Towards an Intelligent Model-based Approach to Automated Sustainability Reporting

Juan de Lara
juan.delara@uam.es
Univ. Autónoma de Madrid
Madrid, Spain

Esther Guerra
esther.guerra@uam.es
Univ. Autónoma de Madrid
Madrid, Spain

Jacinto Corral
jacinto.corral@sygris.com
Sygris
Madrid, Spain

Cristóbal Mora
cristobal.mora@sygris.com
Sygris
Madrid, Spain

Abstract

Environmental, social and governance (ESG) reporting – or simply *sustainability reporting* – is increasingly important for organisations, yet identifying relevant reporting areas and aligning them with evolving standards is a complex, manual task.

To alleviate this situation, we present a model-based intelligent approach for automating the recommendation of sustainability reporting areas based on company profiles. Our solution leverages publicly available ESG reports to build an ESG-specific recommender through a configurable pipeline. This combines information modelling, LLM-based extraction guided by domain models, model-driven data cleaning and consolidation using a lightweight domain-specific language, and machine learning-based classifier training. By separating domain knowledge from processing logic, the approach can be adapted to different reporting standards and compliance frameworks.

We report on an experiment that shows the value of explicit modelling within the approach, but highlights the need for human supervision in some steps. More generally, we discuss the role of models as key artefacts for integrating ESG domain knowledge, guiding AI-based extraction, improving ESG classifier consistency, and enabling reusable ESG reporting automation solutions.

CCS Concepts

• **Computing methodologies** → **Knowledge representation and reasoning**; • **Information systems** → *Information integration*; • **Software and its engineering** → *Domain specific languages.*

Keywords

Model-driven Engineering, Sustainability Reporting, LLMs, Machine Learning, Decision Trees

ACM Reference Format:
Juan de Lara, Esther Guerra, Jacinto Corral, and Cristóbal Mora. 2026. Towards an Intelligent Model-based Approach to Automated Sustainability Reporting. In *ACM/IEEE 29th International Conference on Model Driven Engineering Languages and Systems (MODELS Companion 2026), October 04–09, 2026, Málaga, Spain.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3837062.3839342

This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License.
MODELS Companion 2026, Málaga, Spain
© 2026 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-2903-4/2026/10
https://doi.org/10.1145/3837062.3839342

1 Introduction

European Union (EU) law, specifically the Corporate Sustainability Reporting Directive (CSRD) [8], requires large companies to disclose information on the risks and opportunities they identify in relation to environmental and social factors, as well as on the impacts of their activities on people and the environment. This is formally referred to as *sustainability reporting*, which is commonly discussed under the broader umbrella of environmental, social and governance (ESG) reporting. These disclosure actions support investors, civil society organisations, consumers, and other stakeholders in assessing corporate sustainability performance within the context of the European Green Deal [7].

ESG reporting is also extending beyond the largest companies. Under the CSRD, listed (i.e., traded) small and medium-sized enterprises (SMEs) are also brought into scope, reflecting a broader regulatory trend towards wider sustainability disclosure coverage. In addition, even companies not directly subject to mandatory reporting may be indirectly affected through value-chain requirements, as large reporting entities are required to collect ESG-related data from suppliers and business partners.

Beyond the EU, ESG reporting requirements also exist in other jurisdictions in varying forms, including the United Kingdom [22], the United States [24], and Australia [2]. At the international level, sustainability disclosure frameworks are increasingly structured around two major reference models: the investor-oriented standards developed by the International Sustainability Standards Board (ISSB) [14], and the impact-oriented Global Reporting Initiative (GRI) Standards [12], which together represent complementary approaches to ESG reporting.

ESG reporting is a complex, time-expensive and effort-consuming process for companies. It requires understanding and deciding what needs to be reported (e.g., which themes under the European Sustainability Standard, ESRS [9]), establishing ESG governance, collecting data, calculating metrics, reviewing and validating the collected information, and submitting and publishing a report. Given the difficulty of these tasks, specific software to help in this process becomes necessary.

Sygris[1] is a company specialised in sustainability software, including sustainability reporting, and in the integral management of sustainability strategies, including data collection, analysis, management and visualisation. The software is offered as a web-based lowcode platform[2]. To increase the value of this platform for customers, it will be extended with a new model-based, intelligent solution that automatically recommends reporting areas from the companies' profiles. This solution, which we report on the paper,

[1] https://sygris.com/
[2] https://sygris.com/en/low-code-platform/

profits from the increasing number of companies making ESG reports public[3] to extract relevant data and train an ESG-specific recommender. This work is the result of an ongoing project between Sygris and the modelling & software engineering research group (https://miso.es) at the Universidad Autónoma in Madrid.

The intelligent recommender implements a model-based approach, which makes the construction and training pipeline configurable not only for ESRS reporting, but also for other standards. The pipeline has an information modelling phase; an extraction phase in which a large language model (LLM) uses the information model to extract data from existing reports; a model-based cleaning, validation and consolidation phase guided by a lightweight YAML-based domain-specific language (DSL); and a training phase that produces a trained classifier out of the cleaned data. This paper reports on the stages of the pipeline and on the deployment strategy. We report on an evaluation over a dataset of more than 730 reports (>10Gb), showing the benefits of our model-based approach and the usefulness of explicit validation rules to improve the consistency of the classifier predictions.

Paper organisation. Section 2 overviews ESG sustainability reporting. Section 3 describes our model-based approach for building an ESG report assistant. Section 4 reports the evaluations conducted. Section 5 compares with related work, and Section 6 concludes.

2 Background on Sustainability Reporting

Under the ESRS framework [11], sustainability reporting follows a structured and regulated disclosure system introduced under EU's CSRD. Throughout this paper, ESRS refers to the standards as adopted in 2023 [9], which is the version targeted by our dataset and extraction models. Its purpose is to standardise how companies report sustainability impacts, risks, and opportunities, ensuring that ESG information is comparable, auditable, and decision-useful across organisations. ESRS defines not only what must be reported, but also how this information is structured and integrated into corporate reporting processes.

ESRS is organised into three main categories of topical standards: Environmental, Social, and Governance (cf. Table 1). The Environmental standards address the company's impact on natural systems and resources. These include ESRS E1 on climate change, E2 on pollution, E3 on water and marine resources, E4 on biodiversity and ecosystems, and E5 on resource use and circular economy. Together, these standards cover emissions, environmental degradation, resource consumption, and ecological impacts, forming the environmental pillar of sustainability disclosure.

The Social standards focus on impacts related to people and society across the company's operations and value chain. ESRS S1 covers the company's own workforce, S2 addresses workers in the value chain, S3 focuses on affected communities, and S4 covers consumers and end-users. These standards require companies to disclose information on labour conditions, human rights, equality, health and safety, and broader societal impacts, ensuring that social externalities are systematically reported.

Finally, the Governance standard ESRS G1 addresses business conduct. It requires disclosure of practices related to corporate ethics, anti-corruption, internal controls, and responsible behaviour.

[3] https://insights.efrag.org/reports

Table 1: ESRS Topical Standards and Topics

Category	ESRS Standard	Topic
Environmental	ESRS E1	Climate change
	ESRS E2	Pollution
	ESRS E3	Water and marine resources
	ESRS E4	Biodiversity and ecosystems
	ESRS E5	Resource use and circular economy
Social	ESRS S1	Own workforce
	ESRS S2	Workers in the value chain
	ESRS S3	Affected communities
	ESRS S4	Consumers and end-users
Governance	ESRS G1	Business conduct

In addition, each ESRS in Table 1 is decomposed hierarchically (e.g., ESRS E1 "Climate change" is subdivided in "adaptation to climate change", "mitigation of climate change", and "energy"). Overall, there are 91 leaf subtopics in the 10 ESRS standards.

In practice, ESRS reporting is embedded in the statutory annual management report and is subject to external assurance to ensure data reliability and compliance. Companies must both publish the report publicly and generate it in a structured digital format (e.g., XHTML with ESRS tagging) suitable for regulatory submission and automated processing. This structured digital format relies on the eXtensible Business Reporting Language (XBRL), which encodes ESRS taxonomy concepts as machine-readable tags for automated processing and regulatory exchange [25]. However, only the PDF version of the report is typically made publicly available through corporate channels, while the machine-readable XHTML filing is submitted to regulatory bodies and is not yet broadly accessible. Reports are filed via national systems and, in the future, via the upcoming European Single Access Point (ESAP) [10], which aims to make both human-readable and machine-readable formats available to regulators, investors, and data users.

3 Intelligent Model-based Approach to Sustainability Reporting

One goal of the project was the construction of an intelligent recommender that predicts suitable ESRS topics to report given a company profile. This recommender shall be integrated in a novel lowcode intelligent reporting platform by Sygris, which would guide the reporting process until a final XBRL-compliant report is generated.

Figure 1 shows the pipeline to construct the recommender. As input to the pipeline, we first collected available ESG reports from various sources, including representative companies' web sites and the public database at https://insights.efrag.org/reports. This resulted in a dataset of over 700 ESG documents.

This dataset is the input to the pipeline, which starts with an *extraction* phase (label 1 in Figure 1). This phase extracts high-quality data from each document in the dataset regarding the company profile and the ESRS topics reported. The extraction is automated with an LLM. In addition, to make the process flexible and configurable to other reporting standards, the company features of interest and the ESRS themes are modelled using a YAML-based DSL.

Next, a *pre-processing* phase is applied to the extracted data (label 2). Specifically, the ESRS data are encoded as a 102-dimensional binary vector, where 1 indicates "relevant for reporting" and 0 indicates "not relevant". Similarly, a feature vector with dimension 10

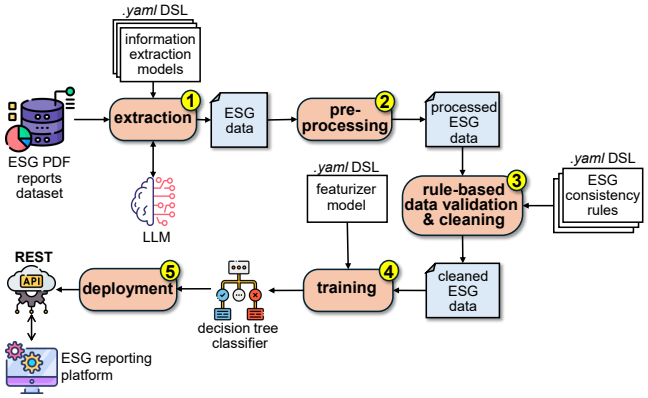

Figure 1: The model-based machine learning pipeline for building an ESG classifier.

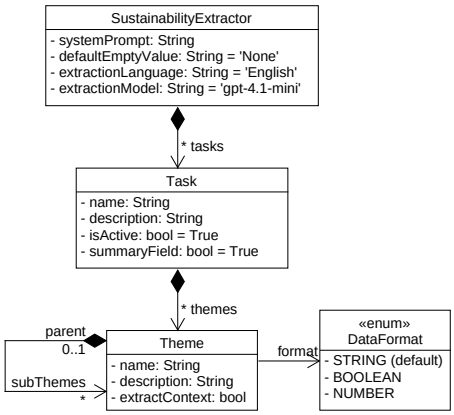

Figure 2: Meta-model for ESG information extraction.

is constructed according to the company profile features modelled in the previous phase.

The next phase performs *validation and cleaning* of the pre-processed data (label 3). This is necessary because the initial LLM-based extraction process may introduce errors: the documents in the dataset are large (the average page length is around 300); include complex text, images and diagrams; and may yield feature vectors with more than 100 features. Consequently, this third phase validates the extracted, pre-processed data against domain knowledge and priors, modelled as rules in a YAML-based DSL. These rules check expected conditions and can correct erroneous cases.

The cleaned and validated ESG data is then used for *training* a classifier based on decision trees (label 4). We use another YAML-based DSL to model how the data columns are to be featurized (i.e., represented and transformed). This design is energy-efficient: the LLM is used only offline during the extraction phase, while the deployed classifier requires significantly fewer computational resources at inference time than an LLM-based alternative would.

The last phase of the pipeline is the *deployment* of the trained classifier within the reporting lowcode platform (label 5). This classifier is not meant to be static, but can undergo periodic retraining once integrated within the platform. Companies can use the platform to obtain suggestions on which ESRS topics to report based on the company features, while enabling the possibility to adjust these suggestions if needed. The platform will use such adjustments to retrain the classifier and improve future recommendations.

Next, we describe the pipeline steps: the ESG information extraction models (Section 3.1), the information extraction agent (Section 3.2), the pre-processing (Section 3.3), the data cleaning and validation rules (Section 3.4), the model-based ESG classifier training (Section 3.5), and the deployment (Section 3.6).

3.1 ESG information extraction modelling

We model the information to be extracted from ESG reports using a YAML-based DSL that allows defining hierarchical information items and their relevant fields. Figure 2 shows its meta-model.

The root class SustainabilityExtractor allows configuring the system prompt, the default empty value, the extraction language, the language model to be used, and the information extraction tasks. Each Task has a name, a description, an isActive boolean flag indicating whether the LLM-based extractor should execute the task (since multiple tasks may be used to process the same document, e.g., one per ESRS topic and another for the company profile), and a summaryField boolean flag that, when selected, will add a boolean summary field to each theme within the task indicating whether the theme was found relevant for the document.

Tasks contain Themes, which may in turn contain arbitrarily nested subThemes. Each Theme has a name, a description, an output data format (string by default), and an extractContext boolean flag that, when selected, generates an LLM-computed field containing a brief verbatim paragraph indicating where the theme was extracted from in the document. The latter flag facilitates early validation of the extraction results by human experts against the original document, while for bulk extraction, the flag is normally disabled.

Given its hierarchical nature, and for a better integration with Python-based machine learning pipelines, this meta-model is provided with a YAML syntax. Each Task is modelled in a different file, and the configuration parameters in the root class are passed in the command line to the extractor.

As an example, Listing 1 shows an excerpt of the task model to extract a company profile from an ESG report. Keyword task in line 1 signals the beginning of the task definition, and keyword fields in line 6 signals the beginning of the definition of its themes. The excerpt declares three relevant features (i.e., objects of type Theme according to the meta-model) to be extracted: the company name (lines 7–10), the sector (lines 11–15), and the reporting currency (lines 17–21). The complete model declares another seven features: products or services that the company offers, country where the company's headquarters are located, world areas where the company has presence, company size, juridic form, annual turnover, and whether it is stock listed.

As a second example, Listing 2 displays the extraction model for ESRS E1 (Climate Change), which comprises three subtopics: Climate Change Adaptation (lines 5–11), Climate Change Mitigation (lines 12–20), and Energy Use (lines 21–27). The flag summaryField is set to True (the default value), and all topics have string data format (the default value). Note that the DSL supports keyword

```yaml
task:
  active: False
  name: company_data
  description: Company data
  summaryField: False
  fields:
    - name: company name
      description: The full legal name of the company
      extractContext: False
      format: str
    - name: sector
      description: |
        Industry sector(s) the company operates in. Choose EXACTLY ONE of
            the categories below. Return the letter of the category:
        (A) Agriculture, Forestry, Livestock and Fishing
        (B) Mining and Quarrying
...
    - name: reporting_currency
      description: |
        Currency used in financial figures throughout the report. Use ISO 4217
            Codes, like USD, EUR, GBP, etc. If the information is not
            available return None.
      extractContext: False
      format: str
```

Listing 1: Excerpt of extraction task for the company profile.

synonyms for fields and themes (line 4), to make it natural for both company profile and ESG data extraction.

```yaml
task:
  name: ESRS E1
  description: Climate Change
  themes:
    - name: Climate Change Adaptation
      description: |
        Include data such as:
        - Physical risk assessments (chronic and acute risks)
        - Adaptation strategies and actions
        - Climate resilience of assets, operations, and value chain
        - Investments in adaptation measures
    - name: Climate Change Mitigation
      description: |
        Include:
        - GHG reduction targets
        - Policies and actions for GHG reduction
        - Decarbonization initiatives and levers
        - Use of renewable energy and energy efficiency measures
        - Scope 1, 2, and 3 GHG emissions (with breakdowns and
            methodologies)
        - Use of internal carbon pricing (if any)
    - name: Energy Use
      description: |
        Include:
        - Total energy consumption (direct and indirect)
        - Share of renewable vs. non-renewable energy
        - Energy efficiency initiatives
        - Energy intensity ratios (e.g., per product, per revenue)
```

Listing 2: Excerpt of extraction task for ESRS E1.

Although our pipeline is instantiated for ESRS reporting, this model-based extraction approach is designed to support alternative sustainability reporting frameworks. For example, replacing the ESRS information extraction model with a GRI information extraction model would allow the pipeline to target GRI topical standards instead of ESRS topical standards. The same flexibility applies to the evolution of a single framework. On 3 July 2026, the EU adopted a delegated act revising the ESRS, reducing total datapoints by over 70% while retaining the same ESG topic coverage. Our extraction is defined over topics rather than datapoints, so this revision leaves the extraction models essentially intact. Still, this precisely shows the benefit of declaring what to extract in a model rather than in code: regulatory change becomes a model edit.

3.2 Model-based ESG information extraction

We have built an LLM-based extraction agent that takes an ESG information extraction model and a document dataset, and produces a *.csv* file with the extracted data. Figure 3 shows its scheme.

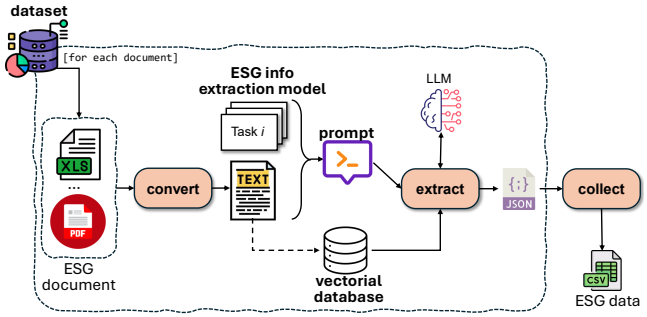

Figure 3: The model-based ESG extraction agent.

The agent traverses each document in the dataset. While our current dataset only contains PDF files, the architecture supports ESG reporting data in multiple formats like excel or PDF. Then, for each document in the dataset, and for each task in the ESG information extraction model, the extractor builds a prompt comprising: (i) system-level extraction instructions, (ii) instructions derived from the task, and (iii) the textual content of the ESG report. In addition, the agent uses *structured output* [18] to instruct the LLM on the expected output format (in JSON), which should comply with the extraction task. This means that the instructions for the structured output are automatically derived from the information extraction model. If several extraction tasks are required for the same document, these are iterated in multiple prompts to avoid overwhelming the LLM with large prompts, instructions and output format requirements.

Since ESG reports tend to be large, the extraction process supports two modes. If the text from the report fits in the LLM context window, then direct prompting is used. Otherwise, the text is chunked and stored in a vector database. Then, the chunks most relevant to the given extraction task are retrieved using a retrieval-augmented architecture (RAG) [17] and included in the prompt.

Once all the documents in the dataset are processed, the extracted data is output in *.csv* format.

3.3 ESG data pre-processing

The extracted data needs to be processed to make it suitable for training a decision tree classifier. Specifically, since we need a binary vector summarising the relevance of each ESRS topic, the content of each cell in the *.csv* is converted into 1 (relevant) or 0 (not relevant).

The extraction tasks typically produce 'None' for topics not found relevant in the report, and a textual summary of the relevance otherwise (for topics with string format). For example, the following is an excerpt of the extracted result for the Climate Change Adaptation topic within ESRS E1 for company XXX[4]: "*Company*

[4]anonymised

XXX has a Climate Transition Plan consistent with its Strategic Plan, aiming to reduce Scope 1 direct emissions by at least 95% by 2050 compared to 2017 and Scope 3 indirect emissions by 25% by 2040 compared to 2019....". These summaries are not discarded, but we plan to use them in the final deployed recommender to provide examples and rationales explaining why a topic is considered relevant.

For extraction tasks that selected summaryField (cf. Section 3.1), the corresponding generated boolean values are also binarised.

3.4 Data cleaning: validation and fixing rules

The extraction process explained in Section 3.2 is prone to errors, as it is made by LLMs by processing extensive documents. To mitigate this issue, we perform a validation and amendment stage in which rules defined in a YAML-based DSL check the extracted data (in the *.csv* file), signal suspicious values, and can trigger corrections (if so desired). Figure 4 shows the meta-model of the DSL.

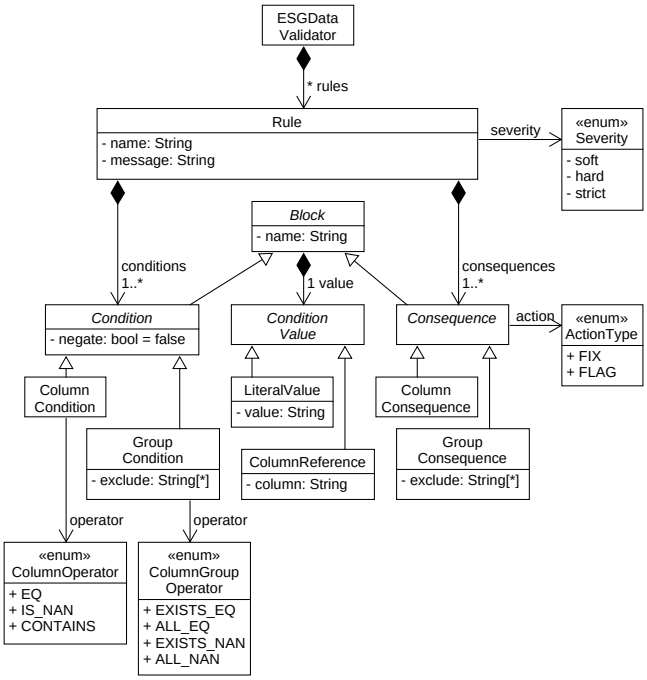

Figure 4: Meta-model of the validation and fixing rules.

Rules have the form *when* ⟨condition⟩ *then* ⟨consequence⟩, expressing that if a pattern is found in some columns of the *.csv* file, then some values are expected in other columns. The conditional of the rule (class Condition) can check values of columns (ColumnCondition), or groups of them (GroupCondition). In the latter case, all columns whose name contain a given literal are sought, and some columns may be listed for exclusion (GroupCondition.exclude). Conditions can be negated via the Condition.negate flag. Expected values in columns or groups are checked via operators (ColumnOperator or ColumnGroupOperator) against literals (LiteralValue) or other column values (ColumnReference). The Consequence of a rule can be used to specify expected column values when the condition pattern is found, and disconformities can be just flagged or fixed (ActionType). Finally, rules declare a severity level (Severity) which indicates the importance of the error.

Next, we illustrate typical usages of the rules. The first one is to check the consistency between a summary field and the columns it summarises. For example, if the summary column is 0, then all columns in its child column group must be 0; whereas if any column in the child column group is 1, so must be the summary column. The two rules in Listing 3 capture this. The first one (lines 2–14) states that when any column starting by esrs_e1_ (excluding the summary column) has a 1, then the summary column should have a 1 as well (otherwise, this requires fixing). Conceptually, this validation is bottom-up, from detail columns to a summary column. The second rule (lines 15–27) is top-down: if the summary column is 0, then all detail columns should be 0. Overall, these two rules ensure the hierarchical coherence of ESRS E1 and its subtopics.

```
1  rules:
2  - name: E1_BOTTOM_UP
3    severity: hard
4    when:
5    - group: esrs_e1_
6      exclude:
7      - esrs_e1_summary
8      op: exists_eq
9      value: 1
10   then:
11   - action: fix
12     column: esrs_e1_summary
13     value: 1
14   message: An E1 subtopic is reported but E1 summary is not
15 - name: E1_TOP_DOWN
16   severity: hard
17   when:
18   - column: esrs_e1_summary
19     op: eq
20     value: 0
21   then:
22   - action: fix
23     group: esrs_e1_
24     exclude:
25     - esrs_e1_summary
26     value: 0
27   message: E1 summary is 0 but an E1 subtopic is reported
```

Listing 3: Hierarchical coherence rules for ESRS E1.

Similar rules may check incoherent empty values, e.g., in summary columns. Listing 4 shows an example for ESRS E1. The rule checks that if all subtopics of ESRS E1 are 0, but the summary column has no value, then the value 0 should be inferred for the summary column.

```
1  rules:
2  - name: E1_NAN_SUMMARY_ALL_CHILDREN_ZERO
3    severity: hard
4    when:
5    - group: esrs_e1_
6      exclude:
7      - esrs_e1_summary
8      op: all_eq
9      value: 0
10   - column: esrs_e1_summary
11     op: is_nan
12   then:
13   - action: fix
14     column: esrs_e1_summary
15     value: 0
16   message: E1 summary is empty but all subtopics are 0: summary inferred as 0
```

Listing 4: Rule cleaning empty value for ESRS E1.

Rules may also be used to assess expected reporting areas given that some other area is reported. Listing 5 shows one example for topics of ESRS E2 and E3. It checks that if water pollution is reported (column esrs_e2_water_pollution), then water discharge should be reported as well (column esrs_e3_water_discharge). Note that while this is normally the case (e.g., the company discharges wastewater containing pollutants), in exceptional cases it may be not (e.g., the company had a pollution incident affecting water but it normally does not discharge water).

```yaml
rules:
  - name: E2_WATER_POLLUTION_IMPLIES_E3_DISCHARGE
    severity: strict
    when:
      - column: esrs_e2_water_pollution
        op: eq
        value: 1
    then:
      - action: fix
        column: esrs_e3_water_discharge
        value: 1
    message: Disclosing water pollution implies water discharge must be
        reported
```

Listing 5: Water pollution implies water discharge reporting.

Rules can also document prior knowledge about the company profile and expected reporting areas. For instance, the rule in Listing 6 captures that, for a company in the energy sector (identified by the value W in the sector column), it is unusual not to report ESRS E1 (Climate Change).

```yaml
rules:
  - name: ENERGY_SECTOR_E1
    severity: soft
    when:
      - column: sector
        op: contains
        value: W
    then:
      - action: fix
        column: esrs_e1_summary
        value: 1
    message: Energy sector company not reporting E1 Climate Change is
        unusual
```

Listing 6: Climate change reporting is relevant for companies in the energy sector.

Other rules may just flag suspicious column value combinations. Listing 7 shows an example signalling that reporting supply chain workforce, but not the own workforce, is unusual.

Even if rules have an expected action (fix, flag), our evaluator can be executed in checkonly mode to flag inconsistencies, or in enforce mode to modify values of discording cells (i.e., changing from 0 to 1 or vice versa) so that the rule is satisfied for the given row. In the latter case, the severity of rules can be used to determine which rules are executed.

Currently, rules check data within the same company, extracted from an ESG report (i.e., same row in the *.csv*). In future work, we will extend the rules to consider several companies, similar to metamorphic relations [21]. These rules may be used to check that if two companies have similar profiles (e.g., same sectors, same geographic operation areas), then they should report the same or similar ESRS topics.

```yaml
rules:
  - name: S2_WITHOUT_S1
    severity: soft
    when:
      - column: esrs_s2_summary
        op: eq
        value: 1
      - column: esrs_s1_summary
        op: eq
        value: 0
    then:
      - action: flag
        column: esrs_s1_summary
        value: 1
    message: Supply chain workforce reported w/o own workforce is unusual
```

Listing 7: Reporting supply chain workforce without reporting own workforce is unusual.

3.5 Model-based ESG classifier training

Since ESG data extraction is driven by a model, the representation and treatment of the features used for training the classifier cannot be hard-coded. Instead, we use a YAML-based DSL to express how the extracted information is to be featurized. Figure 5 shows the meta-model of this DSL.

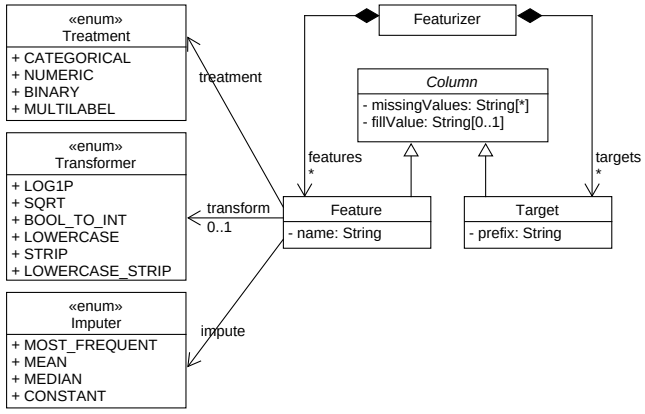

Figure 5: Meta-model of the featurizers.

A Featurizer defines a collection of Features and Targets, the latter being the features that the classifier will predict. Both can declare how they represent missing values, and optionally provide a fill value to be used in that case. Features have also a name, declare how they should be treated (as categorical, numeric, binary, or multi-label values), whether they need transformation, and may choose an imputer for missing values. Transformations depend on the feature treatment: for categorical features, allowed transformations are lowercase, strip, and lowercase+strip; for numeric, log1p and sqrt; and for binary, bool_to_int (which converts True into 1 and False into 0). Imputers provide a default value to missing values using a strategy: the most frequent value, the mean, the median, or the constant specified in attribute fillValue. Targets are identified by a prefix, thus enabling the selection of multiple columns at once.

Listing 8 shows an excerpt of a featurizer model. Lines 3–7 specify how company_data_sector is to be featurized: treated as a categorical variable, converted to lower case, stripped of whitespaces, and initialising missing values (i.e., cells that are empty or contain “NA”,

"None" or "N/A") with the most frequent value. Lines 9–12 specify the treatment for targets. These are all columns whose name starts by "esrs_", and missing values will be filled with 0s.

```
1  training:
2    features:
3      – name: company_data_sector
4        treatment: categorical
5        transform: lowercase+strip
6        impute: most_frequent
7        missing_values: ["NA", "None", "N/A"]
8  ...
9    targets:
10     – prefix: esrs_
11       missing_values: ["NA", "None", "N/A"]
12       fill_value: 0
```

Listing 8: Excerpt of a featurizer model.

The training step is therefore configured through a featurizer model. This approach avoids any code changes when the feature schema evolves. Upon loading, the featurizer is first validated to ensure that all declared treatments, transforms, and imputation strategies are consistent before any data is touched. Features are divided into four treatment types: categorical fields are one-hot encoded after optional normalisation transforms (lowercasing, whitespace stripping); numeric fields are optionally compressed via log or square root transforms before standard scaling; binary fields are mapped to integer flags and passed through unchanged; and multi-label fields, such as subsidiary regions or product categories, are expanded into binary indicator columns.

The classifier is trained as a multi-label model, predicting simultaneously which ESRS topics are relevant for a given company. Our training program can be configured with the classifier type, supporting three tree-based ensemble methods: Random Forest [4], LightGBM [15], and XGBoost [5], all of which are well-suited to tabular data with mixed feature types. To handle the multi-label nature of the problem, the base classifier is wrapped in either a classifier chain that models label dependencies by chaining predictions sequentially, or an independent multi-output wrapper that treats each label independently. Data splitting defaults to iterative stratified splitting, which preserves label distribution across train and test sets in multi-label settings. Training can also be parameterised w.r.t. hyperparameter optimisation, which can be performed via Optuna [1], scikit-learn [20] grid search, or randomised search [3], tuning the base classifier parameters against macro-averaged F1 score using cross-validation.

3.6 ESG classifier deployment

For its integration with the ESG automated reporting tool, the trained classifier is deployed as a REST service. The classifier is exposed through a RESTful API built with FastAPI[5], providing a /predict endpoint that accepts company features and returns a binary prediction for each ESRS topic. At inference time, the service reconstructs the same feature representation used during training.

The system also supports continual learning through a /retrain endpoint. This accepts a set of company features and corresponding ESRS labels, and triggers an asynchronous retraining job. Job lifecycle management is handled via a thread-backed status store, with clients able to poll a /retrain-status endpoint to monitor progress.

[5]https://fastapi.tiangolo.com/

4 Evaluation

In this section, we evaluate our classifier and its construction pipeline to assess the value of the various modelling layers. We aim to answer the following Research Questions (RQs):

RQ1 To what extent do ESG information extraction models produce consistent and reliable company profiles and ESRS labels when applied by LLMs?

RQ2 Does data cleaning using the modelled validation rules improve the logical coherence of the extracted data?

RQ3 Does training on cleaned data improve the logical coherence of classifier predictions?

To assess RQ1, we use two LLMs (*gpt-4.1-mini* and *gemini-2.0-flash*) to extract data from ESG reports using the same information extraction model, and measure inter-LLM agreement. To assess RQ2, we check if the LLM-extracted data violate the validation and fixing rules. For RQ3, we compare the classifier trained with unprocessed data, and the classifier trained with data cleaned by the validation and fixing rules. Since we do not have ground truth data, but automatically extracted data, we do not compare prediction accuracy, but the internal coherence of predictions by applying the validation rules.

Next, Section 4.1 describes the dataset, Sections 4.2–4.4 answer each RQ, and Section 4.5 presents validity threats of the experiment.

4.1 Dataset

The dataset comprises 734 sustainability PDF reports from European companies, collected mainly from https://insights.efrag.org/reports. Of these, 732 were successfully processed, while 2 were discarded due to read errors.

As Table 2 shows, the reports vary in length and file size, reflecting the heterogeneity of sustainability disclosure practices across companies and sectors. Page counts range from 18 to 1,183 pages, with an average of 303.7 pages (SD = 162.9) and a median of 276 pages, suggesting a right-skewed distribution driven by a minority of very extensive reports. File sizes range from under 1 MB to 205.7 MB, with an average of 14.2 MB (SD = 15.6 MB) and a median of 9.6 MB, consistent with the page count distribution. The total corpus amounts to approximately 10.2 GB and 222,313 pages of sustainability disclosure content.

Table 2: Descriptive statistics of the ESG report dataset.

Statistic	File Size (MB)	Page Count
Min	0.3	18
Max	205.7	1,183
Mean	14.2	303.7
Median	9.6	276.0
Std. deviation (SD)	15.6	162.9
Total	10,405.2 MB	222,313 pages

4.2 RQ1: extraction models

Our pipeline uses the ESG information extraction models to build prompts for the LLM. To assess the degree in which LLMs are reliable when used with those prompts, we measure the agreement

of the extracted data by *gpt-4.1-mini* and *gemini-2.0-flash*, starting from the same extraction model.

Table 3 shows the agreement statistics, distinguishing between the extraction of the reporting ESRS topics (102 values) and the company profile (10 values). The agreement score is the fraction of fields on which both LLMs assigned the same value in the whole dataset. We observe that both extraction tasks yield mean agreement scores above 60%. However, the ESRS topic extraction task shows a notably higher standard deviation (18.9% vs 12.5%), indicating that agreement is more variable across companies for ESRS topic extraction than for company profile extraction.

Table 3: Inter-LLM agreement statistics for the extraction of reporting ESRS topics and company profiles.

Metric	Reporting ESRS Topics	Company Profile
# Companies	732	732
# Fields compared	102	10
Mean agreement	61.3%	64.1%
Median agreement	64.2%	63.7%
Std. deviation	18.9%	12.5%
25th percentile	54.7%	55.9%
75th percentile	72.7%	72.5%
Min / Max	0.3% / 100.0%	26.5% / 100.0%
% ≥ 80%	10.0%	9.8%

Figure 6 displays both agreement histograms, where the x-axis is the agreement score, and the y-axis is the percentage of companies having such score. The figure allows inspecting the per-company agreement distribution. We can observe a longer tail of low agreement values (below 20%) for ESRS topic extraction, which is also reflected in the higher standard deviation.

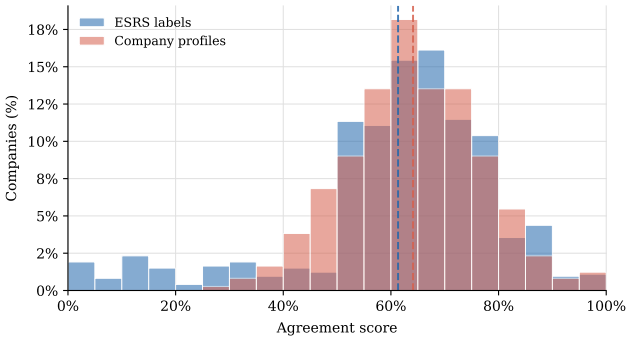

Figure 6: Extraction agreement histograms (mean values are depicted with dashed lines).

Answering the RQ. The results suggest that ESG information extraction models produce moderately consistent outputs across the two LLMs considered, with mean agreement above 60% for both extraction tasks. However, the non-negligible standard deviation values, particularly for reporting ESRS topics, indicates that data extraction guided by ESG information extraction models is not yet fully reliable, and requires further guidelines.

4.3 RQ2: validation rules

To assess the value of our DSL for data cleaning, we defined 30 validation rules checking hierarchical coherence (as in Listing 3) and empty values (as in Listing 4). For hierarchical coherence, we considered top-down rules (a section summary is 0 but one of its subtopics is 1) and bottom-up rules (a subtopic is 1 but the section summary is 0). We applied the rules in checkonly mode to the raw data extracted by each LLM independently, to assess the internal consistency of their outputs prior to any fixing. Table 4 summarises the results, and Figure 7 shows the breakdown by ESRS section and violation direction (top-down or bottom-up). Overall, violations to hierarchical coherence were detected in both directions, but no empty value violation.

Table 4: Internal consistency violations. Top-down (TD): section summary is 0 but a subtopic is 1. Bottom-up (BU): a subtopic is 1 but the section summary is 0.

	gpt-4.1-mini			*gemini-2.0-flash*		
ESRS Section	**TD**	**BU**	**Total**	**TD**	**BU**	**Total**
E1 Climate change	15	8	23	1	1	2
E2 Pollution	98	37	135	16	8	24
E3 Water	258	224	482	49	30	79
E4 Biodiversity	217	87	304	97	25	122
E5 Resource use	49	21	70	3	2	5
G1 Governance	80	21	101	130	38	168
S1 Own workforce	88	8	96	182	40	222
S2 Supply chain	303	31	334	53	21	74
S3 Communities	153	72	225	61	32	93
S4 Consumers	302	135	437	181	17	198
Total	**1,563**	**644**	**2,207**	**773**	**214**	**987**
Per company			**3.01**			**1.35**

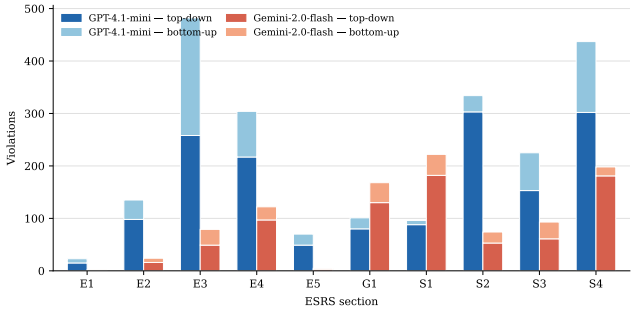

Figure 7: Internal consistency violations per ESRS section and LLM, split by violation direction (top-down / bottom-up).

gpt-4.1-mini produced 2,207 violations across 732 companies (3.01 per company on average), and *gemini-2.0-flash* produced 987 violations (1.35 per company), less than half as many. All violations are from hierarchical coherence rules, representing logical contradictions between a section summary and its subtopics that cannot be attributed to genuine ambiguity. This indicates that *gpt-4.1-mini* is substantially less internally consistent than *gemini-2.0-flash* in its ESRS topic extractions.

Both LLMs produced violations in all ESRS sections, but the distribution differs. *gpt-4.1-mini* concentrates its violations in E3 (Water),

S4 (Consumers), and S2 (Supply chain), with top-down violations dominating in all sections. By contrast, *gemini-2.0-flash* has its highest violation counts in S1 (Own workforce), S4 (Consumers), and G1 (Governance), with a more balanced mix of top-down and bottom-up violations. S4 is the only section that ranks among the most problematic for both LLMs, suggesting that consumer-related disclosures are inherently harder to classify consistently. E1 (Climate change) yields the fewest violations for both models, consistent with it being the most prominent and well-defined ESRS topic.

Answering the RQ. The results indicate that the automated extraction task produces internal inconsistencies in the generated data, which validation rules can help uncover.

4.4 RQ3: enhanced prediction consistency

To address this RQ, we compare the consistency of predictions of four classifiers: two trained on raw data extracted by *gemini-2.0-flash* and *gpt-4.1-mini*, and two trained on the data after being corrected by the 30 validation rules from the previous RQ. For this experiment, we split the dataset in two equally sized subsets of 366 reports each: one for training the classifiers, and the other for obtaining predictions. Since we do not have ground truth data, we do not compare classification performance (e.g., accuracy, F1 score). Instead, we assess the consistency of the predictions by measuring the extent to which they satisfy the validation rules.

Table 5 summarises the results on consistency of predictions on unseen data (i.e., on the second subset of reports). While *gemini-2.0-flash* produces 32 violations affecting 2 companies (0.09 violations per company), *gpt-4.1-mini* produces 316 violations affecting 128 companies (0.86 per company). Both classifiers trained on fixed data produce no violations.

Table 5: Consistency violations across predictions (366 companies, 30 validation rules).

Violations	*gemini*	*gpt*	*gemini-fixed*	*gpt-fixed*
TD rules	24	174	0	0
BU rules	8	142	0	0
Total	32	316	0	0
Per company	0.09	0.86	0.00	0.00

Answering the RQ. Classifiers trained on data cleaned using the validation rules produce more consistent predictions. In our experiment, the predictions generated by these classifiers did not violate any of the consistency constraints encoded by the validation rules.

4.5 Threats to validity

Internal validity. For RQ1, there is no ground truth. Agreement measures inter-LLM consistency but not correctness. Thus, both LLMs could systematically agree on a wrong value, inflating apparent reliability. Without a human-annotated gold standard, we cannot distinguish consensus from shared error. This lack of ground truth also hinders a meaningful precision-based comparison of classifier predictions for RQ3. We leave for future work the manual annotation of reports, and expect to collect ground truth data when the web tool enters in production.

External validity. Our results are specific to *gpt-4.1-mini* and *gemini-2.0-flash*, and other LLMs may produce different agreement levels. Moreover, the 732 companies may not be representative of the full population of reporting companies. For instance, if large companies with detailed reports are over-represented, then agreement scores may be high compared to smaller or less structured reports.

Construct validity. The construct *reliable* in RQ1 is operationalised as inter-LLM agreement, which is an indirect proxy. High agreement is necessary but not sufficient for reliability.

Conclusion validity. The ESRS agreement distribution has a long low tail (standard deviation 19%), thus the mean is not fully representative. Conclusions based on the mean alone may overstate typical performance. Likewise, mean per-company agreement scores treat all topics equally, regardless of how often a topic is relevant. Rare ESRS topics that are almost always not relevant may artificially inflate agreement through trivial consensus.

5 Related Work

In the area of sustainability, we can find solutions directed to help ESG reporting, sustainability management platforms, and AI-based ESG analysis tools. These groups of solutions – both industrial and research-oriented – are analysed next.

Enterprise ESG reporting. In this category, we find platforms like IBM's Envizi[6], Watershed[7] or Workiva[8]. Their common theme is supporting organisations in the collection, consolidation, and disclosure of ESG information. These platforms typically provide workflows for gathering sustainability-related data from heterogeneous sources, mapping indicators to established reporting frameworks (e.g., GRI, SASB, ISSB, and CSRD/ESRS), and generating auditable sustainability reports. Their primary objective is to facilitate the compliance and governance processes associated with ESG disclosure, rather than automatically interpreting existing reports or extracting insights from third-party disclosures.

In research, Hoan *et al.* [13] propose an agentic approach to manage the lifecycle of ESG reporting, where multiple AI agents extract ESG information, verify ESG performance, and update ESG reports based on organisational outcomes.

Sustainability management. This group includes platforms like Enablon[9], novisto[10], pulsora[11] and Sygris[12]. All feature the integration of ESG data management, sustainability strategy execution, and performance monitoring capabilities. These platforms provide organisations with centralised repositories for ESG metrics, tools for managing sustainability initiatives, materiality assessments, carbon accounting functionalities, and support for regulatory compliance. Compared with enterprise ESG reporting platforms, sustainability management solutions focus more broadly on operationalising sustainability practices across an organisation, including target definition, progress tracking, and internal decision support.

AI-based ESG analysis. This is the tool category more closely related to our work. Recent approaches explore the use of LLMs to analyse

[6] https://www.ibm.com/products/envizi/emissions-management
[7] https://watershed.com/
[8] https://www.workiva.com/
[9] https://www.wolterskluwer.com/en/solutions/enablon/
[10] https://novisto.com/
[11] https://www.pulsora.com/
[12] https://sygris.com/

sustainability disclosures and extract structured ESG information from unstructured documents. EulerESG [6] extracts reporting metrics (e.g., energy consumption, greenhouse gas emissions) from ESG PDF reports using LLM-based pipelines. It further provides an LLM-powered chatbot that enables users to query and explore the content of individual sustainability reports. Similarly, ChatClimate [19] provides an LLM-based approach to analyse reports and supports interactive question-answering. Moreover, it assigns a score to the reports reflecting to what extent they follow TCFD guidelines (a UK standard [22]). ESGReveal [26] uses LLMs and RAG to extract sustainability metrics from ESG reports. In contrast, our work focuses on building an ESG recommender system, and the model-based extraction pipeline enables the accommodation of heterogeneous disclosure frameworks and disclosure metrics.

Analysis of ESG data is important for investors (e.g., for risk mitigation, since ESG-compliant companies are less exposed to regulatory fines, supply chain disruptions and reputational damage, and may have better access to finance [23]), and specialised stock market benchmarks track the performance of companies meeting specific ESG criteria. Examples are the S&P 500 ESG Index[13], or the Spanish IBEX ESG Index[14]. Hence, some research works analyse ESG datasets in connection with stock investment [16].

6 Conclusions and Future Work

We have presented a model-based machine learning pipeline for producing ESG recommenders for sustainability reporting. The pipeline includes a model-based extraction phase powered by LLMs, and a cleaning and validation phase that uses rules built with a YAML-based DSL. Our experiments show that LLMs can use the extraction models to produce company profiles and ESRS reporting data with moderate agreement. However, the extraction task is complex, hence validation rules are needed to detect inconsistencies. These results indicate the need for human supervision of extracted data. Our YAML-based validation rules and flexible ESG information extraction models (which support context fields summarising findings in the reports) are directed to ease this task. Classifier prediction improves when trained with automatically fixed data.

This is an ongoing industry–university collaborative research project, with some missing objectives to achieve. Regarding the classifier, a path to explore is to combine data extracted by several LLMs using consolidation strategies (e.g., majority voting). We also aim at finishing the web-based reporting platform itself, and designing strategies for effective retraining of the classifiers.

Acknowledgments

Work sponsored by the Madrid Region (project IA4SustainabilityReport) and the Spanish MICIU (project PID2024-155231OB-I00).

References

[1] Takuya Akiba, Shotaro Sano, Toshiki Yanase, Takeru Ohta, and Masanori Koyama. 2019. Optuna: A Next-generation Hyperparameter Optimization Framework. In *KDD*. ACM, 2623–2631.
[2] Australian Accounting Standards Board (AASB). 2024. Climate-related Financial Disclosure (AASB S2 / ISSB alignment). https://standards.aasb.gov.au/aasb-s2-sep-2024.
[3] James Bergstra and Yoshua Bengio. 2012. Random Search for Hyper-Parameter Optimization. *J. Mach. Learn. Res.* 13, 281–305.
[4] Leo Breiman. 2001. Random Forests. *Mach. Learn.* 45, 1 (2001), 5–32.
[5] Tianqi Chen and Carlos Guestrin. 2016. XGBoost: A Scalable Tree Boosting System. In *KDD*. ACM, 785–794.
[6] Yi Ding, Xushuo Tang, Zhengyi Yang, Wenqian Zhang, Simin Wu, Yuxin Huang, Lingjing Lan, Weiyuan Li, Yin Chen, Mingchen Ju, Wenke Yang, Thong Hoang, Mykhailo Klymenko, Xiwei Zu, and Wenjie Zhang. 2025. EulerESG: Automating ESG Disclosure Analysis with LLMs. http://arxiv.org/abs/2511.21712
[7] European Commission. 2019. European Green Deal. https://commission.europa.eu/strategy-and-policy/priorities-2019-2024/european-green-deal_en.
[8] European Commission. 2022. Corporate Sustainability Reporting Directive. https://eur-lex.europa.eu/legal-content/EN/TXT/?uri=CELEX:32022L2464.
[9] European Commission. 2023. Commission Delegated Regulation (EU) 2023/2772 of 31 July 2023 supplementing Directive 2013/34/EU as regards sustainability reporting standards (ESRS). https://eur-lex.europa.eu/eli/reg_del/2023/2772/oj. Official Journal of the European Union, accessed: June 2026.
[10] European Commission. 2026. European Single Access Point (ESAP). https://www.esma.europa.eu/esmas-activities/data/european-single-access-point-esap. Accessed: June 2026.
[11] European Financial Reporting Advisory Group (EFRAG). [n. d.]. European Sustainability Reporting Standards (ESRS). https://www.efrag.org/sustainability-reporting/esrs. Accessed: June 2026.
[12] Global Reporting Initiative. 2021. GRI Standards. https://www.globalreporting.org/standards/. Accessed: 2026-06-16.
[13] Thong Hoang, Mykhailo Klymenko, Xiwei Xu, Shidong Pan, Yi Ding, Xushuo Tang, Zhengyi Yang, Jieke Shi, and David Lo. 2026. ESG Reporting Lifecycle Management with Large Language Models and AI Agents. http://arxiv.org/abs/2603.10646
[14] International Sustainability Standards Board (ISSB). 2023. IFRS Sustainability Disclosure Standards: IFRS S1 and IFRS S2. IFRS Foundation. Issued June 2023. https://www.ifrs.org/issued-standards/ifrs-sustainability-standards-navigator/
[15] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. 2017. LightGBM: A Highly Efficient Gradient Boosting Decision Tree. In *NIPS*. 3146–3154.
[16] Ook Lee, Hanseon Joo, Hayoung Choi, and Minjong Cheon. 2022. Proposing an Integrated Approach to Analyzing ESG Data via Machine Learning and Deep Learning Algorithms. *Sustainability* 14, 14 (2022).
[17] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. In *NeurIPS*. Curran Associates, Article 793, 16 pages.
[18] Yaxi Lu, Haolun Li, Xin Cong, Zhong Zhang, Yesai Wu, Yankai Lin, Zhiyuan Liu, Fangming Liu, and Maosong Sun. 2025. Learning to Generate Structured Output with Schema Reinforcement Learning. In *ACL (1)*. ACL, 4905–4918.
[19] Jingwei Ni, Julia Bingler, Chiara Colesanti-Senni, Mathias Kraus, Glen Gostlow, Tobias Schimanski, Dominik Stammbach, Saeid Ashraf Vaghefi, Qian Wang, Nicolas Webersinke, Tobias Wekhof, Tingyu Yu, and Markus Leippold. 2023. CHATREPORT: Democratizing Sustainability Disclosure Analysis through LLM-based Tools. In *EMNLP (Demos)*. ACL, 21–51. See https://www.chatclimate.ai/.
[20] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, and Édouard Duchesnay. 2011. Scikit-learn: Machine Learning in Python. *J. Mach. Learn. Res.* 12 (2011), 2825–2830.
[21] Sergio Segura, Gordon Fraser, Ana Belén Sánchez, and Antonio Ruiz Cortés. 2016. A Survey on Metamorphic Testing. *IEEE Trans. Software Eng.* 42, 9 (2016), 805–824.
[22] Task Force on Climate-related Financial Disclosures (TCFD) reporting requirements. 2021. Sustainability reporting requirements. https://www.fca.org.uk/firms/climate-change-sustainable-finance/reporting-requirements.
[23] Albert Tsang, Tracie Frost, and Huijuan Cao. 2023. Environmental, Social, and Governance (ESG) Disclosure: A Literature Review. *The British Accounting Review* 55, 1 (2023), 101149.
[24] U.S. Securities and Exchange Commission. 2024. Final Rule: Climate-Related Disclosures for Investors. https://www.sec.gov/files/rules/final/2024/33-11275.pdf.
[25] XBRL International. 2026. The XBRL Standard. Accessed: June 2026. https://www.xbrl.org/the-standard/
[26] Yi Zou, Mengying Shi, Zhongjie Chen, Zhu Deng, Zongxiong Lei, Zihan Zeng, Shiming Yang, Hongxiang Tong, Lei Xiao, and Wenwen Zhou. 2025. ESGReveal: An LLM-Based Approach for Extracting Structured Data from ESG Reports. *Journal of Cleaner Production* 489 (2025), 144572.

[13] https://www.spglobal.com/spdji/en/indices/sustainability/sp-500-scored-and-screened-index/
[14] https://www.bolsasymercados.es/en/bme-exchange/indices/esg.html