CUSTOMER REVIEW SENTIMENT ANALYSIS THROUGH DEEPLEARNING

A PROJECT REPORT

Submitted by

CHARUMATHI B	**420420104007**
MADHUMITHA G	**420420104030**
SAKTHIUMA S	**420420104040**
SANMATHI S	**420420104042**

in partial fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING

ADHIPARASAKTHI ENGINEERING COLLEGE

MELMARUVATHUR

ANNA UNIVERSITY :: CHENNAI 600 025

MAY 2024

BONAFIDE CERTIFICATE

Certified that this project report titled "**CUSTOMER REVIEW SENTIMENT ANALYSIS THROUGH DEEP LEARNING"** is the bonafide work of **CHARUMATHI B (420420104009), MADHUMITHA G (420420104030), SAKTHIUMA S (42042104040), SANMATHI S(420420104042),** who carried out the work under my vision.

SIGNATURE	**SIGNATURE**
Dr . C . DHAYA, Ph.D.,	**Ms . A . JAYANTHI, M.E.,**
HEAD OF THE DEPARTMENT	**SUPERVISOR**
Department of CSE	Department of CSE
Adhiparasakthi Engineering College	Adhiparasakthi Engineering College
Melmaruvathur - 603 319.	Melmaruvathur - 603 319.

Submitted for the project work (CS8811) and Viva-voce examination held on ....at Adhiparasakthi Engineering College, Melmaruvathur.

INTERNAL EXAMINER **EXTERNAL EXAMINER**

ACKNOWLEDGEMENT

With the divine blessings of **Goddess Adhiparasakthi**, I express my deep gratitude for the spiritual blessings of His holiness **PADMA SHRI ARULTHIRU AMMA** and the divine guidance of **THIRUMATHI AMMA** that has undoubtedly taken us to path of victory in completing this project.

The infrastructural support with all kinds of lab facilities have been a motivating factor in this completion of project work, all because of our **Correspondent Sakthi Thiru Dr. G. B. SENTHILKUMAR B.E.** with great pleasure we take this opportunity to thank him.

For the academic side the constant support from our honorable **Principal Dr . J. RAJA, Ph.D.,** as encourages us to work hard and attain this goal of completing the project.

Our sincere thanks to our **Head of the Department Dr . C . DHAYA, Ph.D.,** who has given us both moral and technical support adding experience to the job we have undertaken.

We thank our **Project Coordinator Mr. K. CHAIRMADURAI, M.E.,** who has led from the front questioning as at the right time and encouraging us most of the time here by extracting quality work from us.

We sincerely thank our **Project Coordinator** and **Supervisor, Ms. A . JAYANTHI, M.E.,** who helped us in crossing in the path to our glory. We also thank other staff members, librarian, and non-teaching staff member of support and motivation in all ourendeavor.

ABSTRACT

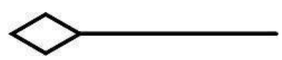
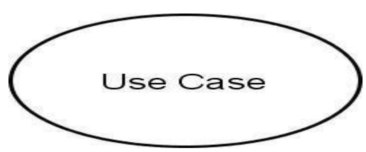
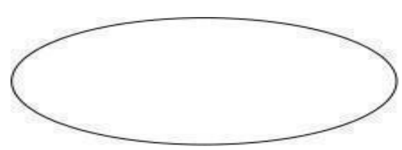
Sentiment analysis, positioned at the forefront of customer feedback interpretation, represents a sophisticated methodology integrating natural language processing and machine learning algorithms. This approach enables businesses to delve deep into the sentiments expressed in customer reviews, distinguishing between positive, negative, and neutral sentiments with an impressive degree of accuracy. By dissecting these sentiments, companies gain nuanced insights into various aspects of their products, services, and customer interactions.

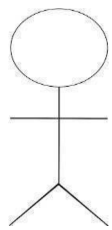
One of the primary benefits of sentiment analysis lies in its ability to unveil underlying patterns within customer feedback. By identifying recurring themes or issues, businesses can pinpoint areas requiring improvement and prioritize strategic initiatives accordingly. For instance, if a particular product receives consistently positive feedback for its user-friendly interface but negative comments regarding its durability, a company can focus its efforts on enhancing product durability while maintaining its intuitive design.

Furthermore, sentiment analysis empowers businesses to tailor their strategies to better align with customer expectations. By understanding the emotions and opinions expressed by customers, organizations can fine-tune their marketing messages, product features, and service offerings to resonate more effectively with their target audience.

TABLE OF CONTENT

LIST OF SYMBOLS

SYMBOL NAME	**NOTATION**	**DESCRIPTION**
Class	Class name Visibility attribute Type=initial value Visibility operation (arglist):return type	Class represents a collection of similar entities grouped together
Association		Association represents a static relationship between classes.
Use case		A use case is an interaction between the system and other external examination.
Relational		It is used for Additional Process Communication
Control flow		It represents the control flow between the state
Data process/State		A circle in DFD represent the vertical dimension the object communication

Object lifeline		An object lifeline represents the vertical dimension then object Communication.
Message		It represents the Message exchanged
Actor		Actors are the user of the system and other external entity that react with the system

LIST OF FIGURES

LIST OF ABBREVIATIONS

NLP	Natural Language Processing
CNN	Convolutional Neural Networks
ML	Machine Learning
SMM	Semantic Matching Model
DL	Deep Learning
LSTM	Long Short-Term Memory
SVM	Support vector Machine
SSD	Single Shot Multibox Detector
YOLO	You Only Look Once
TF IDF	Term Frequency-Inverse Document Frequency
API	Application Programming Interface
UI	User Interface
NN	Neural Network
CV	Open Computer Vision
QR	Question Retrieval

CHAPTER 1

INTRODUCTION

1.1 DOMAIN OVERVIEW

Sentiment analysis is a crucial aspect of mining online user-generated content, particularly in the context of customer reviews. Customer reviews are an essential form of opinionated content that can provide valuable insights for both future customers and merchants. They offer a glimpse into the experiences and perceptions of customers, helping businesses improve their products and services. Deep learning has emerged as an effective means for solving sentiment classification problems, as it intrinsically learns a high-level representation of the data without the need for laborious feature engineering. However, the success of deep learning heavily relies on the availability of large-scale training data. In the context of customer reviews, ratings are often used as weak supervision signals for sentiment classification. These ratings reflect the overall sentiment of customer reviews and can be exploited for sentiment analysis. However, treating binarized ratings as sentiment labels can confuse a sentiment classifier, as a 5star review can contain negative sentences, and positive words can occasionally appear in 1-star reviews. To address this issue, a novel deep learning framework for review sentiment classification has been proposed.

1.2 OVERVIEW OF THE PROJECT

Our project is focused on sentiment analysis of customer reviews using deep learning techniques. The goal is to develop an automated system that can classify customer reviews into positive, negative, and neutral categories. This can help businesses understand customer opinions and improve their products or services accordingly. The research has implemented four deep-learning models, including Long Short-Term Memory, Bidirectional Recursive Neural Networks, Bidirectional Encoder Representations from Transformers, and Convolutional 1D networks. Performance metrics like Precision, Accuracy, Recall, and F1 score have been used for evaluation.

1.3 OBJECTIVE OF THE PROJECT

The objective of the customer review sentiment analysis project through deep learning is to develop a system that can automatically classify customer reviews into positive, negative, and neutral categories. This project aims to leverage deep learning techniques, specifically Long Short-Term Memory, Bidirectional Recursive Neural Networks, Bidirectional Encoder Representations from Transformers, and Convolutional 1D networks, to analyze and evaluate the best model for sentiment analysis of customer reviews. By using performance metrics like Precision, Accuracy, Recall, and F1 score for evaluation, the project seeks to create a framework that can enhance customer satisfaction and assist businesses in making data-driven decisions based on customer feedback

1.4 MOTIVATION

The motivation behind o project is to analyze customer reviews and determine their sentiment using deep learning techniques. This can help businesses to understand customer opinions and preferences, improve their products and services, and make informed decisions based on customer feedback.

CHAPTER 2

LITERATURE SURVEY

A Literature review is a text of a scholarly paper, which includes the current knowledge, including substantive findings, as well as theoretical and methodological contributions to a particular topic. Literature reviews use secondary sources and do not report new or original experimental work. A literature review usually precedes the methodology and results section.

PAPER : 1

Title : "Improving Sentiment Analysis in Social Media by Handling Lengthened Words"

Author : Ashima Kukkar

Year : 2023

Summary: This paper proposes a lexicon-based system that considers lengthened words as they are, instead of omitting or normalizing them. The system calculates the sentiment scores of lengthened words and uses them to determine the overall sentiment level of the person. The dataset used in the paper includes lengthened words, which are words whose characters are repeated more than their original length.

PAPER : 2

Title : " Text Mining and Emotion Classification on Monkey pox Twitter **Dataset : ** A Deep Learning-Natural Language Processing (NLP) Approach"

Author : Ruth Olusegun

Year : 2023

Summary : The study extracts and preprocesses 800,000 datasets, then uses the NRC Lexicon, a Python library, to predict and measure the emotional significance of each text. The project develops deep learning models based on Convolutional Neural Networks (CNN), Long Short-Term Memory (LSTM), Bidirectional LSTM (Bi

LSTM), and the combination of Convolutional Neural Networks and Long Short Term Memory (CLSTM) for emotion classification.

PAPER : 3

Title : " Sentiment Analysis in E-Commerce Platforms A Review of Current Techniques and Future Directions"

Author : Huang Huang

Year : 2022

Summary : A mobile application assists donor to find nearby orphanage and home and lets the donor contact the acceptor and share details about the availability of food and related information in just few clicks. Then the nearby acceptors can see the food ready to be donated and claim that food after confirmation with donor. This application will be an impactful changeover of many lives.

PAPER : 4

Title : " Impact of content characteristics and emotion on behavioral engagement in social media: literature review and research agenda"

Author : Melanie Schreiner

Year : 2022

Summary : This paper contributes to advancing knowledge in the field of social media engagement and provides practical insights for marketers, content creators, and platform designers to optimize their strategies and enhance user engagement.

PAPER : 5

Title : " What social media told us in the time of COVID-19: a scoping review"

Author : Shu-Feng Tsao

Year : 2022

Summary: In this paper by mapping the existing literature, this review identifies gaps, inconsistencies, and areas for future research, underscoring the dynamic nature of social media's influence during crises. The insights derived from this synthesis offer valuable guidance for policymakers, healthcare professionals, researchers, and social media practitioners in effectively leveraging social media to navigate public health emergencies. This includes strategies for enhancing health communication, combating misinformation, supporting mental well-being, and harnessing the collective power of online communities for crisis response and resilience-building efforts.

PAPER : 6

Title : " Sentiment Analysis in Social Media for Competitive Environment Using Content Analysis"

Author : Shahid Mehmood

Year : 2021

Summary: This paper contributes to advancing the realms of competitive intelligence and social media analytics by showcasing the efficacy of sentiment analysis in extracting valuable insights from online discourse. By harnessing sentiment analysis within a competitive context, businesses can derive actionable intelligence to inform strategic decision-making, refine brand positioning, and capitalize on growth opportunities.

PAPER : 7

Title : " Sentiment Analysis of Consumer Reviews Using Deep Learning"

Author : Binmahfoudh A. Hussain, M.

Year : 2022

Summary: This paper presents a novel approach to sentiment analysis of consumer reviews using deep learning techniques. We propose a deep learning model that leverages neural network architectures, such as recurrent neural networks (RNNs) or convolutional neural networks (CNNs), to automatically learn and classify sentiment from textual data. Through extensive experimentation and evaluation on a diverse dataset of consumer reviews, we demonstrate the effectiveness of our approach in accurately identifying and categorizing sentiment polarity

PAPER : 8

Title : " Evaluating Annotated Dataset of Customer Reviews For Aspect Based Sentiment Analysis"

Author : Dimple Chehal

Year : 2021

Summary: This paper discusses an approach to evaluating annotated datasets of customer reviews for Aspect Based Sentiment Analysis (ABSA). ABSA is a technique that breaks down text into its constituent aspects or features and assesses sentiment at a highly specific level, such as the performance of a smartphone's camera or the taste of a restaurant's food. This approach provides businesses with valuable insights into customer opinions, enabling them to identify areas for improvement and make data driven decisions.

PAPER : 9

Title : " Deep Learning Based Sentiment Classification :A Comparative Survey"

Author : Alhassan Mabrouk

Year : 2020

Summary : This paper presents a comprehensive literature-based survey that compares the performance of over 100 deep learning (DL)-based sentiment classification (SC) approaches using public datasets of reviews. The study aims to provide insights into the effectiveness of various DL-based SC models and identify the most promising techniques for future research.

PAPER : 10

Title : " A Survey of Sentiment Analysis from Social Media Data"

Author : Koyel Chakraborty

Year : 2022

Summary: The survey discusses the necessary stages for building a complete sentiment analysis model, including preprocessing, feature extraction, sentiment classification, and evaluation.

CHAPTER 3

SYSTEM ANALYSIS

3.1 EXISTING SYSTEM

Existing systems to spot customer reviews use both people and computers. Humans check for suspicious patterns in content and user behavior, while automated tools analyze the language and sentiments. These tools also learn from historical data and use collaborative methods. Users reporting suspicious reviews is another important part of the process. Ongoing tech improvements continue to make these systems better at catching fake reviews.

3.1.1 DISADVANTAGES

- It supports only for small and medium level data’s only.
- It gives the less accuracy.
- False Positives
- Adaptability Challenges

3.2 PROPOSED SYSTEM

The proposed system for customer review detection introduces advancements in context understanding, dynamic adaptation, privacy, and explainability. It employs a multifaceted approach, prioritizes user feedback, and emphasizes cross-platform standardization. The system aims to optimize resource usage, mitigate biases, and collaboratively address challenges in fake review detection, fostering transparency and user empowerment.

3.2.1 ADVANTAGES

- Enhanced Accuracy
- Adaptability to Deceptive Tactics
- Continuous Learning

3.3 ALGORITHMS

3.3.1 BOOSTING ALGORITHMS

Boosting algorithms are a class of machine learning techniques that aim to enhance the predictive performance of models by combining the strengths of multiple weak learners. Weak learners, often simple decision trees, are sequentially trained on the data, with each subsequent learner focusing on the mistakes of its predecessors. The key idea is to assign higher weights to misclassified instances, thereby emphasizing their importance in subsequent iterations. Gradient Boosting, AdaBoost, and XG Boost are popular boosting algorithms. Gradient Boosting optimizes model errors by minimizing the gradient of the loss function, AdaBoost assigns varying weights to instances based on their classification success, and XG Boost extends these concepts with regularization and parallel processing capabilities. Boosting algorithms excel in improving predictive accuracy and are widely utilized in diverse machine learning applications.

The life of Machine Learning programs is straightforward and can be summarized in the following points:

1. Define a question
2. Collect data
3. Visualize data
4. Train algorithm
5. Test the Algorithm
6. Collect feedback
7. Refine the algorithm
8. Loop 4-7 until the results are satisfying
9. Use the model to make a prediction

Once the algorithm gets good at drawing the right conclusions, it applies that knowledge to new sets of data.

Machine learning can be grouped into two broad learning tasks: Supervised and Unsupervised. There are many other algorithms.

3.3.2 SUPERVISED LEARNING

An algorithm uses training data and feedback from humans to learn the relationship of given inputs to a given output. For instance, a practitioner can use marketing expense and weather forecast as input data to predict the sales of cans.You can use supervised learning when the output data is known. The algorithm will predict new data.

There are two categories of supervised learning:

- Classification task
- Regression task

Classification

Imagine you want to predict the gender of a customer for a commercial. You will start gathering data on the height, weight, job, salary, purchasing basket, etc. from your customer database. You know the gender of each of your customer, it can only be male or female. The objective of the classifier will be to assign a probability of being a male or a female (i.e., the label) based on the information (i.e., features you have collected). When the model learned how to recognize male or female, you can use new data to make a prediction. For instance, you just got new information from an unknown customer, and you want to know if it is a male or female. If the classifier predicts male = 70%, it means the algorithm is sure at 70% that this customer is a male, and 30% it is a female.

The label can be of two or more classes. The above example has only two classes, but if a classifier needs to predict object, it has dozens of classes (e.g., glass, table, shoes, etc. each object represents a class)

Regression

When the output is a continuous value, the task is a regression. For instance, a financial analyst may need to forecast the value of a stock based on a range of feature like equity, previous stock performances, macroeconomics index. The system will be trained to estimate the price of the stocks with the lowest possible error.

Linear regression

Finds a way to correlate each feature to the output to help predict future values.

Logistic regression

Extension of linear regression that's used for classification tasks. The output variable 3is binary (e.g., only black or white) rather than continuous (e.g., an infinite list of potential colors) Classification.

Decision tree

Highly interpretable classification or regression model that splits data-feature values into branches at decision nodes (e.g., if a feature is a color, each possible color becomes a new branch) until a final decision output is made.

Naive Bayes

The Bayesian method is a classification method that makes use of the Bayesian theorem. The theorem updates the prior knowledge of an event with the independent probability of each feature that can affect the event. Regression Classification

Support vector machine Support Vector Machine, or SVM, is typically used for the classification task. SVM algorithm finds a hyperplane that optimally divided the classes. It is best used with a non-linear solver.

Random forest

The algorithm is built upon a decision tree to improve the accuracy drastically. Random forest generates many times simple decision trees and uses the 'majority vote' method to decide on which label to return. For the classification task, the final prediction will be the one with the most vote; while for the regression task, the average prediction of all the trees is the final prediction.

AdaBoost

Classification or regression technique that uses a multitude of models to come up with a decision but weighs them based on their accuracy in predicting the outcome.

Gradient-boosting trees

It is a state-of-the-art classification/regression technique. It is focusing on the error committed by the previous trees and tries to correct it.

3.3.3 UNSUPERVISED LEARNING

In unsupervised learning, an algorithm explores input data without being given an explicit output variable (e.g., explores customer demographic data to identify patterns).

K-means clustering

Puts data into some groups (k) that each contains data with similar characteristics (as determined by the model, not in advance by humans) .

Gaussian mixture model

A generalization of k-means clustering that provides more flexibility in the size and shape of groups (clusters Clustering).

Hierarchical clustering

Splits clusters along a hierarchical tree to form a classification system. Can be used for Cluster loyalty-card customer. Clustering Recommender system help to define the relevant data for making a recommendation.

PCA/T-SNE

Mostly used to decrease the dimensionality of the data. The algorithms reduce the number of features to 3 or 4 vectors with the highest variances.

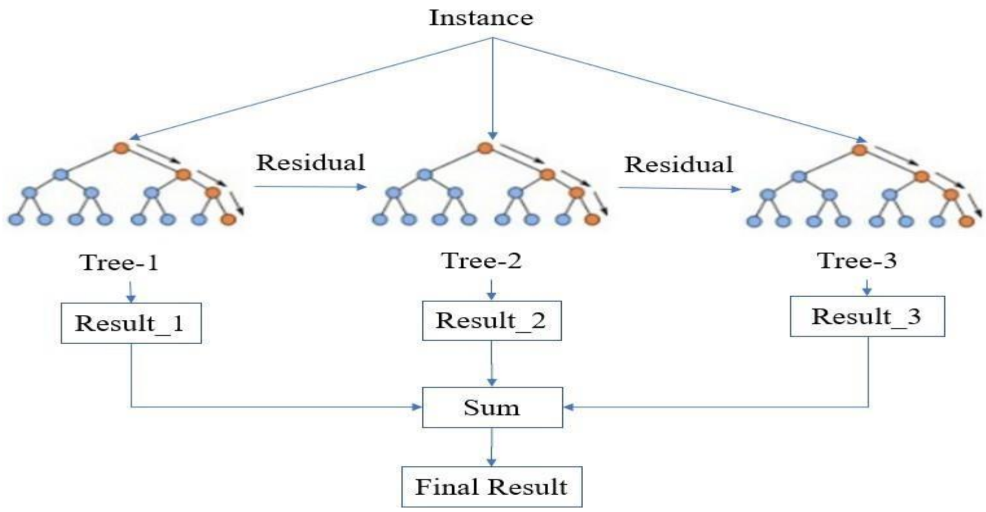


Figure 3.1 Algorithm

CHAPTER 4

SYSTEM SPECIFICATION

Develop a backend system responsible for data processing, model training, and inference. This could be implemented using frameworks like TensorFlow, PyTorch, or scikit-learn. Design a user-friendly frontend interface for interacting with the sentiment analysis system. This could be a web application, mobile app, or desktop application depending on the target platform. Identify and collect customer reviews from various sources such as e-commerce platforms, social media, and review websites. Implement mechanisms for data retrieval, including web scraping or API integration. Gather feedback from users or stakeholders on the accuracy and usefulness of the sentiment analysis results.

4.1 FUNCTIONAL REQUIREMENTS

The system should collect feedback from users or stakeholders on the accuracy and usefulness of sentiment analysis results. It should use feedback to iteratively improve the performance of sentiment analysis models, such as retraining models on new data or fine-tuning hyperparameters. The system should provide a user-friendly interface for users to interact with sentiment analysis results. It should generate reports and visualizations to present sentiment analysis insights in a comprehensible manner .It allow users to customize analysis parameters and view detailed information about sentiment predictions.

4.2 NON-FUNCTIONAL REQUIREMENTS

The system should provide timely responses to sentiment analysis requests, with low latency for real-time or near-real-time analysis. The system should be able to handle a high volume of concurrent sentiment analysis requests efficiently, ensuring high throughput. The system should ensure the privacy and confidentiality of customer data, adhering to data protection regulations and industry standards. The system should monitor key performance indicators (KPIs) related to system health, resource utilization, and user satisfaction to ensure continuous improvement and optimization.

4.3 HARDWARE REQUIREMENTS:

- OS : Windows 7,8 and 10 (32 and 64 bit)
- RAM : 4 GB

4.4 SOFTWARE REQUIREMENTS:

- Anaconda Navigator
- Python language
- Jupiter notebook

4.5 FEASIBILITY STUDY

Assess the availability and accessibility of customer review data from various sources such as e-commerce platforms, social media, and review websites. Determine if sufficient data can be obtained for training and testing machine learning models. Evaluate the suitability of different machine learning algorithms (e.g., Logistic Regression, Support Vector Machines, Naive Bayes, Neural Networks) for sentiment analysis tasks based on the characteristics of the data and the requirements of the project.

Estimate the costs associated with data acquisition, data preprocessing, model development and training, infrastructure setup (e.g., computing resources), deployment, maintenance, and ongoing operation of the sentiment analysis system. Evaluate the potential benefits of implementing the sentiment analysis system, such as improved customer satisfaction, better decision-making, and increased revenue or cost savings, compared to the projected costs.

Developing and administering questionnaires to interested stakeholders such as potential users of the information system.

Observing or monitoring users of the current system to determine their needs as well as their satisfaction and dissatisfaction with the current system. Collecting, examining, and analyzing documents, reports, layouts, procedures, manuals, and any other documentation relating to the operations of the current system. Modeling, observing, and simulating the work activities of the system.

The goal of the feasibility study is to consider alternative information systems solutions, evaluate their feasibility, and propose the alternative most suitable to the organization. The feasibility of a proposed solution is evaluated in terms of its components.

4.5.1 ECONOMICAL FEASIBILITY

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

4.5.2 TECHNICAL FEASIBILITY

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client.

The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

4.5.3 SOCIAL FEASIBILITY

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

4.5.4 OPERATIONAL FEASIBILITY

The ability, desire, and willingness of the stakeholders to use, support, and operate the proposed computer information system. The stakeholders include management, employees, customers, and suppliers. The stakeholders are interested in systems that are easy to operate, make few, if any, errors, produce the desired information, and fall within the objectives of the organization.

Resources: Assess the availability of human resources, technology, infrastructure, and financial resources needed to implement and sustain the project. Consider if the organization has the necessary skills, expertise, and funding to support the project's operational requirements.

Compatibility: Evaluate the compatibility of the proposed system with existing organizational processes, systems, and workflows. Determine if the project can seamlessly integrate into the current operational environment or if significant modifications or adaptations are necessary.

CHAPTER 5

SYSTEM DESIGN AND DEVELOPMENT

5.1 INTRODUCTION

Systems design is the process or art of defining the architecture, components, modules, interfaces, and data for a system to satisfy specified requirements. One could see it as the application of systems theory to product development. There is some overlap and synergy with the disciplines of systems analysis, systems architecture and systems engineering.

5.2 FILE DESIGN

File design is the design of the database and it contains information about the files used in the system. In database design the tables constructed, fields in the table their data types and in the other part it tells about the extensions of the file used in the development.

5.3 INPUT DESIGN

The input design is the process of converting the user-oriented inputs in to the computer-based format. The goal of designing input data is to make the automation as easy and free from errors as possible. For providing a good input design for the application easy data input and selection features are adopted. The input design requirements such as user friendliness, consistent format and interactive dialogue for giving the right message and help for the user at right time are also considered for the development of the project.

5.4 OUTPUT DESIGN

When designing output, systems analyst must accomplish the following;

- ➢ Determine what information to present
- ➢ Decide whether to display, print the information and select the output medium
- ➢ Arrange the presentation of information in an acceptable format

Accomplishing the general activities listed above will require specific decisions, such as whether to use preprinted forms when preparing reports and documents, how many line to plan on printed page, or whether to user graphics and colour. The output design is specified on layout forms, sheets that describe the location characteristics (such as length and type), and format of the column headings and pagination. As we indicated at the beginning of this discussion, these elements are analogous to an architect blueprint that shows the location of each component.

Output Format: Determine the format of the output based on the nature of the information and the needs of the users. It could include textual reports, graphical charts or graphs, tables, dashboards, or a combination of these formats.

Information Organization: Organize the information in a logical and structured manner. Use headings, subheadings, and sections to provide clarity and ease of understanding. Group related data together and use appropriate labels to enhance readability.

5.5 SYSTEM ARCHITECTURE

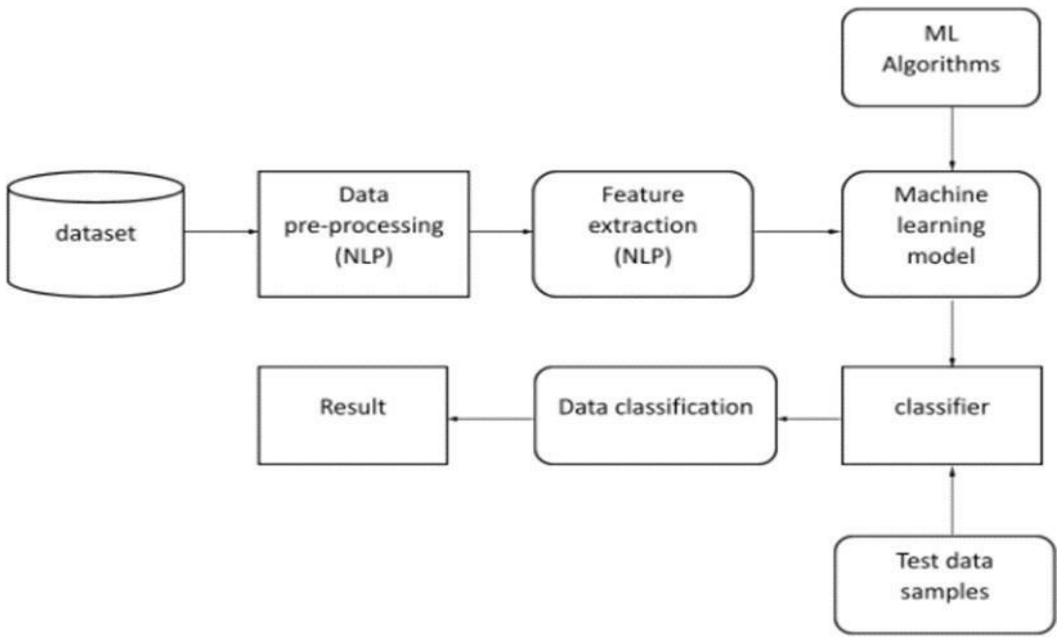


Figure 5.1 System Architecture

5.6 USECASE DIAGRAM

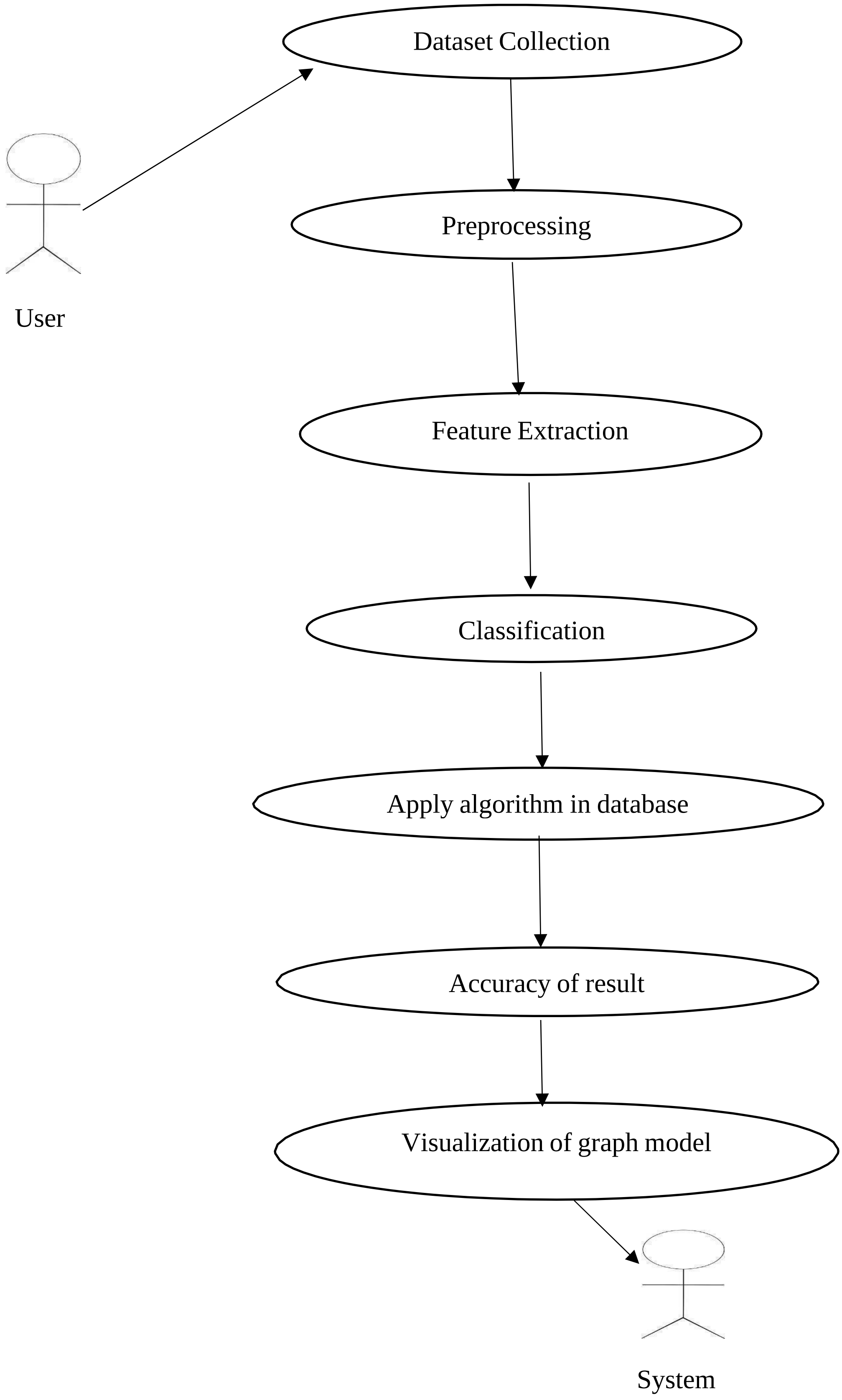


Figure 5.2 Use case Diagram

5.7 DATA FLOW DIAGRAM

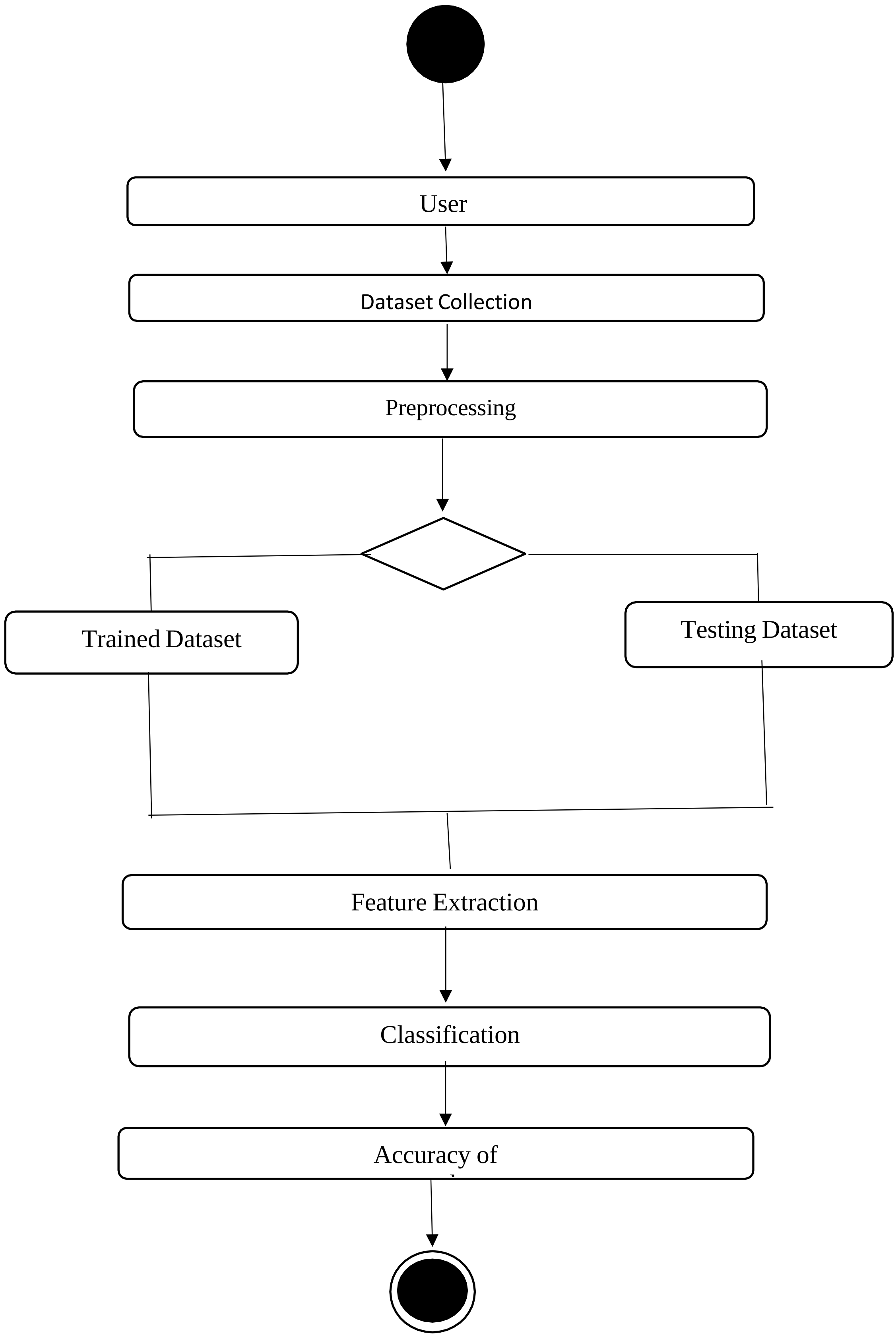


Figure 5.3 Dataflow diagram

5.8 CLASS DIAGRAM

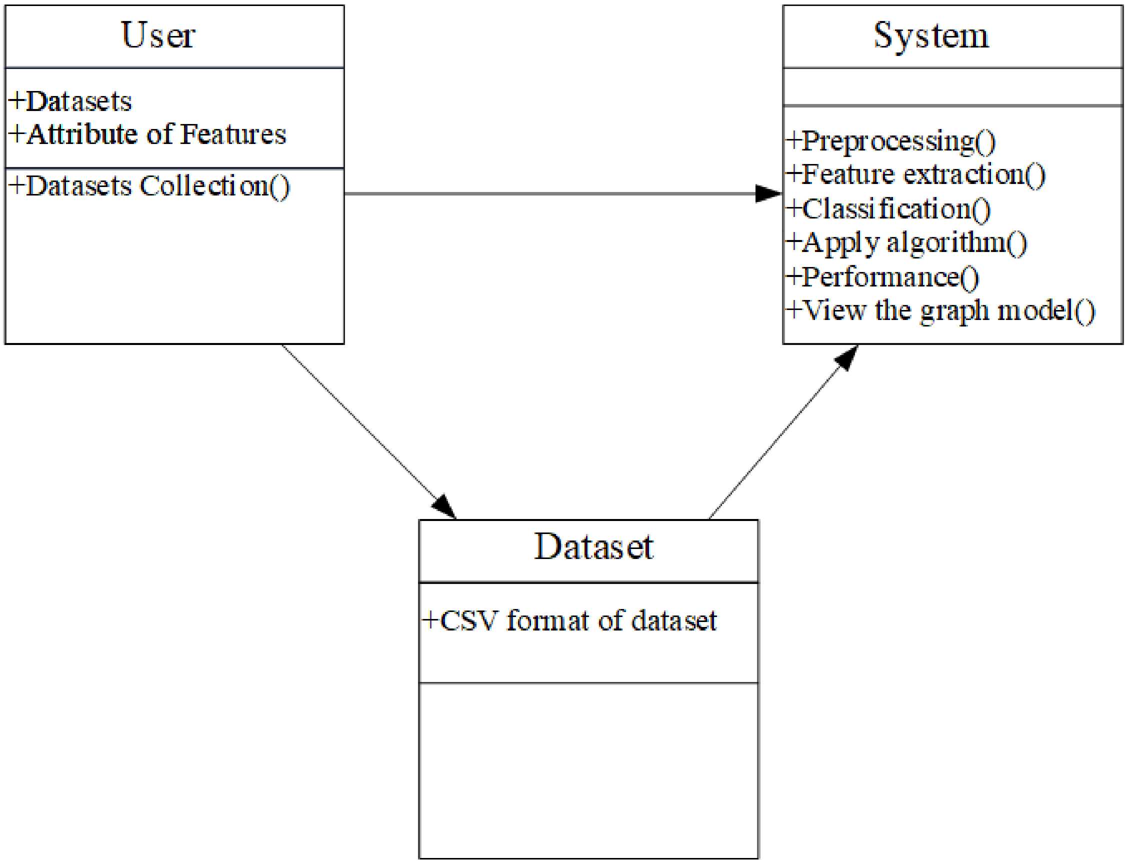


Figure 5.4 Class Diagram

CHAPTER 6

TECHNOLOGIES

6.1 PYTHON LANGUAGE

Python is an object-oriented programming language created by Guido Rossum in 1989. It is ideally designed for rapid prototyping of complex applications. It has interfaces to many OS system calls and libraries and is extensible to C or C++. Many large companies use the Python programming language include NASA, Google, YouTube, BitTorrent, etc. Python programming is widely used in Artificial Intelligence, Natural Language Generation, Neural Networks and other advanced fields of Computer Science. Python had deep focus on code readability & this class will teach you python from basics.

6.2 PYTHON PROGRAMMING CHARACTERISTICS

- It provides rich data types and easier to read syntax than any other programming languages
- It is a platform independent scripted language with full access to operating system API's
- Compared to other programming languages, it allows more run-time flexibility
- It includes the basic text manipulation facilities of Perl and Awk
- A module in Python may have one or more classes and free functions
- Libraries in Pythons are cross-platform compatible with Linux, Macintosh, and Windows
- For building large applications, Python can be compiled to byte-code
- Python supports functional and structured programming as well as OOP
- It supports interactive mode that allows interacting Testing and debugging of snippets of code

- In Python, since there is no compilation step, editing, debugging and testing is fast.

6.3 PYTHON ENVIRONMENT

Python is available on a wide variety of platforms including Linux and Mac OS X. Let's understand how to set up our Python environment.

REQUIREMENTS ANAYLSIS

- Python
- Anaconda Navigator
- Python built-in modules
    - Numpy
    - Pandas
    - Matplotlib
    - Sklearn
    - Seaborm

6.4 APPLICATIONS OF PYTHON PROGRAMMING

6.4.1 WEB APPLICATIONS

You can create scalable Web Apps using frameworks and CMS (Content Management System) that are built on Python. Some of the popular platforms for creating Web Apps are: Django, Flask, Pyramid, Plone, Django CMS. Sites like Mozilla, Reddit, Instagram and PBS are written in Python.

6.4.2 SCIENTIFIC AND NUMERIC COMPUTING

There are numerous libraries available in Python for scientific and numeric computing. There are libraries like: SciPy and NumPy that are used in general purpose computing. And, there are specific libraries like: EarthPy for earth science, AstroPy for Astronomy and so on. Also, the language is heavily used in machine learning, data mining and deep learning.

6.4.3 CREATING SOFTWARE PROTOTYPES

Python is slow compared to compiled languages like C++ and Java. It might not be a good choice if resources are limited and efficiency is a must. However, Python is a great language for creating prototypes. For example: You can use Pygame (library for creating games) to create your game's prototype first. If you like the prototype, you can use language like C++ to create the actual game.

6.4.4 GOOD LANGUAGE TO TEACH PROGRAMMING

Python is used by many companies to teach programming to kids and newbies. It is a good language with a lot of features and capabilities. Yet, it's one of the easiest language to learn because of its simple easy-to-use syntax.

6.4.5 OPENCV PACKAGE

Python is a general purpose programming language started by Guido van Rossum, which became very popular in short time mainly because of its simplicity and code readability. It enables the programmer to express his ideas in fewer lines of code without reducing any readability. Compared to other languages like C/C++, Python is slower. But another important feature of Python is that it can be easily extended with C/C++. This feature helps us to write computationally intensive codes in C/C++ and create a Python wrapper for it so that we can use these wrappers as Python modules. This gives us two advantages: first, our code is as fast as original C/C++ code (since it is the actual C++ code working in background) and second, it is very easy to code in Python. This is how OpenCV- Python works, it is a Python wrapper around original C++ implementation.

Besides that, several other libraries like SciPy, Matplotlib which supports Numpy can be used with this. So OpenCV-Python is an appropriate tool for fast prototyping of computer vision problems.

CHAPTER 7

FEATURES OF ANACONDA NAVIGATOR

Anaconda is a free and open source, easy to install distribution of Python and R programming languages. Anaconda provides a working environment which is used for scientific computing, data science, statistical analysis and machine learning. The latest distribution of Anaconda is Anaconda 5.3 and is released in October, 2018. It has the conda package, environment manager and a collection of 1000+ open source.

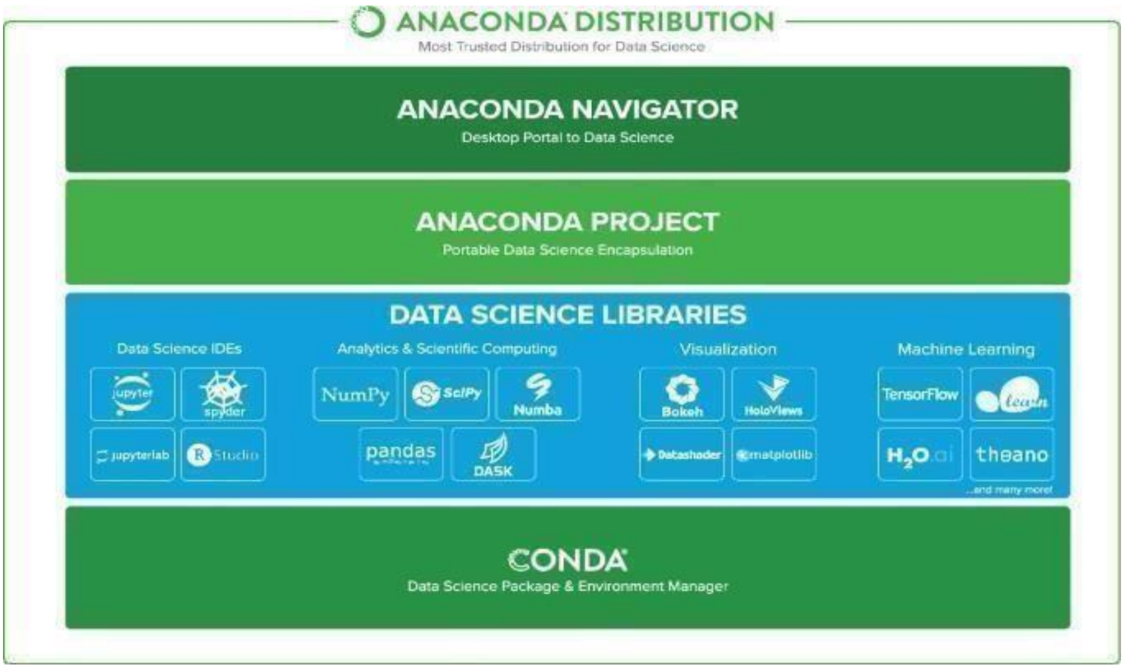


Figure 7.1 Anaconda distribution

What is Anaconda Navigator?

Anaconda Navigator is a desktop graphical user interface (GUI) included in the Anaconda distribution. It allows us to launch applications provided in the Anaconda distribution and easily manage conda packages, environments and channels without the use of command-line commands. It is available for Windows, macOS and Linux.

7.1 APPLICATION PROVIDED IN ANACONDA DISTRIBUTION

The Anaconda distribution comes with the following applications along with Anaconda Navigator.

1. JupyterLab
2. Jupyter Notebook
3. Qt Console
4. Spyder
5. Glueviz
6. Orange3
7. RStudio
8. Visual Studio Code

- **Jupyter Lab:** This is an extensible working environment for interactive and reproducible computing, based on the Jupyter Notebook and Architecture.
- **Jupyter Notebook:** This is a web-based, interactive computing notebook environment. We can edit and run human-readable docs while describing the data analysis.
- **Qt Console:** It is the PyQt GUI that supports inline figures, proper multiline editing with syntax highlighting, graphical calltips and more.
- **Spyder:** Spyder is a scientific Python Development Environment. It is a powerful Python IDE with advanced editing, interactive testing, debugging and introspection features.
- **VS Code:** It is a streamlined code editor with support for development operations like debugging, task running and version control.
- **Glueviz:** This is used for multidimensional data visualization across files. It explores relationships within and among related datasets.
- **Orange 3:** It is a component-based data mining framework. This can be used for data visualization and data analysis

- **R studio:** It is a set of integrated tools designed to help you be more productive with R. It includes R essentials and notebooks. Advanced conda users can also build your own Navigator applications.

How can I run code with Navigator?

The simplest way is with Spyder. From the Navigator Home tab, click Spyder, and write and execute your code. You can also use Jupyter Notebooks the same way. Jupyter Notebooks are an increasingly popular system that combine your code, descriptive text, output, images and interactive interfaces into a single notebook file that is edited, viewed and used in a web browser.

New Features of Anaconda 5.3

Figure 7.3 Anaconda

Compiled with Latest Python release: Anaconda 5.3 is compiled with Python manual.

Better Reliability: The reliability of Anaconda has been improved in the latest release by capturing and storing the package metadata for installed packages.

Enhanced CPU Performance : The Intel Math Kernel Library 2019 for Deep Neural Networks(MKL 2019) has been introduced in Anaconda 5.3 distribution. Users deploying Tensor flow can make use of MKL 2019 for Deep Neural Networks. These Python binary packages are provided to achieve high CPU performance.

7.2 SOFTWARE TESTING

Software testing can be stated as the process of verifying and validating whether a software or application is bug-free, meets the technical requirements as guided by its design and development, and meets the user requirements effectively and efficiently by handling all the exceptional and boundary cases.

The process of software testing aims not only at finding faults in the existing software but also at finding measures to improve the software in terms of efficiency, accuracy, and usability. It mainly aims at measuring the specification, functionality, and performance of a software program or application.

7.2.1 GENERAL

In a generalized way, we can say that the system testing is a type of testing in which the main aim is to make sure that system performs efficiently and seamlessly. The process of testing is applied to a program with the main aim to discover an unprecedented error, an error which otherwise could have damaged the future of the software. Test cases which brings up a high possibility of discovering and error is considered successful. This successful test helps to answer the still unknown errors.

Software testing encompasses various types, including functional testing, performance testing, security testing, usability testing, compatibility testing, and more. Each type focuses on specific aspects of the software's quality.

Testing can be conducted at different levels, such as unit testing, integration testing, system testing, and acceptance testing. Each level verifies different aspects of the software, from individual components to the entire system.

CHAPTER 8

TEST CASES

Testing, as already explained earlier, is the process of discovering all possible weak-points in the finalized software product. Testing helps to counter the working of sub-assemblies, components, assembly and the complete result. The software is taken through different exercises with the main aim of making sure that software meets the business requirement and user-expectations and doesn’t fails abruptly. Several types of tests are used today. Each test type addresses a specific testing requirement.

8.1 TESTING TECHNIQUES

A test plan is a document which describes approach, its scope, its resources and the schedule of aimed testing exercises. It helps to identify almost other test item, the features which are to be tested, its tasks, how will everyone do each task, how much the tester is independent, the environment in which the test is taking place, its technique of design plus the both the end criteria which is used, also rational of choice of theirs, and whatever kind of risk which requires emergency planning. It can be also referred to as the record of the process of test planning. Test plans are usually prepared with signification input from test engineer.

1. UNIT TESTING

Unit testing for customer review sentiment analysis through deep learning involves testing individual components or units of the system to ensure they perform as expected. Since deep learning models are a critical component of sentiment analysis systems, unit testing would primarily focus on testing the functionality and behavior of these models.

2. FUNCTIONAL TESTING

The functional tests help in providing the systematic representation that functions tested are available and specified by technical requirement, documentation of the system and the user manual.

3. INTEGRATION TESTING

Integration testing for customer review sentiment analysis through deep learning involves verifying that individual components of the system work together correctly as a whole. Since sentiment analysis systems often comprise multiple modules such as data preprocessing, model inference, and user interface, integration testing ensures that these components integrate seamlessly**.**

4. SYSTEM TESTING

System testing, as the name suggests, is the type of testing in which ensure that the software system meet the business requirements and aim. Testing of the configuration is taken place here to ensure predictable result and thus analysis of it. System testing is relied on the description of process and its flow, stressing on pre driven process and the points of integration.

5.WHITE BOX TESTING

The white box testing is the type of testing in which the internal components of the system software is open and can be processed by the tester. It is therefore a complex type of testing process. All the data structure, components etc. are tested by the tester himself to find out a possible bug or error. It is used in situation in which the black box is incapable of finding out a bug. It is a complex type of testing which takes more time to get applied.

6. BLACK BOX TESTING

The black box testing is the type of testing in which the internal components of the software is hidden and only the input and output of the system is the key for the tester to find out a bug. It is therefore a simple type of testing. A programmer with basic knowledge can also process this type of testing. It is less time consuming as compared to the white box testing. It is very successful for software which are less complex are straight-forward in nature. It is also less costly than white box testing.

7. ACCEPTANCE TESTING

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

Acceptance testing, also known as user acceptance testing (UAT), is a type of testing conducted to determine if a software system meets the requirements and expectations of the end users or stakeholders. It is the final phase of the testing process before the software is released for production use. Acceptance testing for customer review sentiment analysis through deep learning involves verifying whether the system meets the requirements and expectations of the end-users .

CHAPTER 9

MODULES

9.1 MODULE DESCRIPTION

9.1.1 DATA PREPROCESSING

Data preprocessing is a crucial step in customer review sentiment analysis through deep learning. It involves cleaning, transforming, and preparing the raw text data to make it suitable for input into a deep learning model. Here's a typical data preprocessing pipeline for this task: Data collection, Text cleaning, Tokenization, Stopword removal, Normalization, padding, Vectorization, Splitting data, Handling imbalanced data, Save pre-processed data.

9.1.2 DATA INTEGRATION

Data integration in the context of customer review sentiment analysis involves combining and merging data from multiple sources to create a comprehensive dataset for training and testing deep learning models, Such as Identify data sources, Data extraction, Data cleaning and preprocessing, Data alignment, Duplicate removal, etc….

9.1.3 DATA CLEANING

Data cleaning is a crucial step in preparing customer review data for sentiment analysis through deep learning. It involves identifying and correcting errors, inconsistencies, and irrelevant information in the raw data.

9.1.4 DATA TRANSFORMATION

Data transformation is the process of converting raw customer review data into a format suitable for analysis and modeling. In the context of sentiment analysis through deep learning, data transformation involves converting text data into numerical representations that can be input into machine learning algorithms.

9.1.5 TRAINING MODEL

Training a deep learning model for customer review sentiment analysis involves several steps , including selecting an appropriate architecture ,preparing the data, defining the model ,training the model, and evaluating its performance.

9.1.6 PREDICTION

Predictions for customer review sentiment analysis through deep learning can vary based on the specific model architecture, dataset quality, feature engineering techniques, and hyperparameter tuning. However, here are some general trends and considerations:

- **Improved Accuracy:** Deep learning models, especially recurrent neural networks (RNNs) and transformers like BERT, have shown significant improvements in sentiment analysis tasks compared to traditional machine learning approaches. These models can capture more complex patterns and semantics in text data, leading to higher accuracy.
- **Fine-grained Analysis:** Deep learning models can often provide more nuanced sentiment analysis by distinguishing between different aspects of sentiment (positive, negative, neutral) and even detecting subtle emotions or sentiments.
- **Transfer Learning:** Transfer learning, where pre-trained models are fine-tuned on specific tasks, has become a common practice in deep learning for sentiment analysis. Models pre-trained on large text corpora (such as BERT, GPT, etc.) can be fine-tuned on smaller datasets for specific sentiment analysis tasks, leading to better performance, especially with limited labeled data.
- **Multimodal Analysis:** With the rise of social media and e-commerce platforms, customer reviews may include not only text but also images, videos, or emojis. Deep learning models capable of handling multimodal data are likely to become more prevalent for sentiment analysis tasks to leverage information from various modalities.

- **Interpretability Challenges:** Deep learning models, especially complex architectures like deep neural networks, often lack interpretability, making it challenging to understand why a particular prediction was made. Research in the field is focusing on developing techniques to explain the decisions of deep learning models, which is crucial for applications like customer review sentiment analysis, where trust and transparency are essential.
- **Domain Adaptation:** Customer reviews span various domains, from electronics to hospitality. Deep learning models may require domain adaptation techniques to generalize well across different domains. Domain-specific pre-training or domain-adversarial training methods are examples of approaches used to address this challenge.
- **Real-time Inference:** Deploying deep learning models for real-time sentiment analysis in production environments poses challenges related to computational efficiency and latency. Optimizing model architectures and leveraging hardware accelerators like GPUs or TPUs can help address these challenges.

Data flow diagram:

level 0:

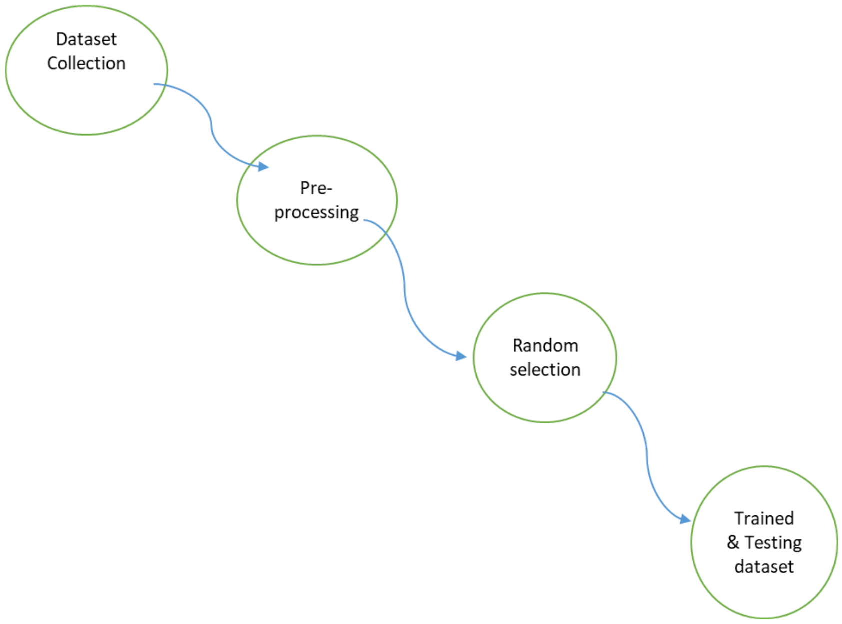


Figure 9.2 level 0

level 1:

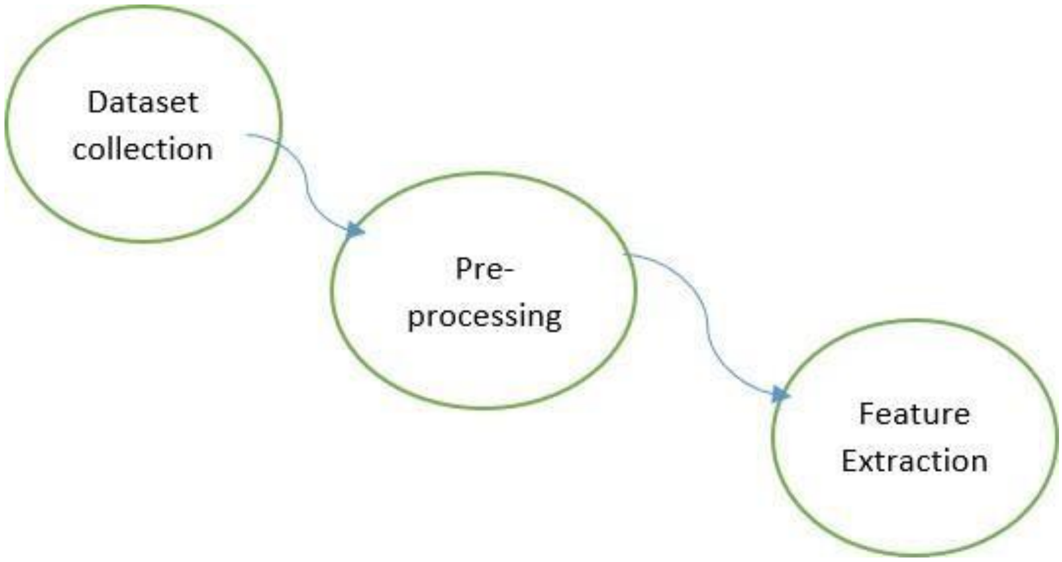


Figure 9.3 level 1

level 2:

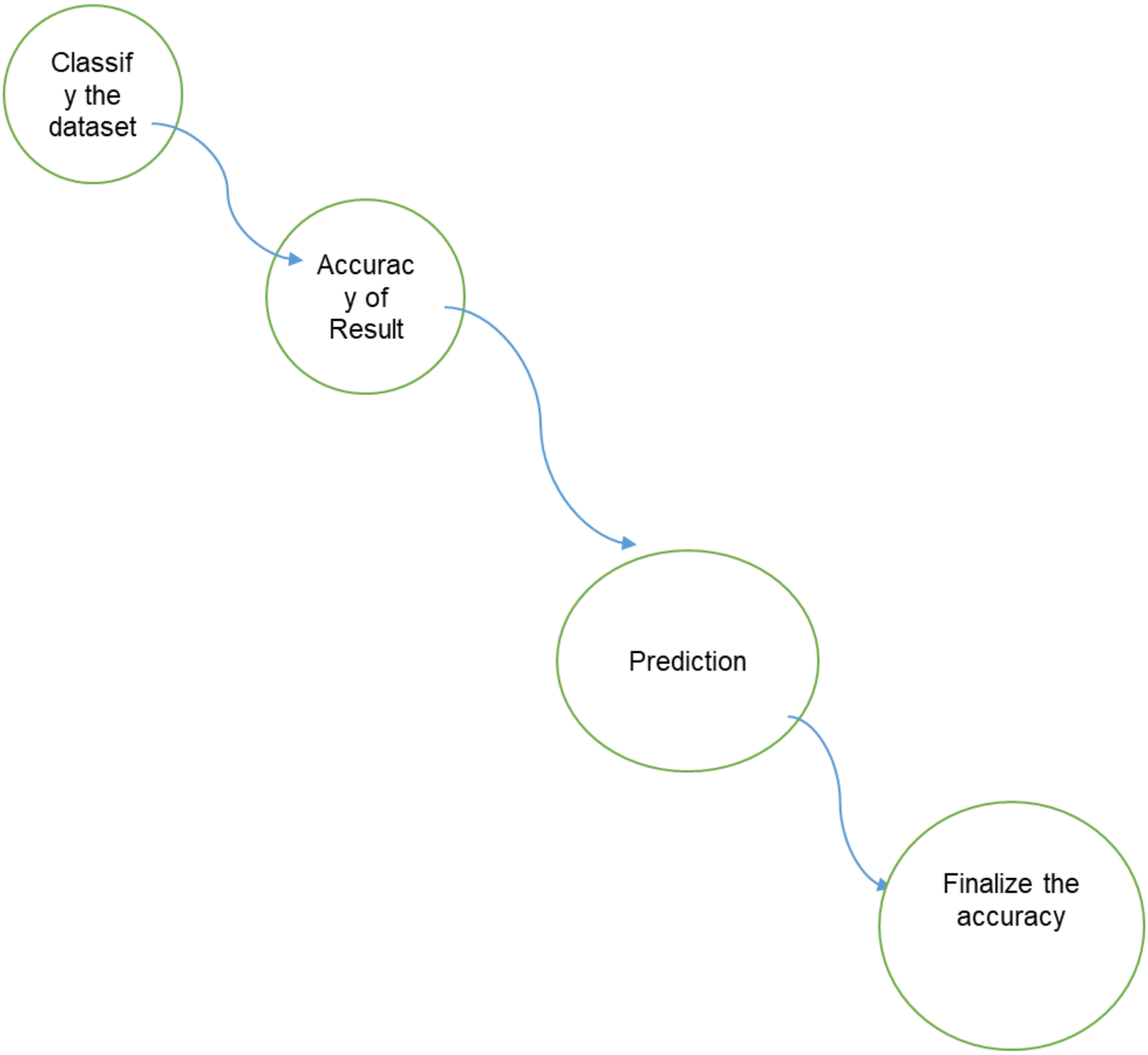


Figure 9.4 level 2

CHAPTER 10
CODING

```
import numpy as np
import pandas as pd
import matplotlib as mpl
import matplotlib.pyplot as plt
from matplotlib import style
from sklearn import preprocessing
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split

# read the data
df=pd.read_csv('Reviews_clean.csv')

df.head()

df.tail()

df.info

df.isnull().sum()

df.describe()

df=df.dropna(axis = 0)

df.shape

df.head()

data_pos=data[data["Rating"].isin([4,5])]

data_pos.head()

data_neg=data[data["Rating"].isin([1,2])]

data_neg.head()

data.Rating.value_counts()
```

```
# Plot histogram grid
df.hist(figsize=(15,15), xrot=-45, bins=10) ## Display the labels rotated by 45 degress

# Clear the text "residue"
plt.show()

import seaborn as sns
sns.kdeplot(df['Rating'])

import seaborn as sns
sns.kdeplot(df['Review Votes'])

sns.boxplot( y=df["Rating"] )

sns.heatmap(df.corr(), annot = True)
plt.show()

import seaborn as sns
sns.barplot(x=data.Rating.value_counts().index,y=data.Rating.value_counts().values)

data_filtered=pd.concat([data_pos[:20000],data_neg[:20000]])

data_filtered.shape

sns.barplot(x=data_filtered.Rating.value_counts().index,y=data_filtered.Rating.value_counts(
).values)

data_filtered["r"]=1

data_filtered["r"][data_filtered["Rating"].isin([1,2])]= 0

data_filtered.head()

data_filtered.tail()

data_filtered.r.value_counts()
```

FEATURE EXTRACTION

```
from nltk.tokenize import word_tokenize
from nltk.corpus import stopwords
from nltk.stem.porter import *
import string
import nltk
```

```
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.metrics import confusion_matrix

import re
def remove_pattern(input_txt,pattern):
    r = re.findall(pattern,input_txt)
    for i in r:
        input_txt = re.sub(i,'',input_txt)#sub(characters we want to keep, removed character
replaced by space, string to work on)

    return input_txt
data_filtered["Reviews"] = np.vectorize(remove_pattern)(data_filtered["Reviews"],
"@[\w]*")
data_filtered["Reviews"] = data_filtered["Reviews"].str.replace("[^a-zA-Z#]", " ")
data_filtered["Reviews"] = data_filtered["Reviews"].apply(lambda x: ' '.join([w for w in
x.split() if len(w)>3 ])) #tokenization
tokenized = data_filtered["Reviews"].apply(lambda x: x.split())
import nltk
from nltk.stem.porter import *
stemmer = PorterStemmer()
tokenized = tokenized.apply(lambda x : [stemmer.stem(i) for i in x])
tokenized.head()
all_words = ' '.join([text for text in data_filtered["Reviews"]])
data_filtered["Reviews"]
#Spilt Train And Test data

from sklearn.model_selection import train_test_split

X_train_data,x_test_data,Y_train_data,y_test_data=train_test_split(data_filtered["Reviews"],
data_filtered["r"],test_size=0.2)
Y_train_data.head()
X_train_data.head()
```

SVM

```
from sklearn.svm import LinearSVC
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.pipeline import Pipeline

tfidf = TfidfVectorizer()

classifier = LinearSVC()

clf = Pipeline([('tfidf',tfidf), ('clf',classifier)])
# it will first do vectorization and then it will do classification
```

```
clf.fit(X_train_data, Y_train_data)

from sklearn.metrics import accuracy_score, classification_report, confusion_matrix

y_pred = clf.predict(x_test_data)
print(y_pred)
cm=confusion_matrix(y_test_data, y_pred)
print(cm)

accuracy_score(y_test_data, y_pred)
print(f'\nClassification Report:\n{classification_report(y_test_data,y_pred)}')
```

MULITINOMIALNB

```
# Multinomial

from sklearn.naive_bayes import MultinomialNB
mul_model = MultinomialNB()

clf = Pipeline([('tfidf',tfidf), ('clf',mul_model)])
# it will first do vectorization and then it will do classification

clf.fit(X_train_data, Y_train_data)

from sklearn.metrics import accuracy_score, classification_report, confusion_matrix

y_pred = clf.predict(x_test_data)
print(y_pred)
cm=confusion_matrix(y_test_data, y_pred)
print(cm)

accuracy_score(y_test_data, y_pred)
```

RANDOMFORESTCLASSIFIER :

```
from sklearn.ensemble import RandomForestClassifier
rf_model = RandomForestClassifier()
clf = Pipeline([('tfidf',tfidf), ('clf',rf_model)])
# it will first do vectorization and then it will do classification

clf.fit(X_train_data, Y_train_data)

from sklearn.metrics import accuracy_score, classification_report, confusion_matrix
```

```
y_pred = clf.predict(x_test_data)
print(y_pred)
cm=confusion_matrix(y_test_data, y_pred)
print(cm)

accuracy_score(y_test_data, y_pred)
```

DECISIONTREE CLASSIFIER

```
from sklearn import tree

tree_model = tree.DecisionTreeClassifier()

clf = Pipeline([('tfidf',tfidf), ('clf',tree_model)])
# it will first do vectorization and then it will do classification

clf.fit(X_train_data, Y_train_data)

from sklearn.metrics import accuracy_score, classification_report, confusion_matrix

y_pred = clf.predict(x_test_data)
print(y_pred)
cm=confusion_matrix(y_test_data, y_pred)
print(cm)

accuracy_score(y_test_data, y_pred)
```

LOGISTIC REGRESSION

```
from sklearn.linear_model import LogisticRegression

lr_model = LogisticRegression()
clf = Pipeline([('tfidf',tfidf), ('clf',lr_model)])
# it will first do vectorization and then it will do classification

clf.fit(X_train_data, Y_train_data)

from sklearn.metrics import accuracy_score, classification_report, confusion_matrix

y_pred = clf.predict(x_test_data)
print(y_pred)
cm=confusion_matrix(y_test_data, y_pred)
```

```
print(cm)

accuracy_score(y_test_data, y_pred)


from tpot import TPOTClassifier
from sklearn.datasets import load_iris
from sklearn.model_selection import train_test_split

import warnings
warnings.filterwarnings('ignore')

iris = load_iris()
iris.data[0:5], iris.target
X = iris.data
target = iris.target
names = iris.target_names

X_train, X_test, y_train, y_test = train_test_split(iris.data, iris.target,
                                                    train_size=0.75, test_size=0.25)
X_train.shape, X_test.shape, y_train.shape, y_test.shape

tpot = TPOTClassifier(verbosity=2, max_time_mins=10)
tpot.fit(X_train, y_train)
print(tpot.score(X_test, y_test))

tpot.fitted_pipeline_

print(tpot.score(X_test, y_test))

sdf = df[['Reviews','Rating']]
sdf.head(2)

def assign_sentiment(Rating):
    if float(Rating) >= 3:
        return  "Positive"
    elif float(Rating) <= 2:
        return "neutral"
    else:
        return "Negative"

sdf['sentiment'] = sdf['Rating'].apply(assign_sentiment)
sdf.head(3)
```

```
sdf.drop('Rating', inplace=True, axis=1)
sdf.head()

sdf['Reviews'] = sdf['Reviews'].astype(str)
sdf.head()
sdf = sdf[sdf['Reviews'].map(len) <= 2000]
sdf.head()
sdf.dropna(inplace=True)
sdf.reset_index(inplace=True, drop=True)

actual_sentiment = sdf['sentiment']

actual_output = pd.DataFrame(actual_sentiment)
actual_output
```

CHAPTER 11

SCREENSHOTS

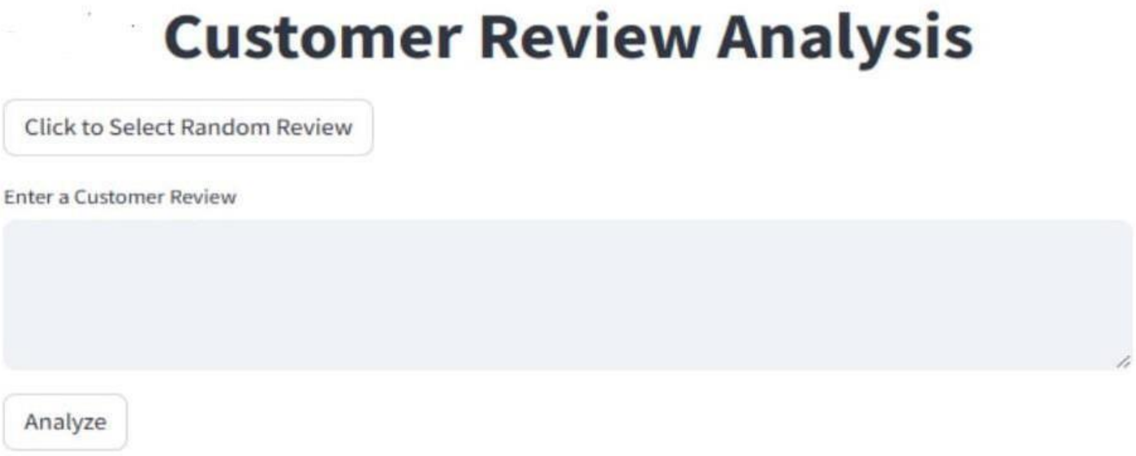


Figure 11.1 Screenshot 1

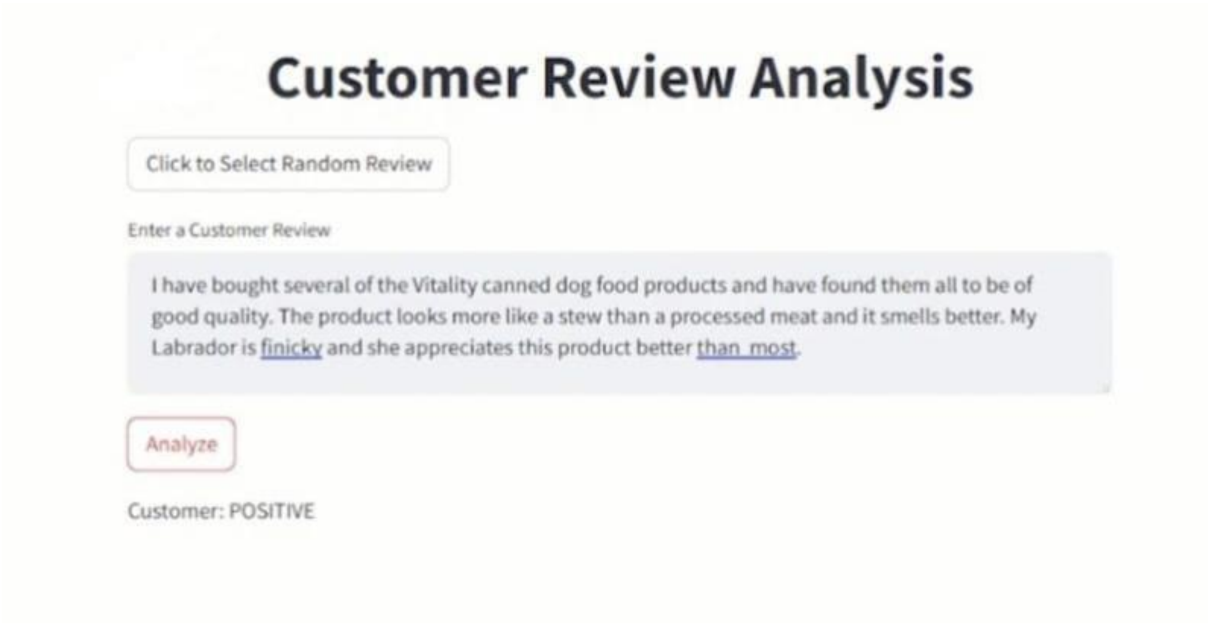


Figure 11.2 Screenshot 2

CHAPTER 12

CONCLUSION

The proposed Customer Review Sentiment analysis Through Deep learning is a significant advancement in the field of sentiment analysis, offering enhanced accuracy through advanced NLP and dynamic adaptation. This system is designed to evaluate the compatibility of customer reviews with their corresponding ratings, which is essential for e-commerce platforms like Amazon.com. The proposed model employs sentiment analysis using deep learning on Amazon.com product review data, converting reviews to vectors using paragraph vector and training a recurrent neural network with gated recurrent unit. This approach incorporates both semantic relationships of review text and product information, providing a more comprehensive analysis. The system also focuses on user empowerment, privacy, and explainability, striving for transparency and ethical practices. It aims to optimize resource usage and mitigate biases, ensuring fairness and practicality. The model also fosters a more trustworthy and informed online environment for users and businesses by tackling review-rating mismatches and promoting cross-platform consistency. The success of deep learning heavily relies on the availability of large-scale training data, which is addressed by leveraging weak supervision signals provided by review ratings. The proposed model also uses Natural Language Processing (NLP) to automate the process of analyzing reviews, achieving a maximum validation accuracy of 79.83% when using Fast Text as word embedding and the Multi-channel deep learning approach.

In conclusion, the proposed Customer Review Sentiment analysis Through Deep learning is a robust and comprehensive system that offers enhanced accuracy, transparency, and ethical practices, fostering a more trustworthy and informed online environment for users and businesses.

REFERENCES

[1]. M. L. Jensen, J. M. Averbeck, Z. Zhang, and K. B. Wright, ‘‘Credibility of anonymous online product reviews: A language expectancy perspective,’’ J. Manage. Inf. Syst., vol. 30, no. 1, pp. 293–324, Jul. 2013.

[2]. G. Cui, H.-K. Lui, and X. Guo, ‘‘The effect of online consumer reviews on new product sales,’’ Int. J. Electron. Commerce, vol. 17, no. 1, pp. 39–58, Oct. 2012.

[3]. D. Zhang, L. Zhou, J. L. Kehoe, and I. Y. Kilic, ‘‘What online reviewer behaviors really matter? Effects of verbal and nonverbal behaviors on detection of fake online reviews,’’ J. Manage. Inf. Syst., vol. 33, no. 2, pp. 456–481, Apr. 2016.

[4]. A. U. Akram, H. U. Khan, S. Iqbal, T. Iqbal, E. U. Munir, and M. Shafifi, ‘‘Finding rotten eggs: A review spam detection model using diverse feature sets,’’ KSII Trans. Internet Inf. Syst., vol. 12, no. 10, pp. 5120–514Oct. 2018.

[5]. M. Luca and G. Zervas, ‘‘Fake it till you make it: Reputation, competition, and yelp review fraud,’’ Manage. Sci., vol. 62, no. 12, pp. 3412–3427, Dec. 2016. Risk Reduction, ISSN: 2212-4209, vol. 37, July 2019, Article 101176.

[6]. Gonzalo A. Ruz, Pablo A. Henríquez, Aldo Mascareño,“Sentiment analysis of Twitter data during critical events through Bayesian networks classifiers,” Future Generation Computer Systems, vol.106, pp. 92-104, May 2020.

[7]. S.-U. Hassan, N.R. Aljohani, N. Idrees et al., “Predicting literature’s early impact with sentiment analysis in Twitter,” Knowledge-Based Systems (2019), vol. 192, !5 March 2020, Article 105383.

[8]. Kashfia Sailunaz , Reda Alhajj, “Emotion and sentiment analysis from Twitter text,” Journal of Computational Science,vol. 36,September 2019, Article 101003.

[9]. Veny Amilia Fitri, Rachmadita Andreswari, Muhammad Azani Hasibuan, “Sentiment Analysis of social media twitter with case of anti-LGBT campaign in Indonesia using Naïve bayes, Decision treeand Random forest algorithm,”Procedia Computer Science, vol.161 , pp. 765–772,2019.M.A. Hassonah, R. Al-Sayyed, A. Rodan et al., An efficient hybrid filter and evolutionary wrapper approach for sentiment analysis of various topics on Twitter, Knowledge-Based Systems (2019), vol. 192, 15 March 2020, Article 105353.

[10]. Huy Tien Nguyen, Minh Le Nguyen, “An ensemble method with sentiment features and clustering support”, Neuro-computing”,vol. 370, pp. 155-165, 22 December 2019.