

Weekly meetings

- TA meetings: every Tuesday at 13:45 (online or on campus)
- Sprint planning meetings: every Tuesday at 15:00 (on campus)

Research Questions

- Main research question:

“How can a data-driven model be used to predict the likelihood that a train path request by an operator will be granted by ProRail based on historical data?”

- Sub-research questions:

A: *“How does time tolerance (earliest and latest preferred departure) affect the probability?”*

B: *“How good is the match between the request of the operator and the response of ProRail?”*

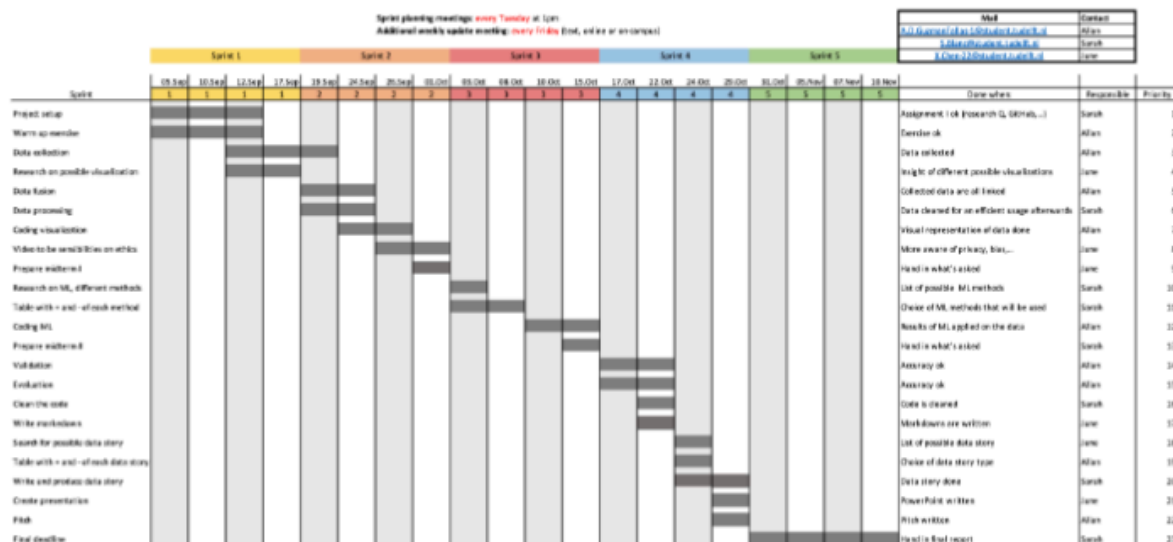
C: *“Do route structure and stabling/service needs influence outcomes?”*

D: *“Are operator, planning phase, and request timing (hour/day/week) predictive?”*

E: *(other questions?)*

Project Backlog

[Gantt diagram]



Changes/updates in the backlog:

- Date: September 16 2025
- New task(s):
- Completed task(s):
- Removed or deprioritized task(s):

- Reasons/context:
- Stakeholder feedback: [what Fulco said in meeting]
- Risks & challenges:
- Other notes: -

- Date: September 17 2025
- New task(s):
- Completed task(s):
- Removed or deprioritized task(s):
- Reasons/context:
- Stakeholder feedback: -
- Risks & challenges: unclear communication between us, course staff and ProRail
- Other notes: see minutes of meetings

- Date: September 23 2025
- New task(s): fix the sprint documentation
- Completed task(s): hand in milestone 1 project proposal, clean dataset
- Removed or deprioritized task(s):
- Reasons/context:
- Stakeholder feedback: meet with TA
- Risks & challenges: -
- Other notes: see minutes of meetings

Minutes of meetings

- **October 10** (on campus):
 1. Midterm presentation with professors, ProRail and TA.

- **October 7** (on campus):
 1. [TA meeting 3](#)
 2. Preparing for midterm presentation

- **September 30** (on campus):
 1. [TA meeting 2](#)
 2. Group meeting: sprint planning

- **September 24** (on campus):
 1. Discussion on feedback

- **September 23** (on campus):
 1. [Communication with TA.](#)

- **September 22** (online and on campus)
 1. Finishing up milestone 1.

- **September 17** (on campus):

1. Discussing how to clean the data.
- **September 16** (online and on campus):
 1. Meeting the group on campus.
 2. Online meeting with ProRail.

Current planned tech stack

1. Operating systems: MacOS and Windows
2. Server-Side Programming: Python
3. Frameworks and libraries
 - a. Machine learning techniques to predict probability
 - b. Data cleaning: pandas, numpy
 - c. Visualisation: matplotlib, seaborn
 - d. Machine learning models: scikit-learn (logistic regression, gradient boosting, random forests)
 - e. Model evaluation: built-in scikit metrics (AUC, precision recall, Brier score)
4. Techniques, iterative approach
 - a. Baseline model: logistic regression with simple engineered features (time-of-day, day-of-week, stops). This provides interpretability and a benchmark.
 - b. Tree-based models: Random Forest and Gradient Boosting to capture nonlinear relationships.
 - c. Advanced feature engineering: incorporating categorical encodings to group orders.
 - d. Lastly for evaluation: apply probability calibration (e.g. Platt scaling, isotonic regression) to ensure probability outputs.
5. Version control: GitHub
6. Development environment: VS Code and Jupyter Notebook
7. Documentation: **notebook in Github?**

Data cleaning

- Order data from operators, spanning over a period of three years
 - 80 input columns (e.g. train characteristics, preferred route and arrival time)
 - 30 additional columns (reaction from ProRail planners)

Rows

To see what values in the rows are useful, the client was consulted and it was discussed that certain values will not be used for the purpose of this model.

- Keep only last 'Ordernummer', all others that are the same, remove these
- Delete rows with Procestype == ANNULERINGEN
- Keep rows with reactie_type == AANGEBODEN or reactie_type == GEEN_AANBIEDING_MOGELIJK
- Delete completely empty rows

- Delete rows that do not have a reaction (reactie_type == empty)
- Delete rows with Aanvraagkanaal == handmatig
- Delete rows with Operator == NSR
- Delete rows with Aanvraag_Fase_Planproces == Jaardienst

Columns

The columns that are needed for the purpose of this research were kept, whereas the other ones were dropped. In this stage the reaction columns, which could hint the model that a request got an offer back, were removed - except for the 'answer column' (Reactie_type). This answer column will later be separated into a different dataset to train the model.

- Keep Tijdstempel
- Keep Ordernummer
- Keep Procestype

Requests

- Keep Aanvraag_vervoerder_verkorting
- Keep Aanvraag_Indienmoment
- Keep Aanvraag_Dienstregelingjaar
- Keep Aanvraag_Routelint (delete after we have created an extra column with information about stops)
- Keep Aanvraag_Eerste_Locatie_Dienstregelpunt
- Keep Aanvraag_Laatste_Locatie_Dienstregelpunt
- Keep Aanvraag_Eerste_Locatie_DienstregelpuntSpoor (?) - we think ProRail can always fulfill this request, but we can check it by keeping this column – maybe a creative way of checking “need for stabling”
- Keep Aanvraag_Eerste_Locatie_Vroegste_Vertrektijd
- Keep Aanvraag_Eerste_Locatie_Laatste_Vertrektijd
- Keep Aanvraag_Laatste_Locatie_Vroegste_Vertrektijd
- Keep Aanvraag_Laatste_Locatie_Laatste_Vertrektijd
- Keep Aanvraag_Eerste_Locatie_Behandelingen
- Keep Aanvraag_Laatste_Locatie_Behandelingen
- Keep Aanvraag_Eerste_Locatie_MaxLengte
- Keep Aanvraag_Eerste_Locatie_MaxSnelheid
- Keep Aanvraag_Eerste_Locatie_MaxGewicht
- Keep Aanvraag_Laatste_Locatie_Behandelingen (is information about stabling. If it contains “opstel” of “opstellen” then the operator wants a stabling service)

Reaction columns

- Keep Reactie_type
- Delete all other columns

Code

Aside from removing rows and columns in the first step, some data needs a different format - such as dates.

At last, the transformed columns are removed.

- Create extra column with number of requested stops (except start and end), derived from Aanvraag_routelint
- Only keep the first and last station of Aanvraag_routelint
- For Aanvraag_Indienmoment, make three columns which extract
 - Moment of day (per half hour)

- Day of week
 - Week of year
- For Aanvraag_Eerste_Locatie_Vroegste_Vertrektijd, make three columns which extract
 - Moment of day (per half hour)
 - Day of week
 - Week of year
- For Aanvraag_Eerste_Locatie_Laatste_Vertrektijd, make three columns which extract:
 - Moment of day (per half hour)
 - Day of week
 - Week of year
- Make a column which extracts the difference between the moment of day (per half hour) Aanvraag_Eerste_Locatie_Vroegste_Vertrektijd and the moment of day (per half hour) Aanvraag_Eerste_Locatie_Laatste_Vertrektijd
- For Aanvraag_Laatste_Locatie_Behandelingen, turn this into binary code to know whether they want service ("opstel")
- Delete column Aanvraag_Laatste_Locatie_Behandelingen
- Delete column Aanvraag_Routelint