Table of Contents

Go to:

- Introduction
- Automation
- Preprocessing 🗓️
- Modelling 📉
- Prediction 📈
- Conclusion ❕

Rain in Australia

FastAPI

The FastAPI application is utilized to expose the core functionalities of the machine learning project:

- making dataset,
- preprocessing,
- model training,
- model prediction.

Generally speaking, making dataset and preprocessing are fast to finish, while model training and prediction can be quite time consuming(depending on the computing resources available), the FastAPI application is designed to run asynchronously.

- The model training and prediction can be started in the background and the FastAPI application can continue to serve other requests(making new dataset or preprocessing).
- Callback functions are injected into model training and prediction to provide real-time updates on the progress and the final results.
- The updates and status can be queried using corresponding endpoints.

Conclusion

- The project does predict the weather of tomorrow by using changing data automatically
- The steps for the making dataset, preprocessing, and modelling are automized by crontab

- For the application 5 Docker containers are used via docker compose. Communicating is managed by docker compose network.
- Model training is tracked by the MLFlow,
- FastAPI application is used to expose the core functionalities of the project.

Outlook

When we had more time for the project we would extend it with:

1. Production-grade orchestration (e.g. Airflow) instead of cron,
   Advantages of Airflow: support monitoring of the models, more complex workflows possible
2. Data and experiment versioning using DagsHub
   Advantages of DagsHub: better tracking of data, code and model artifacts changes over time
3. Better observability and security
   Prometheus and Grafana, authenticated access to the API and Streamlit app
4. End-to-end CI/CD for the ML pipeline (tests, linting and automated deployment of models and API).