

# AI Data Analytics Agent Documentation

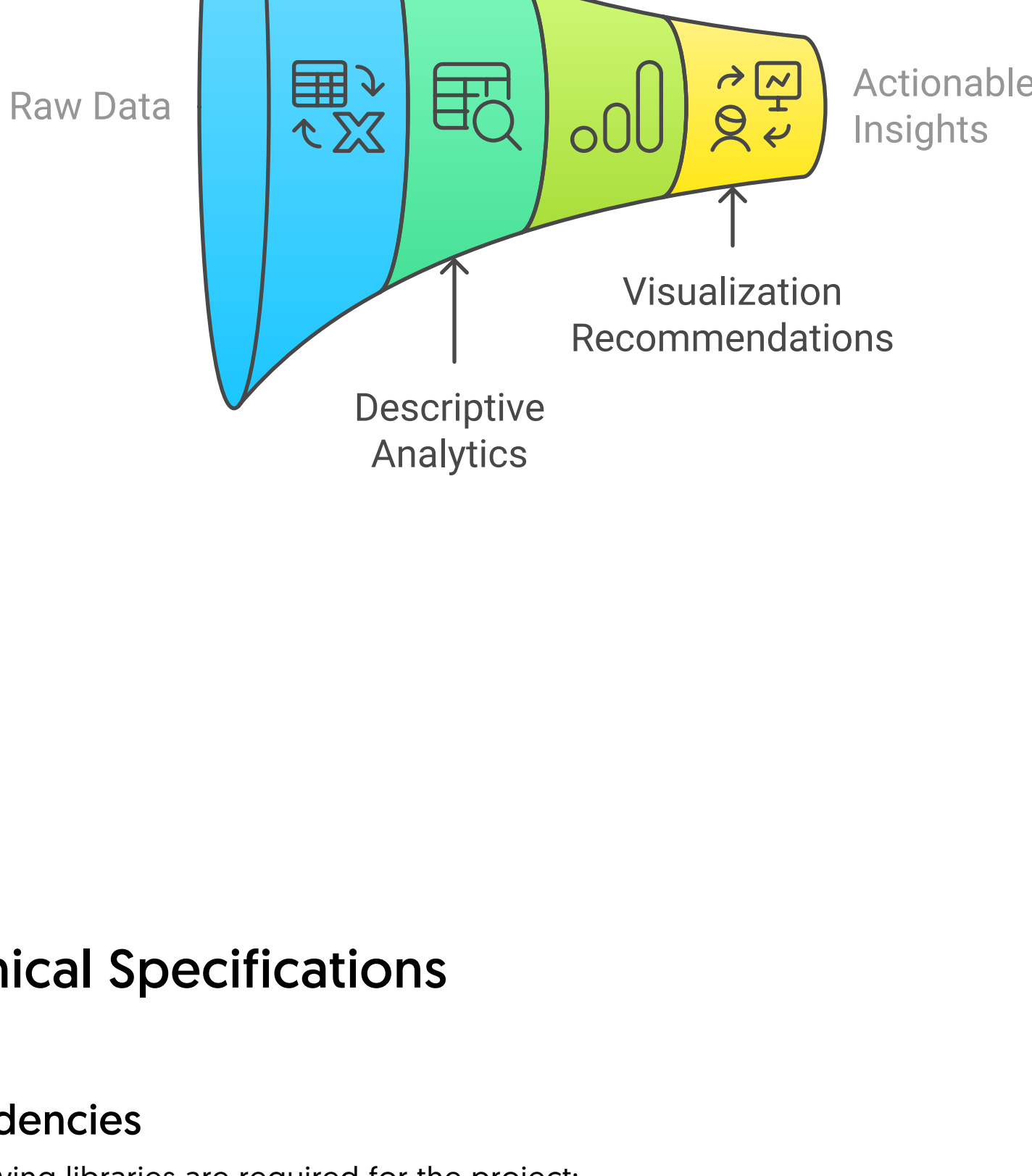
## Abstract

This document provides a comprehensive overview of the AI Data Analytics Agent, a Python application that utilizes the Ollama language model for dataset analysis. Designed to operate on a MacBook Pro with an M3 chip, this tool offers functionalities such as descriptive and predictive analytics, data cleaning suggestions, and visualization recommendations. The following sections detail the project's technical specifications, code structure, function descriptions, user interaction, and instructions for running the application.

## Project Overview

The AI Data Analytics Agent is a Python application that leverages the Ollama language model to analyze datasets and provide insights. This tool is designed to run locally on a MacBook Pro with the M3 chip, utilizing the capabilities of Ollama to perform various analytical tasks, including descriptive analytics, predictive analytics, data cleaning suggestions, and visualization recommendations.

### Data Processing Funnel



## Technical Specifications

### Dependencies

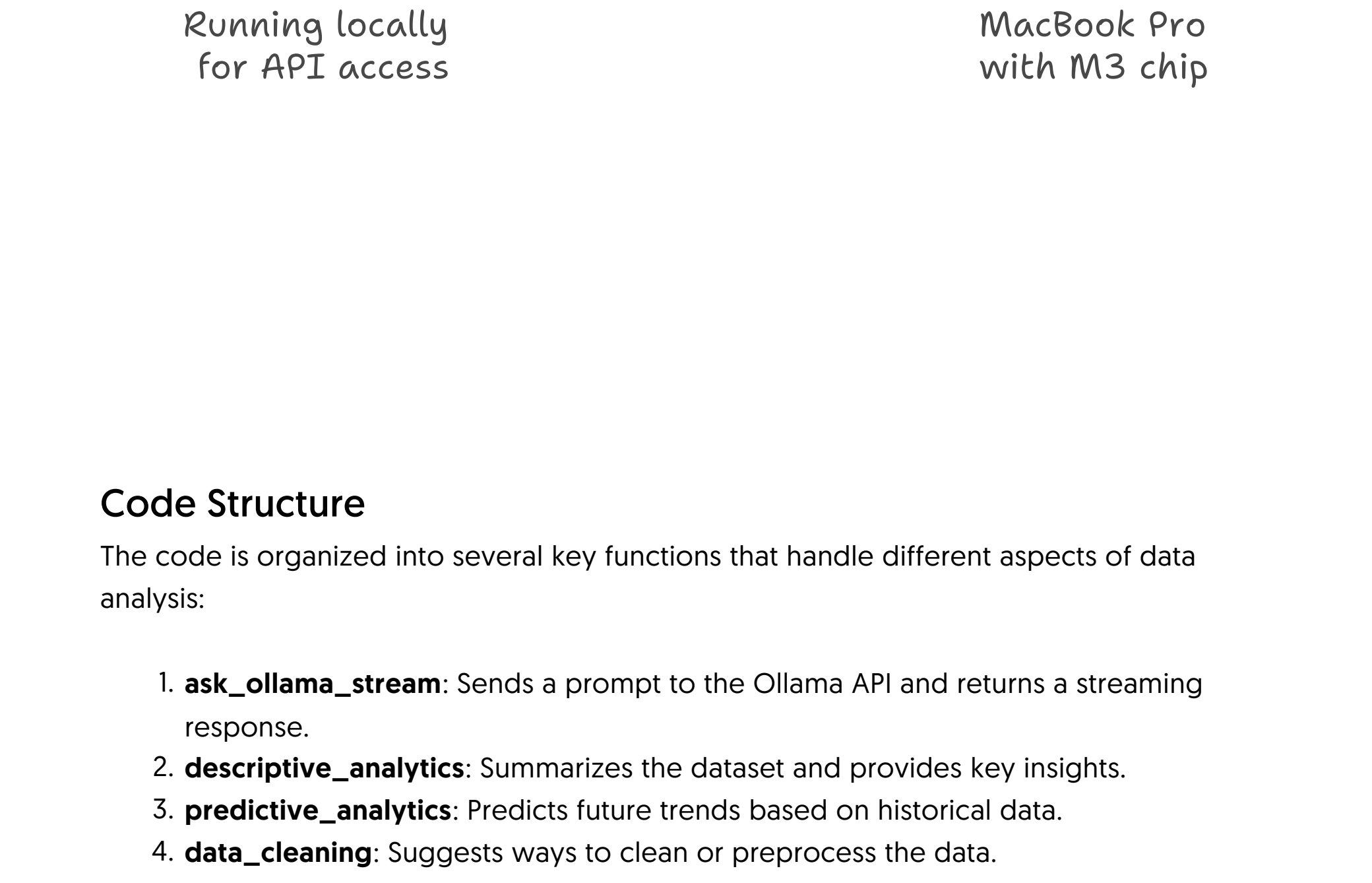
The following libraries are required for the project:

- ollama**: For interacting with the Ollama language model.
- pandas**: For data manipulation and analysis.
- requests**: For making HTTP requests to the Ollama API.
- json**: For handling JSON data.

### Environment Setup

- Platform**: MacBook Pro with M3 chip
- Ollama server**: Running on <http://localhost:11434>
- Python version**: 3.x
- Dataset**: A CSV file containing the data to be analyzed.

## Project Setup Overview

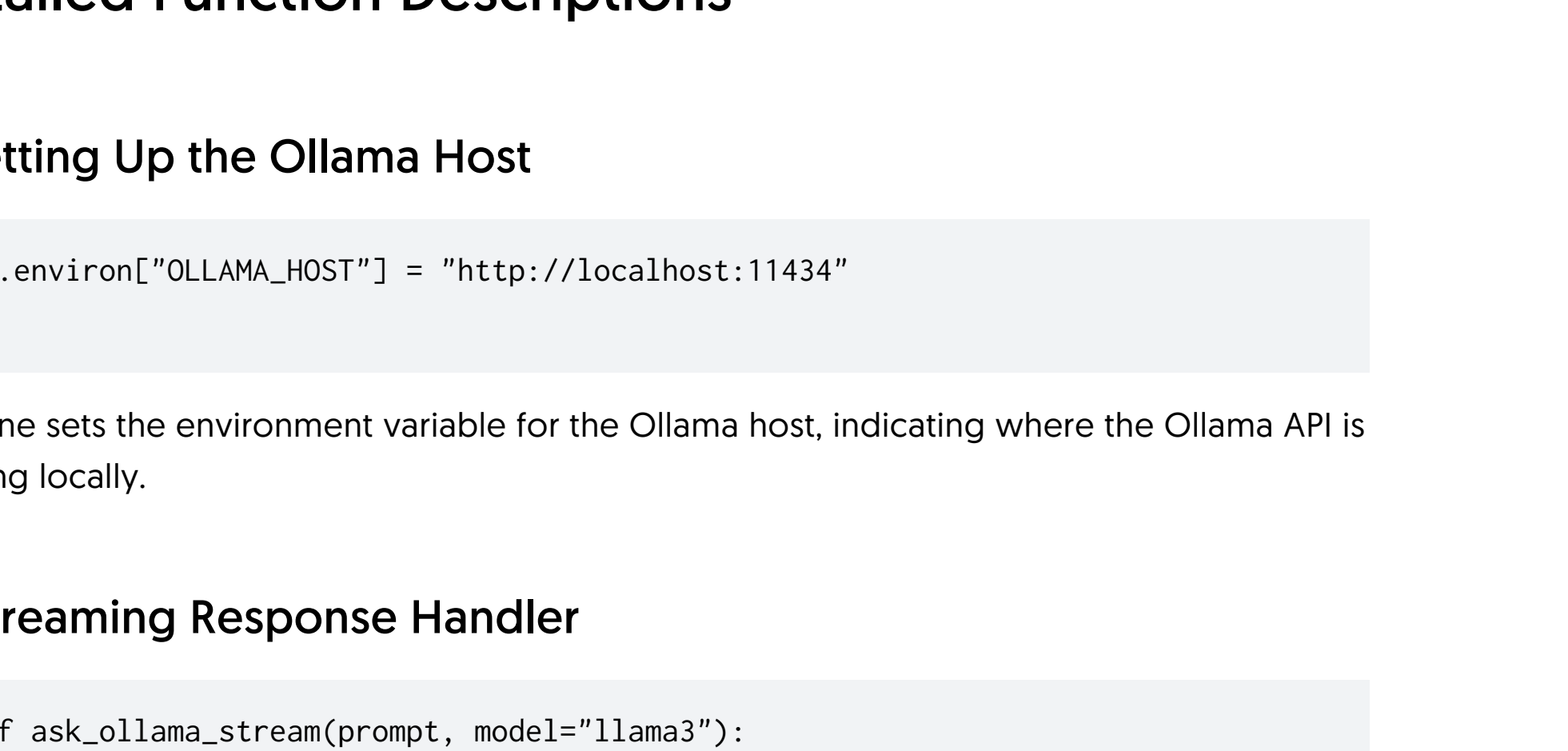


## Code Structure

The code is organized into several key functions that handle different aspects of data analysis:

- ask\_ollama\_stream**: Sends a prompt to the Ollama API and returns a streaming response.
- descriptive\_analytics**: Summarizes the dataset and provides key insights.
- predictive\_analytics**: Predicts future trends based on historical data.
- data\_cleaning**: Suggests ways to clean or preprocess the data.
- visualization\_suggestions**: Recommends the best ways to visualize the data.
- custom\_query**: Allows users to ask custom questions about the dataset.
- main**: The entry point of the application that presents a menu for user interaction.

### Data Analysis Code Structure



## Detailed Function Descriptions

### 1. Setting Up the Ollama Host

```
os.environ["OLLAMA_HOST"] = "http://localhost:11434"
```

This line sets the environment variable for the Ollama host, indicating where the Ollama API is running locally.

### 2. Streaming Response Handler

```
def ask_ollama_stream(prompt, model="llama3"):
    """Sends a question to Ollama and returns streaming response."""
```

#### Parameters:

- prompt**: The input question or prompt to be sent to the Ollama model.
- model**: The specific model to use (default is "llama3").

- Returns**: The accumulated response from the model as a string.

### 3. Data Loading

```
data =
pd.read_csv("/Users/somesh/Downloads/annual-enterprise-survey-2023-financial-yea
r-provisional-size-bands.csv")
```

This line loads the dataset from a specified CSV file into a pandas DataFrame for analysis.

### 4. Analytical Functions

#### Descriptive Analytics

```
def descriptive_analytics(data):
    """Asks Ollama to summarize the dataset and provide key insights."""
```

- Functionality**: Sends a prompt to Ollama to summarize the dataset and extract key insights.

#### Predictive Analytics

```
def predictive_analytics(data):
    """Asks Ollama to predict future trends based on historical data."""
```

- Functionality**: Sends a prompt to Ollama to analyze historical data and predict future trends.

#### Data Cleaning Suggestions

```
def data_cleaning(data):
    """Asks Ollama to suggest ways to clean or preprocess the data."""
```

- Functionality**: Sends a prompt to Ollama to provide recommendations for cleaning or preprocessing the dataset.

#### Visualization Suggestions

```
def visualization_suggestions(data):
    """Asks Ollama to recommend the best way to visualize the data."""
```

- Functionality**: Sends a prompt to Ollama to suggest effective visualization techniques for the dataset.

#### Custom Query Handler

```
def custom_query(data):
    """Allows the user to ask custom questions about the dataset."""
```

- Functionality**: Provides an interactive loop for users to ask custom questions about the dataset, sending each question to Ollama for a response.



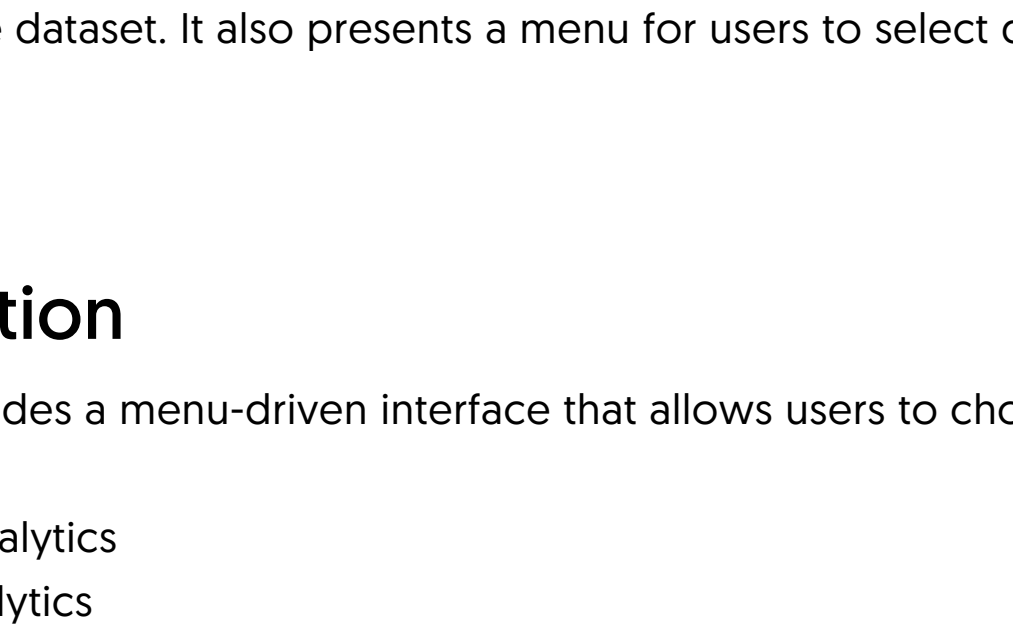
### Overview of Analytical Functions



## 5. Main Function

```
def main():
    print("Welcome to the AI Data Analytics Agent!")
    print("Dataset Preview:")
    print(data.head())
```

### AI Data Analytics Agent Execution Sequence



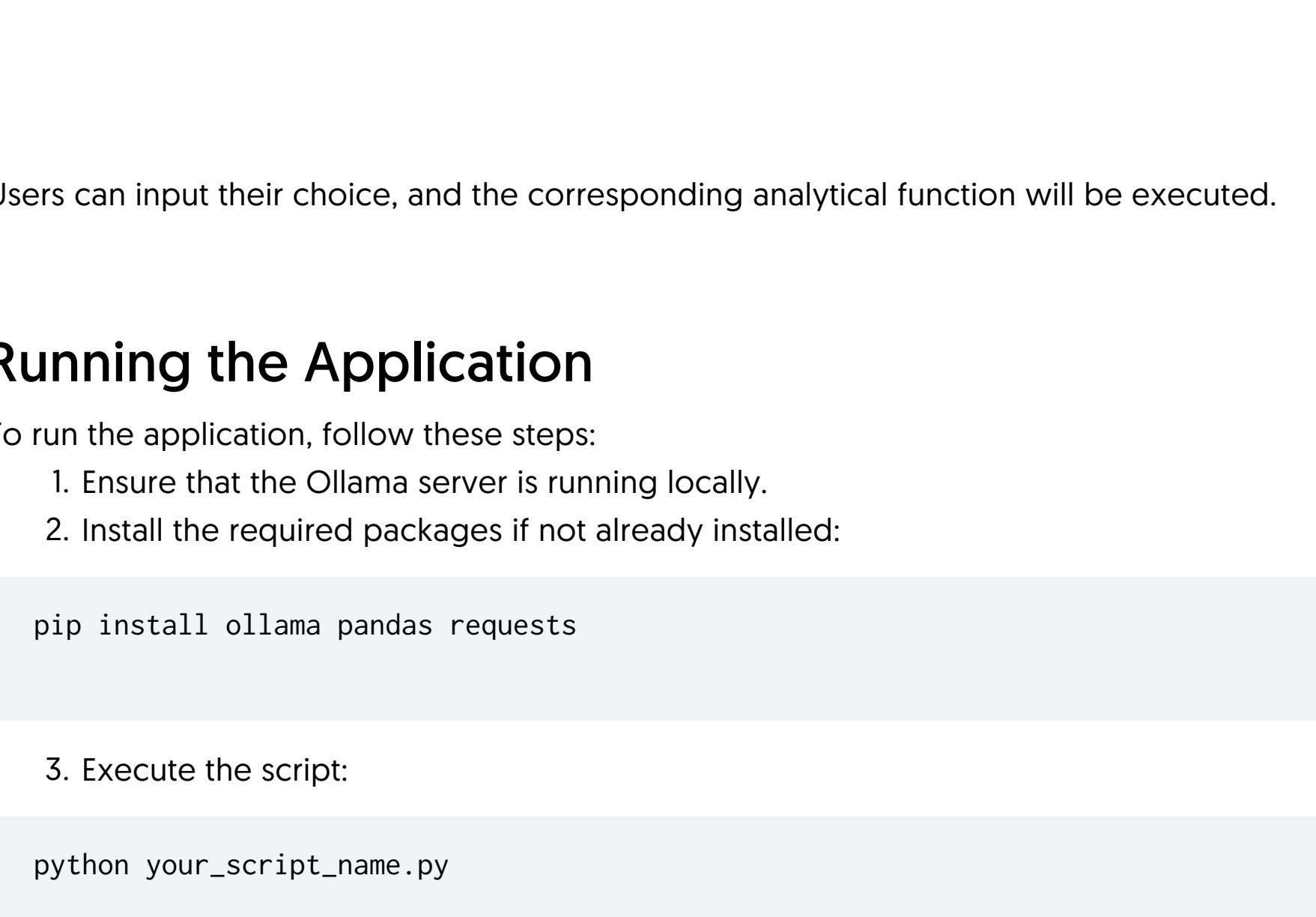
- Functionality**: Initializes the application, displays a welcome message, and shows a preview of the dataset. It also presents a menu for users to select different analytical options.

## User Interaction

The application provides a menu-driven interface that allows users to choose from the following options:

- Descriptive Analytics
- Predictive Analytics
- Data Cleaning Suggestions
- Visualization Suggestions
- Custom Query
- Exit

### Unified Analytical Interface



Users can input their choice, and the corresponding analytical function will be executed.

## Running the Application

To run the application, follow these steps:

- Ensure that the Ollama server is running locally.
- Install the required packages if not already installed:

```
pip install ollama pandas requests
```

- Execute the script:

```
python your_script_name.py
```

## Steps to Run the Application

### Execute Script

Run the specified Python script to launch the application.

### Install Packages

Install necessary packages using pip to prepare the environment.

### Start Ollama Server

Ensure the Ollama server is operational on your local machine.

## Conclusion

The AI Data Analytics Agent is a powerful tool for analyzing datasets using the Ollama language model. It is designed to run locally on a MacBook Pro with the M3 chip, providing users with real-time insights and recommendations. The application is flexible and can be extended with additional features or analytical capabilities as needed.