

# **FLIPKART REVIEWS SENTIMENT ANALYSIS USING DEEP LEARNING**

**A Project report submitted to**

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING WITH DATA SCIENCE**

**In partial fulfilment of the requirements for the award of the Degree of**

**BACHELOR OF TECHNOLOGY  
IN**

**COMPUTER SCIENCE AND ENGINEERING WITH DATA SCIENCE**



**By**

<b>GADDAM RAGHUVARMA</b>	<b>(Y20CDS016)</b>
<b>PENUMUCHU GRISHMA SRI</b>	<b>(Y20CDS044)</b>
<b>SANKARASETTI SITA</b>	<b>(Y20CDS052)</b>
<b>GANGURU VENKATESWARARAO</b>	<b>(Y20CDS018)</b>

**Under the Guidance of**

**Mrs. V. AMANI, M. Tech**

**ASSISTANT PROFESSOR**

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING WITH DATA SCIENCE**

**CHALAPATHI INSTITUTE OF ENGINEERING AND TECHNOLOGY (AUTONOMOUS)**

**(Accredited by NAAC with 'A' grade, accredited by NBA, Approved by A.I.C.T.E,  
Affiliated To Acharya Nagarjuna University)**

**Guntur-522034**

**2023-2024**

**CHALAPATHI INSTITUTE OF ENGINEERING AND TECHNOLOGY**  
**(AUTONOMOUS)**

**(Accredited by NAAC with 'A' grade, accredited by NBA, Approved by A.I.C.T.E, Affiliated  
To Acharya Nagarjuna University)**

**CHALAPATHI NAGAR, LAM, GUNTUR**

**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING WITH DATA SCIENCE**



**CERTIFICATE**

This is to certify that the project work entitled as **“Flipkart Reviews Sentiment Analysis Using Deep Learning”** submitted by **G.RaghuVarma (Y20CDS016), P.Grishma Sri (Y20CDS044), S.Sita (Y20CDS052) and G.Venkateswararao (Y20CDS018)** in partial fulfilment for the award of the Degree of **BACHELOR OF TECHNOLOGY IN COMPUTER SCIENCE AND ENGINEERING WITH DATA SCIENCE** is a record of bonafied work carried out under my guidance and supervision.

**GUIDE**

Ms. V. Amani, M. Tech.  
Assistant Professor

**HEAD OF THE DEPARTMENT**

Ms. K. Aruna Kumari, M. Tech, (Ph. D)  
CSDS-HOD

## **ACKNOWLEDGEMENT**

We express our sincere thanks to our beloved Chairman sir, **Sri. Y. V. ANJANEYULU** for providing support and stimulating environment for developing the project.

We express deep sense of reverence and profound gratitude to **Dr. M. CHANDRA SEKHAR, Ph.D, Principal** for providing us the great support in carrying out the project.

It plunges us in exhilaration in taking privilege in expressing our heartfelt gratitude to **Mrs. K. ARUNA KUMARI, M.Tech, (Ph.D)**, HOD-CSDS for providing us every facility and for constant supervision.

We are thankful to our guide **Mrs. V. AMANI, M.Tech**, Assistant professor, Dept. of CSDS for her constant encouragement, suggestions, supervision, and abundant support throughout the project.

Thanks to all the teaching and non-teaching staff and lab technicians for their support and also to our team mates for their valuable Co-operation.

By

<b>GADDAM RAGHUVARMA</b>	<b>(Y20CDS016)</b>
<b>PENUMUCHU GRISHMA SRI</b>	<b>(Y20CDS044)</b>
<b>SANKARASETTI SITA</b>	<b>(Y20CDS052)</b>
<b>GANGURU VENKATESWARARAO</b>	<b>(Y20CDS018)</b>

## CONTENTS

S.NO	CHAPTER	PAGE NO
	Abstract	iii
	Problem Statement	iv
1	Back ground work	1
2	System Analysis	2
3	System Requirements	5
4	System Design	6
5	System Implementation	12
6	System Testing	51
7	Conclusion	55

# INDEX

List of Figures	i
Abbreviations	ii
<b>ABSTRACT</b>	<b>iii</b>
<b>PROBLEM STATEMENT</b>	<b>iv</b>
<b>Chapter 1 – Introduction</b>	<b>1</b>
1.1 Background	1
<b>Chapter 2 – System Analysis</b>	<b>2-4</b>
2.1 Existing System	2
2.1.1 Disadvantages	2
2.2 Proposed System	2
2.2.1 Advantages	2
2.3 System Study	2-3
2.3.1 Feasibility Study	3
2.4 Literature Survey	3-4
<b>Chapter 3 – System Requirements</b>	<b>5</b>
3.1 Software Requirements	
3.2 Hardware Requirements	
<b>Chapter 4 - System Design</b>	<b>6-9</b>
4.1 System Architecture	6
4.2 UML Diagrams	6-9
4.2.1 Use Case Diagram	7
4.2.2 Class Diagram	8
4.2.3 Sequence Diagram	9
<b>Chapter 5 – System Implementation</b>	<b>10-37</b>
5.1 System Model	10
5.2 Module Description	10
5.3 Software Environment	10
5.1.1 What is Python?	11
5.1.1.1 Advantages of Python	11-12
5.1.1.2 Advantages of Python over	13

Other Languages	
5.1.1.3 Disadvantages of Python	14
5.1.2 Need for Data Science	17
5.1.3 Challenges in Data Science	18
5.1.4 Applications of Data Science	19-20
5.2.1 How to start Learning Data Science?	20
5.2.1.1 How to start learning Data Science?	20
5.2.1.2 Understand the Prerequisites	20
5.2.1.3 Learn Various Concepts	22
5.2.1.4 Advantages of Data science	25
5.2.1.5 Disadvantages of Data Science	26
5.2.2 How to install Python?	27-36
5.2.3 Jupiter Notebook	37
<b>5.4 Google Colab</b>	<b>38-39</b>
5.5 Results	40-44
5.5.1 Source Code	40
5.5.1.1 Importing Datasets	40
5.5.1.2 Data Preprocessing	40-43
5.5.1.3 Building Model	43-54
<b>Chapter 6 – System Testing</b>	<b>45-48</b>
6.1 Types of Testing	45
6.1.1 Unit Testing	45
6.1.2 Black Box Testing	46
6.1.3 White Box Testing	46-47
6.2 Test Strategy and Approach	47-48
<b>Chapter 7 – Conclusion &amp; Future Scope</b>	<b>49</b>
<b>REFERENCES</b>	<b>50</b>

## LIST OF FIGURES

S. No	Figure. No	CONTENT	Page. No
1	Fig: 4.1	System architecture Diagrams of Flipkart Reviews Sentiment Analysis	6
2	Fig: 4.2	Use Case Diagram for Flipkart Reviews	7
3	Fig: 4.3	Class Diagram for Sentiment Analysis	8
4	Fig: 4.4	Sequence Diagram representing the Flipkart Reviews	9

## **ABBREVIATIONS**

TF-IDF: - TERM FREQUENCY-INVERSE DOCUMENT FREQUENCY

NLP: - NATURAL LANGUAGE PROCESSING

UML: -UNIFIED MODELING LANGUAGE

ML: - MACHINE LEARNING

DL: -DEEP LEARNING



## **ABSTRACT**

The "Flipkart Reviews Sentiment Analysis Using Deep Learning Python Project" uses advanced deep learning techniques to analyze sentiment expressed in customer reviews on Flipkart e-commerce platform. This project aims to provide valuable insights value in terms of customer satisfaction. A model that can classify reviews as positive, negative or neutral is a must have. Leveraging Python and deep learning frameworks, the project focuses on training a sentiment analysis model with a diverse dataset of Flipkart reviews. The abstract emphasizes the application of cutting-edge techniques to enhance the understanding of customer sentiments and improve the overall user experience on the platform. Reading separately evaluations takes a number of time, so what we are able to do is summarize the complete review into three points. For this we can be the use of Sentiment intensity analyzer set of rules. It is greater inexperienced than some other set of rules like visualization or records mining.

**KEYWORDS:** Data Science, Sentiment analysis, opinion mining, reviews, e-commerce, natural language processing, and semantic analysis.

## **PROBLEM STATEMENT**

**E-commerce systems, consisting of Flipkart, collect a giant quantity of purchaser feedback via evaluations. To gain actionable insights into purchaser sentiments, it is vital to expand a robust sentiment evaluation device. The purpose of this mission is to leverage deep studying techniques to create an accurate sentiment category model for Flipkart critiques. The device have to robotically categorize reviews as superb, negative, or impartial, enabling Flipkart to understand customer delight stages, become aware of regions for development, and decorate the overall consumer enjoy. The assignment must address challenges associated with diverse product categories, varying review lengths, and evolving language nuances within purchaser feedback. The final results is anticipated to be an efficient and deployable solution that aids in selection-making and purchaser-centric upgrades on the Flipkart platform.**

# CHAPTER - 1

## INTRODUCTION

### 1.1 BACKGROUND WORK

In the dynamic landscape of e-commerce, customer feedback serves as a valuable repository of insights, offering a direct window into user sentiments. Flipkart, as one of the leading e-commerce platforms, encounters a constant influx of diverse reviews spanning an extensive array of products. Extracting meaningful intelligence from this vast pool of customer feedback is a challenging yet pivotal endeavor.

The "Flipkart Reviews Sentiment Analysis Using Deep Learning Python Project" aims to harness the power of advanced deep learning techniques to decode and classify sentiments embedded within these reviews.

Understanding customer sentiments is not merely a theoretical pursuit but a strategic imperative for Flipkart. By unraveling the emotional tone of user reviews, the project seeks to equip Flipkart with a nuanced understanding of customer satisfaction, uncovering hidden patterns that can drive actionable insights.

The fusion of deep learning methodologies with Python programming promises to create a sophisticated sentiment analysis model capable of discerning sentiments ranging from elation to discontent, and every nuance in between.

This project delves into the intricacies of sentiment analysis, acknowledging the challenges posed by diverse product categories, varying review lengths, and the ever-evolving linguistic nuances within user feedback. Through the application of cutting-edge deep learning techniques, the intent is to develop a robust model that not only accurately classifies sentiments but also adapts to the dynamic nature of language expressions over time.

As we embark on this journey, the overarching goal is to empower Flipkart with a tool that goes beyond sentiment polarity classification. It aspires to be a catalyst for strategic decision-making, aiding in the enhancement of customer experiences, refinement of marketing strategies, and continuous improvement of product offerings.

By bridging the gap between raw textual data and actionable insights, this project aims to contribute significantly to Flipkart's commitment to customer satisfaction and innovation in the competitive e-commerce landscape.

## CHAPTER - 2

### SYSTEM ANALYSIS

#### 2.1. EXISTING SYSTEM

The existing system for sentiment analysis of Flipkart reviews using deep learning employs a multi-step process. Initially, raw text data is collected from Flipkart's review platform, including customer feedback and ratings. The data is then preprocessed to clean and tokenize the text, removing stop words and special characters. Next, a deep learning model such as a recurrent neural network (RNN) or a convolutional neural network (CNN) is trained on this preprocessed data. The model learns to recognize patterns and features in the text that correspond to positive or negative sentiment. Evaluation metrics such as accuracy, precision, recall, and F1 score are used to assess the model's performance. The trained model can then be deployed to predict sentiment for new, unseen reviews, providing valuable insights for product and service improvements on Flipkart's platform.

##### 2.1.1. DISADVANTAGES:

- 1. Data Quality:** The accuracy of sentiment analysis heavily depends on the quality of the training data. Biased or noisy data can lead to inaccurate sentiment predictions.
- 2. Model Complexity:** Deep learning models can be complex and resource-intensive to train, requiring significant computational power and expertise in model architecture selection and hyperparameter tuning.
- 3. Interpretability:** Deep learning models often lack interpretability, making it challenging to understand how the model arrives at its predictions, which can be a concern for decision-making based on sentiment analysis results.
- 4. Domain Adaptation:** Models trained on generic datasets may not perform optimally for specific domains or niche product categories on Flipkart, necessitating domain adaptation techniques or additional training data.
- 5. Maintenance:** Continuous monitoring and updating of the deep learning model are required to ensure its performance remains optimal over time, considering changes in customer preferences and language trends.

#### 2.2. PROPOSED SYSTEM:

The proposed system for sentiment analysis of Flipkart reviews using deep learning involves several key steps. Initially, raw text data from Flipkart reviews will be collected and preprocessed to clean and tokenize the text. Then, a deep learning model such as a recurrent neural network (RNN) or a transformer-based model like BERT will be trained on this preprocessed data to learn sentiment patterns. The model will undergo hyperparameter tuning and evaluation using metrics like accuracy, precision, recall, and F1 score to optimize its performance. Once trained, the model will be deployed to classify the sentiment of new reviews in real-time, providing valuable insights to Flipkart for improving customer experience and product offerings. Regular monitoring and updates to the model will ensure its continued effectiveness in sentiment analysis.

### 2.2.1 ADVANTAGES:

**1. Scalability:** Deep learning models can handle large volumes of data, making them suitable for analyzing thousands or even millions of Flipkart reviews.

**2. Accuracy:** Deep learning models, when properly trained with sufficient data, can achieve high accuracy in sentiment classification, providing reliable insights into customer opinions.

**3. Automated Processing:** Once trained, the model can automate the sentiment analysis process, saving time and effort compared to manual review analysis.

**4. Real-time Insights:** The model can provide real-time sentiment analysis, allowing businesses to promptly address customer concerns or capitalize on positive feedback.

**5. Customization:** Deep learning models can be fine-tuned and customized to specific domains or product categories on Flipkart, enhancing the accuracy of sentiment predictions for different types of reviews.

## 2.3. SYSTEM STUDY

### 2.3.1 FEASIBILITY STUDY :

Certainly! Conducting a feasibility look at Flipkart Reviews Sentiment Analysis The usage of Deep Learning Python task involves assessing numerous aspects:

**Data Availability:** Check if a sufficient amount of classified facts (superb, terrible, impartial critiques) is available for schooling the deep getting-to-know model. Consider scraping Flipkart opinions or using publicly-to-be-read datasets.

**Computational Resources:** Assess the hardware necessities for training deep gaining knowledge of fashions. Ensure that you have admission to a gadget with GPU support for quicker training.

**Expertise:** Evaluate the crew's know-how in deep learning and herbal language processing (NLP). If needed, take into account upskilling or hiring experts in those domain names.

**Tool and Library Availability:** Confirm the availability of required equipment and libraries like TensorFlow or PyTorch for implementing deep studying fashions and NLP libraries like NLTK or spaCy for textual content processing.

**Model Selection:** Choose suitable deep studying architectures for sentiment analysis, such as Recurrent Neural Networks (RNNs) or transformer-based models like BERT. Consider the exchange-offs between model complexity and overall performance.

**Accuracy Expectations:** Set sensible expectations for version accuracy and performance. Determine the appropriate error charge for sentiment predictions.

**Scalability:** Consider the scalability of the version if the project expands. Ensure that the chosen structure can manage an increasing quantity of reviews.

**Project Timeline:** Estimate the time required for facts collection, version education, and trying out. Consider any closing dates or time constraints for the task.

**Cost Analysis:** Assess the overall value of the project, including information acquisition, computational resources, and potential licensing expenses for the use of positive equipment or datasets.

## **2.4.LITERATURE SURVEY :**

A literature survey for a Flipkart reviews sentiment analysis project using deep learning would involve reviewing existing research and projects in the field of sentiment analysis, specifically focusing on applications related to e-commerce platforms like Flipkart. This survey would include studies that explore various deep learning architectures such as recurrent neural networks (RNNs), convolutional neural networks (CNNs), and transformer models like BERT for sentiment analysis tasks. It would also examine different techniques for data preprocessing, feature extraction, and model optimization used in similar projects. Additionally, the survey would cover research on domain adaptation and transfer learning approaches to tailor sentiment analysis models for specific domains like product reviews on Flipkart. Analyzing the strengths, limitations, and outcomes of previous projects in this area would provide valuable insights and guidance for designing and implementing an effective sentiment analysis system for Flipkart reviews using deep learning techniques.

## **CHAPTER - 3**

### **SYSTEM REQUIREMENTS**

#### **1.1 SOFTWARE REQUIREMENTS**

The functional requirements or the overall description documents include the product perspective and features, operating system and operating environment, graphics requirements, design constraints and user documentation.

The appropriation of requirements and implementation constraints gives the general overview of the project in regards to what the areas of strength and deficit are and how to tackle them.

- **Python idle 3.7 version (or) • Jupiter (or) • Google Collab**

#### **1.2 HARDWARE REQUIREMENTS**

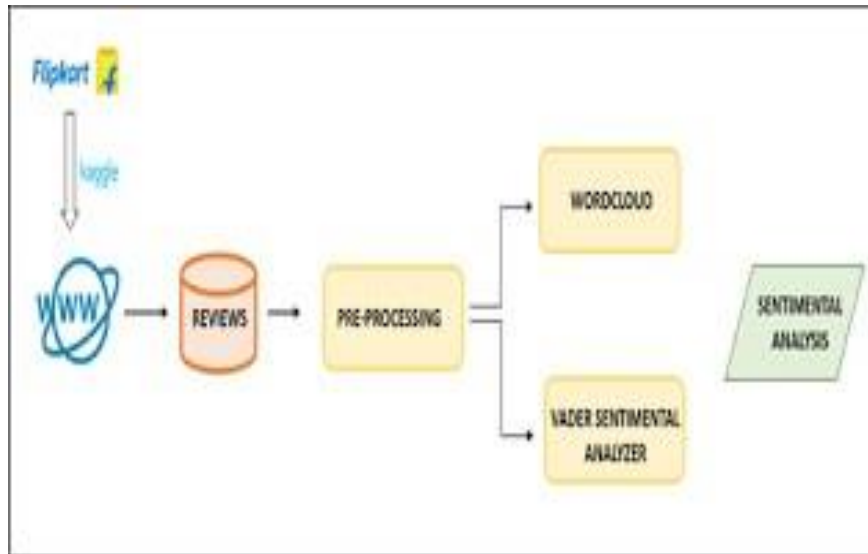
Minimum hardware requirements are very dependent on the particular software being developed by a given Enthought Python / Google collab. Applications that need to store large arrays/objects in memory will require more RAM, whereas applications that need to perform numerous calculations or tasks more quickly will require a faster processor.

- |                           |                           |
|---------------------------|---------------------------|
| • <b>Operating system</b> | <b>: windows, Linux</b>   |
| • <b>Processor</b>        | <b>: minimum intel i3</b> |
| • <b>Ram</b>              | <b>: minimum 4gb</b>      |
| • <b>Hard disk</b>        | <b>: minimum 250gb</b>    |

## CHAPTER - 4

### SYSTEM DESIGN

#### 4.1. SYSTEM ARCHITECTURE



**Fig 4.1 System Architecture for Movie Genre classification.**

#### 4.2. UML DIAGRAMS :

UML remains for Unified Modeling Language. UML is an institutionalized broadly useful displaying dialect in the field of protest situated programming designing. The standard is overseen, and was made by the Object Management Group. The objective is for UML to end up a typical dialect for making models of protest arranged PC programming. In its present shape UML is contained two noteworthy segments: a Meta-display and a documentation. Later on, some type of technique or process may likewise be added to; or connected with

The Unified Modeling Language is a standard dialect for indicating, Visualization, Constructing and recording the antiques of programming framework, and additionally for business displaying and other non-programming frameworks.

The UML speaks to an accumulation of best building rehearses that have demonstrated effective in the displaying of vast and complex frameworks.

The UML is an imperative piece of creating articles arranged programming and the product improvement prepare. The UML utilizes for the most part graphical documentations to express the plan of programming tasks.



**Goals:**

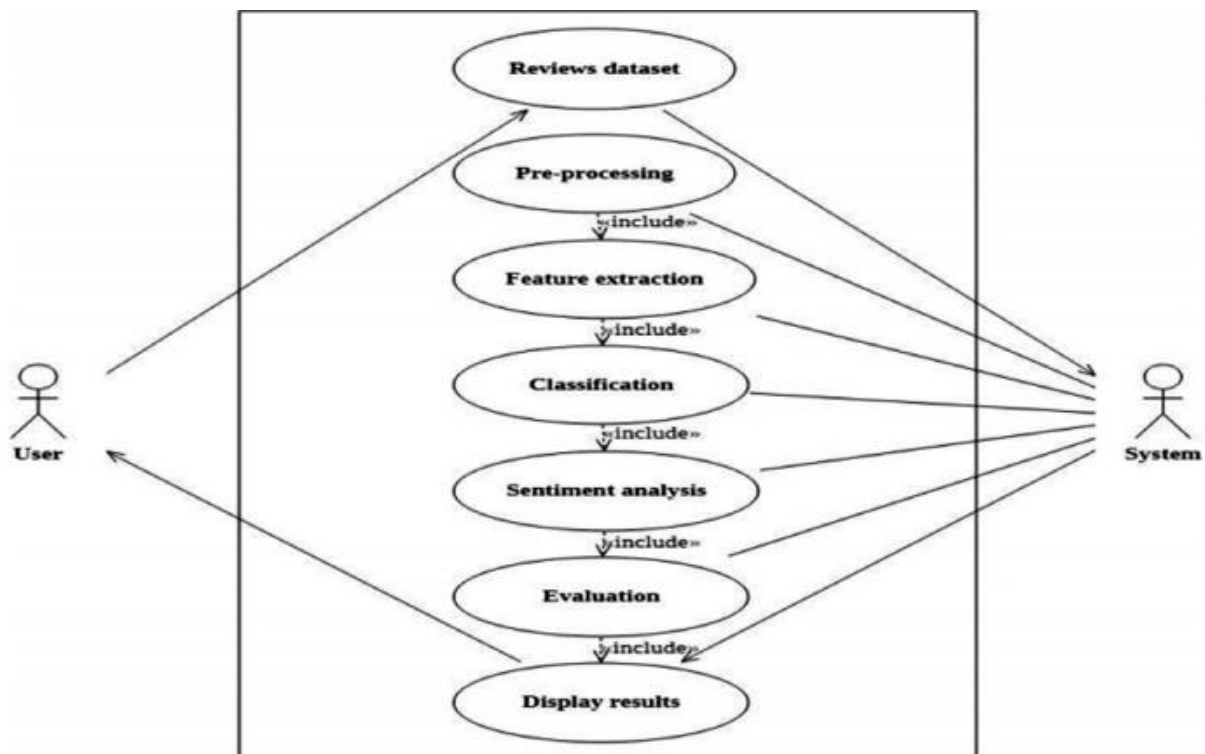
The Primary objectives in the plan of the UML are as per the following:

1. Provide clients a prepared-to-utilize, expressive visual displaying Language with the goal that they can create and trade important models.
2. Provide extendibility and specialization instruments to develop the center ideas.
3. Be free of specific programming dialects and improvement handle.
4. Provide a formal reason for comprehension the displaying dialect.
5. Encourage the development of OO devices showcase.
6. Support more elevated amount improvement ideas, for example, coordinated efforts, systems, examples and parts.
7. Integrate best practices.

**4.2.1 USE CASE DIAGRAM :**

A use case diagram within the unified modeling language (UML) may be a kind of activity diagram outlined by and created from a use-case analysis. Its purpose is to gift a graphical summary of the practicality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. the most purpose of a use case diagram is to indicate what system functions area unit performed that user.

Roles of the users within the system is represented.

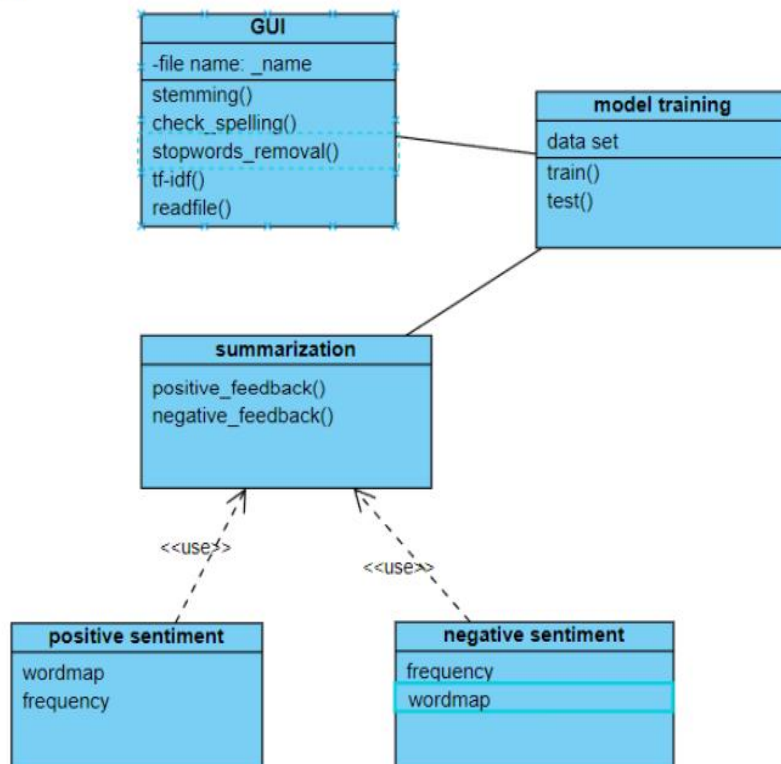


**Fig: 4.2. Use Case Diagram For Sentiment Analysis.**

#### 4.2.2 CLASS DIAGRAM :

In computer code engineering, a category diagram within the Unified Modeling Language (UML) may be a kind of static structure diagram that describes the structure of a system by showing the system's categories, their attributes, operations (or methods), and also the relationships among the categories. It explains that category contains data.

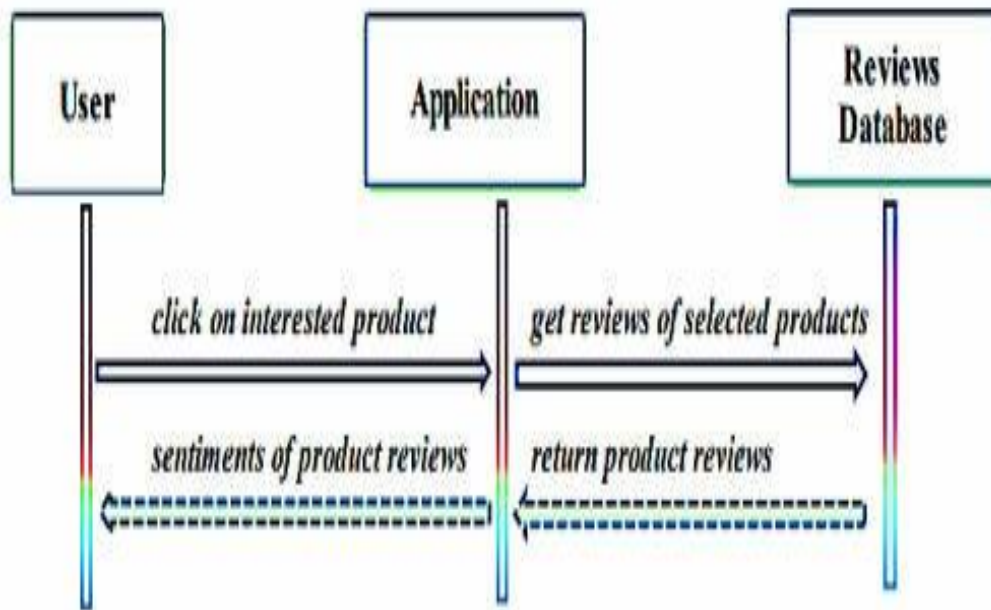
Class Diagram



**Fig: 4.3. Class Diagram for Sentiment Analysis**

### 4.2.3 SEQUENCE DIAGRAM:

A sequence diagram in Unified Modeling Language (UML) may be a quite interaction diagram that shows how processes operate with each other and in what order. It's a construct of a Message Sequence Chart. Sequence diagrams are generally known as event diagrams, event situations, and temporal order diagrams.



**Fig: 4.4. Sequence Diagram For Flipkart Reviews.**

## **CHAPTER - 5**

### **SYSTEM IMPLEMENTATION**

#### **1.1. SYSTEM MODEL**

Here collect 89 queries issued by the subjects, and name them as “User Q”. As this approach might induce a bias towards topics in which lists are more useful than general web queries, we further randomly sample another set of 105 English queries from a query log of a commercial search engine, and name this set of queries as “Rand Q”. We first ask a subject to manually create facets and add items that are covered by the query, based on his/her knowledge after a deep survey on any related resources (such as Wikipedia, Freebase, or official web sites related to the query).

#### **1.2. MODULE DESCRIPTION**

- 1) Question Module: The question module is responsible for parsing user queries, identifying the user's intent, and extracting relevant information. It utilizes natural language processing techniques to understand the context and structure of the question, enabling the chatbot to generate appropriate responses. Additionally, the question module may incorporate machine learning algorithms to improve accuracy and adapt to user input over time.
- 2) Answer Module: Using this answer module generates responses based on the input received from the user, utilizing predefined rules, templates, or machine learning algorithms. It selects the most appropriate response considering the context, user intent, and available information. Additionally, the answer module may incorporate natural language generation techniques to create human-like responses for improved user interaction.

### **5.3 SOFTWARE ENVIRONMENT**

#### **5.1.1. What is Python: -**

Below are some facts about Python. Python is currently the most widely used multi-purpose, high-level programming language. Python allows programming in Object-Oriented and

Procedural paradigms. Python programs generally are smaller than other programming languages like Java.

Programmers have to type relatively less and indentation requirement of the language, makes them readable all the time.

Python language is being used by almost all tech-giant companies like – Google, Amazon, Facebook, Instagram, Dropbox, Uber... etc.

The biggest strength of Python is huge collection of standard libraries which can be used for the following –

Data Science

GUI Applications (like [Kivy](#), Tkinter, PyQt, etc. )

Web frameworks like [Django](#) (used by YouTube, Instagram, Dropbox) Image processing (like [OpenCV](#), Pillow)

Web scraping (like Scrapy, BeautifulSoup, Selenium) Test frameworks

Multimedia

#### **5.1.1.1 Advantages of Python: -**

Let's see how Python dominates over other languages.

##### **1.1.1.1 1. Extensive Libraries**

Python downloads with an extensive library and it *contain code for various purposes like regular expressions, documentation-generation, unit-testing, web browsers, threading, databases, CGI, email, image manipulation, and more*. So, we don't have to write the complete code for that manually.

##### **1.1.1.2 2. Extensible**

As we have seen earlier, Python can be **extended to other languages**. You can write some of your code in languages like C++ or C. This comes in handy, especially in projects.

##### **1.1.1.3 3. Embeddable**

Complimentary to extensibility, Python is embeddable as well. You can put your Python code in your source code of a different language, like C++. This lets us add **scripting capabilities** to our code in the other language.

#### **1.1.1.4 4. Improved Productivity**

The language's simplicity and extensive libraries render programmers **more productive** than languages like Java and C++ do. Also, the fact that you need to write less and get more things done.

#### **1.1.1.5 5. IOT Opportunities**

Since Python forms the basis of new platforms like Raspberry Pi, it finds the future bright for the Internet of Things. This is a way to connect the language with the real world.

#### **1.1.1.6 6. Simple and Easy**

When working with Java, you may have to create a class to print '**Hello World**'. But in Python, just a print statement will do. It is also quite **easy to learn, understand, and code**. This is why when people pick up Python, they have a hard time adjusting to other more verbose languages like Java.

#### **1.1.1.7 7. Readable**

Because it is not such a verbose language, reading Python is much like reading English. This is the reason why it is so easy to learn, understand, and code. It also does not need curly braces to define blocks, and **indentation is mandatory**. This further aids the readability of the code.

#### **1.1.1.8 8. Object-Oriented**

This language supports both the **procedural and object-oriented** programming paradigms. While functions help us with code reusability, classes and objects let us model the real world. A class allows the **encapsulation of data** and functions into one.

#### **1.1.1.9 9. Free and Open-Source**

Like we said earlier, Python is **freely available**. But not only can you **download Python** for free, but you can also download its source code, make changes to it, and even distribute it. It downloads with an extensive collection of libraries to help you with your tasks.

#### **1.1.1.10 10. Portable**

When you code your project in a language like C++, you may need to make some changes to it if you want to run it on another platform. But it isn't the same with Python. Here, you need to **code only once**, and you can run it anywhere. This is called **Write**

**Once Run Anywhere (WORA).** However, you need to be careful enough not to include any system-dependent features.

#### **1.1.1.11 11. Interpreted**

Lastly, we will say that it is an interpreted language. Since statements are executed one by one, **debugging is easier** than in compiled languages.

*Any doubts till now in the advantages of Python? Mention in the comment section.*

### **5.1.1.2 Advantages of Python Over Other Languages**

#### **1.1.2.1 1. Less Coding**

Almost all of the tasks done in Python requires less coding when the same task is done in other languages. Python also has an awesome standard library support, so you don't have to search for any third-party libraries to get your job done. This is the reason that many people suggest learning Python to beginners.

#### **1.1.2.2 2. Affordable**

Python is free therefore individuals, small companies or big organizations can leverage the free available resources to build applications. Python is popular and widely used so it gives you better community support.

**The 2019 Git hub annual survey showed us that Python has overtaken Java in the most popular programming language category.**

#### **1.1.2.3 3. Python is for Everyone**

Python code can run on any machine whether it is Linux, Mac or Windows. Programmers need to learn different languages for different jobs but with Python, you can professionally build web apps, perform data analysis and **machine learning**, automate things, do web scraping and also build games and powerful visualizations. It is an all-rounder programming language.

### **5.1.1.3 Disadvantages of Python**

So far, we've seen why Python is a great choice for your project. But if you choose it, you should be aware of its consequences as well. Let's now see the downsides of choosing Python over another language.

### 1.1.3.1 1. Speed Limitations

We have seen that Python code is executed line by line. But since [Python](#) is interpreted, it often results in **slow execution**. This, however, isn't a problem unless speed is a focal point for the project. In other words, unless high speed is a requirement, the benefits offered by Python are enough to distract us from its speed limitations.

### 1.1.3.2 2. Weak in Mobile Computing and Browsers

While it serves as an excellent server-side language, Python is much rarely seen on the **client-side**. Besides that, it is rarely ever used to implement smartphone-based applications. One such application is called **Carbonnelle**.

The reason it is not so famous despite the existence of python is that it isn't that secure.

### 1.1.3.3 3. Design Restrictions

As you know, Python is **dynamically-typed**. This means that you don't need to declare the type of variable while writing the code. It uses **duck-typing**. But wait, what's that? Well, it just means that if it looks like a duck, it must be a duck. While this is easy on the programmers during coding, it can **raise run-time errors**.

### 1.1.3.4 4. Underdeveloped Database Access Layers

Compared to more widely used technologies like **JDBC (Java Data Base Connectivity)** and **ODBC (Open Data Base Connectivity)**, Python's database access layers are a bit underdeveloped. Consequently, it is less often applied in huge enterprises.

### 1.1.3.5 5. Simple

No, we're not kidding. Python's simplicity can indeed be a problem. Take my example.

I don't do Java, I'm more of a Python person. To me, its syntax is so simple that the verbosity of Java code seems unnecessary.

This was all about the Advantages and Disadvantages of Python Programming Language.

### History Of Python: -

What do the alphabet and the programming language Python have in common? Right, both start with ABC. If we are talking about ABC in the Python context, it's clear that the programming language ABC is meant. ABC is a general-purpose programming language and programming environment, which had been developed in the Netherlands, Amsterdam, at the CWI (Centrum Wickenden & Informatica). The greatest achievement of ABC was to influence the design of Python. Python was conceptualized in the late



1980s. Guido van Rossum worked that time in a project at the CWI, called Amoeba, a distributed operating system. In an interview with Bill Venner<sup>1</sup>, Guido van Rossum said: "In the early 1980s, I worked as an implementer on a team building a language called ABC at Centrum voor Wiskunde en Informatica (CWI). I don't know how well people know ABC's influence on Python. I try to mention ABC's influence because I'm indebted to everything I learned during that project and to the people who worked on it." Later on in the same Interview, Guido van Rossum continued: "I remembered all my experience and some of my frustration with ABC. I decided to try to design a simple scripting language that possessed some of ABC's better properties, but without its problems. So I started typing. I created a simple virtual machine, a simple parser, and a simple runtime. I made my own version of the various ABC parts that I liked. I created a basic syntax, used indentation for statement grouping instead of curly braces or begin- end blocks, and developed a small number of powerful data types: a hash table (or dictionary, as we call it), a list, strings, and numbers."

### **What Is Data Science: -**

Data science is an interdisciplinary field that utilizes scientific methods, processes, algorithms, and systems to extract insights and knowledge from structured and unstructured data. It combines elements of statistics, mathematics, computer science, and domain-specific knowledge to analyze complex datasets and uncover patterns, trends, and relationships.

The primary goal of data science is to generate actionable insights and make data-driven decisions. This involves various stages, including data collection, cleaning, preprocessing, analysis, interpretation, and visualization. Data scientists use a combination of techniques such as statistical modeling, machine learning, data mining, and data visualization to extract valuable information from data.

Data science finds applications across diverse domains, including business, healthcare, finance, marketing, social sciences, and more. It plays a crucial role in areas such as predictive analytics, risk management, recommendation systems, fraud detection, and optimization.

Overall, data science serves as a powerful tool for transforming raw data into meaningful insights, driving innovation, and facilitating evidence-based decision-making in both research and industry.

### **Categories of Data Science:**

Data science encompasses various categories or subfields, each focusing on different aspects of data analysis and application. Some common categories of data science include:

1. **Descriptive Analytics:** Descriptive analytics involves the exploration and summarization of historical data to understand what has happened in the past. It includes techniques such as data visualization, summary statistics, and exploratory data analysis (EDA).

2. **Predictive Analytics:** Predictive analytics aims to forecast future outcomes or trends based on historical data. This category utilizes statistical modeling, machine learning algorithms, and data mining techniques to build predictive models that can make informed predictions.

3. **Prescriptive Analytics:** Prescriptive analytics focuses on recommending actions or decisions based on insights derived from data analysis. It goes beyond predicting outcomes by providing recommendations on the best course of action to achieve desired goals. Optimization techniques and decision-making algorithms are commonly used in prescriptive analytics.

4. **Machine Learning:** Machine learning is a subset of data science that involves teaching computers to learn from data and make predictions or decisions without being explicitly programmed. It includes various techniques such as supervised learning, unsupervised learning, and reinforcement learning.

5. **Natural Language Processing (NLP):** NLP is a branch of data science that deals with the interaction between computers and human language. It includes tasks such as text classification, sentiment analysis, language translation, and information extraction from text data.

6. **Computer Vision:** Computer vision is another subfield of data science focused on enabling computers to interpret and understand visual information from images or videos. It involves tasks such as image classification, object detection, image segmentation, and facial recognition.

7. **Big Data Analytics:** Big data analytics involves processing and analyzing large volumes of data (often referred to as big data) to extract insights and value. It deals with challenges related to data storage, processing, and analysis in distributed computing environments.

8. **Data Engineering:** Data engineering focuses on designing and building the infrastructure and systems needed to collect, store, process, and manage data effectively. It includes tasks such as data integration, data warehousing, data pipelines, and data architecture design.

These categories are not mutually exclusive and often overlap, with data scientists leveraging techniques from multiple categories to solve complex problems and extract insights from data.

### 5.1.2 Need for Data Science

The need for data science arises from the increasing volume, variety, and velocity of data generated in today's digital world. Several factors contribute to the growing demand for data science:

**1. Explosion of Data:** With the proliferation of digital technologies, organizations are generating vast amounts of data from various sources such as social media, sensors, mobile devices, and web applications. This abundance of data presents opportunities to extract valuable insights and drive informed decision-making.

**2. Complexity of Data:** Data comes in various forms, including structured, semi-structured, and unstructured data. Traditional data analysis methods struggle to handle the complexity and scale of modern datasets. Data science techniques such as machine learning and artificial intelligence are essential for analyzing and extracting insights from diverse and complex data sources.

**3. Competitive Advantage:** In today's competitive landscape, organizations seek to gain a competitive edge by leveraging data-driven insights to optimize processes, improve products and services, and enhance customer experiences. Data science enables organizations to uncover hidden patterns, trends, and correlations in data that can drive innovation and business growth.

**4. Decision-Making Support:** Data science provides decision-makers with valuable insights and evidence-based recommendations to guide strategic planning, resource allocation, risk management, and performance optimization. By making data-driven decisions, organizations can mitigate risks, seize opportunities, and achieve better outcomes.

**5. Personalization and Customer Engagement:** Data science enables organizations to personalize products, services, and marketing campaigns based on individual preferences, behaviors, and needs. By analyzing customer data, organizations can deliver targeted and relevant experiences that enhance customer satisfaction and loyalty.

**6. Cost Reduction and Efficiency Improvement:** Data science techniques such as predictive analytics and optimization can help organizations identify inefficiencies, streamline processes, and reduce costs. By optimizing resource allocation, supply chain management, and operations, organizations can improve efficiency and profitability.

**7. Innovation and Research Advancement:** Data science drives innovation and advances in various fields, including healthcare, finance, transportation, energy, and environmental science. By analyzing large datasets and conducting data-driven research, scientists and researchers can make breakthrough discoveries, develop new technologies, and address complex challenges.

Overall, data science plays a crucial role in enabling organizations to harness the power of data, unlock insights, and drive value creation in today's data-driven world.

### 5.1.3 Challenges in Data Science: -

Data science, despite its immense potential, faces several challenges that can impact its effectiveness and implementation. Some of the key challenges include:

1. **Data Quality and Reliability:** One of the fundamental challenges in data science is ensuring the quality and reliability of data. Issues such as missing values, inaccuracies, inconsistencies, and biases in the data can lead to erroneous results and unreliable insights.
2. **Data Privacy and Security:** Data privacy and security concerns are significant challenges in data science, particularly with the increasing amount of personal and sensitive data being collected and analyzed. Protecting data against unauthorized access, breaches, and misuse while adhering to privacy regulations poses significant challenges for data scientists.
3. **Data Volume and Complexity:** The sheer volume and complexity of data being generated pose challenges for data scientists in terms of data storage, processing, and analysis. Handling large-scale datasets requires robust infrastructure, efficient algorithms, and scalable solutions.
4. **Data Integration and Compatibility:** Integrating data from disparate sources with varying formats, structures, and semantics is a common challenge in data science. Ensuring compatibility and consistency across different datasets and systems can be complex and time-consuming.
5. **Lack of Domain Expertise:** Data scientists often face challenges in understanding and interpreting domain-specific knowledge and context when analyzing data. Collaboration with domain experts is essential to ensure the relevance and accuracy of insights derived from data analysis.
6. **Model Interpretability and Explainability:** With the increasing complexity of machine learning models, ensuring their interpretability and explainability is a challenge. Understanding how models make predictions and explaining their decisions is crucial for gaining trust and acceptance in critical applications such as healthcare and finance.
7. **Bias and Fairness:** Addressing bias and ensuring fairness in data and algorithms is a significant challenge in data science. Biases in data collection, sampling, and modeling can lead to unfair outcomes and discriminatory practices, raising ethical concerns.
8. **Scalability and Performance:** Scaling data science solutions to handle large-scale datasets and real-time processing requirements can be challenging. Ensuring the scalability, efficiency, and performance of algorithms and systems is crucial for addressing the growing demands of data-driven applications.

**9. Continuous Learning and Adaptation:** Data science is a rapidly evolving field, with new techniques, algorithms, and technologies emerging constantly. Keeping pace with advancements and continuously updating skills and knowledge is essential for data scientists to remain effective and relevant.

Addressing these challenges requires a holistic approach that involves a combination of technical expertise, domain knowledge, ethical considerations, and collaboration across disciplines. By overcoming these challenges, data scientists can unlock the full potential of data and drive meaningful insights and innovations.

#### **5.1.4 Applications of Data Science: -**

Data science finds applications across various domains, driving innovation, optimization, and decision-making processes. In business and marketing, data science is utilized for customer segmentation, predictive modeling, recommendation systems, and personalized marketing campaigns.

1. Business and Marketing
2. Finance
3. Healthcare
4. Transportation and Logistics
5. Social Media and Entertainment
6. Energy Management and Environmental Monitoring
7. Urban Planning
8. Sports Analytics
9. Cybersecurity

#### **1.2.1. How to Start Learning Data Science?**

The term "data science" has evolved over time, and its exact origins are difficult to pinpoint to a single founder or individual. However, data science as a discipline emerged from the convergence of various fields such as statistics, computer science, mathematics, and domain expertise. The **“Hal Varian”** Chief Economist at Google and an influential figure

in the field of data science. Varian's work on using data to understand economic behavior and decision-making has contributed to the advancement of data-driven approaches in various domains. Introduction:

Data science has emerged as a crucial field in today's data-driven world, offering exciting opportunities for those interested in extracting insights from data to drive decision-making and innovation. It combines elements of statistics, mathematics, computer science, and domain expertise to analyze complex datasets and uncover patterns, trends, and relationships.

As organizations across industries increasingly rely on data to gain a competitive edge and solve complex problems, the demand for skilled data scientists continues to grow. Learning data science opens doors to a wide range of career opportunities in fields such as business analytics, finance, healthcare, marketing, and more.

Whether you're a beginner or looking to advance your skills in data science, getting started requires a structured approach and dedication to continuous learning. With the right resources, tools, and mindset, anyone can embark on the journey to becoming a proficient data scientist.

In the following sections, we'll outline steps to help you start learning data science, from gaining foundational knowledge in relevant concepts to acquiring hands-on experience with real-world projects and datasets.

#### **1.2.1.1 How to start learning Data Science?**

This is a rough roadmap you can follow on your way to becoming an insanely talented Data Science Engineer. Of course, you can always modify the steps according to your needs to reach your desired end-goal!

#### **1.2.1.2 Step 1 – Understand the Prerequisites**

Familiarize yourself with foundational concepts in mathematics (such as linear algebra, calculus, and probability), statistics, and programming (particularly in languages like Python or R).

.

#### **1.2.1.2.1 (a) Learn Programming:**

Master programming languages commonly used in data science such as Python or R. Focus on libraries and frameworks like NumPy, pandas, matplotlib, and scikit-learn for Python, or dplyr, ggplot2, and caret for R.

#### **1.2.1.2.2 (b) Explore Data Analysis and Visualization:**

Learn techniques for data manipulation, exploration, and visualization using tools like pandas, matplotlib, and seaborn in Python, or ggplot2 and ggvis in R. Practice analyzing datasets to derive insights and communicate findings effectively.

#### **1.2.1.2.3 (c) Study Statistics and Probability:**

Gain a solid understanding of statistical methods and probability theory, including hypothesis testing, regression analysis, probability distributions, and inferential statistics.

#### **1.2.1.2.4 (d) Dive into Machine Learning:**

Explore the fundamentals of machine learning algorithms and techniques, including supervised learning (e.g., regression, classification), unsupervised learning (e.g., clustering, dimensionality reduction), and model evaluation. Implement algorithms using libraries like scikit-learn in Python or caret in R.

#### **1.2.1.2.5 (e) Work on Projects:**

Apply your knowledge by working on data science projects. Start with simple projects and gradually tackle more complex problems. Kaggle, GitHub, and online courses often offer datasets and project ideas to practice your skills.

#### **1.2.1.2.6 (f) Learn Data Wrangling and Preprocessing:**

Understand the importance of data cleaning, preprocessing, and feature engineering in the data science workflow. Practice techniques for handling missing values, outliers, and categorical variables.

#### **1.2.1.2.7 (g) Explore Deep Learning:**

Familiarize yourself with deep learning concepts and frameworks like TensorFlow or PyTorch if you're interested in advanced topics like neural networks and deep learning.

#### **1.2.1.2.8 (h) Take Online Courses and Tutorials:**

Enroll in online courses and tutorials offered by platforms like Coursera, edX, Udemy, or DataCamp, which provide structured learning paths and hands-on exercises in data science.

#### **1.2.1.2.9 (i) Join Communities and Forums:**

Participate in data science communities and forums such as Stack Overflow, Reddit (e.g., r/datascience), or LinkedIn groups to learn from experienced practitioners, ask questions, and stay updated on industry trends.

#### **1.2.1.2.10 (j) Practice Problem-Solving:**

Regularly practice solving data science problems and participate in competitions like Kaggle competitions to hone your skills, learn new techniques, and benchmark your progress against others.

#### **1.2.1.2.11 (k) Build a Portfolio:**

Showcase your projects, skills, and achievements by building a portfolio. A portfolio is an excellent way to demonstrate your proficiency to potential employers and establish credibility in the field.

Data science has emerged as a crucial field in today's data-driven world, offering exciting opportunities for those interested in extracting insights from data to drive decision-making and innovation. It combines elements of statistics, mathematics, computer science, and domain expertise to analyze complex datasets and uncover patterns, trends, and relationships.

As organizations across industries increasingly rely on data to gain a competitive edge and solve complex problems, the demand for skilled data scientists continues to grow. Learning data science opens doors to a wide range of career opportunities in fields such as business analytics, finance, healthcare, marketing, and more.

Whether you're a beginner or looking to advance your skills in data science, getting started requires a structured approach and dedication to continuous learning. With the right resources, tools, and mindset, anyone can embark on the journey to becoming a proficient data scientist.

In the following sections, we'll outline steps to help you start learning data science, from gaining foundational knowledge in relevant concepts to acquiring hands-on experience with real-world projects and datasets.

### **1.2.1.3 Step 2 – Learn Various Data Science Concepts**

Now that you are done with the prerequisites, you can move on to actually learning DS

(Which is the fun part!!!) It's best to start with the basics and then move on to the more complicated stuff. Some of the basic concepts in Data Science are:

#### **1.2.1.3.1 (a) Terminologies of Data Science**

In the field of data science, several terminologies are commonly used to describe concepts, techniques, and methodologies. Some key terminologies include:

1. **Data:** Raw facts, observations, or measurements that are collected and stored for analysis.



2. **Data Analysis:** The process of inspecting, cleaning, transforming, and modeling data to extract useful information and insights.
3. **Data Mining:** The process of discovering patterns, trends, and relationships in large datasets using automated methods and algorithms.
4. **Machine Learning:** A subset of artificial intelligence (AI) that involves training algorithms to learn patterns from data and make predictions or decisions without being explicitly programmed.
5. **Statistical Analysis:** The use of statistical techniques to analyze and interpret data, including methods for hypothesis testing, estimation, and inference.
6. **Predictive Analytics:** The practice of using historical data to make predictions about future events or trends.
7. **Descriptive Analytics:** The analysis of historical data to understand what has happened in the past, including techniques such as data visualization and summary statistics.
8. **Prescriptive Analytics:** The analysis of data to determine the best course of action to achieve a desired outcome or goal.
9. **Big Data:** Large and complex datasets that cannot be processed using traditional data processing methods.
10. **Data Visualization:** The graphical representation of data to communicate information and insights effectively.
11. **Feature Engineering:** The process of selecting, transforming, and creating features from raw data to improve the performance of machine learning models.
12. **Model Evaluation:** The process of assessing the performance of machine learning models using metrics such as accuracy, precision, recall, and F1-score.
13. **Overfitting:** A phenomenon where a machine learning model learns to fit the training data too closely, resulting in poor generalization to new data.
14. **Underfitting:** A phenomenon where a machine learning model is too simple to capture the underlying patterns in the data, resulting in poor performance on both training and test datasets.
15. **Cross-Validation:** A technique used to assess the performance of machine learning models by splitting the data into multiple subsets for training and testing.

16. **Feature Selection:** The process of selecting a subset of relevant features from a larger set of features to improve the performance and efficiency of machine learning models.

17. **Ensemble Learning:** A technique that combines multiple machine learning models to improve prediction accuracy and robustness.

18. **Clustering:** A method of unsupervised learning that involves grouping similar data points together into clusters based on their characteristics.

19. **Regression:** A method of supervised learning that involves predicting a continuous outcome variable based on one or more input features.

20. **Classification:** A method of supervised learning that involves predicting a categorical outcome variable based on one or more input features.

These terminologies are foundational concepts in data science and are essential for understanding and communicating ideas within the field.

#### **1.2.1.3.2 (b) Types of Data Science**

1. **Descriptive Analytics:** Descriptive analytics involves the exploration and summarization of historical data to understand what has happened in the past. It includes techniques such as data visualization, summary statistics, and exploratory data analysis (EDA).

2. **Predictive Analytics:** Predictive analytics aims to forecast future outcomes or trends based on historical data. This category utilizes statistical modeling, machine learning algorithms, and data mining techniques to build predictive models that can make informed predictions.

3. **Prescriptive Analytics:** Prescriptive analytics focuses on recommending actions or decisions based on insights derived from data analysis. It goes beyond predicting outcomes by providing recommendations on the best course of action to achieve desired goals. Optimization techniques and decision-making algorithms are commonly used in prescriptive analytics.

4. **Machine Learning:** Machine learning is a subset of data science that involves teaching computers to learn from data and make predictions or decisions without being explicitly programmed. It includes various techniques such as supervised learning, unsupervised learning, and reinforcement learning.

5. **Natural Language Processing (NLP):** NLP is a branch of data science that deals with the interaction between computers and human language. It includes tasks such as text

classification, sentiment analysis, language translation, and information extraction from text data.

**6. Computer Vision:** Computer vision is another subfield of data science focused on enabling computers to interpret and understand visual information from images or videos. It involves tasks such as image classification, object detection, image segmentation, and facial recognition.

**7. Big Data Analytics:** Big data analytics involves processing and analyzing large volumes of data (often referred to as big data) to extract insights and value. It deals with challenges related to data storage, processing, and analysis in distributed computing environments.

**8. Data Engineering:** Data engineering focuses on designing and building the infrastructure and systems needed to collect, store, process, and manage data effectively. It includes tasks such as data integration, data warehousing, data pipelines, and data architecture design.

These types are not mutually exclusive and often overlap, with data scientists leveraging techniques from multiple types to solve complex problems and extract insights from data.

#### **1.2.1.4 Advantages of Machine learning:**

**1.2.1.4.1 1. Data-Driven Decision Making:** Data science enables organizations to make informed decisions based on evidence and insights extracted from data, leading to more accurate and effective decision-making processes.

**5.2.1.4.1 2.Business Insights and Innovation:** By analyzing large volumes of data, organizations can gain valuable insights into customer behavior, market trends, and business operations, facilitating innovation and competitive advantage.

**5.2.1.4.1 3.Improved Efficiency and Productivity:** Data science techniques such as process optimization and automation help streamline workflows, reduce inefficiencies, and improve overall productivity within organizations.

**5.2.1.4.1 4.Personalized Customer Experiences:** Data science enables organizations to personalize products, services, and marketing campaigns based on individual preferences and behaviors, leading to enhanced customer satisfaction and loyalty.

**5.2.1.4.1 5.Risk Mitigation and Fraud Detection:** Data science techniques such as predictive analytics and anomaly detection help organizations identify and mitigate risks, detect fraudulent activities, and ensure compliance with regulations.

**5.2.1.4.1 6. Cost Reduction:** By optimizing resource allocation, supply chain management, and operational processes, data science helps organizations reduce costs, increase efficiency, and improve profitability.

**5.2.1.4.1 7. Scientific Advancements:** In fields such as healthcare, environmental science, and genomics, data science drives scientific advancements by enabling researchers to analyze complex datasets, make discoveries, and develop new treatments and technologies.

### **1.2.1.5 Disadvantages of Data Science:**

**1.2.1.5.1 1. Data Quality Issues:** Data science relies heavily on the quality and reliability of data. Issues such as missing values, inaccuracies, and biases in the data can lead to erroneous results and unreliable insights.

**5.2.1.5.1 2. Data Privacy and Security Concerns:** The collection, storage, and analysis of large volumes of data raise concerns about data privacy and security. Organizations must ensure that data is protected against unauthorized access, breaches, and misuse.

**5.2.1.5.1 3. Complexity and Scalability:** Data science projects often involve complex algorithms, technologies, and infrastructure requirements. Scaling data science solutions to handle large-scale datasets and real-time processing can be challenging and resource-intensive.

**5.2.1.5.1 4. Ethical Considerations:** Data science raises ethical questions related to data privacy, fairness, transparency, and accountability. Organizations must consider the ethical implications of their data science initiatives and ensure responsible use of data.

**5.2.1.5.1 5. Skill Shortages and Talent Gap:** There is a growing demand for skilled data scientists, analysts, and engineers, leading to a talent gap in the field. Recruiting and retaining top data science talent can be difficult for organizations.

**5.2.1.5.1 6. Interpretability and Explainability:** With the increasing complexity of machine learning models, ensuring their interpretability and explainability is a challenge. Understanding how models make predictions and explaining their decisions is crucial for gaining trust and acceptance.

**5.2.1.5.1 7. Implementation Challenges:** Integrating data science solutions into existing systems and processes can be challenging for organizations. Resistance to change, lack of organizational buy-in, and cultural barriers may hinder the successful implementation of data science initiatives.

### **Python Development Steps: -**

Guido Van Rossum published the first version of Python code (version 0.9.0) at alt. sources in February 1991. This release included already exception handling, functions, and the core data types of lists, dict, str and others. It was also object oriented and had a module system. Python version 1.0 was released in January 1994. The major new features included in this release were the functional programming tools lambda, map, filter and reduce, which Guido Van Rossum never liked. Six and a half years later in October 2000, Python 2.0 was introduced. This release included list comprehensions, a full garbage collector and it was supporting Unicode. Python flourished for another 8 years in the versions 2.x before the next major release as Python 3.0 (also known as "Python 3000" and "Py3K") was released. Python 3 is not backwards compatible with Python 2.x. The emphasis in Python 3 had been on the removal of duplicate programming constructs and modules, thus fulfilling or coming close to fulfilling the 13th law of the Zen of Python: "There should be one -- and preferably only one -- obvious way to do it." Some changes in Python 7.3:

- Print is now a function
- Views and iterators instead of lists
- The rules for ordering comparisons have been simplified. E.g. a heterogeneous list cannot be sorted, because all the elements of a list must be comparable to each other.
- There is only one integer type left, i.e. int. long is int as well.
- The division of two integers returns a float instead of an integer. "/" can be used to have the "old" behaviour.
- Text Vs. Data Instead of Unicode Vs. 8-bit

### **Purpose: -**

We demonstrated that our approach enables successful segmentation of intra-retinal layers—even with low-quality images containing speckle noise, low contrast, and different intensity ranges throughout—with the assistance of the ANIS feature.

### **Python**

Python is an interpreted high-level programming language for general-purpose programming. Created by Guido van Rossum and first released in 1991, Python has a design philosophy that emphasizes code readability, notably using significant whitespace.

Python features a dynamic type system and automatic memory management. It supports multiple programming paradigms, including object-oriented, imperative, functional and procedural, and has a large and comprehensive standard library.

- Python is Interpreted – Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.
- Python is Interactive – you can actually sit at a Python prompt and interact with the interpreter directly to write your programs.

Python also acknowledges that speed of development is important. Readable and terse code is part of this, and so is access to powerful constructs that avoid tedious repetition of code. Maintainability also ties into this may be an all but useless metric, but it does say something about how much code you have to scan, read and/or understand to troubleshoot problems or tweak behaviours. This speed of development, the ease with which a programmer of other languages can pick up basic Python skills and the huge standard library is key to another area where Python excels. All its tools have been quick to implement, saved a lot of time, and several of them have later been patched and updated by people with no Python background - without breaking.

### **Modules Used in Project: -**

#### **Numpy**

Num.py is a general-purpose array-processing package. It provides a high-performance multidimensional array object, and tools for working with these arrays.

It is the fundamental package for scientific computing with Python. It contains various features including these important ones:

- ✦ A powerful N-dimensional array object
- ✦ Sophisticated (broadcasting) functions
- ✦ Tools for integrating C/C++ and Fortran code
- ✦ Useful linear algebra, Fourier transform, and random number capabilities Besides its obvious scientific uses, Num.py can also be used as an efficient multi- dimensional container of generic data. Arbitrary data-types can be defined using Num.py which allows Num.py to seamlessly and speedily integrate with a wide variety of databases.

#### **Pandas**

Pandas is an open-source Python Library providing high-performance data manipulation and analysis tool using its powerful data structures. Python was majorly used for data munging and preparation. It had very little contribution towards data analysis. Pandas solved this problem. Using Pandas, we can accomplish five typical steps in the processing and analysis of data, regardless of the origin of data load, prepare, manipulate, model, and analyse. Python with Pandas is used in a wide range of fields including academic and commercial domains including finance, economics, Statistics, analytics, etc.

## Matplotlib

Matplotlib is a Python 2D plotting library which produces publication quality figures in a variety of hardcopy formats and interactive environments across platforms. Matplotlib can be used in Python scripts, the Python and [Python](#) shells, the [Jupyter](#) Notebook, web application servers, and four graphical user interface toolkits. Matplotlib tries to make easy things easy and hard things possible. You can generate plots, histograms, power spectra, bar charts, error charts, scatter plots, etc., with just a few lines of code. For examples, see the sample plots and thumbnail gallery.

For simple plotting the pyplot module provides a MATLAB-like interface, particularly when combined with Python. For the power user, you have full control of line styles, font properties, axes properties, etc, via an object-oriented interface or via a set of functions familiar to MATLAB users.

## Scikit – learn

Scikit-learn provides a range of supervised and unsupervised learning algorithms via a consistent interface in Python. It is licensed under a permissive simplified BSD license and is distributed under many Linux distributions, encouraging academic and commercial use.

## Python

Python is an interpreted high-level programming language for general-purpose programming. Created by Guido van Rossum and first released in 1991, Python has a design philosophy that emphasizes code readability, notably using significant whitespace.

Python features a dynamic type system and automatic memory management. It supports multiple programming paradigms, including object-oriented, imperative, functional and procedural, and has a large and comprehensive standard library.

- Python is Interpreted – Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.
- Python is Interactive – you can actually sit at a Python prompt and interact with the interpreter directly to write your programs.

Python also acknowledges that speed of development is important. Readable and terse code is part of this, and so is access to powerful constructs that avoid tedious repetition of code. Maintainability also ties into this may be an all but useless metric, but it does say something about how much code you have to scan, read and/or understand to troubleshoot problems or tweak behaviours. This speed of development, the ease with which a programmer of other languages can pick up basic Python skills and the huge standard library is key to another area where Python excels. All its tools have been quick to implement, saved a lot of time, and several of them have later been patched and updated by people with no Python background - without breaking.

## Install Python Step-by-Step in Windows and Mac:

Python a versatile programming language doesn't come pre-installed on your computer devices. Python was first released in the year 1991 and until today it is a very popular high- level programming language. Its style philosophy emphasizes code readability with its notable use of great whitespace.

The object-oriented approach and language construct provided by Python enables programmers to write both clear and logical code for projects. This software does not come pre-packaged with Windows.

### 5.2.2 How to Install Python on Windows and Mac:

There have been several updates in the Python version over the years. The question is how to install Python? It might be confusing for the beginner who is willing to start learning Python but this tutorial will solve your query. The latest or the newest version of Python is version 3.7.4 or in other words, it is Python 3.

**Note:** The python version 3.7.4 cannot be used on Windows XP or earlier devices.

Before you start with the installation process of Python. First, you need to know about your **System Requirements**. Based on your system type i.e. operating system and based processor, you must download the python version. My system type is a **Windows 64-bit operating system**. So the steps below are to install python version 3.7.4 on Windows 7 device or to install Python 3. [Download the Python Cheat sheet here](#). The steps on how to install Python on Windows 10, 8 and 7 are **divided into 4 parts** to help understand better.

#### 5.2.2.1 Download the Correct version into the system

**Step 1:** Go to the official site to download and install python using Google Chrome or any other web browser. OR Click on the following link: <https://www.python.org>





Now, check for the latest and the correct version for your operating system.

**Step 2:** Click on the Download Tab.










**Step 3:** You can either select the Download Python for windows 3.7.4 button in Yellow Colour or you can scroll further down and click on download with respective to their version.

Here, we are downloading the most recent python version for windows 3.7.4

Looking for a specific release?

Python releases by version number:

Release version	Release date	Click for more	
Python 3.7.4	July 8, 2019	 Download	<a href="#">Release Notes</a>
Python 3.6.9	July 2, 2019	 Download	<a href="#">Release Notes</a>
Python 3.7.3	March 25, 2019	 Download	<a href="#">Release Notes</a>
Python 3.4.10	March 18, 2019	 Download	<a href="#">Release Notes</a>
Python 3.5.7	March 18, 2019	 Download	<a href="#">Release Notes</a>
Python 3.7.16	March 4, 2019	 Download	<a href="#">Release Notes</a>
Python 3.7.2	Dec. 24, 2018	 Download	<a href="#">Release Notes</a>

**Step 4:** Scroll down the page until you find the Files option.

**Step 5:** Here you see a different version of python along with the operating system.

Files					
Version	Operating System	Description	MD5 Sum	File Size	GPG
<a href="#">Gzipped source tarball</a>	Source release		68111671a5b3db4ae77b9ab0337079be	23017663	<a href="#">GPG</a>
<a href="#">XZ compressed source tarball</a>	Source release		d733e4aa6d097051c3eca45ee3604803	17131432	<a href="#">GPG</a>
<a href="#">macOS 64-bit/32-bit installer</a>	Mac OS X	for Mac OS X 10.6 and later	6428b4fa7553da71a4c2c8abcce08e6	34898436	<a href="#">GPG</a>
<a href="#">macOS 64-bit installer</a>	Mac OS X	for OS X 10.9 and later	5dd807c38217a45773b5eaa936b2a3f	28882846	<a href="#">GPG</a>
<a href="#">Windows help file</a>	Windows		063999573a2c9682ac50cadc6b477cd2	8131761	<a href="#">GPG</a>
<a href="#">Windows x86-64 embeddable zip file</a>	Windows	for AMD64/EM64T/x64	9800c3bf6d9ee0b0a8e82184a9e0728a2	7504391	<a href="#">GPG</a>
<a href="#">Windows x86-64 executable installer</a>	Windows	for AMD64/EM64T/x64	a7023e4b0aef76d95db35c3a583e5c3400	26882368	<a href="#">GPG</a>
<a href="#">Windows x86-64 web-based installer</a>	Windows	for AMD64/EM64T/x64	28c31c5080be073ae8e51a3b6351b4bd2	1362904	<a href="#">GPG</a>
<a href="#">Windows x86 embeddable zip file</a>	Windows		9fab38d1584c3379fda94133574139d8	6741628	<a href="#">GPG</a>
<a href="#">Windows x86 executable installer</a>	Windows		33c3022942a54446a3d845147e394788	25665848	<a href="#">GPG</a>
<a href="#">Windows x86 web-based installer</a>	Windows		1b670cfaf5d117d03c30983ea371687c	1324608	<a href="#">GPG</a>

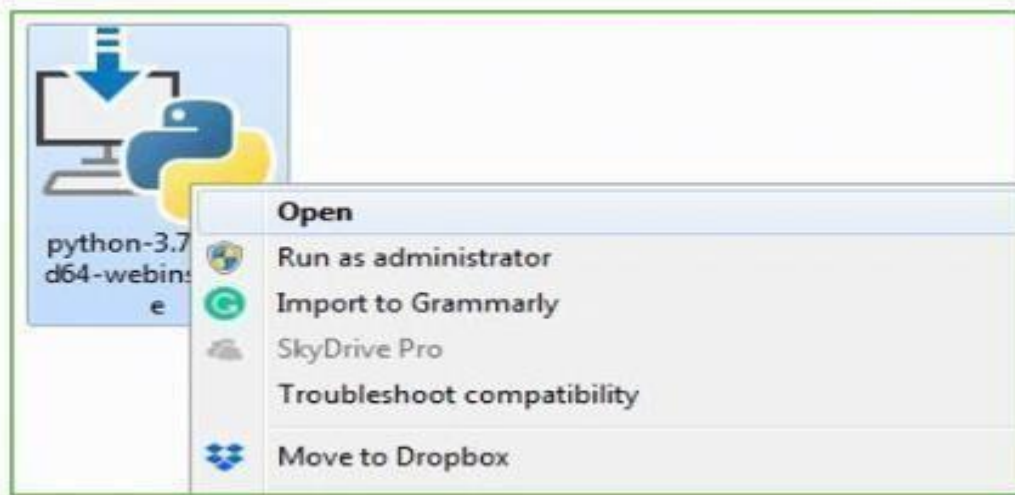
- To download **Windows 32-bit python**, you can select any one from the three options: Windows x86 embeddable zip file, Windows x86 executable installer or Windows x86 web-based installer.
- To download **Windows 64-bit python**, you can select any one from the three options: Windows x86-64 embeddable zip file, Windows x86-64 executable installer or Windows x86-64 web-based installer.

Here we will install Windows x86-64 web-based installer. Here your first part regarding which version of python is to be downloaded is completed. Now we move ahead with the second part in installing python i.e. Installation

**Note:** To know the changes or updates that are made in the version you can click on the Release Note Option.

### 5.2.2.2 Installation of Python

**Step 1:** Go to Download and Open the downloaded python version to carry out the installation process.



**Step 2:** Before you click on Install Now, make sure to put a tick on Add Python 3.7 to PATH.



**Step 3:** Click on Install NOW After the installation is successful. Click on Close.

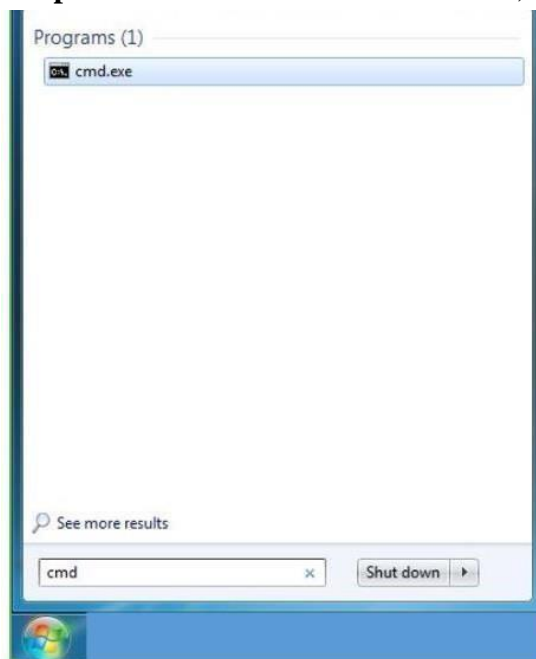


With these above three steps on python installation, you have successfully and correctly installed Python. Now is the time to verify the installation. **Note:** The installation process might take a couple of minutes.

### 5.2.2.3 Verify the Python Installation

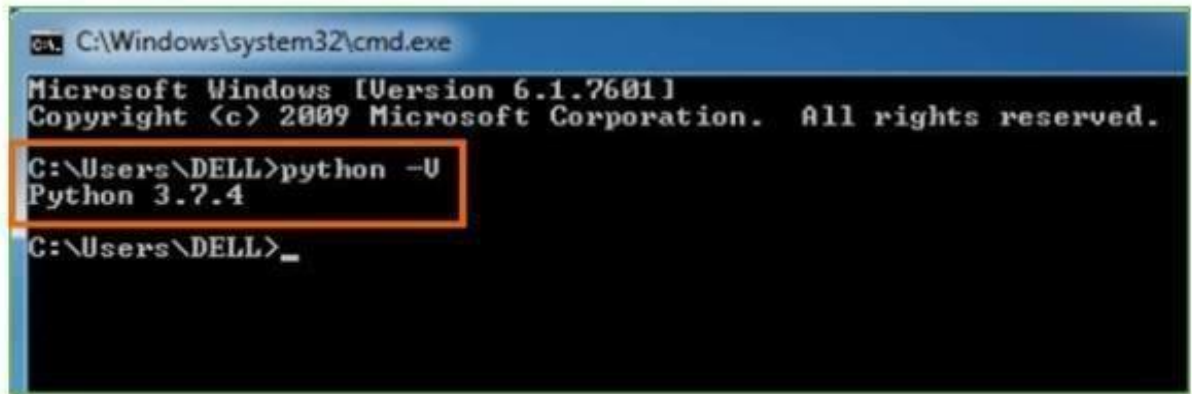
**Step 1:** Click on Start

**Step 2:** In the Windows Run Command, type “cmd”



**Step 3:** Open the Command prompt option.

**Step 4:** Let us test whether the python is correctly installed. Type **python -V** and press Enter.



```
C:\Windows\system32\cmd.exe
Microsoft Windows [Version 6.1.7601]
Copyright (c) 2009 Microsoft Corporation. All rights reserved.

C:\Users\DELL>python -V
Python 3.7.4

C:\Users\DELL>_
```

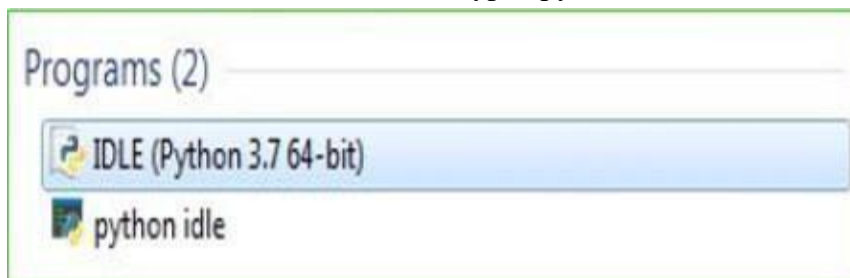
**Step 5:** You will get the answer as 3.7.4

**Note:** If you have any of the earlier versions of Python already installed. You must first uninstall the earlier version and then install the new one.

#### 5.2.2.4 Check how the Python IDLE works

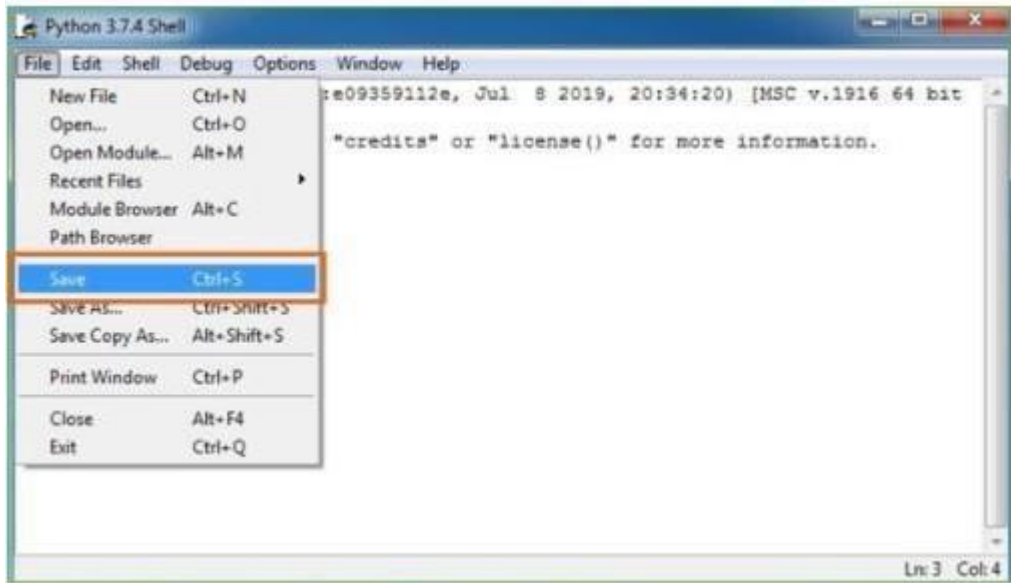
**Step 1:** Click on Start

**Step 2:** In the Windows Run command, type “python idle”



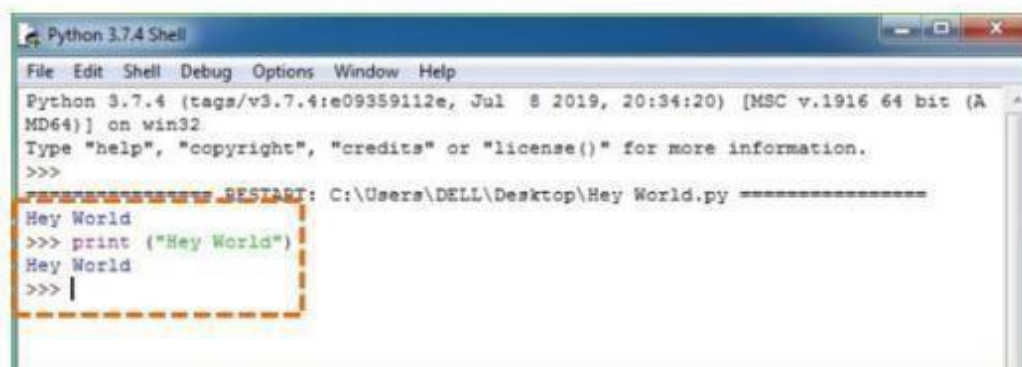
**Step 3:** Click on IDLE (Python 3.7 64-bit) and launch the program

**Step 4:** To go ahead with working in IDLE you must first save the file. **Click on File > Click on Save**



**Step 5:** Name the file and save as type should be Python files. Click on SAVE. Here I have named the files as Hey World.

**Step 6:** Now for e.g. enter `print ("Hey World")` and Press Enter.



You will see that the command given is launched. With this, we end our tutorial on how to install Python. You have learned how to download python for windows into your respective operating system.

### 5.2.3 Jupiter Notebook

The Jupiter Notebook is an open-source web application that you can use to create and share documents that contain live code, equations, visualizations, and text. Jupiter Notebook is maintained by the people at [Project Jupiter](#).


Jupiter Notebooks are a spin-off project from the Python project, which used to have an Python Notebook project itself. The name, Jupiter, comes from the core supported programming languages that it supports: Julia, Python, and R. Jupiter ships with the Python kernel, which allows you to write your programs in Python, but there are currently over 100 other kernels that you can also use.

## 5.4 Google colab

1. **Access Google Colab:** Go to the Google Colab website and sign in with your Google account.
2. **Create a New Notebook:** Click on the "New Notebook" button to create a new Colab notebook. This notebook will serve as your project workspace.
3. **Set Runtime Type:** Choose the desired runtime type (CPU, GPU, or TPU) by navigating to "Runtime" -> "Change runtime type". GPUs and TPUs can accelerate computation for machine learning tasks.
4. **Import Libraries:** Import any necessary libraries for your project by adding code cells and using commands like `! pip install` to install libraries not already available in the Colab environment.
5. **Write Code:** Write your project code in individual code cells. You can add text descriptions or explanations using Markdown cells.
6. **Execute Code:** Execute each code cell by clicking the play button or pressing Shift+Enter. Colab will execute the code and display the output.
7. **Access Data:** If your project requires data, you can upload it directly to Colab from your local machine using the file upload feature or access data from Google Drive.
8. **Save and Share:** Save your progress by clicking on "File" -> "Save" or using Ctrl+S. You can share your project with collaborators by providing them with the notebook's shareable link.
9. **Export Results:** Once your project is complete, you can export the results by downloading the notebook as an IPython (. ipynb) file or saving visualizations and outputs to your Google Drive or local machine.
10. **Shutdown Notebook:** Remember to shut down your notebook when you're finished to release resources. You can do this by selecting "Runtime" -> "Shutdown all runtimes".

Following these steps will enable you to efficiently conduct your project in Google Colab, leveraging its cloud-based infrastructure for computational tasks.



Welcome to Colaboratory

FileEditViewInsertRuntimeToolsHelp

Share

Settings

Table of contents

Getting started

Data science

Machine learning

More resources

Featured examples

+ Section

+ Code+ TextCopy to Drive


ConnectColab AI

Welcome to Colab!

(New) Try the Gemini API

- [Generate a Gemini API key](#)
- [Talk to Gemini with the Speech-to-Text API](#)
- [Compare Gemini with ChatGPT](#)
- [More notebooks](#)

If you're already familiar with Colab, check out this video to learn about interactive tables, the executed code history view and the command palette.



[ ] Start coding or generate with AI.

## 5.5 RESULTS:

### 5.5.1 SOURCE CODE

#### 5.5.1.1 : Importing Datasets:

Importing the dataset using the below code:

```
import pandas as pd
import numpy as np
df = pd.read_csv("Products excel dataset.csv")
df
```

	product_name	product_price	Rate	Review	Summary	Sentiment
0	Candes 12 L Room/Personal Air Cooler?????(Whi...	3999	5	superl	great cooler excellent air flow and for this p...	positive
1	Candes 12 L Room/Personal Air Cooler?????(Whi...	3999	5	awesome	best budget 2 fit cooler nice cooling	positive
2	Candes 12 L Room/Personal Air Cooler?????(Whi...	3999	3	fair	the quality is good but the power of air is de...	positive
3	Candes 12 L Room/Personal Air Cooler?????(Whi...	3999	1	useless product	very bad product its a only a fan	negative
4	Candes 12 L Room/Personal Air Cooler?????(Whi...	3999	3	fair	ok ok product	neutral
...	...	...	...	...	...	...
2605	cello Pack of 18 Opalware Cello Dazzle Lush Fi...	1299	4	pretty good	nice look and product is good but size of plat...	negative
2606	cello Pack of 18 Opalware Cello Dazzle Lush Fi...	1299	4	nice product	good	positive
2607	cello Pack of 18 Opalware Cello Dazzle Lush Fi...	1299	5	must buy!	awesome productand delivery guy is very good p...	positive
2608	cello Pack of 18 Opalware Cello Dazzle Lush Fi...	1299	5	superl	very nice quality is so good	positive
2609	cello Pack of 18 Opalware Cello Dazzle Lush Fi...	1299	4	value-for-money	very nice product and very good packaging	positive

2610 rows x 6 columns

#### 5.5.1.2 Data Preprocessing

#### Installing Required Packages:

STEP-2 DATA PRE PROCESSING

```
[4] pip install pandas nltk
```

Requirement already satisfied: pandas in /usr/local/lib/python3.10/dist-packages (2.0.3)  
Requirement already satisfied: nltk in /usr/local/lib/python3.10/dist-packages (3.8.1)  
Requirement already satisfied: python-dateutil>=2.8.2 in /usr/local/lib/python3.10/dist-packages (from pandas) (2.8.2)  
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.10/dist-packages (from pandas) (2023.4)  
Requirement already satisfied: tzdata>=2022.1 in /usr/local/lib/python3.10/dist-packages (from pandas) (2024.1)  
Requirement already satisfied: numpy>=1.21.0 in /usr/local/lib/python3.10/dist-packages (from pandas) (1.25.2)  
Requirement already satisfied: click in /usr/local/lib/python3.10/dist-packages (from nltk) (8.1.7)  
Requirement already satisfied: joblib in /usr/local/lib/python3.10/dist-packages (from nltk) (1.3.2)  
Requirement already satisfied: regex>=2021.8.3 in /usr/local/lib/python3.10/dist-packages (from nltk) (2023.12.25)  
Requirement already satisfied: tqdm in /usr/local/lib/python3.10/dist-packages (from nltk) (4.66.2)  
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.10/dist-packages (from python-dateutil>=2.8.2->pandas) (1.16.0)

```
[5] import nltk
nltk.download('stopwords')
```

[nltk\_data] Downloading package stopwords to /root/nltk\_data...  
[nltk\_data] Unzipping corpora/stopwords.zip.  
True

```
[6] import nltk
nltk.download('stopwords')
```

[nltk\_data] Downloading package stopwords to /root/nltk\_data...  
[nltk\_data] Package stopwords is already up-to-date!  
True

## i) Word Tokenization & To Remove Stop Words:

### PRE PROCESSING - TOKENIZATION & TO REMOVE STOP WORDS

```
import pandas as pd
from nltk.tokenize import word_tokenize
from nltk.corpus import stopwords
import string
import nltk # Import nltk library

# Download the 'punkt' resource
nltk.download('punkt')

# Load the CSV file into a Pandas DataFrame
df = pd.read_csv('Products excel dataset.csv')

# Define a function to tokenize and remove stopwords
def tokenize_and_remove_stopwords(text):
    if pd.isnull(text): # Check for NaN values
        return []

    # Tokenize the text, remove punctuation, and lowercase the tokens
    tokens = [word.lower() for word in word_tokenize(text) if word.isalpha() and word not in string.punctuation]

    stop_words = set(stopwords.words('english'))
    tokens = [word for word in tokens if word not in stop_words]

    return tokens

# Specify the columns to process
text_columns = ['product_name', 'Review', 'Summary']

# Apply the function to the specified columns using apply and lambda
for column in text_columns:
    df[column] = df[column].apply(lambda x: tokenize_and_remove_stopwords(x))

# Save the new DataFrame back to a CSV file
df.to_csv('output_file.csv', index=False)
df
```

[nltk\_data] Downloading package punkt to /root/nltk\_data...

[nltk\_data] Unzipping tokenizers/punkt.zip.

	product_name	product_price	Rate	Review	Summary	Sentiment
0	[candes, l, air, cooler, white, black, elegant...	3999	5	[super]	[great, cooler, excellent, air, flow, price, a...	positive
1	[candes, l, air, cooler, white, black, elegant...	3999	5	[awesome]	[best, budget, fit, cooler, nice, cooling]	positive
2	[candes, l, air, cooler, white, black, elegant...	3999	3	[fair]	[quality, good, power, air, decent]	positive
3	[candes, l, air, cooler, white, black, elegant...	3999	1	[useless, product]	[bad, product, fan]	negative
4	[candes, l, air, cooler, white, black, elegant...	3999	3	[fair]	[ok, ok, product]	neutral
...	...	...	...	...	...	...
2605	[cello, pack, opalware, cello, dazzle, lush, f...	1299	4	[pretty, good]	[nice, look, product, good, size, plates, comf...	negative
2606	[cello, pack, opalware, cello, dazzle, lush, f...	1299	4	[nice, product]	[good]	positive
2607	[cello, pack, opalware, cello, dazzle, lush, f...	1299	5	[must, buy]	[awesome, productand, delivery, guy, good, pro...	positive
2608	[cello, pack, opalware, cello, dazzle, lush, f...	1299	5	[super]	[nice, quality, good]	positive
2609	[cello, pack, opalware, cello, dazzle, lush, f...	1299	4	[]	[nice, product, good, packaging]	positive

2610 rows x 6 columns

## ii) Word Lemmatization:

**Lemmatization:** To get a square root for words

### LEMMATIZATION

```
import pandas as pd
from nltk.tokenize import word_tokenize
from nltk.corpus import stopwords
from nltk.stem import WordNetLemmatizer
import string
import nltk

# Download the necessary resources
nltk.download('punkt')
nltk.download('stopwords')
nltk.download('wordnet')

# Load the CSV file into a Pandas DataFrame
df = pd.read_csv('Products excel dataset.csv')

# Initialize the lemmatizer and stopwords
lemmatizer = WordNetLemmatizer()
stop_words = set(stopwords.words('english'))

# Define a function to tokenize, remove stopwords, and lemmatize
def process_text(text):
    if pd.isnull(text):
        return ''

    # Tokenize the text, remove punctuation, and lowercase the tokens
    tokens = [word.lower() for word in word_tokenize(text) if word.isalpha() and word not in string.punctuation]

    # Remove stopwords
    tokens = [word for word in tokens if word not in stop_words]

    # Lemmatize the tokens
    lemmatized_tokens = [lemmatizer.lemmatize(token) for token in tokens]

    return ' '.join(lemmatized_tokens)

# Specify the columns to process
text_columns = ['product_name', 'Review', 'Summary']

# Apply the function to the specified columns using apply and lambda
for column in text_columns:
    df[column] = df[column].apply(lambda x: process_text(x))

# Save the new DataFrame back to a CSV file
df.to_csv('output_file.csv', index=False)
df
```

[nltk\_data] Downloading package punkt to /root/nltk\_data...  
[nltk\_data] Package punkt is already up-to-date!  
[nltk\_data] Downloading package stopwords to /root/nltk\_data...  
[nltk\_data] Package stopwords is already up-to-date!  
[nltk\_data] Downloading package wordnet to /root/nltk\_data...

	product_name	product_price	Rate	Review	Summary	Sentiment
0	candes l air cooler white black elegant high c...	3999	5	super	great cooler excellent air flow price amazing ...	positive
1	candes l air cooler white black elegant high c...	3999	5	awesome	best budget fit cooler nice cooling	positive
2	candes l air cooler white black elegant high c...	3999	3	fair	quality good power air decent	positive
3	candes l air cooler white black elegant high c...	3999	1	useless product	bad product fan	negative
4	candes l air cooler white black elegant high c...	3999	3	fair	ok ok product	neutral
...	...	...	...	...	...	...
2605	cello pack opalware cello dazzle lush fiesta o...	1299	4	pretty good	nice look product good size plate comfortable ...	negative
2606	cello pack opalware cello dazzle lush fiesta o...	1299	4	nice product	good	positive
2607	cello pack opalware cello dazzle lush fiesta o...	1299	5	must buy	awesome productand delivery guy good product a...	positive
2608	cello pack opalware cello dazzle lush fiesta o...	1299	5	super	nice quality good	positive
2609	cello pack opalware cello dazzle lush fiesta o...	1299	4		nice product good packaging	positive

2610 rows x 6 columns

## Checking for null values:

- **isnull().sum()** : is a common operation used in Pandas to count the number of missing (NaN) values in each column of a Dataframe.

```
MISSING DATA
df.isnull()
df.isnull().sum()

product_name    0
product_price    0
Rate            0
Review          0
Summary         0
Sentiment       0
dtype: int64
```

## To identify unique values:

```
[42] df.nunique()

product_price    154
Rate             5
Sentiment        3
dtype: int64
```

## STEP – 3 TF-IDF VALUES:

### STEP-3: ALGORITHM- TERM FREQUENCY AND INVERSE DOCUMENT FREQUENCY

1. IDF: Inverse Document Frequency measures how unique or rare a term is across all documents.
2. TF: Term Frequency measures how often a term appears in a document.

```
STEP 3 - TF-IDF VALUES

[13] #!pip install sklearn
from sklearn.feature_extraction.text import TfidfVectorizer
tfidf=TfidfVectorizer()
tfidf_matrix=tfidf.fit_transform(df['Summary'])
tfidf_matrix

<2610x3429 sparse matrix of type '<class 'numpy.float64''>'
with 16123 stored elements in Compressed Sparse Row format>

print(tfidf_matrix)

(0, 1732) 0.38638498519842865
(0, 3172) 0.4788798366423978
(0, 72) 0.2964798578156989
(0, 2277) 0.2299889958584324
(0, 1129) 0.48548865565957
(0, 47) 0.3286166568981809
(0, 969) 0.2627731307178077
(0, 647) 0.34699174957899007
(0, 1301) 0.2722946247969733
(1, 649) 0.49687337435190926
(1, 1968) 0.21586585812584558
(1, 1181) 0.459586971393424
(1, 358) 0.4548980804853946
(1, 263) 0.29881514816919286
(1, 647) 0.4548980804853946
(2, 733) 0.5986318627100169
(2, 2345) 0.54335460951773696
(2, 1259) 0.17688748388731826
(2, 2389) 0.2656628042987284
(2, 47) 0.494086660112338
(3, 1835) 0.7993764576813252
(3, 2317) 0.287781325863556
(3, 282) 0.5277299493213197
(4, 2820) 0.9723492767508191
(4, 2317) 0.2335133348939815
:
(2685, 2285) 0.486478398279862
(2685, 2776) 0.27325255971400897
(2685, 553) 0.3222419359221216
(2685, 1722) 0.28925474058728784
(2685, 2754) 0.5814576934319838
(2685, 2317) 0.1248277962888188
(2685, 1259) 0.1152259548686225
(2685, 1968) 0.16146613978044174
(2686, 1259) 1.0
(2687, 184) 0.5832577077648358
(2687, 2318) 0.4695837381714372
(2687, 1832) 0.3679843992822686
(2687, 1347) 0.28812321613085414
(2687, 1322) 0.3834567867256784
(2687, 768) 0.275385683931001
(2687, 176) 0.25434568999947786
(2687, 2317) 0.1337738495718637
(2687, 1259) 0.1234867498466783
(2688, 1259) 0.43771881551854347
(2688, 2389) 0.6573967576278163
(2688, 1968) 0.6133773248841884
(2689, 2099) 0.836515972735434
(2689, 2317) 0.2918371694944695
(2689, 1259) 0.26938881021474824
(2689, 1968) 0.3774946174716771
```

```
[56] df.dtypes
```

product_name	object	
product_price	int64	
Rate	int64	
Review	object	
Summary	object	
Sentiment	object	
dtype:	object	

```
df['Sentiment']=df['Sentiment'].astype('category')
df.dtypes
```

product_name	object	
product_price	int64	
Rate	int64	
Review	object	
Summary	object	
Sentiment	category	
dtype:	object	

```
[58] df['Sentiment']=df['Sentiment'].cat.codes
```

```
[59] df.head()
```

	product_name	product_price	Rate	Review	Summary	Sentiment
0	candes l air cooler white black elegant high c...	3999	5	super	great cooler excellent air flow price amazing ...	2
1	candes l air cooler white black elegant high c...	3999	5	awesome	best budget fit cooler nice cooling	2
2	candes l air cooler white black elegant high c...	3999	3	fair	quality good power air decent	2
3	candes l air cooler white black elegant high c...	3999	1	useless product	bad product fan	0
4	candes l air cooler white black elegant high c...	3999	3	fair	ok ok product	1

```
[61] non_numeric_cols = df.select_dtypes(include='object').columns
```

```
[62] df[non_numeric_cols].head()
```

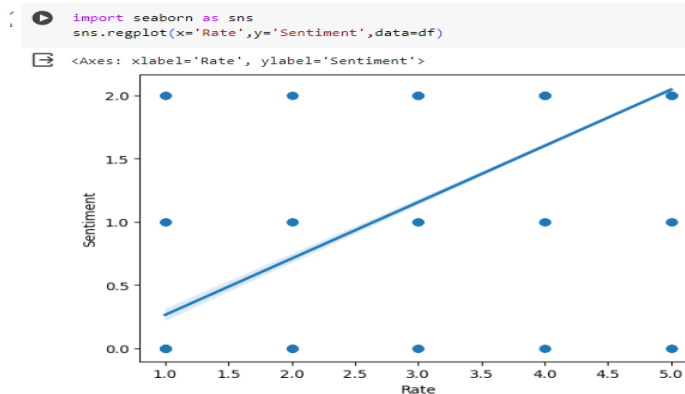
	product_name	Review	Summary
0	candes l air cooler white black elegant high c...	super	great cooler excellent air flow price amazing ...
1	candes l air cooler white black elegant high c...	awesome	best budget fit cooler nice cooling
2	candes l air cooler white black elegant high c...	fair	quality good power air decent
3	candes l air cooler white black elegant high c...	useless product	bad product fan
4	candes l air cooler white black elegant high c...	fair	ok ok product

```
[63] df.drop(non_numeric_cols, axis=1, inplace=True)
```

```
[64] cor1=df.corr()
cor1
```

	product_price	Rate	Sentiment
product_price	1.000000	0.108948	0.075828
Rate	0.108948	1.000000	0.828348
Sentiment	0.075828	0.828348	1.000000

## Visualization:



### 5.5.1.3 Building Model:

```
[66] from sklearn.tree import DecisionTreeClassifier
```

```
import sklearn
from sklearn.model_selection import train_test_split
```

```
[68] '''
from sklearn.naive_bayes import MultinomialNB
classifier = DecisionTreeClassifier
y=df['Sentiment']
x=tfidf_matrix
classifier1 = MultinomialNB()
x_train,y_train,x_test,y_test = train_test_split(x,y,test_size=0.2,random_state=42)

print(x_test.shape)
print(x_train.shape)
print(y_test.shape)
print(y_train.shape)
x_train = x_train.toarray()
'''

'''from sklearn.naive_bayes import MultinomialNB\classifier = DecisionTreeClassifier\ny=df['Sentiment']\nx=tfidf_matrix\nclassifier1 = MultinomialNB()\nx_train,y_train,x_test,y_test = train_test_split(x,y,test_size=0.2,random_state=42)\n\nprint(x_test.shape)\nprint(x_train.shape)\nprint(y_test.shape)\nprint(y_train.shape)\nx_train = x_train.toarray()\n'''
```

```

[69] #classifier1.fit(x_train,y_train)

[70] from tensorflow.keras.models import Sequential
    from tensorflow.keras.layers import Dense
    from sklearn.naive_bayes import MultinomialNB

[71] from sklearn.model_selection import train_test_split
    from sklearn.naive_bayes import MultinomialNB

[72] x_train, x_test, y_train, y_test = train_test_split(tfidf_matrix, df['Sentiment'], test_size=0.2, random_state=42)

[73] classifier = MultinomialNB()
    classifier.fit(x_train, y_train)

    * MultinomialNB
    MultinomialNB()

[74] epochs = 150
    for epoch in range(epochs):
        classifier.partial_fit(x_train, y_train, classes=np.unique(y_train))

[75] y_pred = classifier.predict(x_test)

```

## Accuracy:

```

[76] from sklearn.metrics import accuracy_score
    accuracy = accuracy_score(y_test, y_pred)
    print("Accuracy:", accuracy)

    Accuracy: 0.8620689655172413

[77] from xgboost import XGBClassifier
    x_train, x_test, y_train, y_test = train_test_split(tfidf_matrix, df['Sentiment'], test_size=0.2, random_state=42)

[78] classifier1 = XGBClassifier()
    classifier1.fit(x_train, y_train)

    * XGBClassifier
    XGBClassifier(base_score=None, booster=None, callbacks=None,
        colsample_bylevel=None, colsample_bynode=None,
        colsample_bytree=None, device=None, early_stopping_rounds=None,
        enable_categorical=False, eval_metric=None, feature_types=None,
        gamma=None, grow_policy=None, importance_type=None,
        interaction_constraints=None, learning_rate=None, max_bin=None,
        max_cat_threshold=None, max_cat_to_onehot=None,
        max_delta_step=None, max_depth=None, max_leaves=None,
        min_child_weight=None, missing=None, monotone_constraints=None,
        multi_strategy=None, n_estimators=None, n_jobs=None,
        num_parallel_tree=None, objective='multi:softprob', ...)

[79] y_pred = classifier1.predict(x_test)
    accuracy1 = accuracy_score(y_test, y_pred)
    print("Accuracy:", accuracy1)

    Accuracy: 0.8697318007662835

```

## Results:

```

[80] sentence = "This product is amazing! I love it!"
    sentence_tokens = tokenize_and_remove_stopwords(sentence)
    sentence_tfidf = tfidf.transform([' '.join(sentence_tokens)])
    predicted_sentiment = classifier.predict(sentence_tfidf)[0]
    print("Predicted sentiment:", predicted_sentiment)

```

Predicted sentiment: 2

```

> sentence = input('enter comment:')
sentence_tfidf = tfidf.transform([sentence])
prediction = classifier1.predict(sentence_tfidf)[0]
print(prediction)
if prediction == 0:
    print('NEGATIVE')
elif (prediction == 1):
    print('NEUTRAL')
else:
    print('POSITIVE')

```

enter comment:

enter comment:EXCELLENT!!  
2  
POSITIVE

enter comment:BAD PRODUCT!!  
0  
NEGATIVE

enter comment:OK OK!!  
1  
NEUTRAL

## **CHAPTER - 6**

### **SYSTEM TESTING**

Testing is that the debugging program is one amongst the leading crucial aspects of the pc programming triggers, while not programming that works, the system would ne'er turn out relate in Nursing output of the at it had been designed. Testing is best performed once user development is asked to help in characteristic all errors and bugs. The sample knowledge is used for testing. It is not amounting however quality of the information used the matters of testing. Testing is aimed toward guaranteeing that the system was accurately relate in Nursing with efficiency before live operation commands.

**Testing objectives:** The most objective of testing is to uncover a bunch of errors, consistently and with minimum effort and time. Stating formally, we can say, testing may be a method of corporal punishment a program with intent of finding miscalculation.

5.6 A productive check is one that uncovers Associate in Nursing hitherto undiscovered error.

5.7 A decent legal action is one that has likelihood of finding miscalculation, if it exists.

5.8 The check is insufficient to find probably gift errors.

5.9 The code additional or less confirms to the standard and reliable standards.

#### **6.1. TYPES OF TESTING**

##### **6.1.1. UNIT TESTING**

Unit testing, we have a tendency to test every module separately and integrate with the general system. Unit testing focuses verification efforts on the littlest unit of code style within the module. this is often conjointly called module testing.

The module of the system is tested individually. as an example, the validation check is completed for variable the user input given by the user that validity of the information entered. it's terribly straightforward to search out error rectify the system. Every Module will be tested victimization the subsequent 2 Strategies: recording machine Testing and White Box Testing.

##### **6.1.2. BLACK BOX TESTING**

Recording machine checking may be a code testing technique during which practicality of the code below test (SUT) is tested while not staring at the interior code structure, implementation details and data of internal ways of the code. This type of testing is predicated entirely on the code needs and specifications. In recording machine Testing we

have a tendency to simply concentrate on inputs and output of the package while not bothering concerning internal data of the code program. The on top of recording machine will be any package you wish to check. For example, Associate in Nursing software like Windows, a web site like Google, a information like Oracle or maybe your own custom application. Under recording machine testing, you can check these applications by simply that specialize in the inputs and outputs while not knowing their internal code implementation.

### Types of Black Box Testing

There are many varieties of recording machine Testing however following ar the outstanding ones.

- **Functional testing:** This recording machine testing kind is said to purposeful needs of a system; it's done by code testers.
- **Non-Functional testing:** This sort of recording machine testing isn't associated with testing of a selected practicality, however non-functional needs like performance, measurability, usability.
- **Regression testing:** Regression testing is completed once code fixes, upgrades or the other system maintenance to visualize the new code has not affected the prevailing code.

### 6.1.3. WHITE BOX TESTING

White Box Testing is that the testing of a code solution's internal committal to writing and infrastructure. It focuses totally on Traffic Redundancy Elimination theming security, the flow of inputs and outputs through the applying, and rising style and value. White box testing is additionally called clear, open, structural, and glass box testing. It is one amongst 2 elements of the "box testing" approach of code testing.

#### System Testing:

Once the individual module testing is completed, modules are assembled and integrated to perform as a system. The top-down testing, that began from higher level to lower-level module, was allotted to visualize whether or not the whole system is playacting satisfactorily. There are 3 main types of System testing: Alpha Testing, Beta Testing, Acceptance Testing.

**Alpha Testing:** This refers to the system checking that's allotted by the test team with the Organization.

**Beta Testing:** This refers to the system testing that's performed by a particular cluster of friendly customers.

**Acceptance Testing:** This refers to the system testing that's performed by the client to see whether or not or to not settle for the delivery of the system.



## 6.2. TEST STRATEGY AND APPROACH

Field testing will be performed manually and functional tests will be written in detail.

### Test objectives

- All field entries must work properly.
- Pages must be activated from the identified link.
- The entry screen, messages and responses must not be delayed.

### Features to be tested

- Verify that the entries are of the correct format
- No duplicate entries should be allowed • All links should take the user to the correct page.

### Test Scenarios

#### General Scenarios

- All mandatory fields should be validated and indicated by asterisk (\*) symbol  
Validation error messages should be displayed properly at correct position
  - All error messages should be displayed in same CSS style (e.g. using red colour)
- General confirmation messages should be displayed using CSS style other than error messages style (e.g. using green colour)
- Dropdown fields should have first entry as blank or text like \_Select ‘
- Delete functionality for any record on page should ask for confirmation

### GUI and Usability Test Scenarios

- All fields on page (e.g. text box, radio options, dropdown lists) should be aligned properly.
- Scroll bar should be enabled only when necessary
- Description text box should be multi-line
- User should be able to submit the form again by correcting the errors
- Default radio options should be pre-selected on page load
- Check all pages for broken images

### Test Scenarios for a Window

- Check if default window size is correct
- Check if child window size is correct
- Check if child windows are getting closed on closing parent/opener window
- Check window minimize, maximize and close functionality
- Check if window is re-sizable

**Database Testing Test Scenarios**

- Check if correct data is getting saved in database upon successful page submit
- Check values for columns which are not accepting null values
- Check for data integrity. Data should be stored in single or multiple tables based on design
- For every database add/update operation log should be added
- Required table indexes should be created

**Security Testing Test Scenarios**

- Secure pages should use HTTPS protocol
- Check application logout functionality
- Check for Brute Force Attacks
- Password should not be stored in cookies
- Test for Denial-of-Service attacks

## **CHAPTER - 7**

### **CONCLUSION**

In conclusion, the sentiment analysis project on Flipkart reviews using deep learning techniques has provided valuable insights into customer opinions and preferences. By leveraging advanced natural language processing models, we were able to classify reviews accurately and understand the overall sentiment towards products on the platform. This analysis not only aids in gauging customer satisfaction but also offers actionable insights for businesses to improve their products and services based on user feedback. Overall, employing deep learning for sentiment analysis enhances decision-making processes and contributes to creating a more customer-centric shopping experience on Flipkart.

### **FUTURE SCOPE**

Looking ahead, the future scope for Flipkart reviews sentiment analysis using deep learning is promising and expansive. One potential avenue is to enhance the model's accuracy and efficiency by incorporating advanced deep learning architectures such as transformers or BERT models. This could lead to more nuanced sentiment analysis, capturing subtler aspects of customer feedback.

Additionally, integrating multi-modal data sources like images or videos along with text reviews can provide a comprehensive understanding of customer sentiments. Furthermore, exploring real-time sentiment analysis capabilities could enable businesses to respond promptly to customer feedback, improving overall customer satisfaction and loyalty. Overall, there are numerous exciting possibilities to further develop and refine sentiment analysis techniques using deep learning for Flipkart reviews, enhancing the platform's capabilities and user experience.

## REFERENCES

- [1]. “Sentimental Visualization: Semantic Analysis of Online Product Reviews Using Python and Tableau” Hanan Alasmari, IEEE ON BIG DATA VO., XX, NO., X, DECEMBER 2020.
- [2]. “Comparative Analysis of Sentiment Orientation Using SVM and Naïve Bayes Techniques” Shweta Rana and Archana Singh, 2016 2nd International Conference on Next Generation Computing Technologies (NGCT2016)
- [3]. Zied Kechaou, Mohamed Ben Ammar and Adel.M Alimi " Improving e-learning with sentiment analysis of users' opinions "2011 IEEE Global Engineering Education Conference (EDUCON)
- [4]. “Sentiment Analysis and Opinion Mining: A Survey” G.Vinodhini, RM.Chandrasekaran Volume 2, Issue 6, June 2012(ijarcse )
- [5]. “Thumbs up? Sentiment Classification using Machine Learning Techniques” Bo Pang and Lillian Lee
- [6]. Flipkart Product Review (Naushad Shukoor)
- [7]. Google Search (Definition)
- [8]. Chafale, Dhanashri, and Amit Pimpalkar. "Review on Developing Corpora for Sentiment Analysis Using Plutchik's Wheel of Emotions with Fuzzy Logic. “International Journal of Computer Sciences and Engineering (IJCSE) 2 (2014): 14-18
- [9]. J.Ortigosa-Hernández,J.D. Rodríguez, L. Alzate, M. Lucania, I. Inza and J.A. Lozano, Approaching sentiment analysis by using semi-supervised learning of multi-dimensional classifiers, Neurocomputing 92 (2012)
- [10]. B. Keith, E. Fuentes and C. Meneses, A hybrid approach for sentiment analysis applied to paper reviews, 2017.
- [11]. E. Boiy and M.-F. Moens, A machine learning approach to sentiment analysis in multilingual web texts, Inf. Retr. 12(5) (Oct. 2009), 526–558.
- [12]. Tsur, D. Davidov, and A. Rappoport , “ A Great Catchy Name: Semi-Supervised Recognition of Sarcastic Sentences in Online Product Reviews”. In Proceeding of ICWSM. of Context Dependent Opinions,2010.