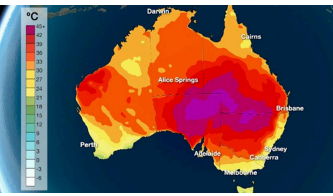


AT2 - Machine Learning as a Service



The Brief:

You are working for a company called Open Meteo that is providing weather forecasts to its client. A year ago, a new program was launched by the new CTO to modernise their technology stack. They succeeded to re-architecture their historical offering and services via API. Customers can now self-serve and integrate with the different services by themselves: <https://open-meteo.com/> ↗ (<https://open-meteo.com/>).

The next phase of this program is to embed AI into their services. You have been chosen to work on a project to train ML models to predict rain in the future.

You are tasked to build 2 different models that will be deployed into production as API (on Render):

- a predictive model using a Machine Learning algorithm to accurately predict if it is going to rain (not precipitation) or not (binary classification) in a week time (exactly + 7 days from a given date) in Sydney.
- a predictive model using a Machine Learning algorithm to accurately predict the cumulated volume of precipitation fall (not rain only) in mm (regression) within the next 3 days time in Sydney

You will decide which performance metrics to use for these models.

Dataset

You will have to design your dataset for this project using historical data for Sydney (latitude: -33.8678, longitude: 151.2073).

Historical weather data is available via the publicly available API:

https://open-meteo.com/en/docs/historical-weather-api#json_return_object ↗ (https://open-meteo.com/en/docs/historical-weather-api#json_return_object)

This API provides you with information for a given point in time (hourly or daily). You will have to design the target variables for both of your models taking into account the period of prediction for each.

You will have to decide which parameters can be used as features for your models. Be wary of any data leaks that can be fed to your models during training.

You can use any data prior to 2025 as training, validation and test sets. All the data from 2025 onwards should be used as data in production (i.e. not accessible during the project).

Git Repos:

Each student will set up 2 PRIVATE github repositories:

- One used for experimentation following the cookiecutter data science template
- Another one for the FastAPI application

Each student needs to extend the functionalities of their custom Python Package (set up for AT1), deploy them on TestPyPI and use them for this assignment.

Git Repo for Experimentation:

Each student will set up a Github repository for your experiments with the following requirements:

- Your Github repository needs to be private
- You need to provide admin access the following users to your private Github repository:
 - anthony.so@uts.edu.au (<mailto:anthony.so@uts.edu.au>)
 - reasmey.tith@uts.edu.au (<mailto:reasmey.tith@uts.edu.au>)
 - Natalia.Tkachenko@uts.edu.au (<mailto:Natalia.Tkachenko@uts.edu.au>)
 - Huy.Nguyen-1@uts.edu.au (<mailto:Huy.Nguyen-1@uts.edu.au>)
 - savinay.singh@uts.edu.au (<mailto:savinay.singh@uts.edu.au>)
 - TheHai.Bui@uts.edu.au (<mailto:TheHai.Bui@uts.edu.au>)
- You need to use Cookiecutter Data Science template to setup your project structure
- You need to create 2 sub-folder called **'rain_or_not'** and **'precipitation_fall'** inside both **'notebooks/** and **'models/** folders
- The notebooks (ipynb files) need to be stored in the correct sub-folders in **'notebooks/**
- Use the following naming convention for your notebooks:

36120-25SP-<student_id>-experiment_<number>

Such as: **36120-25SP-149874-experiment-1.ipynb**

- Save your best models artefacts in the **'models/** folder
- Save your experiment reports in the **'reports/** folder
- Provide the **pyproject.toml** and **requirements.txt** files at the root of your repository
- Don't forget to include some instructions in the **'README.md'** file for setting up your environment and running your code.
- You are required to build custom modules (function, classes) and publish them on TestPyPI
- **'github.txt'**: text file containing the link to your Github repository for your experimentation phase

Git Repo for API:

Each student will set up a separate Github repository for the API with the following requirements:

- Your Github repository needs to be private
- You need to provide admin access the following users to your private Github repository:
 - anthony.so@uts.edu.au (<mailto:anthony.so@uts.edu.au>)
 - reasmey.tith@uts.edu.au (<mailto:reasmey.tith@uts.edu.au>)
 - Natalia.Tkachenko@uts.edu.au (<mailto:Natalia.Tkachenko@uts.edu.au>)
 - Huy.Nguyen-1@uts.edu.au (<mailto:Huy.Nguyen-1@uts.edu.au>)
 - savinay.singh@uts.edu.au (<mailto:savinay.singh@uts.edu.au>)
 - TheHai.Bui@uts.edu.au (<mailto:TheHai.Bui@uts.edu.au>)
- This repository needs to comply with the following structure:
 - **'app'**: folder that will contain the **'main.py'** with the code of the FastAPI app
 - **'models'**: folder that will contain the trained models from your experiments and will be loaded by your FastAPI app

- `pyproject.toml` and `requirements.txt`: text files containing the list of Python packages that need to be installed for running your FastAPI app
 - `Dockerfile`: text file containing the Docker instructions to build and launch your Docker container for your FastAPI app
 - `github.txt`: text file containing the link to your Github repository for your FastAPI app
- Here are the list of expected API endpoints:

Endpoints	Description
<code>/</code> (GET)	Displaying a brief description of the project objectives, list of endpoints, expected input parameters and output format of the model, link to the Github repo related to this project
<code>/health/</code> (GET)	Returning status code 200 with a string with a welcome message of your choice
<code>/predict/rain/</code> (GET)	<p>Returning the prediction on if it will rain in exactly 7 days or not.</p> <p>Here are the expected input parameters for this endpoint:</p> <ul style="list-style-type: none"> • <code>date</code>: date from which the model will predict rain or not in a week's time (i.e. date + 7 days). The expected date format is YYYY-MM-DD. <p>The expected output for this prediction needs to be a JSON format with the following structure:</p> <pre>{ "input_date": "2023-01-01", "prediction": { "date": "2023-01-08", "will_rain": TRUE, } }</pre>
<code>/predict/precipitation/fall/</code> (GET)	<p>Returning the predicted cumulated sum of precipitation (in mm) within the next 3 days.</p> <p>Here are the expected input parameters for this endpoint:</p> <ul style="list-style-type: none"> • <code>date</code>: date from which the model will predict the sales on. The expected date format is YYYY-MM-DD. <p>The expected output for this prediction needs to be a JSON format with the following structure:</p> <pre>{ "input_date": "2023-01-01", "prediction": { "start_date": "2023-01-02", "end_date": "2023-01-04", "precipitation_fall": "28.2", } }</pre>

- You will need to deploy this repository on Render. You should be able to send predictions requests to both of your models.

Expected Deliverables:

1. Zip file, submitted on Canvas, containing project code, artefacts, instructions or any relevant documents for the Experimentation Repository
2. Zip file, submitted on Canvas, containing project code, artefacts, instructions or any relevant documents for the API Repository
3. Final report, submitted on Canvas, describing the trained models, performance achieved, API structure, instructions for running predictions. The report should not exceed 3000 words. Don't forget to include links to your Github repository and API url.
4. Accessible FastAPI and models in Render (deployment using free-tier account)

Submission:

All assignments need to be submitted before the due date on Canvas. Penalties will be applied for late submission.

Templates

You need to use the following templates:

- [Jupyter notebook for experimentation \(https://canvas.uts.edu.au/courses/36524/files/10036483?wrap=1\)](https://canvas.uts.edu.au/courses/36524/files/10036483?wrap=1) [↓ \(https://canvas.uts.edu.au/courses/36524/files/10036483/download?download_frd=1\)](https://canvas.uts.edu.au/courses/36524/files/10036483/download?download_frd=1)
- [Final report \(https://canvas.uts.edu.au/courses/36524/files/10036481?wrap=1\)](https://canvas.uts.edu.au/courses/36524/files/10036481?wrap=1) [↓ \(https://canvas.uts.edu.au/courses/36524/files/10036481/download?download_frd=1\)](https://canvas.uts.edu.au/courses/36524/files/10036481/download?download_frd=1)

Criteria	Ratings					Pts
1. Comprehensibility, quality, reliability, robustness and readability of code	20 to >17.0 Pts HD 🏆 Completed above 90% of the requirements (Github repos, notebooks, FastAPI, custom Python package functionalities) 🏆 Clearly and effectively documented code (README, code comments, docstring) 🏆 Code is efficient, easy to understand, and maintain 🏆 Code executes without errors and fully covered with unit tests	17 to >15.0 Pts D ✅ Completed between 80-90% of the requirements (Github repos, notebooks, FastAPI, custom Python package functionalities) ✅ Clearly documented code (README, code comments) ✅ The code is fairly efficient without sacrificing readability and understanding ✅ Code executes with minor errors and partially covered with unit tests	15 to >13.0 Pts C 🟢 Completed between 65-80% of the requirements (Github repos, notebooks, FastAPI, custom Python package functionalities) 🟢 Basic code documentation has been completed (README, code comments) 🟢 A logical solution that is relatively easy to follow but it is not the most efficient 🟢 Code executes with minor errors and limited coverage of unit tests	13 to >10.0 Pts P ⚠️ Completed between 50-65% of the requirements (Github repos, notebooks, FastAPI, custom Python package functionalities) ⚠️ Very limited code documentation included (code comments) ⚠️ A difficult to understand and inefficient solution ⚠️ Code executes with major errors and no unit tests	10 to >0 Pts F ❌ Completed less than 50% of the requirements (Github repos, notebooks, FastAPI, custom Python package functionalities) ❌ No code documentation included ❌ The code is poorly organized and very difficult to read ❌ Code executes with critical errors	20 pts
2. Quality of results and recommendation and depth of discussion of ethics/privacy issues (including matters related to Indigenous people), value, benefits, risks and recommendation for business stakeholders and final users in final report	20 to >17.0 Pts HD 🏆 Description of data preparation methods/approaches is clear, accurate, detailed enough and demonstrates good understanding of ML theory and/or practice 🏆 The ML approach, relevance and justification are clearly formulated and well-substantiated using academic or external literature 🏆 Describe the performance of the models with proper analysis and explanation for any variations in performance 🏆 Critical affected parties (both direct and indirect) are identified 🏆 Solution and ethical analysis is logical and clearly presented at a level that reflects extensive reflection and insight. 🏆 Key recommendations to achieve business objectives with strong justifications and proper references. Good relevance between your analysis and the organisational strategies	17 to >15.0 Pts D ✅ Description of data preparation methods/approaches is clear, accurate, and detailed enough to reproduce the work ✅ The ML approach, relevance and justification are well-connected using some grounding in academic or external literature ✅ Describe the performance of the models with reasonable analysis and explanation for any variations in performance ✅ Critical directly affected parties are identified ✅ Solution and ethical analysis is logical and clearly presented. ✅ Key recommendations to achieve business objectives with clear justifications and references	15 to >13.0 Pts C 🟢 Description of data preparation methods/approaches is clear but some details are missing for reproducibility of work 🟢 The ML approach, relevance and justification are broadly described. Limited academic or external literature used to substantiate arguments. 🟢 Describe the performance of the models with partial analysis, but the explanation for any variations in performance may be clearer 🟢 Most critical affected parties (both direct and indirect) are identified. 🟢 Solution and ethical analysis is logical and clear. The analysis may be superficial at some level. 🟢 Good attempt in making business recommendations, though the justifications may be a bit weak	13 to >10.0 Pts P ⚠️ Not detailed and/or accurate depiction of data preparation methods ⚠️ The ML approach, relevance and justification are unclear. Limited academic or external literature used to substantiate arguments. ⚠️ Describe the performance of the models with some analysis, but the explanation for any variations in performance may be insufficient ⚠️ Most critical directly affected parties are identified. ⚠️ Solution and ethical analysis is too generic and not specific enough ⚠️ Attempt in making business recommendations, though the justifications are vague or not relevant	10 to >0 Pts F ❌ Demonstrates no clear focus or development of data preparation methods ❌ Missing elements of the ML approach, relevance and justification ❌ No description or inadequate description of the performance of the models and the explanation for any variations in performance may be absent or inadequate. ❌ Affected parties are not identified completely. Major players critical to analysis are not identified. ❌ Analysis was not carried out sufficiently and is fundamentally flawed. Solution may be trivial or illogical. ❌ Limited or vague business recommendations on the topics	20 pts
3. Justification of decisions made with clear and strong evidence supporting claims (business objectives, data transformation performed, models selected, hyperparameters selected and accuracy of results)	25 to >21.0 Pts HD 🏆 Hypotheses are testable and accurately stated in acceptable format 🏆 Data analysis shows excellent understanding and identification of data anomalies and main characteristics of the data 🏆 Uses correct and complete quantitative and/or qualitative analysis to make relevant and correct decisions 🏆 Thorough data preparation that demonstrates strong understanding of critical data quality issues and ML requirements 🏆 Well-chosen algorithms, measurements and analysis that illustrate how the baseline and trained models perform on all sets 🏆 Many aspects of evaluation are discussed and a clear conclusion is drawn, with direct reference to the purpose of the experiment	21 to >18.0 Pts D ✅ Hypotheses are testable and clearly stated in acceptable format ✅ Data analysis shows a good understanding and identification of main data anomalies and main characteristics of the data ✅ Quantitative and/or qualitative analysis is given to support a relevant decision, but it is either only partially correct or partially complete ✅ Substantial data preparation that demonstrates good understanding of data quality issues and ML requirements ✅ Well-chosen algorithms, measurements and analysis that illustrate how the baseline and trained models perform but on the training set only ✅ A clear conclusion is drawn from the work reported and a defended proposal for further investigation is proposed, with clear links to both the work reported and the domain of application.	18 to >16.0 Pts C 🟢 Hypotheses are too broad and not easily testable 🟢 Data analysis shows some good understanding and identification of data anomalies and characteristics of the data 🟢 Reasonable decision is made but quantitative and/or qualitative analysis is lacking details 🟢 Adequate data preparation that demonstrates some understanding of data quality issues and ML requirements 🟢 Minor issue with the choice of algorithms, measurements or analysis for the baseline and trained models 🟢 A rounded, balanced summary of the work is presented with a justified proposal given	16 to >12.0 Pts P ⚠️ Hypotheses are generic and not specific ⚠️ Data analysis shows limited understanding and identification of data some anomalies and some characteristics of the data ⚠️ An incorrect quantitative and/or qualitative analysis or major error is given to support a decision. ⚠️ Partial data preparation and missing critical data preparation steps ⚠️ Major issue with the choice of algorithms, measurements and analysis for the baseline and trained models ⚠️ A summary of the work is presented and a proposal made	12 to >0 Pts F ❌ Hypothesis is poorly stated, missing or not relevant ❌ No attempt to perform data analysis or not relevant ❌ Either no reasonable decision is made or, if present, is not based on quantitative and/or qualitative analysis ❌ Irrelevant or no data preparation steps performed ❌ No analysis performed or incorrect choice of algorithms or measurements for the baseline model and trained models ❌ Answer does not demonstrate adequate engagement with the problem nor a qualitative understanding of the work reported	25 pts
4. Appropriateness of communication style to audience in final report (summarisation, quality of findings, recommendations, visualisations, readability, clarity)	20 to >17.0 Pts HD 🏆 Report is well-written and presented professionally, appropriate use of formality items, professional use of references 🏆 Information and ideas are presented in a logical sequence which flows naturally and is engaging to the audience 🏆 Accurate observation table, data table, and graph are present. The data graph is properly set up and is supported by the data in the chart 🏆 Interpretation of data displays contains all critical elements and relevant details, and contains no irrelevant data	17 to >15.0 Pts D ✅ Well-written report that could be more concise or less error-prone ✅ Information and ideas are presented in a logical sequence which is followed by the reader with little or no difficulty ✅ Observation table or data table needs to be properly labeled to make sense of the data and /or graph needs to be properly labeled to better sense of the results ✅ Interpretation of data displays contains some critical elements and some relevant details, and additional irrelevant information may be included	15 to >13.0 Pts C 🟢 Most aspects are covered with good efforts, though minor mistakes or typos may exist 🟢 Information and ideas are presented in an order that the audience can follow with minimum difficulty 🟢 Data missing from the observation table, data table, and or graph 🟢 Interpretation of data displays contains few critical elements and few relevant details	13 to >10.0 Pts P ⚠️ Reasonably-written report with grammar, mechanics, or structural issues ⚠️ Information and ideas are presented in an order that the audience can follow with minimum difficulty but lacks relevance or details ⚠️ Includes appropriate but not accurate displays of the data ⚠️ Interpretation of data displays contains few critical elements and few relevant details, and includes irrelevant information	10 to >0 Pts F ❌ Poorly presented, many typos, missing key sections, poor use of formality items, badly structured report ❌ Information and ideas are poorly sequenced (the author jumps around). The audience has difficulty following the thread of thought ❌ Does not include appropriate displays of the data ❌ Interpretation of data is absent or blatantly incorrect	20 pts
5. Robustness of deployed API services and relevance for	15 to >12.0 Pts HD	12 to >11.0 Pts D	11 to >9.0 Pts C	9 to >7.0 Pts P	7 to >0 Pts F	15 pts

Criteria	Ratings					Pts
documentation and instructions for deploying models and running web applications	🔧 API handles edge cases, invalid inputs, and concurrency with graceful error responses and fallback mechanisms. 🧑 Includes CI/CD pipelines or scripts (e.g., Docker, GitHub Actions) for seamless deployment and updates. 🧑 Clear, well-structured, and complete documentation covering setup, deployment, usage, and troubleshooting.	✅ API handles most expected inputs and errors with basic validation and error messages. ✅ Provides reproducible deployment steps, possibly with Docker or shell scripts, though not fully automated. ✅ Covers all major aspects of deployment and usage, though may lack advanced troubleshooting or visuals.	👍 Works for standard inputs but lacks robustness for edge cases or concurrent requests. 👍 Deployment is possible but requires manual steps or assumptions not documented. 👍 Covers setup and usage but lacks clarity or completeness in deployment instructions.	⚠️ API works but lacks input validation or fails under unexpected conditions. ⚠️ Deployment instructions are vague or incomplete, requiring guesswork. ⚠️ Documentation is missing key steps or lacks structure and clarity.	❌ API fails to run or crashes under normal conditions. ❌ No instructions or tools provided for deploying the model or API. ❌ Documentation is absent or unusable for setup or deployment.	
Total points: 100						