Yann GUEGUEN

2024 - 2025

Food Tracking project

Contents

1 Project Overview

This personal project was initiated during my data engineering apprenticeship to test and showcase my skills in both **data engineering** and **data science**. The goal is to build a complete **data pipeline**, from data ingestion to visualization, while integrating modern industry practices such as automation, orchestration, and deployment.

The project is a full-fledged data processing and analysis pipeline designed to simulate food tracking. It leverages **fake data** generated by a sensor data simulation application, which mimics user interactions from a fictional food tracking app. The simulated data is exposed through an **API hosted on Render**, allowing seamless retrieval and integration into the data pipeline.

The project processes these user-generated datasets through an **ETL** (Extract, Transform, Load) workflow, automating data ingestion, cleaning, and transformation. To ensure efficient workflow execution, it incorporates orchestration with **Apache Airflow**, enabling automated scheduling and task management.

The data pipeline follows an **ETL** (**Extract, Transform, Load**) workflow:

1. **Data Extraction**: The API is queried to fetch the latest sensor data.
2. **Transformation & Cleaning**: The dataset undergoes preprocessing, validation, and aggregation.
3. **Storage**: The cleaned data is stored in an **Amazon S3** bucket.
4. **Orchestration**: **Apache Airflow** automates task scheduling and dependencies.
5. **Machine Learning & AI**: Predictive models and analytical algorithms are applied to detect trends and extract insights from the processed data.
6. **Visualization**: An interactive **Streamlit** dashboard is used for real-time analysis.

To maintain high standards of data integrity and code reliability, a **CI/CD pipeline** has been implemented. It automates testing and deployment processes, ensuring that every transformation preserves data quality. **Unit tests** are executed after each transformation to validate the correctness of the processed data. Additionally, each code change is verified through automated tests before being merged on **GitHub**, preventing regressions and maintaining robust implementation standards.

For deployment, **Docker is planned but not yet implemented**. The objective is to create a reproducible and scalable environment, leveraging containerization to facilitate deployment.

For data visualization, I have developed an interactive **Streamlit** web application. The application successfully renders meaningful visualizations aligned with the data. However, deploying the app has proven challenging due to AWS credentials management. Since the application fetches data directly from an **Amazon S3**, I am currently facing deployment issues related to authentication and secure access to cloud storage.

Moving forward, my goal is to refine the deployment process while maintaining data security and accessibility, ensuring a robust and user-friendly platform for data exploration.

1.1 Tech Stack

The project integrates multiple tools and libraries to streamline data processing and visualization:

- **Backend & API**: FastAPI (planned), Render (API hosting)
- **Data Processing**: **Pandas**, **DuckDB**, **PyArrow**
- **Orchestration**: **Apache Airflow**, **boto3** (AWS SDK)
- **Storage**: **Amazon S3**
- **Visualization**: **Streamlit**, **Plotly**, **Seaborn**, **Matplotlib**
- **Machine Learning** (planned): **Scikit-learn**, **Scipy**
- **CI/CD & Deployment**: **GitHub Actions**, **pytest**, **Docker** (planned)

1.2 Project Structure

The project follows a structured and modular architecture to ensure scalability and maintainability. Below is an overview of the key directories:

```
IM/
  .github/              # CI/CD workflows and automation scripts
  AI/                   # Machine learning and AI models
  aws_s3/               # Scripts for AWS S3 data handling
  data/                 # Raw and processed datasets
  data_engineering/      # Core ETL pipeline and data processing scripts
  DB/                   # Database configurations and management scripts
  env/                  # Virtual environment (ignored in version control)
  model_and_weight/     # Pre-trained model weights and configurations
  streamlit_app/        # Streamlit-based data visualization application
  web_site/             # Web-based UI for interacting with the project
  .dockerignore         # Files excluded from Docker images
  .env                  # Environment variables (ignored in version control)
  .gitignore            # Git ignored files
  requirements.txt      # Project dependencies
  README.md             # Project documentation
  project_structure.py  # Script to generate project structure
  list_s3_dataframes.py # Utility script to list S3 data
```

This structure ensures a clear separation of concerns between data processing, storage, AI models, and visualization components.

2 Sensor API: Generating Simulated Food Tracking Data

2.1 Overview

Figure 2: *Hier ist eine Bildunterschrift, die das Bild beschreibt. Dabei sollte das Bild und die Unterschrift für sich stehen können. Ein Wort wäre zu wenig.*

Figure 1: *mmm*

Ein zentrales Gestaltungselement ist eine Tabelle, in der wichtige Ergebnisse übersichtlich zusammengetragen werden können, vgl. *Tabelle 1*. Es ist auf die Tabellenüberschrift im Gegensatz zu einer Abbildungsunterschrift zu achten.

Table 1: *Tabellenüberschrift.*

	A	B
Y	XXX	XXX
Z	XXX	XXX

Grundlegend sind auch Listen[1] für einen klaren Überblick:

- erster Punkt
- zweiter Punkt
- dritter Punkt

[1] Es gibt auch nummerierte Listen.

2.2 Web Scrapping

2.2.1 Reaktionsgleichungen

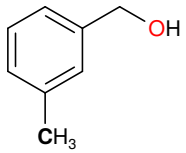
An dieser Stelle kann eine Reaktionsgleichung eingefügt werden. Dies erfolgt aber in einer eigenen Umgebung, die wieder mit `begin{equation}` und `end{equation}` "eingeklammert" wird. Dieser Baustein befindet sich in Overleaf.com in der mitgelieferten Datei: `bausteine.tex`.

$$\mathrm{A} + \mathrm{B} \longrightarrow \mathrm{AB} \tag{1}$$

Wenn nun auf die Reaktionsgleichung (s. *Equation 1*) Bezug genommen werden soll, muss hier auf das Label mit `\autoref{label}` referenziert werden.

2.2.2 Strukturformeln

Wie es für die Naturwissenschaften üblich ist, kommt man um Strukturformeln nicht drum rum. Entweder man bindet die Grafik als PNG oder PDF ein oder man erstellt sich mit `\chemfig{..}` selber eine:



(2)

2.2.3 Diagramme

Die Krönung in einem wissenschaftlichen Dokument ist natürlich das selbsterstellte Diagramm. Die Daten aus *Abbildung 3* stammen aus einem Experiment zur Extraktion von Chlorophyll und anschließender Spektroskopie mit dem PASCO-Spektrometer. Die Daten wurden in LibreOffice ausgedünnt und entsprechend angepasst.

2.2.4 Mathematik & Physik

Hier spielt LaTeX seine großen Vorteile aus. Hier können alle Arten von Gleichungen erstellt werden. Hier für gibt es im Internet endlos viele Hilfen. Auf Wikipedia kann man hier aber verweisen:
→ `wikipedia.org`

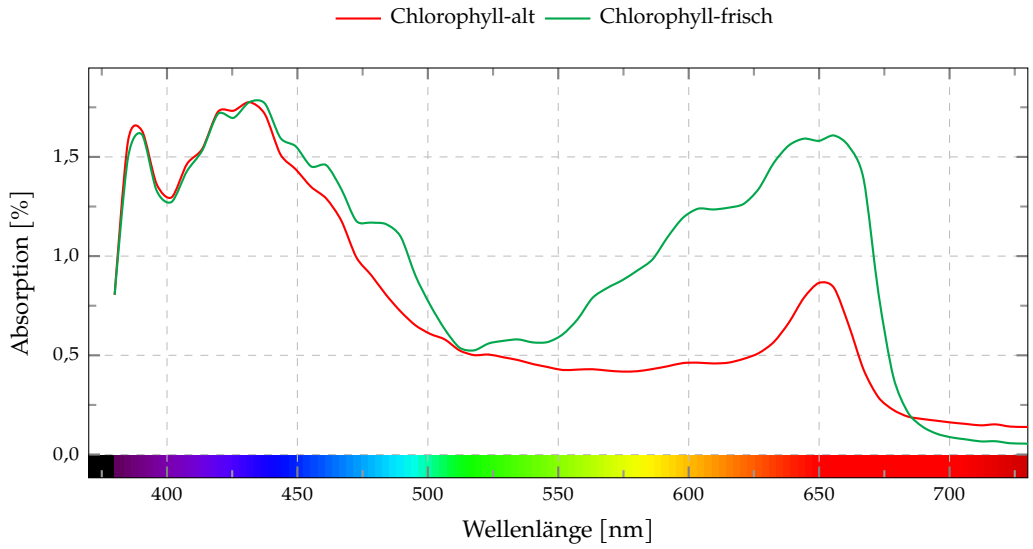


Figure 3: *Dieses Diagramm wird nicht im Haupt-Dokument erzeugt, sondern von extern eingebunden. So bleibt der LaTeX-Code sauber.*

$$\begin{aligned} f(x) &= x^2 - 4 \\ x^2 - 4 &= 0 \\ x^2 &= 4 \\ x &= \pm\sqrt{4} \\ x &= \pm 2 \end{aligned} \tag{3}$$

2.3 Zusätzliche Unterteilung

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.[1–3]

References

[1] G. D. Greenwade, *Demojournal* **1993**, *14*(3), 342–351, doi: 130.1.1/jpb0013.

[2] H. Zauder, U. Schauder, B. Angst, *Die neuen Erkenntnisse der Prüfungsangst*, 4th Aufl., Grusl-Verlag, Berlin, **2032**.

[3] B. Wachter, H. Huber, *Webtext ohne Sinn*, **1993**, doi: 10.1098/10234er5, `webpause.de`, gef.: 1.2025.