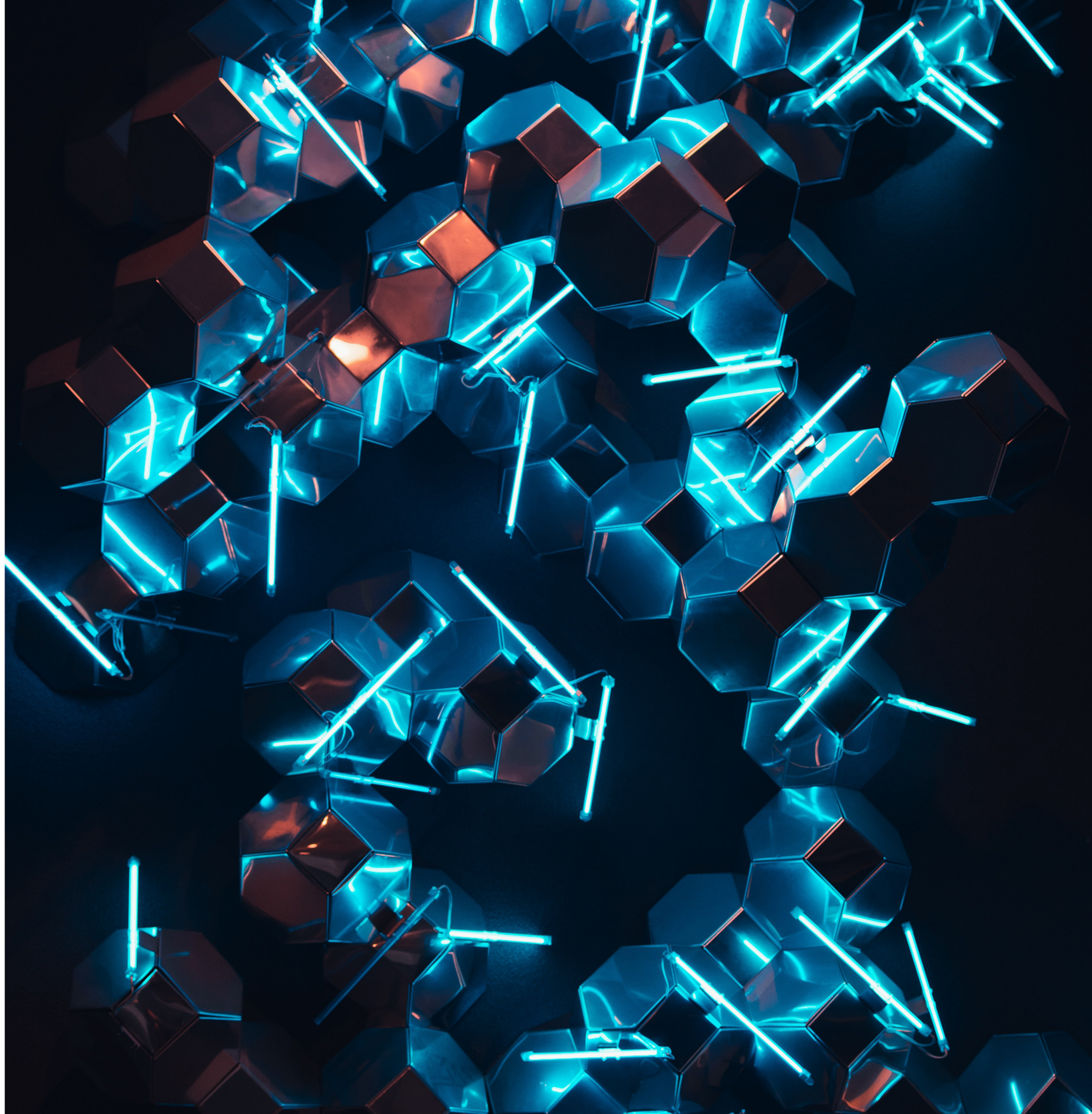
nPlan

Model Performance 101

March 2023



01
Contents

02
Background

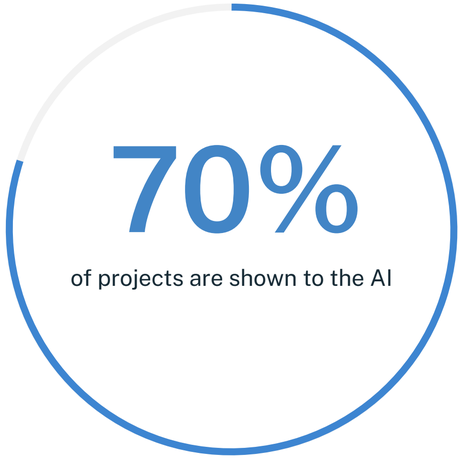
When evaluating Artificial Intelligence it is important to make sure that it is accurate. We pride ourselves on the rigour of our methods, and keep our models to very high standards.

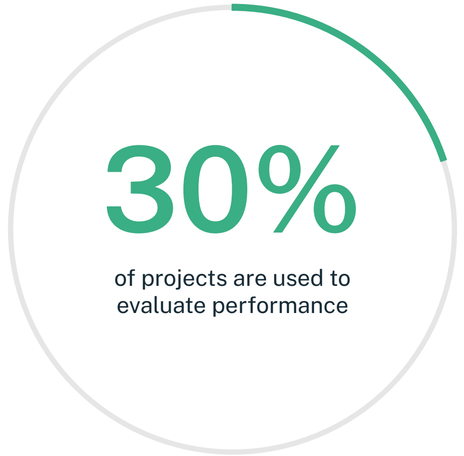
This document is designed to convey both these standards and rigour, as well as explain some of the metrics we use, in a way that does not require heavy statistical training, whilst still reporting real performance data.

How this report is generated

When we train our AI, **only 70% of projects are shown to the AI,** with the remaining 30% being used to evaluate performance. This guarantees that the AI can “generalise” well: generalisation is a property of machine learning to be able to make good predictions in contexts that it has not seen before. It does so by learning “patterns” rather than exact data.

Because we do this 70/30 project allocation randomly, we perform this test several times with different splits every time. In statistics this is called “cross-validation”, and is a way of making sure that we’re not just “getting lucky”. All the metrics and data that we include in this report have been verified in this way, on our entire dataset.





02
Background

The dataset

Whilst each client's specific performance and data remain private, here are a few details about our global dataset to give an idea of what sort of insights the AI is building internally. This information is accurate as of Feb 22nd 2023.

Please note that given the manner in which we query activities from the bottom up rather than top-down from the project, we do not provide estimates of the number of schedules for any industry or project type.

Instead, we are able to provide more detailed performance numbers for a range of specific sub-categories on request, after data has been shared with us.

Baselines used in this report

To measure model performance, we measure and report accuracy and calibration metrics from our ML model compared to a baseline of using a fixed distribution over all activities.

We use the same methodology as the best-in-class monte carlo methods (PERT and Log-Normal) distributions as baselines, since these are the most commonly used ones in practice.

Dataset Size

358,871,102
activities

539,569
project schedules

Dataset Composition (industries)

- Energy (nuclear and renewables)
- Oil & Gas
- Utilities
- Commercial buildings
- Construction
- Infrastructure (Rail/Road/ Airports etc)
- High-rise residential and commercial
- Transportation
- Ship building and repair
- Space Exploration
- Military and Logistics
- Research facilities (Particle Accelerators, Biochem facilities, Labs etc)

Project maturity

- Pre-investment
- Design and Engineering
- Construction
- Commissioning

Schedule detail levels

- Summary
- PMO/Owner review
- Tier 1 General Contractor
- Sub-contractor

03
Our AI understands projects

We decided to test our AI's building blocks (Large Language Models and Graph Neural Networks) - which look a lot like ChatGPT - to see what they understood about the projects they were learning from. We devised the hardest tasks that we could come up with for our models:

- **Automatically complete a schedule by filling in missing activities (think predictive text but for schedules)**
- **Break down summary activities into component options, and generate multiple ways of doing so**

We then tested this by showing experts (schedules, project managers, and our own in-house risk engineers) three generated options and a real one, and asked them to see if they could tell which one was real. **About 80% of the time the experts chose an option that was generated by our AI** instead of the real version of the schedule.

04
Predictive ability

Traditional Monte-Carlo based methods are typically based on picking a form of distribution to apply to activities in the schedule.

Our AI automatically generates these distributions with infinite granularity and no theoretical limitation on the length of distribution. This means that, unlike triangular distributions, we don't make the assumption that an activity "can never take more than 40% longer", which is a very common assumption made in standard QSRA processes.

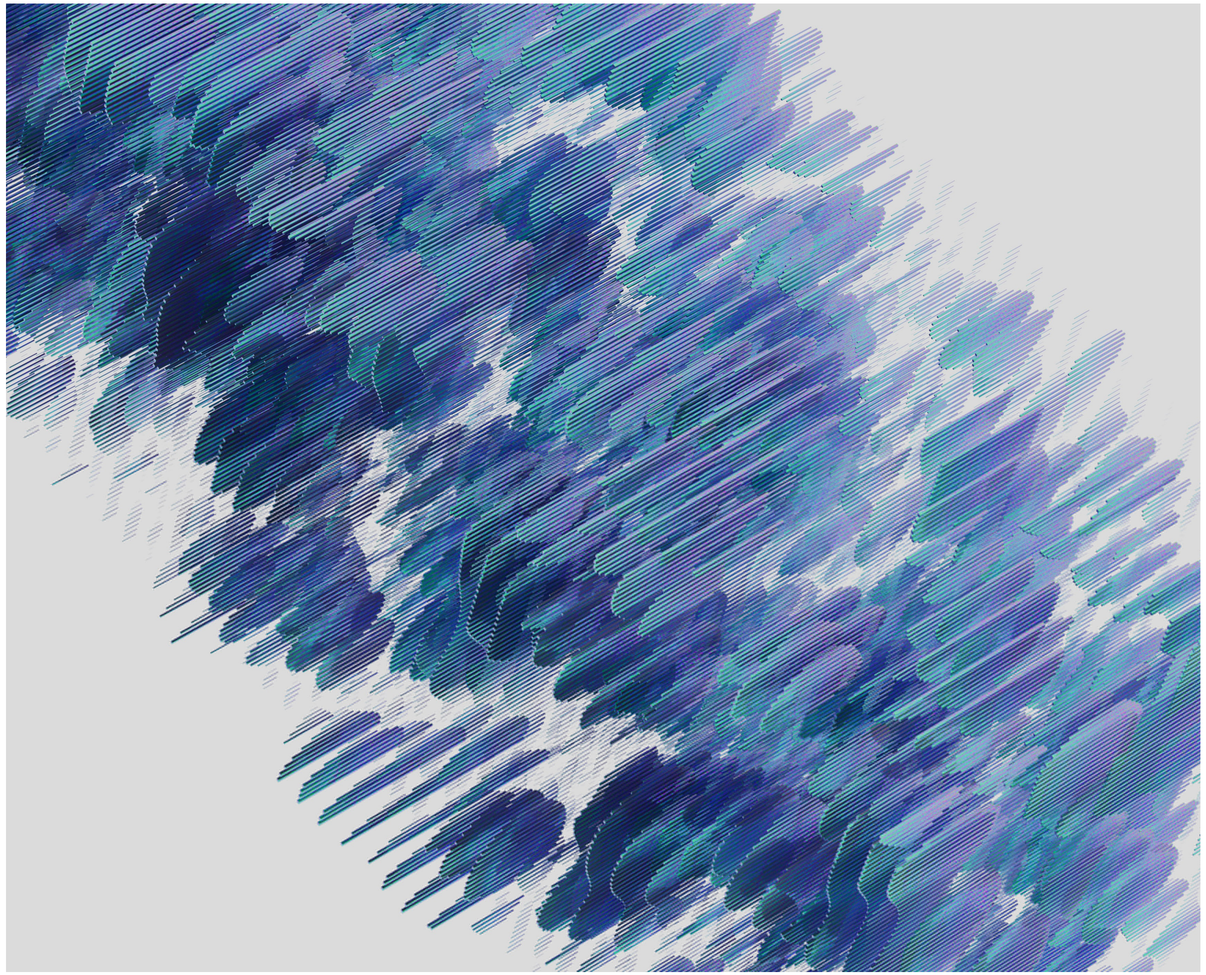
Activity-level delay forecasts

nPlan is **significantly better** at predicting activity-level distributions on several metrics, as shown in Table 1.

Method/forecast metric	CRPS	MAE	Likelihood
PERT	703.5	1514.1	0.012
Log-normal	105.2	237.3	0.01
GNN	64.1	106.3	0.44

Table 1: Activity-level performance compared to typical PERT methods: test set had 14,184 projects from various sectors and industries, approximately proportional to their representation in the full dataset as reported above.

04
Predictive ability



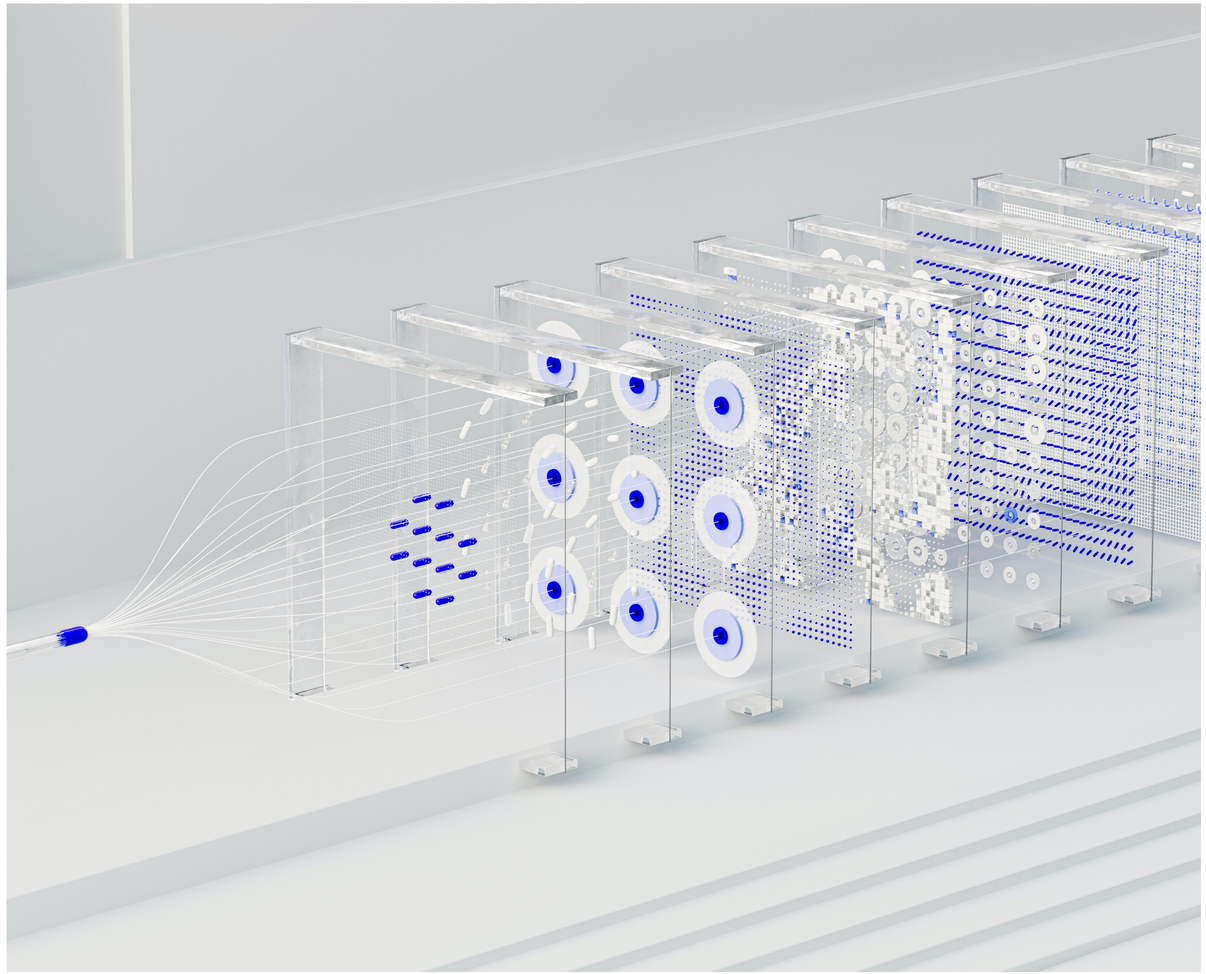
Rather than measuring the “correctness” of a particular single P-value of a forecast, Continuous Ranked Probability Score (CRPS) assesses the quality of the output distribution over its entire range, representing the overall quality of the learned forecast.

For task duration forecasts we include Mean Absolute Error (MAE) and Likelihood metrics in addition to CRPS to better gauge model performance. MAE is the average difference between any sampled duration from the forecast’s expectation and the actualised one, averaged over all schedules in the entire test set.

We report both CRPS and MAE in working hours, and a perfect score for each would be 0.

Likelihood is the probability density around the task’s actualised duration, averaged over the entire test set.

This metric measures local distribution quality and approximates the likelihood of sampling the duration for a task. The perfect score here would be 1.0, meaning that the model always forecasts the exact delay with full certainty.



04
Predictive ability

When we only look at delayed activities, this difference is even more obvious, with nPlan being able to identify **47.2% of all activities that are delayed by more than 50%.** PERT was not able to identify any delays of this kind whatsoever.

It is also to be noted that, given these are actualised activities, they had not been identified or mitigated at the time by project teams and/or any risk analysis methodologies that might have been employed at the time, meaning that **47.2% of these real-world delays are preventable using nPlan.**

Project-level delay forecasts

Once we combine the activity level forecasts we have a project-level outcome for which our AI produces a probability distribution.

We have shown elsewhere how traditional Monte-Carlo methods are flawed and biased towards underestimating uncertainty, and Table 1 shows that our AI is **both more specific and more calibrated in its predictions** than any PERT method.

[1]We consider a delay “identified” if a model captures the actualised delay with a probability larger than 30%

04
Predictive ability

We use the probability distributions outputted by the activity forecasting model on each activity to run Monte-Carlo simulations and estimate probability distributions for each project's end date.

This gives a forecasted probability distribution for the project completion date, which we then use to compute a forecasted probability distribution of relative project delay error:

$$\textit{delay multiplier} = \frac{\textit{forecasted project duration}}{\textit{actual project duration}}$$

Notice that a perfectly accurate forecast would produce a single delay multiplier with value 1.0.

We measure model accuracy to forecast project delays using common statistics over the distances from the observed true values to the sampled forecasted values using the tested method.

We report in Table 2 various P-values, and average of the delay multiplier distribution. For each of these, the closer to 1, the better. CRPS, calculated in the same way as for the activity-level analysis, is also reported; the closer to 0 the better.

The perfect values for median and mean are 1.0, whilst for CRPS the perfect score is 0.

Further, when we look at all projects in our test dataset that were delayed, our AI is able to **identify 47.9% of projects delayed by more than 30% from just the baseline schedule** (1514 of 13913 projects in our test dataset).

Method/forecast metric	CRPS	P10, P50, P90	Average
PERT	0.847	1.53, 2.04, 2.77	2.103
Log-normal	0.532	1.28, 1.63, 2.35	1.746
GNN	0.169	1.02, 1.14, 1.41	1.191

Table 2

05
Conclusion

We find that, in the entirety of our dataset, our AI is able to identify around half of the delayed activities, as well as half of the delayed projects.

It is important to note that, given that these delays are actual delays in a schedule, they were not forecasted or prevented by whichever process was in place at the time.

We therefore conclude that **about half of all the delays that were not prevented during these projects were identified by nPlan, and therefore potentially still preventable right from the outset of a project, when the baseline was created.**

For those delayed projects, we were also able to identify around half of the activities that caused those delays which were not prevented or mitigated by the project teams at the time.