

I

# InnoClass AI

## EPO Spring codefest 2025

A platform to manage AIs and data for SDG classification

Prepared for EPO

Created by Henry Fritz Chatchueng Tamo (Working Student at Deloitte GmbH, Computer Science  
Student at University of Düsseldorf)

and

Rodolphe D’Inca (EPO Patent examiner)

Demonstrator available on <https://innoclass.alkemata.com>

username: epo

code: codechallenge25

# 1. Executive summary

We have developed **InnoClass**, a sustainable AI platform that classifies patents according to the 17 United Nations Sustainable Development Goals (SDGs).

Instead of optimising a single model in isolation, we built an end-to-end platform that automates **data extraction, model training, deployment and monitoring**. Key features include:

- **SDG demonstrator** – a working pipeline that tags patent texts with SDG labels and exposes the results through an API and a web application.
- **Carbon-footprint monitoring** – energy sensors (CarbonCode) track hardware usage, and the system defaults to CPU inference or off-the-shelf models to minimise GPU time.
- **Reproducible pipelines** – data and model workflows are orchestrated by *Dagster*, ensuring every run is logged, versioned and easy to rerun with new data or parameters.

## 2. Problem analysis and Exploration of the solution space

### 1. The challenge

*Use AI to classify patents according to the Sustainable Development Goals (SDGs).*

Our initial step involved a precise technical definition of the problem and an examination of existing solutions for classifying **Sustainable Development Goals (SDGs)**. Previous efforts by various organizations and companies, particularly for scientific texts, often employed **lexical characterization** using keyword lists. More advanced solutions utilized **Large Language Models (LLMs)** like BERT (e.g., SDGBert) for classifying texts into specific SDGs.

---

### Challenges with Existing LLMs

Fine-tuning LLMs for SDG classification necessitates **reference classifications**, which are labor-intensive to compile. We used one such reference classification to initiate our training. While **ChatGPT 3** demonstrated the ability to classify patents accurately without explicit SDG labels (leveraging its prior knowledge), its current use is impractical due to **cost and time constraints**. This indicates a need for a more **lightweight and specialized model**.

Our initial exploration involved analyzing the problem and investigating potential solutions through several exploratory notebooks. While these notebooks are archived, they are not ergonomically usable in their

current state.

---

## Understanding the SDGs and Classification Difficulties

The 17 SDGs are characterized by both a short and an official long title. Their broad nature, covering a vast range of technical fields (or even non-technical fields like SDG 17), makes developing precise reference classifications challenging due to subjective interpretation. A common approach seen in prior research involves using **multiple classifiers with majority voting**. While we didn't directly implement this, the concept of integrating multiple classifiers could refine our developed reference classification. We emphasize the critical importance of a robust mechanism for defining the reference classification, as it directly impacts the LLM's effectiveness and the quality assurance for users.

---

## Exploring Zero-Shot and Few-Shot Learning

A more ideal solution would be to employ a **zero-shot or few-shot LLM** (essentially a decoder like ChatGPT) to circumvent the need for extensive training reference sets. However, attempting to use a local model like **Llama 2**, which is lighter than ChatGPT, proved unconvincing. Llama 2 possesses significantly fewer parameters than ChatGPT, and its intrinsic context is insufficient for comprehending complex technical texts. While training such a model on scientific or patent texts could enhance its knowledge, its size would likely lead to substantial computational costs.

---

## LLM Context and Vocabulary Alignment

An LLM's intrinsic context refers to the general knowledge it acquires during its training. The SDG titles typically use simple, straightforward language, which contrasts sharply with the highly technical vocabulary found in patents. Therefore, when selecting or training LLMs for SDG classification of patents, it's crucial to consider the training datasets used. The LLM needs to be trained on data that enables it to effectively bridge the gap between this simpler vocabulary and the more specialized, technical terms prevalent in patents. This allows the model to establish meaningful connections between the two linguistic styles.

## 2. Classification vs. Semantic Search for SDG Alignment

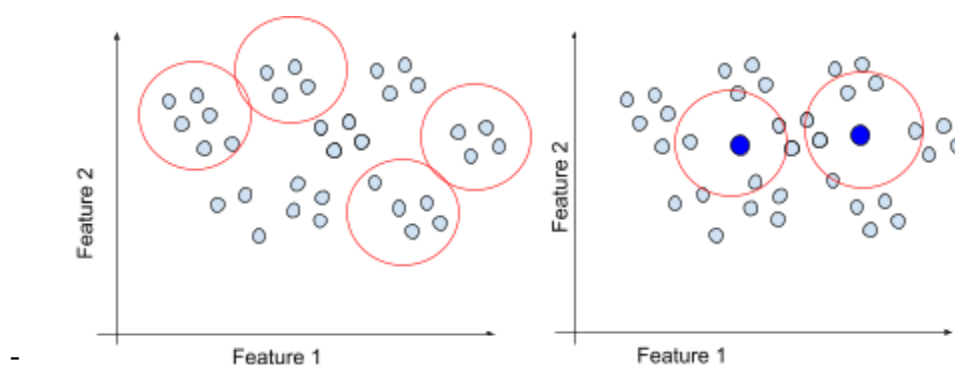
Initially, our primary challenge revolved around defining the appropriate conceptual framework: classification or semantic search. This decision fundamentally dictates the model architecture and training methodology. Traditional classification groups semantically similar documents. However, standard classifiers, lacking inherent knowledge of content labels, typically gauge "distance" between textual features to identify close neighbors. This approach is effective for narrow, homogeneous technical categories like Cooperative Patent Classification (CPC) codes.

However, SDGs present a unique challenge: a single SDG can encompass vastly disparate technical fields. In

such cases, common features might fall below the detection threshold of traditional classifiers. The crucial distinction with SDGs lies in their titles, which provide a rich semantic context. This led us to hypothesize that semantic search would be a more suitable approach for patent classification than pure classification. The core idea is to use the SDG title as a query (or "prompt") and retrieve patents with semantic content closely aligned with that SDG.

### Focusing on Encoders for Semantic Search

For these reasons, our focus shifted to encoders like BERT and particularly SBERT (Sentence-BERT), which are better suited for semantic search. SBERT excels at generating embeddings (vector representations) for entire sentences in a feature space. The semantic content of a sentence is encoded into a mathematical vector, where each dimension represents a specific semantic feature. The similarity between two texts can then be quantified by measuring the distance between their corresponding encoded vectors.



*Simplified comparison of classification and semantic search in feature space. The documents are represented in light blue. On the left, the classifier in red looks for close neighbours and detects groupings of documents: the classes. On the right, the label of the classes is used in dark blue and embedded in feature space: they serve as anchor for the semantic search (circle in red) to find documents around them.*

We further explored various aspects of this model, including:

- **Asymmetric vs. Symmetric Queries:** This involved considering the length disparity between the SDG title and the patent text. One idea was to break down the patent text into smaller sentences for more symmetric comparisons.
- **Pretraining:** We investigated pretraining methods like Masked Language Modeling (MLM) or TSDAE (Transformer-based Sequential Denoising Autoencoder) to enhance the model's understanding of domain-specific language.
- **Reranking with a Cross-Encoder:** As a final step, we studied the possibility of using a cross-encoder for reranking. A cross-encoder jointly encodes the SDG and the text, allowing for a more nuanced understanding of their interrelationships and highlighting direct connections.

### 3. Patent Specifics and Content Extraction for SDG Classification

Having addressed the core classification methodology and model selection, our attention shifted to the specific documents for classification: patents. While most existing studies focus on scientific articles, patents possess distinct characteristics that are advantageous for our task. Patents are typically more structured and

often exhibit greater clarity.

Of particular importance is the "background of the invention" section. This section provides crucial context regarding the problem the invention aims to solve. It generally contains less legalistic language and offers insights into the invention's broader context, which directly aligns with the objectives of SDG classification. Furthermore, this section is usually located near the beginning of the patent and is relatively concise. The main drawback is that it can sometimes be a neglected part of the document and, due to its lower legal significance, may not always be rigorously scrutinized by examiners. Despite this, its informational value for SDG mapping remains significant.

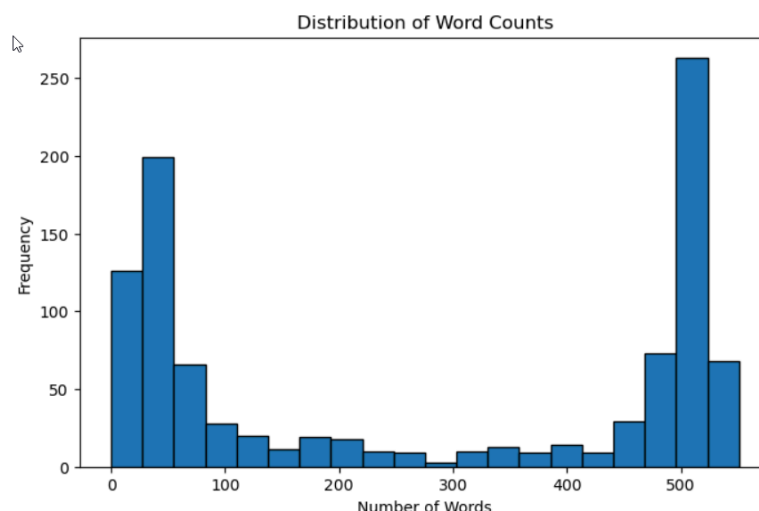
## Extracting Relevant Patent Information

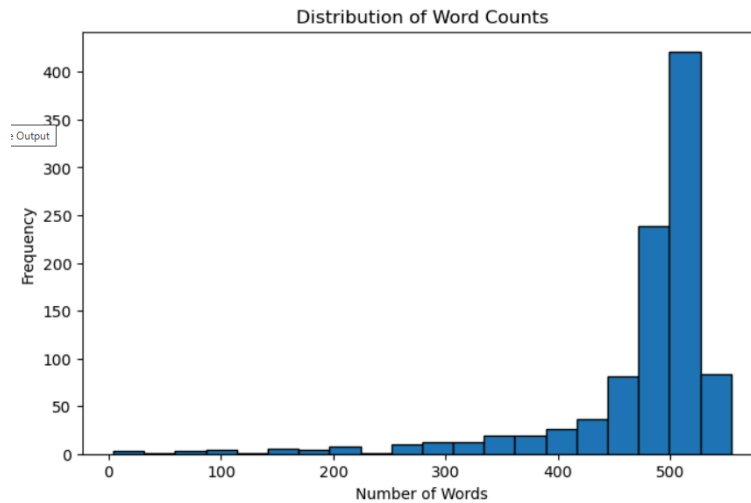
We initially explored using T5 for summarization to extract this background information, even when not explicitly labeled. However, concerns regarding computational time and the potential for embedding an already LLM-summarized text into another LLM led us to revert to a more traditional, lexical extraction solution.

Our approach involved a three-step process:

1. **Heading Extraction:** We first attempted to extract content under headings containing keywords such as "background."
2. **Keyword Location in Paragraphs:** If no explicit headings were found (e.g., due to OCR issues), we then searched for specific keywords within the `<p>` tags of the text.
3. **First 500 Words Extraction:** As a fallback, if neither of the above methods yielded results, we extracted the first 500 words of the document. The goal was to capture as much useful information as possible within approximately 600 tokens, suitable for input into our embedders.

To support this extraction process, we developed SQL queries in BigQuery. These queries allowed us to extract text distributions based on various criteria, such as length, time period, or random selection. This capability proved invaluable for assessing different scenarios and refining our extraction strategy.





*Evolution of the extracted text length distribution during the exploratory phase*

## 3. Design of the solution

### A. Project Specifications

This project considers three primary user types and their corresponding use cases:

1. **European Patent Office (EPO) Internal Users:** The EPO requires a robust platform capable of generating and continuously updating patent classifications in a streaming fashion.
2. **Casual Users:** This group needs a user-friendly web application for easily searching and consulting patents within a specific SDG category. This application would serve as an enrichment to existing EPO documentation (e.g., EPODOC).
3. **Advanced Users:** These users require an extension to the existing EPO API, enabling them to develop custom tools and applications based on the SDG classification data.

Beyond user-specific needs, the system must adhere to the following general requirements:

- **Meaningful Results:** The classification outputs must be reasonably accurate and relevant.
- **Deployability:** The demonstrator should be easily deployable for production environments.
- **Scalability:** The system must scale effectively with increasing document volumes, supporting multi-server architectures and GPU acceleration.
- **Sustainability:** The approach should prioritize limiting computational power consumption, or at least provide mechanisms to measure energy usage.
- **Flexibility and Robustness:** The system needs to be adaptable and resilient to the rapid advancements in Artificial Intelligence.

### B. Core Principle

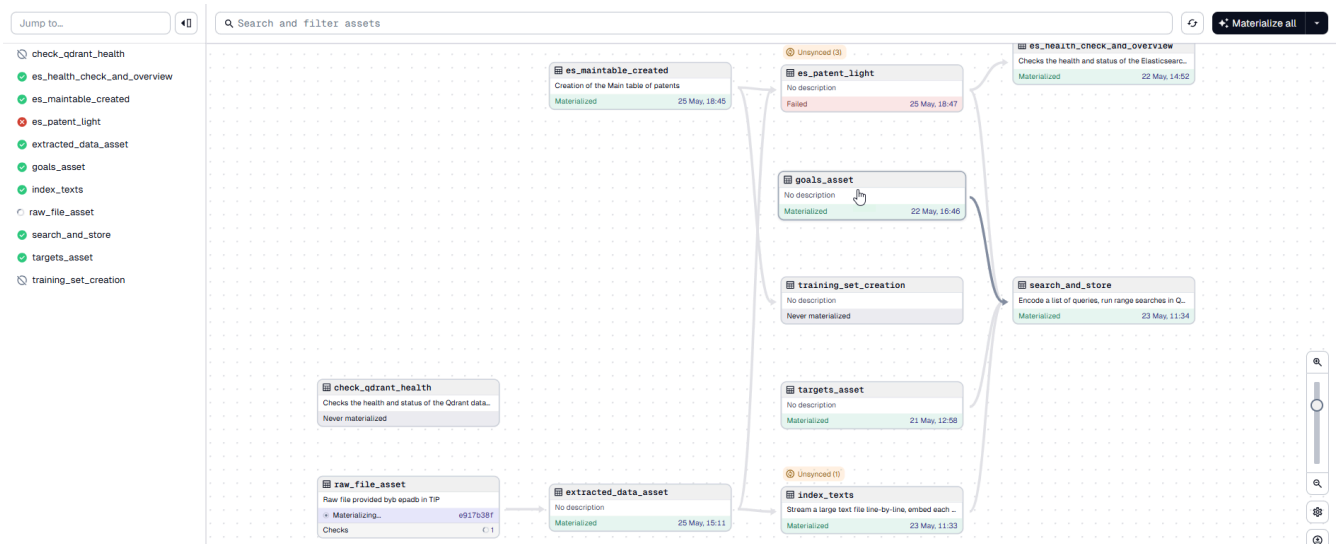
The proposed solution operates on the following principle:

1. **Lexical Extraction:** Relevant background excerpts are extracted from patent documents using lexical methods.
2. **Vector Embedding:** Both the extracted patent texts and the SDG labels are converted into high-dimensional numerical vectors (embeddings).
3. **Nearest Neighbor Range Search:** A semantic search is performed to find the nearest neighbors between the patent text embeddings and the SDG label embeddings. A predefined similarity threshold is applied to identify the most relevant documents and assign them to the corresponding SDG category.

## C. System Architecture

The system architecture is divided across two platforms:

1. **The TIP EPO Platform:** This platform hosts notebooks used for data extraction and fine-tuning models based on user feedback.
2. **A Dedicated Linux Server:** This server hosts the data pipeline, which performs the following functions:
  - Transforms data received from the TIP notebooks (via SSH).
  - Extracts relevant background information from patents.
  - Performs vector embedding.
  - Saves the resulting data in databases.
3. The data pipelines are managed by a data orchestrator (Dagster) using Python assets, ensuring compatibility with the exploratory work done on the TIP platform. The system logs all data runs, including parameters and results, and uses sensors connected to the databases and assets to signal updates.



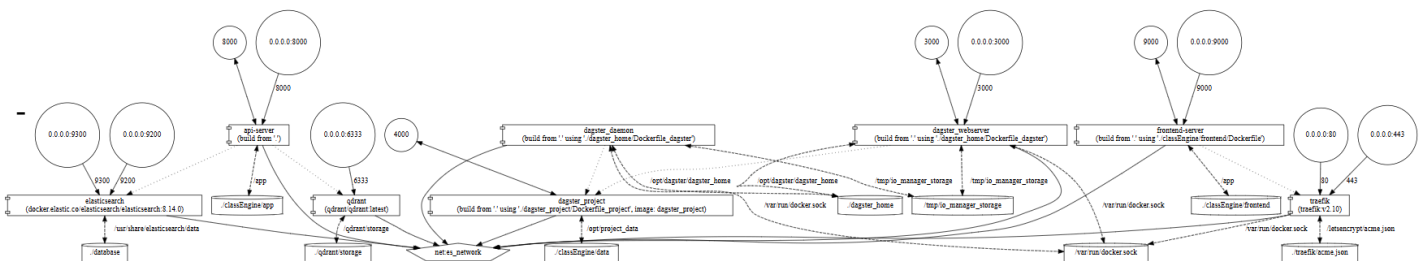
*Dagster pipeline orchestrator*



The following key components are used:

- **Transformers (SBERT, SPECTER):** These tools, based on SBERT and utilizing a SPECTER model trained on scientific data, are used to embed both extracted patent texts and SDG labels into vectors. This vector representation captures the semantic content of sentences in both the texts and labels. The base model is automatically downloaded and installed using the HuggingFace libraries.
- **QDrant Vector Database:** QDrant is used to store the calculated vectors along with their payload (ID, patent ID). It was chosen over ElasticSearch for its native vector database design (rather than an added feature), GPU support for indexing, and its ability to perform range searches. This range search capability is crucial for retrieving vectors with a similarity score above a given threshold for a specific SDG vector.
- **ElasticSearch Database:** A "standard" ElasticSearch database is used to locally replicate a small patent database, populated from the EPO database (via the TIP notebooks). This local database includes the added SDG categories and fields for user feedback on classification and for flagging documents for manual review or fine-tuning. ElasticSearch was selected due to its existing use within the office, minimizing deployment effort. While it offers powerful lexical search capabilities (e.g., BMS2) and hybrid search algorithms (e.g., HRR), a full semantic search approach was chosen, as the inclusion of lexical constraints was found to destabilize the broader semantic search based on the SDG labels.
- **FastAPI Server:** A Python-based FastAPI server, served by Uvicorn, provides the interface between the internal data pipelines and the external world (the web application and TIP tools). Due to time constraints, authentication safeguards were not implemented for the demonstrator. However, authentication solutions are highly dependent on the specific operational deployment and existing authentication systems.
- **Web Application:** A web application, built using React, Material UI, and Vite, provides essential functionalities, including a search page with user feedback mechanisms, an interface to the Dagster orchestrator for pipeline management, and a tool for preparing reference files used for fine-tuning the LLM model.
- **TIP Notebook Code Examples:** Code examples are provided within a TIP notebook to demonstrate how to use the FastAPI server to access the classification system.

4. All server-side components are containerized and managed by Docker Compose, behind a Traefik reverse proxy. The container structure is detailed below:



Structure of the Docker services

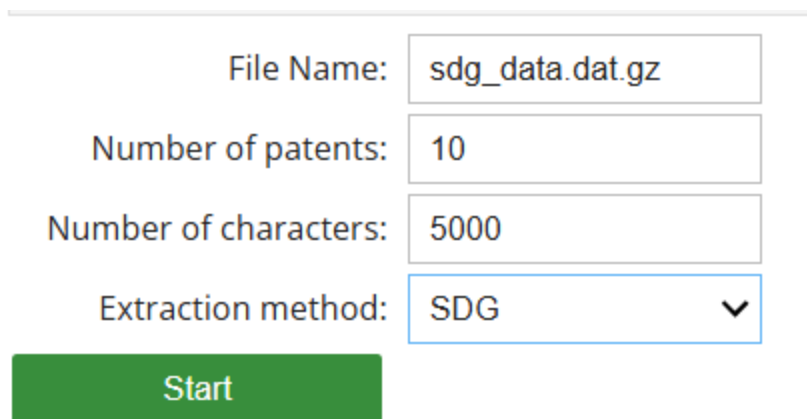
## C. Modes of Operation

The platform is organized around four distinct pipelines, all adhering to the principles outlined previously:

### 1. Classification Pipeline

This pipeline orchestrates the process of classifying patents from data ingestion to final categorization:

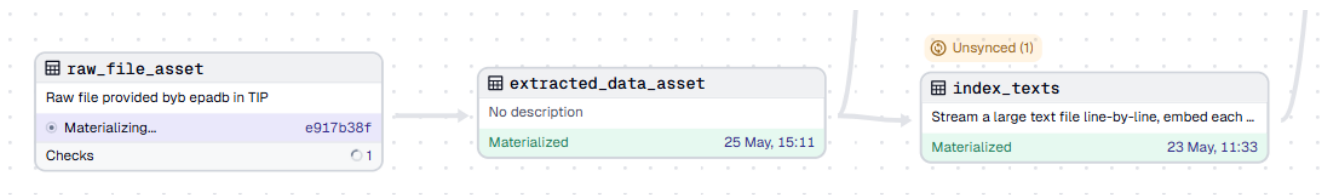
- **Data Ingestion (TIP Platform):** The process begins on the TIP EPO Platform, where patent data is loaded from the central EPO database. A user interface allows for selecting various data distributions:
  - **Random:** A random selection of patents.
  - **Time-based:** Data split equally across ten year ranges.
  - **Length-based:** Distribution according to document length.
  - **SDG-Referenced:** Documents containing explicit references to SDGs, useful for initial reference classification and fine-tuning (SDG numbers are automatically detected and extracted).
  - **Bulk Access:** Batch ingestion of data to populate the server's ElasticSearch database. Once selected, the data and metadata are compressed into a JSONL file, named, and sent via SSH to the main server, which hosts the pipeline orchestrator.



The image shows a web-based form for selecting data from the EPO database. It contains four input fields: 'File Name' with the value 'sdg\_data.dat.gz', 'Number of patents' with the value '10', 'Number of characters' with the value '5000', and 'Extraction method' with a dropdown menu showing 'SDG'. Below these fields is a green 'Start' button.

*Notebook interface in TIP for selection of data in the EPO database*

- **Server-Side Processing (Dagster):** From the main server, the pipeline is managed by Dagster. It detects new files, extracts background data, embeds this data into vectors, stores these vectors in the vector database, and performs a range search for each SDG label. The classification results are then stored in the main ElasticSearch database. Dagster is a data-oriented orchestrator, with the pipeline composed of interconnected "assets" representing data at different stages. For example, assets include the raw files from the EPO database, the extracted background data, and the vectorized data in the vector database. Each asset is programmed in Python, allowing for significant reuse of code from the exploratory phase.



*Ingestion pipeline with corresponding assets for data transformation*

- **Asset Materialization and Logging:** For an asset to "exist" (i.e., for its data to be processed), it must be "materialized," triggering a run based on a specific configuration (e.g., filename, database index). Every run is extensively documented, and logs are stored, facilitating the comparison of different parameters across runs. This architecture is crucial for managing the large number of parameters and models encountered during development. Most assets are also equipped with Carboncode sensors to evaluate their energy consumption during materialization.

## 2. Search Pipeline

The search pipeline enables users to query classified patents:

- **Initiation:** This pipeline can be initiated either from the web application or from the TIP Notebook client. It sends a request to the FastAPI server with the desired search parameters.
- **Endpoint and Interaction:** We've developed a single endpoint that accepts one or more SDGs as parameters. The FastAPI server then queries the Elasticsearch database with this request and sends the results back to the client.
- **User Interface:** The web application, built with React, presents a simplified search interface to the user, and the results are displayed in a clear, organized manner.

### Search SDGs

Select SDGs

Enter Keywords (comma-separated)

SEARCH

*Search interface in the webapp*

### METHOD AND SYSTEM FOR SUSTAINABLE DEVELOPMENT GOAL (SDG)

#### PERFORMANCE ASSESSMENT OF AN ENTERPRISE

Extracted Text: The disclosure herein generally relates to the field of performance assessment, and, more particularly, to method and system for sustainable development goal performance assessment of an enterprise. S

SDGs: SDG1 (Score: 1.00), SDG2 (Score: 1.00), SDG3 (Score: 1.00), SDG4 (Score: 1.00), SDG5 (Score: 1.00), SDG6 (Score: 1.00), SDG7 (Score: 1.00), SDG8 (Score: 1.00), SDG9 (Score: 1.00), SDG11 (Score: 1.00), SDG13 (Score: 1.00), SDG14 (Score: 1.00), SDG15 (Score: 1.00) | Targets: N/A

Up Votes: 0 | Down Votes: 0

### A COMPUTER-IMPLEMENTED MODEL FOR MEASURING AND REPORTING SOCIAL IMPACTS OF SOCIAL ENTERPRISES

Extracted Text: The invention belongs to the field of data processing systems or methods, specially adapted for financial, managerial, supervisory, and administrative purposes, more precisely to the field of data pro

SDGs: SDG1 (Score: 1.00), SDG2 (Score: 1.00), SDG3 (Score: 1.00), SDG4 (Score: 1.00), SDG5 (Score: 1.00), SDG6 (Score: 1.00), SDG7 (Score: 1.00), SDG8 (Score: 1.00), SDG11 (Score: 1.00), SDG12 (Score: 1.00), SDG13 (Score: 1.00), SDG14 (Score: 1.00), SDG15 (Score: 1.00), SDG17 (Score: 1.00) | Targets: N/A

Up Votes: 0 | Down Votes: 0

### PERFORMANCE MEASURING SYSTEM MEASURING SUSTAINABLE DEVELOPMENT RELEVANT PROPERTIES OF AN OBJECT, AND METHOD THEREOF

Extracted Text: Performance Measuring System Measuring Sustainable Development Relevant Properties Of An Object, and Method Thereof Field of the Invention The present invention relates to performance measuring system

SDGs: SDG1 (Score: 1.00), SDG2 (Score: 1.00), SDG3 (Score: 1.00), SDG4 (Score: 1.00), SDG5 (Score: 1.00), SDG7 (Score: 1.00), SDG8 (Score: 1.00), SDG9 (Score: 1.00), SDG10 (Score: 1.00), SDG11 (Score: 1.00), SDG13 (Score: 1.00), SDG17 (Score: 1.00), SDG121 (Score: 1.00) | Targets: N/A

Up Votes: 0 | Down Votes: 0

This pipeline allows users to provide valuable feedback on the classification results, contributing to continuous improvement:

#### 4. User-Feedback Driven Model Improvement

This pipeline closes the loop, using validated user feedback to refine the classification model:

- **Reference File Management:** The data manager can review and validate the reference file directly within the web application. The interface displays the patent title, the extracted background text (to verify extraction accuracy), and the associated classification, which can be checked or unchecked. Once reviewed, the file can be marked as "validated" and "confirmed as reference." (Note: There is currently a minor bug where a changed file moves to the end of the list.)

The screenshot shows the InnoClass web application. At the top, there's a blue header with the 'InnoClass' logo and a 'Menu' button. Below the header, there's a navigation bar with a 'Filter by Reference' dropdown set to 'Not Validated', a 'Show All' button, and an 'APPLY FILTERS & FETCH FIRST' button. The main content area displays a patent document titled 'CROWDFUNDING 4.0: A NOVEL INFLUENCE-BASED GLOBAL FUNDRAISING PLATFORM AND SYSTEM'. Below the patent text, there's a section for 'Sustainable Development Goals (SDGs)' with a table for selection.

Select	SDG
<input type="checkbox"/>	SDG1: No Poverty
<input type="checkbox"/>	SDG2: Zero Hunger
<input type="checkbox"/>	SDG3: Good Health and Well-being
<input type="checkbox"/>	SDG4: Quality Education
<input type="checkbox"/>	SDG5: Gender Equality
<input type="checkbox"/>	SDG6: Clean Water and Sanitation

Checking page for patents with user feedback

- **Model Fine-Tuning:** These validated reference data can be extracted from the pipeline as a CSV file and copied to the TIP platform. A dedicated fine-tuning notebook then processes this data, splitting it into training, validation, and test sets. An existing model can be loaded and fine-tuned using this new data.

```
--- Starting Model Fine-tuning ---
Error displaying widget: model not found

[3706/3706 32:44, Epoch 1/1]

Step  Training Loss  Validation Loss  Sts-dev Pearson Cosine  Sts-dev Spearman Cosine
-----
1000      0.087200      No log              0.820134              0.791458
2000      0.073700      No log              0.845762              0.808162
3000      0.069800      No log              0.855036              0.814432
3706      0.072200      No log              0.858982              0.817386

Model fine-tuning complete. Model saved to 'output/sbert_finetuned_sdg'.
```

Finetuning of SBERT model in TIP notebook

- **Testing and Deployment:** The fine-tuning process is followed by a testing phase, which provides essential metrics on the model's performance. Once the model is trained and validated, it can be transferred to the main server and integrated into the pipeline as the new classifier. At this point, all existing data will need to be re-classified, either in a single batch or, preferably, progressively.

## 4. Conclusion and perspective

We have developed a comprehensive demonstrator platform specifically designed for managing data and AI models related to **SDG classification**. This strategic choice was driven by the rapid evolution of AI models, which can quickly render existing models and their associated parameters obsolete. This flexible platform is also extensible, capable of supporting other use cases and diverse AI tasks.

The broad interpretation of SDGs necessitates a flexible model that can adapt the **reference data** used for validating classifications. Furthermore, our work on Sustainable Development Goals inherently required us to uphold these same principles. Recognizing that 30 teams might concurrently optimize similar models using powerful GPUs, leading to potentially damaging redundancies, our approach was to provide a platform that helps coordinate and implement these optimizations. We've also integrated **Carbon sensors** into our code to evaluate electrical consumption, operating on the principle that "what is well known, is well managed." To avoid detrimental redundancies, we recommend coordinating this SDG work with other relevant organizations, enabling data and trained models to become shared resources.

We acknowledge that this project is currently a demonstrator and would require detailed knowledge of the hardware infrastructure and specific usage patterns for full production deployment. However, all essential components are in place, and the effort required to establish a fully operational platform would be minimal.

Looking ahead, it appears that databases built on vast amounts of collected data, such as the one we've created, may soon become outdated. With the emergence of even more powerful AI, possessing **agentic capabilities** and utilizing **Model Context Protocol servers**, it seems more probable that AI will autonomously search for and classify raw data on-the-fly using its intrinsic knowledge. But that, as they say, is another story.