Kingdom of Saudi Arabia
Ministry of Education
King Faisal University
College of Computer Sciences & Information Technology

Automating Datasets Cleansing Using ChatGPT-4

A project submitted
in partial fulfillment of the requirements for the degree of
Bachelor of Science in Computer Science

by

Abdulaziz Sami Alshateeb (219009863)

Ali Hussain Alqattan (219022038)

Ali Hassan Alghazal (219023289)

Ali Abbas Almubarak (219036649)

Supervised by

Dr. Abdulelah Abdallah Algosaibi

Committee Member

Mr. Conrado Padua Vizcarra

June, 2024

ACKNOWLEDGEMENT

We would like to express our sincere gratitude to our project supervisor Dr. Abdulelah Algosaibi, and our graduation project committee members for their guidance, support, and encouragement throughout this project proposal. They were always available to answer our questions and provide us with valuable feedback. We are grateful for their expertise and mentorship.

We would also like to thank all of CCSIT faculty members, for their support and guidance during our time as students. We have learned a lot from them, and we are grateful for the opportunity to have studied at such a prestigious institution.

UNDERTAKING

This is to declare that the project entitled "Automating Datasets Cleaning using ChatGPT-4" is an original work done by the undersigned, in partial fulfillment of the requirements for the degree of Bachelor of Science at the Department of Computer Science at Computer Science, College of Computer Sciences and Information Technology, King Faisal University.

All the analysis, design, and system development have been accomplished by the undersigned. Moreover, this project has not been submitted to any other college or university.

Abdulaziz Sami Alshateeb

Ali Hussain Alqattan

Ali Hassan Alghazal

Ali Abbas Almubarak

ABSTRACT

The project's primary objective is to automate the data cleansing phase by integrating ChatGPT-4, thereby revolutionizing the process of building accurate models in data science. This innovative approach aims to address common data quality issues, including handling missing values, identifying and managing outliers, and encoding categorical variables, all of which are pivotal for ensuring the reliability and accuracy of machine learning models. The integration of ChatGPT-4 is anticipated to bring about a paradigm shift in data science workflows, promoting efficiency, accuracy, and consistency in the data preprocessing phase. The ultimate goal is to empower data scientists with a powerful tool that can comprehend their specific needs, generate context-aware data cleansing strategies, and contribute to the overall enhancement of model-building processes.

The main purpose of this project is to leverage the advanced capabilities of ChatGPT-4 in automating and streamlining the data preprocessing stage in data analysis and machine learning applications. This not only enhances efficiency but also allows data scientists to focus more on designing and fine-tuning models, rather than getting bogged down with the intricacies of data preprocessing. Ultimately, this could lead to the development of more robust and precise models, thereby pushing the boundaries of what’s possible in data science.

The problem we are trying to solve is to reduce the time and effort cost of data cleaning reduce human error while cleaning data and guarantee the correctness of data.

In conclusion, this project's primary objective is to transform the landscape of data science by automating the data cleansing phase through the integration of ChatGPT-4. By addressing common data quality issues. The anticipated paradigm shift in data science workflows is poised to bring about increased efficiency, accuracy, and consistency in the data preprocessing phase. Ultimately, the integration of ChatGPT-4 serves as a powerful tool to comprehend specific data scientist needs, generate context-aware data cleansing strategies, and contribute to the overall enhancement of model-building processes. By leveraging ChatGPT-4's advanced capabilities, the project not only streamlines data preprocessing but also empowers data scientists to focus more on model design and fine-tuning, pushing the boundaries of what's achievable in data science. The core problem addressed by this project is the reduction of time and effort in data cleaning, mitigating human errors, and ensuring the correctness of data, thus contributing to the advancement and efficiency of data science endeavors.

Keywords: ChatGPT-4, LLMs, Data Science, Data Cleansing, Data preprocessing, OpenAI.

Table of Contents

List of Tables ix
List of Figures x
1 Introduction 1
  1.1 Background 2
  1.2 Motivation 3
  1.3 Problem Statement 4
  1.4 Innovation And Utility 6
  1.5 Scope And Degree of Challenge 6
2 Comprehensive Analysis of Related Work 8
  2.1 Generative Pretrained Transformer (GPT) 8
  2.2 Prompt Engineering 10
  2.3 Related Work 11
    2.3.1 Research name: Towards Automated Data Cleaning Workflows 11
    2.3.2 Research name: Large Language Models as Data Preprocessors 13
    2.3.3 Research name: Leveraging ChatGPT API for Enhanced Data Preprocessing in NatUKE 13
    2.3.4 Research name: The Role of ChatGPT in Data Science: How AI-Assisted Conversational Interfaces Are Revolutionizing the Field 14
    2.3.5 Research paper Name: Automated Data Preprocessing for Machine Learning-Based Analyses 15
    2.3.6 Research Paper: Is a prompt and a few samples all you need? Using GPT-4 for data augmentation in low-resource classification tasks 16
  2.4 Data Cleansing Tools 16
    2.4.1 OpenRefine 17
    2.4.2 Trifacta Wrangler 18
    2.4.3 WinPure 18
    2.4.4 Tableau Prep 19
  2.5 Objectives 19
  2.6 Expected Outcomes 20

- 2.7 Methodology 20
  - 2.7.1 Approach 20
  - 2.7.2 Visual Representation 21
- 3 Identification Of Alternative Solutions 27
  - 3.1 List of Alternatives 27
  - 3.2 Analysis of Alternatives 27
- 4 Project Requirements 28
  - 4.1.1 Functional Requirements: 29
  - 4.1.2 Non-Functional Requirements: 30
- 5 Discussion Of Tools and Techniques Used During Project Proposal 31
  - 5.1.1 Lucid Chart 31
  - 5.1.2 Gantt.io 31
  - 5.1.3 Sequencediagram.org 32
  - 5.2 Appropriate Analysis 33
  - 5.2.2 Comparison Between AI Models 37
- 6 Details of partial implementation conforming to the design of the proposal phase 38
  - 6.1 Frontend Development 38
    - 6.1.1 Landing Page 38
    - 6.1.2 Home Page 39
    - 6.1.3 Datasets Page 39
    - 6.1.4 Clean With ChatGPT 39
  - 6.2 GPTs (DataGPT) 40
  - 6.3 Use Case 40
    - 6.3.1 Dataset Source 40
    - 6.3.2 Manual analysis 41
    - 6.3.3 ChatGPT Response 42
    - 6.3.4 Relationships Between Features 42
  - 6.4 Challenges and Limitations 43
- 7 Commands of tools and techniques being used during project implementation 44
  - 7.1 Project Management 44
  - 7.2 Web Development 44
    - 7.2.1 BootstrapMade 44
    - 7.2.2 UI Components 45

7.2.3 GPT Customization 46
7.3 Users' credentials 47
8 Preliminary outcomes/results 48
8.1 DataGPT Website 48
8.2 Customized GPT 48
9 Analysis of the preliminary result through comparison, validation, or verification 49
9.1 Website Development 49
9.2 Complete Use case 49
9.2.1 Features Analysis 50
9.2.2 Statical Analysis 51
9.2.3 Identify Dataset Issues and Suggested Actions 56
Issue 56
Description 56
Suggested Actions 56
Missing Values 56
Some features have missing values. 56
Impute missing values using statistical methods (mean, median, mode) or predictive modeling. 56
Outliers 56
Features like pktcount, bytecount, tx_bytes, and rx_bytes have extremely high maximum values. 56
Identify and treat outliers using z-score, IQR, or capping. 56
Negative Values 56
The pktrate feature contains negative values. 56
Investigate and correct negative values in the pktrate feature. 56
High Standard Deviation 56
Features like bytecount, pktcount, tx_bytes, rx_bytes, and tot_kbps show high variability. 56
Normalize or standardize these features to reduce the impact of outliers and improve algorithm performance. 56
Imbalanced Classes 56
The label feature is imbalanced, with one class more prevalent. 56
Use oversampling, undersampling, or synthetic data generation (SMOTE) to balance the classes. 56
Duplicate Rows 56

There are duplicate rows in the dataset. 56
Identify and remove duplicate rows to ensure data integrity.. 56
Inconsistent Data Types 56
src and dst IP addresses are categorical but not processed as such. 56
Convert categorical data into appropriate formats (one-hot encoding or label encoding). ... 56
Anomalous Values... 56
The pktrate feature has nonsensical negative values. 56
Investigate and correct anomalies or errors in the data recording. 56
9.2.4 Relationship Between Features... 57
9.2.5 Limitations and Recommendations 58
10 Remarks on preliminary results and intermediate conclusions 58
11 Details of project implementation conforming to the project proposal 60
11.1 Project Objectives and Deliverables 60
11.2 Team Collaboration and Roles.. 60
11.3 Mastery of tools and techniques used in project implementation 60
11.3.1 Auth0.. 61
11.3.2 openAI API 62
12 Overall project outcome/achievements 63
12.1 Landing Page 63
12.2 Home Page... 65
12.3 My Datasets Page... 66
12.4 Data Cleansing Page 67
13 Analysis of overall results through comparison, validation, or verification..................... 68
13.1 Comparison with Traditional Methods 68
13.2 Validation through Use Cases.. 68
14 Comprehensive remarks on overall project outcome and achievements.......................... 69
14.1 Conclusion 69
14.2 Future Work... 70
References 71

List of Tables

Table 1.5.1 Data Cleansing Tasks and Challenges with ChatGPT-4 6
Table 2.1.1 Advantages of Data Cleansing Automation with ChatGPT-4..................................... 9
Table 4.1.1.1 Functional Requirements.. 29
Table 4.1.2.1 Non-Functional Requirements... 30
Table 6.1.3.1 Instructions of Sequencediagram.org 32
Table 5.2.2.1 Bing Styles 37
Table 5.2.2.2 Comparative Analysis of AI Models.. 37
Table 6.2.1 steps to create costume GPTS 40
Table 6.3.2.1 Dataset Features Analysis 41
Table 6.3.3.1 Dataset features analysis by ChatGPT 42
Table 7.2.2.1 UI Assists Tools 45
Table 7.2.3.1 Prompt for building our custom GPTS.. 46
Table 7.3.1 Auth0 use cases and solution... 47
Table 9.2.1.1 Features Analysis DDoS Dataset.. 50
Table 9.2.2.1 Statical Analysis DDoS Dataset 51
Table 9.2.3.1 Identification of Dataset Issues………………………………..………………...56
Table 9.2.4.1 Features Relationships…………………………………………..……………57
Table 9.2.5.1 Limitations and Recommendations…………………………………..…………58

List of Figures

Figure 1.3.1 General Framework... 5
Figure 2.2.1 Example of COT 11
Figure 2.3.1.1 Research Paper.. 12
Figure 2.7.1.1 Waterfall.. 21
Figure 2.7.2.1.1 Use-case Diagram 22
Figure 2.7.2.2.1 Sequence Diagram 23
Figure 2.7.2.3.1 Activity Diagram.. 25
Figure 2.7.2.3.2 Activity Diagram.. 26
Figure 9.2.2.1 Bar Chart of Protocol 55
Figure 9.2.2.2 Bar Chart of Label... 55
Figure 11.3.1.1 Auth0 Email Verification……………………………………………………..61
Figure 11.3.1.2 Code Snippt……………………………………………………………...62
Figure 12.1.1 Landing Page..…………………………………………………………………63
Figure 12.1.2 Landing Page..…………………………………………………………………64
Figure 12.1.3 Landing Page..…………………………………………………………………64
Figure 12.2.1 Home Page……..………………………………………………………………65
Figure 12.3.1 My Datasets Page…..………………………………………………………….66
Figure 12.4.1 Chat Page...………..……………………………………………………………67

1 Introduction

Data science stands at the point of a transformative age with the arrival of ChatGPT-4, an advanced language model designed to automate the data cleansing phase. The goal of this project is to aid data scientists in building accurate machine-learning models. ChatGPT-4 ensures data accuracy and integrity by automating processes including addressing missing values, controlling outliers, and encoding categorical variables [1].

The primary objective of this project is to harness the capabilities of ChatGPT-4 to bring about a paradigm shift in data science workflows. The model's integration is expected to streamline the data preprocessing phase, thereby promoting efficiency, accuracy, and consistency.

As data science continues to grow in complexity and scope, the demands on data scientists to produce robust and precise models have never been greater. The traditional data preprocessing stage, often characterized by its time-consuming and error-prone nature, has been a bottleneck in the workflow. This project addresses this challenge head-on by leveraging the advanced capabilities of ChatGPT-4 to automate and refine the data preprocessing stage. As a consequence, the procedure is more efficient, freeing up data scientists to focus on building and improving models instead of being bogged down in the specifics of data cleansing.

The problem that this project seeks to solve is multifaceted. It aims to reduce the time and effort costs associated with data cleaning, minimize human error, and guarantee the correctness of the data.

In conclusion, ChatGPT-4 is a new tool that makes preparing data for machine learning faster and more accurate. It automates the cleaning of data, which saves time and reduces mistakes. This project uses ChatGPT-4 to improve how data scientists work, allowing them to focus on building better models without worrying about data issues.

1.1 Background

This project aims to integrate the advanced capabilities of ChatGPT-4 with data preprocessing tasks in the field of data science. ChatGPT-4, a generative pre-trained transformer, is renowned for its ability to understand and generate human-like text across various domains and contexts. Its architecture, combined with a vast amount of training data, allows it to exhibit impressive language capabilities, making it a valuable tool for a wide range of language-related tasks.

One of the key areas where ChatGPT-4 can make a significant impact is data cleaning, a critical yet time-consuming process that involves the correction or removal of inaccuracies, inconsistencies, and redundancies in datasets. By introducing ChatGPT-4 into this process, we aim to streamline and automate these tasks. With its advanced natural language processing capabilities, ChatGPT-4 can understand and interpret complex data structures, identify errors, and suggest appropriate corrections. This not only reduces the manual effort involved in data cleaning but also enhances the accuracy and reliability of the cleaned data.

In addition to data cleaning, ChatGPT-4 can also be utilized in other data preprocessing tasks in natural language processing (NLP), such as tokenization, stemming, lemmatization, and stop word removal. These tasks convert raw text into a usable format for machine-learning models. Furthermore, dimensionality reduction methods like Principal Component Analysis (PCA) can be applied to high-dimensional data to mitigate computational complexity and potential overfitting.

In essence, the integration of ChatGPT-4 with data preprocessing aims to enhance the overall quality, accuracy, and generalizability of models, playing a critical role in their effectiveness in real-world scenarios. By automating these tasks, data scientists can focus more on building and refining models, leading to more efficient and effective outcomes in data science projects.

1.2 Motivation

The motivation behind this project lies in the potential of ChatGPT-4, a generative pre-trained transformer, to revolutionize the field of data science, particularly in the realm of data cleaning and preprocessing.

ChatGPT-4, trained on a diverse range of tasks and a vast amount of data, can understand and generate human-like text across various domains and contexts. Its architecture, known as Transformer, excels in capturing long-range dependencies in data, making it well-suited for understanding and generating coherent and contextually relevant responses. This capability allows ChatGPT-4 to exhibit impressive language capabilities and generate contextually appropriate responses.

In natural language processing (NLP), preprocessing includes tokenization, stemming, lemmatization, and stop-word removal to convert raw text into a usable format for machine learning models. Dimensionality reduction, using methods like Principal Component Analysis (PCA), is applied to high-dimensional data to mitigate computational complexity and potential overfitting.

Ultimately, the integration of ChatGPT-4 with data preprocessing aims to enhance the overall quality, accuracy, and generalizability of models, playing a critical role in their effectiveness in real-world scenarios. By automating these tasks, data scientists can focus more on building and refining models, leading to more efficient and effective outcomes in data science projects.

1.3 Problem Statement

The primary focus of this project is to tackle the issue of inefficient and manual dataset cleansing. Currently, the process of cleaning datasets involves a lot of time and effort, leading to potential human errors. This slows down data analysis and machine learning applications, hindering their efficiency. Thus, there's a need for an automated solution that can streamline and simplify this entire process. Our project aims to leverage the power of ChatGPT-4, an advanced language model, to effectively automate dataset cleansing. By doing so, we aim to provide a highly specific, clear, and straightforward solution that enhances the accuracy and reliability of data used in various applications.

The following figure illustrates the general framework of the project:

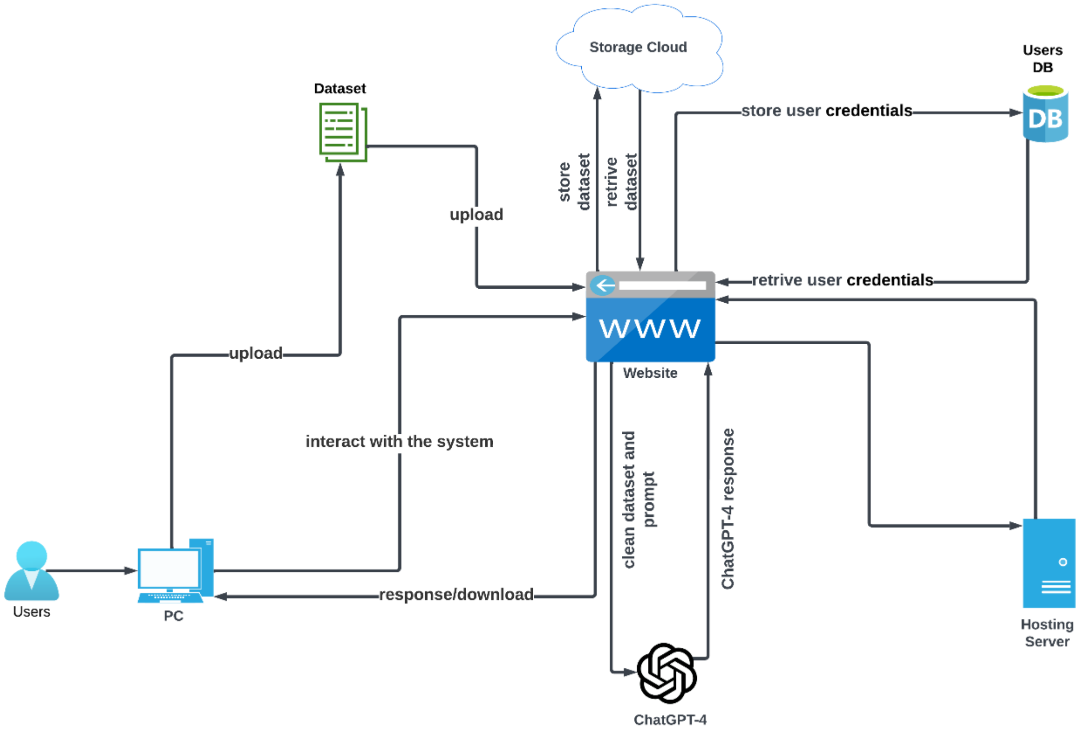


Figure 1.3.1 General Framework

This framework provides a high-level overview of the components involved in the system and how different components will interact with each other to achieve its main goal and provide a cleaned dataset to the user using ChatGPT-4.

1.4 Innovation And Utility

The innovative aspect of our project lies in the creation of a customized version of the ChatGPT-4 model (GPTs) [2], specifically designed to assist data scientists in dataset cleaning. This unique adaptation of ChatGPT-4 aims to automate and streamline the often tedious and complex process of data cleaning, thereby enhancing efficiency and accuracy in data science. By intelligently identifying and correcting errors, inconsistencies, and redundancies in datasets, our customized ChatGPT-4 model not only improves the quality of data but also allows data scientists to focus more on their core tasks of data analysis and model building. This innovation represents a significant stride in the field of data science, showcasing how advanced AI technologies can be harnessed to simplify and optimize data preprocessing tasks.

1.5 Scope And Degree of Challenge

This project aims to develop an automated data cleansing system utilizing ChatGPT-4, a large language model with advanced natural language processing capabilities. The system will be designed to handle various data cleansing tasks, including:

Task	Description
Identifying and correcting data errors	ChatGPT-4 will be employed to detect and rectify errors in data, such as typos, inconsistencies, and outliers
Handling missing values	The framework will implement strategies to impute missing values in datasets, ensuring data completeness and consistency
Standardizing data formats	ChatGPT-4 will be utilized to standardize data formats, such as dates, names, and addresses, ensuring uniformity and compatibility for downstream applications.
Removing duplicates	The framework will identify and remove duplicate records within datasets to eliminate redundancies and improve data integrity
Enhancing data quality	The overall data quality will be enhanced by addressing data inconsistencies, eliminating anomalies, and ensuring data accuracy.

Table 1.5.1 Tasks and Challenges

While automating data cleansing with ChatGPT-4 offers promising benefits, there are challenges to consider:

1. **Model Training:** Training ChatGPT-4 models requires large amounts of clean data, which can be difficult to obtain.
2. **Domain Expertise (Domain Engineering):** Understanding the context and degrees of specific data domains may require additional domain-specific training for ChatGPT-4.
3. **Explain ability:** Ensuring the explainability of ChatGPT-4's decisions and actions is crucial for trust and transparency.
4. **Data Privacy:** Protecting sensitive data during the cleansing process is paramount to maintaining data privacy and compliance.

2 Comprehensive Analysis of Related Work

This section provides a concise comprehensive analysis of the project's foundation. It delves into essential topics such as the Generative Pretrained Transformer (GPT), Prompt Engineering, and a review of Related Work in the field. Through this exploration, this section aims to provide the essential background knowledge required to fully understand and value the project.

2.1 Generative Pretrained Transformer (GPT)

Generative Pretrained Transformer, commonly known as GPT, is a type of artificial intelligence language model [3]. It's a complex mathematical representation of text or other types of media that allows a computer to perform tasks such as interpreting and producing language, recognizing or creating images, and solving problems. This is achieved in a way that seems similar to how a human brain works. GPT is a natural language system that can be used to answer questions, translate languages, and generate text in response to a prompt. It's not limited to natural language processing; you can use it for a variety of tasks depending on how you train the model. One ethical concern is that GPT models can inadvertently generate offensive content.

ChatGPT-4 is the most advanced system developed by OpenAI. It's designed to produce safer and more useful responses. GPT-4 can solve difficult problems with greater accuracy, thanks to its broader general knowledge and problem-solving abilities. It's more creative and collaborative than ever before, capable of generating, editing, and iterating with users on creative and technical writing tasks.GPT-4 surpasses ChatGPT in its advanced reasoning capabilities. GPT-4 is capable of handling over 25,000 words of text, allowing for use cases like long-form content creation, extended conversations, and document search and analysis [2].

Automating data cleansing with ChatGPT-4 offers several advantages over traditional manual methods:

Advantages	Description
Improved Data Quality	ChatGPT-4's ability to detect and rectify errors, impute missing values, and standardize formats significantly improves data quality, leading to more accurate and reliable analysis.
Increased Efficiency	Automation eliminates the need for manual data cleaning, saving time and effort for data analysts and scientists
Reduced Costs	Automation minimizes the need for specialized data-cleaning personnel, reducing labor costs and overhead expenses.
Enhanced Scalability	Automated frameworks can handle large and complex datasets efficiently, making them suitable for enterprise-level applications.
Accessibility	Automation democratizes data cleansing, making it accessible to a wider range of users with varying technical expertise.

Table 2.1.1 Advantages of Data Cleansing with ChatGPT4

Large LLMs today, such as GPT-4, are tuned to follow instructions and are trained on large amounts of data; so, they are capable of performing some tasks.

2.2 Prompt Engineering

Prompt engineering is a crucial skill for effectively communicating with advanced AI systems like ChatGPT. It involves providing structured instructions to these systems to guide their rule enforcement, and process automation, and to control the quality and quantity of their outputs. Think of prompts as a specialized form of programming tailored to modify how these AI models interact and respond. Similarly, prompt patterns serve as a method for sharing knowledge, much like software patterns, offering repeatable solutions to frequently encountered issues in generating outputs and interacting with these large-scale AI models [4].

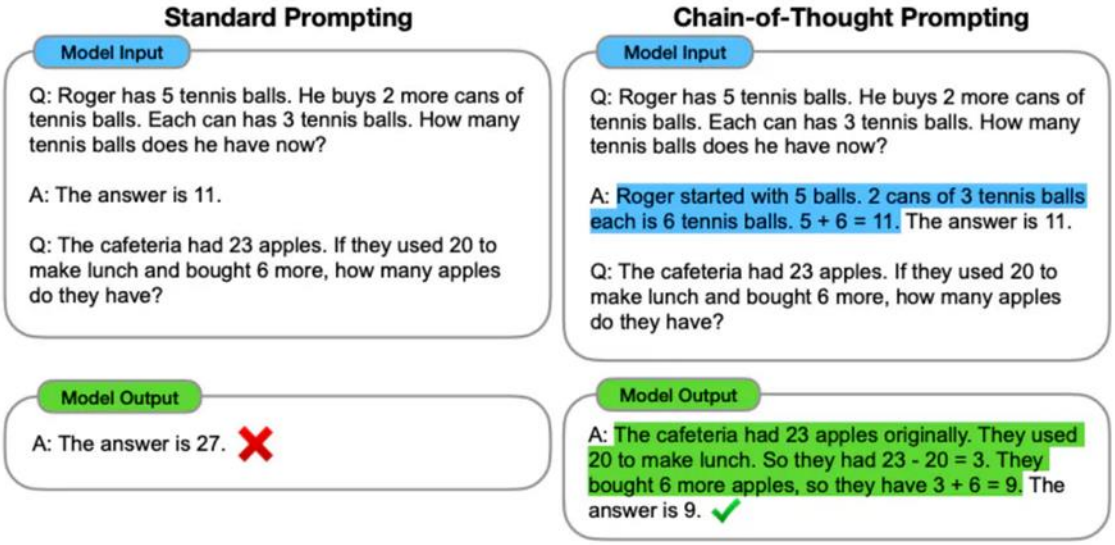
Prompt engineering skills help to better understand the capabilities and limitations of large language models (LLMs). Researchers use prompt engineering to improve the capacity of LLMs on a wide range of common and complex tasks such as question answering and arithmetic reasoning.

Some useful techniques help to reach accurate results such as:

- Chain of thought:

In 2022, Google researchers Wei et al. proposed Chain-of-Thought (CoT) prompting, an approach that encourages LLMs to break down a complex “thought” (an LLM’s response) into intermediate steps by providing a few demonstrations to the LLM (few-shot learning). They found that CoT prompting boosted LLMs’ performance at complex arithmetic, commonsense, and symbolic reasoning tasks, all types of tasks resistant to the improvements that scaling laws granted to LLMs in other areas. [5]

Example of COT:



Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?

A: The answer is 11.

Q: The cafeteria had 23 apples. If they used 20 to make lunch and bought 6 more, how many apples do they have?

A: The answer is 27. ✗

Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?

A: Roger started with 5 balls. 2 cans of 3 tennis balls each is 6 tennis balls. 5 + 6 = 11. The answer is 11.

Q: The cafeteria had 23 apples. If they used 20 to make lunch and bought 6 more, how many apples do they have?

A: The cafeteria had 23 apples originally. They used 20 to make lunch. So they had 23 - 20 = 3. They bought 6 more apples, so they have 3 + 6 = 9. The answer is 9. ✓

Figure 2.2.1 Example of COT

2.3 Related Work

This section will explore the research conducted in other search-related studies, highlighting the challenges encountered and the strategies implemented to address these issues throughout their investigations.

2.3.1 Research name: Towards Automated Data Cleaning Workflows

The problem addressed in this project is the challenge of data quality in AI-based technologies, which heavily rely on trustworthy and clean data [6]. Traditional data cleaning approaches often require prior knowledge about the dataset to select and configure the appropriate cleaning methods. However, for unknown datasets, it's unrealistic to have a priori knowledge of all potential data quality problems and formulate all necessary constraints at the outset.

They used the similarities between past cleaning tasks and current cleaning tasks to assess tool applicability to new datasets. They used tools such as NADEEF, Open Refine, and Katara and their detection strategy are Histogram, Gaussian, Gaussian Mixture, and Partitioned Histogram Modeling.

The following diagram shows how the application works:

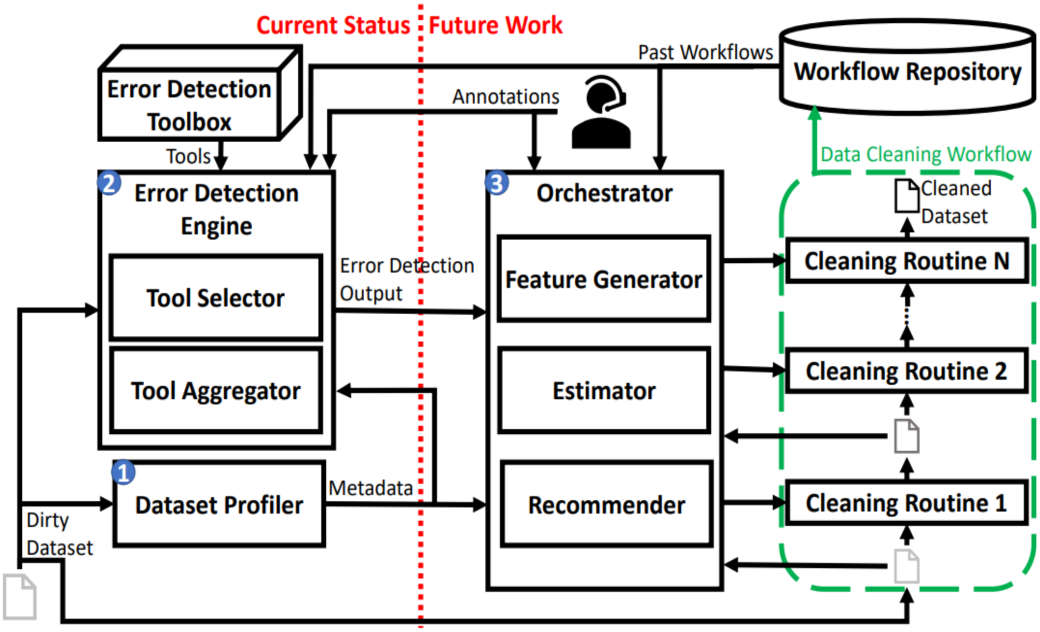


Figure 2.3.1.1 Research Paper

The project has made progress in automating data cleaning tasks, particularly in error detection. The focus is on understanding and utilizing metadata, establishing connections between metadata and data quality, and extending active learning to the correction of errors in the data.

2.3.2 Research name: Large Language Models as Data Preprocessors

This research paper experiments using LLMs as preprocessors and they apply the experiments on different models and evaluate the performance.

In this paper, they evaluate three Large Language Models (LLMs): GPT-3.5-turbo-0301 (referred to as GPT-3.5), GPT-4-0314 (referred to as GPT-4), and Vicuna-13B (referred to as Vicuna). The respective temperature parameters for these models are set at 0.75, 0.65, and 0.2. For SM tasks, we use 3 few-shot examples, and for other tasks, they use 10. The default batch prompting method is random batching. The batch size ranges for GPT-3.5, GPT-4, and Vicuna are [10, 20], [10, 15], and [1, 2], respectively. As baselines, they employ GPT-3 (text-davinci-002) with the best settings for all four tasks, HoloClean and HoloDetect for ED, IMP for DI, SMAT for SM, and Ditto for EM. As these baselines have been evaluated, they use these results as a reference.

Performance comparisons are presented in the result table (table 4.2 in the research paper) [1]. Both GPT-3.5 and GPT-4 generally surpass those reported on three out of the four tasks: DI, SM, and EM. For DI and SM, both models achieve superior performance than previous methods, particularly for SM. GPT-4 emerges as the victor on 4 out of 7 datasets for EM, with GPT-3.5 presenting strong competition. Vicuna provides reasonable answers on a very limited number of datasets for EM, achieving around 50% F1 score.

2.3.3 Research name: Leveraging ChatGPT API for Enhanced Data Preprocessing in NatUKE

The research paper focuses on enhancing the performance of knowledge extraction algorithms, particularly in the context of natural product knowledge extraction from academic literature. The key problem addressed in the paper is the improvement of data preprocessing in the existing NatUKE (A Benchmark for Natural Product Knowledge Extraction from Academic Literature) project using ChatGPT, a state-of-the-art language model developed by OpenAI.

The researchers leverage ChatGPT's natural language processing capabilities to preprocess data for knowledge graph embedding algorithms. The process involves several steps:

1. PDF Extraction: Extracting text data from PDF documents.

2. Data Cleaning: Removing unwanted characters, and error detection from Optical Character Recognition (OCR).
3. Text Slicing: Segmenting text into smaller units for easier processing.
4. Prompt Generation: Designing prompts for ChatGPT to generate relevant information.
5. Setting up OpenAI Call: Interacting with the OpenAI API, managing parameters such as response length and temperature.
6. Model Invocation: Calling the ChatGPT model with prompts and receiving generated text.
7. Result Retrieval: Storing and processing the generated results.
8. Post-Processing: Cleaning and structuring the output data from ChatGPT.
9. Training and Benchmarking: Using the processed data for further model training and evaluating performance against various metrics.

The improvements in results were most notable in the extraction of bioactivity (B), collection site (L), and isolation type (T) for DeepWalk and Node2Vec. EPHEN algorithm showed considerable improvements across all properties, with bioactivity (B) and isolation type (T) being the most significantly impacted [7].

2.3.4 Research name: The Role of ChatGPT in Data Science: How AI-Assisted Conversational Interfaces Are Revolutionizing the Field

The research paper explores the use of ChatGPT in data science to streamline workflows, enhance decision-making by analyzing unstructured data, and improve dataset quality with synthetic data. It also addresses ChatGPT's adaptability, coherent responses, and potential issues like bias and plagiarism. The paper aims to demonstrate how ChatGPT can support data scientists in becoming more efficient and addresses ethical concerns, proposing future research directions to further integrate AI tools like ChatGPT in data science.

ChatGPT enhances the efficiency of data science by handling routine tasks like data preparation, preprocessing, and model development. This automation allows data scientists to focus on complex challenges, such as crafting accurate predictive models and refining data visualization methods. Moreover, ChatGPT can sift through unstructured data sources, including customer comments and online critiques, to discover key insights that inform better decision-making strategies.

Based on the abstract and conclusion you provided, the results of the research paper indicate that ChatGPT can significantly aid data scientists by automating parts of their workflow, such as data preparation and model interpretation. It also shows promise in enhancing decision-making through the analysis of unstructured data and the generation of synthetic data to enrich datasets. However, the paper also acknowledges the limitations of ChatGPT, including potential biases and the risk of plagiarism. The key contribution of the paper is demonstrating the practical applications of ChatGPT in data science, while also highlighting areas for improvement and future exploration to ensure responsible use of AI in this field [8].

2.3.5 Research paper Name: Automated Data Preprocessing for Machine Learning-Based Analyses

The research paper you're referring to addresses the problem of data preprocessing in Machine Learning (ML). While there have been advancements in performance enhancement through AutoML libraries for tabular datasets, the field of data preprocessing itself has not seen significant progress. The paper proposes an automated pipeline for advanced preprocessing steps like feature engineering, feature selection, target discretization, and sampling. These steps are designed to improve the performance of ML models, especially for datasets with inter-feature dependencies.

The paper also introduces a new sampling method called "Bin-Based sampling" for general data sampling and during feature engineering to create and select new features efficiently. This method has been shown to reduce preprocessing time and improve model performance by about 4-7% with baseline models like RandomForest on OpenML datasets. The improvement is marginal with AutoML libraries. The authors suggest that future work could focus on parallelizing the pipeline to further reduce feature engineering time. The main goal of the research is to enhance ML model performance through better preprocessing techniques without the need for domain knowledge.

In the research paper, the authors tackled the challenge of improving data preprocessing for machine learning. They created an automated system that streamlines complex steps like feature engineering and selection. A key innovation is their Bin-Based sampling method, which speeds up the process while maintaining data quality. This approach proved to be effective for datasets with related features, leading to better machine-learning model performance. However, it's less useful for small datasets.

The research paper presents a new way to prepare data for machine learning, making the process faster and more efficient. The authors developed an automated system that includes a novel sampling method, leading to better performance of machine learning models on certain datasets. The improvement is notable with traditional models and slight with advanced AutoML libraries. The system is particularly effective for larger datasets with complex features but less so for smaller ones. The authors aim to improve the system's speed in future work. Essentially, the paper offers a smarter approach to data preprocessing that can boost model accuracy and reduce time spent on data preparation [9].

2.3.6 Research Paper: Is a prompt and a few samples all you need? Using GPT-4 for data augmentation in low-resource classification tasks

The project addresses the challenge of acquiring and annotating data for machine learning, which is costly and time-consuming, especially in complex domains with limited resources. The goal is to explore whether synthetic data generated by language models like GPT-4 and ChatGPT can augment small datasets and improve classification tasks’ performance.

The researchers used GPT-4 and ChatGPT to create synthetic data samples, testing two strategies: preserving the original label distribution and balancing it. They trained a multilingual language model on both real and synthetic data, assessing the performance with progressively larger training sample sizes and evaluating the zero-shot capabilities of GPT-4 and ChatGPT.

In the sentiment analysis task, synthetic data underperformed compared to human-annotated data, with little improvement beyond 2000 samples. For the hate-speech detection task, the model trained on crowdsourced data achieved high accuracy, but classifying offensive language remained challenging. In the social dimension classification, zero-shot approaches outperformed trained models, especially in identifying underrepresented classes, with balanced data strategies showing sample efficiency in training [10].

2.4 Data Cleansing Tools

In the domain of data science, the integrity and quality of data are paramount. This system showcases real-life data cleansing tools that are essential for ensuring the accuracy and usability of datasets. The purpose of these tools is to detect and fix mistakes, irregularities, and superfluous data that may affect the credibility of research outcomes and business intelligence. They leverage advanced algorithms and intuitive interfaces to streamline the data cleansing process, enhancing efficiency and reducing the likelihood of errors. From removing duplicate entries to standardizing data formats, these tools play a crucial role in preparing raw data for meaningful analysis, ultimately contributing to more informed decision-making and reliable outcomes.

2.4.1 OpenRefine

OpenRefine is a powerful free, open-source tool for working with messy data: cleaning it; transforming it from one format into another; and extending it with web services and external data [11].

The main features of OpenRefine are:

- **Faceting:** Drill through large datasets using facets and apply operations on filtered views of your dataset.
- **Clustering:** Fix inconsistencies by merging similar values thanks to powerful heuristics.
- **Reconciliation:** Match your dataset to external databases via reconciliation services.
- **Infinite Undo/Redo:** Rewind to any previous state of your dataset and replay your operation history on a new version of it.
- **Privacy:** Your data is cleaned on your machine, not in some dubious data laundering cloud.
- **Wikibase:** Contribute to Wikidata, the free knowledge base anyone can edit, and other Wikibase instances.

2.4.2 Trifacta Wrangler

Trifacta Wrangler is another one of the top data-cleaning tools on the market. This interactive and transformational tool enables data analysts to clean and prepare data very quickly compared to other tools. Because it concentrates on data analysis, less time is required for formatting. Trifacta Wrangler also relies on machine learning (ML) algorithms to recommend common data transformations and aggregations [12].

Advantages of Trifacta Wrangler:

- Less formatting time.
- Focus on data analysis.
- Quick and accurate.
- Machine learning algorithm suggestions.

2.4.3 WinPure

WinPure is a highly regarded data cleansing tool that offers a suite of features to quickly and accurately clean, scrub, and deduplicate data. It’s designed for both business and IT users, providing unmatched matching accuracy and speed for enterprise-level data cleansing tasks [13].

Features and benefits of WinPure:

- Advanced Data Profiling: Analyzes the quality of your data by checking formats, types, completeness, and value counts.
- WinPure CleanMatrix: A sophisticated method to apply a range of data-cleaning processes to your data.
- Intelligent Data Matching: Finds the most accurate matches with minimal false positives, thanks to its advanced data matching engine.

2.4.4 Tableau Prep

Tableau Prep is a data preparation tool that simplifies the process of cleaning, shaping, and combining data for analysis within Tableau. It’s designed to provide a visual and direct path to prepare your data, allowing you to see the effects of your data transformations instantly [14].

Features of Tableau Prep:

- Visual Data Preparation: Offers a visual experience where you can see row-level data, profiles of each column, and your entire data preparation process.
- Connect to Various Data Sources: Ability to connect to data on-premises or in the cloud, whether it’s a database or a spreadsheet.
- Immediate Results: Changes made to the data are reflected immediately, even on millions of rows of data.
- Smart Features: Tableau Prep can analyze your data and recommend cleaning operations to quickly fix problems in your data fields.

2.5 Objectives

The primary objectives of this project are:

1. To develop a robust automated data cleansing framework powered by ChatGPT-4.
2. To evaluate the effectiveness of ChatGPT-4 in handling various data cleansing tasks.
3. To demonstrate the applicability of the framework to real-world datasets.
4. To contribute to the advancement of automated data cleansing techniques using large language models.

2.6 Expected Outcomes

- reliable customized ChatGPT-4 model for dataset cleaning.
- user-friendly interface to interact with users.
- Increased Data Understanding:
    - ChatGPT-4 could provide insights into the nature of the dataset by identifying patterns, relationships, or anomalies that might not be immediately apparent during manual inspection.

2.7 Methodology

In the forthcoming methodology section, we will unveil the systematic approach and methodologies employed in the dataset-cleaning project utilizing ChatGPT-4. This section serves as a comprehensive guide to the strategies and techniques applied to refine and enhance the dataset. It will delve into the intricacies of leveraging ChatGPT-4's language processing capabilities for data cleaning, illustrating the step-by-step process and detailing the functionalities integrated into the system.

2.7.1 Approach

Plan-driven project management is a method where the project’s features and how they are to be developed are planned out in advance. This approach is based on project engineering management methods and is typically used for large software development projects. [15]

The Waterfall model is a sequential design process in software engineering where progress flows steadily downwards (like a waterfall) through various phases. These phases typically include requirements gathering, system design, implementation, testing, deployment, and maintenance. Each phase is distinct and must be completed before the next one begins.

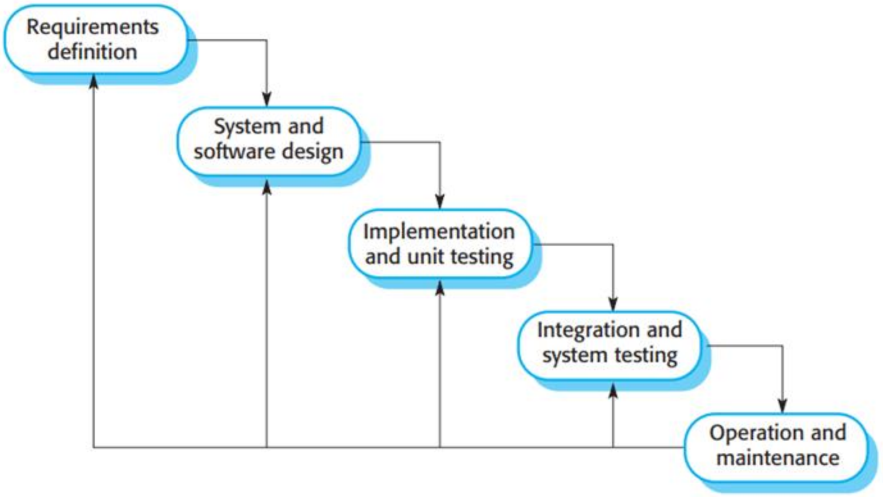

Requirements definition
System and software design
Implementation and unit testing
Integration and system testing
Operation and maintenance

Figure 2.7.1.1 Waterfall

There are some reasons why we chose a plan-driven approach for building this project:

Complexity: Cleaning datasets can be a complex process, especially when dealing with large volumes of data. A plan-driven approach allows for a detailed roadmap to be created, outlining each step of the process. This can help manage the complexity and ensure that no steps are missed.

Quality Assurance: A plan-driven approach allows for thorough quality checks at each stage of the project. This can help ensure that the cleaned datasets meet the necessary standards and are ready for further analysis.

2.7.2 Visual Representation

In this section, we provide an in-depth visual representation of the system. This is accomplished by utilizing a general framework that outlines the system's architecture and operations. We also used a use-case diagram to demonstrate the system-user interactions, highlighting the specific tasks that users can execute within the system. Lastly, we use sequence diagrams to show the information flow and event sequence within the system. Together, these visual aids offer a thorough and integrated understanding of the system's functionality and user engagement.

2.7.2.1 Use-case

The following use case diagram shows how the system's actors interact with the system:

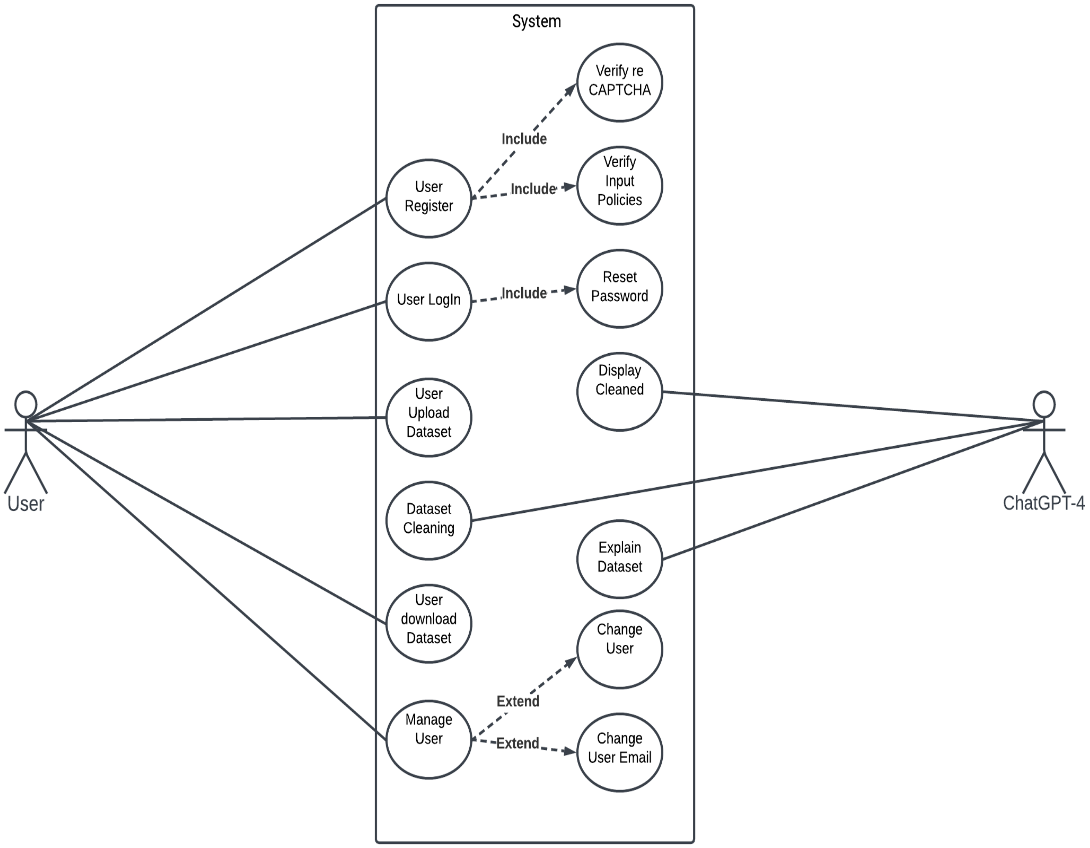


Figure 2.7.2.1.1 Use-case

The user can sign up for a new account and he has to follow the registration policy and verify using CAPTCHA to reduce the attempt to DOS/DDOS attacks.

The system is designed to empower users by offering a seamless and secure experience for updating their account information. Through this functionality, users gain the capability to independently modify both their password and email address.

Once the user registers on the website he/she could interact with the tool. It starts when he/she uploads a dirty dataset and asks to clean the dataset then the ChatGPT-4 model will clean the dataset and give a brief explanation about the dataset such as features nature of values and how the model can handle the obstacles that lead to a dirty dataset such as missing values, empty data, outliers, etc. … After that, it will display a cleaned dataset and it provides the ability to download it.

2.7.2.2 Sequence UML Diagram

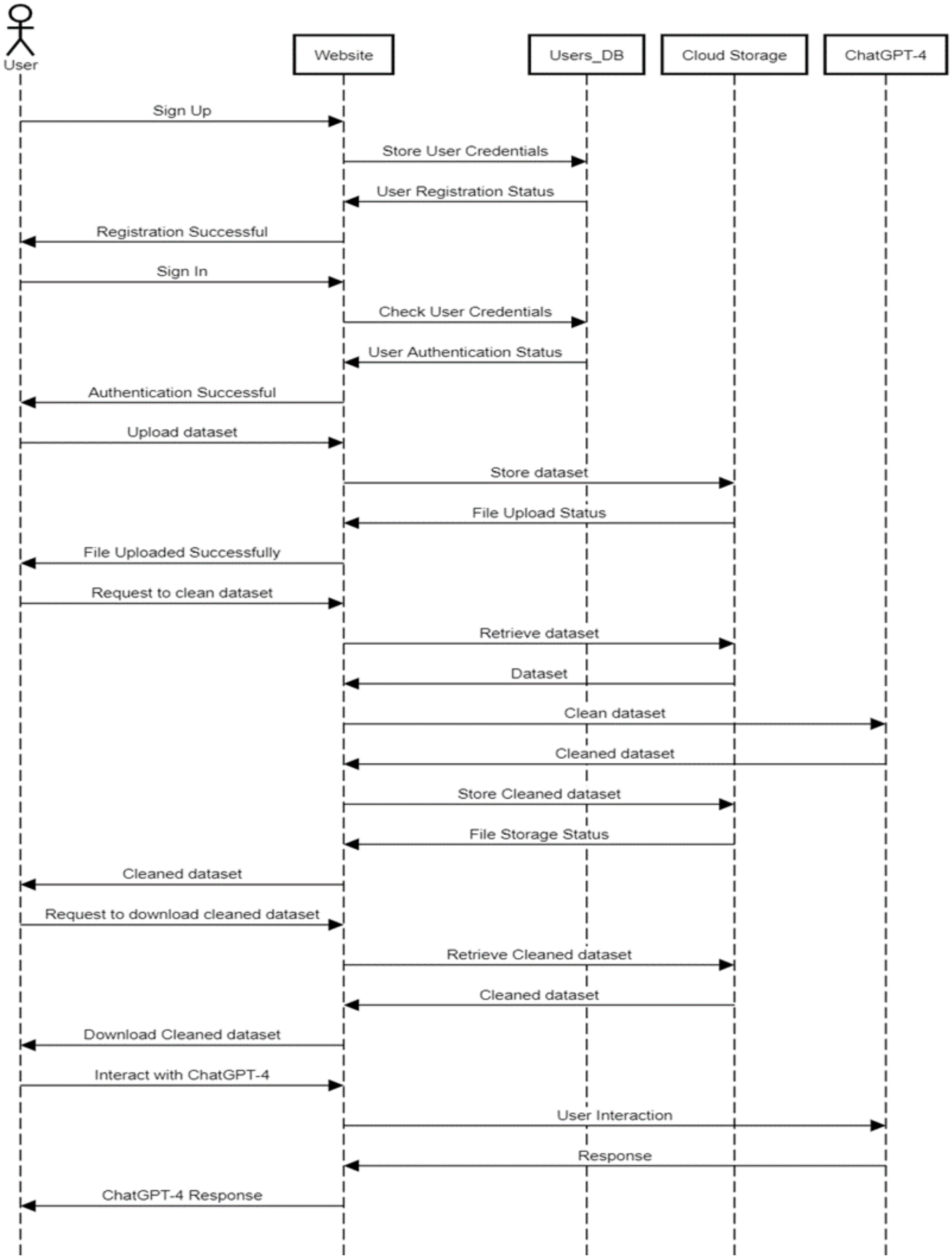


Figure 2.7.2.2.1 Sequence UML Diagram

The sequence diagram describes a series of interactions between a user and the system.

the user signs up on the website. The website stores the user’s credentials in the Users_DB. The Users_DB then returns the status of the user’s registration to the website, which informs the user that the registration was successful.

the user signs into the website. The website checks the user’s credentials in the Users_DB. The Users_DB returns the status of the user’s authentication to the website, which then informs the user that the authentication was successful.

The user then uploads a dataset to the website. The website stores the dataset in the cloud storage system. The cloud storage system returns the status of the file upload to the website, which then informs the user that the file was uploaded successfully.

The user requests the website to clean the dataset. The website retrieves the dataset from the cloud storage system and sends it to ChatGPT-4 to be cleaned. ChatGPT-4 returns the cleaned dataset to the website, which then stores the cleaned dataset back in the cloud storage system. The cloud storage system returns the status of the file storage to the website, which then informs the user that the dataset has been cleaned.

The user requests to download the cleaned dataset. The website retrieves the cleaned dataset from the cloud storage system and allows the user to download it.

the user interacts with ChatGPT-4 through the website. The website sends the user’s interaction to ChatGPT-4, which then returns a response. The website presents the response from ChatGPT-4 to the user.

This sequence of interactions allows the user to sign up, sign in, upload a dataset, have it cleaned by ChatGPT-4, download the cleaned dataset, and interact with ChatGPT-4, all through the website.

2.7.2.3 Activity Diagram

UML activity diagrams show the activities in a process and the flow of control from one activity to another. They are used to model both business processes and software processes.

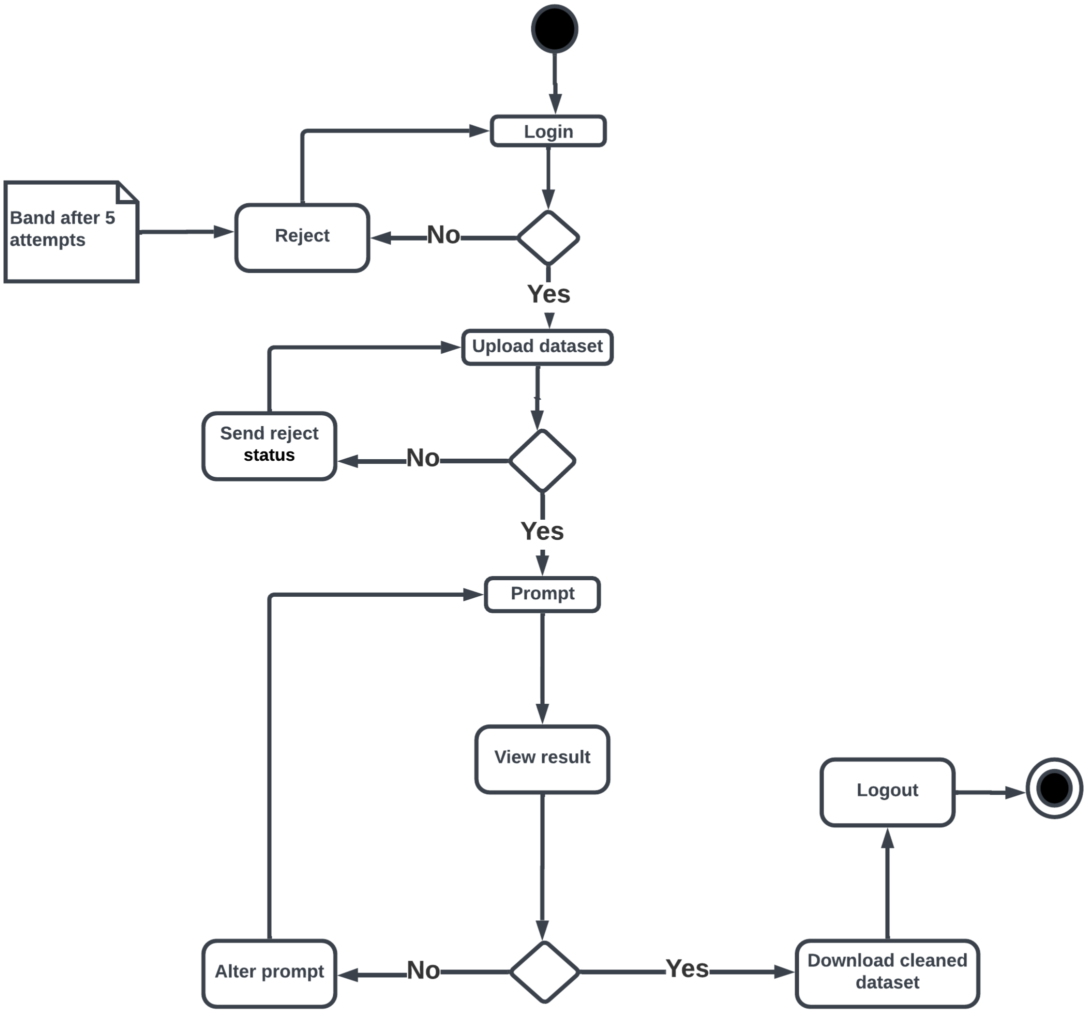


Figure 2.7.2.3.1 Activity Diagram1

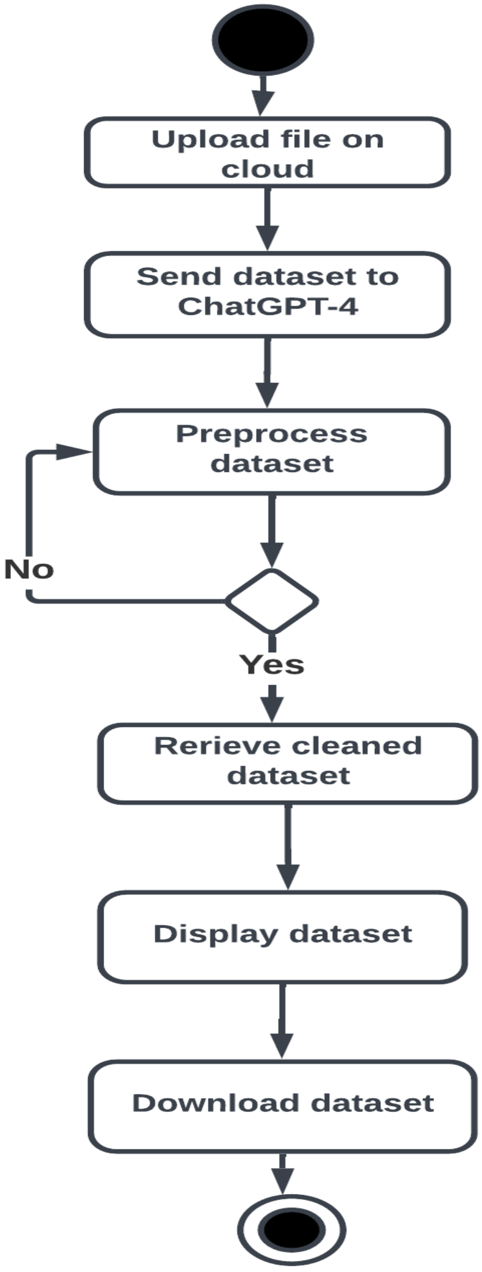


Figure 2.7.2.3.2 Activity Diagram2

This activity diagram shows how the dataset is taken from the user then prepossessed and retrieved.

3 Identification Of Alternative Solutions

This section delves into the exploration of various alternative solutions for our project, which aims to leverage ChatGPT-4 for automated dataset cleaning. The identification and analysis of potential solutions are crucial for selecting the most effective approach to meet our project goals.

3.1 List of Alternatives

We consider the following solutions for dataset cleaning:

1. **Conventional Data Cleaning Tool**: Will build software that follows predefined rules for data cleaning.
2. **Custom Scripting**: Tailored scripts developed for specific dataset cleaning tasks.

3.2 Analysis of Alternatives

Conventional data cleaning tools are software applications designed to automate the process of correcting or removing incorrect, corrupted, duplicated, or improperly formatted data within a dataset.

Custom scripting means writing special code to clean data. This code is made just for a certain set of data and helps fix any mistakes or organize the data better. It’s like creating a custom tool for a specific job. The code is written in languages that are good for working with data, like Python, R, or SQL.

4 Project Requirements

The MoSCoW technique to prioritize the priority of non-functional requirements and functional requirements is a simple yet effective way to prioritize tasks or requirements in a project. It stands for Must-have, Should-have, Could-have, and Won't-have.

1. Must-have: These are non-negotiable requirements that are essential for the project to be successful. If a must-have is not included, the project delivery should be considered a failure.

2. Should-have: These are important requirements that are not critical for the launch but add significant value. They can be scheduled for a future release without impacting the current one. The project or product still functions without them, but their inclusion would be beneficial.

3. Could-have: These are desirable requirements that are not necessary but could improve user experience or customer satisfaction with a small development cost. They are included if time and resources permit.
4. Won't-have: These are the least critical, lowest payback items, or those not appropriate at the current time. Stakeholders agree that these requirements will not be implemented in the current delivery but may be considered in the future.

4.1.1 Functional Requirements:

ID	Name	Description	Rationale	Priority
FR-101	Integration of ChatGPT-4	This project necessitates the seamless integration of ChatGPT-4 to automate the data cleansing phase. This involves establishing the required infrastructure and interfaces for ChatGPT-4 to engage with the data and execute necessary operations.	ChatGPT is a 4-based project that uses the power of the model.	M
FR-102	Data Cleansing	The system should be able to identify and handle missing values, manage outliers, and encode categorical variables.	The project aims to clean data using scientific methods.	M
FR-103	Error Mitigation	The system shall mitigate the errors on the dirty dataset.	One of the aspects of the project is to reduce human errors and help data scientists build accurate models.	S
FR-104	Context-Aware	The system should be able to generate data cleansing strategies based on the specific needs of the data scientists.	ChatGPT-4 should have the ability to justify the data cleansing actions.	C
FR-105	User Interface	The system should provide an intuitive user interface for users to interact smoothly with the system.	Facilitates ease of use and enhances user experience.	S
FR-106	Support for Multiple Data Formats	The system should support a wide range of data formats for input and output.	Enhances compatibility with different data storage systems.	S
FR-107	Publish on the GPT store	Once the customized GPT is developed and tested, it needs to be published on the GPT store	Sharing the customized GPT on GPTstore allows for collaboration and wider adoption	M

Table 4.1.1.1 Functional Requirements

4.1.2 Non-Functional Requirements:

ID	Name	Description	Priority
NFR-101	Performance	The system must perform efficiently to process and clean datasets in a reasonable amount of time.	M
NFR-102	Fault tolerance	The system should gracefully handle any errors or interruptions during the cleaning process.	S
NFR-103	Availability	Continuous availability is crucial for an automated system to be used when needed.	M
NFR-104	Usability	While important, the usability can be enhanced over time as users interact with the system.	C
NFR-105	Accessibility	Making the system accessible to a wider range of users.	C
NFR-106	Reliability	The system should reliably clean datasets.	S
NFR-107	Security	The system must apply security methods to protect the user’s data and the system itself.	M

Table 4.1.2.1 Non-Functional Requirements

5 Discussion Of Tools and Techniques Used During Project Proposal

This section will detail the specific tools used to create diagrams, charts, and sequence flows, as well as the techniques employed to ensure a structured and efficient proposal process. We will discuss how each tool has contributed to the clarity and organization of our project, and how the techniques have streamlined our workflow, ultimately leading to a robust and well-defined proposal for this project.

5.1.1 Lucid Chart

A lucid chart has proven to be an essential resource in our project. It has allowed us to construct detailed and understandable diagrams that effectively depict the various elements and procedures of our project. These diagrams have been instrumental in helping us visualize and comprehend the intricacies of our project, and they have also improved the quality of our project report by offering visual depictions of our work.

We have used it in:

1. General Framework diagram of the project.
2. Use case diagram.
3. Activity diagram.

5.1.2 Gantt.io

It has allowed us to create detailed Gantt charts that effectively map out the Proposed working plan the timeline and the progression of our project.

We have used it in:

- Gantt chart.

5.1.3 Sequencediagram.org

It has allowed us to create detailed sequence diagrams that effectively depict the interaction or communication between different components in our project.

This tool built a sequence diagram based on a specific set of instructions; the following table shows examples of these instructions:

Instruction	Description
title Title **A->B: info**	This instruction creates a title for the diagram
participant Participant	individual components in the system that interact with each other.
actor Actor	Users interact with the system
A->B: request **A<--B: response**	message sent from one participant. reply from the receiving participant back to the initiating participant.

Table 6.1.3.1 Instructions of Sequencediagram.org

In conclusion, the tools and techniques employed during the project proposal phase have been critical in shaping the project. Lucid Chart provided a platform for creating clear and comprehensive diagrams, which facilitated a deeper understanding of the project's framework and use cases. Gantt.io enabled the development of detailed Gantt charts, which outlined the project's timeline and ensured a well-organized approach to meeting milestones. Sequencediagram.org offered the means to visualize the interactions between project components, enhancing communication and coordination within the team.

These tools not only contributed to the clarity and organization of the project but also ensured that the proposal process was structured and efficient.

5.2 Appropriate Analysis

ChatGPT-4's NLP prowess promises to revolutionize data cleaning, but caution is key. Traditional methods struggle with complex data, while unfettered AI use can be risky. This section advocates for a critical analysis approach, where we leverage ChatGPT-4's strengths strategically. We'll identify tasks where its nuanced language understanding shines while acknowledging limitations and carefully interpreting its outputs. This isn't about surrendering control; it's about collaborating and wielding AI as a tool under our watchful guidance. Let's dance with data, ChatGPT-4 by our side, but with critical thinking leading the steps.

5.2.1.1 Data Preprocessing

Data preprocessing is a critical step in the data mining and data analysis process that transforms raw data into a format that can be understood and analyzed by computers and machine learning. Raw, real-world data in the form of text, images, video, etc., is often messy, containing errors and inconsistencies, and doesn’t have a regular, uniform design. [16]

The following subsection will discuss each step in detail.

5.2.1.2 Duplicate Entries

It's crucial to always scan the dataset for any duplicate entries. In certain real-world scenarios, these duplicates may hold significance. In such cases, it's typically beneficial to consolidate them into a single entry, while adding an extra column to denote the count of unique entries. However, there are instances where duplication is merely a byproduct of the data generation process. For instance, if the data is extracted by selecting specific columns from a larger dataset, there wouldn't be any duplicates when considering the other columns.

5.2.1.3 Multiple Entries for a Single Entity

This scenario is somewhat more intriguing than mere duplicate entries. Often, each real-world entity should logically correspond to a single row in the dataset. This is usually because some of the entries have become outdated, leaving only one row that is currently accurate.

5.2.1.4 Missing Entries

Often, when some entities are not included in a dataset, they share certain features that exclude them. For instance, imagine you have a record of all transactions from the past year. You group these transactions by customer and calculate the total transaction size for each one. This dataset will have one row per customer, but any customer who didn't make any transactions in the past year won't be included at all. In such a case, you can match the derived data with a known list of all customers and fill in the correct values for the missing ones.

Sometimes, data is missing because it was never collected for some entities. For example, two factories might produce a certain product, but only one of them collects this specific data.

5.2.1.5 NULLs

NULL entries usually indicate that we don't have specific information about an entity. But why is that?

The simplest reason is that there might have been an error in the data collection process. The implications of this depend on the situation.

When it comes to analytics, many algorithms can't process NULLs. In such cases, it's often necessary to replace the missing values with a reasonable substitute. This could be an estimate based on other data fields, or you might just use the average of all the non-null values.

In some instances, NULL values occur because the data was never collected. For example, one factory might take certain measurements while producing widgets, but another factory might not. The comprehensive data table for all widgets will then have

NULLs for the data that wasn't collected by a particular factory. Because of this, whether a variable is NULL can sometimes be a significant feature. The factory that produced the widget could be a crucial factor in what you're trying to predict, regardless of the other data you've collected.

5.2.1.6 Huge Outliers

Sometimes, a significant deviation in the data is due to a truly unusual event. How to handle this depends on the situation.

In some cases, it might be best to remove these outliers from the dataset. For instance, when analyzing web traffic, you're typically interested in predicting human page views. A sudden surge in recorded traffic is more likely to be the result of a bot attack than human activity.

In other scenarios, outliers could simply indicate missing data. Some storage systems don't support the explicit concept of a NULL value, so a predetermined value is used to represent missing data. If you notice many entries with identical, seemingly random values, this could be what's happening.

5.2.1.7 Out-of-Date Data

In many databases, each row comes with a timestamp indicating when it was added. When an entry is updated, the original data isn't overwritten; instead, a new row with a current timestamp is inserted. As a result, many datasets contain entries that are no longer valid but can be useful if you're trying to trace the database's history.

5.2.1.8 Artificial Entries

A lot of industrial datasets contain synthetic entries that are intentionally added to the actual data. This is typically done to test the software systems that handle the data.

5.2.1.9 Irregular Spacings

Many datasets contain measurements taken at regular intervals. For instance, you might have website traffic data recorded every hour or the temperature of an object measured at each inch. Most algorithms that handle this kind of data assume that the data points are evenly spaced, which can be a problem if they're not.

If the data comes from sensors measuring something like temperature, you usually have to use interpolation techniques to create new values at evenly spaced points.

A specific case of irregular spacing occurs when two entries have the same timestamps but different values. This typically happens because timestamps are only recorded with a certain level of precision. If two measurements are taken within the same minute and time is only recorded up to the minute, their timestamps will be the same.

5.2.2 Comparison Between AI Models

In this section, we will be comparing three prompts AI models: Bing, BARD, and ChatGPT-4. Each of these models has its unique strengths and capabilities, and they have been developed with different objectives and methodologies in mind.

In Bing AI there are three styles of conversation which are illustrated below table:

Style	Answer Length	Content-Type	Example Prompt	Example Answer
Creative	Longer	Imaginative	Write a poem	Roses are red, violets are blue, I’m a chatbot, and I like you.
Precise	Shorter	Factual	Write a code	print("Hello, world!")
Balanced	Medium	Informative	Write an essay	An essay is a piece of writing that presents a topic and an argument. It usually consists of an introduction, a body, and a conclusion.

Table 5.2.2.1 Comparison Between AI Models

With the diversity of AI models, the following table compares ChatGPT-4 with the other models:

Feature			
Developer	Google	OpenAI	Microsoft
Model	LaMDA	GPT-4	GPT-4
Parameters	137B	100T	100T
Input	Text	Text, Image	Text, Image
Output	Conversational	Various	Imaginative, Factual, Informative
Dataset	Infinite	Internet	Internet
Cost	Free	$20/month	Free
Limitations	Memory, Accuracy, Ethics	Memory, Accuracy, Ethics	Memory, Accuracy, Ethics

Table 5.2.2.2 ChatGPT4 vs Other AI Models

6 Details of partial implementation conforming to the design of the proposal phase

This section outlines the front-end development of the DataGPT website and discusses the web pages that we tried to make as user-friendly as possible. Also, another section discusses a use case for data discovery which is a part of the data science life cycle. The system architecture design shows the proper architecture for this project followed by the challenges and limitations that occurred during this milestone.

6.1 Frontend Development

The website is designed to be user-friendly, functional website deliver the main service of this project. In the following sections, we will delve into the details of these pages, highlighting their functionalities and how they contribute to an intuitive and seamless user experience.

The initial results will be shown in the Preliminary Outcomes/results section.

6.1.1 Landing Page

For developing the landing page we used Bootstrapmade, we chose this template because of its simplicity and modernity. It consists of multiple sections. The first section is the hero section which illustrates the main feature of this project.

The feature section contains the main system features with a short description for each one. The FAQ section contains some important questions that the user may need to know about.

The team section contains the developer’s information and Linktr.ee links to reach out to the developers.

6.1.2 Home Page

The interface welcomes users with a simple animation upon their arrival. Additionally, it enables the user to navigate my datasets page and chat page. By doing that, we improved the user experience.

6.1.3 Datasets Page

This page displays all the datasets uploaded by the user, presented as cards that include the following: dataset information, download button, and delete button for each dataset. Also, at the top of the page, there are the uploading button and search box.

6.1.4 Clean With ChatGPT

Since OpenAI does not provide custom GPTs API, we developed a simple chat page to enable the users to interact with ChatGPT4. Also, there is a button that redirects the user to our custom GPT (DataGPT) on the GPT store.

6.2 GPTs (DataGPT)

GPTs are custom versions of ChatGPT that users can tailor for specific tasks or topics by combining instructions, knowledge, and capabilities. They can be as simple or as complex as needed [17].

The following table shows how to create a GPT:

Step	Instruction
1	Go to https://chat.openai.com/gpts/editor or select your name and then “My GPTs”
2	Click on “Create a GPT”
3	Use the Create tab to communicate with the GPT Builder. Suggest ideas like, 'Make a creative who helps generate visuals for new products' or 'Make a software engineer who helps format my code.'
4	In the Configure tab, name your GPT and set its description. This is also where you select the actions your GPT can take, such as browsing the web or creating images.
5	To finalize and share your GPT, click on “Publish” and decide if you want to share it with others.

Table 6.2.1 Steps to Create Custom GPT

6.3 Use Case

This use case is designed to demonstrate the capability of ChatGPT4 vs human being capabilities in preprocess data. We have chosen a dataset related to food prices in poor countries, this dataset could be used to build machine learning models. The details will be explained in the following sections.

6.3.1 Dataset Source

This dataset is taken from Worldbank [18]. Afghanistan, Armenia, Burundi...and 33 more - Monthly food price estimates by product and market (worldbank.org). This dataset is about food pricing from 2007 to 2024. In the following section, we will compare between human being response vs ChatGPT4 response. The goal of our experiment is to check out the ChatGPT-4 capabilities when it comes to with dealing open data and trying to extend new features & data.

The experiment is divided into two parts, the first part is about the ability of ChatGPT-4 to understand the context of the dataset and discover the relationships between features.

The second part will be about the generated content and it will be detailed in the milestone-4 report.

6.3.2 Manual analysis

At this stage, we will only explore the data and features of this dataset, advanced preprocessing steps will be shown in the next milestone.

Monthly price estimates at market/commodity level (all available countries)

In the below table there is a description of the features:

Feature	Description
Country	----
Region	The specific region within the country.
Market	The specific market within the region.
Product	The specific product type.
Date	DD/MM/YYYY
Open	The opening price of the product.
High	The highest recorded price.
Low	The lowest recorded price.
Close	The closing price.
Currency	The currency in which the prices are denoted.

Table 6.3.2.1 Dataset Features Analysis

The Open, High, Low, and Close prices are directly related to each other as they represent different aspects of the product's price within a single day. This can provide insights into the volatility of the product's price.

The Date feature can be used to track changes in the product's price over time, which can help identify trends or patterns.

The Country, Region, and Market features can provide context for the price data, as prices can vary based on location due to factors like local supply and demand conditions, taxes, and transportation costs.

6.3.3 ChatGPT Response

Feature	Description
Country	The country where the market data was collected.
Region	A more specific area within the country.
Market	The particular market within the region.
Product	The type of product, in the displayed rows is 'bread'.
Date	The date when the data was recorded.
Open	The opening price of the product on the given date.
High	The highest price of the product on the given date.
Low	The lowest price of the product on the given date.
Close	The closing price of the product on the given date.
Currency	The currency in which the prices are recorded.

Table 6.3.3.1 Dataset Features Analysis by ChatGPT

6.3.4 Relationships Between Features

- **Temporal Relationship** (Date with Open, High, Low, Close): The date feature has a direct relationship with the Open, High, Low, and Close prices. These prices fluctuate over time due to various factors such as supply and demand, seasonality, and economic conditions.
- **Geographical Relationship** (Country, Region, Market with Prices): The geographical features (Country, Region, Market) are directly related to the prices (Open, High, Low, Close). Different regions may have different price levels due to varying local economic conditions, availability of the product, and other regional factors.
- **Product and Prices:** While the dataset displayed focuses on 'bread', it's implied that the dataset could include other products. The type of product is directly related to its pricing data, as different products will have different price points and fluctuations based on their unique supply and demand dynamics.
- **Currency and Prices:** The currency is related to the pricing data as it defines the unit of measurement for the prices. Additionally, currency value fluctuations can affect the perceived price changes when comparing internationally.

6.3.4.1 Feature Extraction and Analysis

For analysis, you might extract features such as:

- **Monthly Price Change:** The difference in closing prices between months to analyze price trends.
- **Volatility:** Measured as the difference between the High and Low prices within a period, indicating market stability or instability.
- **Average Price:** Calculated over a specific period to understand the general price level.

6.4 Challenges and Limitations

During the implementation, we faced several challenges, including issues related to data availability and front-end design. A major obstacle we faced was getting access to private datasets, a critical component of our data availability concerns. We intended to speak with our supervisor about these issues to get advice and possible fixes. We aimed to navigate these obstacles efficiently, ensuring the project's progress without compromising the quality or integrity of our work. Through collaborative problem-solving and leveraging our supervisor's expertise, we hoped to overcome these barriers and complete our project.

We faced some issues related to frontend design due to the limitations of skills in UI/UX knowledge. To resolve that we used a mixture of prebuild templates and UI elements we have done some research to find the best UI/UX practices.

Further challenges may be discovered during the development of the backend.

As OpenAI does not provide a custom GPTs API, we developed a simple chat website that allows users to communicate with ChatGPT-4.

7 Commands of tools and techniques being used during project implementation

This section will talk about the tools that have been used during the implementation phase. A variety of tools assisted us in accomplishing the assigned tasks and it will be illustrated in this section.

7.1 Project Management

Trello is the visual work management tool that empowers teams to ideate, plan, manage, and celebrate their work together in a collaborative, productive, and organized way. Users can create boards for different projects, add lists to represent stages or categories of tasks, and populate these lists with cards that represent individual tasks. Cards can be moved between lists as tasks progress from one stage to the next, and they can be assigned to specific team members, given due dates, and enriched with detailed descriptions, attachments, comments, and more.

7.2 Web Development

In terms of front-end, we used HTML, CSS, and JS. Along with other tools to meet the usability requirements of the system as shown in the next sections.

7.2.1 BootstrapMade

BootstrapMade offers easy-to-use, customizable templates suitable for various websites. With options from free basic templates to premium plans with more features, they cater to both beginners and professional developers. This website helps users quickly create professional-looking websites without much effort.

We have used the template shown below figure:

Link of the template: https://bootstrapmade.com/ninestars-free-bootstrap-3-theme-for-creative/

7.2.2 UI Components

In order to achieve the requirement of having a user-friendly system when it comes it UI/UX there are some websites that provide a prebuilt awesome joyful components to be used in the project. The following table shows some of these websites that inspired us and we used components from it.

Website	Link
unDraw	https://undraw.co/
ICONS8	https://icons8.com/
UIVerse	https://uiverse.io/
LottieFiles	https://lottiefiles.com/
Shadcn/ui	https://ui.shadcn.com/
FreeFrontend	https://freefrontend.com/
coolors	https://coolors.co/

Table 7.2.2.1 UI Assists Tools

7.2.3 GPT Customization

This section illustrates the process of customizing DataGPT using GPT builder. The following table shows the used prompts and the response by the GPT builder.

Prompt
I want to make a GPT expert in data science and artificial intelligence and all its sub-domains, with all the latest data science and artificial intelligence knowledge and information. Also, the main purpose of this GPT is to be an expert in data preprocessing methods.
The main purpose of this GPT is to assist with data cleaning, dealing with open data, discovering relationships between features, and describing datasets. Suggest solutions that will lead to better data quality and a better understanding of the data nature. Data enriching and diving into details will help the user, so one of your tasks is to ensure that happens. The GPT should only answer the queries that are related to the specified fields. If the user asks for something irrelevant, the answer should be polite and mention the domain the GPT covers.
You should use formal language when chatting with users.
You should ask clarifying questions instead of creating general-purpose responses.
You should avoid racism, and bias toward a specific group, race, political, or atheist beliefs. You should only focus on the things you are instructed to do.
When the user gives approval for the changes on the dataset, create a PDF technical report for the changes and how they affect the dataset. then generate the new dataset in the same format as the original (excel, csv,...)
When the user asks for info about the owner of this GPT, you should mention this custom GPT is done by CS students from King Faisal University, computer science department. As a partial requirement of the bachelor's degree. In case the user would like to reach us for suggestions inquiries, or issues provide the project’s email address: **datagpt2024@gmail.com**
Check the report template that I have uploaded named "Report Template" you should provide to each user after finishing the preprocessing task.
Note that you should follow the same structure as the template. if the user asks to add an additional section, make sure it follows the main structure. the report must be in PDF or Word extension to ensure the readability.
At the beginning of each new conversation with users, provide a short description of what this GPT is about and list the tasks that this GPT can do.
In order to avoid generating too many unwanted files, ask the user if there is anything he/she wants before generating the report. You can use some statements like "Is there anything else like me to do?" at the end of each of your responses.

Table 7.2.3.1 Prompt for building our custom GPTS

7.3 Users' credentials

Security is one of the most important aspects of software engineering nowadays and always has a high priority over other aspects. Building a security tool from scratch is not a preferred practice so, there comes the role of identity and access management solutions.

Auth0 is a flexible, drop-in solution to add authentication and authorization services to applications. teams and organizations can avoid the cost, time, and risk that come with building their own solution to authenticate and authorize users.[auth0 website]

Here are some challenges /use cases and Auth0 solutions which is beneficial and help reduce the threats and risks:

Challenge/ Use cases	Auth0 Solution
Adding user authentication and authorization to your application	Flexible drop-in solution, avoiding development costs and risks.
User login options	Username/password, social accounts (Facebook, Twitter).
User profile retrieval	Customize UI and apply authorization policies.
API Security	Secure your API with OAuth 2.0.
Single Sign-On (SSO)	Implement SSO across multiple applications.
Secure API access	Securely access APIs from JavaScript front-end and mobile apps.
Web app authentication with SAML	Utilize SAML for user authentication in web apps.
Passwordless login	Enable login with one-time codes via email or SMS.
Data breach protection	Receive notifications and manage compromised user accounts.
DDoS attack prevention	Block suspicious IP addresses with failed login attempts.
Enterprise Directory Service Federation	Allow employee login to internal/external apps using existing credentials.
User management solution	Focus on your app; Auth0 handles user management (reset, creation, provisioning, blocking, deletion).
Multi-factor Authentication (MFA)	Enforce MFA for accessing sensitive data.

Table 7.3.1 Auth0 use cases and solution

8 Preliminary outcomes/results

This section presents the preliminary outcomes of the project's implementation phase, highlighting initial findings and observations. It is important to acknowledge that, given the project's progressive development, some of these early results—particularly in areas such as UI/UX—may evolve significantly by Milestone 4. This evolution is a natural aspect of the iterative development process, allowing for refinement and improvement based on feedback and further analysis. A more comprehensive discussion will be provided in the next section “Analysis of the preliminary result through comparison, validation or verification”.

8.1 DataGPT Website

The DataGPT website is a user-friendly website that enables users to interact with the OpenAI customized GPT that we developed to assist them with data preprocessing. The website consists of multiple pages that will be detailed in the subsequent section.

The webpages are done successfully, and the only remaining part is developing the backend and services integration.

8.2 Customized GPT

The initialization is done smoothly and successfully, now our custom GPT can deal with various type of datasets. It prove its effectiveness by testing different datasets with different domians.

9 Analysis of the preliminary result through comparison, validation, or verification

This section will provide a deeper look into the details of the final results and discuss it. Then, we will validate the accomplished task with the requirements of the project.

9.1 Website Development

There is a diversity in web technology and many services that help creating well end websites.

For the front end, we have used HTML, CSS, and JS. For CSS we have used the Bootstrap framework which enables us to have well-organized pages. To have a clue about how the pages will look, first, we prototype the interfaces using Visly [19]. This website offers tools for creating user interface (UI) designs, especially focusing on wireframes and prototypes. It's designed to be easy to use, enabling teams of all sizes and skill levels to brainstorm and create attractive app wireframes.

Second, we have used auth0 to create signup/sign-in pages and it will be our user's authentication and authorization service and the integration will be completed in the back-end development phase. Auth0 [20] will handle critical security concerns such as MFA, and users' credentials.

Also, we decide to DigitalOcean [20]which is our option for cloud services. DigitalOcean's blend of intentional simplicity, affordability, and flexibility in its cloud services makes it an appealing option for individual developers, high-growth startups, and more generally ISVs and SMEs (small and medium enterprises) seeking a robust platform alternative from the hyperscalers.

9.2 Complete Use case

This use case is designed to demonstrate the capabilities of ChatGPT4 in preprocessing and analysis tasks. The ChatGPT proves its effectiveness in understanding datasets and identifying dataset issues such as outliers, missing and null values, and other data issues. Assessing data quality, discovering and simplifying relationships between features. Finally, it provides suggestions, recommendations, and aims of a dataset. The details will be explained in the following sections.

9.2.1 Features Analysis

In this section, we will only explore the data and features of this dataset, advanced preprocessing steps will be shown in the next sections.

The below table describes the features of the datasets:

Feature	Description
dt	Timestamp of the record
switch	Switch ID
src	Source IP address
dst	Destination IP address
pktcount	Packet count
bytecount	Byte count
dur	Duration in seconds
dur_nsec	Duration in nanoseconds
tot_dur	Total duration
flows	Number of flows
srcDevice	Source device identifier
dstDevice	Destination device identifier
rate	Rate
drop	Drop status (dropped packets)
spkts	Number of sent packets
dpkts	Number of dropped packets
pktrate	Packet rate
Pairflow	Pair flow identifier
Protocol	Protocol used (e.g., UDP, TCP)
port_no	Port number
tx_bytes	Transmitted bytes
rx_bytes	Received bytes
tx_kbps	Transmission rate in kilobits per second
rx_kbps	Reception rate in kilobits per second
tot_kbps	Total transmission rate in kilobits per second
label	Label (classification target)

Table 9.2.1.1 Features Analysis DDoS Dataset

As shown above, DataGPT describes the features even though we did not include the metadata with the dataset.

9.2.2 Statical Analysis

Feature	Count	Unique	Top	Frequency	Mean	Std Dev	Min	25%	50%	75%	Max
dt	**104345.0**	**-**	**-**	**-**	**17927.514169342085**	**11977.642654625488**	**2488.0**	**7098.0**	**11905.0**	**29952.0**	**42935.0**
switch	**104345.0**	**-**	**-**	**-**	**4.214260386218793**	**1.9563265690385538**	**1.0**	**3.0**	**4.0**	**5.0**	**10.0**
src	**104345**	**19**	**10.0.0.3**	**11491**	**-**	**-**	**-**	**-**	**-**	**-**	**-**
dst	**104345**	**18**	**10.0.0.7**	**18020**	**-**	**-**	**-**	**-**	**-**	**-**	**-**
pktcount	**104345.0**	**-**	**-**	**-**	**52860.95474627438**	**52023.241459901285**	**0.0**	**808.0**	**42828.0**	**94796.0**	**260006.0**
bytecount	**104345.0**	**-**	**-**	**-**	**38186596.6435766**	**48777482.4821864**	**0.0**	**79576.0**	**6471930.0**	**76203544.0**	**147128002.0**
dur	**104345.0**	**-**	**-**	**-**	**321.49739805453066**	**283.5182322732592**	**0.0**	**127.0**	**251.0**	**412.0**	**1881.0**
dur_nsec	**104345.0**	**-**	**-**	**-**	**461388039.67607456**	**277001898.1559739**	**0.0**	**234000000.0**	**418000000.0**	**703000000.0**	**999000000.0**
tot_dur	**104345.0**	**-**	**-**	**-**	**321886485437.7306**	**283402930204.56494**	**0.0**	**127000000000.0**	**252000000000.0**	**413000000000.0**	**1880000000000.0**
flows	**104345.0**	**-**	**-**	**-**	**5.654233552158704**	**2.9500364378025274**	**2.0**	**3.0**	**5.0**	**7.0**	**17.0**
packetins	**104345.0**	**-**	**-**	**-**	**5200.383468302267**	**5257.001450176361**	**4.0**	**1943.0**	**3024.0**	**7462.0**	**25224.0**
pktperflow	**104345.0**	**-**	**-**	**-**	**6381.715290622455**	**7404.777807924153**	**-130933.0**	**29.0**	**8305.0**	**10017.0**	**19190.0**
byteperflow	**104345.0**	**-**	**-**	**-**	**4716149.793281902**	**7560116.255042452**	**-14642594.0**	**2842.0**	**552168.0**	**9728112.0**	**14953872.0**
pktrate	**104345.0**	**-**	**-**	**-**	**212.2106761224783**	**246.85512346906947**	**-4365.0**	**0.0**	**276.0**	**333.0**	**639.0**
Pairflow	**104345.0**	**-**	**-**	**-**	**0.6009871100675643**	**0.48969776575268287**	**0.0**	**0.0**	**1.0**	**1.0**	**1.0**

Protocol	**104345**	**3**	**ICMP**	**41321**	**-**	**-**	**-**	**-**	**-**	**-**	**-**
port_no	**104345.0**	**-**	**-**	**-**	**2.331093967128276**	**1.08433291278482**	**1.0**	**1.0**	**2.0**	**3.0**	**5.0**
tx_bytes	**104345.0**	-	-	-	**93252643.2279266**	**151937991.69541538**	**2527.0**	**4743.0**	**4219610.0**	**135639774.0**	**1269981973.0**
rx_bytes	**104345.0**	**-**	**-**	**-**	**93280389.4836456**	**133000449.20233938**	**856.0**	**3539.0**	**13383386.0**	**143927708.0**	**990596183.0**
tx_kbps	**104345.0**	-	-	-	**998.8997556183813**	**2423.4716178146223**	**0.0**	**0.0**	**0.0**	**251.0**	**20580.0**
rx_kbps	**103839.0**	**-**	**-**	**-**	**1003.8114196014985**	**2054.887033976864**	**0.0**	**0.0**	**0.0**	**557.0**	**16577.0**
tot_kbps	**103839.0**	-	-	-	**2007.5787420911217**	**3144.4371725399496**	**0.0**	**0.0**	**4.0**	**3838.0**	**20580.0**
label	**104345.0**	**-**	**-**	**-**	**0.3908572523839187**	**0.48794481491435904**	**0.0**	**0.0**	**0.0**	**1.0**	**1.0**
Feature	**Count**	**Unique**	**Top**	**Frequency**	**Mean**	**Std Dev**	**Min**	**25%**	**50%**	**75%**	**Max**
dt	**104345.0**	**-**	**-**	**-**	**17927.514169342085**	**11977.642654625488**	**2488.0**	**7098.0**	**11905.0**	**29952.0**	**42935.0**
switch	**104345.0**	-	-	-	**4.214260386218793**	**1.9563265690385538**	**1.0**	**3.0**	**4.0**	**5.0**	**10.0**
src	**104345**	**19**	**10.0.0.3**	**11491**	**-**	**-**	**-**	**-**	**-**	**-**	**-**
dst	**104345**	**18**	**10.0.0.7**	**18020**	-	-	-	-	-	-	-
pktcount	**104345.0**	**-**	**-**	**-**	**52860.95474627438**	**52023.241459901285**	**0.0**	**808.0**	**42828.0**	**94796.0**	**260006.0**

Table 9.2.2.1 Statical Analysis DDoS Dataset

This table provides a comprehensive statistical analysis of a dataset with 104,345 records, summarizing key features related to network traffic data. Here is a brief overview of the columns:

1. **dt**: Number of data transfer records, with a mean of 17,927.51 and a standard deviation of 11,977.64.

2. **switch**: Indicates the switch number, averaging 4.21 with a standard deviation of 1.96.
3. **src**: Source IP addresses, with 19 unique values, the most frequent being 10.0.0.3 (11,491 occurrences).
4. **dst**: Destination IP addresses, with 18 unique values, the most frequent being 10.0.0.7 (18,020 occurrences).
5. **pktcount**: Packet count, with an average of 52,860.95 and a standard deviation of 52,023.24.
6. **bytecount**: Byte count, with a mean of 38,186,596.64 and a standard deviation of 48,777,482.48.
7. **dur**: Duration in seconds, averaging 321.50 with a standard deviation of 283.52.
8. **dur_nsec**: Duration in nanoseconds, with a mean of 461,388,039.68 and a standard deviation of 277,001,898.16.
9. **tot_dur**: Total duration, averaging 321,886,485,437.73 with a standard deviation of 283,402,930,204.56.
10. **flows**: Number of flows, with an average of 5.65 and a standard deviation of 2.95.
11. **packetins**: Number of packet-ins, with a mean of 5,200.38 and a standard deviation of 5,257.00.
12. **pktperflow**: Packets per flow, averaging 6,381.72 with a standard deviation of 7,404.78.
13. **byteperflow**: Bytes per flow, with a mean of 4,716,149.79 and a standard deviation of 7,560,116.26.
14. **pktrate**: Packet rate, averaging 212.21 with a standard deviation of 246.86.
15. **Pairflow**: Indicates whether the flow is paired, averaging 0.60 with a standard deviation of 0.49.
16. **Protocol**: Network protocol used, with 3 unique values, the most common being ICMP (41,321 occurrences).
17. **port_no**: Port number, averaging 2.33 with a standard deviation of 1.08.
18. **tx_bytes**: Transmitted bytes, with a mean of 93,252,643.23 and a standard deviation of 151,937,991.70.

19. **rx_bytes**: Received bytes, averaging 93,280,389.48 with a standard deviation of 133,000,449.20.
20. **tx_kbps**: Transmission rate in kbps, with a mean of 998.90 and a standard deviation of 2,423.47.
21. **rx_kbps**: Reception rate in kbps, averaging 1,003.81 with a standard deviation of 2,054.89.
22. **tot_kbps**: Total kbps, with a mean of 2,007.58 and a standard deviation of 3,144.44.
23. **label**: Label indicating classification, with a mean of 0.39 and a standard deviation of 0.49.

Each feature's statistical properties, such as count, unique values, top values, frequency, mean, standard deviation, and percentiles, provide a detailed understanding of the dataset's distribution and variability.

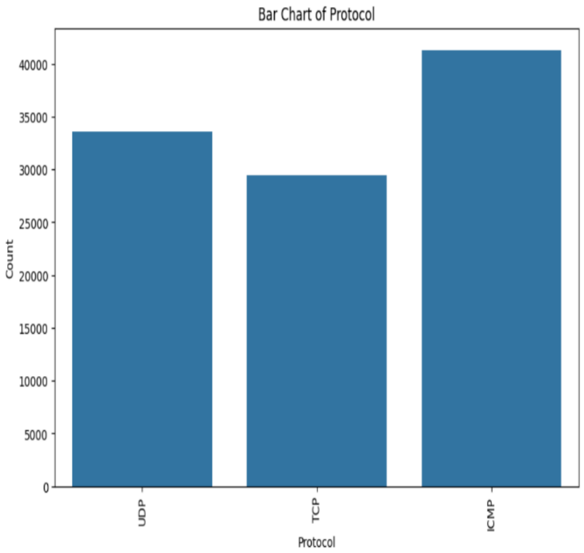


Figure 9.2.2.1 Bar Chart of Protocol

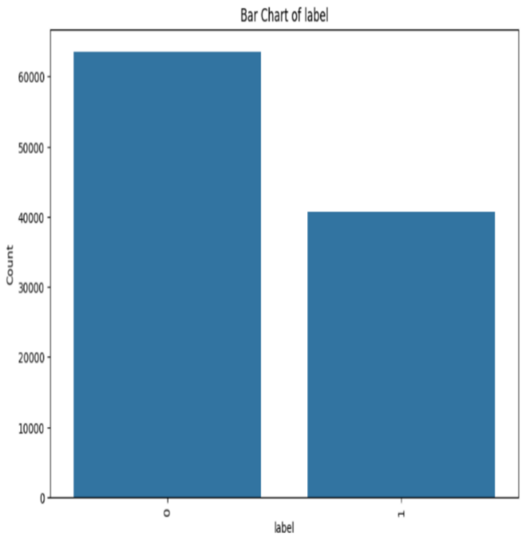


Figure 9.2.2.2 Bar Chart of Label

The bar chart of protocol represents the frequency of different network protocols within the dataset. The x-axis shows the three protocols: UDP, TCP, and ICMP, while the y-axis represents the count of occurrences for each protocol.

- **UDP**: Appears around 35,000 times.
- **TCP**: Appears around 30,000 times.
- **ICMP**: Appears around 40,000 times, making it the most frequent protocol in this dataset.

The bar chart of label represents the frequency of labels within the dataset. The x-axis shows the two labels: 0 and 1, while the y-axis represents the count of occurrences for each label.

- Label 0: Appears around 65,000 times.
- Label 1: Appears around 40,000 times.

Label 0 is more frequent than label 1 in this dataset.

9.2.3 Identify Dataset Issues and Suggested Actions

DataGPT is capable of identifying issues within datasets. The table below shows the identified issues and suggestions for actions to resolve them.

Issue	Description	Suggested Actions
Missing Values	Some features have missing values.	Impute missing values using statistical methods (mean, median, mode) or predictive modeling.
Outliers	Features like pktcount, bytecount, tx_bytes, and rx_bytes have extremely high maximum values.	Identify and treat outliers using z-score, IQR, or capping.
Negative Values	The pktrate feature contains negative values.	Investigate and correct negative values in the pktrate feature.
High Standard Deviation	Features like bytecount, pktcount, tx_bytes, rx_bytes, and tot_kbps show high variability.	Normalize or standardize these features to reduce the impact of outliers and improve algorithm performance.
Imbalanced Classes	The label feature is imbalanced, with one class more prevalent.	Use oversampling, undersampling, or synthetic data generation (SMOTE) to balance the classes.
Duplicate Rows	There are duplicate rows in the dataset.	Identify and remove duplicate rows to ensure data integrity.
Inconsistent Data Types	src and dst IP addresses are categorical but not processed as such.	Convert categorical data into appropriate formats (one-hot encoding or label encoding).
Anomalous Values	The pktrate feature has nonsensical negative values.	Investigate and correct anomalies or errors in the data recording.

Table 9.2.3.1 Dataset Issues and Suggested Actions

9.2.4 Relationship Between Features

Feature 1	Feature 2	Correlation Coefficient
dt	switch	0.05
dt	pktcount	0.01
dt	bytecount	0.02
dt	dur	-0.01
dt	tx_bytes	0.01
dt	rx_bytes	-0.01
switch	pktcount	0.07
switch	bytecount	0.06
switch	tx_bytes	0.05
switch	rx_bytes	0.06
pktcount	bytecount	0.98
pktcount	dur	0.42
pktcount	tx_bytes	0.70
pktcount	rx_bytes	0.72
bytecount	dur	0.42
bytecount	tx_bytes	0.70
bytecount	rx_bytes	0.72
dur	tx_bytes	0.28
dur	rx_bytes	0.30
tx_bytes	rx_bytes	0.95

Table 9.2.4.1 Relationship Between Features

1. **Strong Correlations**:
    - **pktcount** and **bytecount** have a very high positive correlation (0.98), indicating that as the number of packets increases, the byte count also increases proportionally.
    - **tx_bytes** and **rx_bytes** also have a strong positive correlation (0.95), suggesting a relationship between transmitted and received bytes.
2. **Moderate Correlations**:
    - **pktcount** and **dur** have a moderate positive correlation (0.42), indicating that longer durations tend to have more packets.
    - **pktcount** and **tx_bytes** (0.70), as well as **pktcount** and **rx_bytes** (0.72), show that packet counts are moderately related to both transmitted and received bytes.
3. **Weak Correlations**:
    - Most other features have weak or negligible correlations with each other, such as **dt** with most features, indicating little to no linear relationship.

9.2.5 Limitations and Recommendations

Limitation	Description	Recommendations
Missing Values	Some features contain missing values which can bias analyses and affect model performance.	Impute or remove missing values using statistical methods or predictive modeling.
Outliers	Outliers in features like pktcount, bytecount, tx_bytes, and rx_bytes can distort analyses.	Identify and treat outliers using z-score, IQR, or capping techniques.
Negative Values	pktrate feature contains unrealistic negative values indicating data collection errors.	Investigate and correct negative values in the pktrate feature.
Imbalanced Classes	The label feature is imbalanced, leading to potential bias in models.	Use oversampling, undersampling, or synthetic data generation (SMOTE) to balance the classes.
Duplicate Rows	Presence of duplicate rows can skew analysis and results.	Identify and remove duplicate rows to ensure data integrity.
High Variability	High standard deviation in features like bytecount, pktcount, tx_bytes, rx_bytes, and tot_kbps.	Normalize or standardize these features to reduce impact of outliers and improve model performance.
Inconsistent Data Types	Categorical data like src and dst IP addresses not processed as categorical features.	Convert categorical data into appropriate formats (one-hot encoding or label encoding).
Limited Feature Set	Lack of important features that could enhance predictive modeling.	Enrich dataset with additional contextual information and relevant features.
Temporal Component	Temporal dependencies and patterns are not explicitly modeled or analyzed.	Analyze temporal patterns to better understand trends and improve predictive models.
Anomalous Values	Features like pktrate have nonsensical values indicating potential data quality issues.	Investigate and correct anomalies or errors in the data recording.
Multicollinearity	High correlations between features like pktcount and bytecount, tx_bytes and rx_bytes.	Combine highly correlated features to reduce dimensionality and multicollinearity.
Lack of Context	Absence of contextual information about the network environment or data collection conditions.	Enrich the dataset with additional contextual information to improve robustness and relevance of analysis.

Figure 9.2.5.1 Limitations and Recommendations

10 Remarks on preliminary results and intermediate conclusions

Our experience with ChatGPT-4 to automate dataset cleansing has been both thrilling and educational. Our journey has uncovered hitherto unseen subtleties, and working with the data has been likened to solving a challenge. We examine the initial findings in this section and derive intermediate conclusions that act as road markers for our future work.

Chat GPT-4 works excellently at describing data and explaining the data in detail for various fields, and this allows the user to understand it excellently.

ChatGPT-4 cleans data relatively well and explains all the actions it performs in the data cleaning process, but it sometimes suggests operations to clean data that it cannot perform, and these may affect the result of the final file.

The good thing here is that the user can set special rules for the data cleaning process that match the user's desire.

ChatGPT-4 in data cleaning Sometimes the data cleaning process works in a way that may not be consistent with the desires of the users, such that sometimes normalization works in an undesirable way.

DataGPT tries to resolve the issue in ChatGPT-4 when it comes to some generation and not implementing some data preprocessing tasks.

11 Details of project implementation conforming to the project proposal

In this section, we detail the execution of our project, with a focus on how our implementation adhered to the plans and expectations set forth in the original project proposal. We outline the methods employed, the progress according to the planned timeline, the challenges faced, and the solutions devised to maintain alignment with our initial goals.

11.1 Project Objectives and Deliverables

As outlined in our initial proposal, the primary objective was to automate the data cleansing process using ChatGPT-4, enhancing the efficiency and accuracy of data modeling. We successfully implemented features that handle missing values, detect outliers, standardize data formats, and other data preprocessing solutions. These deliverables were completed as planned and conformed to the project's specifications.

11.2 Team Collaboration and Roles

As a group, we agreed to come up with a fine-ending project that meets the expectations. To ensure that we set up regular weekly meetings and use project management tools like Trello to organize our work keep team members aligned with each other and follow up the project update states. All the team members were involved in all tasks including website development, GPT customization, and many other tasks.

11.3 Mastery of tools and techniques used in project implementation

This section highlights our proficiency with the diverse tools and techniques deployed during the implementation of our project. We'll discuss how each was effectively utilized to meet our project objectives and efficiency throughout the development process.

11.3.1 Auth0

Auth0 was essential in our project for managing authentication and authorization. It helped us secure user access and integrate security features seamlessly into our application. Below, we explain how Auth0 improved our security infrastructure and made user management easier.

- We customized the login form and made it compatible with our system design.
- We enable users to sign up / sign in using their social accounts (google&microsoft).
- We have configured multifactor authentication (MFA), so the user can authenticate using email (OTP).
- Auth0 provides the users, database, and all needed information such as details, history, permission, and more which makes it easier to manage user data instead of creating it from scratch.
- We have added a custom action for email verification after signing up and it’s illustrated in the following diagram:

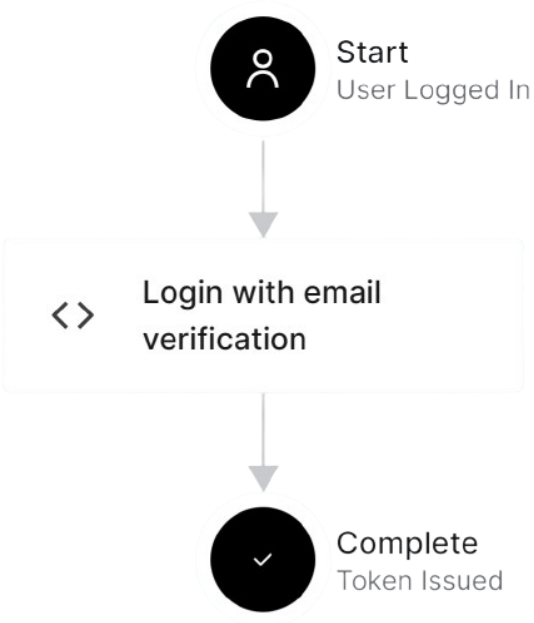
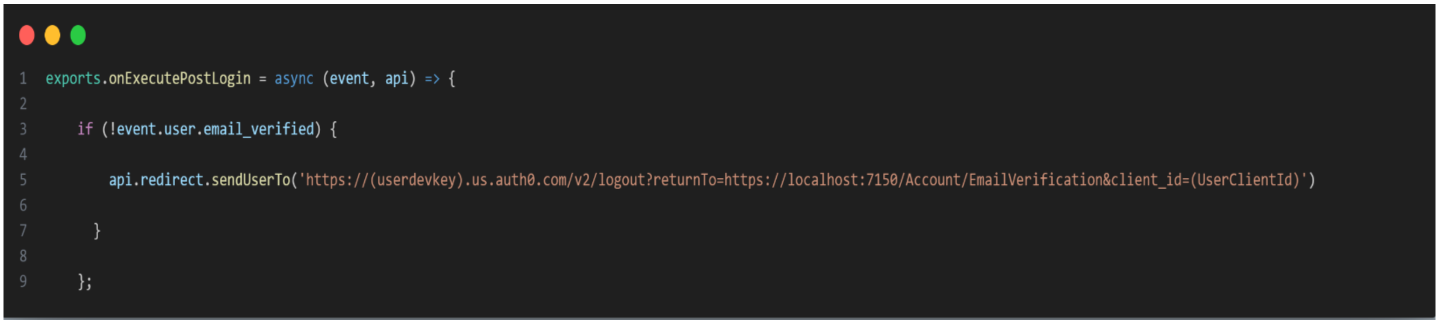


Figure 11.3.1.1 Auth0 Email Verification Diagram

Here is the code:



```
exports.onExecutePostLogin = async (event, api) => {

    if (!event.user.email_verified) {

        api.redirect.sendUserTo('https://(userdevkey).us.auth0.com/v2/logout?returnTo=https://localhost:7150/Account/EmailVerification&client_id=(UserClientId)')

      }

    };
```

Figure 11.3.1.2 Auth0 Email Verification Code

11.3.2 openAI API

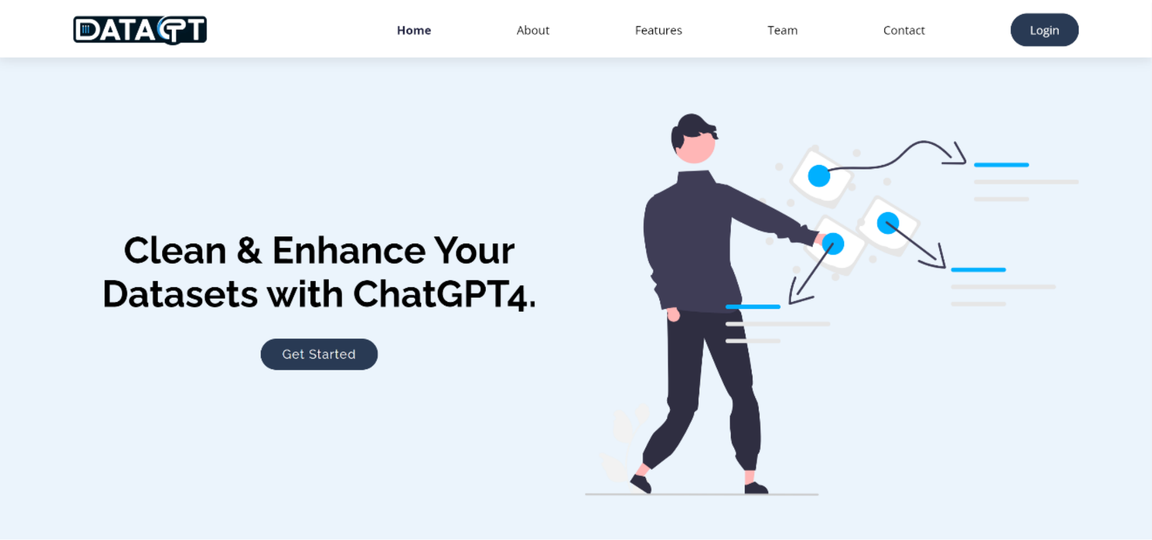
We utilized the OpenAI API to integrate ChatGPT into our system, enabling users to interact with ChatGPT directly through our website. This integration allows for seamless communication and enhances user engagement by providing real-time, AI-driven responses to inquiries.

12 Overall project outcome/achievements

Our project has successfully integrated ChatGPT-4. The data cleansing process enhances both the efficiency and accuracy of data preprocessing. This streamlined approach provides a user-friendly interface for interacting with the system. Below are the key outcomes and achievements:

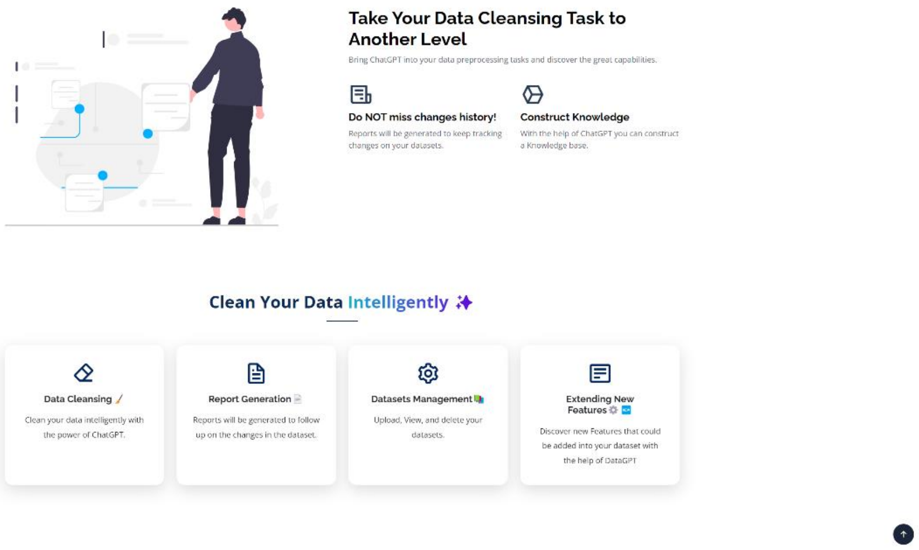
12.1 Landing Page

The landing page serves as the entry point for our application. It provides a clear and concise overview of the project, highlighting its core functionalities and the value it offers. Here, you'll gain a quick understanding of the project's main idea and its key features.



Clean & Enhance Your Datasets with ChatGPT4.

Figure 12.1.1 Landing Page



Take Your Data Cleansing Task to Another Level

Bring ChatGPT into your data preprocessing tasks and discover the great capabilities.

Do NOT miss changes history!

Reports will be generated to keep tracking changes on your datasets.

Construct Knowledge

With the help of ChatGPT you can construct a Knowledge base.

Clean Your Data Intelligently ✨

Data Cleansing

Clean your data intelligently with the power of ChatGPT.

Report Generation

Reports will be generated to follow up on the changes in the dataset.

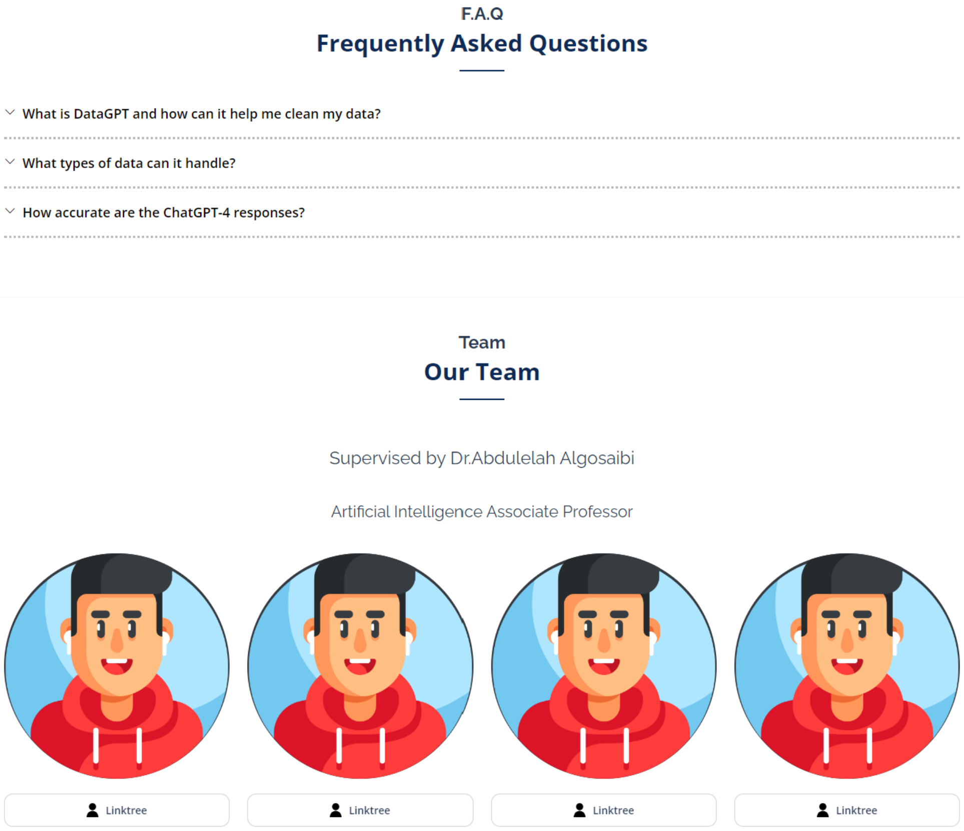
Datasets Management

Upload, View, and delete your datasets.

Extending New Features

Discover new Features that could be added into your dataset with the help of DataGPT

Figure 12.1.2 Landing Page



F.A.Q

Frequently Asked Questions

- What is DataGPT and how can it help me clean my data?
- What types of data can it handle?
- How accurate are the ChatGPT-4 responses?

Team

Our Team

Supervised by Dr.Abdulelah Algosaibi

Artificial Intelligence Associate Professor

Figure 12.1.3 Landing Page

12.2 Home Page

The central hub for interacting with the application. This streamlined interface provides easy access to the essential features and functionalities you need to get started.

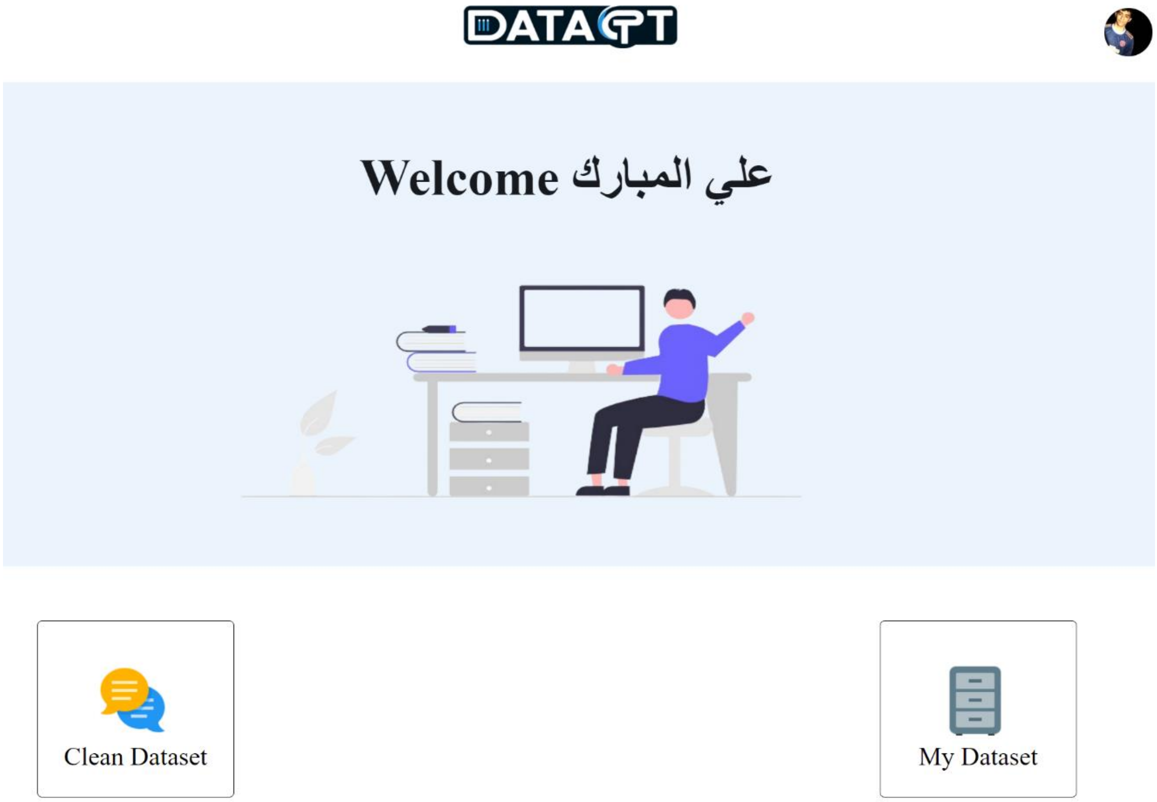


Figure 12.2.1 Home Page

12.3 My Datasets Page

My datasets page enables the user to effortlessly manage his/her datasets within the application. This interface provides a centralized platform for:

- Upload dataset: Upload your datasets quickly and easily for processing and analysis.
- Download dataset: Download your datasets.
- Delete dataset: Remove datasets you no longer need with just a few clicks, ensuring efficient data organization.

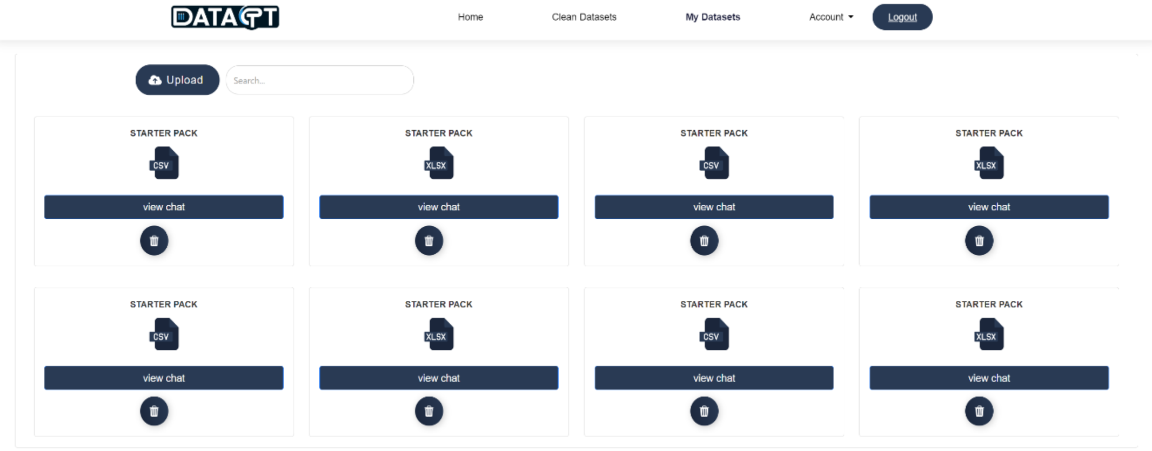
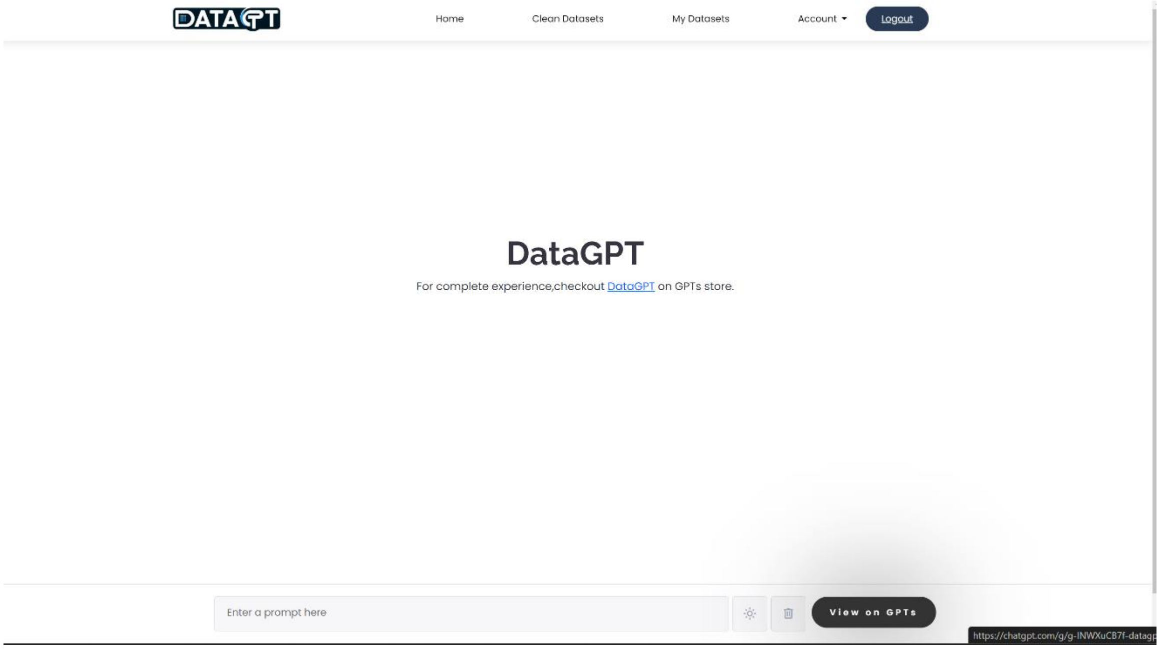


Figure 12.3.1 My Datasets Page

12.4 Data Cleansing Page

As OpenAI does not offer a custom GPTs API, we created a basic chat page allowing users to interact with ChatGPT-4. Additionally, there is a button that directs users to our custom GPT, DataGPT, available on the GPT store.



For complete experience,checkout DataGPT on GPTs store.

Figure 12.4.1 Data Cleansing Page

13 Analysis of overall results through comparison, validation, or verification

In this section, we perform a thorough analysis of the results obtained from the integration of ChatGPT-4 into our data cleansing processes, by comparing, validating, and verifying them against established standards.

13.1 Comparison with Traditional Methods

We begin by comparing the performance of our custom ChatGPT-4 with traditional data cleansing methods. This involves quantifying improvements in terms of time saved, error reduction, and overall data quality enhancements.

13.2 Validation through Use Cases

Validation is performed by applying the system to real-world datasets and observing its ability to correctly cleanse and preprocess data. Specific use cases, such as handling datasets with high volumes of missing values or erroneous entries, are presented along with detailed before-and-after comparisons to illustrate the system's capabilities.

14 Comprehensive remarks on overall project outcome and achievements

14.1 Conclusion

This project's primary objective was to automate the data cleansing phase by integrating ChatGPT-4, enhancing the efficiency, accuracy, and consistency of data preprocessing within data science workflows. We successfully implemented ChatGPT-4 to handle cleansing tasks like error identification and correction, missing value imputation, and data standardization. As a result, the quality of data cleansing improved, providing more reliable and accurate data for subsequent analytical processes. The development of a user-friendly interface has also enhanced the system's accessibility, enabling easy interaction with the cleansing process.

14.2 Future Work

As we continue to develop and enhance our project, we are excited about the possibilities for making our data cleansing tool even more effective and user-friendly. Here are the key areas we could plan to do it on for future improvements:

- **Adding New Features:** We could add new tools to the system, like features that can predict and correct errors automatically or provide real-time suggestions while cleaning data.

- **Handling Bigger Data Sets:** We could make our system robust enough to handle larger amounts of data, which would be beneficial for bigger organizations or more complex projects.

- **More Customization:** We aim to allow users to tailor the cleaning process to better fit their specific needs, depending on their industry or the particular types of data they work with.

- **Using Advanced Models:** We also may look into integrating newer and more advanced large language models (LLMs) that could enhance the system's ability to understand and process complex datasets.

References

[1] "Large Language Models as Data Preprocessors," [Online]. Available: https://arxiv.org/pdf/2308.16361.pdf.

[2] OpenAI, "OpenAI," [Online]. Available: https://openai.com/blog/introducing-gpts.

[3] "GPT (Generative Pre-trained Transformer) – A Comprehensive Review on Enabling Technologies, Potential Applications, Emerging Challenges, and Future Directions," [Online]. Available: https://arxiv.org/pdf/2305.10435.pdf.

[4] "Guide, Prompt Engineering," [Online]. Available: https://www.promptingguide.ai/.

[5] "google developers\machine learning," [Online]. Available: https://developers.google.com/machine-learning/resources/prompt-eng .

[6] "Towards Automated Data Cleaning Workflows," [Online]. Available: https://ceur-ws.org/Vol-2454/paper_8.pdf.

[7] "Leveraging ChatGPT API for Enhanced Data," [Online]. Available: https://teams.microsoft.com/l/message/19:7a683b35a246456fad6bed93cb4363c9@thread.v2/1701899615507?context=%7B%22contextType%22%3A%22chat%22%7D.

[8] H. H. a. E. S. Silva, "The Role of ChatGPT in Data Science: How AI-Assisted Conversational Interfaces Are Revolutionizing the Field," [Online]. Available: file:///C:/Users/Huawei/Downloads/BDCC-07-00062-with-cover%202.pdf.

[9] "Automated Data Preprocessing for Machine Learning Based Analyses," [Online]. Available: https://www.researchgate.net/publication/361026018_Automated_Data_Preprocessing_for_Machine_Learning_Based_Analyses.

[10] "Is a prompt and a few samples all you need? Using GPT-4 for data augmentation in low-resource classification tasks," [Online]. Available: https://arxiv.org/pdf/2304.13861.pdf.

[11] "OpenRefine," [Online]. Available: https://openrefine.org/.

[12] "Alteryx," [Online]. Available: https://www.alteryx.com/about-us/trifacta-is-now-alteryx-designer-cloud.

[13] "Winpure," [Online]. Available: https://winpure.com/.

[14] Tableau. [Online]. Available: https://www.tableau.com/products/prep.

[15] I. Sommerville, Software Engineering TENTH edition, PEARSON.

[16] F. CadY, The Data Science HandbooK.

[17] "The Role of ChatGPT in Data Science," [Online]. Available: https://www.mdpi.com/2504-2289/7/2/62.