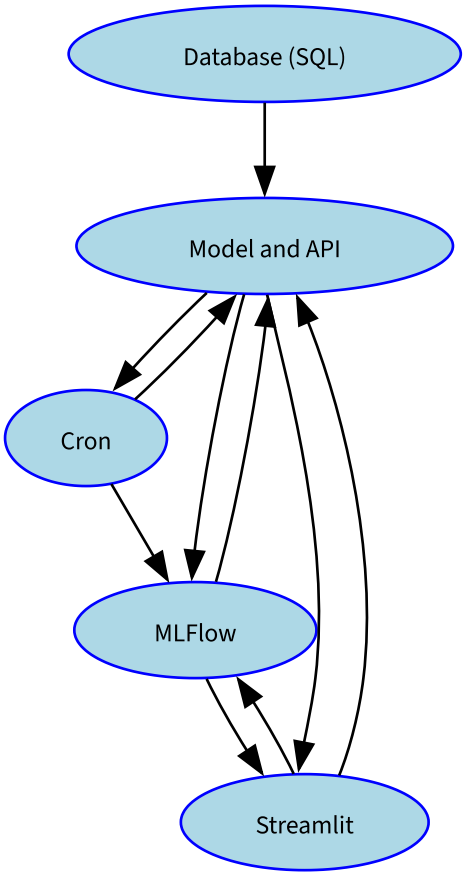
Table of Contents

Go to:

- Introduction
- Automation
- Preprocessing
- Modelling
- Prediction
- Conclusion

Rain in Australia

Project Structure



Dockerisation

five docker containers: cron, MySQL, MLFlow, Streamlit and model services

- cron container for automating the process
- MySQL container hosts all the raw data
- MLFlow container hosts the mlflow server for storing and restoring the best models
- model container hosts the data substracting, data preprocessing, training, predicting, and

FastAPI services

- Streamlit container hosts the Streamlit app
- to start(or build if not exists) the docker compose use docker compose up and docker-compose.yml - file
- to open the Streamlit app, in your browser, go to http://localhost:8501/
- to visit the MLflow server, in your browser, go to http://localhost:8080/

Automation

using crontab to automate the process:

- calls cron_pipeline.sh every 10 minutes
- the script calls the FastAPI endpoints in the model container in the following order:
- make dataset (chooses a random part of the original data to simulate changes in the data)
- preprocess data
- train model

MySQL Database

The MySQL database container hosts the raw data. The data is stored in a table called weather_data:

- the process takes the big weatherAUS.cvs from this source: https://www.kaggle.com/datasets/jsphyg/weather-dataset-rattle-package?resource=download
- and converts it into a SQL database by creating first an empty tabel with the column definition and then import the data.
- make_dataset.py then can filter and delete specific columns or parameter (eg. location) to create a smaller subset of the data for preprocessing and modelling.

- eg. 20 % of the data is randomly chosen to simulate changes in the data over time