

The
Alan Turing
Institute

Plans for DECOVID Model Monitoring Framework

Jack Roberts, Ed Chalstrey
Research Engineering Group



Context

Machine learning/statistical models represent the data they were trained on:

- If the data changes, a model may no longer perform as expected.

DECOVID is happening in a rapidly evolving environment:

- Dataset will change; e.g. new hospitals, new measurements.
- Response to Covid-19 will change; e.g. new treatments, number of patients.

Context

Machine learning/statistical models represent the data they were trained on.

- If the data changes, a model may no longer perform as expected.

Performance of models trained for DECOVID is likely to decrease over time.

DECOVID is happening in a rapidly evolving environment:

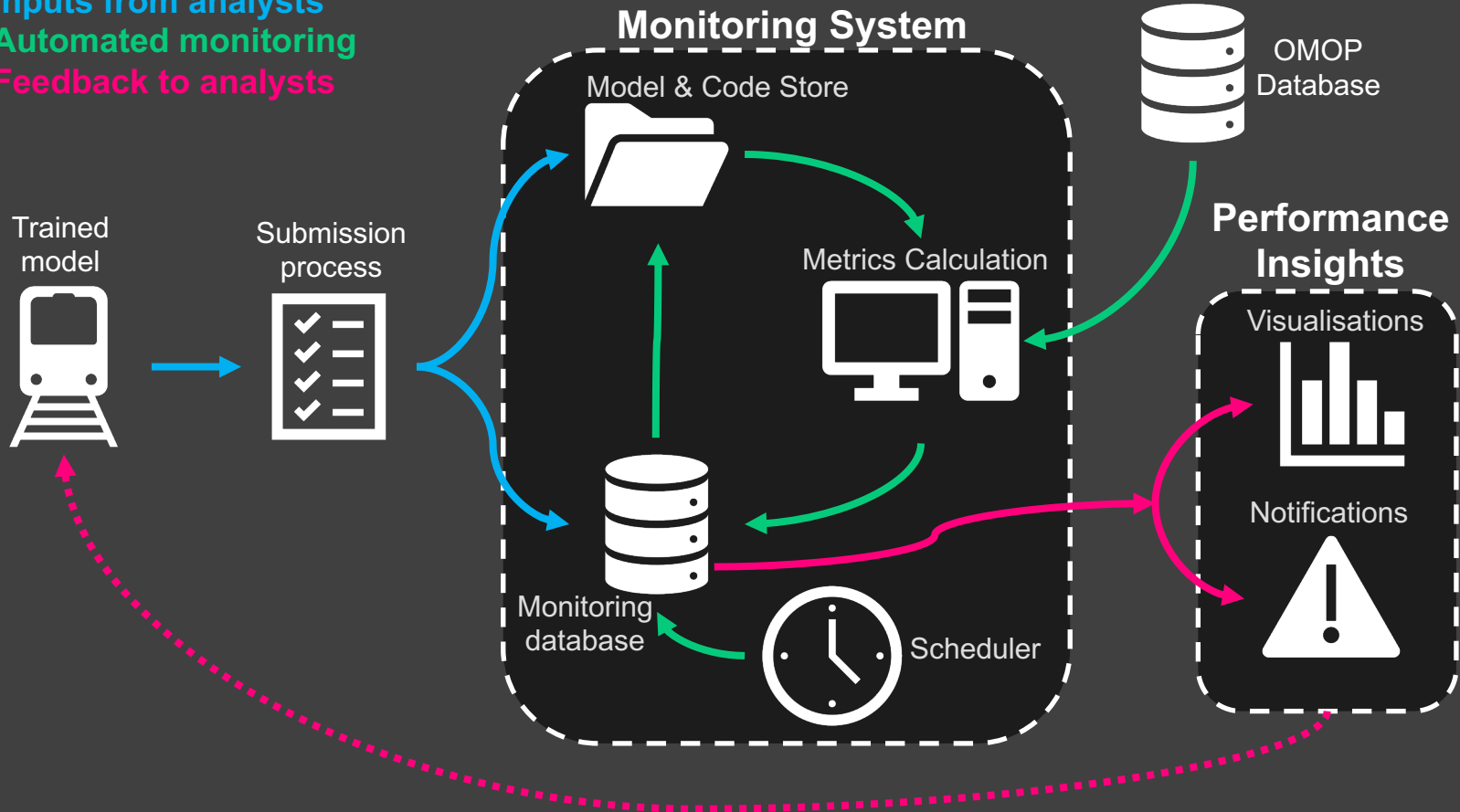
- Dataset will change; e.g. new hospitals added, different measurements.
- Response to Covid-19 will change; e.g. new treatments, number of patients.

Aim

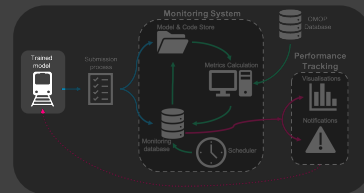
- DECOVID outputs to be reproducible, replicable, robust and relevant.
Continually appraise models and validate their utility on current data.
- Provide a system where analysts can automatically track the performance of their models on new data as it becomes available.
Reduce the burden on analysts of needing to re-validate their models.

Proposed System Design

Inputs from analysts
Automated monitoring
Feedback to analysts



1. Analyst team trains a model



- With whichever tools and workflow that team prefers (but we're focusing on Python & R).
- Saves the pre-trained model.

Aim for the system to place no constraints
on the analysts at this stage



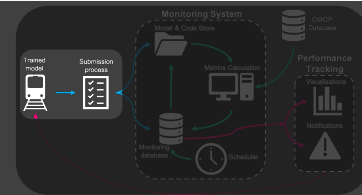
2. Prepare model for submission

Write a script that:

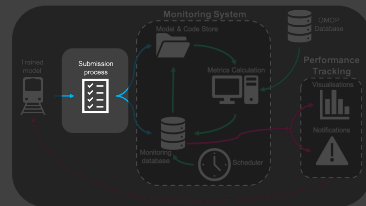
1. Gets data between a **start date** and **end date** from a given **database**.
2. Calculates predictions for that data using the pre-trained model.
3. Calculates N **single-valued performance metrics**.
4. Saves those metrics to file with a **known format** and location.

Define the **environment** to run the script in:

- All package versions (e.g. conda env, requirements.txt, renv...)
- Command to activate environment and run script.



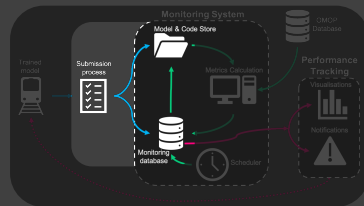
3. Submit the model



- Metrics script, environment + all other required code, model files etc.
- Metadata: Team details, research question, model description & version, etc.
- Reference values for all metrics (and the data subset these were calculated on).



4. Validate Model Reproducibility



Verify we can:

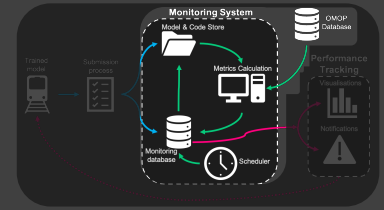
1. Setup the environment.
2. Calculate the metrics with no errors.
3. Get metric values that match the specified reference values.

If not: Feedback and work together to resolve any issues.

If yes: Add the model to the monitoring database.



5. Track Performance Metrics

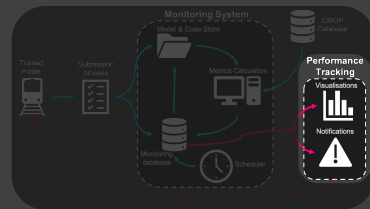


As new data becomes available:

- Automate recalculation of metrics for all models on the new data.
- Log the results (or runtime errors) to the monitoring database.



6. Evaluate and Distribute Results



Visualisations

- Performance changes over time.
- Comparison between different models.



Notifications

- If metrics drop below a threshold, notify analyst team.
- Team investigates source of the problem (then perhaps submits a new model version).

Links



GitHub Repo:

<https://github.com/alan-turing-institute/DECOVID-dataaccess>

("monitor" directory)



Contact:

jroberts@turing.ac.uk, DECOVID Slack, GitHub: @jack89roberts