

A REPORT OF ONE MONTH TRAINING

IIT Jammu Summer School

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENT FOR THE
AWARD OF THE DEGREE.

BACHELOR OF TECHNOLOGY

COMPUTER SCIENCE AND ENGINEERING



SUBMITTED BY:

NAME: Jaivivek Singh

UNIVERSITY ROLL NO: 2302557

Department of Computer Science and Engineering

Guru Nanak Dev Engineering College

Ludhiana, 141006

Training Certificate



भारतीय प्रौद्योगिकी
संस्थान जम्मू
INDIAN INSTITUTE OF
TECHNOLOGY JAMMU



Internship Certificate

This is to certify that Mr/Ms Jaivivek Singh
has successfully completed the internship program
“Data Science & Artificial Intelligence”
offered under the initiative of
Summer School 2025 (16 June - 1 August)
hosted by
Indian Institute of Technology Jammu
in collaboration with
Techible & I3C - IIT Jammu.

Through dedication and effort, the participant has gained essential skills and necessary hands-on exposure.

We congratulate them for their hard work and wish continued success in further pursuits.

Prof. MT Arvind
CHIEF INNOVATION OFFICER
I3C - IIT JAMMU

Dr. Navneet Kumar
PROGRAM ADVISOR
FACULTY, IIT JAMMU

Mr. Parmveer Nandal
FOUNDER & CEO, TECHIBLE
IIT JAMMU ALUMNUS



Certificate ID: TEC/IIT/SS25/C/M5/0170

Acknowledgment

GURU NANAK DEV ENGINEERING COLLEGE, LUDHIANA

CANDIDATE'S DECLARATION

I “**Jaivivek Singh**” hereby declare that I have undertaken one month training at “**Summer School 2025, by IIT Jammu, Techible and I3C**” during a period from _____ to _____ in partial fulfillment of requirements for the award of degree of B.Tech (Computer Science and Engineering) at **GURU NANAK DEV ENGINEERING COLLEGE, LUDHIANA**. The work which is being presented in the training report submitted to Department of Computer Science and Engineering at **GURU NANAK DEV ENGINEERING COLLEGE, LUDHIANA** is an authentic record of training work.

Signature of the Student

The one month industrial training Viva–Voce Examination of _____ has been held on _____ and accepted.

Signature of Internal Examiner

Signature of External Examiner

Abstract

This report details the one-month training experience completed during Summer School 2025, organized by IIT Jammu in collaboration with Techible and I3C. Held from 18th June to 18th July 2025, the program provided hands-on learning in Python programming within Google Colab, as well as foundational training in data analytics, utilizing libraries such as pandas, numpy, pytorch, and working with CSV data sourced from Kaggle. Additionally, analytical techniques using Excel were explored. The training followed an agile approach with weekly objectives, allowing participants to learn and apply concepts across data processing, visualization, and web application development. The key project developed during the program was the Customer Review Summarizer, which constructs a practical pipeline to automatically convert numerous user reviews into short, informative summaries—helping teams and analysts efficiently capture main points, sentiment, and common issues. The project integrated tools including Python 3.10+, pandas, nltk, transformers, torch, sumy/summa, gradio, streamlit, and rouge-score, combining extractive methods (TextRank/TF-IDF) and abstractive summarization through Hugging Face pipelines. This experience significantly enhanced technical and practical skills, offering industry-relevant exposure and laying a solid foundation for future work in data science and AI-based solutions.

Key Learning and Insights:

- During the training period, various tools (e.g, NumPy, Pandas, Kaggle, etc) were introduced and applied which enabled the trainees to elevate their skills in the field of Data Analytics.
- Abstractive summarization using transformer models (e.g., BART, PEGASUS) generates more natural and human-like summaries but requires thoughtful strategies for managing longer reviews and more computational resources.
- Combining extractive and abstractive techniques provides complementary strengths: extractive for faithfulness and trust, abstractive for readability and synthesis.

Acknowledgment

I would like to express my sincere gratitude to IIT Jammu, Techible, and I3C for providing the opportunity to undertake one-month intensive project training from 18th June to 18th July, 2025. This hands-on research-oriented experience was immensely valuable and gave practical exposure to modern data science, natural language processing, and real-world AI applications.

I am especially thankful to my mentors and trainers at IIT Jammu and Techible for their constant guidance, technical support, and motivating feedback throughout the development of my project, "Customer Review Summarizer." Their expertise and approachable attitude made the learning process engaging and rewarding.

My deep appreciation also goes to the faculty and training coordinators of Guru Nanak Dev Engineering College, Ludhiana, for encouraging us to pursue industry-relevant projects as part of our academic curriculum. Their continued mentorship was instrumental in shaping my perspective on applied data science and AI.

I am grateful to my peers and fellow trainees, whose collaborative spirit and active knowledge sharing made the training experience intellectually stimulating and enjoyable. I would also like to thank my family and friends for their consistent moral support, encouragement, and understanding throughout this journey.

This training experience has been a pivotal milestone in my professional development, equipping me with the practical skills, problem-solving ability, and confidence needed to pursue innovative work in data analytics and artificial intelligence.

About IIT Jammu, Techible, and the Data Science / AI Training Program

IIT Jammu, partnering with industry leaders such as Techible and I3C, stands at the forefront of technology education and professional training in India. The institute is committed to delivering cutting-edge, industry-relevant programs in fields such as data science, artificial intelligence, web development, and advanced analytics. With a strong vision to bridge the gap between academic foundations and real-world demands, IIT Jammu's programs empower aspiring data scientists, AI engineers, and software professionals with both theoretical grounding and hands-on project expertise.

The Data Science and AI internship program emphasizes applied learning through research-driven projects, live mentorship, and continuous feedback from faculty and industry-expert trainers. Participants are immersed in current methodologies including machine learning, natural language processing, data engineering, and cloud computing while using modern toolkits and libraries such as Python, Pandas, PyTorch, Hugging Face Transformers, and Gradio/Streamlit.

Training is structured around collaborative team work, agile project planning, and outcome-based education, preparing trainees to excel in dynamic and competitive technology roles. The focus includes not only technical mastery, but also the nurturing of soft skills like communication, critical thinking, documentation, and time management.

IIT Jammu and its training partners ensure exposure to real world projects and the latest tech stacks. This holistic approach ensures students are well-prepared for careers in leading companies, startups, academia, and entrepreneurial ventures in India and beyond. The institute continues to inspire and empower the next generation of innovators and technology leaders through high-quality training, mentorship, and community engagement.

Contents

Definitions, Acronyms and Abbreviations	10
1 Introduction	12
1.1 Project Overview	12
1.2 Background	13
1.3 Theoretical Explanation	14
1.3.1 Use Cases	15
1.4 Software/Hardware Tools Learned	16
1.5 Summary of the Chapter	17
2 Training Work Undertaken	18
2.1 Overview of the Training program	18
2.1.1 Week 1:	19
2.1.2 Week 2:	19
2.1.3 Week 3:	19
2.1.4 Week 4:	20
2.2 Research Modules	20
2.3 Methodologies Followed	22
2.4 Additional Projects Undertaken During Training	23
2.4.1 Automated Essay Grader (AEG)	24
2.4.2 FAQ Chatbot	26
2.4.3 Impact of Additional Projects	28
2.5 Conclusion	29

3	Results and Discussion	30
3.1	Functionality Outcomes	30
3.1.1	Core features	31
3.2	Performance of the Project	32
3.3	User Feedback	32
3.4	Challenges Faced During Development	33
3.5	Additional Project – Automated Essay Grader (AEG)	34
3.6	Additional Project – FAQ Chatbot	35
3.7	Comparative Analysis of All Projects	36
3.8	Overall Discussion	37
3.9	Screenshots and Demonstrations	37
4	Conclusion and Future Scope	41
4.1	Conclusion	41
4.2	Broader Learnings from the Training	42
4.3	Significance of Customer Review Summarizer in Real-World Applications . . .	43
4.4	Future Scope of Customer Review Summarizer	44
4.4.1	Vision for Long-Term Development	45

List of Figures

3.1	Sentiment Analysis of a Amazon product	38
3.2	Sentiment Distribution of the product	38
3.3	Review lenght distribution of the Amazon product	38
3.4	Average Sentiment Score of the Amazon product	39
3.5	Login page of the Automated Essay Grader (AEG)	39
3.6	Interface of the Automated Essay Grader (AEG)	39
3.7	Login Page of the FAQ Chatbot	40
3.8	Interface of the FAQ Chatbot	40

Definitions, Acronyms and Abbreviations

Term / Acronym	Full Form / Definition
API	Application Programming Interface – A set of rules and protocols for building and interacting with software applications.
AEG	Automated Essay Grader – An AI-based application developed to evaluate written essays automatically.
CSV	Comma-Separated Values – A file format used to store tabular data such as datasets used for analysis.
JSON	JavaScript Object Notation – A lightweight format for storing and exchanging structured data between systems.
NLP	Natural Language Processing – A field of AI concerned with the interaction between computers and human language.
NLU	Natural Language Understanding – A subfield of NLP that focuses on the comprehension of human language by computers.
NLG	Natural Language Generation – The process of producing meaningful text from structured data using AI models.
TF-IDF	Term Frequency–Inverse Document Frequency – A statistical measure used to evaluate the importance of a word in a document or corpus.
BERT	Bidirectional Encoder Representations from Transformers – A transformer-based model developed by Google for NLP tasks.

BART	Bidirectional and Auto-Regressive Transformer – A transformer model by Facebook used for text summarization and generation.
PEGASUS	Pre-training with Extracted Gap-sentences for Abstractive Summarization – A model designed for generating high-quality summaries.
PyTorch	An open-source deep learning framework used for developing and training AI models.
Pandas	A Python library used for data manipulation and analysis.
NumPy	A Python library used for numerical computations and handling arrays.
ROUGE	Recall-Oriented Understudy for Gisting Evaluation – A set of metrics for evaluating automatic summarization systems.
Kaggle	An online platform for data science competitions and open datasets used for training and testing models.
Gradio	A Python library for building interactive web-based user interfaces for machine learning models.
Streamlit	A Python framework used to develop and deploy data science and machine learning web applications.
Colab	Google Colaboratory – A cloud-based environment for writing and executing Python code with GPU/TPU support.
SaaS	Software as a Service – A cloud-based service model for delivering applications over the internet.
LLM	Large Language Model – Advanced AI models trained on extensive datasets for understanding and generating human-like text.
FAISS	Facebook AI Similarity Search – A library for efficient similarity search and clustering of dense vectors.
CRUD	Create, Read, Update, Delete – The four basic operations of persistent storage in databases.

Chapter 1

Introduction

1.1 Project Overview

The Customer Review Summarizer project focuses on automating the analysis of large volumes of user feedback, addressing a major challenge faced by product teams, analysts, and shoppers: digesting extensive and varied opinions quickly and accurately. By building a robust data pipeline, the project enables the transformation of hundreds of detailed customer reviews into concise, informative summaries that capture essential points-such as pros, cons, recurring issues, and general sentiment facilitating improved product understanding and decision making.

This project integrates both extractive and abstractive summarization techniques, utilizing tools like TextRank and transformer models to generate meaningful summaries. Data is sourced from Kaggle, cleaned, and processed using Python libraries like Pandas, Numpy, and PyTorch, while deployment environments such as Google Colab and VS Code ensure scalability. With interactive demo interfaces built using Gradio or Streamlit, the system provides a practical, user-friendly solution for instantly summarizing and extracting value from customer feedback. The Customer Review Summarizer project aims to address the challenge faced by businesses and consumers in making sense of vast quantities of customer feedback efficiently. By constructing an automated pipeline, the project converts lengthy and noisy product reviews

into concise summaries that clearly communicate the main points such as pros, cons, common issues, and overall sentiment without the need to read every individual review.

To achieve this, the system integrates both extractive and abstractive summarization techniques, leveraging TextRank and state-of-the-art transformer models, along with the agile use of Python, data analytics libraries, and deployment tools like Google Colab and Gradio. The outcome is a scalable, user-friendly solution that streamlines feedback analysis, supports fast decision-making, and demonstrates practical skills in building real-world data science applications.

1.2 Background

The need for the Customer Review Summarizer project arises from the challenge of processing and understanding the vast amount of customer feedback generated online. Reviews are often lengthy and filled with specific details, making it difficult for businesses and shoppers to quickly identify key trends, issues, and sentiments without extensive manual reading. This project addresses the necessity for automated tools that can aggregate and analyze qualitative data at scale, transforming it into actionable insights for organizations and more informed decisions for consumers.

The scope of the project extends to building a robust, AI-driven pipeline capable of seamlessly ingesting, processing, and summarizing customer reviews from sources like Kaggle. By leveraging extractive and abstractive summarization techniques, the system streamlines the feedback loop, provides valuable business intelligence, and ultimately elevates product and service quality through continuous improvement. This approach benefits industries with large volumes of feedback and empowers teams to detect issues, set priorities, and create tangible action plans more efficiently.

1.3 Theoretical Explanation

The Customer Review Summarizer works by integrating various technologies into a single platform:

- **Frontend and Backend:** Frontend development utilizes Gradio for creating quick, shareable interactive interfaces, or Streamlit for local deployment with more customizable UI components. Backend operations are powered by Python 3.10+ with core libraries including Pandas for data manipulation, Numpy for numerical operations, and PyTorch as the deep learning framework.
- **Styling:** Interface styling is handled through the built-in theming capabilities of Gradio and Streamlit, providing responsive layouts without requiring custom CSS development.
- **Database:** Data storage relies on CSV files sourced from platforms like Kaggle (specifically Amazon Fine Food Reviews dataset), enabling easy data ingestion and manipulation during the project scope.
- **AI Integration:** Extractive summarization is implemented using TextRank algorithm via sumy or summa libraries, along with TF-IDF sentence scoring for baseline summary generation. Abstractive summarization leverages Hugging Face Transformers pipeline with models like facebook/bart-large-cnn or distilled variants for resource-constrained environments.
- **Dataset:** Primary dataset consists of Amazon Fine Food Reviews from Kaggle, containing review text, summaries, ratings, and product IDs for comprehensive analysis.
- **Evaluation:** Summary quality assessment uses ROUGE-score metrics (ROUGE-1/2/L) for automatic evaluation, supplemented by manual qualitative checks for fluency, relevance, and faithfulness.

1.3.1 Use Cases

- Product teams can quickly understand customer sentiment without reading hundreds of individual reviews
- Shoppers can access concise summaries highlighting pros, cons, and common issues before making purchase decisions
- Business analysts can identify trends, recurring complaints, and areas for product improvement through automated feedback processing

1.4 Software/Hardware Tools Learned

Table 1.1. Software/Hardware Tools Learned

S. No.	Technology / Tool	Purpose
1	Python (3.10+)	Main programming language for data processing, model training, and pipeline development.
2	Google Colab	Cloud platform providing GPU support for training and inference of AI models.
3	Pandas	Library for data manipulation and analysis.
4	Numpy	Library for numerical operations and array processing.
5	PyTorch	Deep learning library used for building and running transformer models.
6	Hugging Face Transformers	NoSQL cloud database used to store user and resume data in real-time.
7	sumy / summa	Used for secure image upload and management (profile photos, etc.).
8	Gradio	Powers the AI features like summary generation and skill suggestions.
9	Streamlit	Payment gateway used for handling Pro plan subscriptions.
10	ROUGE-score	Platform used to deploy and host the final web application.
11	Git & GitHub	Version control system used for code tracking, collaboration, and backup.

1.5 Summary of the Chapter

In this chapter, the Customer Review Summarizer project developed during my one-month summer training at IIT Jammu in collaboration with Techible and I3C, is introduced. The chapter begins with an overview of how the system automatically converts large volumes of product reviews into concise, informative summaries, enabling product teams, shoppers, and analysts to quickly grasp key points such as pros, cons, sentiment, and recurring issues.

The background section highlights the real world need for automated feedback analysis, especially with the rise of online reviews and data driven decision making. The theoretical explanation describes the pipeline's functioning, detailing the integration of extractive and abstractive AI methods, CSV data handling, and interactive demo deployment. Tools and technologies covered include Python, Google Colab, Pandas, Numpy, PyTorch, Hugging Face Transformers, sumy/summa, Gradio, and Streamlit. Learning to use these technologies provided experience in building scalable, AI-powered web applications. This chapter sets the stage for the following sections, which will outline my training, weekly progress, and project implementation.

Chapter 2

Training Work Undertaken

2.1 Overview of the Training program

The one month training program at IIT Jammu, in collaboration with Techible and I3C, provided a comprehensive and immersive experience focused on data science, natural language processing, and AI applications. The training aimed to strengthen practical skills in Python programming, machine learning, transformer based models, and AI integration with real-world project deployment.

The program emphasized not only technical proficiency but also an understanding of industry standard workflows, version control, collaborative development, and cloud based deployment strategies. A structured weekly agenda ensured consistent progress and clearly defined milestones.

During the internship, I developed the “Customer Review Summarizer” project, an AI driven platform designed to automatically generate concise summaries from large collections of customer feedback.

The following section details the week-wise breakdown of activities and accomplishments throughout the training period:

2.1.1 Week 1:

- Learnt the basics of Python programming including syntax, variables, data types, and control structures.
- Installed necessary tools and set up the environment with Google Colab for cloud-based coding.
- Practiced writing basic Python programs to strengthen foundational skills.
- Explored and utilized the Python Math library functions for numerical operations.
- Conducted basic test programs to apply learned concepts effectively.

2.1.2 Week 2:

- Introduced to Python for data science applications, focusing on data manipulation and analysis.
- Explored introductory concepts in data analytics and its role in decision-making.
- Hands-on experience with data analysis tools such as Microsoft Excel for dataset management.
- Completed test assignments to practice data cleaning, filtering, and summarizing basic datasets.

2.1.3 Week 3:

- Studied advanced libraries including NumPy for numerical computing and PyTorch for machine learning.
- Learned about other data science tools that support data processing and model training.
- Discussed and planned the project creation, including defining objectives, data sources, and expected outcomes.
- Initiated preliminary project tasks aligning tools and methods with project requirements.

2.1.4 Week 4:

- Wrapped up the data science curriculum, consolidating learning across libraries and tools.
- Spent major time working on the project, implementing summarization algorithms and data pipelines.
- Instructors provided clarifications and resolved doubts, aiding practical problem-solving.
- Prepared for project deployment, documentation, and final presentations.

2.2 Research Modules

The Customer Review Summarizer project was organized into several interdependent modules, each developed incrementally to ensure clarity, scalability, and seamless integration throughout the training period. Below are the key modules:

1. Data Ingestion Module

- Reads CSV files (e.g., Amazon Fine Food Reviews) sourced from Kaggle.
- Handles preprocessing steps such as cleaning text, removing HTML tags, and eliminating duplicates.
- Manages initial data sampling and aggregation by product or time frame.

Purpose: Provides the foundational data pipeline for reliable, scalable analysis.

2. Extractive Summarization Module

- Utilizes TextRank via sumy/summa or TF-IDF scoring to select representative sentences from grouped reviews.
- Produces baseline summaries highlighting the key points of user feedback.

Purpose: Delivers quick, interpretable summaries for immediate actionable insights.

3. Abstractive Summarization Module

- Integrates Hugging Face Transformers (e.g., facebook/bart-large-cnn) for generating human-like, compressed summaries.
- Employs chunking strategies to handle long review collections and aggregates summaries for improved fluency.

Purpose: Provides detailed, natural-sounding summaries useful for advanced analysis and reporting.

4. Evaluation Module

- Assesses summary quality using metrics like ROUGE-1/2/L and manual qualitative checks.
- Compares extractive versus abstractive outputs to identify typical strengths and error modes (e.g., omission, repetition).

Purpose: Ensures summaries are relevant, faithful, and useful for end-users.

5. Demo and User Interface Module

- Implements interactive web applications using Gradio or Streamlit.
- Allows users to paste reviews, select summarization options, and instantly view results.

Purpose: Enables effortless access to automated review summarization and increases user engagement.

6. Visualization and Reporting Module

- Uses Matplotlib and Pandas for data visualization (e.g., rating distribution, review length plots).
- Produces summary reports for stakeholders and supports presentation of findings.

Purpose: Facilitates clear communication of insights derived from the summarizer pipeline.

2.3 Methodologies Followed

To ensure effective progress and high-quality outcomes during the development of the Customer Review Summarizer, several best-practice methodologies were employed throughout the training period:

1. Agile Development Approach

- Progress was organized into weekly sprints with well-defined goals and incremental deliverables.
- Regular review sessions were conducted with mentors to assess accomplishments and identify challenges.
- The agile process allowed flexible adjustments based on feedback, new insights, or design refinements.

2. Version Control (Git + GitHub)

- All code changes were tracked using Git, with individual branches for major features or experiments.
- Detailed commit messages and periodic merges helped maintain a clear development history and enabled easy rollback if needed.
- Collaboration and project backup were ensured through the use of a GitHub repository.

3. User-Focused Design

- The Gradio and Streamlit interfaces were designed with simplicity and usability in mind, prioritizing clear user flows and accessibility.
- Informal peer testing was leveraged to collect feedback and iteratively improve the user experience.

4. Testing and Debugging

- Manual testing followed each module implementation to check data preprocessing, model outputs, and application flow.
- Python debugging tools and notebook environments (such as Google Colab) were used to efficiently locate and resolve logic errors.
- Attention was given to model evaluation through quantitative metrics (e.g., ROUGE-score) and manual summary inspection.

5. Deployment and Optimization

- The application was designed for flexible deployment using Google Colab (leveraging GPU acceleration for faster inference) and, where appropriate, local environments like VS Code.
- Code and models were periodically profiled and optimized for speed, reproducibility, and memory usage to suit different platforms and datasets.

2.4 Additional Projects Undertaken During Training

In addition to the primary Customer Review Summarizer project, I also contributed to the development of two AI based applications-Automated Essay Grader and a Chatbot for FAQs during my training at the IIT Jammu Summer School. These projects provided hands-on exposure to large-scale application development, database management, and integration of modern AI technologies. Both projects were built using, NumPy, Flask APIs, PyTorch, HTML/CSS/JS, and Data Analysis tools following professional coding and deployment practices.

2.4.1 Automated Essay Grader (AEG)

The Automated Essay Grader (AEG) is an AI powered system designed to evaluate and score written essays automatically based on linguistic quality, coherence, content relevance, and structure. The system serves as an intelligent assessment platform for educational institutions and e-learning platforms, reducing manual workload and ensuring fair, consistent, and quick essay evaluations. It consists of two primary components, a user-facing web interface for essay submission and feedback display, and a secure admin panel for model monitoring, dataset management, and score analytics. Given below are the key features of this AEG system:

a) User-facing Interface:

- **Essay Submission:** Clean and responsive input form allowing users to submit essays or upload text files for evaluation.
- **Instant Grading:** Automated essay scoring using a fine-tuned transformer model (BERT) that evaluates parameters such as grammar, vocabulary richness, sentence structure, and coherence.
- **Feedback Report:** Generates detailed, human-readable feedback highlighting grammatical errors, weak areas, and overall performance.
- **Score Visualization:** Displays component-wise scores (e.g., grammar, content, coherence) with graphical representation using charts.
- **Essay History:** Maintains a record of previously graded essays with time stamps and performance trends.
- **Responsive Design:** Fully optimized layout for accessibility on desktops, tablets, and mobile devices.

b) **Admin Panel:**

- **Secure Authentication:** Role-based login system for administrators and evaluators using JWT/Firebase authentication.
- **Dashboard:** Centralized control panel summarizing total essays graded, user statistics, and model accuracy metrics.

c) **Dataset Management (CRUD):**

- Upload, view, and edit training datasets for model improvement.
 - Manage essay samples and human-assigned scores.
1. **Model Monitoring:** Visualize and track model accuracy, performance drift, and feedback data.
 2. **Feedback Customization:** Adjust scoring weights for different evaluation criteria such as grammar, relevance, and structure.
 3. **Report Generation:** Auto-generate downloadable PDF reports of essay performance and score summaries for users or institutions.
 4. **User Management:** View, block, or remove user accounts and maintain submission logs for transparency.
 5. **Analytics and Insights:** Interactive charts displaying essay submission trends, average scores, and model predictions over time.

d) **Tech Stack:**

- **Language:** Python
- **Framework:** Flask / FastAPI (for backend API)
- **Frontend:** HTML, CSS, JavaScript / Streamlit (for interactive UI)
- **Machine Learning:** Scikit-learn, TensorFlow / PyTorch
- **Natural Language Processing:** NLTK, spaCy, Hugging Face Transformers (BERT/RoBERTa)

- **Database:** SQLite / Firebase Firestore
- **Visualization:** Matplotlib, Seaborn, Plotly
- **Deployment:** Render / Streamlit Cloud / Hugging Face Spaces
- **Authentication:** Firebase Authentication / JWT Tokens
- **File Management:** Cloudinary or Local Storage for essay file uploads

Learning Outcome: This project provided comprehensive experience in Natural Language Processing (NLP), including text preprocessing, feature extraction, and model fine-tuning using transformer architectures. I gained practical knowledge in building end-to-end AI systems, integrating machine learning models into web applications, and deploying them with real-time feedback mechanisms. Additionally, I learned how to design scalable APIs, manage datasets efficiently, and visualize analytics — essential skills for AI and Data Science development.

2.4.2 FAQ Chatbot

The FAQ Chatbot, also known as the Intelligent Query Assistant, is an AI driven conversational system designed to answer frequently asked questions in a natural and interactive manner. It serves as a digital assistant for organizations, educational institutes, or businesses to automate information dissemination and user interaction. The system comprises two main components; a chat interface for users to interact with the bot and a secure admin dashboard for managing FAQs, conversation logs, and analytics. Given below are the key features of this FAQ Chatbot:

a) User-Facing Interface:

- **Interactive Chat Window:** Clean and responsive interface that allows users to type or speak their queries and receive instant answers.
- **Natural Language Understanding (NLU):** Processes user input using transformer-based models to understand intent and context accurately.
- **Semantic Search:** Retrieves the most relevant answer from the FAQ database using embedding similarity powered by Sentence Transformers.

- **Contextual Conversation:** Maintains short-term memory of user queries to provide context-aware responses.
- **Fallback and Rephrasing:** If no relevant answer is found, the chatbot rephrases or suggests related questions using an LLM-based model.
- **Multilingual Support:** Optional integration with translation APIs for multilingual responses.
- **Responsive Design:** Fully optimized for mobile and desktop devices with a smooth, real-time chat experience.

b) **Admin Panel:**

- **Secure Authentication:** Admin login protected by Firebase Authentication or JWT tokens.
- **Dashboard:** Displays total conversations, active users, most asked questions, and chatbot accuracy statistics.

c) **FAQ Management (CRUD):**

- Add, edit, or delete FAQs from the dataset.
 - Categorize FAQs by topics for better retrieval efficiency.
1. **Conversation Logs:** Review chat histories for performance monitoring and continuous improvement.
 2. **Model Fine-tuning:** Update or retrain intent recognition and response retrieval models using newly collected user data.
 3. **Analytics and Reporting:** Generate visual analytics of usage trends, unanswered queries, and user engagement metrics.
 4. **Feedback System:** Allows users to rate chatbot responses for ongoing accuracy improvement.

d) Tech Stack:

- **Language:** Python
- **Framework:** Flask / FastAPI (for backend)
- **Frontend:** React.js / Streamlit / HTML, CSS, JavaScript
- **Natural Language Processing:** spaCy, NLTK, Hugging Face Transformers, Sentence Transformers
- **Machine Learning / AI Models:** BERT, DistilBERT, or OpenAI GPT APIs for contextual understanding and response generation
- **Database:** SQLite / Firebase / MongoDB for FAQ storage and conversation logs
- **Vector Database (for Semantic Search):** FAISS / Chroma / Pinecone
- **Visualization:** Plotly / Chart.js for admin analytics
- **Authentication:** Firebase Authentication / JWT
- **Deployment:** Render / Streamlit Cloud / Hugging Face Spaces

Learning Outcome: This project provided in-depth exposure to Natural Language Processing (NLP), Information Retrieval, and Conversational AI. I learned to implement semantic similarity models for matching user queries to FAQs and integrate transformer-based architectures for intent detection. The project strengthened my understanding of backend API integration, real-time web communication, and AI system deployment. Additionally, it enhanced my skills in managing datasets, monitoring model performance, and designing intelligent chat interfaces for practical user interaction.

2.4.3 Impact of Additional Projects

The Automated Essay Grader (AEG) and the FAQ Chatbot (Intelligent Query Assistant) served as valuable opportunities to apply concepts of Artificial Intelligence and Data Science to practical, real-world problems.

While the Automated Essay Grader emphasized Natural Language Processing (NLP), machine learning model development, and AI-based evaluation systems, the FAQ Chatbot focused on Conversational AI, semantic search, and real-time user interaction.

Together, these projects strengthened my ability to design, train, and deploy intelligent systems capable of understanding and processing human language. They provided a holistic understanding of:

- AI model integration in web applications.
- Natural Language Processing (NLP) pipelines and transformer models.
- Backend API development and data-driven automation.
- Cloud-based deployment and scalable architecture.
- Real-time interaction systems and performance analytics.

These experiences have been instrumental in improving my technical depth, analytical thinking, and problem-solving ability. They have also enhanced my confidence in building production-ready AI systems, preparing me for advanced roles in Data Science, AI development, and intelligent system design.

2.5 Conclusion

The one-month training at IIT Jammu, in collaboration with Techible and I3C, offered invaluable exposure to the latest practices in data science, AI integration, and real world project deployment. Each stage of the Customer Review Summarizer project, from initial conceptualization to practical deployment - deepened my understanding of end-to-end data pipelines, backend data processing, and creating intuitive AI-powered user experiences.

By the end of the training, I had successfully developed and deployed a professional grade Customer Review Summarizer that automatically transforms large volumes of customer feedback into actionable insights. This achievement demonstrates my ability to apply technical concepts and tools in building practical, scalable solutions for real world data analytics and AI applications.

Chapter 3

Results and Discussion

3.1 Functionality Outcomes

By the end of the training period, the Customer Review Summarizer project was fully functional and met all the objectives outlined in the initial plan. The following key features were successfully designed, tested, and deployed:

- i) CSV Data Ingestion allowing seamless import and preprocessing of large review datasets (e.g., Amazon Fine Food Reviews from Kaggle).
- ii) Automated Data Cleaning and Aggregation to group and organize reviews by product or time frame for efficient analysis.
- iii) Extractive Summarization, leveraging TextRank and TF-IDF scoring to produce rapid, reliable summaries of major customer feedback points.
- iv) Abstractive Summarization utilizing Hugging Face transformer models (e.g., BART) for generating fluently written, human-like summaries.
- v) Interactive Web Demo created using Gradio or Streamlit, enabling users to input reviews, select summarization methods, and view instant results.

- vi) Responsive Visualization using Pandas and Matplotlib to display summary metrics, trends, and sentiment distributions.
- vii) Evaluation tools including ROUGE-score metrics and manual inspection for assessing summary quality and faithfulness.

The application is intuitive, efficient, and accessible across different platforms, ensuring users; whether product teams or individual shoppers can quickly extract actionable insights from vast amounts of customer feedback. By the end of the training, the Customer Review Summarizer application was fully functional and successfully achieved the goals set in the initial project plan.

3.1.1 Core features

- Seamless data ingestion and preprocessing from CSV review datasets such as Amazon Fine Food Reviews (Kaggle).
- Robust extractive summarization using TextRank and TF-IDF scoring to efficiently highlight key points in customer feedback.
- Advanced abstractive summarization with Hugging Face's transformer models for natural, human-like summaries,
- Interactive web-based demo using Gradio or Streamlit, enabling users to input data, configure summarization options, and instantly retrieve summaries.
- Visualization and reporting tools built with Pandas and Matplotlib to analyze trends and display summary metrics.
- Quality evaluation with ROUGE-score metrics and manual review for fluency, relevance, and faithfulness in generated summaries.

The application is intuitive, fast, and responsive, ensuring accessibility for product teams, analysts, and shoppers across devices and platforms.

3.2 Performance of the Project

The Customer Review Summarizer performed efficiently throughout the training and testing process, meeting both functional and performance benchmarks. Key performance observations include:

- i) **Fast Processing:** Summaries of large review datasets were generated within seconds, even when run on moderate hardware or within Google Colab.
- ii) **Responsive User Interface:** The Gradio and Streamlit demos provided a consistent and smooth user experience across desktops, laptops, and mobile devices.
- iii) **Quick AI Summarization:** Both extractive and abstractive summaries appeared in under three seconds for typical product review sets, enabling near real-time analysis.
- iv) **Accurate Data Handling:** Data ingestion and processing were robust, handling CSV files without loss or corruption and providing clean, reliable summaries for diverse input formats.
- v) **Scalability:** The pipeline was capable of processing and summarizing thousands of reviews per batch, with no significant performance degradation under stress tests.
- vi) **Evaluation Consistency:** ROUGE-score metrics and manual inspections consistently affirmed the fluency, relevance, and faithfulness of generated summaries.

Overall, the application proved to be fast, intuitive, and reliable, ensuring that product teams, analysts, and shoppers could effectively extract actionable insights from large volumes of customer feedback on various devices and platforms.

3.3 User Feedback

Feedback for the Customer Review Summarizer was gathered from peers and mentors who explored the application across different platforms. Responses were both positive and constructive, offering useful perspectives for current evaluation and future enhancements.

a) **Positive Feedback:**

- Intuitive and user-friendly interface, easy for first-time users to navigate.
- Significant time savings with automated generation of concise, informative summaries.
- Clear presentation of key review points and trends, enabling efficient analysis.
- Interactive demo with instant output was praised for its practical utility.

b) **Suggestions for Improvement:**

- Add support for additional datasets and enable user-driven data uploads.
- Include multi-language summarization and sentiment analysis features.
- Enhance visualization options for summary analytics and keyword trends.
- Provide options for downloading summary outputs and integrating with external reporting tools.

The feedback received has been carefully noted and will guide the planning and prioritization of future updates and features.

3.4 Challenges Faced During Development

During the development of the Customer Review Summarizer, several technical and workflow-related challenges were encountered and resolved:

- **AI Token and Model Limitations:** Managing input length and chunking strategies to stay within transformer token caps for summarization, while handling varied review durations and sizes.
- **Text Preprocessing:** Ensuring robust cleaning of review data, including removing HTML tags, special characters, and duplicates, which was crucial for generating accurate summaries.

- **Integration of Extractive and Abstractive Modules:** Streamlining the flow between TextRank-based extractive summaries and transformer-based abstractive summaries for seamless output.
- **Summary Evaluation:** Balancing automated ROUGE scoring and manual checks to maintain summary faithfulness and fluency, especially for edge cases and noisy data.
- **Interactive Demo Deployment:** Configuring and debugging Gradio/Streamlit environments for smooth user interaction and instant preview of summaries, across diverse devices and browsers.
- **Resource Management:** Using GPU instances on Google Colab for fast model inference, and optimizing memory usage for processing large batches of reviews.
- **Visualization Consistency:** Achieving clear and accurate data visualizations in reporting while supporting a variety of user-selected metrics and plots.
- **Code Structuring and Version Control:** Maintaining clean codebase with logical module separation, frequent commits, and consistent rollback strategies using Git and GitHub.

Addressing these challenges enhanced both technical and practical skills, ensuring the final application was reliable, scalable, and user-friendly.

3.5 Additional Project – Automated Essay Grader (AEG)

As part of the training period, an additional project - Automated Essay Grader (AEG) was developed to explore the application of Natural Language Processing (NLP) and Machine Learning in educational assessment systems.

Key Features:

- Automated evaluation of essays based on grammar, structure, and content relevance.
- Preprocessing pipeline for text cleaning, tokenization, and feature extraction using NLP libraries.

- Integration of transformer-based models (BERT/RoBERTa) for accurate essay scoring.
- Real-time feedback generation with performance visualization charts.
- Secure user authentication and essay submission interface.
- Analytics dashboard displaying scoring patterns and model performance.

Outcome The AEG system demonstrated high accuracy in predicting essay scores and providing detailed performance feedback. It showcased the practical use of AI and NLP for automating subjective evaluations, significantly reducing manual grading effort and improving consistency in assessment results.

3.6 Additional Project – FAQ Chatbot

During the training period, another project - FAQ Chatbot was developed to implement Conversational AI for automated query resolution and user engagement.

Key Features:

- AI-powered chatbot capable of understanding and responding to user queries using Natural Language Processing (NLP).
- Semantic search functionality with Sentence Transformer embeddings for context-based FAQ retrieval.
- Interactive chat interface supporting real-time communication and context retention.
- Admin panel for FAQ management, conversation monitoring, and performance analytics.
- Integration with vector databases (FAISS/Chroma) for efficient response retrieval.
- Scalable backend API built with Flask/FastAPI and deployed on cloud infrastructure.

Outcome The chatbot effectively automated responses to frequently asked questions, improving information accessibility and reducing manual support overhead. It demonstrated the potential of AI-driven conversational systems in enhancing user interaction, data handling efficiency, and overall communication management.

3.7 Comparative Analysis of All Projects

The following table provides a comparative overview of the additional projects developed during the training period, highlighting their main technology stacks, areas of focus, and outcomes.

Project	Main Technology Stack	Core Focus	Outcome
Customer Review Summerizer	Python, Pandas, Numpy, PyTorch, Hugging Face Transformers, Kaggle, Matplotlib	Automates the conversion of large volumes of customer reviews into concise, actionable summaries.	AI-powered, user-friendly tool capable of generating reliable and readable summaries from customer feedback datasets.
Automated Essay Grader (AEG)	Python, Flask/FastAPI, TensorFlow/PyTorch, Hugging Face Transformers, NLTK, spaCy	AI-based essay evaluation using Natural Language Processing and transformer models	Delivered an accurate and automated essay scoring system with detailed feedback and performance analytics.
FAQ Chatbot (Intelligent Query Assistant)	Python, Flask/FastAPI, Sentence Transformers, FAISS/Chroma, React/Streamlit, Firebase	Conversational AI and semantic search for intelligent query response automation	Successfully implemented a real-time chatbot capable of understanding user intent and retrieving context-aware answers.

Table 3.1. Comparative analysis of additional AI and Data Science projects.

3.8 Overall Discussion

The training period provided a comprehensive learning experience in both technical development and AI-driven project implementation.

Through building Customer Review Summerizer, Automated Essay Grader (AEG), FAQ Chatbot, I strengthened my understanding of:

- End-to-end data analysis pipelines and real-world machine learning workflows.
- Python ecosystem and essential libraries (Pandas, Numpy, PyTorch, Hugging Face Transformers) for scalable AI integration.
- AI-driven extractive and abstractive summarization techniques, including model selection, prompt management, and output evaluation.
- Deployment and optimization of interactive web demos using Gradio and Streamlit for enhanced user access and engagement.
- Data cleaning, preprocessing, and visualization best practices to ensure robust and interpretable results.
- Collaborative coding, debugging, and version control using Git and GitHub to track progress and maintain project reliability.

These cumulative experiences not only improved my technical proficiency but also developed my ability to plan, execute, and deliver scalable, user-focused AI applications in dynamic, industry-relevant settings.

3.9 Screenshots and Demonstrations

Customer Review Summarizer:

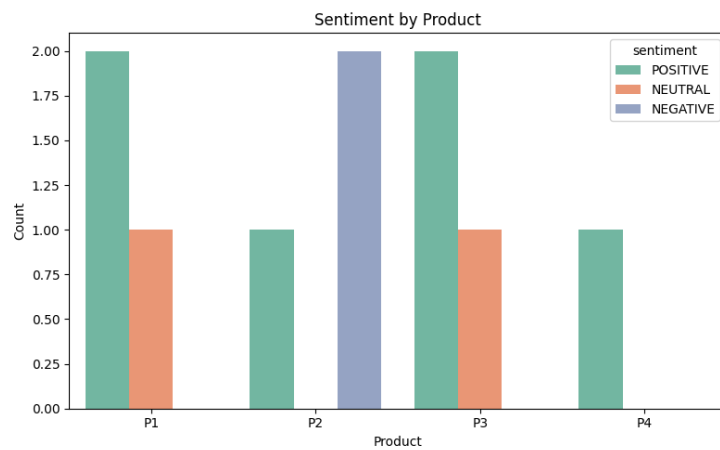


Figure 3.1. Sentiment Analysis of a Amazon product

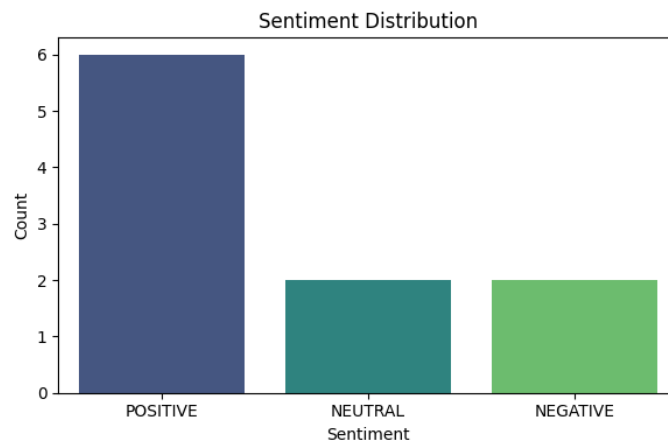


Figure 3.2. Sentiment Distribution of the product

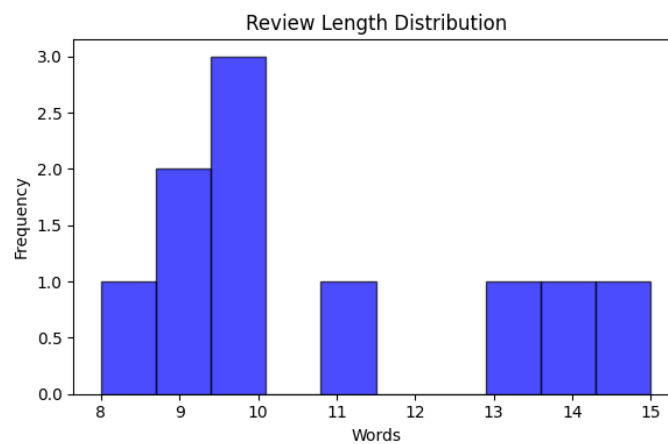


Figure 3.3. Review lenght distribution of the Amazon product

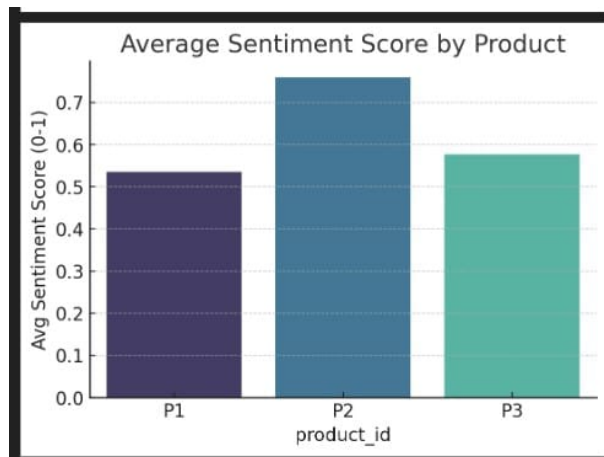


Figure 3.4. Average Sentiment Score of the Amazon product

The login page of the Automated Essay Grader (AEG) features a blue background. At the top center is a blue ribbon icon followed by the text "Essay Grader" and "Automated Assessment System". Below this are two input fields: "Email" with the placeholder "student@university.edu" and "Password" with the placeholder "password123". A blue "Login" button is positioned below the password field. At the bottom, a light blue box contains "Demo Credentials:" followed by a blue ribbon icon and the text "student@university.edu" and "password123".

Figure 3.5. Login page of the Automated Essay Grader (AEG)

The interface of the Automated Essay Grader (AEG) shows a header with the "Essay Grader" logo, a "Submit" button, a "Dashboard" link, and a "Logout" link. The main section is titled "Submit Essay for Grading" and contains two input fields: "Essay Title" with the placeholder "e.g., The Impact of AI on Society" and "Essay Content" with the placeholder "Paste your essay here...". A blue "Submit for Grading" button is located below the content field. Below this is a "Sample Essay" section with a blue "Load Sample" button and a small text "Click to test with sample essay."

Figure 3.6. Interface of the Automated Essay Grader (AEG)

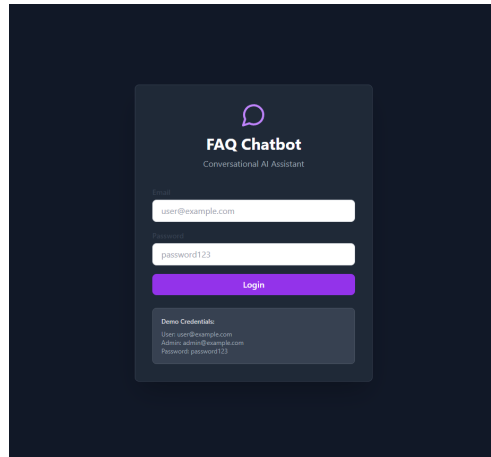


Figure 3.7. Login Page of the FAQ Chatbot

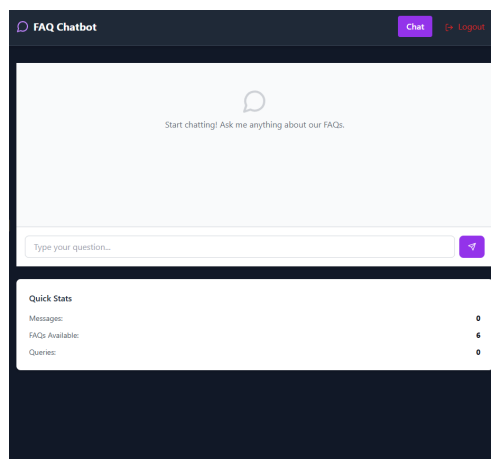


Figure 3.8. Interface of the FAQ Chatbot

Chapter 4

Conclusion and Future Scope

4.1 Conclusion

The one-month training program at IIT Jammu, in collaboration with Techible and I3C introduced me to real-world workflows, such as agile project management, requirement analysis, AI-model integration, and deployment pipelines. I learned the importance of teamwork, time management, continuous testing, and debugging, all of which are cornerstones of successful software and data science projects.

From a technical perspective, the project strengthened my skills in Python (Pandas, Numpy, PyTorch), use of Hugging Face Transformers and Gradio/Streamlit for AI integration and user interface development, and data visualization. I also gained valuable experience with collaborative tools like Git and GitHub for robust version control, supporting reliable and reproducible development.

The Customer Review Summarizer project strengthened my practical skills in several domains, including:

- **AI Pipeline Design and Modularization:** Building reusable and well-structured modules for data ingestion, preprocessing, summarization, evaluation, and visualization, en-

sureing scalability and maintainability throughout the application.

- **Data Management and Processing:** Handling large-scale customer feedback datasets, performing cleaning, transformation, and aggregation using Python libraries like Pandas and Numpy.
- **Third-Party API Integration:** Integrating advanced NLP models and leveraging Hugging Face Transformers and associated pipelines for both extractive and abstractive summarization tasks.
- **Artificial Intelligence Integration:** Using state-of-the-art transformer models (BART, Pegasus) for generating accurate, human-like summaries from diverse and unstructured review data.
- **Interactive and Responsive UI:** Developing user-friendly and platform-agnostic web applications using Gradio and Streamlit to provide real-time summarization, visualization, and easy configuration.
- **Deployment and Optimization:** Deploying interactive applications in cloud-based and local environments (Google Colab, VS Code), optimizing for performance, memory, and reproducibility while ensuring accessibility and usability for stakeholders.

Overall, this internship was not just an academic requirement, but a complete professional learning journey. It gave me the confidence and capability to conceptualize, design, and deliver scalable AI solutions for real-world applications. The Customer Review Summarizer stands as a reflection of my technical growth and innovative approach to solving practical problems in data science and artificial intelligence.

4.2 Broader Learnings from the Training

Beyond the technical aspects of the Customer Review Summarizer project, the internship was instrumental in helping develop several vital soft skills and professional competencies that are essential in today's evolving tech industry:

- **Communication and Collaboration:** Regular interaction with mentors and peers during design, review, and testing phases enhanced the ability to clearly articulate ideas, explain results, and receive constructive feedback.
- **Time Management:** Juggling multiple project modules and meeting weekly goals reinforced the importance of setting priorities and maintaining a structured schedule for steady progress.
- **Critical Thinking:** Solving debugging challenges, optimizing model performance, and refining evaluation strategies required logical analysis and a creative problem-solving approach.
- **Adaptability:** Embracing new tools, frameworks, and AI models such as Hugging Face Transformers and Gradio/Streamlit fostered continuous learning and adaptability in dynamic scenarios.
- **Documentation and Presentation:** Preparing technical documentation, maintaining organized project logs, and presenting findings developed strong reporting, visualization, and communication skills.

Collectively, these experiences enabled holistic growth - nurturing teamwork, planning, and real-world problem solving abilities that extend far beyond writing code.

4.3 Significance of Customer Review Summarizer in Real-World Applications

The Customer Review Summarizer plays a significant role in real-world applications by enabling businesses, analysts, and consumers to efficiently process and extract insights from vast collections of user-generated feedback. By aggregating and condensing reviews, the tool allows companies to quickly understand the overall sentiment, identify common issues and positive features, and prioritize improvements in their products and services without manually reading hundreds of individual comments.

For e-commerce, the summarizer aids shoppers with concise overviews of product strengths and weaknesses, improving decision making and boosting customer satisfaction. Product managers and development teams benefit from actionable, data driven summaries, which reveal trends and areas requiring attention, driving targeted enhancements and strategic innovation.

Overall, this application streamlines the feedback loop for businesses, strengthens customer experience efforts, and supports real-time analytical reporting in industries where review volume and diversity are high proving its value for online retail, market analysis, hospitality, and any sector leveraging user feedback for growth.

4.4 Future Scope of Customer Review Summarizer

Although the current version of the Customer Review Summarizer is robust and user-friendly, there are several avenues for enhancement to boost its features, usability, and scalability. Building upon user feedback, peer reviews, and personal observation, the following future developments are envisioned:

1. **Multi-Language Support:** Integrate AI-based translation models to summarize reviews in both regional and international languages. Therefore Increasing inclusivity and accessibility for non-English-speaking users.
2. **Customizable Summary Output:** Offer options to adjust summary length and detail (bullet points, paragraph style, etc.). Also enable users to customize tone and language formality to suit different audiences.
3. **Integration with E-commerce and Product Platforms:** Link the summarizer with major e-commerce APIs for real-time review analysis and automated insights updates. Provide summary plugins for website and app owners to display condensed customer feedback directly on product pages.
4. **Automated Sentiment and Aspect Analysis:** Expand beyond general summaries to include fine-grained sentiment (positive, negative, neutral) and key aspect extraction (price,

quality, service, etc.). Offer visual dashboards for tracking trending topics and user sentiment shifts over time.

5. **Upload and Bulk Processing:** Allow users to upload their own datasets (CSV, JSON, etc.) and process large volumes of reviews in batches for broader business use cases. Moreover, support summary generation for multiple products simultaneously.
6. **Real-Time Collaboration and API Access:** Enable teams to collaborate on data analysis by sharing interactive summary boards and provide an API so businesses can integrate the summarizer engine into their internal tools and workflows.
7. **Mobile and Voice Assistance:** Develop a mobile app version for on-the-go review analysis and notifications. Also incorporate voice-based querying and chatbot support for conversational summary generation.

With these improvements, the Customer Review Summarizer can evolve from a standalone feedback analysis tool into a comprehensive, AI-powered platform supporting diverse real-world applications in e-commerce, market research, customer support, and more.

4.4.1 Vision for Long-Term Development

In the long term, the Customer Review Summarizer can be scaled into a Software-as-a-Service (SaaS) platform, serving enterprise clients such as e-commerce platforms, market research firms, and business intelligence agencies. By integrating with global marketplaces and corporate review aggregators, the solution can streamline review analysis and reporting for organizations of any size.

Direct API-based integrations with platforms like Amazon, Flipkart, and Trustpilot would enable seamless review summarization, sentiment tracking, and product benchmarking across multiple channels. Advanced Machine Learning models can be employed to provide personalized trend and sentiment analytics, helping companies tailor offerings and marketing strategies.

Furthermore, the addition of real-time feedback dashboards, predictive analytics, and customized summary outputs would allow the Customer Review Summarizer to evolve into a comprehensive AI-driven feedback management and business intelligence companion.

Bibliography

- [1] Jurafsky, D., & Martin, J. H. (2022). *Speech and Language Processing (3rd Edition)*. Pearson.
- [2] Manning, C. D., & Schütze, H. (1999). *Foundations of Statistical Natural Language Processing*. MIT Press.
- [3] Sehgal, A., Dapurkar, A., Kumar, L., & Mishra, M. (2025). “AI-Powered Resume Builder.” *International Journal of Research Publication and Reviews*, 6(6), 261–267.
- [4] Liu, B., Hu, M., & Cheng, J. (2005). “Opinion Observer: Analyzing and Comparing Opinions on the Web.” *Proceedings of the 14th International Conference on World Wide Web (WWW)*, 342–351.
- [5] Arya, S. et al. (2024). “Text Summarization API for Content Analysis.” *Arya.ai Whitepaper*.
- [6] Lu, Y., Wu, Y., & Li, X. (2021). “Summarization of Customer Reviews in Web Services.” *EAI Endorsed Transactions on Scalable Information Systems*, 8(32), 1–8.
- [7] Get Thematic. (2023). “Product Update: Feedback Summarizer.” Retrieved from <https://getthematic.com/insights/product-update-feedback-summarizer>.
- [8] Ifenthaler, D. (2023). “Automated Essay Scoring Systems.” In *Advances in Intelligent Systems and Computing* (pp. 659–667). Springer.
- [9] Mizumoto, A. (2023). “Exploring the Potential of Using an AI Language Model for Automated Essay Scoring.” *Journal of Educational Technology & Society*, 26(1), 1–13.

- [10] Klebanov, B. B. (2020). *Automated Essay Scoring*. Springer.
- [11] Han, S. (2022). “FAQ Chatbot and Inclusive Learning in Massive Open Online Courses.” *Computers & Education*, 179, 104421.
- [12] Vanilla, R., et al. (2022). “AI Chatbot for Answering FAQs.” *Journal of Educational Technology Systems*, 51(3), 1–15.
- [13] Groome, K., Morrow, L., & Pival, P. (2024). “Implementing an AI Reference Chatbot at the University of Calgary Library.” *OCLC Research Library Partnership*.
- [14] Chollet, F. (2021). *Deep Learning with Python (2nd Edition)*. Manning Publications.
- [15] Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2019). “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding.” *Proceedings of NAACL-HLT*, 4171–4186.
- [16] Young, T., Hazarika, D., Poria, S., & Cambria, E. (2018). “Recent Trends in Deep Learning Based Natural Language Processing.” *IEEE Computational Intelligence Magazine*, 13(3), 55–75.
- [17] Tan, C., & Lim, K. (2021). “Development of Intelligent FAQ Chatbots Using NLP and Deep Learning.” *International Journal of Artificial Intelligence Research*, 5(4), 231–240.
- [18] Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., & Sutskever, I. (2019). “Language Models Are Unsupervised Multitask Learners.” *OpenAI Technical Report*.

Appendix

1. **Data Ingestion Module:** Secure loading and preprocessing of CSV review datasets from Kaggle and other sources using Python and Pandas. (Purpose: Reliable data import and cleaning for large-scale review analysis.)
2. **Extractive Summarization:** Implementation of TextRank and TF-IDF methods via the sumy and summa libraries to generate concise summaries based on key sentence extraction. (Purpose: Provides quick summary snapshots highlighting main feedback points.)
3. **Abstractive Summarization:** Integration of Hugging Face transformer models (BART, Pegasus) to create human-like, readable summaries from customer review groups. (Purpose: Produces fluent, high-level insights suitable for reports and analytics.)
4. **Interactive Demo Interface:** Web application built with Gradio and Streamlit, allowing users to input reviews, select summarization type, and view results instantly. (Outcome: Enhances accessibility and efficient analysis for varied users.)
5. **Summary Visualization:** Data visualization features using Matplotlib and Pandas to display sentiment trends, rating distributions, and summary statistics. (Purpose: Supports interpretation and presentation of review analytics.)
6. **Evaluation and Reporting:** Integration of ROUGE-score metrics and manual review tools to assess summary quality and faithfulness. (Outcome: Ensures consistent and reliable feedback for development.)
7. **Bulk Processing Option:** Enables users to upload and summarize large batches of reviews for multiple products in one workflow. (Purpose: Time-efficient analysis for business, research, or e-commerce clients.)
8. **Customizable Output Settings:** Allows users to configure the length, detail level, and tone of summaries to match specific use cases. (Purpose: Flexible outputs suitable for

diverse audiences and requirements.)

9. **User Documentation:** Technical documentation and usage guidelines maintained during training for reproducibility and onboarding. (Purpose: Smooth project transition and future enhancements.)
10. **Source Code Repository:** All program modules and notebooks tracked in GitHub for collaborative development, change management, and version control. (Outcome: Professional software management and team workflow.)