem Statement	2
1.3.1 Input	2
1.3.2 Process	2
1.3.3 Output	2
1.4 Scope of the Project	2
1.5 Objectives	3
1.6 Review of Literature	3
1.7 Organization of the Report	9
1.8 Summary	10
2. System Requirement Specification	11
2.1 Hardware Requirement	11
2.2 Software Requirement	11
2.3 Interfaces	11
2.4 Summary	12
3. High Level Design	13
3.1 Design consideration	13
3.1.1 Data Collection	14

3.2 System Architecture 15

3.2.1 Data Preprocessing 16

3.2.2 Feature Selection 16

3.2.3 Developing the Predicate Model 17

3.3 Specification using Use Case Diagram 17

3.4 Data Flow Diagram 18

3.5 State Chart Diagram for Proposed System 19

3.6 Summary 21

4. **Board Work** **22**

4.1 Data Collection 22

4.2 Pre-processing 23

4.3 Data cleaning 24

4.4 Data Transformation 25

4.5 Explanation of Board Work 25

4.6 Text Classification (Naive Bays) 27

4.7 KNN 30

5. **Conclusion and Future Enhancement** **36**

5.1 Conclusion 36

5.2 Future 36

6. **Reference** **37**

List of Figures

Page No

Figure No.	Figure Name	
Figure 3.1	Architecture Diagram of Proposed Approach	16
Figure 3.2	Use Case Diagram of Proposed Approach	18
Figure 3.3	Data Flow Diagram of Proposed System	19
Figure 3.4	State Chart Diagram of model using KNN And Naïve Bays	20
Figure 4.1	Data Set	22
Figure 4.2	Clustered tables for True data	24
Figure 4.3	Clustered table for False data	24

CHAPTER 1

INTRODUCTION

1.1 Introduction About The Project

Millions of users generate large volumes of data online per day. For various applications, it is important to be able to process this data in real time and understand the general opinions of users without having to manually sort through this data. Various rule-based, machine learning-based, and a combination of the two techniques exist in performing sentiment analysis. One of the drawbacks of sentiment analysis however, especially for machine learning- based approaches, is that the usability of a sentiment classifier is largely dependent on the type of training data being fed to the classifier.

Users behave in varying ways across various different online platforms. This difference may arise due to a variety of reasons such as the subjects being discussed, the anonymity offered by a social media platform, the level of informality expected from a platform, and the amount of expertise held by an average user of a platform. As a result, when a sentiment classifier is introduced to a completely new testing data, it produces biased results due to the unfamiliarity of a new domain

There are multiple scenarios that require an unbiased classifier to be able to perform well on unseen training data. For example, there may not exist sufficient training data for new online websites to successfully train their own sentiment classifiers. In this case, having an unbiased classifier trained on a different set of training data can be a useful alternative. In addition to this, we can also consider the use for sentiment classifiers in understanding the reception of users to a newly launched product. Different kinds of products have different jargons in their reviews that denote sentiments. In this case, if a method exists to make a sentiment classifier unbiased then it can be successfully used to test the sentiment of a newly launched products without having to worry about the jargons In this thesis, we will start with exploring the various issues in training data that need to be overcome while using a sentiment classifier. We will then proceed to test out the various methods for training a sentiment classifier. After that, we will analyze the bias that occurs when a sentiment classifier is cross- trained on a different set of training dataWe will then propose various methods to mitigate this bias and compare the results to denote the best possible solution.

1.2 Motivation

Detecting and classifying ChatGPT contents using deep transformer models can be motivated by several reasons, each addressing different concerns and objectives. Here are some key motivations for such tasks: Content Moderation and Compliance, Filtering Offensive Language, Compliance with Policies and Regulations, Addressing Abuse and Harassment, Customizing Responses, Identifying Sensitive Information, Protecting Against Malicious Use, Handling Ambiguity, Adapting to Dynamic Contexts.

The motivation for detecting and classifying ChatGPT contents using deep transformer models stems from a combination of user safety, compliance, privacy protection, model improvement, and ethical considerations, all of which contribute to a more responsible and effective deployment of conversational AI systems.

1.3 Problem statement

The problem is to develop a deep learning model based on transformer architecture to detect and classify text generated by ChatGPT (a language generation model) from human- generated text. The aim is to differentiate between chat conversations that are produced by a human versus those that are generated by the AI model.

1.3.1 Input:

A dataset containing raw news articles, which are collected from internet or some other sources.

1.3.2 Processing:

- Machine learning algorithms: Naive Bayes,
- Classification: KNN .

1.3.3 Output:

- Classifies wheather the given content is ChatGPT generated or Human written.
- Accurately classify news as either real or fake.
- And predicts the percentage of truth if the news is real.

1.4 Scope of the Project

- Expanding the scope of the project to include detection of more types of misinformation, such as propaganda, and conspiracy theories and wheather it is generated by ChatGPT .

- Developing a real-time system that can analyze news articles as they are published to quickly identify and flag potentially Chat GPT Text stories.
- Integrating the Chat GPT Text detection model using GLTR(Gaint language model testroom) with existing social media platforms to automatically flag potentially Chat GPT Text stories, helping to prevent the spread of misinformation.

1.5 Objectives

- To collect and prepare a diverse set of training data that includes various scenarios and aspects of Chat GPT generated text i.e which includes news articles.
- To explore and use different machine learning algorithms such as Naive Bayes,Random forest,Logistic Regression,CNN,k-nearest neighbors as classifier to find the best algorithm for detecting Chat GPT Text.
- To develop a machine learning model that can accurately authenticate whether a given news article is real or fake.
- To evaluate the percentage of truth if the news is real.
- To continuously refine and improve the model to adapt to the ever-changing landscape of Chat GPT Text.

1.6 Literature Review

This research revolves around observing and mitigating classification bias in text obtained from social media data. This involves researching multiple subdomains of data science related to natural language processing, classification, and bias. Before diving into the research details, we first look into existing work across these domains. The scope of this literature review involves analyzing methodologies to implement the various parts of this research. We start with defining what bias is in terms of data science, We then proceed to explore research in text classification and then move on to sentiment analysis and finally cross-domain classification.

[1] Bias

In the current day and age, data science is heavily used across multiple domains for recommendations, classification and predictions. Multiple machine learning algorithms have been created and used depending on the problem statement. While these algorithms are not discriminatory by themselves, bias can seep into many applications due reasons such as unfair representation in training data, insufficient training data, or faulty feature selection. 5 Eirini

Ntoutsi and colleagues [8] discuss the history of bias in data mining along with ways to mitigate bias depending on the use cases. A system is said to be biased when it has a higher probability of coming to one conclusion over others. While not inherently undesirable, various machine learning problems need to do away with bias to ensure that end-users are not being discriminated against. Bias mitigation techniques focus on one of the three approaches – collecting data on bias awareness , modifying the machine learning algorithms to mitigate bias, and finally adjusting the results post-learning to account for bias. Through experiments, we will determine the presence of bias in a text classification problem (i.e, sentiment classifiation) by training on a Reddit dataset and testing on a Twitter dataset. We will then explore methods to identify the presence of bias in the outcomes followed by testing ways to mitigate this bias.

[2] **Text Classification**

Text classification is not a new problem. Social media websites, online retail stores, and blogs are among numerous sources that generate a large amount of user text. Given the volume of text generated, it is often imperative to devise methods to classify text for analysis without any human involvement. Before implementing and testing one strategy for text classification, it is worth mentioning that there is no “best" strategy for text classification. In order to determine what strategy works the best, often the structure of the input data needs to be considered. Common strategies to process and classify text include, but are not limited to nearest neighbor methods such as KNN, Multi-class SVMs, Web-based categorization techniques, semantic labeling, and random walks.

[3] Sentiment Analysis

Sentiment classification is a form of text classification problem discussed earlier. The rise in end-user generated, opinionated online content has led to the need of classifying text on the basis of the sentiment behind them. Sentiment analysis allows a classifier to gauge the underlying sentiment in some text. This is particularly useful in use cases involving reviews of various products and services. The work by Walaa Medhat [15] provides a systematic study of various sentiment analysis algorithms and situations where their use are appropriate. A strategic training of a sentiment analysis classifier includes processing the text, identifying the sentiment, feature selection, and finally classification of the sentiment. There are many issues with data extracted from user generated content that can negatively impact a sentment analysis classifier. These issues are both semantic as well as syntactic. Under semantic issues, we consider the paper by Iti Chaturvedi [11] that discusses the issue in sentiment analysis related to differentiating between opinions and facts. This paper discusses the methodologies to analyze and remove data instances with neutral sentiments prior to training classifiers. In this paper, this sub-task is referred to as detection of subjectivity in sentiment analysis. They seek to prevent forcefully classifying data instances into positive and negative sentiments when they are neutral or fact-based. Feature selection is an essential precursor to training the classifier. It is important to extract the correct subset of features from a given text in order to accurately represent the sentiment-related factors. Some popular options to select such features in text are detecting the polarity of objectives in text, identifying the presence of negations in text, and noting the frequencies of using certain terms in speech. 7 Feature selection is followed by selecting a classification algorithm. The algorithm is responsible for training the classifier. Sentiment classifiers can be roughly categorized into three broad categories. These are the machine learning-based approaches, lexicon-based approaches, and a hybrid of the previous two approaches. For our project, we have considered sentiment analysis approaches for social media and product reviews datasets. A survey by Lin Yue [17] focuses solely on sentiment analysis performed on social media data. This paper provides an extensive study of the current status of sentiment analysis algorithms. They discuss sentiment analysis approaches under granularity oriented sentiment analysis, document level sentiment analysis as well as sentiment analysis based on individual words and sentences. In therms of methodologies, they broadly classify sentiment analysis approaches in terms of supervised learning, unsupervised learning and finally semi-supervised learning approaches. For the datasets, they consider Twitter as well as Facebook datasets in order to test and analyze various approaches.

[4] Machine Learning-based Approaches

Various existing machine learning algorithms can be redesigned in order to fit a sentiment classification problem. In this case, the training file consists of a set of records along with their target sentiment class. Machine learning algorithms can further be broken down into supervised and unsupervised learning algorithms. Supervised machine learning algorithms are appropriate when we have training data that already contain the target sentiment class for each record. Traditional machine learning classifiers such as Naive Bayes classifiers, Bayesian networks, Neural networks, Decision tree classifiers, and Support Vector Machines all have comparable accuracies in sentiment classification. For the scope of this project, since we have tweets and 8 comments with their associated targets, we can initially consider supervised learning algorithms for training the classifiers. Support Vector Machines are especially useful in classification of text documents. This is due in part to the sparsity of the features in a text document and the relatedness of the features. Chen [4] focuses on utilizing support vector machines to classify sentiment polarity of user reviews. A lot of research has been done into utilizing machine learning for classifying sentiments for social media data. Twitter provides one of the best sources to extract text classification data from as metrics such as likes and retweets provide a good metric for labeling the data. A paper by YungMingLi [18] focuses more on classifying tweets using support vector machines while also adding the feature of user credibility when training the classifier. A survey by Anastasia Giachanou and Fabio Crestani [9] analyzes and compares various methods to perform sentiment classification on Twitter data. In this survey, they discuss various Twitter specific data challenges that need to be dealt with to perform sentiment analysis. These include length of text, relevance of topics, correctness of grammar, sparsity of data, stop words and multilingual content. They go ahead to discuss various features present in this text based dataset. These include semantic features such as the different kinds of negations and opinion words present in the text as well as syntactical features such as term frequencies, bigrams, n grams and parts of speech. Machine learning approaches are discussed that use SVMs to predict the sentiments of the data instances as positive, negative and neutral Tweets. They further discuss evaluation metrics such as F measure, precision, recall and accuracy in order to evaluate the classifiers.

[5] Other Approaches

Sentiment classification algorithms are not limited to using machine learning approaches. In the case where target classes are not available, a lexicon-based approach can be used to determine the sentiment behind classes. This can involve incorporating dictionaries to identify positive and negative words, statistical approaches to identify patterns or corpus-based approaches that make use of opinion words along with connectives and conjunctions. As the scope of this project is to identify and mitigate bias, our focus is more towards a machine learning-based approach. We also look into a more hybrid learning model that involves lexicon based segregation before utilizing machine learning for training. The work of Dalibor [3], shows that it is apparent that we have various challenges when it comes to analysis of user generated text. These include, but are not limited to, presence of sarcasm in text, ironic sentences, fake news, indistinguishable facts and opinions and unstructured data. The algorithm selected needs to do away with challenges that harm the problem settings.

[6] Applications of Sentiment Classification

The work of [2] studies sentiment analysis in the movie reviews domain. Movie reviews are some of the more convenient datasets to perform sentiment analysis on due to widely available databases for training and a clear sentiment class in the form of the ‘movie ratings’ provided by the users along with their text-based reviews. The authors employ three primary machine learning methods for their study. These are namely Naive Bayes classifier, maximum entropy classifiers, and support vector machines. These machine learning models employ standard bag-of-words features. Such an implementation has a provably higher accuracy than random baselines and classification models with selected unigrams in predicting sentiments across the 10 movie-reviews domain. We note that the work of [7] best resembles our approach for sentiment classification. This approach involves using word2vec to convert Chinese words into their corresponding vectors. Support Vector Machines are then employed to train the classifier into detecting sentiments. Another similar line of research is the work of [12], which deals with sentiment analysis on social media data obtained from Twitter. The obtained tweets are preprocessed and cleaned to include parts-of-speech tags as well as reduce words to their origins. The sentiment analysis problem is then further broken down into two sub-problems, namely the target-dependent classification problem and the target-independent classification problem. The target-dependent problem deals with sentiment classification in tweets with the correct sentiment class provided. This proved to display a greater accuracy as compared to a target-independent classifier. The latter focuses more on the content and lexicons in the tweets rather than their subject. The authors further improved accuracy rates by employing a graph-based optimization framework to assign sentiments to tweets based on retweets. While we discuss text base sentiment analysis problems in this project, it is worth looking into sentiment analysis for user generated content that involves input data in forms other than that of text. A survey by Alessandro Ortis [1] discusses sentiment analysis in terms of user responses to various images. Features in this case are features pertaining to Valence, Arousal and Control. They discuss classifying the polarity of images in terms of five finegrained sentiments - anger, fear, disgust, surprise and sadness.

[7] Parts of Speech

Any classification problem involves feature engineering and often a way to generate the best possible subset of features that are relevant to the problem 11 at hand. Sentiment analysis deals with the detection of the emotions behind a user’s textual inputs. From analyzing the English language, we can hypothesize that various parts of speech play an important role in concluding sentiments. With this in mind, we take a look into ‘parts of speech’ as a possible way fir feature engineering. The use of parts of speech tags as features for sentiment analysis has been explored before. The study by [5] employs the use of parts of speech tags as a method to weigh certain lexicons prior to training a machine learning classifier for sentiment analysis.

The study initially trains a sentiment analysis classifier without any special weighing. After this, a classifier is retrained based on the following subsets of features: nouns, verbs, adjectives, and adverbs. After testing the classifier on these subsets, they are retrained based on all permutations and combinations of the four subsets and the optimal subgroup of POS tags is selected. These subsets are assigned higher weights when training a classifier. Their results show that adjectives and adverbs seem to have high relevance in determining the sentiment behind a given text. This is later tested and incorporated into our project to improve accuracy. Similarly the work of [16] discusses Parts of Speech in sentiment analysis for Twitter as a method to reduce the high dimensionality of a text corpus by reducing the number of features in a text. This paper explores this goal through a proposed method that helps to determine how related a parts of speech feature is to the overall sentiment of a given text (using $\chi 2$). This dependency is then used to create a composite set of features that is weighted based on the relevance of the feature.

[8] Cross-Domain Classification

The final stage in detection of bias during classification is testing a classifier on completely new text. Cross-classification in text-based applications involves training a classifier over one dataset and then testing its accuracy on another dataset obtained from an unseen domain. Theoretically, this would lead to lower accuracies as user behaviors across online platforms typically vary in terms of language and format of the generated text. There is also the issue of the kinds of content generated. Reddit, being anonymous, typically has larger amounts of controversial comments and ‘troll’ comments as compared to Twitter, where the identity of the poster is generally known. Our objective is to spot the bias in languages used across Reddit and Twitter by employing sentiment classification techniques. We can further analyze the potential for removing such a bias. Various state of the art algorithms exist to perform cross-domain classification of text documents. The most popular out of these are the spectral feature alignment and structural correspondence learning algorithms. For instance, [6] provides an interesting approach towards training classifiers across multiple domains. Their work expands into the previously discussed concept of target-dependent classifiers. The dataset involves reviews taken for two separate products or ‘targets’. For the sake of the experiment, four different targets are taken - books, DVDs, electronics, and kitchen appliances. Cross domain classification is a challenging problem when it comes to text-based domains as the features of the two training domain do not align with each other. This problem can be partially solved by creating a sentiment specific thesaurus. Here, lemmatized sentences can be broken down to create feature vectors that contain words that express similar sentiments to each other. Another approach to cross-domain sentiment classification is covered by the work of [10]. The paper also considers the reviews domain in sentiment 13 classification. Different categories of products may have different wordings for equally positive and equally negative reviews. With this notion, training a classifier on one domain will give inaccurate results when training on a completely new domain. The address this problem using a Topical Correspondence Transfer (TCT) approach, where different domains are represented using a term-document matrix and their relationships are explored. In other approaches of cross-domain classification, we explore the research paper by Paola Zola [19] that discusses cross-domain classification in terms of two completely different domains of user generated data. In this case, data obtained from websites such as Amazon and Tripadvisor are used to train a machine learning based sentiment classifier, This classifier is then used to test on datasets from social media such as Facebook and Twitter. Various evaluation metrics such as F score, accuracy and ROC are then used to test the efficiency of cross-classifying.

1.7 Organization of the Report

The report has been organized into the below chapters**:**

Chapter 1 – Introduction: This chapter presents a brief description about chatgpt generated text or human generated text using machine learning techniques.

Chapter 2 – System Requirements Specification: As the name suggested the second chapter consisting of specific Requirement, software and hardware Requirements that used in this project. Also, we summarise this chapter at the end.

Chapter 3 – High Level Design: This chapter contains design consideration, architecture of the proposed system, and use case diagrams.

Chapter 4- Conclusion and Future Enhancements: The chapter 8 explains about the detail functionalities and description of each of them.

1.8 Summary

The first chapter describes the importance of our project. Section 1.1 gives the introduction of the project topic. In section 1.2 the motivation of project has been discussed. Section 1.3 the problem statement has been explained and the scope of the project is described in the section 1.4. And objectives are presented in the section 1.5. Section 1.6 gives details of the literature survey reviews; the important papers referred. Finally, section 1.7 gives the organization of the report.

CHAPTER 2

SYSTEM REQUIREMENT SPECIFICATION

Software requirement specification captures all the functionalities to be implemented in this project. System requirement specifications gathered by extracting the appropriate information to implement the system. It is the elaborative conditions which the system needs to, attain. Moreover, the System requirement specifications delivers a complete knowledge of the system to understand what this project is going to achieve without any constrains on how to achieve this goal. This System requirement specifications does not provide the information to outside characters but it hides the plan.

Specific Requirements

Software requirement specification captures all the functionalities to be implemented in this project.

2.1 Hardware Requirements

- Core : Pentinum 4 or higher
- RAM : 2 GB(min)
- Storage : 80 GB

2.2 Software Requirements

- Operating system : Microsoft, Windows 10 or higher
- Coding Language: Python,HTML,Java Script and MySQL
- Browser : Any of Mozilla,Opera,Chrome etc.

2.3 Interfaces

The Jupyter Notebook combines three components.

The notebook web application: An interactive web application for writing and naag code interactively and authoring notchook documents

Kernels: Separate processes started by the notebook web application that ranse ende in a given language and returns output hack to the notebook web application. The kernel also handles things like computations for interactive widgets, tab completinet and introspection.

Notebook documents: Self-contained documents that contain a represetation of all content visible in the notebook web application, including inputs and outputs of the computations, narrative text, equations, images, and rich media representalines of objects. Each notebook document has its own kernel.
Flask web application

The notebook web application enables users to edit code in the browser, with automatic syntax highlighting, indentation, end tah completion introspection

Run code from the browser, with the results of computations attached to the code which generated them

Create and use interactive JavaScript widgets, which bind interactive user interface controls and visualizations to reactive kernel side compinations

2.4 Summary

The chapter 2 considers all the system requirements which we require to develop this proposed system. Section 2.1 grants specific requirements like programming languages, frameworks are being used and under which platform this project has been done in detail. The hardware requirements for this project have been explained in section 2.2. The software requirements are clearly explained in section 2.3. Finally, the interfaces are clearly explained in section 2.4.

CHAPTER 3

HIGH LEVEL DESIGN

A high-level design discusses an overview of how a system should work and the top level components that comprises the proposed solution. It should have very little details about implementation ie, no explicit class definition and in cases not even details such as data base type (relational or object) & programming language and platform. High Level design gives an overview of the system flow. However, this gives more information for the user to understand the logic. Here we see the basic knowledge about the system design and architecture. The issues that we see in this part which are the primary component for the design Software architecture is the first step to analyze and consider all requirements for a software and attempt to define a structure which is able to fulfill them. For this also, non- Functional requirements have to be considered such as scalability and maintainability.

Following are the issues that we see in this part which are the primary component for the design. This first design step has to be more or less independent of the programming language. Here we see the basic knowledge about the system design and architecture. The issues that we see in this part which are the primary component for the design High level design provides an over view of an entire system, identifying all its elements at some level of abstraction. This contrasts with low level design which exposes the detailed design of each of the elements. Design overview-as the project proceeds the needs is to provide an overview of how the various sub systems and components of the system fit together. In both cases the high level should be complete view of the entire system breaking it down into smaller parts that are more easily understood. To minimize the maintenance overhead as construction proceeds and the lower level is done it is best that high level design is only elaborated to the degree needed to satisfy this need.

3.1 Design Considerations

The design consideration briefs about how the system behaves for the boundary environments and what action should be taken if the abnormal case happens. Some of the Sign considoratoins are:

Data Collection

Data Preprocessing

Developing the Prediction Model

3.1.1 Data Collection

Data Collection is a process of gathering and measuring information on targeted variables in a systematic way. Formal data collection process is required as it ensures the data is defined and accurate so that the decisions based on the data are valid. The data required for the cardiac arrhythmia prediction is the heterogeneous genomes which vary from each individual. It has a large dataset with complex genetic structures which has to be handled to remove the noisy and inconsistent data

Data Preprocessing

The Preprocessing of genetic data includes the following:

Data Cleaning

In the previous section, we have discussed multiple issues with the training data that impact the accuracy of a sentiment classifier. While some of those issues need to be handled by making adjustments to a classifier or penalizing the outputs of the classifier, a reasonable number of these issues can be resolved by cleaning the data before training the classifier.

The following steps have been taken to clean the data to make it appropriate for a classifier

Removing stop-words

For this issue, we make use of the nltk ’stopwords’ library. This library provides a list of stop words that are irrelevant to sentiment analysis. Every instance of the input data is processed and the words matching the members of this set are removed.

Reducing variations in text

The PortStemmer class in the nltk package takes a word and reduces them to their root words. Preprocessing text in such a way ensures that words with the same roots get converted to the same vectors, thus reducing the number of features in a meaningful way**.**

Removing numbers and special characters

We use Regex pattern matching to remove special characters and numbers present in a text to make them more suitable for a sentiment classifier.

Removing words from a different language

We make use of a dictionary to ensure that the words that are being fed to a classifier are a part of the English language and can be reliably used in a classifier. Data instances containing no English words are skipped while vectorizing the words.

Fixing spelling mistakes

Spelling mistakes can be fixed by taking words that are misspelled and using a dictionary to find the closest matching words. These words are then replaced with their correct spellings. The same words may be misspelt in multiple places in different ways and so fixing all spelling mistakes ensures a reduction in noisy data

3.2 System Architecture

The below Figure 3.1 shows the system architecture is the conceptual design that defines the structure and behaviour of a system. An architecture description is a formal description of a system, organized in a way that supports reasoning about the structural properties of the system. It defines the system Components or building blocks and provides a plan from which products can be procured, and systems developed, that will work together to implement the overall system. The purpose of system architecture activities is to define a comprehensive solution based on principles, concepts, and properties logically related to and consistent with each other. The solution architecture has features, properties, and characteristics which satisfy, as far as possible, the problem or opportunity expressed by a set of system requirements (traceable to mission business and stakeholder requirements) and life cycle cloudscapes (eg, operational, support) and which are implementable through technologies (eg, mechanics, electronics, hydraulics, software, services, procedures, human activity) System Architecture is abstract, conceptualization-oriented, global, and focused to achieve the mission and lifecycle concepts of the system. It also focuses on high- level structure in systems and system elements. It addresses the architectural principles, concepts, properties, and characteristics of the system-of-interest. It may also be applied to more than one system, in some cases forming the common structure, pattern, and set of requirements for classes or families of similar or related systems.

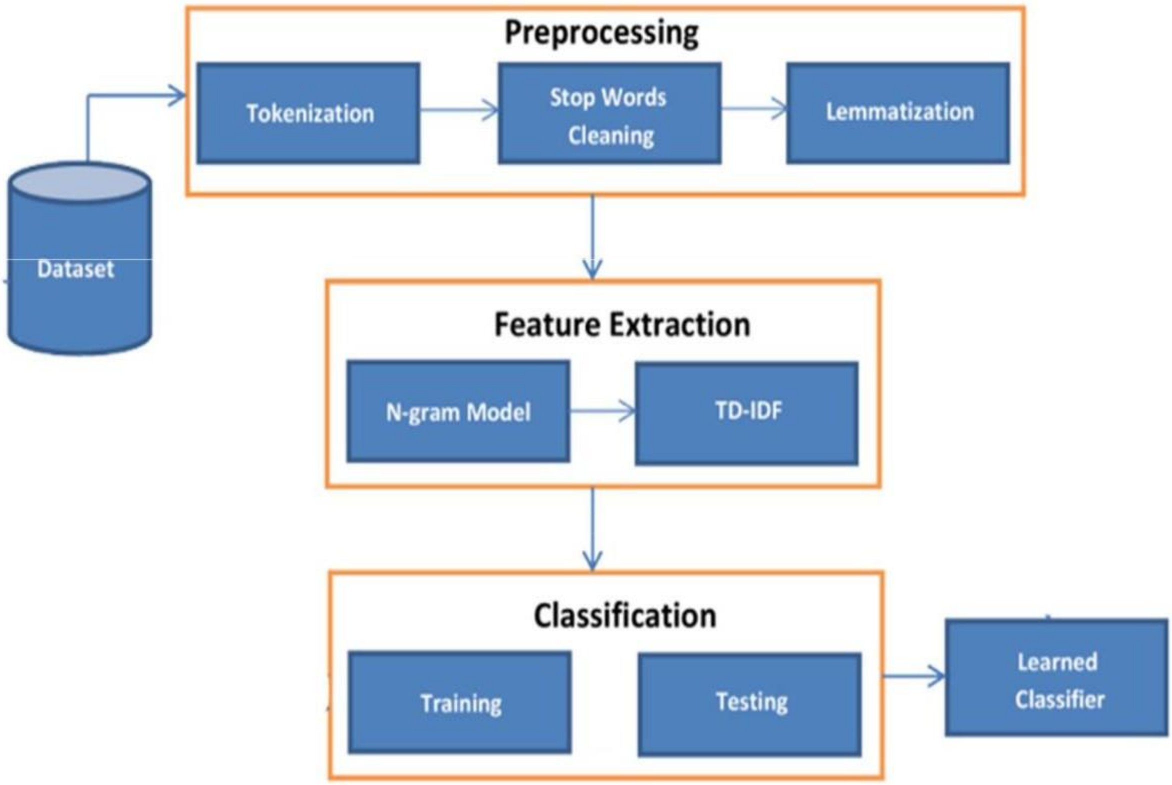


Figure 3.1: System Architecture of the Proposed Approach

3.2.1 Data Preprocessing

It is a method that uses statistical techniques to combine results from different studies and obtain a quantitative estimate of the overall effect of a particular intervention or variable on a defined outcome. The collected data were synthesized to remove irrelevant features. For example, the PR column was irreverent to develop a prediction model, thus it was removed. To handle null values, list wise deletion technique was applied where a particular observation was deleted if it had one or more missing values. Then to extract unnecessary features from the dataset.

3.2.2 Feature Selection

Feature selection is the process of identifying and selecting a subset of relevant features (variables, attributes, or inputs) from a larger set of features in order to build a predictive model or reduce the dimensionality of the data. In the feature selection use the Random-Forest algorithm. The preprocessed data has may features while the classification method that we have used is time consuming process. Is needed to reduce the time cost and also get the important featured that are maximally correlated with the output class.

3.2.3 Developing the Prediction Model

To generate prediction of fake news, algorithms had been developed and their accuracy were tested. After attaining results from various types of supervised learning like Naive Bayes, KNN; SVM was found to be highly feasible with higher accuracy than the other algorithms. Further modifications were made to the algorithm to attain even better results.

3.3 Specification Using Use case Diagram

A use case defines a goal-oriented set of interactions between external entities and the system under consideration. The external entities which interact with the system are its actors. A set of use cases describe the complete functionality of the system at a particular level of detail and it can be graphically denoted by the use case diagram.

A use case diagram at its simplest is a representation of a user's interaction with the system that shows the relationship between the user and the different use cases in which the user is involved. A use case diagram can identify the different types of users of a system and the different use cases and will often be accompanied by other types of diagrams as well.

In software and systems engineering, a use case is a list of steps, typically defining interactions between a role (known in Unified Modeling Language (UML) as an "actor") and a system, to achieve a goal. The actor can be a human, an external system, or time. In systemsengineering, use cases are used at a higher level than within software engineering, often representing missions or stakeholder goals. The detailed requirements may then be capturedin Systems Modeling Language (SysML) or as contractual statements. The Sequence of activities that are carried out are the same as the other diagrams. Use case for this module indicates the users interaction with the system as a whole rather than individual modules .All the encryption mechanisms are carried out via the login page that redirects the user to the particular functionality that he or she wishes to implement .

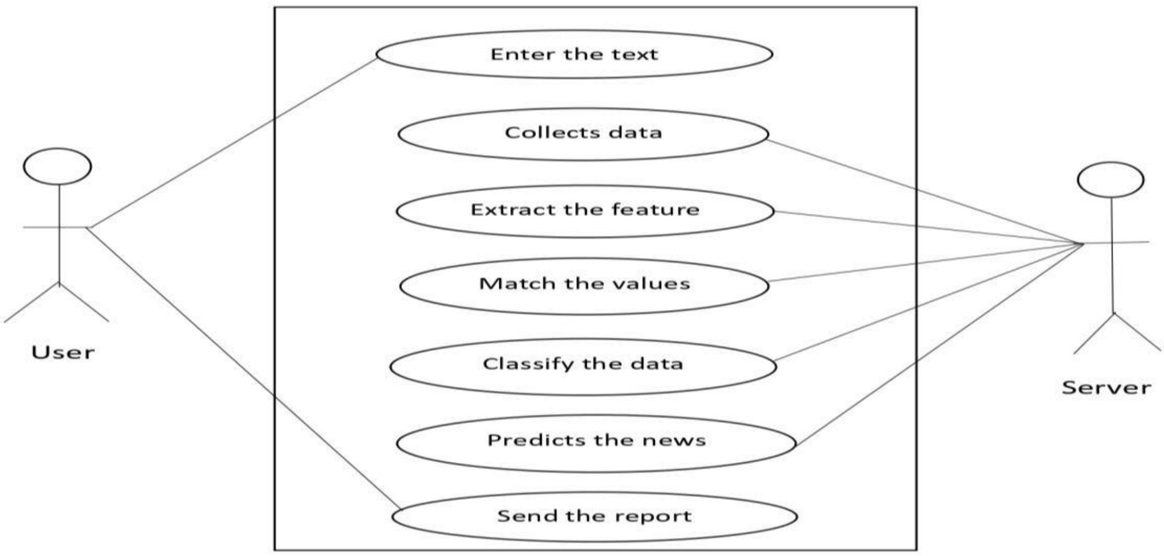
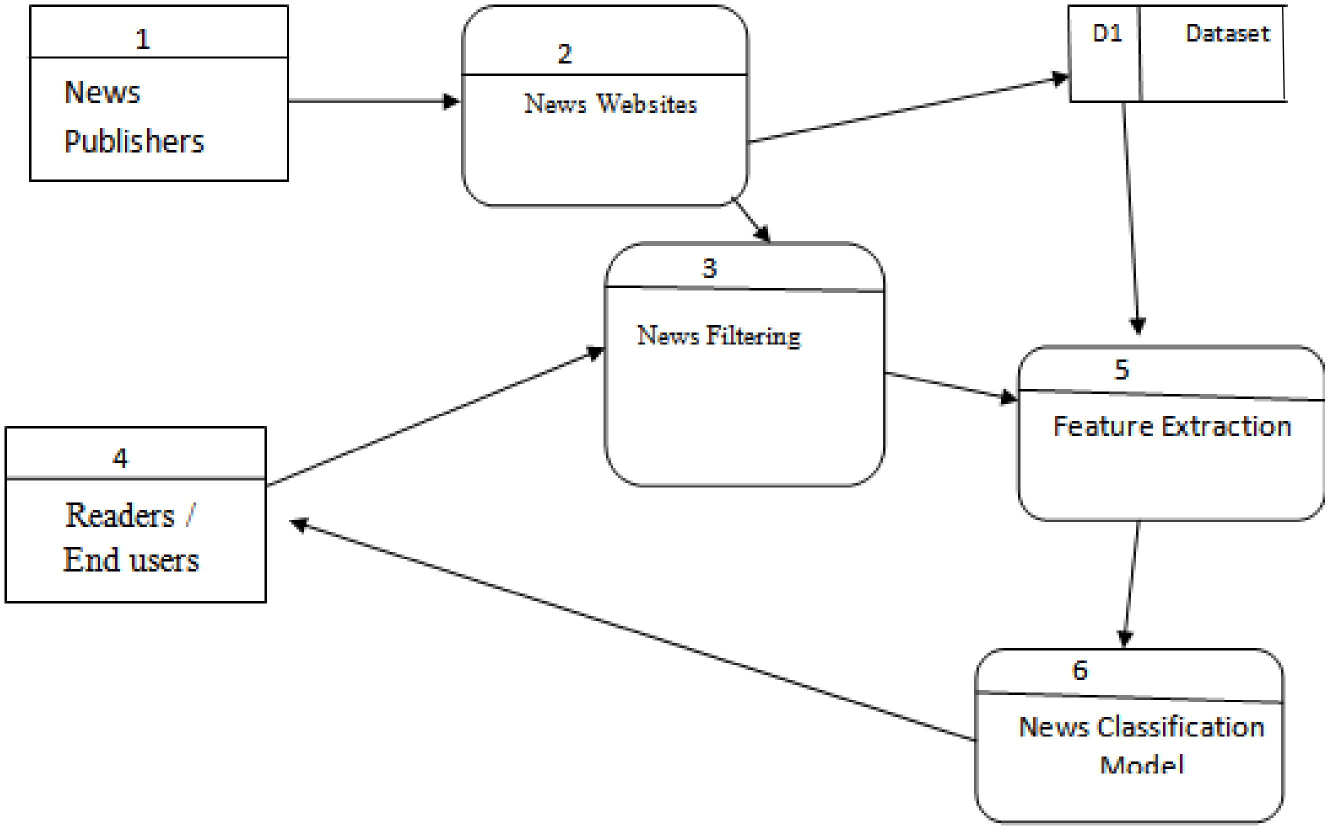


Figure 3.2: Use Case diagram for Proposed Model

3.4 Data Flow Diagram

The DFD is also called as bubble chart. It is a simple graphical formalism that can be used to represent a system in terms of the input data to the system, various processing carried out on these data, and the output data is generated by the system. A data-flow diagram (DFD) is a way of representing a flow of a data of a process or a system. The DFD also provides information about the outputs and inputs of each entity and the process itself. A data-flow diagram has no control flow, there are no decision rules and no loops. Specific operations based on the data can be represented by a flowchart.

There are several notations for displaying data-flow diagrams. For each data flow, at least one of the endpoints (source and / or destination) must exist in a process. The refined representation of a process can be done in another data-flow diagram, which subdivides this process into subprocesses. The data-flow diagram is part of the structured-analysis modelling tools. When using UML, the activity diagram typically takes over the role of the data-flow diagram. A special form of dataflow plan is a site-oriented data-flow plan.



3.3 : Data Flow Diagram For Proposed System

3.5 State Chart Diagram

A State chart is a type of diagram that represent an algorithm, workflow or process. State chart can also be defined as diagrammatic representation of an algorithm (step by step approach to solve a task). The below Figure 3.6 shows the State chart the step as poxes of various kind, and their order by connecting the boxes with arrows. In the given dataset of cardiac stored as a database file is fetched is preprocessed and then the processed data is sent for visualization and feature analysis process. After selecting the particular attributes of the dataset from the previous process, the resulting data is split into 80% and 20% as training and testing data respectively. The model is trained with data available for training. The trained model is then used for validating the data with the set of testing data, then the results are analyzed, If the results are satisfied then the model is given to the application users for their own data analysis A State chart is a type of diagram that represent an algorithm, workflow or process. State chart can also be defined as diagrammatic representation of an algorithm (step by step approach to solve a task). The State chart shows the step as boxes of various kind, and their order by connecting the boxes with arrows. In the given dataset of cardiac stored as a database file is fetched is preprocessed and then the processed data is sent for visualization and feature analysis process. The model is trained with data available for training. The trained model is then used for validating the data withthe set of testing data, then the results are analyzed, If the results are satisfied then the model is given to the application users for their own data analysis.

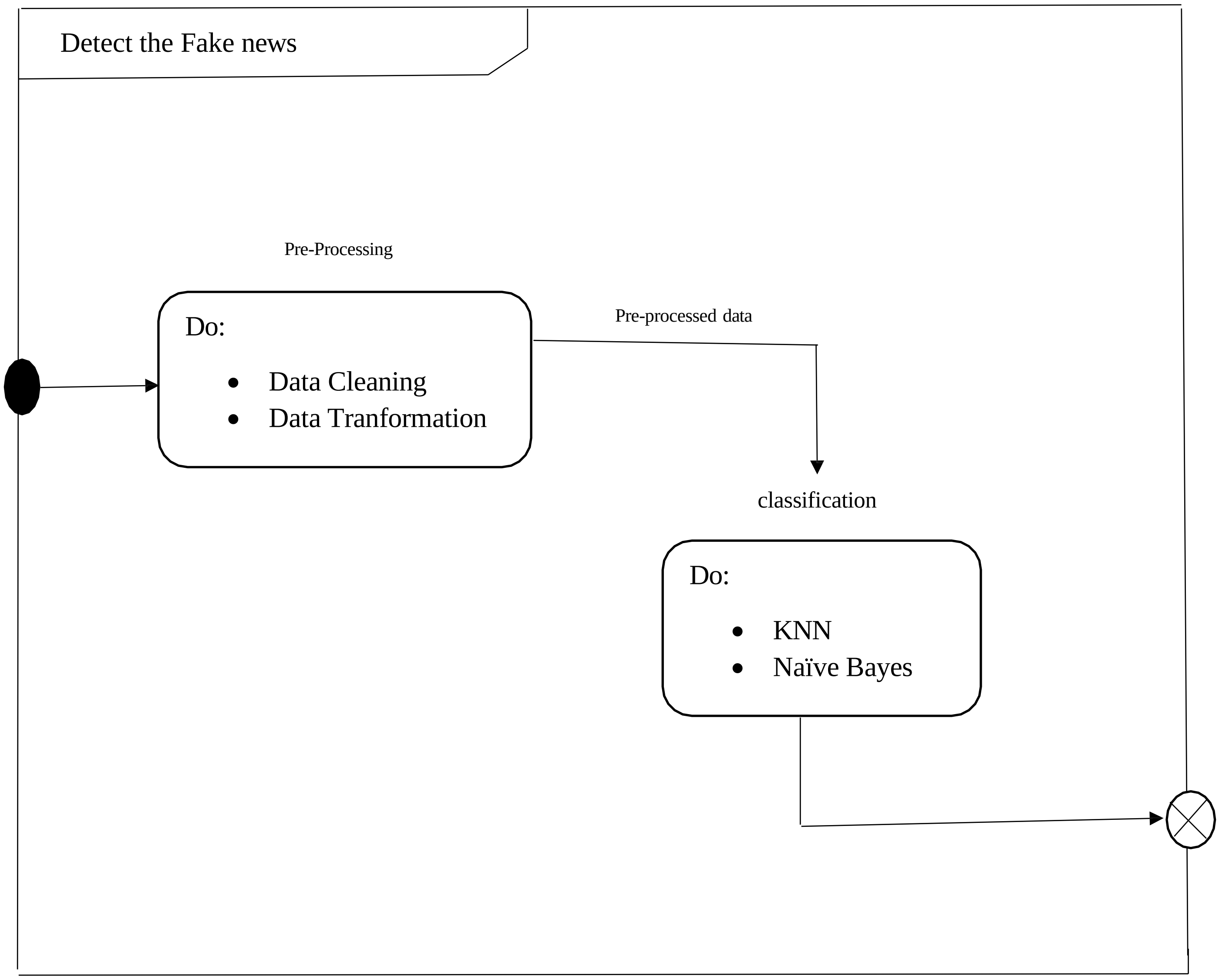


Figure 3.4: State Chart Diagram of Model Using KNN And Naïve Bays

3.6 Summary

This Chapter discusses the high-level design. In section 3.1, design considerations brief about how the system behaves for the boundary environments, in section 3.1.1, Data Collection is defined, In Section 3.1.2 Data Preprocessing methods are specified, preprocessing of data is provided. Then section 3.1.3 gives the details of data prediction modelling. The brief overview of the system architecture is defined in the section 3.2 and the specification with use case diagram is briefed in section 3.3 and section 3.4 represents the data flow diagram.

CHAPTER 4

BOARD WORK

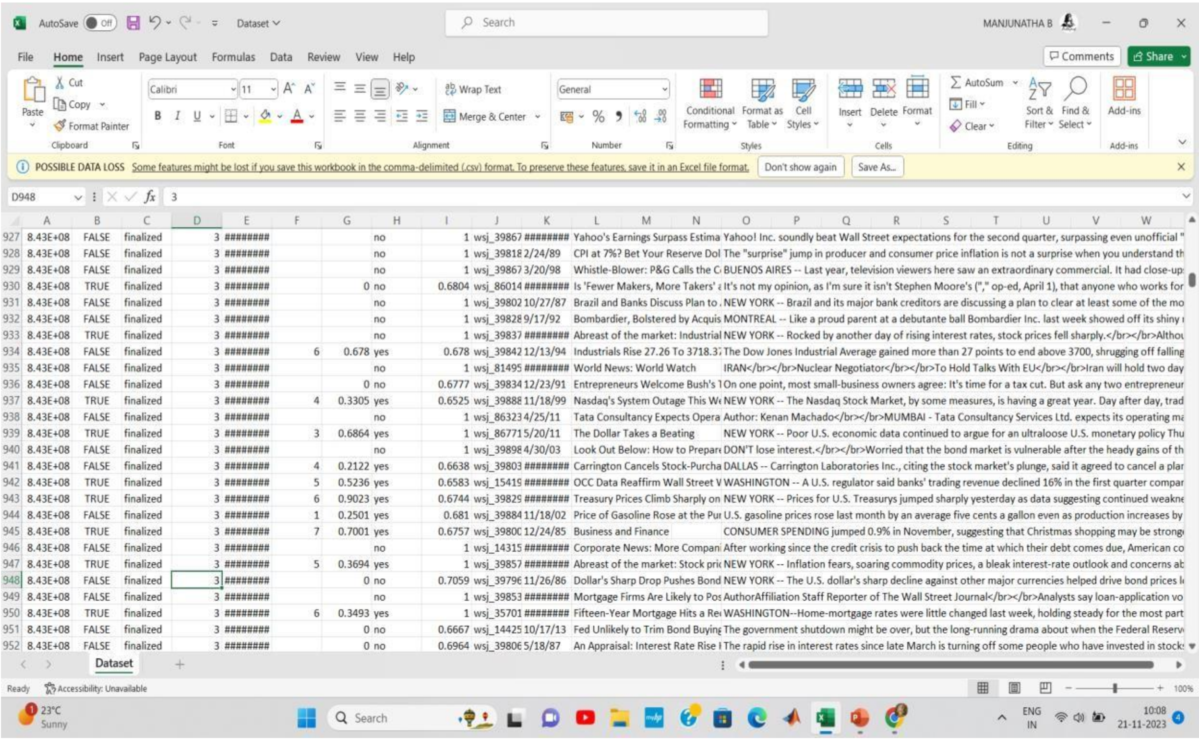


Figure 4.1: Data Set

4.1 Data Collection

- The dataset is collected from “Kaggel.com” website.

We will analyse the datasets that have been used for our experiments. For this project, we will obtain user-generated text data from two different domains and observe the similarities in bias trends and sentiment analysis accuracies across both the domains. The datasets we have picked are Reddit datasets and Twitter datasets obtained from the social media domain as well as the wine reviews and book reviews datasets obtained from the product reviews domain.

In the following sections, we will analyse the breakdown of this data, the noise present in it and methods to mitigate this noise in order to make the data more suitable for sentiment analysis.

Our data has been sourced from social media posts and review datasets. We start with smaller subsets of data and proceed to using larger datasets to measure the change in accuracy and compare the eventual similarities between the datasets.

4.2 Pre-Processing:

In Machine Learning preprocessing refers to cleaning and organizing data for building and training the Model.

It includes several steps:

1. Importing Dataset.

2. Finding Missing Data.

3. Selection of required Attributes.

4. Splitting the Dataset into Training and Testing.

4.3 Data cleaning

It is the process of cleaning the data by filling in the missing values , resolving the inconsistency and removing the outliers.

- Manually enter the missing data.
- Using attribute mean.

SL.NO	GOLDEN	POSITIVITY
1	F	4
2	F	
3	F	3
4	T	6
5	F	4
6	T	7
7	F	1

8	**T**	**9**
9	**T**	**6**
10	**F**	**4**
11	**T**	**8**
SL.NO	**GOLDEN**	**POSITIVITY**
1	**T**	**6**
2	**T**	**7**
3	**T**	**9**
4	**T**	**6**
5	**T**	**8**

Table 4.2:Clustered tables for True data

SL.NO	GOLDEN	POSITIVITY
1	**F**	**4**
2	**F**	
3	**F**	**3**
4	**F**	**4**
5	**F**	**1**
6	**F**	**4**

Table 4.3:Clustered table for False data

Missing value at index 2 is

Found using mean = (4+3+4+1+4)

5= 3

4.4 Data Transformation

- Data Normalization
- For positivity { New data = Current data *10 }
- For positivity confidence { New data= Current data(after rounding Off)*100 }

Current Points	New Points
4	40
5	50
6	60
1	10
7	70

Current Points	New Points
0.2122	(0.2,20)
0.5236	(0.5,50)
0.9023	(0.9,90)
0.2501	(0.25,25)
0.7001	(0.7,70)

4.5 Explanation of Board Work

Steps included in these tasks are: Removing punctuation marks and extra spaces

Step 1: **Tokenization** - We have used tokenization to generate a sequence of words from the user's input query.

Removing stop words - Most of the common words like 'want', 'are', and' can, which we don't need to be considered while processing is removed for improving the performance of the system.

- Below figure shows how the Raw Data / Text have got Tokenized from the sentence given below,

Tokenization

Raw text : How can I get my Address changed in my loan account.

Step 2: Tokenizing the sentence to the simple form by removing the stop words like shown in the given below figure, so that the bot can understand the tokenized words to give the response.

Removing the stop words

Tokenized words : Address , Changed , Loan , Account

Stop Words : How , Can , get , in ,my

Step 3: **Lemmatization** - We have used WordNet Lemma-tizer for getting the lemma (root form of the word) of each token. E.g., 'processing' and 'process' should be considered equal while it getting processed. So, for getting 'process' from 'processing', lemmatization is used.

. **Lemmatization**

Raw Text : How can I get my address changed in my loan account**.**

Consider an example to show how the multiple types of lemmas that convert it into the bag of words called lemmatization. Consider an example that the bot gets easily understand to give the response

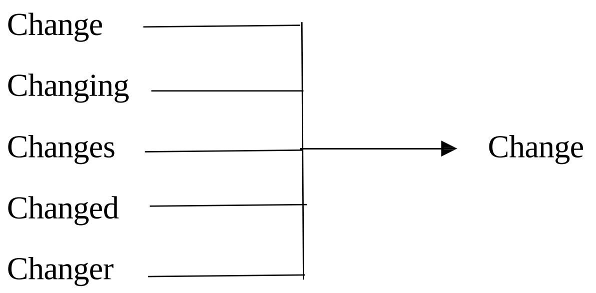


Fig: Lemmatization example

Step 4: Vectorization-The text/data is converted to vectorized format using Bag of Words concept. BOW is a method for preparing text for input to our machine learning algorithm.

Vectorization
Bag of Words

4.6 Text Classification : The Naïve Bayes algorithm

The bag of words representation

y(

I love this movie! It's sweet, but with satirical humor. The dialogue is great and the adventure scenes are fun… It manages to be whimsical and romantic while laughing at the conventions of the fairy tale genre. I would recommend it to just about anyone. I've seen it several times, and I'm always happy to see it again whenever I have a friend who hasn't seen it yet.

)=c

y(

great	2
love	2
recommend	1
laugh	1
happy	1
...	...

)=c

Bayes’ Rule

$$P(C\,|\,D) = \frac{P(D\,|\,C)P(C)}{P(D)}$$

Where:

P(C) = Independent probability of event C.

P(D) = Independent probability of event D.

P(C|D) = Conditional Probability of C given D.

P(D|C) = Conditional Probability of D given C.

Example : 1

- Vectors from the bag of words are represented in matrix format.
- Which includes most repeated words or based on weight of the words with respect to that sentence.
- In order to prove good or bad ,true or false , and for spam detection we will mostly go for this algorithm.

Here we make use of bayes theorem in order to solve this problem.

Sen1 = The food is delicious

Sen2 = the food is bad

Sen3 = Food Bad

Sen4 = Food is delicious

Sen5 = bad

	The	**food**	**delicious**	**bad**	**output**
Sentence1	1	1	1	0	1
Sentence2	1	1	0	1	0
Sentence3	0	1	0	1	0

Sentence4	0	1	1	0	1
Sentence5	0	0	0	1	0

- Here we go on computing the probability values in this theorem:
- P(y=yes/sentence)
- P(y=yes)/(x1,x2,x3,…..,xn)
- P(y)*p(x1/y)*p(x2/y)*p(x3/y)

sentence = group of words=((x1,x2,x3,…..,xn))

Example:

Finding Probability for yes :

P(yes)=P(y=yes)*P(x1/y=yes)*P(x2/y=yes)

P(yes)=2/5 * 1/2 * 2/4 * 2/2

P(yes)=0.1

Finding Probability for No :

P(no)=P(y=no)*P(x1/y=no)*P(x2/y=no)

P(no)=3/5 * 2/3 * 1/1 *3/3

P(no)=0.4

To Normalize these values :

For yes : $\frac{0.1}{0.1+0.4}$

= 0.1/0.5

= 0.20

For No : = 1 - 0.20

= 0.80

Which one has the highest probability that is considered as the output of the sentences (that may be either good or bad review or the given review is true or false based on judgement formed and opinion on each.

KNN : K- Nearest Neighbours

- The K-Nearest Neighbours (K-NN) algorithm is a popular Machine Learning algorithm used mostly for solving classification problems.
- In this article, you'll learn how the K-NN algorithm works with practical examples.
- We'll use diagrams, as well sample data to show how you can classify data using the K-NN algorithm. We'll also discuss the advantages and disadvantages of using the algorithm.

How does the k-nearest neighbours algorithm work?

The K-NN algorithm compares a new data entry to the values in a given data set (with different classes or categories).

Based on its closeness or similarities in a given range (**K**) of neighbours, the algorithm assigns the new data to a class or category in the data set (training data). Let's break that down into steps:

step #1 - assign a value to **K**.

Step #2 - calculate the distance between the new data entry and all other existing data entries (you'll learn how to do this shortly). Arrange them in ascending order.

Step #3 - find the **K** nearest neighbours to the new entry based on the calculated distances. **Step #4 -** assign the new data entry to the majority class in the nearest neighbors. Don't worry if the steps above seem confusing at the moment. The examples in the sections that follow will help you understand better.

How to Calculate Euclidean Distance in the K-Nearest Neighbours Algorithm

We have a new entry but it doesn't have a class yet. To know its class, we have to calculate the distance from the new entry to other entries in the data set using the Euclidean distance formula.

Here's the formula: $\sqrt{(X_2-X_1)^2+(Y_2-Y_1)^2}$

Where:

- X_2 = New entry’s positivity.
- X_1= Existing entry's positivity.
- Y_2 = New entry’s positivity-confidence.
- Y_1 = Existing entry's positivity-confidence

Let's get started!

POSITIVITY	POSITIVITY CONFIDENCE	CLASS
40	20	False
50	50	True
60	90	True
10	25	False
70	70	True
60	10	False
25	80	True

POSITIVITY	POSITIVITY CONFIDENCE	CLASS
20	35	?

Distance #1 For the first row, d1:

POSITIVITY	POSITIVITY CONFIDENCE	CLASS

40	20	False

$d1 = \sqrt{(20 - 40)^2 + (35 - 20)^2}$

$= \sqrt{400 + 225}$

$= \sqrt{625}$

$= 25$

POSITIVITY	POSITIVITY CONFIDENCE	CLASS	DISTANCE
40	20	False	25
50	50	True	?
60	90	True	?
10	25	False	?
70	70	True	?
60	10	False	?
25	80	True	?

For the first row, d2:

POSITIVITY	POSITIVITY CONFIDENCE	CLASS	DISTANCE
50	50	False	?

$d2 = \sqrt{(20 - 50)^2 + (35 - 50)^2}$

$= \sqrt{900 + 225}$

$= \sqrt{1125}$

$= 33.54$

- Same procedure for all the rows to calculate the distance.

Here's the table with the updated distance:

POSITIVITY	POSITIVITY CONFIDENCE	CLASS	DISTANCE
40	20	False	25
50	50	False	33.54
60	90	True	?
10	25	False	?
70	70	True	?
60	10	False	?
25	80	True	?

Here's what the table will look like after all the distances have been calculated:

POSITIVITY	POSITIVITY CONFIDENCE	CLASS	DISTANCE
40	20	False	25
50	50	False	33.54
60	90	True	68.01
10	25	False	14.14

70	70	True	61.03
60	10	False	47.17
25	80	True	45

Let's rearrange the distances in ascending order:

POSITIVITY	POSITIVITY CONFIDENCE	CLASS	DISTANCE
10	25	False	14.14
40	20	False	25
50	50	False	33.54
25	80	True	45
60	10	False	47.17
70	70	True	61.03
60	90	True	68.01

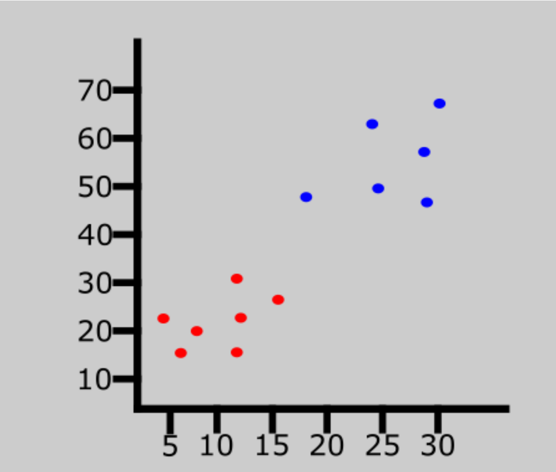
How to Choose the Value of K in the KNN Algorithm

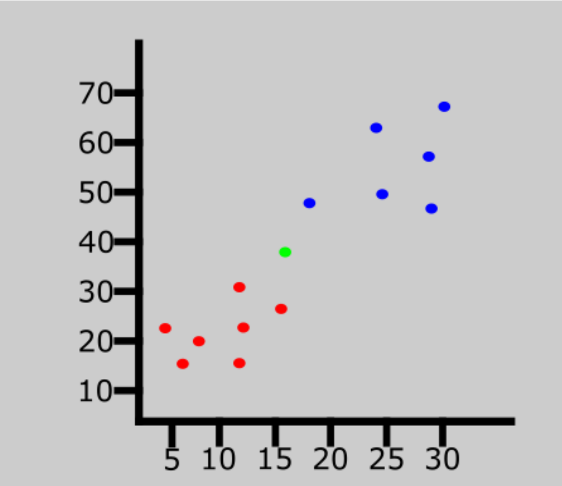
There is no particular way of choosing the value **K**, but here are some common conventions to keep in mind:

- Choosing a very low value will most likely lead to inaccurate False predictions.
- The commonly used value of **K** is 5.
- Always use an odd number as the value of **K**.

With the aid of diagrams, this section will help you understand the steps listed in the previous section.

Consider the diagram below:





- We'll then assign a value to **K** which denotes the number of neighbours to consider before classifying the new data entry. Let's assume the value of **K** is 3.
- Since the value of **K** is 3, the algorithm will only consider the 3 nearest neighbours to the green point (new entry). This is represented in the graph above.
- Out of the 3 nearest neighbours in the diagram above, the majority class is False so the new entry will be assigned to that class.

Trained data sets:

- In the positivity & positivity Confidence attribute,

Distance = 0 < 40

So Consider it as False news.

- In the positivity & positivity Confidence attribute,

Distance = 0 > 40, So Consider it as True news.

CHAPTER 5

CONCLUSION AND FUTURE ENHANCEMENT

5.1 Conclusion

In conclusion, our project on the detection and classification of ChatGPT contents using deep transformer models has demonstrated the effectiveness of leveraging advanced neural networks for analyzing and categorizing textual data. The utilization of deep transformer models has proven to be a robust approach in discerning different types of content within ChatGPT interactions. Through meticulous training and validation, our model exhibits notable accuracy in distinguishing various categories of text, contributing to enhanced content moderation and understanding. This project underscores the potential of leveraging cutting-edge technology for refining the capabilities of language models like ChatGPT in real-world applications.

5.2 Future Enhancement

For future enhancements, consider implementing continuous learning mechanisms to adapt the model to evolving linguistic patterns and user behaviors. Additionally, exploring multi-modal approaches, integrating visual and contextual cues, could further improve the accuracy and richness of content classification. Incorporating user feedback loops and sentiment analysis could enhance the model's ability to understand and respond appropriately to the emotional context of conversations. Furthermore, optimizing computational efficiency and scalability will be crucial for real-time deployment in large-scale applications. Finally, staying abreast of advancements in transformer architectures and NLP techniques will enable the integration of state-of-the-art methods to continually enhance the project's performance and adaptability.

REFERENCES

[1] Mhlanga, D. Open AI in Education, the Responsible and Ethical Use of ChatGPT Towards Lifelong Learning; Springer: Berlin/Heidelberg,Germany, 2023.

[2] M. Granik and V. Mesyura, "Fake news detection using naive Bayes classifier," 2017 IEEE First Ukraine Conference on Electrical and Computer Engineering (UKRCON), Kiev, 2017, pp. 900-903

[3] Kai Shu, Amy Sliva, Suhang Wang, Jiliang Tang, and Huan Liu, “Fake News Detection on Social Media: A Data Mining Perspective” arXiv:1708.01967v3 [cs.SI], 3 Sep 2017.

[4] Giovanni Maria Farinella Alessandro Ortis and Sebastiano Battiato. “Survey on Visual Sentiment Analysis”. In: Computer Vision and Pattern Recognition (May 2020), p. 30. URL: https://arxiv.org/abs/2004. 11639.

[5] Shivakumar Vaithyanathan Bo Pang Lillian Lee. “Thumbs up? Sentiment Classification using Machine Learning Techniques”. In: Conf. on Empirical Methods in Natural Language Processing (EMNLP) (May 2002). URL: https://arxiv.org/abs/cs/0205070.

[6] Dalibor Buži´c. “Sentiment Analysis of Text Documents”. In: Central European Conference on Information and Intelligent Systems (Apr. 2019), pp. 215–221. URL: https : / / search - proquest - com . libezproxy2 . syr . edu / docview / 2366654646 / abstract / 96B44A2143344F84PQ / 1 ? accountid=14214.

Global Conference Hub

CERTIFICATE

OF PARTICIPATION

PROUDLY PRESENTED TO

Sushma D S

Assistant Professor, Rajeev institute of technology Hassan

for Attending & Giving an oral Presentation for the Paper Entitled

Detection and classification of chatgpt generated contents using deep transformer models

in International Conference on Advancements in Science, Technology and Management (ICASTM) 2024 - Jointly Organized by Department of Computer Science, St. Xavier's College of Management & Technology, Patna, Bihar, India & Global Conference Hub, Coimbatore, Tamilnadu, India on 25/04/2024 & 26/04/2024.

Rakesh Kr. Pathak

Dr. Rakesh Kumar Pathak
Conference Convener - ICASTM 2024,
Department of Computer Science, St. Xavier's College of Management & Technology, Patna, Bihar, India.

Martin Poras

Fr. Dr. Martin Poras SJ
Principal, St. Xavier's College of Management & Technology, Patna, Bihar, India.

Global Conference Hub

CERTIFICATE

OF PARTICIPATION

PROUDLY PRESENTED TO

Pooja Cn

UG Student, Rajeev institute of technology Hassan

for Attending & Giving an oral Presentation for the Paper Entitled

Detection and classification of chatgpt generated contents using deep transformer models

in International Conference on Advancements in Science, Technology and Management (ICASTM) 2024 - Jointly Organized by Department of Computer Science, St. Xavier's College of Management & Technology, Patna, Bihar, India & Global Conference Hub, Coimbatore, Tamilnadu, India on 25/04/2024 & 26/04/2024.

Rakesh Kr. Pathak

Dr. Rakesh Kumar Pathak
Conference Convener - ICASTM 2024,
Department of Computer Science, St. Xavier's College of Management & Technology, Patna, Bihar, India.

Martin Poras

Fr. Dr. Martin Poras SJ
Principal, St. Xavier's College of Management & Technology, Patna, Bihar, India.


Global Conference Hub
CERTIFICATE
OF PARTICIPATION
PROUDLY PRESENTED TO
Varsha Hs
UG Student, Rajeev institute of technology Hassan
for Attending & Giving an oral Presentation for the Paper Entitled
Detection and classification of chatgpt generated contents using deep transformer models
in International Conference on Advancements in Science, Technology and Management (ICASTM) 2024 - Jointly Organized by Department of Computer Science, St. Xavier's College of Management & Technology, Patna, Bihar, India & Global Conference Hub, Coimbatore, Tamilnadu, India on 25/04/2024 & 26/04/2024.
Dr. Rakesh Kumar Pathak
Conference Convener - ICASTM 2024,
Department of Computer Science, St. Xavier's College of Management & Technology, Patna, Bihar, India.
Fr. Dr. Martin Poras SJ
Principal, St. Xavier's College of Management & Technology,
Patna, Bihar, India.


Global Conference Hub
CERTIFICATE
OF PARTICIPATION
PROUDLY PRESENTED TO
Yasir Hussain
UG Student, Rajeev institute of technology Hassan
for Attending & Giving an oral Presentation for the Paper Entitled
Detection and classification of chatgpt generated contents using deep transformer models
in International Conference on Advancements in Science, Technology and Management (ICASTM) 2024 - Jointly Organized by Department of Computer Science, St. Xavier's College of Management & Technology, Patna, Bihar, India & Global Conference Hub, Coimbatore, Tamilnadu, India on 25/04/2024 & 26/04/2024.
Dr. Rakesh Kumar Pathak
Conference Convener - ICASTM 2024,
Department of Computer Science, St. Xavier's College of Management & Technology, Patna, Bihar, India.
Fr. Dr. Martin Poras SJ
Principal, St. Xavier's College of Management & Technology,
Patna, Bihar, India.


Global Conference Hub
CERTIFICATE
OF PARTICIPATION
PROUDLY PRESENTED TO
Yashash
UG Student, Rajeev institute of technology Hassan
for Attending & Giving an oral Presentation for the Paper Entitled
Detection and classification of chatgpt generated contents using deep transformer models
in International Conference on Advancements in Science, Technology and Management (ICASTM) 2024 - Jointly Organized by Department of Computer Science, St. Xavier's College of Management & Technology, Patna, Bihar, India & Global Conference Hub, Coimbatore, Tamilnadu, India on 25/04/2024 & 26/04/2024.
Dr. Rakesh Kumar Pathak
Conference Convener - ICASTM 2024,
Department of Computer Science, St. Xavier's College of Management & Technology, Patna, Bihar, India.
Fr. Dr. Martin Poras SJ
Principal, St. Xavier's College of Management & Technology,
Patna, Bihar, India.