DataAnalyzer Technical Documentation

Table of Contents

---

1. Introduction

DataAnalyzer is a powerful tool designed for data scientists and analysts to efficiently process and visualize large datasets. This dynamic software provides a robust set of features, including data cleaning, statistical analysis, and customizable visualizations, making it an essential component for any data-driven organization.

2. System Requirements

Hardware Requirements

- **CPU:** 2 GHz dual-core processor or higher
- **RAM:** Minimum 8 GB (16 GB recommended)
- **Disk Space:** At least 500 MB of free disk space

Software Requirements

- **Operating System:** Windows 10 or higher, macOS Mojave or higher, or a compatible Linux distribution
- **Python:** Version 3.8 or higher
- **Node.js:** Version 14 or higher
- **Database:** PostgreSQL 10 or higher

3. Installation

Step 1: Clone the Repository

```
git clone https://github.com/example/data-analyzer.git
cd data-analyzer
```

Step 2: Install Backend Dependencies

```
cd backend
pip install -r requirements.txt
```

Step 3: Set Up the Database

- Create a PostgreSQL database.

- Update the database connection settings in `config.py`.

Step 4: Run the Backend Server

```
python app.py
```

Step 5: Install Frontend Dependencies

```
cd frontend
npm install
```

Step 6: Run the Frontend Application

```
ng serve
```

4. Configuration

Environment Variables

- Set up the following environment variables in your `.env` file:
  - `DATABASE_URL`: Database connection string
  - `SECRET_KEY`: A secure random key for session management

Configuration File

- Modify `config.py` to adjust settings like the default data format, API keys, and logging levels.

5. Usage

Importing Data

1. Navigate to the "Data Import" section.
2. Select the file format (CSV, Excel, JSON) and upload your dataset.

Cleaning Data

- Access the "Data Cleaning" tab to handle missing values, duplicates, and outliers.
- Choose the appropriate cleaning methods and apply them to your dataset.

Analyzing Data

- Use the "Analysis" feature to select statistical tests and methods.
- Configure parameters and run the analysis to view results.

Visualizing Data

- Go to the "Visualization" section to create interactive charts.
- Choose the data and visualization type (e.g., bar chart, line graph) to generate reports.

6. API Reference

Authentication

- **POST /api/auth/login**
  - **Description:** Log in to the application.
  - **Request Body:** `{ "username": "string", "password": "string" }`
  - **Response:** `{ "token": "JWT token" }`

Data Operations

- **POST /api/data/import**
  - **Description:** Import data into the application.
  - **Request Body:** `{ "file": "file data" }`
  - **Response:** `{ "status": "success", "message": "Data imported successfully." }`

Analysis

- **POST /api/data/analyze**
  - **Description:** Run statistical analysis.
  - **Request Body:** `{ "analysisType": "string", "parameters": {...} }`
  - **Response:** `{ "results": {...} }`

7. Troubleshooting

Common Issues

- **Unable to connect to the database**
  - Check your database credentials in `config.py`.
  - Ensure PostgreSQL is running.
- **Errors during data import**
  - Verify the file format and ensure there are no corrupt files.

Debugging Tips

- Enable debugging mode in `config.py` to view detailed error logs.
- Check the console for any runtime errors.

8. Frequently Asked Questions (FAQs)

Q: What types of data can DataAnalyzer handle?
A: DataAnalyzer supports CSV, Excel, JSON, and SQL databases.

Q: Is there a limit to the size of datasets?
A: While there is no strict limit, performance may vary based on system resources.

Q: Can I integrate DataAnalyzer with other tools?
A: Yes, DataAnalyzer provides APIs for integration with external applications.