University of Potsdam
Faculty of Science
Institute of Environmental Science and Geography
Institute of Physics and Astronomy
Climate, Earth, Water, & Sustainability

Master Thesis
for the award of the academic degree
Master of Science (M.Sc.)
at the University of Potsdam

Forecasting the Success of Environmental and Sustainability Activities in International Development Using Language Models

Potsdam, 4 February 2026

Submitted by:
Morgan Rivers
rivers@uni-potsdam.de
Matriculation No.: 829112

First reviewer: Prof. Christian Kuhlicke
Second reviewer: Dr. Ivan Kuznetzov

Abstract

Abstract in English

International aid and cooperation creates a profound difference in the rate of development in growing economies, improves the lives of the world's poorest, and often safeguards the environment and materially promotes sustainability. However, international aid has non-significant rates of failure in achieving its objectives. There have been few attempts in the literature to create models to explicitly predict the success of aid activities, and none focused on environmental outcomes. This thesis produces a forecasting system for the overall success of international aid activities at time of evaluation from the International Aid and Transparency Initiative (IATI) database, combining classical statistical methods with modern language model techniques. I apply novel techniques to improve prediction accuracy, including quantifiable information used in previous studies, averaging the success rates of semantically similar activities, and using the reasoning abilities and information gathering ability of large language models (LLMs) to improve forecasts. I first assess overall success ratings on a scale from approximately 1 to 6. Testing against the validation set, 300 later-starting activities in a dataset of 1,300 environmental and sustainability improving activities for 4 reporting organizations, the full forecasting system improves on the proxy for human forecasting skill baseline with a probability of forecasting which activity will have a higher rating from 65 % to 77 % [95% CI: 73 %, 81 %]. The system also explains 26 % [95% CI: 14 %, 37 %] of the variance in true ratings ($R^2 = 0.26$) in the validation set, compared to 10 % for a "pick the most common rating from reporting org" baseline, and 0% for the human forecasting skill proxy. I also construct a novel aggregate outcome cost-effectiveness metric, as an independent check that ratings correlate with better success metrics. Using similar methods, my forecasting system predicts which activity will have a higher cost-effectiveness among random pairs of activities in the IATI database with a success probability of 60 % [95% CI: 55 %, 66 %]. I find a weak, positive correlation between ratings and cost-effectiveness of outcomes. However, the correlation in model forecasts is much higher, indicating that the learnable signal is similar between rating and cost-effectiveness prediction. I also release a freely available dataset of LLM-generated activity grades, summaries, success ratings, and various other quantitative activity outcomes and extracted information for 1,800 IATI activities. This work lays the foundation to improve decision making for a wide range of initiatives and policies in developing countries and also in other data-rich institutional contexts.

Abstract auf Deutsch

Will do, once abstract is finalized

Table of Contents

1 Background: Prior work in LLM Forecasting and Forecasting International Aid Impact

1.1 Introduction

Proposal I set out to analyze the applicability and composability of machine learning techniques in forecasting the success of international aid activities using what could be known about the activities at the approval stage. The standard approach in the literature in assessing aid impact and success in environmental domains has historically involved econometric techniques, linear regression models, and occasionally more modern nonlinear models such as Random Forest or XGBoost (**goldembergMindingGapAid2025**). However, a thorough assessment of methods to forecast activity success in this domain has yet to be conducted.

This thesis proposes the use of Large Language Models (LLMs) and statistical models to implement judgemental forecasting to predict how effective interventions will be in the context of international aid activities affecting the environment. I obtain metadata and pdf files from thousands of activities, and separate each record into information about the activity available at approval and evaluation information about how successful the activity was. I use this data to answer the following research questions in this work:

1. How do LLM forecasting methods compare to human proxies and statistical models in forecasting international aid overall success ratings and quantitative outcomes?
2. Do forecasting methods using state-of-the-art natural language processing methods meaningfully improve on simpler baseline forecasting heuristics?
3. How do differing methods of combining LLM and statistical forecasting compare in this domain?
4. What methods improve the accuracy of free-form (qualitative) forecasts in this domain?
5. What aspects of the activity available in my dataset at the beginning of the activity lead to higher or lower ratings?
6. How does forecasting quantitative activity outcomes compare to forecasting ratings?

I use language models to mimic the reasoning and data gathering skills of trained forecasters, in an attempt to replicate the success at using judgemental forecasting from language models in geopolitical forecasting to the adjacent domain of forecasting development cooperation outcomes affecting the environment. I then compare this approach to standard statistical methods, and utilize novel techniques which combine statistical and judgemental

AI forecasting, to investigate how the advantages of both can be put to best use in improved forecasting of international aid activity outcomes. Given the difficulty of field testing ideas, policymakers and funding agencies often rely on expert forecasts on how an intervention will meet its intended goals to select which interventions will be implemented (Hewitt et al., n.d.). Replacing or augmenting that advisory role could greatly improve decision making in the context of international aid.

This method differs in two key ways from prior literature, which have analyzed international aid evaluations. The first difference is that while many works in the literature have attempted to assess correlations between quantitative features and aid evaluation ratings, they have not focused on what knowledge would be available at approval for the activity, and do not assess out-of-time generalization of these correlations. In this work, I assess out-of-time generalization of feature importance. This is critical, because in order to improve aid decision making, one must assess the ability of models to forecast the outcomes of future interventions, not simply statically analyze a corpus of past activities. The second difference is that in addition to standard statistical models, I implement judgemental AI forecasting, as a supplement to standard statistical models.

I will briefly review current progress in event outcome prediction in developmental aid and cooperation interventions affecting the environment, and then discuss progress with LLMs in adjacent domains.

Within the domain of LLM use, there have been early attempts at using them to improve decision making for the environment. A recent tool called "climsight" summarizes and aggregates information about climate adaptation and mitigation (Koldunov and Jung, 2024), but stops short of making forecasts towards adaptation. Machine learning and LLMs have been used to collect over 80,000 articles about climate adaptation and provide analysis about which areas of implementation are lacking and point out gaps in attention towards promising categories of policies.

Limited work has also been done using LLMs such as ChatGPT-4 (GPT-4) to serve as data sources for policy deliberation and multi-criteria assessment of climate and sustainability interventions, finding GPT-4 is in rough agreement with the policy rankings of human experts for the expected outcomes (Bina et al., 2025). However, very little is done to improve on GPT-4's abilities, the assessment was made on only a few dozen generic policy examples, and no attempt was made to compare outcomes between these policies and real-world outcomes. Despite these limitations, the findings are promising. For multiple criteria decision making (MCDM), GPT-4 provided a useful collaborative starting point, eased the process of considering multiple criteria effectively, and aided policy deliberation on climate change and sustainability.

One attempt which focused on specific outcomes of activities found their model using "embeddings" of LLMs could explain 70% of the variance of the unexplained residual from

control variables on relevant country-level sector outcomes from the World Bank World Development Indicators, and assessed the performance of nonlinear models including the random forest model used in this work. However, they include features that could not be known at the beginning of the activity (e.g. actual duration), and do not assess out-of-time generalization, instead splitting randomly within the dataset, nor do they explicitly assess prediction performance for ratings. Furthermore, in replicating and extending their method, I found their model worse than random chance at out-of-time prediction of outcomes, indicating severe issues with out-of-time generalization, which I could only explain by what I discovered to be validation and test set data leakage into their training set.

2 Methods

This thesis implements an LLM-based forecasting method for ratings, a statistical model for ratings free of any LLM features, a system combining the best LLM method and the best statistical model, with the advantages of both, and finally a modified system designed to predict categories of quantitative activity outcomes extracted from activity evaluation documents. The system is built forecasting what the evaluation results will be for thousands of IATI records containing both a pre-intervention description of the activity, as well as an ex-post evaluation of the results. To do so, thousands of pdfs were downloaded, ranked from most to least relevant for forecasting future outcomes or evaluating the end result of the activities, had their pages ranked and graded for relevance to the task, had quantitative and qualitative descriptions and results transcribed into a unified format.

2.1 Data Sources

After considering several data sources for prediction, including the OpenAlex publication repository of peer-reviewed evaluation documents and abstracts, the IDEAL database of ex-post evaluations, the 3ie development database, I decided to use the IATI database, due to its substantial quantity of information available in textual format and extractable from the database records, and its sheer size. While ex-post evaluations may provide sufficient information to describe the activity, it may introduce "future leakage" to rely on language models to completely remove information about the eventual outcome. Furthermore, although many millions of evaluations are available, it proved difficult to reliably identify and de-duplicate academic papers regarding evaluations of environmental interventions. The IDEAL database and 3ie were in the dozens or hundreds of records for environmental topics.

The IATI database has reliable start and end dates, and typically several recorded outcomes, and usually an overall evaluation rating on a six point scale within linked evaluation pdfs. It also reliably marks the reporting organization, allowing for an intelligent unification of the rating scales, and sometimes provides a "results" section where outcomes of an activity can sometimes be found. It is quite common for several activity information documents to be uploaded near the beginning of the activity, and several years later, at least one ex-post evaluation of the activity is uploaded as well, or results of key quantitative outcomes are sometimes directly recorded in the IATI database. The status of the activity is extremely commonly reported, including if it is in the planning or completion/finalization stages, which is helpful information for forecasting.

The downside of the IATI database is it is sometimes inconsistent between reporting organizations as to how the data are filled in, and the format of documents was not always PDF, requiring conversion scripts. Also, many download links were not functioning or required custom web-scraping scripts to properly extract project documents in pdf format from the original websites where project documents were hosted. Dates of documents and especially planned start or end dates, or actual start or end dates, were often missing, leading to frequent exclusion of projects. Furthermore, approximately 30% of IATI activities do not have an activity category code, leading to a further exclusion of environmental or sustainability related activities.

In addition to IATI, I also analyzed the AidData database, which has laboriously double-coded data entry, which introduces less error than the LLM and regex data extraction techniques I used. However, I find that the delay in hand-coding the ratings and results leads to a paucity of recent environmental activities, and the lack of document links in AidData was a key downside that made it difficult to use AidData. After downloading AidData, I found the overlap between AidData and my training set was less than 30%, which meant any data enrichment would run into issues of data availability for the remaining 70%.

2.2 Data Filtering

IATI records for prediction

Out of the approximately 800,000 international aid activities recorded in IATI, I first reduced the set of activities of interest to 7,575 records which aimed to improving the environment, sustainability, or climate adaptation in a developing country or countries, had an appraisal/intervention description document, and an outcome evaluation or progress report document, both of which could be converted to PDF format. Links to these documents were then downloaded where possible (see the next section for details).

Once documents were downloaded and converted to pdf format, activities were further

filtered so that they had at least one document describing the activity, and had a metadata date before 1/4 of the activity implementation period, as well as at least one ex-ante activity at least 3/4 through the activity period. The latest activity document also had to have a metadata date at least one year before the earliest evaluation document. An exception was project appraisal documents from the World Bank, or Project Information Documents, which were found to reliably not leak future information, and this was judged to be more trustworthy than extracting the creation date embedded in the activity document. Activities not meeting these requirements were also excluded, leaving 3,225 activities.

After passing all these filters, I finally attempted to extract the activity rating from the evaluation document using two separate methods. Because rating tendencies are systematically different for different reporting orgs, I needed sufficient data for training, validation, and testing for all organizations. I also restricted activities to those that were marked as "completed" in order to ensure comparability between rating scales, as the only activities not marked as “completed” were relatively recent and would dominate the held-out test set. I determined there were sufficient data for four reporting organizations: The World Bank (957 activities), BMZ/KFW/GIZ (240 activities), the Asian Development Bank (ADB) (156 activities) and the UK Foreign Commonwealth and Development Office (FCDO) (127 activities).

The activity filtering for the topic was done by-hand to filter only those activities relating to improving the environment or sustainability. These were:

- 14015: Water resources conservation (including data collection)
- 14020: Water supply and sanitation - large systems
- 14021: Water supply - large systems
- 14022: Sanitation - large systems
- 14032: Basic sanitation
- 14050: Waste management/disposal
- 23110: Energy policy and administrative management
- 23111: Energy sector policy, planning and administration
- 23112: Energy regulation
- 23183: Energy conservation and demand-side efficiency
- 23210: Energy generation, renewable sources - multiple technologies
- 23220: Hydro-electric power plants
- 23230: Solar energy for centralised grids
- 23231: Solar energy for isolated grids and standalone systems
- 23232: Solar energy - thermal applications
- 23240: Wind energy
- 23250: Marine energy
- 23260: Geothermal energy

- 23270: Biofuel-fired power plants
- 23350: Fossil fuel electric power plants with carbon capture and storage (CCS)
- 23360: Non-renewable waste-fired electric power plants
- 23410: Hybrid energy electric power plants
- 23510: Nuclear energy electric power plants and nuclear safety
- 23610: Heat plants
- 23630: Electric power transmission and distribution (centralised grids)
- 23631: Electric power transmission and distribution (isolated mini-grids)
- 23642: Electric mobility infrastructures
- 31130: Agricultural land resources
- 31210: Forestry policy and administrative management
- 31220: Forestry development
- 31281: Forestry education/training
- 31282: Forestry research
- 31291: Forestry services
- 32174: Clean cooking appliances manufacturing
- 41010: Environmental policy and administrative management
- 41020: Biosphere protection
- 41030: Biodiversity
- 41081: Environmental education/training
- 41082: Environmental research

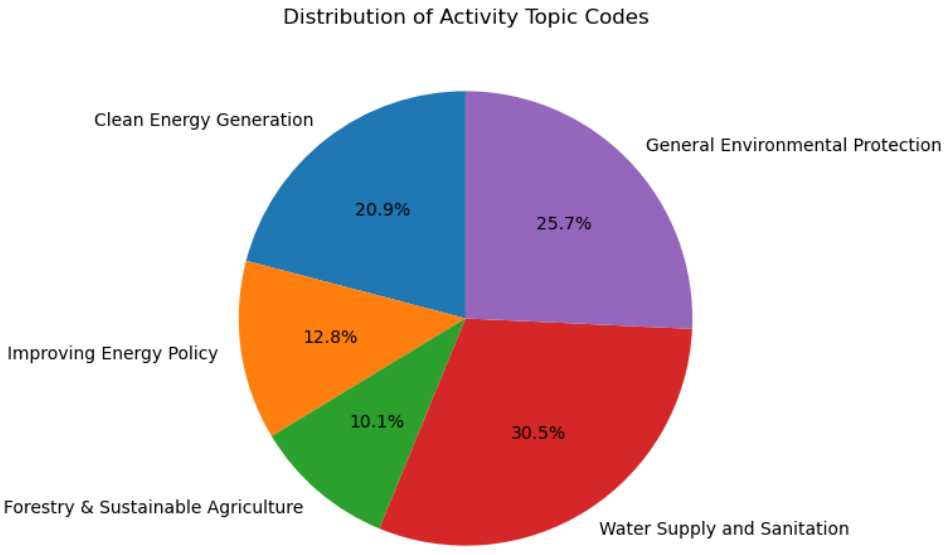

Figure 1: The split of topics analyzed from the dataset.

Document-related Filtering

The IATI database contains a collection of thousands of links to pdfs, word documents,

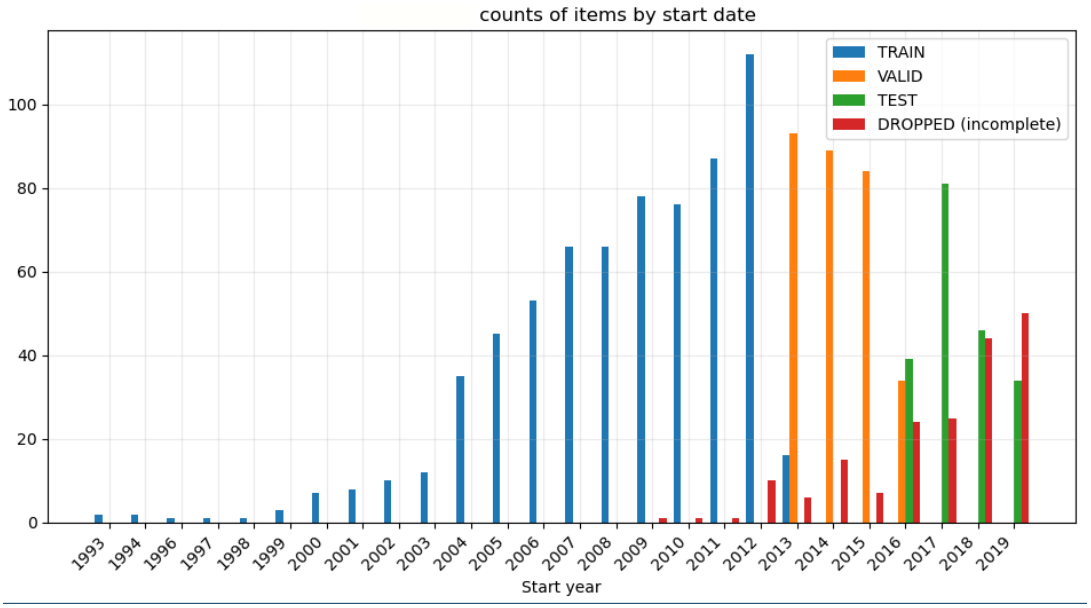

Figure 2: The activities included for forecasting ratings, with the splits by count year. Incomplete activities, shown in red, were not used for prediction. Activity ids used for outcome prediction were given the same temporal boundaries.

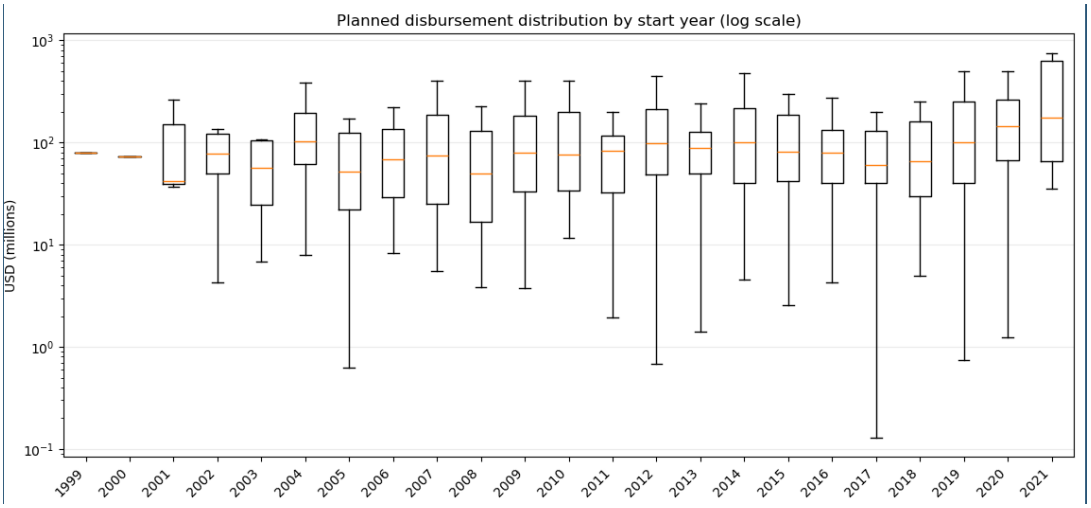

Figure 3: Total disbursements for IATI activities used for ratings, by year. There is no clear trend over time for activity size in the database.

html documents, and other document formats. These were first automatically converted to pdf format via a custom python script, and subsequently needed to pass several criteria before being used as documents for forecasting.

I first wrote a script that directly downloaded these and converted them to pdf format. Next, I look at the pdf metadata date, and determine the creation date of the pdf files. I find this is more often closer to the date of the specific activity description document or activity evaluation document (as determined by reading the document) than metadata at the url indicating upload date, or the date entered in IATI for the document. UNDP results had the URL specifically included in the JSON payload populating their website,

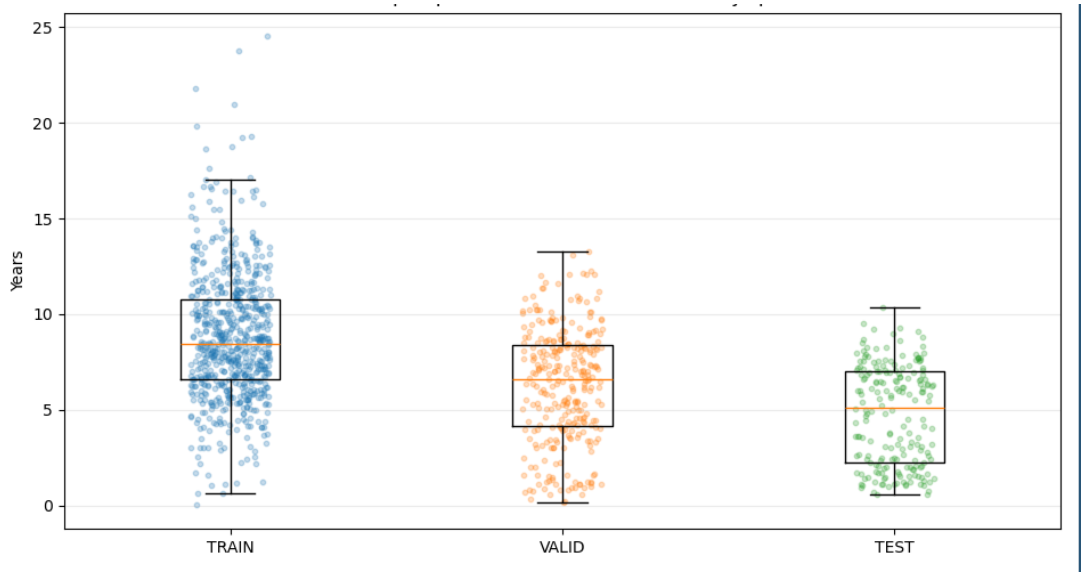

Figure 4: The durations of activities per split. More recently starting activities tend to be shorter, as they have not yet had time to complete and be evaluated. The out-of-distribution nature of validation and test sets increase the challenge of generalizing patterns from the training data.

so the latest year indicated in that evaluation payload was used instead for the UNDP results.

All documents are tagged in IATI with one or more of the following tags per document: “Pre- and post-project impact appraisal”, “Objectives / Purpose of activity”, “Intended ultimate beneficiaries”, “Conditions”, “Budget”, “Summary information about contract”, “Review of project performance and evaluation”, “Results, outcomes and outputs”, “Memorandum of understanding (If agreed by all parties)”, “Tender”, or “Contract”.

I mark documents with “Objectives / Purpose of activity” or “Summary information about contract”, tags as preliminary “baseline” documents - those representing information about the activity before it begins. Documents with “Review of project performance and evaluation” or “Pre- and post-project impact appraisal” tags are marked as preliminary evaluation documents. In order to provide sufficient information to forecast with and sufficient information to evaluate that forecast, I require at least one “baseline” document and at least one “outcome” document per activity, with the baseline document at least one year prior to the evaluation document (based on the uploaded document metadata date). I also require that the activity status code is not “Pipeline/identification”. Instead, activities are allowed to be in implementation, finalization, closed, cancelled, or suspended, such that either a final or preliminary evaluation document is possible.

I filtered further to ensure that all activity document labels which were "Conditions", "Budget", "Tender", "Contract" with no other tags were excluded, as these were typically purely legal context, often containing very little evaluation or useful additional activity information.

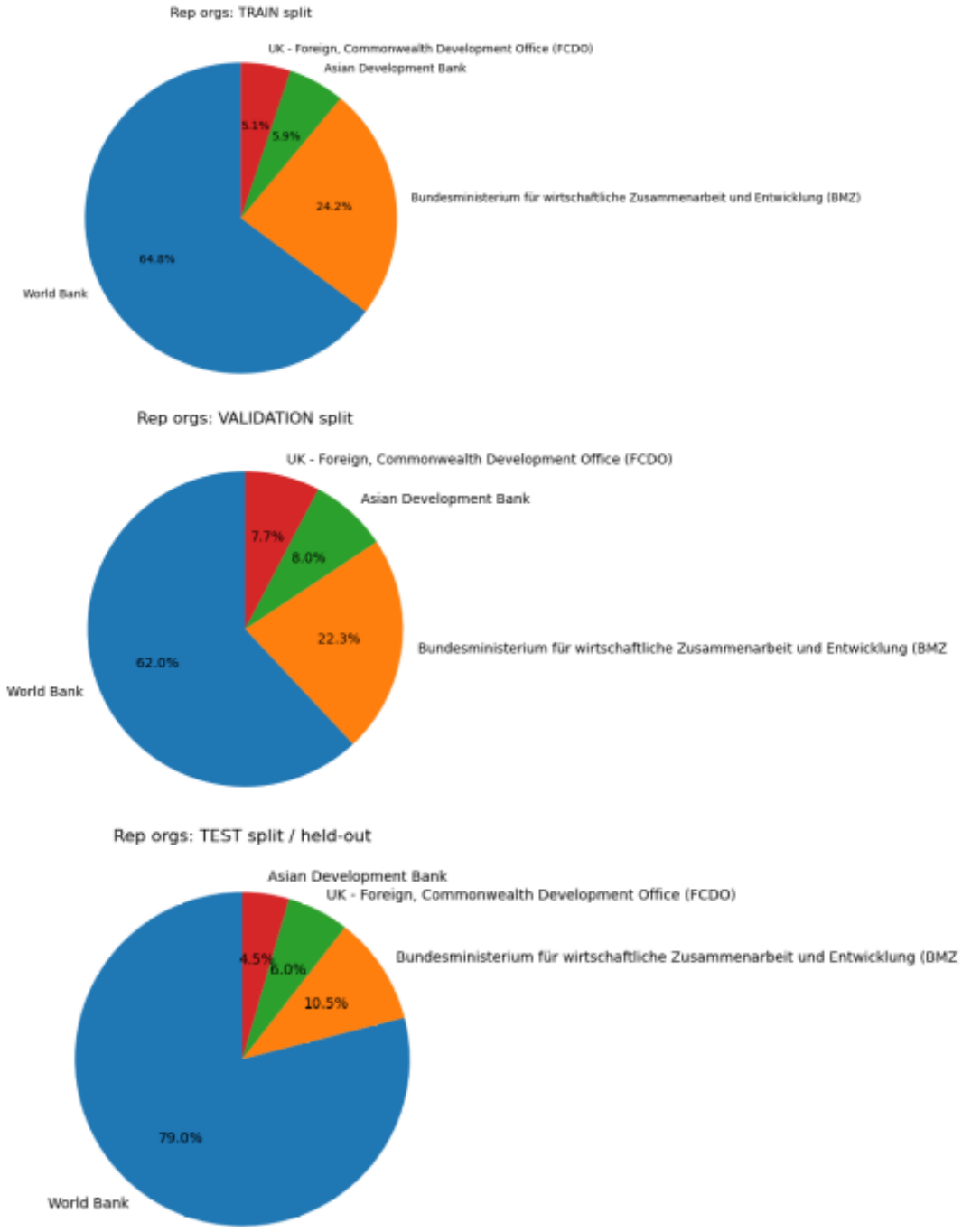

Figure 5: The breakdown of reporting orgs in the dataset which were used for training, validation and testing. The validation and test set are in this way significantly out-of-distribution.

‘’

The date for the documents were determined using (in descending preference where available) the pdf's "created on" or "last modified" in its metadata, or the date indicated in the IATI record for the document. This ordering preference was determined as the PDF metadata dates were found to more reliably match the stated date of authorship of the documents better than the "IATI date" recorded within the activity record. These dates were usually available in the metadata of the original PDF, ODT, DOC, or DOCX file. Experimenting with different date options revealed that out of 400 randomly selected PDFs, the closest available date to the true date of authoring the document was the "created by" date, then the "last modified" date, then the "IATI date". The median difference for the date when selecting this ordering was 22 days different than the date indicated in the document itself as determined by feeding the first 3 pages of each of those 400 PDF documents to *gemini-2.5-flash*.

To ensure pdf metadata dates were appropriate, an analysis was undergone to ensure the procedure for selecting the date of activity documents was valid. If the date of the activity is too early, it could lead to documents authored well after the project start leaking future information. To test this, 400 random pdf documents downloaded were uploaded to gemini and the pdf metadata dates were inspected. PDF metadata dates were discovered to have a median difference of 22 days from the date indicated on the document as extracted by gemini. It was discovered that while approximately 10% of the dates were more than a year after the actual date indicated on the document (such that an activity was actually authored earlier than the start date of the activity), a concerning 0.5% of documents had a creation pdf metadata date later than the date extracted directly from the document.

In order to ensure the forecasts were all based on project information available only roughly at the beginning of the activity, a search was undergone through the information available to the model when forecasting to ensure the forecasting was based only on what could have been known at the beginning of the activity. Approximately 10 activities with pdf metadata dates more than a year earlier than the true authoring of activity documents would be expected, based on the 0.5% rate of ">1 year too early" errors from the date analysis.

To prevent any information leakage, which could be due to incorrect dates as well as incorrect marking of the start date of the activity in IATI, or significant progress being made within the first quarter of the activity where documents are allowed, approximately 40 random chatgpt-generated activity summaries (see the next section) were inspected, with none indicating advanced progress, indicating less than 2.5% of activities should be of concern. A selective search for phrases revealed some activities had made clear progress on targeted outcomes. Consequently, a python script with 6,800 separate search terms was used to further search for inappropriate documents. Exact string search terms were made, with variants of phrases including, "on track", "ongoing project has been performing", "ongoing project is performing", "already made considerable progress", "key milestones

already achieved", "significant progress had been made", "the programme has already made considerable progress", etc. This led to the review of approximately 150 additional activities, and the discovery of 21 activities with clear progress on key project milestones. Progress such as the formation of planning committees or initial disbursements of funds to the implementing organization were not considered grounds for exclusion, given that these milestones are unlikely to be substantially informative. However, extension activities or Phase II / Phase III activities were not excluded, unless significant progress had already been made on the extension or phase being evaluated.

In order to properly extract accurate overall success ratings for each activity and useful textual information about the project for forecasting, I processed each pdf document using the following data processing pipeline:

2.3 Preliminary Data Processing

All document pages had their rotation detected, and were rotated to vertical before processing via the Gemini API. Documents with ".odt", ".doc", or ".docx" extensions were converted to pdfs with a custom script. The pages when converted to pdfs were counted and zero-page documents were excluded.

1. Ranking documents Documents were ranked from most to least useful for forecasting the outcome, or evaluating the results, respectively. *gemini-2.5-flash* structured output with direct pdf input was used to make the rankings. Only documents with c- or better grades on a grading scale from a+ to f were considered for the next stage. Also, the documents were ranked from most to least informative for forecasting among the baseline documents, and most to least valuable for ex-post evaluation among the outcome documents. Baseline documents that were closest to the activity start, and the latest outcome documents were preferenced. Documents with sufficient detail but not excessive lengths, such as executive summaries, were prioritized. Documents that were duplicates in a non-English language were excluded if the equivalent was available in English. For outcomes, if there were multiple progress reports, all the earlier ones were excluded and only the latest were kept in the rankings. After ranking, 2,312 documents had sufficiently informative activity information and activity evaluation documents.

2. Categorizing pages within documents The highest ranking documents were then split into 3-page chunks. Each 3-page chunk was sent in pdf form to *gemini-2.5-flash*. The pages were categorized differently based on whether the document was a baseline or outcome. Categories for outcomes allowed retrieval based on whether final evaluation in quantitative or qualitative form are present on the page, deviations from plans or other types of outcomes were detailed, or if the pages were simply overviews of the activity. Specifically, the allowed categorizations were "condensed summary", "sub activities

outlined", "detailed implementation plans", "broad objectives", "possible outcomes", "quantitative targets", "qualitative targets", "risks as word or numeric", "risks or dangers generally", "plans to address key risks", "positive indicators", "progress reports", "similar cases outcomes", "implementation context country", "contextual challenges", "financing details", "budget and legal", "who implements", "whether part of larger program", "partner identity or skill", "whether skin in the game", "other stakeholder engagement", or "activity monitoring details" for baseline document pages, and "expected outcomes", "deviation from plans", "preliminary results", "final outcomes", "delays or early completion", "over or under spending", "overview as was planned", or "unrelated to evaluation" for outcome document pages. Only one category choice among these was possible per page.

In order to exclude irrelevant pages, the pages were also given a second category, for outcome document pages as "glossary", "blank page", "table of contents", "outcome evaluation", "activity description", "references", or "other", and for baseline document pages the same categories were options, in addition to "core activities", "theory of change", "targets", "broader context", and "preliminary results". Only one category choice among these was possible per page.

3. Extracting Ratings Two separate methods were used to extract rankings. The first method sent each individual outcome page ranked above 7/10 for relevance to evaluation, or with a "quantitative targets" categorization, to *gemini-2.5-flash* to extract any overall ratings, and a second script summarized the overall ratings into a single value for the document. However, this was often insufficient to capture the overall ratings. Another "fallback" script involved a custom generated word search with approximately 500 different rephrasings of "overall rating", "final result", "synthesized score", etc, in English, and searched the pdfs directly for an exact match on those terms, prioritizing pages with one or more exact text matches of such terms. Otherwise, if such words could not be found, the earliest pages in the document which were not categorized as "blank page", "appendix", "glossary", "table of contents", "references", or "activity description" were included and *gemini-2.5-flash* was queried to extract the overall rating from the documents.

For BMZ/GIZ/KFW documents, activity baseline documents were extremely rare. For this reason, the evaluation document was treated as a baseline document for the purposes of forecasting activity success. Categorization for these evaluations also was via the "baseline" document method described above. When grading or summarizing the features of the evaluation document, *gemini-2.5-flash* was instructed to only describe what could have been known at the beginning of the activity, and to under no circumstances reveal the final outcome of the activity.

4. Interpreting Ratings Ratings were reported both with the rating itself, as well as a maximum and minimum possible rating. The World Bank rating scale from 1 ("Highly Unsatisfactory") to 6 ("Highly Satisfactory") was used as the template rating, and other

ratings were attempted to match against this scale. Notably, BMZ/GIZ/KFW ratings were inverted to reach this scale. A "Satisfactory" score was considered equivalent to scores such as "successful", "On Track", or "met expectations". Scores listed as percentages or fractions were re-scaled to the 1 to 6 scale as well. In order to ensure ratings were fairly compared, only the top four most common organizations with ratings were included for training and validation of the forecasting system.

2.3.1 Outcome Extraction

In addition to extracting ratings, a similar approach was used with *gemini-2.5-flash* to extract quantitative outcomes. I extracted quantitative outcomes from all pages which were categorized as outcomes, and marked as containing quantitative information. Unlike with ratings, I did not limit to the top 4 reporting organizations, as reporting ratings is more susceptible to between-reporting-organization variation and gaming than the reportable quantitative outcomes of projects.

Once these PDF pages were extracted, I employed a combination of manual examination of the extracted outcomes and a report of common bigrams to identify outcome variables that could be compared between projects. For each common outcome category, I came up with a list of words and phrases that would commonly match reports of these outcome variables in the description, as well as appropriate units.

Keyword- and unit-based outcome parsers

For each outcome category, I implemented a dedicated parser that scans the extracted quantitative description, baseline, target, outcome, and units. Each parser follows the same general pipeline: (i) *filtering* by description keywords and unit constraints, (ii) *sanitization* of numeric values (e.g., dropping negative sentinels or implausible magnitudes), (iii) *normalization* to a canonical unit (e.g., hectares, tonnes, people), and (iv) *aggregation* to a single activity-level outcome.

Comparable outcome categories

Using manual inspection and frequency statistics over common bigrams, I defined a set of outcome categories that recur across evaluations and are interpretable across projects. The final set included:

- **Cost–benefit ratios (B/C):** benefit-cost ratio outcomes.
- **Rates of return:** economic rate of return (ERR/EIRR) and financial rate of return (FRR/FIRR), in percent.
- **Emissions reductions:** CO_2 or CO_2e reductions (total or per-year) in tonnes.

- **Water and sanitation connections:** counts of service connections, either new or repaired.
- **Pollution load removed:** wastewater pollutant load reductions (e.g., BOD/COD/nutrients), in tonnes and categorized by time basis (total, per-year, per-day).
- **Forest indicators:** trees/seedlings planted (counts) and area-based forest outcomes (reforested, under management, protected) in hectares.
- **Irrigation outcomes:** increases in irrigated area (hectares), computed as a positive increase relative to baseline where available.
- **Energy outcomes:** installed generation capacity, in MW (or occasionally GWh where the source reported capacity in energy units).
- **Air quality (PM2.5):** PM2.5 reductions reported as concentration, emissions, or percent, kept as separate distributions.
- **Clean cooking stoves:** counts of stoves distributed/installed.
- **Agricultural yields:** yield increases expressed either as level changes (normalized to tonnes per hectare when possible) or as percent increases.

Each category required a custom python script, primarily because evaluation reports often contain multiple related indicators (e.g., component-level versus project-level rates of return) and frequently use heterogeneous units or phrasing.

Across all outcome categories, I used custom Python scripts to identify activity-level outcomes. Indicators were detected via keyword and unit constraints, implausible or non-numeric values were dropped, and quantities were standardized to canonical units (e.g., people, tonnes, hectares, MW, t/ha). Domain-specific filters reduced false positives (e.g., wastewater context for pollution loads, water context for connections, agriculture context for yields), textual multipliers and unit abbreviations were parsed to normalize magnitudes, and time bases (total vs. annual) were inferred.

While I originally simply took the mean of extracted outcomes, I found the extracted values were often error prone, so instead I took all matching values and compiled them, sending them to *gemini-2.5-flash* for aggregation into a final baseline, target, and outcome value where available for all outcome categories and activities.

Finally, I used the total disbursement for the activity reported by IATI and determined a rough estimate of the USD per unit outcome (see "Splitting Disbursements" below), except for Benefit-Cost Ratios, Rates of Return, and agricultural yield outcomes. I found on investigation that most outcomes were approximately log-normally distributed. I took the log10 of all categories except Benefit-Cost ratios, rates of return, and agricultural yields.

Splitting disbursements

Unfortunately, I did not have access to outcome-level funding splits from the IATI database. In order to roughly represent the fact that dollar-per-unit spending can be allocated across several outcomes, I wrote a custom algorithm to evenly allocate total activity expenditures to what are usually distinctly funded outcomes. My procedure assigns each activity's total expenditure across the outcome components it reports, so later cost-per-unit calculations do not implicitly treat multi-outcome activities as having multiple full budgets. Benefit/cost ratios and economic and financial rate of return are excluded from monetary allocation, and other outcomes are eligible for splitting. To avoid double-counting when two indicators are simply alternative measurements of the same underlying result, closely related indicators are first grouped into shared conceptual buckets, such as pairing protected area with area under management, pairing different yield-increase measures, and pairing tree planting with reforested area.

Once the components are bucketed, the algorithm gives each bucket an equal share of the activity's allocatable funding. Every component inside a bucket inherits that same share, meaning components that are "alternative measures of the same thing" share one portion of the allocations rather than each taking a slice.

Carbon dioxide reductions are handled as a special case because they can act as a summary metric that overlaps with other mitigation outputs. If CO_2 reductions are reported without any closely linked mitigation outputs (such as improved stoves, added generation capacity, or trees planted), then CO_2 reductions receive an equal share like any other bucket. If CO_2 reductions are reported alongside any of those linked outputs, it inherits the combined allocation already assigned to the linked outputs present for that activity. This prevents CO_2 from inflating allocated spending when it is a co-reported consequence of other outcomes.

2.4 Baseline Methods

Three relatively simple baseline methods were attempted, to ensure the relatively complex and expensive LLM-based methods are better than simpler approaches. I choose three simple baseline methods, in order to ensure the forecasts were significantly better than the baseline methods for activity success forecasting.

Prediction baseline: always predict the most common rating for the reporting organization

This baseline technique provides a sanity check that more sophisticated methods are worthwhile. Because the prediction task is inherently difficult with much of the variation in outcomes unable to be forecasted at the outset of the activity, this is a relatively strong baseline.

Prediction baseline: Ridge Regression with Reporting Organization and Risk Score

As a rough proxy for the forecasting ability of human evaluators, a ridge regression model was trained given both the llm grade for the degree of risk assessed at the beginning of the project, as well as the dummy variable representing which organization was doing the rating. While by construction the full forecasting system also includes these features and thus with sufficient regularization must perform better than this baseline, it does provide some insight into how skilled aid evaluators are at assessing the overall risk of a low rating. This score was only calculated for activities where the llm was able to successfully extract the grade for how risky the project is from the baseline documents. This method suffers from potential inconsistency in llms extracting the overall risk from baseline documents, and it may be the case that aid evaluators in general, when stating the overall risks of the project,

Ridge Regression Trained with non-LLM categories

In order to justify the addition of non-LLM categories, we use the baseline statistical categories apparent in prior literature and train a General Linear Model (GLM) on the outputs. Features include

- planned activity duration
- planned total disbursement
- whether the activity is primarily loan or grant-based
- the one-hot encoded reporting organization
- the Country Policy and Institutional Assessment (CPIA) score from the World Bank for that country
- the scope of the activity on a scale from 1-7, ranging from local to global
- the log(GDP/capita) of the countries where the activity takes place weighted by the percentage of the activity performed in each country.
- *gemini-2.5-flash*-generated evaluation on a score from 0 to 100 of:
  - how well financed the activity is
  - the activity integratedness within the broader activity ecosystem
  - the expected implementer performance
  - the ease of targeted outcomes
  - the degree of contextual challenge
  - the overall risk level
  - the activity's overall technical complexity.

The activity start date was not used, as there was no clear linear pattern with regards to overall activity success over time in the training data.

2.5 Experimental methods

Various methods were used to obtain experimental results. I describe them below.

Non-Parametric Bootstrap

The non-parametric bootstrap is a method used to diversify the training data, increasing the diversity in models that are trained many times. It can be used both for ensuring methods robustly improve performance on a diversity of different training setups, and in the case of training the random forest, increases independence between trees. This works by randomly sampling the same number of samples as exist in the training set, with replacement (the same training point may repeat more than once, at random).

GLM using IATI Features and Grades

A GLM is trained with ridge regression to reduce overfitting on noise and improve generalization.

Nearest Neighbor (Vector Similarity)

I first constructed a similarity test using features including countries of the activity, GDP per capita as described previously, the scope of the activity, and the implementing and funding organization ID. I found however that this similarity test significantly underperformed compared to the semantic similarity of the *gemini-2.5-flash*-generated summary of the activity documents. I first weight the similarity proportional to its embedding semantic similarity score, and tested a cutoff for averaging 1, 3, 7, 10, 15, and 20 nearest neighbors using the Gemini embeddings model *gemini-embedding-001*. I found 15 nearest neighbors was the highest-performing using this method, and thus use the weighted average of the nearest neighbor ratings to predict the overall activity score. Although the nearest neighbor method was used to collect examples for the LLM prompt, it was found that simply taking the weighted mean of ratings underperformed the "most common rating" method.

Random Forest

The Random Forest method is a statistical algorithm which constructs an ensemble of decision trees which would produce the correct output on the training data, and averages those decision trees. The averaging nature of the random forest algorithm reduces overfitting on the training data. The algorithm is inherently "regularized", penalizing an overly complex decision tree. The decision trees split based on value ranges of the features. By reducing the depth of the trees (the number of decision points where the decision tree splits), we can reduce the memorization of the training data from the trees, and improve generalization of the model. Each decision also only considers a random fraction of the features, encouraging each tree to be more independent of each other and improving generalization further. The bootstrap method is also used to train trees, encouraging tree

independences.

XGBoost

The XGBoost (Extreme Gradient Boosting) method is a statistical algorithm which constructs an ensemble of decision trees sequentially, where each subsequent tree attempts to correct the errors made by the previous trees. Unlike the Random Forest which trains trees independently and averages their predictions, XGBoost builds trees iteratively, with each new tree focusing on the residual errors of the ensemble.

The algorithm incorporates both L1 and L2 regularization terms to penalize model complexity and prevent overfitting on the training data. The decision trees split based on value ranges of the features, using a splitting criterion that accounts for the gradient and hessian of the loss function. By limiting the depth of the trees (the number of decision points where the decision tree splits) and controlling the learning rate, we can reduce the memorization of the training data and improve generalization of the model. Each decision also considers only a random fraction of the features (column subsampling), and each tree is trained on a random fraction of the training samples (row subsampling), which helps prevent overfitting and improves model robustness.

Language Model Embeddings

Statistical models are unable to explicitly capture textual information. Accordingly, I insert this information via usage of embeddings, using the same Gemini embeddings model *gemini-embedding-001* of LLM-generated "targets" field as a semantic representation of what each activity is trying to accomplish. This follows (**goldembergMindingGapAid2025**) in extracting the World Bank Project Development Objectives (PDO), as I find they are very similar to the LLM-generated outputs. I also attempted PDO extraction using regex methods, but found the results were noisy on matching, especially on non-world-bank projects, and did not improve prediction performance as much as embeddings on the llm-generated targets. I first normalize the LLM-extracted targets text into a stable canonical form (removing formatting artifacts, unescaping, splitting on separators, dropping "NO RESPONSE" tails, and deduplicating near-identical chunks). I then embed the cleaned targets text for each activity with the *gemini-embedding-001* model, yielding a single high-dimensional vector per activity that summarizes activity objectives (targets) in a continuous latent space.

I then replicate (**goldembergMindingGapAid2025**) and compress target embeddings using a two-stage dimensionality reduction pipeline. First, I apply Principle Component Analysis (PCA) to reduce the embedding vectors to 50 dimensions and fit UMAP on the PCA outputs to produce 2D and 3D coordinate maps. I find the 3D embeddings (umap_x, umap_y, umap_z) perform better on prediction than 2D. This preserves enough local topology that activities with similar targets remain near each other in the compressed space. I also find qualitatively, that sectors with similar 2D vectors cluster

around the activity environmental category, as in (**goldembergMindingGapAid2025**). While (**goldembergMindingGapAid2025**) find significant signal in deviations from average embeddings for countries or sectors, I do not find these theorized degree of "contextualization" features aid forecasting skill when I add them to my model.

By counting the word occurrences between features higher or lower on the UMAP x,y, and z axes, it is possible to investigate what aspects are reported by the embeddings, and the common sectors which they were categorized in. Broadly, low x values correspond to forestry, agriculture, water, and managemant related terms while high x correspond to energy sector and financing related terms. Low y correspond to similar terms as high x, both mostly in the energy sector. High y terms related to biodiversity, wastewater, and conservation. The z axis seems to relate more to urban vs rural: Low z corresponds to wildlife and undp related terms as well as "rural", while high z corresponds to sanitation and wastewater, as well as "urban" and "city". The z axis may also correlate slightly with contextualization, where the more "cookie cutter" objectives with regards to the sector in question are typically correlated with low z values. Lastly, the z axis also has a weak negative correlations with the number of countries: low z values tend to occur in fewer countries.

Embeddings for Sector Clustering

A key component of the efficacy of projects is how they allocate their funding. To do so, I query *gemini-flash-2.5* to identify the breakdown of outcome-related funding, where available in the baseline documents. The allocations of funding are required to approximately sum to the total expenditure of the project, with approximately 55% of projects having their budgets identified. If no high-scoring finance or budget pages were identified in the categorization step, I fallback to the first 10 pages of the highest rank baseline document.

In order to make the resulting dataset of budgets usable by the statistical model, I use the embeddings model to cluster the descriptions of the subsectors into 15 sector clusters (having tried 10, 15, and 20 clusters, I find 15 provides the optimal performance on the validation set). I sum the allocations within each cluster and report as a fraction of the total as a feature for the statistical model.

2.5.1 LLM Forecasting Method

Multiple studies have measured zero-shot LLM forecasting capability against the base model performance, and found better general ability base models tend to perform better on forecasting tasks (Halawi et al., 2024) (Karger et al., 2024): In one study with dozens of base models and a dynamically updating benchmark on prediction market forecasting questions, an inverse linear relationship was found between the human preference of a

model's answer (in terms of an ELO score) and the Brier score, and similarly a log-linear inverse relationship between the compute used to train the model and the Brier score (Karger et al., 2024).

In order to guard against leakage of information from the training, I selected deepseek as my forecasting model, due to its strong performance comparable to other models which have similar training cutoff dates (2023 for Deepseek V3.2).

The LLM forecasting method was decided upon by iteratively inspecting both the quality of the response, and the overall accuracy of the forecasts made by the LLM. To generate the LLM forecasting methods, *gemini-2.5-flash* was prompted with a series of "mock forecasts", generated by *gemini-2.5-pro*. The "mock forecast" used relevant pages retrieved by ranking the categorized topics by forecast informativeness and retrieving 10 pages of the most relevant activity data and 10 pages of the most relevant evaluation data, prioritizing pages marked as "deviations from plans", "delays or early completion", or "over or under spending" with a minimum forecasting relevance score of 3/10, and otherwise returning the pages with the highest forecasting relevance score.

To generate each mock forecast, we constructed a retrieval-augmented input consisting of up to 10 baseline pages and up to 10 outcome/evaluation pages per activity. Baseline pages were selected from high-scoring passages in predefined "forecast-informative" categories (e.g., objectives, implementation plans, risks, financing details, contextual challenges, and stakeholder/implementer information), using a high relevance threshold (minimum categorization score of 9) and including nearby pages when insufficient high-scoring pages were available. Outcome pages were selected from outcome documents emphasizing deviations from plans (including deviations, delays/early completion, and over/under-spending), using a lower relevance threshold (minimum score of 3) and likewise including surrounding pages to reach the target count when needed. We then merged these retrieved excerpts with activity metadata (title, scope, planned start/end dates, planned financing totals when present) and brief model-generated baseline summaries (activity description and risk summary) before prompting Gemini to write a forecast from the ex-ante perspective. Importantly, the prompt required the model to end by outputting the *known* final evaluation rating for that activity (derived from the merged ratings file and converted into scale-specific text via `get_ratings_text`), while also instructing it to ground the narrative in the retrieved evaluation pages and to return "NO RESPONSE" if the evaluation excerpts did not contain sufficient justification for the assigned rating.

The most semantically relevant activities which ended approximately at or before the start of the activity being forecasted was then retrieved (see Section 2.5. In addition, the activity "risks" were inserted before each mock forecast, to provide context for the example. Each mock forecast was structured in a way similar to the highest performing scratchpad method from (Halawi et al., 2024).

A series of features including the activity title, start date, and activity location were injected into the prompt to provide context for the activity, as well as a *gemini-2.5-flash*-generated summary.

Finally, the distribution of rating outcomes was inserted into the prompt, in order to prevent collapse towards only a few ratings.

2.5.2 LLM Prompting Strategies

The full prompt template for the LLM Forecast is shown in Figure 6.

Few-Shot Block In both methods, I use a k-nearest-neighbors (KNN) few-shot block of semantically similar activities in the training data (see Section 2.5 for how semantic similarity was determined). I selected a range of nearest neighbors. I asked the language model to extrapolate lessons about rating scales for the most similar "Highly Unsatisfactory", "Unsatisfactory" or "Moderately Unsatisfactory", the most similar "Moderately Satisfactory", and the most similar "Satisfactory" or "Highly Satisfactory" rated examples in the training data. A selection of $k = 3$ summarized mock forecasts was found to perform better $k = 1$, 5, or 7.

Each example activity in the few-shot block included (i) key metadata (title and, where available, location and a brief summary), (ii) a short "risks" summary, (iii) the retrospective mock forecast analysis, and (iv) the final evaluation outcome label.

Additional Prompts Two additional prompts were given, and inserted into the final forecast: (1) reasons the activity may have been evaluated as "Moderately Satisfactory" or worse, (2) reasons the activity may have been evaluated as "Moderately Satisfactory" or better.

The forecasting prompt required a structured response format that explicitly considered both lower- and higher-outcome arguments on the rating scale and ended with a single-line prediction. Concretely, the model was instructed to: (1) provide reasons the overall success might be rated `{midpoint_low_text}` or lower, (2) provide reasons it might be rated `{midpoint_high_text}` or higher, (3) aggregate considerations and select exactly one of the `{num_options}` outcomes, and (4) output the final forecast on the last line beginning with `FORECAST:` followed by only the chosen option.

Finally, I appended a short description of the empirical distribution of rating outcomes in the training data. This was found to reduce mode-collapse toward a narrow subset of ratings.

Ensembling Ensembling is simply averaging many individual forecasts of the same model, prompted in slightly different ways. I found ensembling was a relatively weak method while validating, so I reserve the ensembling method for the final held-out set, which will

```
SYSTEM:
You are an experienced international aid decision maker with a quantitative
mindset.  Respond with a comprehensive, thorough forecast of what the overall
evaluation rating of the activity will be, from the options of {options_text}.

USER:
Forecast what the outcome will be for this activity.

### Lessons from similar activities ###
{knn_summary_text}
### End lessons ###

### Additional specific information about the activity that you summarized ###
{rag_synthesis_additional_info}
### End of additional information you summarized ###

ACTIVITY ID: {activity_id}
ACTIVITY TITLE: {activity_title}
ORIGINAL PLANNED START DATE: {planned_start}
ORIGINAL PLANNED END DATE: {planned_end}
ACTIVITY SCOPE: {activity_scope}
PLANNED TOTAL DISBURSEMENT (USD): {planned_total_disbursement_usd}
ACTIVITY LOCATION(S): {locations}
LOCATION GDP PER CAPITA, USD: {gdp_percap}
PARTICIPATING ORGANIZATIONS: {reporting_orgs}
IMPLEMENTING ORGANIZATION CATEGORY: {either "Government" or "NGO", otherwise
line not inserted}

ACTIVITY DESCRIPTION: {chatgpt_description}
ACTIVITY TARGETS: {targets_summary}
ACTIVITY CONTEXT: {activity_context}
ACTIVITY COMPLEXITY: {complexity_details}
ACTIVITY INTEGRATEDNESS: {how_integrated_description}
FINANCING DETAILS: {finance_summary}
IMPLEMENTER PERFORMANCE CONTEXT: {implementer_performance_text}
ACTIVITY RISKS:
{risks_summary}
ACTIVITY POSSIBILITIES: {possibilities_summary}

{training-set rating distribution text}
Here are a few reasons that you said the answer might be "Moderately
Satisfactory" or worse:
{insert_stage_s1_answer_here}
Here are a few reasons that you said the answer might be "Satisfactory" or
better:
{insert_stage_s2_answer_here}

YOUR TASK:
Aggregate your considerations above.  Think like a superforecaster (e.g. Nate
Silver).  On the very last line of your response, write 'FORECAST: ' followed
by exactly one option from this rating scale with no extra words:
{options_text}

Respond only in English.
```

Figure 6: Single-method multi-stage forecasting prompt. Stages s1 and s2 are run as separate calls, and their outputs are inserted into the final (s3) prompt via {insert_stage_s1_answer_here} and {insert_stage_s2_answer_here}.

come once this thesis has reached completion.

Fine Tuning In past work in a similar domain, fine-tuning significantly improved forecasting performance (Wen et al., 2025). I attempted to fine-tune *gemini-2.5-flash* using Vertex AI. Do do so, I used Direct Preference Optimization (DPO), which requires one example of a good prompt, and one example of a bad prompt. Out of a random sample of 100 activity IDs in the training data, I forecasted the final forecast stage 5 separate times using *deepseek-V3.2*. I used 50 pairs where the model forecasted one rating increment closer than another rating, as the pairs of forecasts.

It was often the case that there were multiple options for the best or worse forecast due to the limited 6-point scale. In order to choose among forecasts that were equally good or equally bad, I also took the embeddings of the forecasts and found the consine similarity to the embeddings of the mock forecast, and to embeddings of the outcome document and averaged these similarity scores. The most similar among equally good ratings were chosen as the good example for the DPO training pair, and the least similar among equally bad ratings as the bad example.

Once the 50 pairs were identified, I ran the fine-tuning using default settings over 20 epochs from Vertex AI (Learning rate multiplier of 1, Adapter size of 4, Beta of 0.1).

2.6 Scoring Metrics

Accuracy The percent of the time the correct rating is forecasted. Non-integers are rounded to integers.

Side Accuracy The percent of correctly predicted "Satisfactory" or above vs "Moderately Satisfactory" or below (above or below 3.5). Approximately 50% of the training dataset sits above and approximately 50% sits below this boundary.

RMSE (Root Mean Square Error) Take the square of the difference between every prediction and the true value, take the mean of all such squared values, then take the square root. Measure of "average" distance. Lower is better. On a scale from 0 to 5, therefore worst possible value is 5, best possible value is zero. This method heavily penalizes predictions that are significantly incorrect.

MAE (Mean Absolute Error) Measure of "average" distance, by taking the mean of the absolute value of the residuals. Lower is better. On a scale from 0 to 5, therefore worst possible value is 5, best possible value is zero. This method does not heavily penalize predictions that are significantly incorrect.

Coefficient of Determination (R^2) R^2: Coefficient of determination. Theoretically equals zero, if we always choose the mean (however using the training set mean results in a lower score on the test set in the baseline measure below). If more than 1 regressors are

included, R^2 is the square of the coefficient of multiple correlation and can be negative. Measures proportion of the variation in the dependent variable that is predictable from the independent variable. Higher is better. This method generally does not penalize outliers significantly.

Adjusted R^2 Adjusted R^2 is a version of R^2 that accounts for the number of regressors in the model. Unlike plain R^2, it penalizes adding predictors that do not meaningfully improve fit, making it more appropriate when comparing models with different numbers of features. It can decrease when irrelevant regressors are included, and it can be negative. Higher is better. While it penalizes extra parameters that may lead to overfitting, adjusted R^2 within a training set does not reflect model skill as accurately as out-of-time R^2.

Pairwise Probability Pairwise probability is evaluated as the proportion of pairs of individual predictions in the validation or test set that were correctly ordered from lower to higher rating. This method is insensitive to global shifts in ratings, which may occur due to events like the COVID 19 pandemic. It also is insensitive to calibration of the spread of possible outcomes. Given the significant noise related to globally relevant challenges, it can represent a more achievable and informative metric than R^2 or MAE for model forecasting skill

2.6.1 Grading Free Form Forecasts

In addition to extracting the LLM forecasted rating, I also use *gemini-2.5-flash-lite* to grade free-form textual format forecasts. In order to do so, I had *gemini-2.5-flash* first summarize pages categorized as highly relevant to outcome evaluations, obtaining 10 pages with a score of at least 3/10 marked as "deviation_from_plans", "delays_or_early_completion", or "over_or_under_spending", or failing that, that were graded as highly relevant to activity evaluation. If no such pages were categorized, the first 10 pages of the highest ranked activity outcome evaluation were used. Next, for all ensembles of LLM forecasts, a grade was given for how similar the forecast was to the activity outcome using *gemini-2.5-flash-lite*. Grading was performed based on two key criteria: accurately identification of likely drivers of activity success or failure, and identification of likely outcomes being forecasted.

In accordance with the typical US grading scheme definitions, I define an "F" grade as 55 and an A+ grade as approximately 97, with other grades defined in even intervals. I provide *gemini-2.5-flash-lite* the following rubric:

Grading scale

- A+/A/A-: Excellent forecast, highly accurate, attention on key drivers, multiple major events forecasted accurately.

- B+/B/B-: Good forecast, mostly accurate. A mix of correct and incorrect, but at least one major outcome was forecasted. At least one key driver identified.
- C+/C/C-: Adequate forecast, partially accurate or reasonable. Focus was adequate. Major outcomes incorrect, but some smaller aspects were correct.
- D+/D/D-: Poor forecast, although perhaps one or two small correct things. Mostly inaccurate. Wrong focus on drivers.
- F: Failed forecast, completely wrong or unsupported.

3 Results

I provide a summary of the data made available from this work, provide an analysis of the success of the various methods at forecasting evaluation ratings, and provide preliminary results on the ability for the model to forecast specific outcomes. In order to provide a robust, generalizable forecast of activity success, two primary strategies were employed. The first was an LLM-implemented judgmental forecasting method. The second method was to use statistical methods. Both strategies were found to have separate strengths and weaknesses:

Strengths

LLM Forecasting

- Can explicitly reason and identify missing information
- General reasoning skills may transfer across domains
- Can produce text-based forecasts of specific events

Statistical Models

- Cheap and fast to iterate
- Can use large numbers of features without context limits
- Can incorporate LLM forecasts and grades as features
- Efficiently generalize over large datasets
- Easier to prevent future leakage
- Mature methodology

Weaknesses

LLM Forecasting

- Expensive and slow to iterative
- Difficult to calibrate
- Fine-tuning is expensive, and unavailable for best models
- Limited interpretability of model reasoning
- Best models are closed-source
- Risk of training-data leakage
- Limited context window constrains attention and calibration

Statistical Models

- Cannot perform explicit reasoning
- Cannot directly use rich textual or world knowledge
- More prone to overfitting dataset-specific quirks

After consideration of several metrics, I decided the Pairwise probability and R^2 were the most appropriate for assessing model skill in the domain of ratings and activity quantitative outcomes. R^2 is sensitive to bias and outliers, which are important for assessing absolute predictive accuracy. However, a common use-case in aid funding decision making is comparing a pair of activities, or even a suite of many activities. In such a use-case, the Pairwise probability is more representative of what is needed. Furthermore, a weakness in R^2 is that global shifts in the ratings and outcomes due to

external factors can be unpredictable, and a model may by-chance capture these shifts without any real improvement in forecasting skill. Both are clearly interpretable: always forecasting the mean value is 0, equivalent to 0% of variance explained for R^2, while 50% is equivalent to the pairwise forecast of random chance.

3.1 Forecasting Ratings with LLMs

One research question of this thesis is how do differing methods of llm forecasting apply in in the context of forecasting international aid success ratings. I find that more generally capable models are consistently better forecasters. I find otherwise little evidence that any other forecasting techniques improve model performance. The strategy of incorporating additional information relevant to the forecasting question as well as references to similar activities does not clearly improve forecasting skill. I find no evidence of data leakage contaminating the ratings. I also find little evidence that fine-tuning can improve forecasting accuracy in this domain, although a thorough assessment of the potential for fine-tuning *gemini-2.5-flash* was cost prohibitive. I also find language models on their own consistently significantly underperformed statistical models in forecasting skill.

Using statistical models as the drivers of LLM forecasts, I find that the assessed similarity between forecasts and summaries of outcome evaluations strongly correlate with improved ability to predict ratings, and that statistical models significantly improve the quality of LLM free-form forecasts.

In addition to extracting the LLM forecasted rating, I also assess LLM grades. I find that the assessed similarity between forecasts and summaries of outcome evaluations strongly correlate with improved ability to predict ratings (see Figure **??**). Furthermore, I find that when the predicted rating using the best statistical model is injected into the prompt, LLM forecasts of activities become much more similar to the summaries of their outcome evaluations. This provides evidence that while language models can accurately forecast the reasons activities would succeed or fail if given the evaluation ratings, they generally lack the ability to correctly weigh the various factors to produce a calibrated forecast.

While ratings are not clearly correlated with better similarity grades, RAG, KNN retrieval, and extra reasoning of the models (stages 1 and 2) consistently improve free-form textual similarity to the outcome, albeit by a small margin. Simply presenting the grading model with the *gemini-2.5-flash* summary of risks of the project from the project documents lead to a forecasting score of 80.2 (between a "C+" and a "B-", corresponding to an "Adequate" forecast, with only "Adequate", rather than a good, focus on the drivers). However, the risks did not typically include any affirmative forecasts of outcomes, which may in part explain the low ratings. The best forecasting method in terms of similarity

to outcomes was the forced ratings with a large amount of context (RAG+KNN) and included stage 1 and 2, achieving a grade of 88.5, which corresponds to a "B+" grade given by *gemini-2.5-flash-lite* (which sits midway between "Good" and "Excellent", meaning the forecasts were on average between "mostly accurate" and "highly accurate", at at least one major outcome was forecasted, and at least one key driver of the outcome was identified on average). Without any additional methods, *deepseek-v-3.2* can typically achieve a "B" grade on average, while use of the statistical model or of *gemini-3-pro* can lift scores to the "B+" range. The improvement in ability to forecast overall ranking improves significantly as well.

Fine Tuning I found the training loss for *gemini-2.5-flash* steadily reduced over the course of the fine-tuning, reducing by 95%. I tested this on an early subset of the validation data which excluded BMZ activities. The performance results were mixed. While the R^2 performance improved modestly, it remained worse than simply guessing the mean rating in the validation set, and remained far below the performance of the more powerful *gemini-3-pro*. Furthermore, the DPO objective is specifically intended to improve pairwise performance between two pairs of forecasts. However, the Pairwise probability score was slightly worsened by fine-tuning. The cost for the fine-tuning of 20 epochs for 50 pairs was close to 140 euros. Therefore, scaling up larger training points for fine-tuning does not seem to be a promising strategy, either to reach the performance of *gemini-3-pro*, or to reach the performance of the more powerful combined statistical models.

RAG and KNN I find small differences in in forecasting skill for the addition of RAG and KNN on forecasting ratings. However, the free-form similarity scores consistently improve on the validation set when RAG and KNN context is included. This indicates that while the LLM is able to identify more important information when reasoning about the final forecast, it is insufficiently calibrated to use the additional information it has gathered to improve the forecast itself.

LLM Model Selection I find no consistent difference between *gemini-2.5-flash* and *deepseek-v3.2* in forecasting skill. However, *gemini-3-pro* appears to be significantly better at forecasting than both, in both sets of activities achieving higher similarity grades, higher R^2, and higher Pairwise probability than other models. The only exception to this is the models which have been instructed to report an overall forecast which matches the forecast from statistical models.

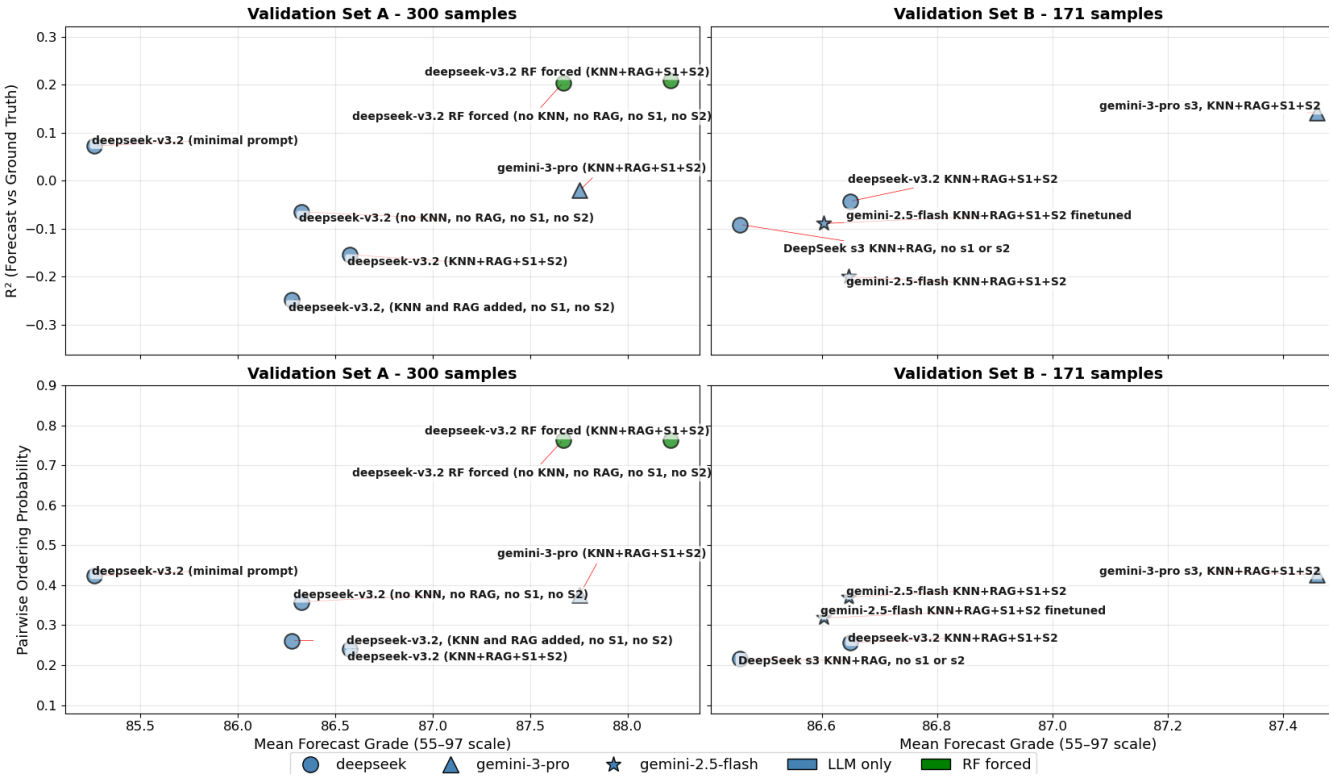

Figure 7: A comparison of various LLM forecasting methods. Notably, *gemini-3-pro* outperforms other models when considering similarity grades. While information injection appears to help the forecasts come to their conclusions for the right reasons, it also appears to harm forecasting skill. When only the Activity ID, title, and a *gemini-2.5-flash* summary of the most important pages for activity forecasting were injected, the LLM performed better on absolute ratings. When forced, somewhat better free-form grades are achievable than without forcing, but only if additional information such as RAG and KNN is given to the model.

3.2 Forecasting Ratings with Statistical Models

Overall, the forecasting system I produce is capable of forecasting evaluation ratings significantly above chance on out-of-time activities. Compared to prior work (**ashtonPuzzleMissingPieces20**
I report a value consistent with an adjusted R^2 with training set (I report an R^2 of 0.34 and an adjusted R^2 of 0.29 for within-training set correlations on primarily world bank ratings, while others report at maximum an adjusted R^2 of 0.3 (**goldembergMindingGapAid2025**). I was not able to identify comparable out-of-time, time-ordered split analysis in the literature. As expected for forecasting under data distribution shift, my results on the out-of-time validation set were somewhat weaker, with an R^2 of 0.23 for the random forest model, and an R^2 of 0.26 when incorporating the language model forecasting results correction + recency model.

3.2.1 Overfitting Corrections

R^2 was chosen as the "Adjusted R^2" has been used in similar work to evaluate model performance in the development aid literature and penalize overfitting by reducing the reported R^2 as a function of the number of input parameters. Mirroring similar reported methods in the literature, I calculated adjusted R^2 on the training points. While this is sensitive to overfitting, it is a common practice in the development aid literature. However, I find adjusted R^2 within the training set is highly sensitive to the specific parameters of the RF model and the subsequent degree of overfitting, such that adjusted R^2 increases to above 0.6 with default random forest parameters, while performance on the validation set drops (See Table 1. I conclude that adjusted R^2 should not be used as a measure of forecast skill.

Relative to the default `RandomForestRegressor` configuration (e.g., `n_estimators`=100, `max_depth`=None, `min_samples_split`=2, `min_samples_leaf`=1, `max_features`=1.0, `bootstrap`=True, `ccp_alpha`=0.0), the specification used here deliberately constrains model capacity in ways that typically reduce overfitting. Trees are explicitly depth-limited (`max_depth`=14 rather than unbounded) and splits are only permitted when nodes contain substantially more data (`min_samples_split`=20 and `min_samples_leaf`=20), which smooths forecasts by limiting fine-grained partitioning of the feature space. In addition, using a smaller feature subset at each split (`max_features`=0.488) increases tree diversity and reduces variance relative to the default that considers all features. The model also uses row subsampling (`max_samples`=0.86), further reducing variance by injecting additional randomness into each tree's training set. Overall, compared to defaults, these choices trade some bias for a meaningful reduction in variance, making the fitted ensemble less susceptible to overfitting.

3.2.2 Embedding Targets

The language model derived features modestly aided forecast accuracy, in aggregate providing an improvement of about 7% additional explanation of the variance of outcomes out of the 26% discoverable by the RF model (See Table 1). The finance, integratedness, implementer_performance, targets, context, risks, and complexity features were directly inserted as grades from the model.

3.2.3 Recency and LLM Adjustment Ridge Regression

I wanted to both correct the random forest model for temporal distribution shift (e.g., changing reporting practices, evaluation standards, portfolio composition, and macro conditions), and incorporate any usable information from the direct LLM forecasts. Even

when the input features are stable, the conditional relationship $p(y \mid x)$ can drift, so a model trained on older activities can become mis-calibrated on newer ones. Furthermore, I found the LLM forecasts were significantly correlated with prediction error in the validation set.

Residual-correction formulation

Let $\hat{y}_i^{\mathrm{RF}}$ be the random forest prediction for activity i, and let $\hat{y}_i^{\mathrm{LF}}$ denote the LLM Forecast. I define the random-forest residual on the i'th activity as:

$$r_i := y_i - \hat{y}_i^{\mathrm{RF}}.$$

I then fit a ridge regression model to predict residuals from a small feature vector consisting of the RF prediction and (optionally) the LLM Forecast:

$$\hat{r}_i := \beta_0 + \beta_1 \hat{y}_i^{\mathrm{RF}} + \beta_2 \hat{y}_i^{\mathrm{LF}},$$

with an ℓ_2 penalty on (β_1, β_2) controlled by `alpha` (ridge strength). The corrected prediction is:

$$\hat{y}_i^{\mathrm{corr}} := \mathrm{clip}_{[0,5]}\left(\hat{y}_i^{\mathrm{RF}} + \lambda \hat{r}_i\right),$$

where λ is a scaling factor (set to 1.0 in my experiments) and clipping enforces the valid rating range between 0 and 5.

This is a simple stacked model: the RF provides the base signal, and ridge regression learns an adjustment to remove systematic residual error that appears in the recent/LLM-covered slice.

I tested two separate methods:

1. Recency correction (RF re-calibration on recent activities). In this variant I remove the LLM forecast entirely fixing $\beta_2 = 0$, but still calculate an offset β_0 and scaling β_1 on the 150 latest training examples.

2. LLM-informed correction (recency + LLM Forecast). In this variant, the ridge model uses both the RF prediction and the LLM Forecast as covariates on the activities where $\hat{y}_i^{\mathrm{LF}}$ is available. This allows the correction to learn a mapping from $(\hat{y}_i^{\mathrm{RF}}, \hat{y}_i^{\mathrm{LF}})$ to the residual r_i, effectively learning when the LLM Forecast contains signal about systematic RF error on the recent slice. The correction is applied only to activities where $\hat{y}_i^{\mathrm{LF}}$ exists; otherwise, forecasts fall back to the uncorrected RF output.

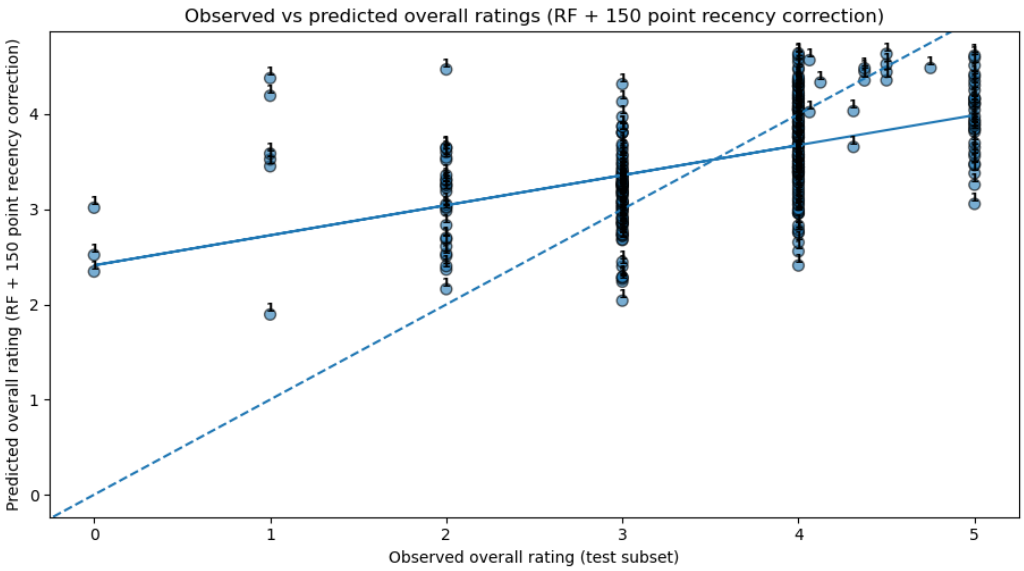

Figure 8: The scatter of observed vs predicted points on the validation set for the LLM-corrected RF prediction. $R^2 = 0.26$, Pairwise probabilty of 77 %.

Table 1: Validation performance in forecasting ratings across forecasting methods. Rows are sorted by ascending R^2 (higher is better). RMSE and MAE are lower-is-better, others are better if higher. Bold indicates the best value in each metric column. Variation between side-accuracy methods ("Moderately Satisfactory" or lower vs "Satisfactory" or lower) were not statistically significant. The "recency correction" variants combine models using the 150 latest-starting activities in the training set for calibration/combination. The "no LLM features" Random Forest excludes the following features: finance, integratedness, implementer_performance, targets, context, risks, complexity, umap3_x, umap3_y, umap3_z.

Method	R^2 ↑	RMSE ↓	MAE ↓	Side Acc. ↑	Acc. ↑	Pairwise ↑
RF + LLM Forecast + recency	**0.258**	**0.815**	0.590	**0.760**	**0.547**	**0.767**
RF + recency	0.254	0.817	0.593	0.750	**0.547**	0.767
XGBoost only	0.233	0.829	0.618	0.713	0.487	0.759
Random Forest (default params)	0.214	0.839	0.622	0.703	0.507	0.757
RF + LLM Forecast + recency (rounded)	0.203	0.845	**0.535**	**0.760**	**0.547**	0.537
Random Forest only	0.203	0.845	0.641	0.710	0.520	0.767
Ridge GLM + Random Forest (mean)	0.198	0.848	0.644	0.697	0.523	0.756
XGBoost, no LLM features	0.189	0.852	0.661	0.700	0.493	0.729
Random Forest, no LLM features	0.186	0.854	0.658	0.710	0.517	0.749
Ridge GLM	0.156	0.869	0.668	0.683	0.507	0.722
Mode of reporting-org score baseline	0.102	0.897	0.579	0.700	0.530	0.426
Ridge Baseline (risks + org only)	0.017	0.938	0.709	0.570	0.433	0.651

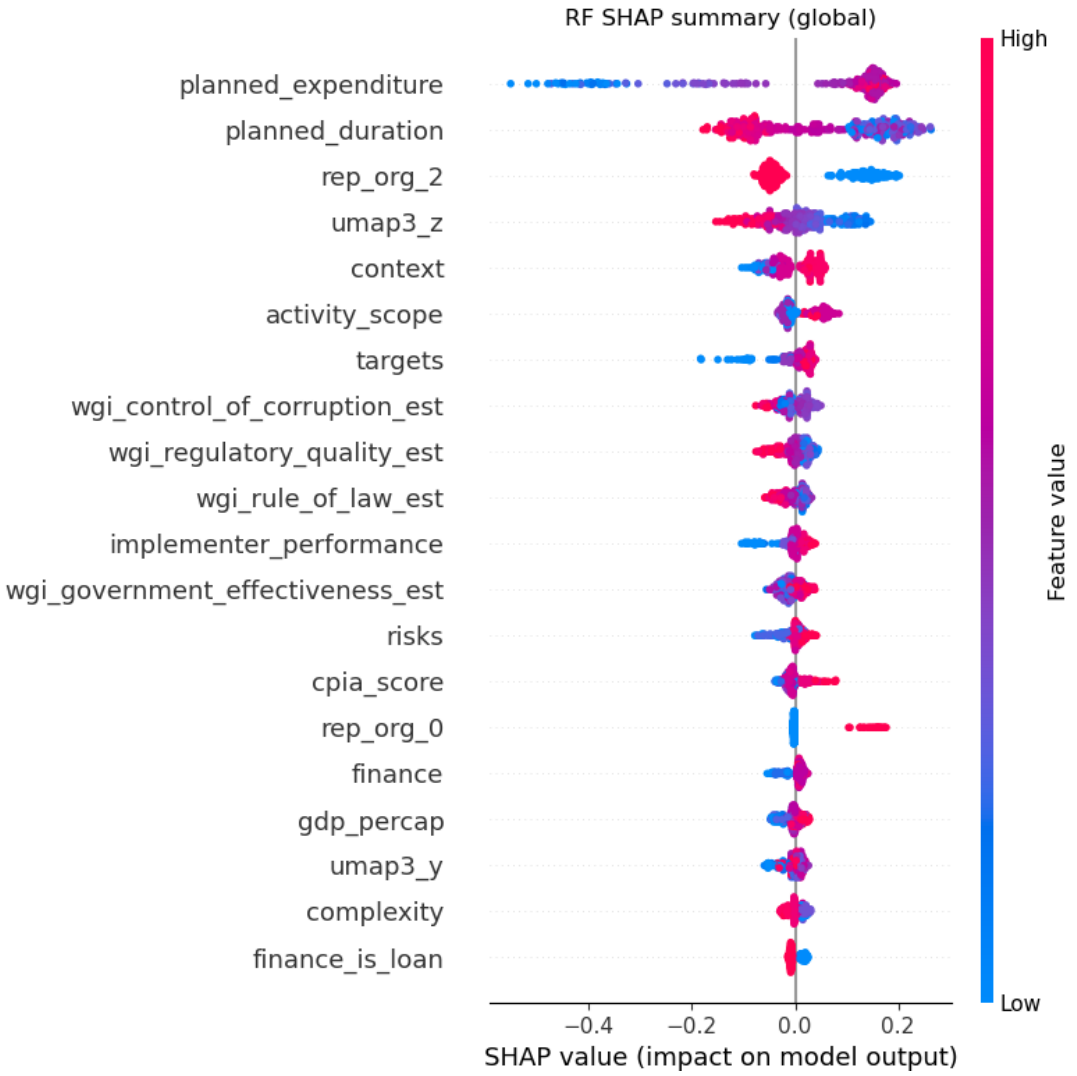

Figure 9: A SHAP analysis of ratings on the validation set from the RF. Red indicates an increase in the value of the feature, while blue indicates a below-average value. Points to the right of zero shift ratings up, points to the left of zero shift ratings down. Overall, planned expenditure, duration, and the organization for rating were the most important features for the random forest model.

3.3 Forecasting Cost-Effectiveness

In general, cost-effectiveness forecasts were weaker than ratings. A similar work found an adjusted $R^2 > 0.7$ for outcome ratings including in forecasting beneficiaries reached (**goldembergMindingGapAid2025**), but this was including actual rather than planned durations, actual rather than planned financial disbursements, and several features including breakdowns of per-sector funding for activities and manager performance ratings from AidData that I did not include in my dataset. My attempt to replicate their result revealed that they likely had training data leakage, but I have not had a response to my inquiries about whether the final code used to produce the tables in their paper was indeed the code containing the bug. Even if they correctly implemented their method, there are some reasons to think my outcomes would be harder to predict. Their paper did not predict on cost-effectiveness, making outcome prediction a much easier task when given overall program spending. Also, the outcomes with high detected correlation measured a lagged 5-year country-level indicator, which is less susceptible to reporting or extraction error, while my data were extracted directly from the outcomes using *gemini-2.5-flash*.

Several factors contributed to the difficulty of forecasting specific outcomes from extracted IATI data:

- Rarity of quantitative outcomes
- Unclear apportioning of funding towards each outcome, which is challenging to extract.
- Inconsistent measurement styles and definitions of terms like Benefit-Cost Ratio, which effects would be included in Economic Rate of Returns.
- Incorrect aggregation of multiple ratings within the documents. I find inaccurate aggregation was initially (artificially) increasing my prediction accuracy, as the errors were more predictable than the outcomes themselves.

Outcome model training and evaluation For each outcome distribution in Table 2, I trained a random-forest regression model containing similar features and hyperparameters to the model used for forecasting ratings, but with the rating target replaced by the relevant activity-level outcome. A one-hot encoded dummy variable for which outcomes being averaged was also included. The ratings were included as a feature to aid learning about activity success. Models were trained using activity IDs in the training split with non-missing outcomes and evaluated on the validation activities. The counts reported in Table 2 correspond to the number of activities available in the validation split for each outcome. Outcome distributions with fewer than 10 activities in either the training or validation split were excluded.

Table 2: Predictive performance of cost-effectiveness outcome RF model on the validation set. R^2, Pairwise ordering probability, and Spearman correlation are shown with 95% bootstrap confidence intervals. In general, only the aggregate had sufficient data to produce a meaningfully predictive model, although the model is only able to distinguish more cost-effective activities 60% of the time. Outcomes are sorted by R^2 (descending). The Benefit-Cost Ratios had identical forecasts for all examples, hence a pairwise and Spearman could not be properly calculated. All other outcomes had fewer than 10 items in training or validation.

Outcome	R^2	Pairwise (%)	Spearman	MAE	$N_{\text{train}}/N_{\text{val}}/N_{\text{test}}$
All selected outcomes (z-aggregate activity mean)	0.05 [-0.03, 0.13]	60 [54, 67]	0.31 [0.13, 0.46]	0.56	682 / 299 / 299
Water Connections Connections (log10(connections))	-0.01 [-0.39, 0.13]	66 [54, 77]	0.43 [0.07, 0.69]	0.70	71 / 25 / 25
Economic Rate Of Return Percent (percent)	-0.01 [-0.19, 0.03]	53 [45, 60]	0.09 [-0.14, 0.28]	15.39	293 / 103 / 103
Co2 Emission Reductions Tonnes Co2E (log10(tonnes co2e))	-0.03 [-0.32, 0.00]	41 [27, 58]	-0.28 [-0.62, 0.18]	1.46	38 / 23 / 23
Benefit Cost Ratios Ratio (ratio)	-0.05 [-0.42, -0.00]	N/A	N/A	0.33	44 / 18 / 18
Financial Rate Of Return Percent (percent)	-0.15 [-1.19, 0.03]	51 [39, 64]	0.01 [-0.32, 0.36]	15.20	147 / 43 / 43
Generation Capacity (log10(Mw))	-0.32 [-1.13, 0.02]	62 [47, 75]	0.39 [-0.06, 0.68]	0.73	44 / 20 / 20

In addition, a single aggregate Z-score was calculated, which subtracts the mean value of each outcome (including ratings) and divides by the standard deviation in the training set. For each activity, the mean value of the z-scores was taken for all dependent variables, including the rating. Due to its high predictability, I theorize that the z-score is a stronger indicator of activity success than activity rating alone, due to the prevalence of "gaming" activity ratings (**goldembergMindingGapAid2025**).

I find outcome prediction to be highly challenging, and my model does not beat simple "predict the mean validation zagg" for outcome prediction (see Figure 2). However, it appears that the model is able to distinguish cost-effectiveness between pairs of progress, at a rate of 60% accuracy, compared to random chance of 50%. The top 3 features of the model are UMAP x axis (negative), the planned duration (positively correlated), and the expenditure.

In keeping with the results of (**goldembergMindingGapAid2025**), I do not find a significant correlation between activity ratings and z-scored outcomes. I found an overall pearson correlation of only 0.07 between cost-effectiveness and ratings over the entire dataset for zagg (N=566).

In conclusion, I find that outcomes cannot be reliably predicted using these methods, revealing the need for more work to extract sufficient data for reliable outcome prediction. Furthermore I find that obtaining cost-effectiveness metrics is more challenging than obtaining ratings, in contrast to the claims from prior literature (**goldembergMindingGapAid2025**).

I do find that failing to divide by the total disbursement as marked in iati increases predictability. This is in part because there is some noise inherent in the total expenditure, and also because when not dividing by expenditure, the model learns to predict linearly higher outcomes correlating with the expenditure. When dividing by each activity's disbursement for those that are marked as dollar-per-unit in Table 2, and looking at all outcomes except ratings, the correlation on z-scores drops to 0.05, (95% CI: -0.03, 0.13) and pairwise probability of 60% (95% CI: 54%, 67%) ($n_val = 299$), compared to an R^2 of 0.10 (95% CI -0.0982, 0.2625) and a pairwise probability of 68% (95% CI: 0.62, 0.74) (n_val=84) when forecasting the z-score for CO_2 emission reductions, generation capacity, and water connections. While pairwise ordering ability for outcomes is statistically significant, the overall ability is weak and appears to be largely a function of expenditure and sector clusters.

Overall, little can be concluded from individual outcome correlations. For more detailed work, expert coding is likely required for robust extraction of outcomes, and funding breakdowns for projects should be used to more accurately evaluate cost-effectiveness, rather than the course assumption that all funding for a project goes to all outcomes. New water or sanitation piping connections is a more clearly comparable outcome, although there may be systematic differences in the costs of sanitation connections and water supply that are not disambiguated by the model.

Despite these limitations, it appears that even a coarse coding of directly comparable activity outcomes is likely to provide a more robust ordering of overall activity cost-effectiveness than evaluator ratings alone, although outcome preditions are not statistically significantly better than forecasting the mean value for the outcome in a given sector in directly forecasting a single activity's cost-effectiveness.

4 Conclusion

I conclude by summarizing my findings regard each research question posed in the introduction.

How do LLM forecasting methods compare to statistical models in forecasting international aid overall success ratings and quantitative outcomes? LLM forecasts themselves consistently underperform statistical models in this domain, while having the disadvantage of being costlier and more difficult to iterate with. However, they

provide meaningfully accurate free-form forecasts when combined with statistical models.

Do forecasting methods using state-of-the-art natural language processing methods meaningfully improve on simpler baseline forecasting heuristics?

Across the board, LLM forecasting methods I investigate do not perform well at forecasting activity ratings when compared to statistical models.

Do forecasting methods using state-of-the-art natural language processing methods meaningfully improve on simpler baseline forecasting heuristics?

The results in this work support the evidence that several methods can improve on baseline heuristics. More sophisticated models such as random forests beat out simpler ridge regression models on every metric. Baseline metrics, such as a linear regression using the organization ID and risks, and a "predict the median rating for this organization" were above random chance, but fell short of more sophisticated methods.

How do differing methods of combining LLM and statistical forecasting compare in this domain? Embeddings of language models inserted into statistical models significantly improve forecasting ability. Furthermore, using embeddings as a means of selecting nearest neighbors and as a component of RAG retrieval gathered information that reliably increased the similarity grades between LLM forecasts and the eventual outcome over all tested LLM forecast configurations. Embeddings were also effective as a method for clustering activity disbursements.

Language models themselves were helpful in extracting grades for various aspects of the forecast, which as a group improved the statistical model forecast. While direct averaging did not demonstrate improved performance, I found that training a simple model to use the residual between the LLM and statistical model modestly improved forecasting skill across most metrics. There is some evidence that training a secondary model to incorporate the final LLM prediction with a random forest prediction can also improve the overall forecast skill.

What methods improve the accuracy of free-form (qualitative) forecasts in this domain? It appears that the only reliable way of improving ratings over a range of different configurations was 1. to switch the model from the somewhat smaller *deepseek-v3.2* or *gemini-2.5-flash* models to the more expensive but more generally capable *gemini-3-pro* model, 2. to incorporate additional retrieval of information via RAG and inserting outcomes from similar past activities using the KNN technique and 3. to directly prompt the model to come to the same conclusion as the random forest model.

Additional prompts to elicit reasons the outcome may go well or badly (stages 1 and 2 in my methods) sometimes deteriorated scores in certain model configurations, although the best performing models used these stages in its context window when forecasting. Similarly, fine-tuning sometimes helped, and sometimes deteriorated scores.

What aspects of the activity available in my dataset at the beginning of the activity lead to higher or lower ratings? As others in the literature have reported, increased duration and planned expenditures tend to correlate with higher activity ratings (Vivalt, 2020) (**ashtonPuzzleMissingPieces2023**) (Eilers et al., 2025). However, there is moderate evidence from the cost-effectiveness random forest model that cost-effectiveness of quantitative outcomes does not clearly correlate with increased spending. Rating organizations tend to differ systematically, and by-sector differences in ratings appear to be more significant in general than regional differences. However, this was not the core focus of this work and needs further research.

How does forecasting quantitative cost-effectiveness activity outcomes compare to forecasting ratings? The current dataset appears insufficient to forecast activity outcomes, both in numbers of quantitative outcomes and noisiness inherent in their extraction. There is some evidence in the validation set that an aggregate measure of cost-effectiveness can be used to rank promising activities, with a 60% chance of a correct ordering compared to 50% which would be arrived at by random chance.

Works Cited

References

Bina, Rachel et al. (Feb. 2025). "On Large Language Models as Data Sources for Policy Deliberation on Climate Change and Sustainability". In: DOI: 10.2139/ssrn.5123359. (Visited on 08/18/2025).

Eilers, Yota et al. (2025). "Volume, Risk, Complexity: What Makes Development Finance Projects Succeed or Fail?" In: *The World Bank Economic Review* (). DOI: 10.1093/wber/lhaf001. (Visited on 09/02/2025).

Halawi, Danny et al. (Nov. 2024). "Approaching Human-Level Forecasting with Language Models". In: *The Thirty-eighth Annual Conference on Neural Information Processing Systems*. (Visited on 08/18/2025).

Hewitt, Luke et al. (n.d.). "Predicting Results of Social Science Experiments Using Large Language Models". In: ().

Karger, Ezra et al. (Oct. 2024). "ForecastBench: A Dynamic Benchmark of AI Forecasting Capabilities". In: *The Thirteenth International Conference on Learning Representations*. (Visited on 08/28/2025).

Koldunov, Nikolay and Thomas Jung (Jan. 2024). "Local Climate Services for All, Courtesy of Large Language Models". In: *Communications Earth & Environment* 5.1, p. 13. ISSN: 2662-4435. DOI: 10.1038/s43247-023-01199-1. (Visited on 08/24/2025).

Vivalt, Eva (Dec. 2020). "How Much Can We Generalize From Impact Evaluations?" In: *Journal of the European Economic Association* 18.6, pp. 3045–3089. ISSN: 1542-4766. DOI: 10.1093/jeea/jvaa019. (Visited on 09/18/2025).

Wen, Jiaxin et al. (June 2025). *Predicting Empirical AI Research Outcomes with Language Models*. DOI: 10.48550/arXiv.2506.00794. arXiv: 2506.00794 [cs]. (Visited on 08/18/2025).

Erklärung zur akademischen Integrität / Declaration of Academic Integrity

Hiermit versichere ich an Eides statt, dass ich die vorliegende Arbeit selbstständig und nur mit den angegebenen Quellen und Hilfsmitteln (z. B. Nachschlagewerke oder Internet) angefertigt habe. Alle Stellen der Arbeit, die ich aus diesen Quellen und Hilfsmitteln dem Wortlaut oder dem Sinne nach entnommen habe, sind kenntlich gemacht und im Literaturverzeichnis aufgeführt. Weiterhin versichere ich, dass weder ich noch andere diese Arbeit weder in der vorliegenden noch in einer mehr oder weniger abgewandelten Form als Leistungsnachweise in einer anderen Veranstaltung bereits verwendet haben oder noch verwenden werden. Die Arbeit wurde noch nicht veröffentlicht oder einer anderen Prüfungsbehörde vorgelegt. */ I hereby certify under penalty of law that I have prepared this thesis independently and only using the cited sources and resources (e.g., reference works or the internet). All passages of the thesis that I have taken from these sources and resources, either verbatim or in spirit, are cited and listed in the bibliography. Furthermore, I certify that neither I nor anyone else has used or will use this thesis, either in its present form or in a more or less modified form, as evidence in another course. This thesis has not yet been published or submitted to another examining authority.*

Potsdam, 4 February 2026