



MLOps practices at KNMI

The Collaborative Quantitative Impact Forecasting use case

EMS2025

Alice Alfonsi

Jasper van Nieuwenhuizen Rosina Derks Margo van der Stam Luca Trani Jouke de Baar



Transitioning AI/ML research to operation is crucial to deliver reliable services and decision support that safeguard society



- Innovation, experimentation
- Data (somewhat) available
- Proof of concept Model performance Novelty
- Publications, Reports



- Robustness, reliability
- Established data streams
- Explainability and transparency Flexible model serving Continuous learning
- Operational product



KNMI MLOps initiative

- Official dedicated agile team since 2025
 4 developers, 1 scrum master, 1 product owner
- Goals
 - Operationalise AI/ML products
 - Develop and maintain AI/ML platform for researchers
 - Provide guidelines and promote best practices
 - Foster collaboration and exchange knowledge



AI/ML to better
understand impacts of
extreme and
compound events thus
improving our
communication
and information
capabilities

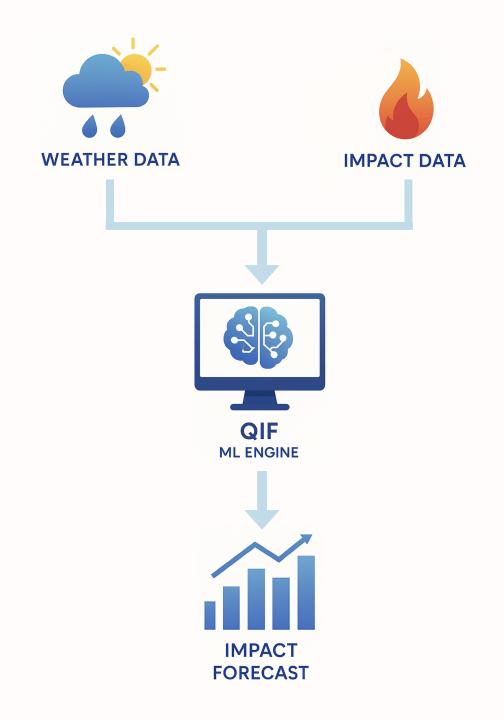




C-QIF for wildfires

Cooperative quantitative impact forecasting

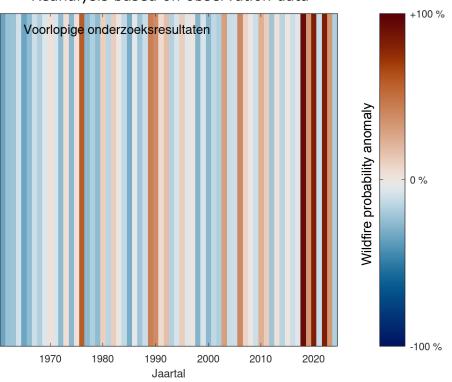
- Data-driven framework relying on historical weather and wildfire records
- Delivers probabilistic daily forecast of wildfire occurrence in the Netherlands
- Cloud-based, real-time operations for scalable, continuous data processing
- Supports operational resource allocation and public communication
- Stakeholders: KNMI weather room, NIPV





Wildfires in The Netherlands

Reanalysis based on observation data



- Wildfire occurrence and intensity are rising
- Assessing impact is complex due to many factors at play
 - High number of wildfires anticipates severe impact
 - Time is crucial in preventing high impact
- Urgency to shift from reactive to proactive approach





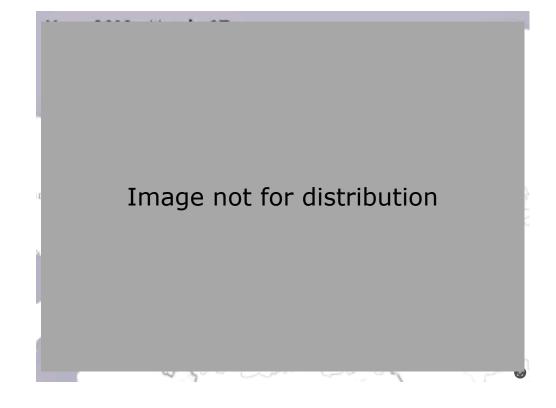
MVP: observations-based

Based on KNMI De Bilt observations and NIPV number of wildfires dataset

- All fires classified as wildfire from 2017
- Daily mean windspeed and relative humidity, min/max temperature, precipitation, month and averages over 14-days

Focus

- Test the full base infrastructure early
- Provide hindcast validation of the model



Around 750-1000 incidents per year



Approach building blocks

 \Diamond

OPERATION: Real-time dashboard, monitoring, secure and maintainable model

TRACKING: Model development and data tracking in MLFlow, reproducibility

DATA: Automated data stream, preprocessing and storage

INFRASTRUCTURE: Scalable and secure AWS cloud, CI/CD pipelines

SOFTWARE PRACTICES: Clean code, documentation, testing

GOVERNANCE: Guidelines, requirements, continuous feedback



Technical timeline

MARCH 2025
RESEARCH
PROTOTYPE

AUGUST 2025

OBSERVATIONS

MVP

OPERATIONAL FORECAST

- Local Octave code
- Historical wildfire records + KNMI observations/ forecast
- M5' ensemble model

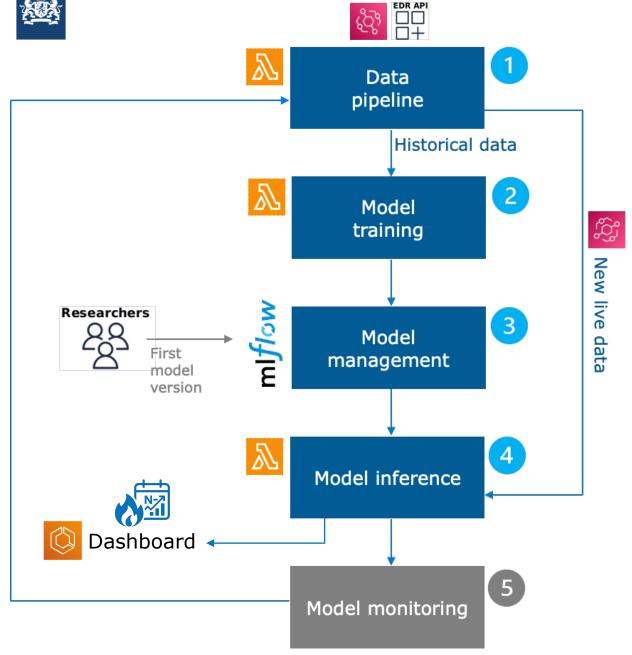


- Port to Python &
 Evaluate model alternatives
- Identify required data and sources, processing and storage
- Build infrastructure and pipelines
- Write documentation

- Cloud-based python
- Historical wildfire records + KNMI observations
- XGBoost QuantileRegressor model



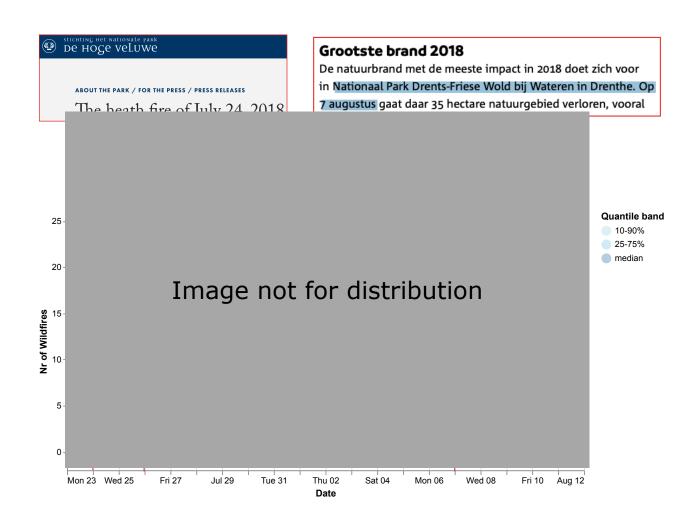
- Automated model tracking in self-hosted MLflow, manual version promotion
- Automated historical and live observation data retrieval and processing
- Input and output data stored as parquet dataset partitioned by date
- Manual model monitoring and training
- Automated daily inference results with incoming live data as dashboard service







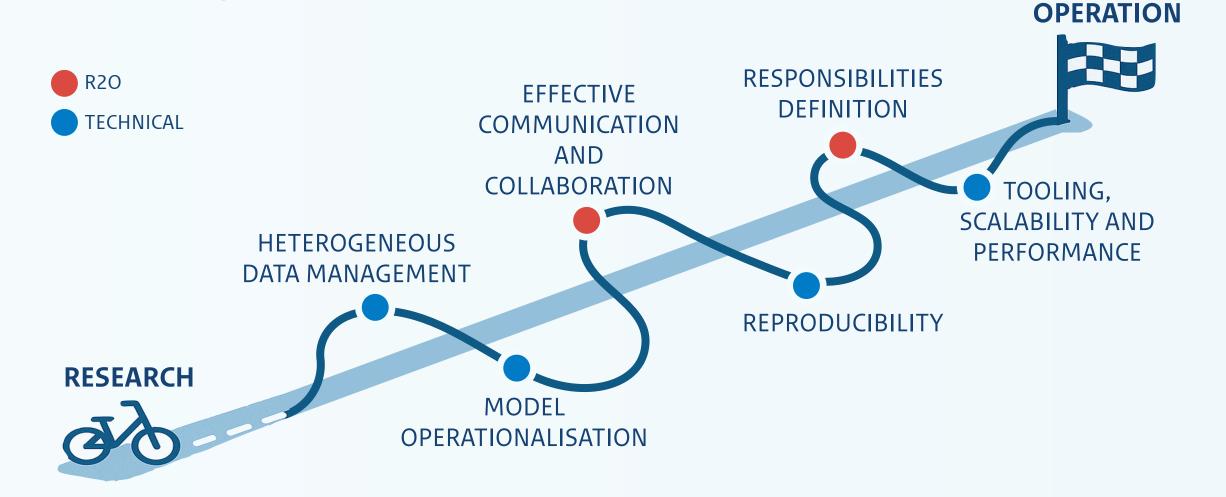
Hindcast dashboard



- Operational framework with observation data in place
- First hindcast results reflect expectations
- Model research and validation ongoing

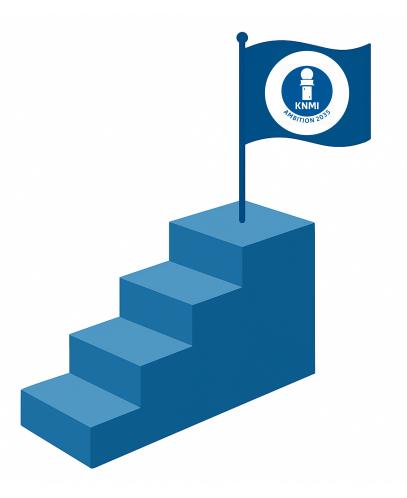


Challenges and lessons





Outlook



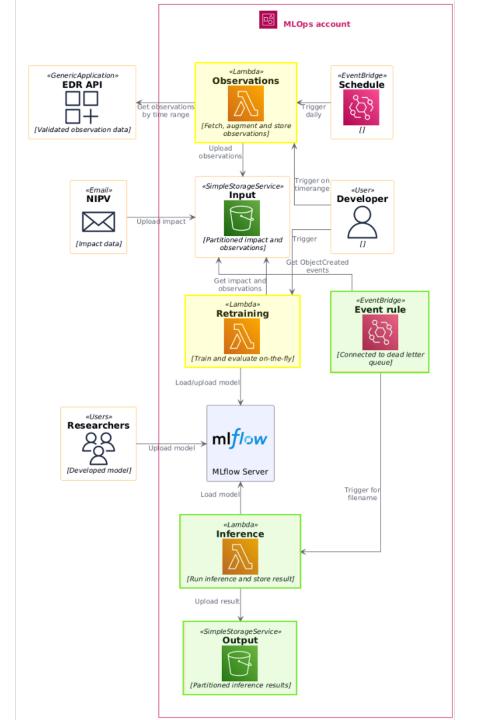
- Towards real-time operational service forecast using ECMWF inputs
- Operational testing and validation with users (KNMI weather room and NIPV)
- Extend the approach to other applications
- Build embedded MLOps culture at KNMI



Backup slides



AWS infrastructure Diagram





Project phases

Requirements:

- Stakeholders & users are identified
- Business value is known and documented

Go / No Go from PO MLOps

| What:

Product:

Team (R)

Business Analyst (C)

Researcher (C)

Architect (C)

Technical

feasibility

This phase can be done by only part

· Monitoring / feedback loop is

· Input data for both training and

· Estimate how much work it is.

· Who is responsible for what?

· Repo with reproducible research code

· Archive original research code in Teams

inference are known and accessible.

of the MLOps team (i.e. 2 people)

Reproduce research with data

· Create high level plan

· High level plan

Actors: PO-(A,R) Portfolio Management (A,R) Researcher (C) Stakeholders (C) Team (I)

Business Value Determination

Prioritization of research projects by PO of MLOps

What:

Research which can be brought to operation

- · High business value
- Risk management

Product:

 Portfolio management of research products



Requirements:

Products are delivered.

Go / No Go from MLOps



Ready for MVP

Requirements:

Go / No Go from MLOps



Actors:

PO (A)
Team (R)
Architect (C)
Stakeholders (teams+ business) (C)

Requirements:

Agreement from business

Go / No Go from MLOps



Actors: PO (A) Team (R) Stakeholders (business) (I) Chain partners (C)

PoC

This phase can be optional

- does the code need rewriting to Python?

Actors: PO (A)

Team (R)

Architect (I)

Researcher (C)

Stakeholders - business (I)

- does the model needs experimenting?

What:

- Python code which reproduces research concept with data
- · Discuss model with MLOps and researcher

Product:

- Documented comparison of model implementations
- · Updated repo with Python research code

.

MVP

- · Discuss with stakeholders & users on final product
- · Ensure training is possible
- · Coordinate with other team plannings
- · Implement inference on AWS (if best solution)
- Discuss with stakeholders (internal & external) on data delivery, availability, responsibilities and contingency place.
- · Do the risk analysis

Product:

- · MVP is demo-able to business
- · Contingency plan
 - · Retraining plan
 - · Model drift detection plan
 - · Availability agreements with partners

Operation

What:

- Implement observability
- · Implement all integrations

Product:

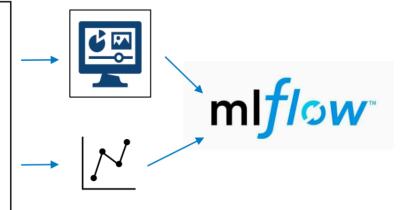
 Final product includes all integrations, is monitored and is updated according to contingency plan



Python model choice

Several implementations:

- Octave to python
- Scikit-learn (RandomForestClassifier, binary classification, QuantileRegressor, HistGrad ientBoostingRegressor)
- Ranger
- Quantile Forest
- XGBoost (quantile regression)



Criteria:

- License and maintenance of library
- Performance metrics such as: RMSE, skill, pinball losses, prediction interval coverage and width
- Integration with other frameworks (e.g. shap for explainability)
- Ease of use and training time
- Visualizations of forecast





Explainability for the weather room

