

AT3 - Data Product with Machine Learning

Start Assignment

- Due Friday by 23:59
- Points 100
- Submitting a file upload
- File types pdf, doc, docx, zip, ipynb, and py



The Brief:

You have been tasked to build a data product for users interested in investing in cryptocurrencies.

You will have to build a Streamlit app that lets users select one of the 4 cryptocurrencies: Bitcoin, Ethereum, XRP and Solana.

The user will be able to see historical information related to the selected cryptocurrency. You will decide what information and how much information you want to display. You will have to source your information via API calls to the following sources:

- <https://docs.kraken.com/api/docs/rest-api/get-ohlcv-data> → <https://docs.kraken.com/api/docs/rest-api/get-ohlcv-data>
- <https://docs.coingecko.com/v3.0.1/reference/coins-id-ohlcv> → <https://docs.coingecko.com/v3.0.1/reference/coins-id-ohlcv>
- <https://developers.tokenmetrics.com/reference/tokens> → <https://developers.tokenmetrics.com/reference/tokens>
- <https://developers.coindesk.com/documentation/data-api/introduction> → <https://developers.coindesk.com/documentation/data-api/introduction>

Each student will need to choose the specific type of information to be displayed on the Streamlit app and build logics to display them in a separate Streamlit Tab in the final app.

The user will also be able to assess predictions from ML models about the chosen cryptocurrency. Each model will need to predict a HIGH price the next day (day +1).

Each student will need to train a model with a different algorithm from the other team members on a single token. Each student will need to submit their best model and make it available in the Streamlit app.

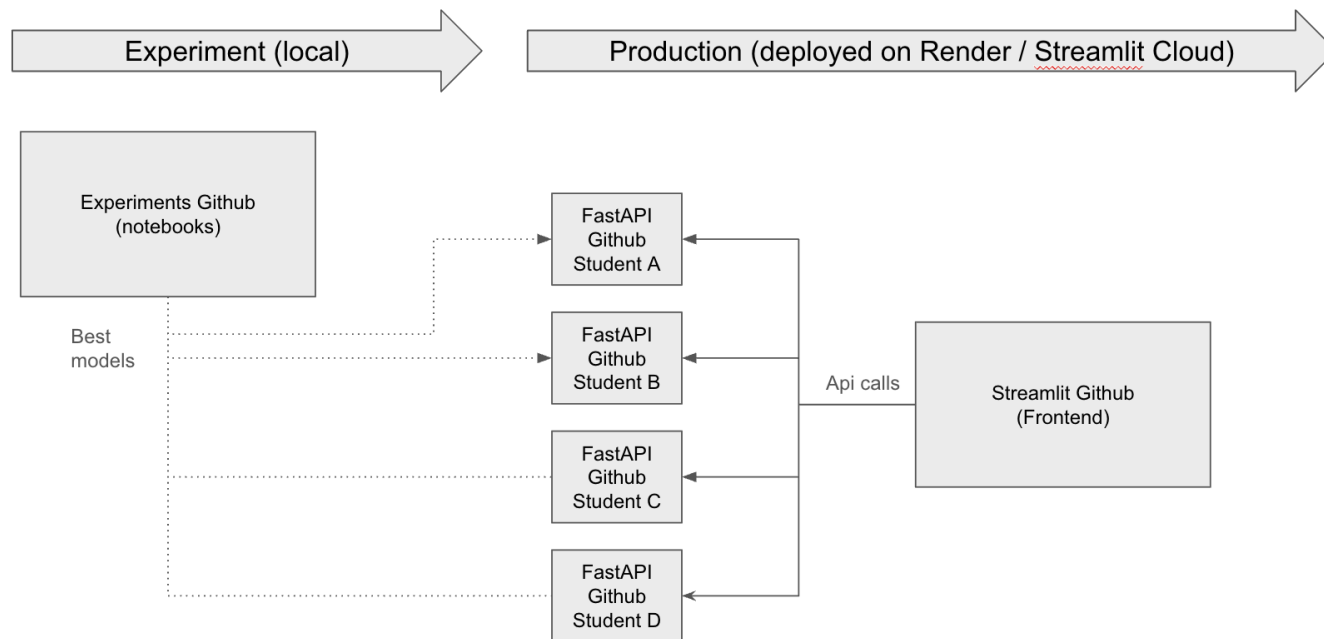
Git Repos:

Each **group** will set up several PRIVATE github repositories:

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

Each **student** will set up a separate PRIVATE github repository for hosting their own models and served via FastAPI.

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 **group** 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
- The notebooks (ipynb files) need to be stored in **`notebooks/`**
- Use the following naming convention for your notebooks:

36120-25SP-group<number>-<student_id>-AT3-experiment_<number>

Such as: **36120-25SP-group1-149874-AT3-experiment-1.ipynb**

- Save your best models artefacts in the **`models/`** 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.
- **`github.txt`**: text file containing the link to your Github repository for your experimentation phase

Git Repo for Streamlit:

Each **Group** 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 Streamlit app
 - **`students`**: folder that will contain the code for rendering Streamlit components from each student (separate them with a dedicated Python script for each student)
 - **`pyproject.toml`** and **`requirements.txt`**: text files containing the list of Python packages that need to be installed for running your Streamlit app
 - **`Dockerfile`**: text file containing the Docker instructions to build and launch your Docker container for your Streamlit app
 - **`github.txt`**: text file containing the link to your Github repository for your FastAPI app

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[\(mailto:reasmey.tith@uts.edu.au\)](mailto:reasmey.tith@uts.edu.au)Natalia.Tkachenko@uts.edu.au[\(mailto:Natalia.Tkachenko@uts.edu.au\)](mailto:Natalia.Tkachenko@uts.edu.au)Huy.Nguyen-1@uts.edu.au[\(mailto:Huy.Nguyen-1@uts.edu.au\)](mailto:Huy.Nguyen-1@uts.edu.au)savinay.singh@uts.edu.au[\(mailto:savinay.singh@uts.edu.au\)](mailto:savinay.singh@uts.edu.au)TheHai.Bui@uts.edu.au[\(mailto:TheHai.Bui@uts.edu.au\)](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/<token>`</code> (GET)	Returning the prediction on the trained model (HIGH price of the token the next day)

Python Requirement:

For all your code you need to follow these requirements:

- Python 3.11.4
- scikit-learn 1.5.1
- Pandas 2.2.2
- Jupyter Lab 4.2.3
- Fastapi 0.111.0
- Uvicorn 0.30.1
- Joblib 1.4.2
- Streamlit 1.36.0
- Xgboost 2.1.0
- Hyperopt 0.2.7
- Lightgbm 4.4.0
- Lime 0.2.0.1
- Wandb 0.17.4

Deliverables:

- Zip file (1 for each **group**), submitted on Canvas, containing project code, artefacts, instructions or any relevant documents for the Experimentation Repository. Use the following naming convention: 36120-25SP-AT3-Group<group_id>-experiments.zip (such as: **36120-25SP-AT3-Group01-experiments.zip**)
- Zip file (1 for each **group**), submitted on Canvas, containing project code, artefacts, instructions or any relevant documents for the Streamlit App Repository. Use the following naming convention: 36120-25SP-AT3-Group<group_id>-streamlit.zip (such as: **36120-25SP-AT3-Group01-streamlit.zip**)
- Zip files (1 for each **student**), submitted on Canvas, containing project code, artefacts, instructions or any relevant documents for the FastAPI Repository. Use the following naming convention: 36120-25SP-AT3-Group<group_id>-api.zip (such as: **36120-25SP-AT3-Group<group_id>-api.zip**)
- Final report describing the machine learning models and their hyperparameters or neural networks architecture, performance achieved, deployment. The report should not exceed 15000 words. Use the following naming convention: 36120-25SP-AT3-Group<group_id>-report.docx (such as: **36120-25SP-AT3-Group01-report.docx**)
- Accessible Streamlit App on Streamlit Community Cloud, FastAPI and models on Render

Note: Your Streamlit App, FastAPI and models need to be accessible online respectively on Streamlit Community Cloud and Render. A series of test will be performed on it and will be part of the marking

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 \(https://canvas.uts.edu.au/courses/36524/files/10039499?wrap=1\)](https://canvas.uts.edu.au/courses/36524/files/10039499?wrap=1) 
(https://canvas.uts.edu.au/courses/36524/files/10039499/download?download_frd=1)
- [Final report \(https://canvas.uts.edu.au/courses/36524/files/10039498?wrap=1\)](https://canvas.uts.edu.au/courses/36524/files/10039498?wrap=1) 
(https://canvas.uts.edu.au/courses/36524/files/10039498/download?download_frd=1)

Dataset:

You can download the dataset for the experimentation phase [here \(https://canvas.uts.edu.au/courses/36524/files/10429924?wrap=1\)](https://canvas.uts.edu.au/courses/36524/files/10429924?wrap=1). [↓](https://canvas.uts.edu.au/courses/36524/files/10429924/download?download_frd=1)
(https://canvas.uts.edu.au/courses/36524/files/10429924/download?download_frd=1)

Data Dictionary:

Name	Description
timeOpen	Timestamp (ISO 8601) of the start of this time series interval.
timeClose	Timestamp (ISO 8601) of the end of this time series interval.
timeHigh	Timestamp (ISO 8601) of the high of this time series interval.
timeLow	Timestamp (ISO 8601) of the low of this time series interval.
name	The CoinMarketCap cryptocurrency ID
open	Opening price for time series interval.
high	Highest price during this time series interval.
low	Lowest price during this time series interval.
close	Closing price for this time series interval.
volume	Adjusted volume for this time series interval.
































marketCap	Market cap by circulating supply for this time series interval.
timestamp	Timestamp (ISO 8601) of when the conversion currency's current value was referenced for this conversion.

36120-AT3

Criteria	Ratings					Pts
1. Comprehensibility, quality, reliability, robustness and readability of code	25 to >21.0 Pts HD Completed above 90% of the requirements (Github repos, notebooks, Streamlit, FastAPI, custom Python package functionalities) Code is efficient, easy to understand, and maintain Clearly and effectively documented code (README, code comments, docstring) Code executes without errors	21 to >18.0 Pts D Completed between 80-90% of the requirements (Github repos, notebooks, Streamlit, FastAPI, custom Python package functionalities) The code is fairly efficient without sacrificing readability and understanding Clearly documented code (README, code comments) Code executes without errors	18 to >16.0 Pts C Completed between 65-80% of the requirements (Github repos, notebooks, Streamlit, FastAPI, custom Python package functionalities) A logical solution that is relatively easy to follow but it is not the most efficient Basic code documentation has been completed (README, code comments) Code executes with minor errors	16 to >12.0 Pts P Completed between 50-65% of the requirements (Github repos, notebooks, Streamlit, FastAPI, custom Python package functionalities) A difficult to understand and inefficient solution Very limited code documentation included (code comments) Code executes with non-critical errors	12 to >0 Pts F Completed less than 50% of the requirements (Github repos, notebooks, Streamlit, FastAPI, custom Python package functionalities) The code is poorly organized and very difficult to read No code documentation included Code executes with critical errors	25 pts
2. Quality of results and recommendation and depth of discussion of ethics/privacy issues (including matters related to Indigenous people), value, benefits,	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,	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-	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	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	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	20 pts

Criteria	Ratings					Pts
risks and recommendation for business stakeholders and final users in final report	<p>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</p>	<p>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</p>	<p>described. Limited academic or external literature used to substantiate arguments. OK Describe the performance of the models with partial analysis, but the explanation for any variations in performance may be clearer OK Most critical affected parties (both direct and indirect) are identified. OK Solution and ethical analysis is logical and clear. The analysis may be superficial at some level. OK Good attempt in making business recommendations, though the justifications may be a bit weak</p>	<p>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</p>	<p>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</p>	
3. Justification of decisions made	20 to >17.0 Pts HD	17 to >15.0 Pts D	15 to >13.0 Pts C	13 to >10.0 Pts P	10 to >0 Pts F	20 pts

Criteria	Ratings						Pts
with clear and strong evidence supporting claims in final report (business objectives, data transformation performed, models selected, hyperparameters selected and accuracy of results)	<p>🏆 Excellent problem/opportunity statement. Clear, realistic and easy-to-understand set of well-consistent project goals. 🏆 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</p>	<p>✅ Good problem/opportunity statement. Clear and reasonable set of project goals, with good consistency ✅ 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</p>	<p>👍 Include problem/opportunity statement. Reasonable set of consistent project goals 👍 Data analysis shows some 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</p>	<p>⚠️ Shows reasonable attempt to a problem/opportunity statement, small, incomplete and set of goals with poor consistency ⚠️ 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</p>	<p>❌ No attempt to state problem/opportunity 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</p>		

Criteria	Ratings					Pts
	clear conclusion is drawn, with direct reference to the purpose of the experiment	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.				
4. Breadth of evidence of collaborative work (e.g. meeting minutes, details of contributions etc)	20 to >17.0 Pts HD  Led the group while doing their part  Participated in all group meetings  Helps direct group in assigning tasks and meeting goals  Almost always listens to, shares with, and supports the efforts of others. Tries to keep people working well together	17 to >15.0 Pts D  Did more than the required work  Participated in most group meetings  Participates in assigning tasks and setting goals.  Usually listens to, shares with and supports the efforts of others. Does not cause "waves" in the group	15 to >13.0 Pts C  Did their part of the required work  Participated in some group meetings  Participates marginally in assigning tasks and setting goals.  Often listens to shares with, and supports the efforts of others.	13 to >10.0 Pts P  Could have done more of the required work  Participated in few group meetings  Watches but doesn't participate in assigning tasks and setting goals.  Rarely listens to, shares with, and supports the efforts of others.	10 to >0 Pts F  Did not do any of the expected work  Not participated in any group meetings  Didn't participate to task assignment and goal settings  Had difficulty working with others	20 pts
5. Robustness of deployed API services and web app and relevance for documentation and instructions for	15 to >12.0 Pts HD  API handles edge cases, invalid inputs, and concurrency with graceful error responses and fallback mechanisms. 	12 to >11.0 Pts D  API handles most expected inputs and errors with basic validation and error messages.  Provides reproducible	11 to >9.0 Pts C  Works for standard inputs but lacks robustness for edge cases or concurrent requests.  Deployment is possible	9 to >7.0 Pts P  API works but lacks input validation or fails under unexpected conditions.  Deployment instructions are vague	7 to >0 Pts F  API fails to run or crashes under normal conditions.  No instructions or tools provided for deploying the model or API. 	15 pts

Criteria	Ratings					Pts
deploying models and running web applications	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. 🏆 Web app is fully functional, responsive web app that integrates seamlessly with the API and reflects real-time predictions.	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. ✅ Web app works well with the API, with minor UI or integration issues.	but requires manual steps or assumptions not documented. 🟡 OK Covers setup and usage but lacks clarity or completeness in deployment instructions. 🟡 OK Web app is mostly functional but may have bugs or inconsistent behavior.	or incomplete, requiring guesswork. ⚠️ Documentation is missing key steps or lacks structure and clarity. ⚠️ Web app is partially functional or has major usability or integration issues.	Documentation is absent or unusable for setup or deployment. ❌ Web app is non-functional or does not connect to the API.	
Total points: 100						