Advanced Programming
The Final Project

Course Instructor: PHD Candidate , Maryam Babaei

June 6, 2024

Introduction

Welcome to your Advanced Programming course final project! This project is aimed at conducting an in-depth analysis of a given dataset and creating an application that provides comprehensive insights and visualizations. The objective is to analyze the dataset using various statistical and analytical techniques. You will implement a series of steps, including data retrieval, preprocessing, descriptive data analysis, and advanced statistical analysis. Additionally, you will develop an API to expose the analysis results and create an HTML representation of the API documentation for a user-friendly interface.
The application you will be building should enable users to retrieve, analyze, and visualize data effectively. Users will be able to access detailed statistical summaries, perform custom visualizations, and explore advanced analytical techniques through a well-structured API.
Throughout this document, we will delve into the various components and features of this data analysis and API application. It is our intention that this project serves as an opportunity for you to apply your advanced programming skills, gain hands-on experience with data analysis, API development, and HTML documentation, and showcase your ability to develop complex software systems.
Note that this project can be completed in groups of 2 or 3. Larger groups will have to implement additional features that are optional for smaller groups. A project mentor will be assigned to your team to offer guidance throughout the project's development.
Let's dive into the world of data analysis and API development and embark on an exciting journey of programming, innovation, and teamwork!

Analysis Section

Introduction

The Analysis Section is the foundational step in comprehending and deriving meaningful insights from a dataset. This section is designed to systematically explore the data, identify key patterns, and understand the relationships among various variables. By employing descriptive data analysis, we aim to uncover the underlying structure of the dataset, which sets the stage for more advanced analytical techniques. The primary objectives include data cleaning, identifying distributions, detecting anomalies, and visualizing the results to facilitate effective decision-making. This process is crucial for transforming raw data into actionable knowledge.

1. Explain the Main Subject and Main Goals for Analysis

Main Subject

The primary subject of this analysis is to investigate and understand the dataset provided. The analysis involves identifying patterns, anomalies, and correlations within the data to derive meaningful insights.

Main Goals

- **Identify Challenges and Goals:** Understand the specific challenges faced and define clear research goals for analyzing the data.
- **Key Variables and Distribution:** Identify key variables, their distributions, and how they interact with each other.
- **Descriptive Data Analysis:** Perform descriptive data analysis to understand the data's nature and internal patterns.
- **Data Preprocessing:** Determine necessary preprocessing steps to make the dataset suitable for analysis.
- **Hidden Patterns and Correlations:** Identify hidden patterns and correlations between variables.
- **Visualization:** Effectively visualize the extracted information to communicate findings.
- **Dataset Description:** Provide a detailed summary of the dataset including its sources, structure, and key attributes to inform decision-making.
- **Dataset Limitations:** Recognize the limitations of the dataset and their impact on conclusions.

2. Implementation of Descriptive Data Analysis

Visualization and Numerical Analysis

- **Visualization:** Utilize various graphical representations such as histograms, bar charts, scatter plots, and heat maps to visually explore the data.
- **Numerical Analysis:** Calculate statistical measures such as mean, median, mode, range, variance, and standard deviation to understand the distribution and central tendency of the data.

3. Data Preprocessing

Necessary Steps

- **Cleaning Data:** Remove or correct erroneous data entries.
- **Handling Missing Values:** Impute or discard missing values based on their significance and impact.
- **Normalizing/Standardizing Data:** Apply normalization or standardization to ensure data is on a comparable scale.
- **Transforming Variables:** Perform necessary transformations to address skewness or non-normal distributions.

4. Hidden Patterns and Correlations

Techniques

- **Correlation Matrices:** Use correlation matrices to identify linear relationships between variables.
- **Scatter Plots:** Visualize potential correlations and relationships between pairs of variables.
- **Advanced Statistical Tests:** Apply Chi-square tests, T-tests, or ANOVA to uncover deeper patterns and relationships (for students 1400).

5. Visualization of Findings

Effective Communication

- **Bar Charts:** Represent categorical data effectively.
- **Line Graphs:** Show trends over time.
- **Heat Maps:** Display correlations and interactions between variables.
- **Advanced Visualizations:** Use network graphs or other advanced techniques for complex data relationships.

6. Application of Findings to Decision-Making

Practical Implications

- **Actionable Decisions:** Translate data insights into practical decisions or policy recommendations.
- **Understanding Impact:** Consider the potential impact and feasibility of the proposed actions in real-world scenarios.

7. Limitations of the Dataset

Recognizing Limitations

- **Data Completeness:** Assess the completeness of the dataset and its representativeness.
- **Scope of Conclusions:** Understand the scope of conclusions that can be confidently drawn given the dataset's limitations.
- **Future Research:** Identify new problems or challenges that arise from the analysis, suggesting further investigation and study.

API Section

Introduction

The API Section focuses on the design and implementation of Application Programming Interfaces (APIs) that facilitate seamless interaction with the dataset and the analytical tools developed. APIs are crucial for enabling access to data and analytical functionalities, allowing users to retrieve, process, and analyze data programmatically. In this section, we outline the various endpoints available, each designed to perform specific tasks such as data retrieval, statistical summarization, visualization generation, and correlation analysis. The primary objective is to provide a user-friendly interface for developers and data analysts to interact with the dataset, ensuring efficient and effective data manipulation and analysis. By standardizing these interactions, the APIs enhance the accessibility and usability of the data and its associated insights.

General Tasks for Showing Results via APIs

1. **Data Retrieval**
   - **Endpoint:** `/api/data`
   - **Description:** Fetch the cleaned and processed dataset.
   - **Methods:** `GET`
   - **Response Format:** JSON
2. **Statistical Summary**
   - **Endpoint:** `/api/summary`
   - **Description:** Retrieve statistical summaries and descriptive analysis results.
   - **Methods:** `GET`

- **Response Format:** JSON
- **Features:**
  - Mean, median, mode
  - Variance, standard deviation
  - Range, quartiles

3. **Visualization**
   - **Endpoint:** `/api/visualization`
   - **Description:** Generate and retrieve visualizations such as charts and plots.
   - **Methods:** `GET`
   - **Query Parameters:**
     - `type` (e.g., histogram, bar_chart, scatter_plot)
     - `variables` (e.g., var1, var2 for scatter plot)
   - **Response Format:** Image/JSON (base64 encoded images)

4. **Correlation Analysis**
   - **Endpoint:** `/api/correlation`
   - **Description:** Perform and retrieve results of correlation analysis.
   - **Methods:** `GET`
   - **Response Format:** JSON
   - **Features:**
     - Correlation matrix
     - Scatter plots for variable pairs

5. **Advanced Statistical Tests**
   - **Endpoint:** `/api/stat_tests`
   - **Description:** Execute advanced statistical tests like Chi-square, T-tests, or ANOVA.
   - **Methods:** `POST`
   - **Request Format:** JSON
   - **Response Format:** JSON
   - **Features:**
     - Specify test type and variables
     - Return p-values, test statistics

6. **Preprocessing Steps**
   - **Endpoint:** `/api/preprocess`
   - **Description:** Execute and document preprocessing steps taken on the dataset.
   - **Methods:** `POST`
   - **Request Format:** JSON
   - **Response Format:** JSON
   - **Features:**
     - Clean data
     - Handle missing values
     - Normalize/standardize data
     - Transform variables

7. **Pattern Detection**
   - **Endpoint:** `/api/patterns`

- **Description:** Run algorithms for hidden pattern detection and correlation analysis.
- **Methods:** `GET`
- **Response Format:** JSON
- **Features:**
  - Identify clusters
  - Detect outliers
  - Discover frequent itemsets

8. **Insights and Decisions**
   - **Endpoint:** `/api/insights`
   - **Description:** Retrieve actionable insights and recommended decisions or policies.
   - **Methods:** `GET`
   - **Response Format:** JSON
   - **Features:**
     - Summarize key findings
     - Provide actionable recommendations
9. **Dataset Limitations**
   - **Endpoint:** `/api/limitations`
   - **Description:** Document and explain the dataset's limitations and their impact on the analysis.
   - **Methods:** `GET`
   - **Response Format:** JSON
   - **Features:**
     - Data completeness
     - Representativeness
     - Timeliness
10. **Future Research Suggestions**
    - **Endpoint:** `/api/future_research`
    - **Description:** Suggest further research areas based on current findings.
    - **Methods:** `GET`
    - **Response Format:** JSON
    - **Features:**
      - Identify new research questions
      - Highlight areas needing deeper investigation

HTML Representation

Goal

The goal of this section is to create an HTML representation of the API documentation. This will provide a user-friendly, web-based interface for accessing and understanding the various endpoints and functionalities offered by the APIs.

Requirements

HTML Structure

- Use standard HTML5 elements to structure the document.
- Include appropriate headings and paragraphs for clear organization.

Content

- Clearly describe each API endpoint, including the URL, HTTP method, and expected response format.
- Provide descriptions and key features of each API.

Formatting

- Use lists to itemize features and query parameters.
- Ensure the content is easy to read and navigate.

Metadata

- Include a `<title>` tag to set the document's title.
- Ensure proper `<head>` and `<body>` sections.

Technical Details

- Include a `DOCTYPE` declaration for HTML5.
- Ensure all elements are properly nested and closed.

Useful Tools and Packages

For the implementation of the project using Python and FastAPI, along with the analysis of data, the following tools and packages will be utilized:

- **Python:** A versatile programming language widely used for web development, data analysis, machine learning, and more.
- **FastAPI:** A modern, fast (high-performance) web framework for building APIs with Python 3.7+ based on standard Python type hints.
- **NumPy:** A fundamental package for scientific computing with Python, providing support for large, multi-dimensional arrays and matrices, along with a collection of mathematical functions to operate on these arrays.
- **Pandas:** A powerful and easy-to-use open-source data analysis and manipulation tool built on top of the Python programming language. It offers data structures and operations for manipulating numerical tables and time series.
- **Matplotlib:** A comprehensive library for creating static, animated, and interactive visualizations in Python. It is widely used for generating plots, charts, histograms, and other graphical representations of data.
- **Seaborn:** A data visualization library based on Matplotlib that provides a high-level interface for drawing attractive and informative statistical graphics.
- **Scikit-learn:** A simple and efficient tool for predictive data analysis, built on NumPy, SciPy, and Matplotlib. It provides various machine learning algorithms and tools for data mining and data analysis.

These tools and packages will be instrumental in the successful implementation of the project, enabling efficient development, data analysis, and visualization.

Evaluation

The evaluation criteria for the final project are as follows:

- **Code Compilation and Execution:** Your code should compile and run without any errors. It should be tested thoroughly to ensure all functionalities work as expected.
- **Code Quality:** Your code should be well-organized, readable, properly commented, and follow clean code principles. Use appropriate Python naming conventions for classes, methods, and variables.
- **Presentation:** You will be required to present your code to a team of judges for the final evaluation. Ensure all team members have sufficient mastery over the project to effectively present and discuss the implementation details.

Adhering to these evaluation criteria will contribute to the successful completion and assessment of your final project.

Submission

To submit your project, follow these steps:

1. **Push to GitHub:** Push the project's code to its repository on GitHub, ensuring that all changes are uploaded.
2. **README File:** Complete the README file with relevant information about your project, including setup instructions, usage guidelines, and any additional details.
3. **Merge Branches:** Merge your development branches with the main branch of the project to consolidate all changes.

The deadline for submitting your project is Friday, July 5 (15th of Tir). You must be prepared to present your code on Saturday, July 6 (16th of Tir).

Good luck and happy coding!