Internship Dissertation Submitted for the partial fulfillment of the **M.Sc. (Integrated) AIML**degree to the Department of AI&ML and Data Science

# Internship Dissertation

## Semester-X

# *OPENAI QUERYCONNECT USING LANGCHAIN*

*SUBMITTED TO* *UNDER THE GUIDANCE OF*

SANSKAR TECHNOLAB


*BY* *RAJVI SAVLIYA*

,

*ZEEL RATHI* *TEAM LEAD*

*MSC (INT) AIML*

Department of AIML & Data Science

School of Emerging Science and Technology

Gujarat University


**April, 2024**

APRIL 2024

# DECLARATION

This is to certify that the research work reported in this dissertation entitled "OpenAI Querry Connect with Langchain" for the partial fulfilment of M.Sc. as a part of M.Sc. (Integrated) in Artificial Intelligence and Machine Learningdegree is the result of investigation done by myself.

Place: Ahmedabad

Name of Student
ZEEL RATHI

Date: 30-04-2024

# ACKNOWLEGEMENT

I would like to express my sincere gratitude to all those who have supported and assisted me throughout the course of my research and writing of this thesis.

I would like to express my profound gratitude to Dr.Ravi Gor sir, Head of department of AIML/Data science for all his support. He has always supported me and encouraged me to overcome many difficulties. I am extremely grateful to him for his valuable suggestions and guidance that he has provided me during my research work.

I would like to thank my supervisor Rajvi Savaliya, Akash Singh for her invaluable guidance and supportthroughout this project. Their expertise, encouragement, and constructive criticism have been instrumental in shaping my research and helping me to refine my ideas.

I would also like to thank the staff for their feedback and support in the development of my research. Their insights and suggestions have been incredibly helpful in shaping my argumentsand strengthening my analysis.

Furthermore, I would like to thank my friends for their support, encouragement, and assistance throughout this project. Their discussions, feedback, and encouragement have been invaluable inhelping me to stay motivated and focused throughout the research and writing process.

Finally, I would like to express my gratitude to my family for their unwavering support and encouragement. Their love, patience, and understanding have been the foundation of my academic journey, and I am incredibly grateful for their continued support.

~ Zeel Rathi

Index

# Chapter 1

# Abstract & Key Words

# Abstract

In the era of burgeoning data volumes, effective communication with and comprehension of data is paramount. "QueryConnect" emerges as a novel solution, integrating the prowess of OpenAI and Langchain technologies to bridge the gap between data and human understanding. Leveraging OpenAI's advanced language models, the platform facilitates natural language interactions with CSV datasets, enabling users to engage in dialogue, summarize data insights, and analyze content seamlessly. Langchain, with its robust text processing capabilities, augments the platform's functionalities, ensuring efficient data handling and comprehension. "QueryConnect" empowers users to navigate through complex datasets effortlessly, unlocking insights and fostering meaningful conversations with their data. This abstract encapsulates the essence of "QueryConnect," an innovative tool poised to revolutionize the intersection of data analytics and natural language processing.



**Keywords**: OpenAI, Langchain, Natural Language Processing, Data Analysis, CSV Datasets, Conversational Interface, Summarization, Data Comprehension, Advanced Language Models

# Chapter 2

# Introduction

In today's data-centric world, the ability to extract meaningful insights from vast datasets is crucial for informed decision-making and innovation. With the advent of advanced natural language processing (NLP) technologies and large language models (LLMs) such as OpenAI's GPT series, there has been a paradigm shift in how we interact with and derive value from data. The project "QueryConnect: Bridging Data and Conversations with OpenAI and Langchain" embodies this shift, aiming to empower users to seamlessly bridge the gap between their data and conversational interfaces.

Background:

The rapid advancement of NLP technologies has enabled machines to understand and generate human-like text with unprecedented accuracy and fluency. OpenAI's GPT series stands at the forefront of this revolution, showcasing the immense potential of LLMs in various domains, including data analysis. Concurrently, Langchain, a Python module designed to facilitate the utilization of LLMs, provides a standardized interface for accessing and leveraging these powerful models in data analytics tasks.

Objective:

The primary objective of the "QueryConnect" project is to democratize access to advanced data analytics capabilities by enabling users to engage in natural language conversations with their data. By integrating OpenAI's LLMs with Langchain, the project seeks to empower users across domains to interact with their datasets intuitively, facilitating exploration, analysis, and interpretation of complex data.

Approach:

The project adopts a comprehensive approach, leveraging the

complementary strengths of OpenAI and Langchain to achieve its objectives. OpenAI's GPT series serves as the backbone for natural language understanding and generation, enabling users to pose queries and requests in plain English. Langchain provides the necessary infrastructure and tools to integrate LLMs seamlessly into data analysis workflows, ensuring efficient processing and interpretation of data.

Key Features:

"QueryConnect" offers a range of features designed to enhance the user experience and facilitate meaningful interactions with data. These include:

- Natural language querying: Users can interact with their datasets using conversational interfaces, posing questions and requests in everyday language.
- Summarization: The platform can generate concise summaries of dataset contents, enabling users to quickly grasp key insights and trends.
- Analytical capabilities: "QueryConnect" supports various analytical tasks, including trend analysis, anomaly detection, and correlation identification, empowering users to derive actionable insights from their data.

Expected Outcomes:

By bridging the gap between data and conversations, "QueryConnect" aims to democratize access to data analytics and empower users to make informed decisions based on data-driven insights. The platform's intuitive interface and powerful analytical capabilities are expected to streamline the process of data analysis, enabling users to unlock the full potential of their datasets.

# Chapter 3

# Basic Terminology

There are Several key terminologies are utilized, reflecting the function nalities and components of the application:

1. Streamlit:

- Streamlit is a Python library used for creating interactive web applications.
- It simplifies the process of building data-focused web apps by allowing developers to write Python scripts to create user interfaces.
- With Streamlit, developers can easily integrate data visualization, input widgets, and other interactive components into their applications.
- In the "QueryConnect" project, Streamlit is utilized to create the user interface for interacting with CSV data.

2. Langchain:

- Langchain is a Python module designed to facilitate the use of Large Language Models (LLMs) in various applications.
- It provides a standardized interface for accessing and utilizing LLMs such as GPT (Generative Pre-trained Transformer) models.
- Langchain simplifies the process of integrating LLMs into applications by abstracting away complexities associated with model initialization, inference, and management.

3. CSVLoader:

- CSVLoader is a component responsible for loading CSV (Comma-Separated Values) files into the application.
- CSV files are commonly used to store tabular data, where each row represents a record and columns represent

attributes.

- The CSVLoader in the "QueryConnect" project ensures that CSV data can be efficiently loaded and processed for further analysis or interaction.

4. ChatOpenAI:

- ChatOpenAI is a class representing an OpenAI language model tailored for conversational interactions.
- It allows users to engage in natural language conversations with the application, posing questions or requests and receiving responses.
- The ChatOpenAI class encapsulates functionalities for interacting with OpenAI's language models, such as generating text based on user inputs.

5. Recursive Character Text Splitter:

- Recursive Character Text Splitter is a text processing component used to split documents into smaller chunks based on character-level recursion.
- This splitting mechanism enables the processing of large textual datasets by breaking them down into manageable segments.
- The Recursive Character Text Splitter enhances the efficiency and scalability of text processing tasks within the "QueryConnect" application.

6. FAISS (Facebook AI Similarity Search):

- FAISS is a vector similarity search library developed by Facebook AI Research.
- It provides efficient algorithms for searching and retrieving similar documents or data points based on their embeddings.

- In the context of the "QueryConnect" project, FAISS is utilized to perform similarity searches on text embeddings for tasks such as document retrieval or content recommendation.

7. Runnable With Message History:

- Runnable With MessageHistory is a component that incorporates functionality for managing message history within the application.
- It allows the application to maintain context and continuity in conversations by storing and accessing previous messages exchanged between the user and the system.
- This component ensures seamless interactions and coherent responses during conversational interactions within the "QueryConnect" application.

8. Summarization Chain:

- Summarization Chain is a processing chain designed to summarize textual data, condensing it into concise and informative summaries.
- It leverages OpenAI language models to generate summaries of documents or datasets, enabling users to extract key insights and trends from large volumes of text.
- The Summarization Chain enhances the utility of the "QueryConnect" application by providing a summarization feature for efficiently digesting textual information.

9. Pandas:

- Pandas is a popular Python library for data manipulation and analysis.
- It provides data structures and functions for efficiently handling structured data, particularly tabular data such as

CSV files.

- Pandas simplifies tasks such as data loading, cleaning, filtering, and analysis, making it a versatile tool for working with datasets in data science and machine learning projects.

10. API Key:

- An API Key is a unique authentication key required to access certain external services or APIs securely.
- It serves as a form of authorization, allowing developers to access and interact with external resources such as the OpenAI API.
- In the "QueryConnect" project, an API Key is necessary for authenticating and accessing the OpenAI API, enabling functionalities such as natural language processing and generation.

These elaborations provide a deeper understanding of the fundamental components and functionalities employed in the "QueryConnect" project, highlighting their roles and significance within the application architecture.

# Chapter 4

# Literature Survey

A brief literature review for this project, focusing on related works and technologies used:

1. Streamlit and Interactive Data Analysis Tools:
    - Streamlit has gained popularity as a framework for building interactive web applications for data science and machine learning tasks.
    - It simplifies the process of creating interactive dashboards and applications, allowing developers to focus on data analysis and visualization.
    - Several studies and tutorials explore the use of Streamlit for various applications, including data exploration, machine learning model deployment, and interactive data analysis.

2. Conversational AI and Natural Language Processing (NLP):
    - Conversational AI and NLP technologies play a crucial role in enabling natural language interactions with computer systems.
    - Research in this area focuses on developing AI models capable of understanding and generating human-like responses in natural language.
    - Chatbot systems powered by advanced NLP models such as GPT (Generative Pre-trained Transformer) have been extensively studied and deployed in various domains, including customer service, education, and entertainment.

3. Text Summarization Techniques:
    - Text summarization is a well-studied task in NLP, aiming to generate concise and informative summaries of longer texts.
    - Traditional approaches include extractive summarization, where sentences or phrases are selected from the

original text, and abstractive summarization, where new sentences are generated to capture the key information.
- Recent advancements in deep learning, particularly with transformer-based models like GPT, have led to significant improvements in abstractive summarization tasks.

4. Data Analysis and Visualization with Python Libraries:
    - Python libraries such as Pandas, NumPy, and Matplotlib are widely used for data analysis and visualization tasks.
    - Pandas provides powerful tools for manipulating structured data, making it a popular choice for working with CSV files and other tabular data formats.
    - Matplotlib and Seaborn offer robust visualization capabilities, allowing users to create a wide range of plots and charts to explore and communicate insights from the data.

5. OpenAI API and Language Models:
    - The OpenAI API provides access to state-of-the-art language models like GPT-3 and GPT-4, which have demonstrated impressive capabilities in natural language understanding and generation.
    - Researchers and developers have explored various applications of these models, ranging from text generation and summarization to language translation and question answering.
    - Integrating the OpenAI API into applications enables developers to leverage the power of these language models for a wide range of NLP tasks.

By combining these technologies and methodologies, the "QUERY CONNECT" project aims to provide a user-friendly platform for

interacting with CSV files, enabling conversational querying, summarization, and analysis of tabular data.and financially sustainable digital communities, transcending traditional barriers in various sectors.

# Chapter 5

# Methodology

# Technology Stack

Technology used in this project:

1. Python:

- Purpose: Python is a versatile programming language known for its simplicity and readability.
- Usage: It serves as the primary language for developing the backend logic and scripting various components of the application.
- Features: Python offers extensive libraries and frameworks for data processing, natural language processing, web development, and more, making it well-suited for building complex applications like QueryConnect.

2. Streamlit:

- Purpose: Streamlit is a Python library used for building interactive web applications.
- Usage: It provides an easy-to-use interface for creating user-friendly web interfaces with minimal code, making it suitable for showcasing data analysis and visualization results.
- Features: Streamlit allows developers to write Python scripts to create web apps without requiring knowledge of web development languages like HTML, CSS, or JavaScript.

3. Langchain:

- Purpose: Langchain is a Python module used for integrating Large Language Models (LLMs) into applications.

- Usage: It provides a standardized interface for accessing and utilizing LLMs, such as GPT models, simplifying the process of incorporating natural language understanding and generation capabilities.
- Features: Langchain offers components for text processing, embedding generation, similarity search, and document summarization, facilitating advanced natural language processing tasks.

4. FAISS (Facebook AI Similarity Search):

- Purpose: FAISS is a library developed by Facebook AI Research for efficient similarity search and clustering of large-scale datasets.
- Usage: It is used in the project for vector similarity search tasks, such as retrieving similar documents based on their embeddings generated from textual data.
- Features: FAISS provides algorithms optimized for searching large collections of vectors, making it suitable for tasks like information retrieval and recommendation systems.

5. OpenAI API:

- Purpose: The OpenAI API provides access to pre-trained GPT models developed by OpenAI.
- Usage: It is leveraged in the project to access state-of-the-art language models for tasks such as conversational interactions, document summarization, and data analysis.
- Features: The OpenAI API offers various GPT models with different capabilities, allowing developers to choose models based on their specific requirements for performance and functionality.

6. Pandas:

- Purpose: Pandas is a Python library used for data manipulation and analysis, particularly for structured tabular data.
- Usage: It is employed in the project for tasks such as loading CSV files, preprocessing data, and performing data analysis operations before presenting insights to the user.
- Features: Pandas provides data structures like DataFrame and Series, along with functions for indexing, filtering, aggregating, and visualizing data, making it a powerful tool for data exploration and manipulation.

7. Asyncio:

- Purpose: Asyncio is a Python library for asynchronous programming.
- Usage: It is utilized in the project to manage asynchronous interactions with the OpenAI API and other asynchronous tasks, enhancing the responsiveness and scalability of the application.
- Features: Asyncio enables non-blocking I/O operations and parallel task execution, allowing the application to handle multiple concurrent tasks efficiently without blocking the main thread.

8. Tempfile:

- Purpose: The Tempfile module in Python is used to create temporary files and directories.
- Usage: In the project, Tempfile is employed for handling uploaded CSV files. When a user uploads a CSV file, Tempfile creates a temporary file to store the uploaded data temporarily during processing.
- Features: Tempfile ensures efficient and secure management of temporary data generated during the

processing of user-uploaded files. Temporary files are automatically deleted when they are closed or when the program exits, preventing clutter and potential security risks.

9. RecursiveCharacterTextSplitter:

- Purpose: RecursiveCharacterTextSplitter is a component provided by the Langchain module for splitting text documents into smaller chunks.
- Usage: It is used in the project to split CSV data into manageable chunks for processing and analysis.
- Features: RecursiveCharacterTextSplitter allows developers to specify parameters such as chunk size and overlap to customize the splitting process according to the requirements of the application. It helps optimize resource usage and improve the efficiency of text processing tasks.

10. CSVLoader:

- Purpose: CSVLoader is a component provided by the Langchain module for loading CSV files into memory.
- Usage: It is utilized in the project to load CSV data from uploaded files or local storage for further processing and analysis.
- Features: CSVLoader handles various aspects of CSV file loading, such as specifying file paths, handling different encodings, and parsing CSV data into structured formats like DataFrames. It ensures compatibility with different CSV file formats and encodings, enhancing the robustness of the application.

11. ChatPromptTemplate and MessagesPlaceholder:

- Purpose: ChatPromptTemplate and MessagesPlaceholder are components provided by the Langchain module for constructing chat prompts and placeholders within conversational interactions.
- Usage: They are used in the project to define templates for chat interactions between users and the application, incorporating placeholders for user input, historical messages, and contextual information.
- Features: ChatPromptTemplate and MessagesPlaceholder facilitate the creation of interactive conversational interfaces, allowing developers to design flexible and customizable chat experiences. They enable dynamic generation of prompts and responses based on user input and system context, enhancing the naturalness and coherence of conversations.

12. RunnableWithMessageHistory:

- Purpose: RunnableWithMessageHistory is a component provided by the Langchain module for executing chat runnables with message history management.
- Usage: It is utilized in the project to execute chat interactions between users and the application while maintaining a history of messages exchanged during the conversation.
- Features: RunnableWithMessageHistory orchestrates the flow of messages between users and the application, managing message history storage and retrieval. It enables seamless integration of chatbot functionality with persistent message tracking, allowing for context-aware interactions and continuity in conversations.

Overall, these technologies work together to enable the QueryConnect application to interact with CSV data through natural language conversations, summarize data for insights, and analyze

datasets effectively within a web-based environment.

## Methodology:

The methodology of the "QueryConnect" project involves several key steps, including data preprocessing, model selection, implementation of chatbot functionalities, and integration with Streamlit for web application development. Below is a detailed elaboration of the methodology:

1. Requirement Analysis:

- The project starts with a thorough analysis of the requirements, including the need for interacting, analyzing, and summarizing CSV files using natural language processing (NLP) techniques.

2. Selection of Tools and Libraries:

- The project utilizes various tools and libraries to implement the desired functionalities:
- Streamlit: A Python library for building interactive web applications.
- Langchain: A modular NLP framework for building conversational agents and text processing pipelines.
- OpenAI GPT Models: Pre-trained language models provided by OpenAI for natural language understanding and generation tasks.
- FAISS: A library developed by Facebook AI Research for efficient similarity search and clustering of large-scale datasets.

3. Data Preprocessing:

- Before processing CSV files, text preprocessing steps may be applied, such as removing special characters, lowercasing text, and handling missing values.
- CSV files are loaded using the Langchain CSVLoader component, which handles different file encodings and formats.

4. Implementation of Chat Functionality:

- The application offers a chat interface where users can interact with CSV data using natural language.
- Upon uploading a CSV file, the text data is split into smaller chunks using the RecursiveCharacterTextSplitter for efficient processing.
- The FAISS library is used to create a vector store from the text chunks, enabling fast similarity search operations.
- The ChatOpenAI model from Langchain is employed to provide conversational responses based on user queries. The model is initialized with parameters such as temperature for controlling response creativity.
- Messages exchanged during the conversation are stored using ChatMessageHistory to maintain conversation context and history.

5. Implementation of Summarization Functionality:

- Users can upload a CSV file to generate a summary of its content.
- The text data from the CSV file is split into chunks, and a summarization chain is initialized using the ChatOpenAI model.
- The summarization chain processes the text chunks to generate a concise summary of the content, leveraging the capabilities of the selected language model.

6. Implementation of Analysis Functionality:

- Users can upload a CSV file to perform analysis tasks, such as identifying column names or extracting insights.
- The CSV data is loaded into a pandas DataFrame, and a conversational agent is created using the create_pandas_dataframe_agent function from Langchain.
- The agent interacts with the user to understand their query and provide relevant insights or information based on the CSV data.

7. Integration with Streamlit:

- The application is built using Streamlit, allowing for easy creation of interactive web interfaces directly from Python scripts.
- Streamlit components such as st.write, st.sidebar, and st.file_uploader are used to design the user interface and handle user interactions.
- The application layout is configured using st.set_page_config to define the title and layout of the web page.

8. Deployment:

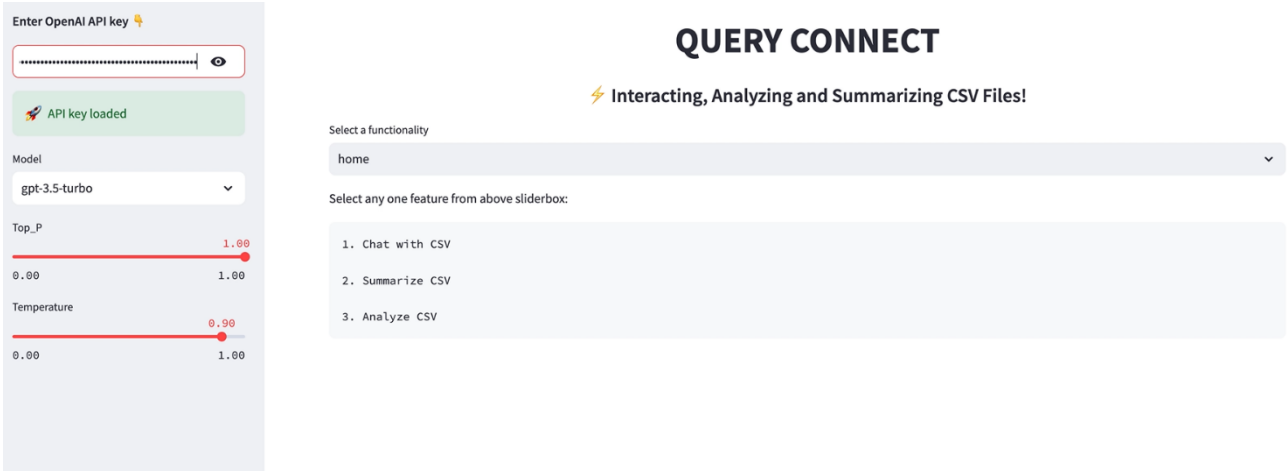
- Once the application development is complete, it can be deployed to a web server or cloud platform for public access.
- Streamlit provides built-in support for deployment to platforms like Heroku, allowing for seamless deployment and hosting of the application.

By following this methodology, the "QueryConnect" project achieves its goal of enabling users to interact with, analyze, and summarize CSV files using natural language, providing a user-friendly and intuitive interface for data exploration and processing.

# Chapter 6

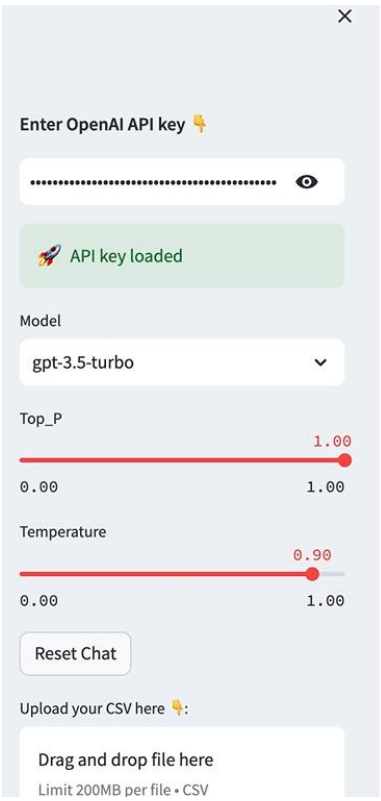
# Results & Discussions

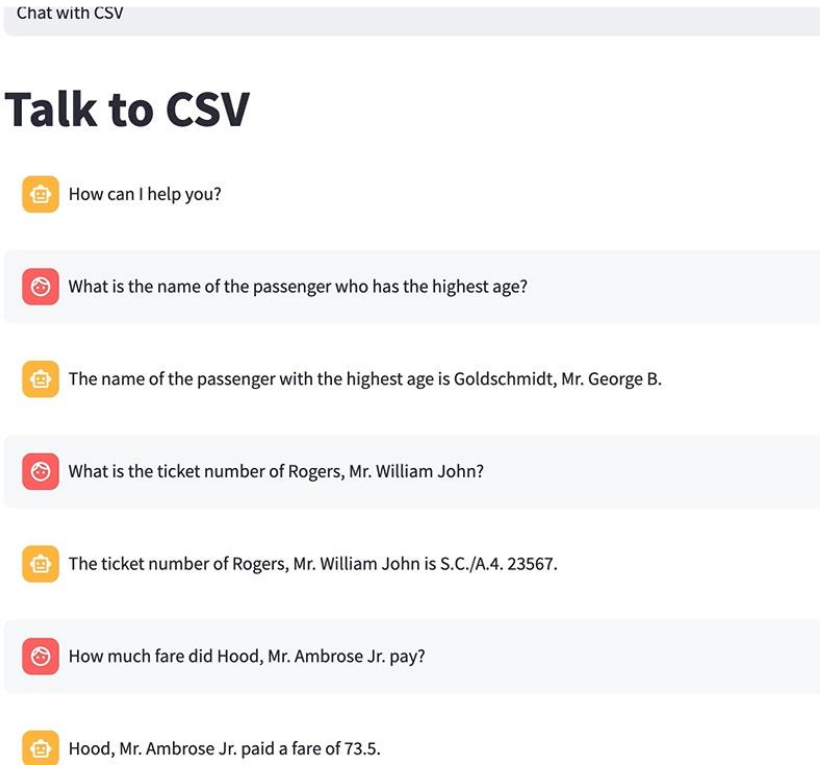
On loading the OpeanAI API Key, we can chat, summarize and analyze the uploaded csv file.



1. Chat with CSV:

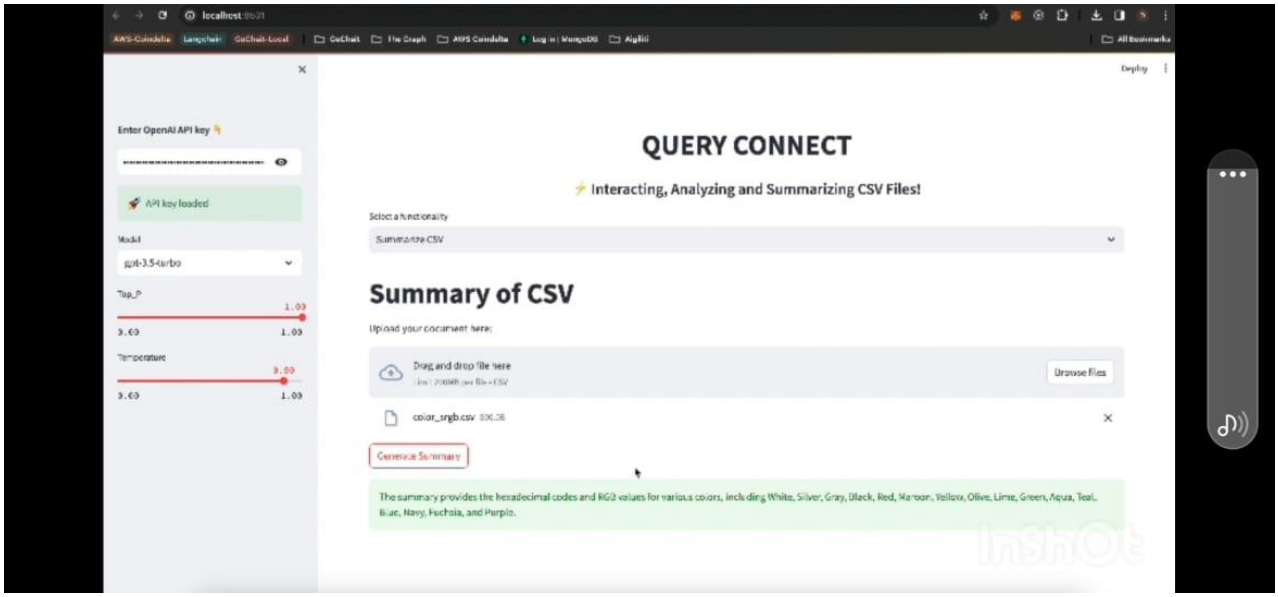
- Users can engage in a conversation with the data in the uploaded CSV file.
- The application uses an AI model to provide responses based on the context of the conversation and the content of the CSV file.
- This can be useful for querying specific information or gaining insights from the data in a conversational manner.



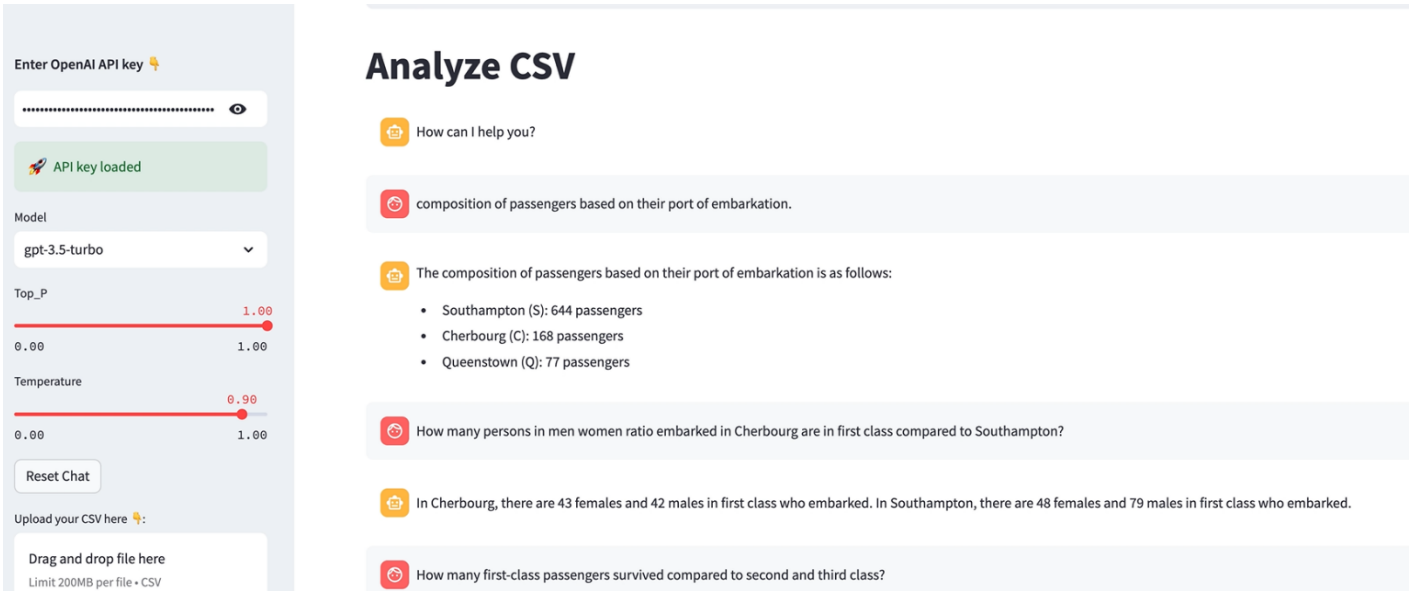
2. Summarize CSV:
   - Users can obtain a summary of the content within the uploaded CSV file.

   - The application utilizes an AI model to process the text and generate a concise summary, which can help users quickly understand the key points or insights contained in the data.



3. Analyze CSV:
    - Users can ask questions about the content of the CSV file, and the application provides relevant answers based on the data.
    - This feature allows for interactive exploration of the CSV data, enabling users to gain deeper insights or extract specific information of interest.

.



# Chapter 7
# Conclusions

The "QueryConnect" project represents a significant advancement in the realm of data exploration and analysis by leveraging natural language processing (NLP) techniques to interact with, analyze, and summarize CSV files. Through the development of an intuitive web application, users are empowered to engage with their data in a conversational manner, facilitating a more seamless and efficient workflow.

In conclusion, the "QueryConnect" project offers the following key contributions and benefits:

1. Enhanced User Experience:
    - By providing a chat interface for interacting with CSV data, the project offers a user-friendly and intuitive way to explore and analyze datasets.
    - Users can simply upload their CSV files and converse with the application to perform various tasks without the need for complex commands or programming knowledge.

2. Efficient Data Analysis:
    - With features such as summarization and analysis functionalities, users can quickly gain insights from their data without manual effort.
    - The summarization capability allows for the generation of concise summaries, while the analysis functionality enables tasks such as column name identification and data exploration.

3. Integration of Advanced NLP Models:
    - The project leverages state-of-the-art NLP models, such as OpenAI's GPT models, to provide intelligent responses and generate summaries that capture the essence of the input data.

- This integration enhances the quality of interactions and insights generated by the application.

4. Streamlined Deployment:
    - The application is built using Streamlit, a Python library for building interactive web applications, which facilitates easy deployment to web servers or cloud platforms.
    - This streamlined deployment process ensures accessibility and scalability for users.

5. Versatility and Adaptability:
    - The modular architecture of the project allows for easy customization and extension to accommodate additional functionalities or adapt to different use cases.
    - Users can tailor the application to suit their specific needs and integrate new features as desired.

Overall, the "QueryConnect" project represents a significant advancement in data interaction and analysis, offering a versatile and user-friendly solution for exploring and extracting insights from CSV files using natural language processing techniques. With its intuitive interface and powerful capabilities, the project has the potential to streamline data analysis workflows and empower users across various domains to make informed decisions based on their data.

# Chapter 8

# Future Scope

The "QueryConnect" project presents a promising foundation for bridging the gap between data and conversations, but its potential extends far beyond its current implementation. Here are several avenues for future development and expansion:

1. Integration with Additional Data Sources:
    - Currently, "QueryConnect" supports interactions with CSV datasets. However, expanding the platform to integrate with a wider range of data sources, such as databases, APIs, and real-time streaming data, would enhance its versatility and applicability across various domains.

2. Enhanced Natural Language Understanding:
    - Improving the natural language understanding capabilities of the platform can enable more nuanced interactions with users. This includes handling complex queries, understanding context, and supporting multi-turn conversations to provide more accurate and relevant responses.

3. Advanced Analytical Capabilities:
    - Incorporating advanced analytical techniques, such as predictive modeling, clustering, and sentiment analysis, can further enrich the insights derived from data. By integrating machine learning algorithms and statistical methods, "QueryConnect" can offer more sophisticated analyses and predictive capabilities.

4. Customization and Personalization:
    - Providing users with the ability to customize and personalize their interactions with the platform can enhance user engagement and satisfaction. This includes features such as customizable dashboards, saved queries, and personalized recommendations based on

user preferences and historical interactions.

5. Integration with External Tools and Services:
    - Integrating "QueryConnect" with external tools and services, such as business intelligence platforms, data visualization tools, and collaboration software, can extend its functionality and facilitate seamless workflows for data analysis and decision-making.

6. Security and Compliance Enhancements:
    - Strengthening security measures to ensure the confidentiality, integrity, and availability of data is paramount. Implementing robust encryption techniques, access controls, and compliance features to adhere to data protection regulations will enhance trust and adoption among users.

7. Scalability and Performance Optimization:
    - As the volume and complexity of datasets grow, ensuring the scalability and performance of the platform becomes essential. Optimizing algorithms, leveraging parallel processing techniques, and utilizing cloud computing resources can enhance the platform's scalability and responsiveness.

8. User Feedback and Iterative Development:
    - Continuously soliciting user feedback and incorporating it into the development process is crucial for refining and enhancing the platform. Conducting user studies, gathering feature requests, and prioritizing improvements based on user needs will drive the evolution of "QueryConnect" in line with user expectations.

By exploring these avenues for future development, "QueryConnect" can evolve into a comprehensive and

indispensable tool for empowering users to interact with their data in a conversational and insightful manner, unlocking new possibilities for data-driven decision-making and innovation.

# Chapter 9

# References

[1]
Rachith Aiyappa, Jisun An, Haewoon Kwak, and Yong-Yeol Ahn. 2023. Can we trust the evaluation on ChatGPT? arXiv:2303.12767 [cs.CL]
[2]
Daniel Bobrow. 1964. Natural language input for a computer problem solving system. (1964).
[3]
Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Je□rey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language Models Are Few-Shot Learners. In Proceedings of the 34th International Conference on Neural Information Processing Systems (Vancouver, BC, Canada) (NIPS'20). Curran Associates Inc., Red Hook, NY, USA, Article 159, 25 pages.
[4]
Ting-Rui Chiang and Yun-Nung Chen. 2018. Semantically-aligned equation generation for solving and reasoning math word problems. arXiv preprint arXiv:1811.00720 (2018).
[5]
Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayana Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Je□ Dean, Slav Petrov, and Noah Fiedel. 2022. PaLM: Scaling Language Modeling with Pathways. arXiv:2204.02311 [cs.CL]
[6]
Charles R Fletcher. 1985. Understanding and solving arithmetic word problems: A computer simulation. Behavior Research Methods, Instruments, & Computers 17, 5 (1985), 565–571.
[7]
James Hiebert. 2013. Conceptual and procedural knowledge: The case of mathematics. Routledge.
[8]
Danqing Huang, Jing Liu, Chin-Yew Lin, and Jian Yin. 2018. Neural math word problem solver with reinforcement learning. In Proceedings of the 27th Interna-

tional Conference on Computational Linguistics. 213–223.
[9]
Danqing Huang, Shuming Shi, Chin-Yew Lin, and Jian Yin. 2017. Learning Fine-Grained Expressions to Solve Math Word Problems. In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics, Copenhagen, Denmark, 805–814. https://doi.org/10.18653/v1/D17-1084
[10]
Zhenwen Liang, Wenhao Yu, Tanmay Rajpurohit, Peter Clark, Xiangliang Zhang, and Ashwin Kalyan. 2023. Let GPT be a Math Tutor: Teaching Math Word Problem Solvers with Customized Exercise Generation. CoRR abs/2305.14386 (2023). https://doi.org/10.48550/arXiv.2305.14386 arXiv:2305.14386
[11]
Zhenwen Liang, Jipeng Zhang, Lei Wang, Wei Qin, Yunshi Lan, Jie Shao, and Xiangliang Zhang. 2022. MWP-BERT: Numeracy-Augmented Pre-training for Math Word Problem Solving. arXiv:2107.13435 [cs.AI]
[12]
Qianying Liu, Wenyv Guan, Sujian Li, and Daisuke Kawahara. 2019. Tree-structured Decoding for Solving Math Word Problems. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP). Association for Computational Linguistics, Hong Kong, China, 2370–2379. https://doi.org/10.18653/v1/D19-1241
[13]
Jordan Meadows, Marco Valentino, Damien Teney, and Andre Freitas. 2023. A Symbolic Framework for Systematic Evaluation of Mathematical Reasoning with Transformers. arXiv:2305.12563 [cs.CL]
[14]
Anirban Mukherjee and Utpal Garain. 2008. A review of methods for automatic understanding of natural language mathematical problems. Arti□cial Intelligence Review 29 (2008), 93–122.
[15]
Alexander Scarlatos and Andrew Lan. 2023. Tree-Based Representation and Generation of Natural and Mathematical Language. arXiv:2302.07974 [cs.CL]
[16]
Paulo Shakarian, Abhinav Koyyalamudi, Noel Ngu, and Lakshmivihari Mareedu. 2023. An Independent Evaluation of ChatGPT on Mathematical Word Problems (MWP). arXiv:2302.13814 [cs.CL]
[17]
James R Slagle. 1965. Experiments with a deductive question-answering program. Commun. ACM 8, 12 (1965), 792–798.
[18]
Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. Advances in neural information processing systems 30 (2017).
[19] Lieven Verscha□el, Stanislaw Schukajlow, Jon Star, and Wim Van Dooren. 2020. Word problems in mathematics education: A survey. ZDM 52 (2020), 1–16.

[20]
Lei Wang, Yan Wang, Deng Cai, Dongxiang Zhang, and Xiaojiang Liu. 2018. Translating a Math Word Problem to a Expression Tree. In Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics, Brussels, Belgium, 1064–1069. https://doi.org/10.18653/v1/D18-1132
[21]
Lei Wang, Dongxiang Zhang, Lianli Gao, Jingkuan Song, Long Guo, and Heng Tao Shen. 2018. Mathdqn: Solving arithmetic word problems via deep reinforcement learning. In Proceedings of the AAAI conference on arti□cial intelligence, Vol. 32.
[22]
Yan Wang, Xiaojiang Liu, and Shuming Shi. 2017. Deep Neural Solver for Math Word Problems. In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics, Copenhagen, Denmark, 845–854. https://doi.org/10.18653/v1/D17-1088
[23]
Widodo Winarso and Toheri Toheri. 2021. An analysis of students' error in learning mathematical problem solving: The perspective of David Kolb's theory. Turkish Journal of Computer and Mathematics Education (TURCOMAT) 12, 1 (2021), 139–150.
[24]
Tianyu Wu, Shizhu He, Jingping Liu, Siqi Sun, Kang Liu, Qing-Long Han, and Yang Tang. 2023. A Brief Overview of ChatGPT: The History, Status Quo and Potential Future Development. IEEE/CAA Journal of Automatica Sinica 10, 5 (2023), 1122–1136. https://doi.org/10.1109/JAS.2023.123618
[25]
Zhipeng Xie and Shichao Sun. 2019. A Goal-Driven Tree-Structured Neural Model for Math Word Problems. In Ijcai. 5299–5305.
[26]
Bakman Ye□m. 2007. Robust Understanding of Word Problems with Extraneous Information. arXiv:math/0701393 [math.GM]
[27]
Dongxiang Zhang, Lei Wang, Luming Zhang, Bing Tian Dai, and Heng Tao Shen. 2020. The Gap of Semantic Parsing: A Survey on Automatic Math Word Problem Solvers. IEEE Transactions on Pattern Analysis and Machine Intelligence 42, 9 (2020), 2287–2305. https://doi.org/10.1109/TPAMI.2019.2914054