CRISP-DM Report

Data Mining Course Project

DSAI track

Anastasia Pichugina
Anna Gromova
Arthur Gubaidullin
Adewuyi Israel
Khush Patel

May 2025

Contents

1 Business Understanding

1.1 Business Objectives

1.1.1 Background

Since scientific research became publicly available to everyone via the Internet, more and more researchers have been appearing online. More people have started reading about new technologies and, as they develop, trying to contribute to scientific fields. ArXiv is a free distribution service and an open-access archive for nearly 2.4 million scholarly articles in the fields of physics, mathematics, computer science, quantitative biology, quantitative finance, statistics, electrical engineering and systems science, and economics [[1]]

However, some industries are developing faster than others, and gaps in the distribution of research areas arise. We were approached by I. Ivanov, head of the Marketing Department at ArXivData, to solve this problem, highlighted by the steering committee, to improve the user experience and the efficiency of researchers (our problem areas).

The current solution consists of manual literature reviews and citation analyses, which is very time-consuming, biased, and close to failure, which can lead to irrelevant results and lost time and profits. To solve this problem, our company proposes to use data and technologies to understand which areas are developing worse than others, because new research in uncommon areas can lead to a boost in science and the growth of other areas, provided that research is related or new approaches emerge.

1.1.2 Business Objectives

Automatic identification of underrepresented or underexplored research areas ("gaps") in the arXiv dataset becomes the main business objective. Automatization can bring more accurate and relevant research gaps, they will motivate researchers to be pioneers in the topic and invest efforts in new undiscovered areas, and extremely new and breakthrough research conducted to fill these gaps will enable businesses to cover even larger areas and continue to maintain the bar for advanced service.

Additional business questions from a customer that might appear are:

- Which subfields show slowing growth despite high initial activity?
- Which areas are leading in terms of research quantity?
- How can unpopular but promising researchers find and connect with more experienced ones?

In addition to the main objectives we have several business requirements:

- Results must be interpretable by domain experts
- Outcomes must be relevant

The steering Committee will check our results for veracity.

We expect this project and achieved goals will bring us and ArXivData the following benefits: duplicated efforts in saturated fields are decreased, underrepresented areas had more funding from international funds and institutes, acceleration of innovative outcomes in novel directions

1.1.3 Business Success Criteria

We assume to assess the prosperity of our project by:

- 30% of prioritized gaps receive targeted funding or institutional hiring within 12 months.
- 20% decrease in submissions to saturated categories within 2 years post-implementation.

ArXivData's leads will keep an eye on papers publication and assess the success.

1.2 Assess Situation

1.2.1 Inventory of Resources

The first part of the situation assessment is resources inventory. Regarding the Data and Knowledge sources we are supposed to get a dataset with publications on ArXiv, that will be a structured online source and source of textual data with titles, abstracts, and categories which papers refer to.

We have no powerful available hardware for Machine Learning tasks, however we can rent them from Yandex Cloud. Also we can utilize Google Colab or Kaggle platforms. Also, during the implementation there will be such tools as Python (for the data mining task) and its frameworks (scikit-learn, pytorch, nltk, etc.).

1.2.2 Requirements, Assumptions and Constraints

The project requires a massive dataset with papers and comprehensible columns, a well-defined schedule with clear timelines, and powerful hardware: CPU and GPU for model training. Also, full access to data must be allowed and the produced model must be interpretable and accurate. We assume the dataset contains meaningful and genuine information without fake records. Key constraints are time limit and computing power.

1.2.3 Risks and Contingencies

The project is susceptible to risks such as a lack of computing resources, financial constraints, and lack of time. Also, there is a risk that researchers will not be interested in rare topics and will refuse to conduct them on their own.

1.2.4 Costs and Benefits

As we already mentioned, for the data mining problem we need data and computational resources. The obtained dataset should be massive, thus we need about 5GB of SSD for its storage. Furthermore, effective model training requires external GPU powers, available RAM, and space for saving model weights. Yandex Cloud allows one to rent these tools, so we can reserve as much power as we need. Here we can conclude that SSD, RAM, and GPU are what we look for, and compute the project's cost per month: 11.91*10 + 0.28*24*30 + 1.05*24*30 = 1076.7 rubles where 11.91 0.28 and 1.05 monthly fee for SSD (5Gb for data + 5Gb for weights), hourly fee for RAM and hourly fee for GPU correspondingly.

As a benefit, we will get:

- More papers with rare topics: identified underexplored areas help researchers focus on high-impact and novel topics instead of saturated fields.

- Science boost:
  - Minimized duplicated efforts by flagging overlapping studies early and focus on small topics is increased.
  - Highlighted interdisciplinary opportunities for cross-domain partnerships.
- Absence of human factor: subjective expert opinions will be replaced with quantifiable and automated gap metrics.
- Money is saved: funding on overstudied topics are reduced.

1.3 Determine Data Mining Goals

1.3.1 Data Mining Goals

The primary business goal is to automatically identify underrepresented areas to guide researchers and funding institutions toward novel topics. To achieve this, the data mining goals are:

1. Topic discovery via unsupervised clustering: group research papers into semantically coherent clusters to uncover natural research trends, independent of predefined arXiv categories using 'abstract' and 'title' NLP embeddings.
2. Find the underrepresented clusters that may indicate research gaps. This will bring us and our customers to a clear understanding of niche areas (smaller clusters = smaller area's study).
3. Preserve scalability and robustness: process the dataset efficiently. Giant models for embeddings generation and clustering might not be as effective as they seem to be. Start with small models and analyze their performance. In case of lack of accuracy proceed with bigger models.

Our Data Mining problem type is clustering with unsupervised learning: primary method for topic discovery.

All these points lead to informative outcomes about underrepresented fields, which can be researched more. According to our Business Objectives, this will motivate explorers to become pioneers in those topics.

1.3.2 Data Mining Success Criteria

Our technical success and models performance will be measured by:

- Density-Based Validity (DBCV) more than 0.2. This metric ensures clusters are structurally meaningful.
- Silhouette Score more than 0: minimal separation required (accepting trade-offs for coverage of rare topics).

Clustering results allow to identify areas which are overpopulated and underrepresented.

2 Data Understanding

2.1 Data Collection

The dataset was collected by Cornell University to analyze the state of scientific research from 2007 to 2025. The data set was extracted from the Arxiv website and published on Kaggle platform, from which our team downloaded it. No problems were encountered during the data collection phase.

2.2 Data Description

The data were acquired in dictionary format as a ".json" file which later were transformed into a tabular format as a ".csv" file. The basic "surface" features of the dataset are as follows:

- Quantity of data: 2,689,088 unique datapoints
- Dataframa size: 146 Mb
- Number of features: 7
- Feature data types distribution:
  - Object/Categorical: 5
  - Object/List: 1
  - Date/time: 1
- Features description:
  - "ID": paper unique identifier
  - "Title": paper title
  - "Category": paper category (general_category.more_specific_category). General categories are: Computer Science, Economics, Electrical Engineering and Systems Science, Mathematics, Physics, Quantitative Biology, Quantitative Finance, Statistics.
  - "Abstract": paper abstract
  - "Authors": paper authors
  - "Authors_parsed": authors parsed to list[list[string]]. However, the inner data type is still string and not list.
- Missing values: 0 (0.0%)

2.3 Data Exploration

Here is more precise description of each variable:

- "Title" - a title of the research paper;
- "Category" - each paper have one or more categories resulting in 120 unique categories;

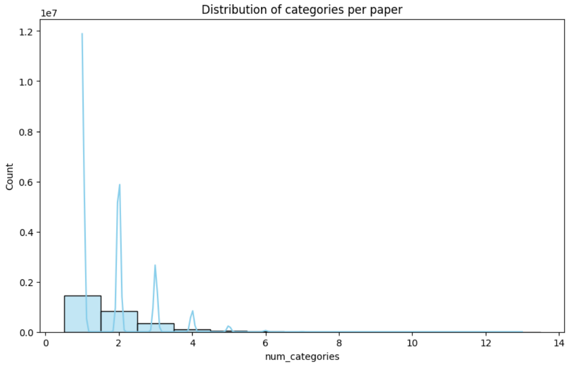

Figure 1: Distribution of categories per paper.

- ”Abstract” - a classic abstract paragraph that contains main information about paper;
- ”Authors” and ”Authors_parsed” - a list of authors for each paper. Unfortunately, it has only a short version of the name and the surname (with exceptions). The most frequent author participated in 2600 papers;
- ”Update date” - the update/publication date for each paper. Ranges from May 2007 to March 2025.

2.3.1 Categories

Each research paper may have more than one category. The top 5 most popular categories are (also see Fig 2):

Category	Count
cs.LG	210055
hep-ph	186039
hep-th	172311
quant-ph	159114
cs.CV	148938

As it was mentioned, some papers have more than 1 category. We checked the distribution of categories amount per paper, and most of them have 1, 2, or 3 categories (Fig 1). There are papers with more categories but it is not a frequent situation. Most categories occur in less than 75 000 research papers.

Also, Fig 3. demonstrates how category popularity was changing over time. Computer Science categories got a huge boost after the 2017.

To see categories connectivity we built a network graph (Fig 5). Here we noticed that cs.GL (Computer Science, General Literature) isn’t a familiar category. We assume that authors don’t mark their papers as ”General Literature” because their work contain more exploratory meaning and not explanations of some areas. We decided not to include the edges with less than 10 occurrences (categories were met together less than 10 times) to avoid mess on the plot. So, we can conclude that some categories tend to be met in a paper together more frequently than other.

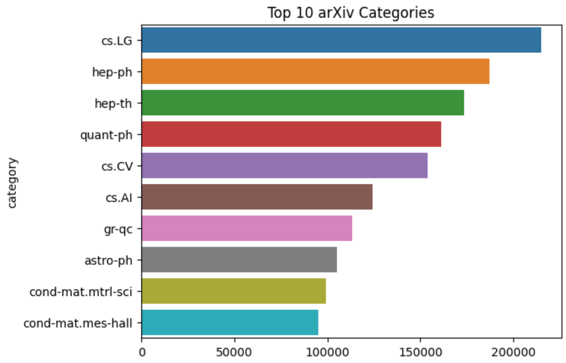

Figure 2: Top 10 most popular categories.

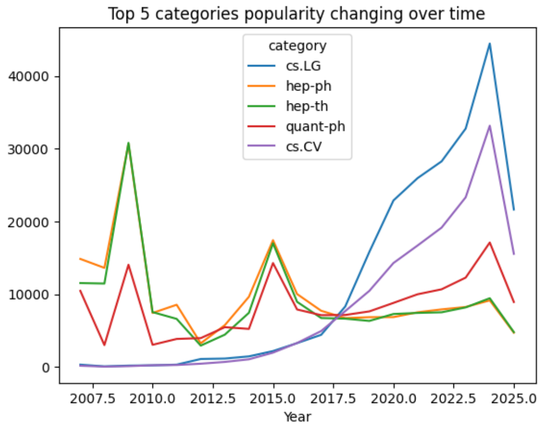

Figure 3: Top 5 categories over time.

We also checked paired co-occurences another diagram with categories connectivity (Fig 4). Here we see that the most collaborative categories are math and cs (Computer Science) topics.

The EDA notebook can be found here

2.3.2 Authors

Initially, we had 2 variables that represent authors:

- Authors: String that contains author short names and surnames separated either by ”,” or ”and”;
- Authors parsed: String that contains each author name and surname enclosed in the brackets. However, it is still a string, so, parsing and transforming this string into a list of strings with author names may be more difficult than parsing ”Authors” variable.

Here we see that most of papers have 1-4 authors, and 2 authors is the most frequent bar (Fig 6). Moreover, we see that the average number of authors per paper increases over time (Fig 7)

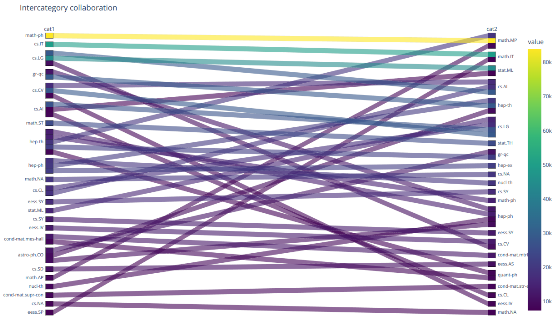

Figure 4: Top 10 most popular categories.

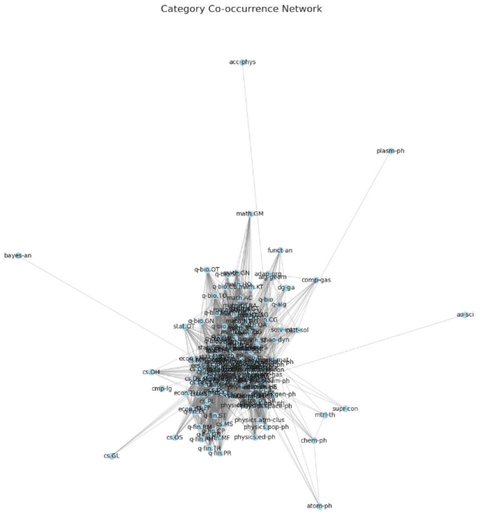

Figure 5: Top 5 categories over time.

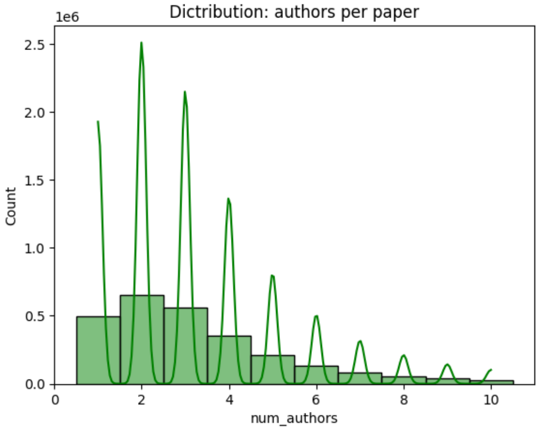

Figure 6: Distribution of authors per paper.

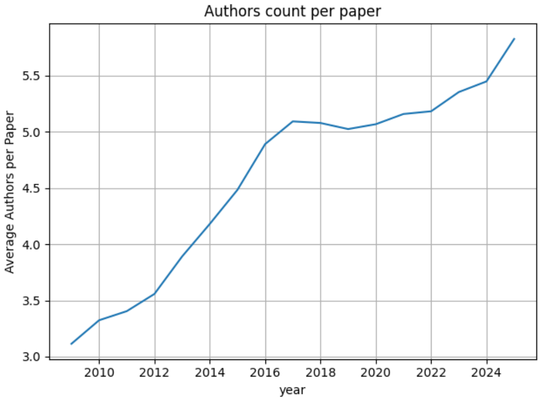

Figure 7: Average authors per year.

2.3.3 Update date

As it was stated above, the dates of paper update range from May 2007 to March 2025.

However, if we take a look at the amount of papers that were updated at the earliest date (2007-05-23) you will see this:

Top 5 most active authors:

Author	Count
Y. Zhang	2644
Yang Liu	1987
Y. Wang	1893
J. Wang	1891
Z. Wang	1701

Figure 8: Top authors by publication count.

Update date	Count
2007-05-23	129984
2007-05-24	45
2007-05-25	64
2007-05-28	30
2007-05-29	58

Such amount of updates at the 23rd of May 2007 may mean that every research paper that has "Update date" = 2007-05-23 were published/updated no later than the 23rd of May 2007:

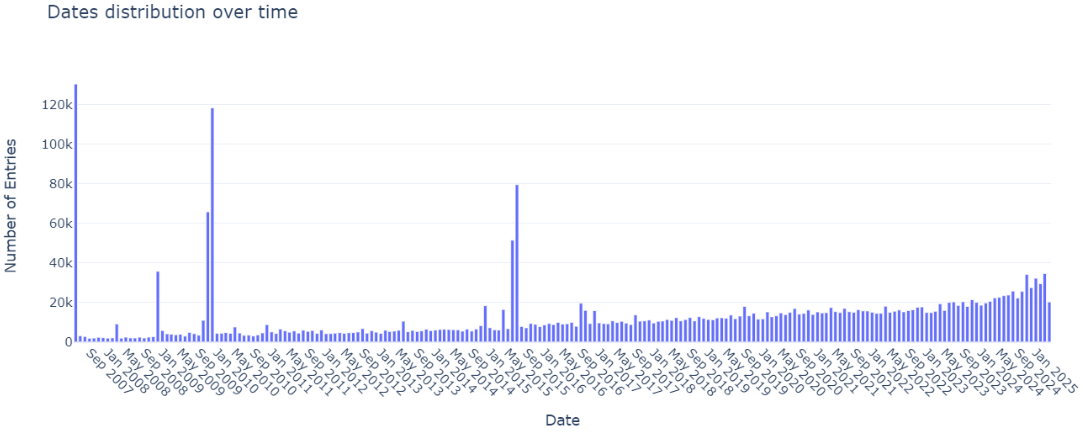

Figure 9: Dates distribution over time.

As we can see from the image above "peaks" present among different dates:

- As we discussed earlier, the first one may mean that quite a lot of publications were published/updated before the 23rd of May 2007;
- The second one occured during the period from the beginning of October to the end of November of 2009;

- The third one occured during the period from the beginning of May to the end June of 2015.

2.3.4 Abstract

To understand the distribution of abstracts' words we built a plot with top 30 words (Fig 10). We see, that the most popular are: ”model”/”models”, ”data”, ”results”, ”quantum”, ”energy”. Although, the most of them are general (”two”, ”method”, ”study”, ”theory”, ”large”), we understand that a lot of papers relate to Data Science field (a lot of words ”model”), physics, and probably ecology (word ”energy” might appear in papers of both (physics and biology/ecology) categories).

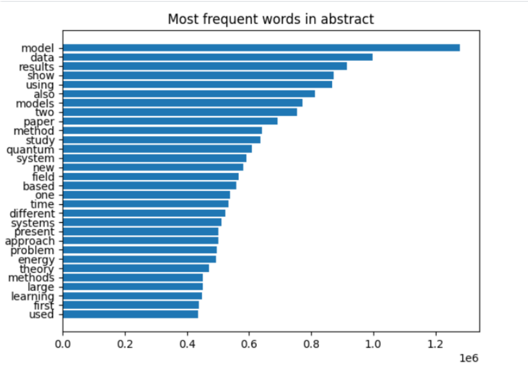

Figure 10: Most frequent words in abstract.

2.4 Data Quality

After completing the basic Exploratory Data Analysis (EDA), we may conclude that the dataset covers a sufficient range of research fields based on categories of publications and large amount of researchers contributing to the world of research.

This dataset can be used for the following tasks:

- Trend Analysis and Forecasting;
- Research Impact Prediction;
- Author Disambiguation and Collaboration Networks;
- Automated Paper Categorization;
- Research Gap Identification.

3 Data Preparation

3.1 Select columns

In relevance with our business goals and by extension data mining goals, we have decided to use the columns: Title, Abstract and Update date. The title and abstract columns will be used to generate the embedding, which will will be used to cluster the paper by topics. The Update date column will be used to analyze the clusters across time.

3.2 Clean data

As mentioned in the data exploration, the dataset doesn't have any vacant cells!

3.3 Construct data

During the modelling phase, we concatenate the Title and Abstract column and the new text string is what we feed into the embedding model. The team decided to do the concatenation on the fly, as this makes it easier to work with the data.

3.4 Integrate Data

For our project we don't need any additional datasets except the ArXiv data. So we implement no data ingestion here.

3.5 Format data

As mentioned in the data description, the original data is in .json format and we convert it to .csv format. This makes reading, analyzing, modeling and working with the data much easier.

4 Modelling

4.1 Modelling Technique

HDBSCAN (Hierarchical Density-Based Spatial Clustering of Applications with Noise) was selected as the primary clustering technique due to its ability to identify clusters of varying shapes and densities, automatically determine the number of clusters, handle noise effectively, and outperform traditional DBSCAN for varying-density clusters.

It's also important to note that clustering is an unsupervised task without ground truth labels. The modelling pipeline consists of three stages:

- **Embedding**: Titles and abstracts concatenated during the data processing stage are passed to an embedding model to generate high-dimensional vectors capturing semantic content.
- **Dimensionality Reduction**: UMAP (Uniform Manifold Approximation and Projection) reduces the dimensionality of embeddings to improve clustering performance and computational efficiency.

- **Clustering**: HDBSCAN clusters the reduced embeddings, with hyperparameter optimization to ensure optimal configuration.

4.2 Modelling Assumptions

The following assumptions were made:

- All papers have complete title and abstract information (confirmed during data exploration).
- Text embeddings adequately capture semantic content of research papers.
- Euclidean distance is appropriate for measuring similarity between reduced embeddings.

4.3 Test Design

Clustering performance was evaluated using:

- **Silhouette Score**: To measure how similar papers are to their own cluster compared to other clusters.
- **Noise Ratio**: Fraction of papers classified as noise.
- **DBCV (Density-Based Clustering Validation)**: Validates density-based clustering quality.
- **Temporal Trend**: Tracks the rate of change in cluster paper counts over time.

Clusters were labeled by selecting the closest 30 papers to each cluster center, feeding their abstracts to an LLM, and generating concise summaries. Predefined arXiv categories were not used as ground truth to allow discovery of natural topic clusters.

4.4 Building Model

Two experiments were conducted to balance model quality and computational feasibility given the dataset's 2.7 million rows.

- In the first experiment, we sampled 100K rows from 2.7M rows, ran the pipeline with a larger and more advanced model.
- In the second experiment, we used the full dataset with a significantly smaller embedding model and ran the pipeline with this as well.

We did not have good priors on what the optimal hyperparameters would be, both for UMAP dimensionality reduction and for the HDBSCAN model. This led us to conduct a **hyperparameter search**.

4.4.1 Hyperparameter Search

For HDBSCAN, we ran the hyperparameter search over the following parameters

- `min_cluster_size`: [20, 25, 50, 100]
- `min_samples`: [3, 5, 7, 10, 15, 20]
- `alpha`: [1.0, 1.5]

and kept the following parameters constant

- `cluster_selection_method`: ['eom'],
- `hdbscan_use_approximate_predict`: [True],
- `cluster_selection_epsilon`: [0.0],

For UMAP dimensionality reduction, we ran hyperparameter search over the following variables

- `umap_n_components`: [2, 5, 10, 20, 30, 40]
- `umap_n_neighbors`: [5, 10, 15, 20]

In total, this gave us 1152 possible combinations to try.

4.4.2 Experiment 1: Sampled Dataset with SOTA Embedding Model

A random sample of 100,000 papers was selected to test a state-of-the-art (SOTA) embedding model, `gte-Qwen2-2.5B-instruct`, a 1536-dimensional 2.5B transformer-based model optimized for semantic similarity. The pipeline was:

- **Embedding**: Generated 1536-dimensional embeddings for concatenated title and abstract.
- **Hyperparameter Search**: Conducted a grid search over HDBSCAN parameters and UMAP parameters. We selected the optimal configs by filtering the configs for those that yielded
  - `Noise ratio` > 0.3
  - `num_clusters` > (1 + 3.322 * log10(data_size))
  - Of the configuration left, we take the config with the max silhouette score.

  Code snippet for the optimal config selection looks like:

```
def select_best_config(results, max_noise=0.3, min_clusters=None):
    df = pd.DataFrame(results)

    # Auto-calculate min_clusters if not provided
    if min_clusters is None:
        min_clusters = calculate_min_cluster_threshold(df)

    # Filter valid configurations
    valid_configs = df[
        (df[’noise_ratio’] < max_noise) &
        (df[’n_clusters’] >= min_clusters)
    ]

    if len(valid_configs) == 0:
        print("Warning: No configurations met criteria!")
        print("Relaxing noise constraint...")
        valid_configs = df[df[’n_clusters’] >= min_clusters]

    # Select config with highest silhouette score
    best_idx = valid_configs[’silhouette’].idxmax()
    return valid_configs.loc[best_idx].to_dict()
```

The result of the most optimal config was as follows:

  - `min_cluster_size`: 50
  - `min_samples`: 3
  - `alpha`: 1.0
  - `umap_n_components`: 2
  - `umap_n_neighbors`: 10

- **Dimensionality Reduction**: Applied UMAP to reduce embeddings to `umap_n_components` dimensions.
- **Clustering**: Applied HDBSCAN with the optimal configuration based on Silhouette Score, DBCV, and noise ratio, as described above.
  - **Labeling**: We use an LLM to label each clusters
  - **Temporal trends**: We define this as the rate of change of each cluster i.e the rate at which each cluster increases. A code snippet is shown below

```
def analyze_trends(trends_pivot, recent_years=5, low_count_threshold=0.035, min_papers=
    total_papers = trends_pivot.sum().sum()
    low_count_absolute = total_papers * low_count_threshold

    analysis = pd.DataFrame({
        ’Total_Papers’: trends_pivot.sum(),
        ’Mean_Annual_Papers’: trends_pivot.mean(),
        ’Recent_Papers’: trends_pivot.tail(recent_years).sum()
    })

    # Calculate growth rate (average annual change, normalized by mean papers)
    growth_rates = []
    for cluster in trends_pivot.columns:
        counts = trends_pivot[cluster]
        if counts.mean() > 0:
```

```
        annual_change = counts.diff().mean() / counts.mean()
    else:
        annual_change = 0
    growth_rates.append(annual_change)
analysis['Growth_Rate'] = growth_rates
```

- **Selection**: Then we select the top 12 clusters by growth rate and the bottom 12 clusters as well, and this is present in out visualization in Fig. 12

Results:

- **Silhouette Score**: 0.219, indicating average cluster separation.
- **Noise Ratio**: 0.041, with 4.1% of points classified as noise. This is great, as almost all the papers belong to one cluster or another.
- **DBCV**: 0.58, reflecting strong density-based cluster validity.
- **Cluster Count**: 239 clusters identified, with sizes ranging from 50 to 27K papers. See `cluster_size_100000.csv`

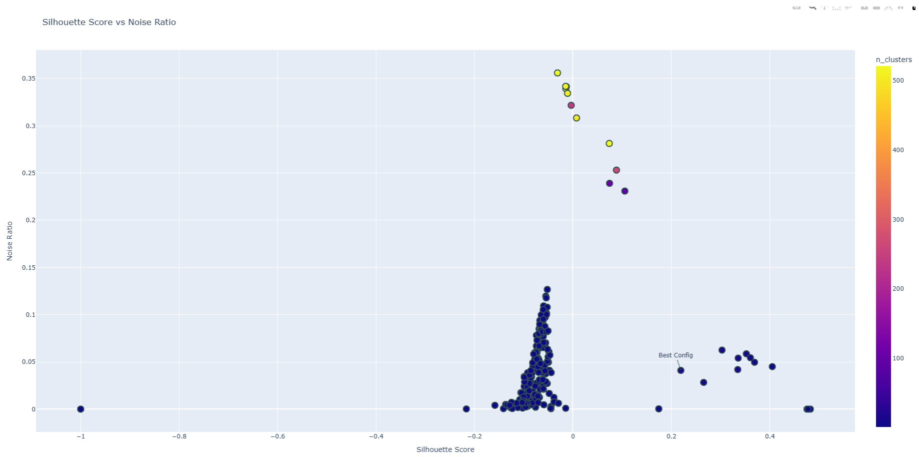

Figure 11: Visualization of hyperparameter search (Silhouette score VS Noise ratio)

A more interactive visualization file is present on the repository.

4.4.3 Experiment 2: Full Dataset with Efficient Embedding Model

The full 2.7 million-row dataset was processed using a lightweight embedding model which is really used in the literature: `sentence-transformers/all-MiniLM-L6-v2`, a 384-dimensional, 22M transformer model designed for speed. The pipeline was:

- **Embedding**: Generated 384-dimensional embeddings for concatenated title and abstract.

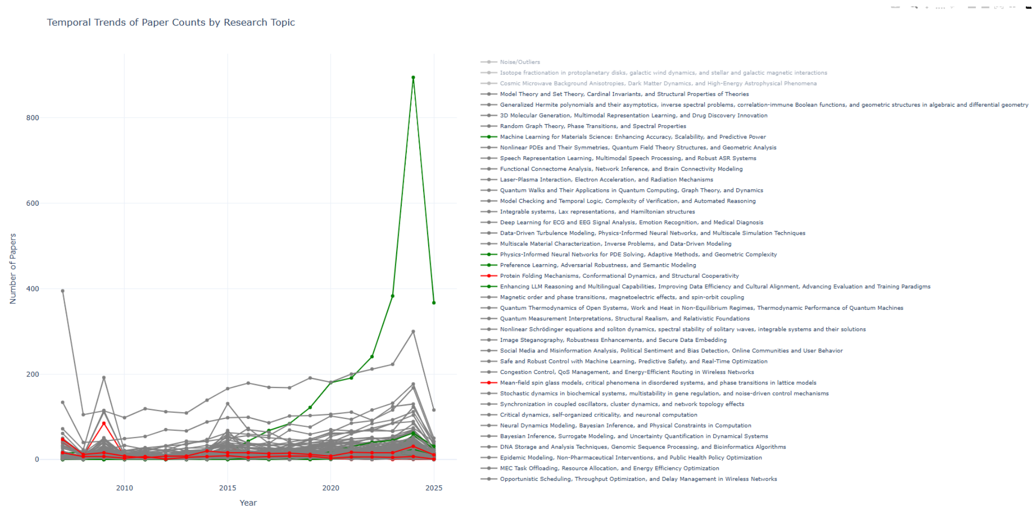

Figure 12: Visualization of the 239 clusters identified

- **Hyperparameter Search**: We did not conduct this due to computational limits. To use best config though, we used the result from the hyperparameter search from the 100K step, and manually tweaked the `n_components` for a reasonable noise ratio.
- **Dimensionality Reduction**: Applied UMAP with `n_components=7`, based on the optimal configuration from Experiment 1.
- **Clustering**: Applied HDBSCAN using the optimal hyperparameters from Experiment 1.

Results:

- **Silhouette Score**: 0.002, indicating poor cluster separation.
- **Noise Ratio**: 0.43, with a significant portion of the papers being classified as noise.
- **DBCV**: 0.22
- **Cluster Count**: 638 clusters identified

4.5 Metrics Comparison

The SOTA model on the sampled dataset yielded higher Silhouette Score (.219 vs .002) and DBCV (0.58 vs. 0.22), reflecting better cluster quality due to richer embeddings. However, the lighter model on the full dataset produced more clusters (239 vs. 638), capturing a broader range of topics despite a higher noise ratio (0.43 vs. 0.041). The optimal configuration from the sampled experiment proved slightly effective for the full dataset, ensuring consistency while scaling to the larger data volume.

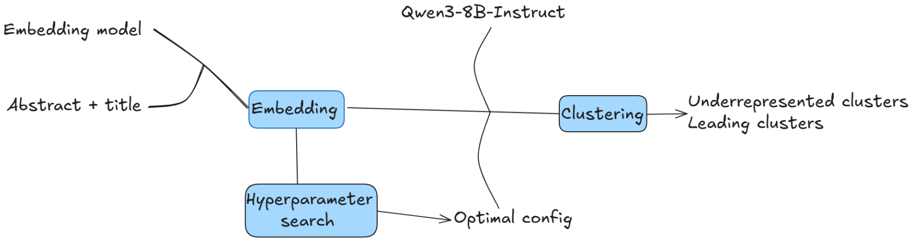

Figure 13: Description of the pipeline.

5 Evaluation

After completing the modeling phase, we did a carefull evaluation of our results to make sure they properly address the business objectives. We examined both the technical and practical quality of the models.

5.0.1 Assessment of Data Mining Results with Respect to Business Success Criteria

Our evaluation showed promising results in meeting the business objectives. The first experiment using the sampled dataset model achieved a Silhouette Score of 0.219 and DBCV of 0.58, indicating meaningful cluster separation. We successfully identified 239 distinct research clusters. This allowed us to analyze growth trends across different fields. The temporal analysis revealed both growing and declining research areas, which addresses the business need for identifying underrepresented fields.

We used LLM to label the clusters, which allowed us to meet our business requirement for interpretable results, as domain experts can easily understand the cluster descriptions.

Our approach provides a systematic method to find potential areas for investment. The decrease in saturated category submissions will require longer-term monitoring after deployment.

5.0.2 Approved Models

Based on our evaluation, we approved the model from Experiment 1 (100K sample with gte-Qwen2-2.5B-instruct embeddings) for several reasons. First, it achieved significantly better clustering metrics than the full-dataset model. Second, its lower noise ratio (4.1% vs 43%) means more papers are properly classified. While this model was trained on a sample, we believe it can represent the research picture better.

5.1 Review Process

We did a thorough review of our data mining process to identify potential improvements or oversights. The main strength was our systematic approach to hyperparameter tuning, which ensured optimal model performance. We also used the most relevant and permitted features (Title, Abstract and Update date) for our modeling and analysis.

However, there were some limitations. Because of computational constraints we could not run the optimal embedding model on the full dataset. What is more some interdisciplinary papers might be misclassified due to our clustering approach. This could be improved by receiving domain expert feedback to validate clusters first.

The data quality remained high throughout the project, all transformations were properly documented and reproducible. We only used available attributes that would be present for future analysis.

5.2 Determine Next Steps

5.2.1 List of Possible Actions

We considered several options for the next step:

1. **Deploy the approved model**: Implement the clustering system for arXiv to identify research gaps. This would immediately provide value but might miss some patterns from the full dataset.

2. **Full dataset processing**: Get additional computing resources to run the better embedding model on all 2.7M papers. This would improve results but require more time and budget.

3. **Hybrid approach**: Use the approved model for initial deployment while processing the full dataset in parallel. This balances immediate needs with long-term improvements but is complex to manage.

4. **Expand model features**: Incorporate citation networks or author information to enhance clustering. This could provide richer insights but would require additional data preparation.

5.2.2 Decision

We decided to proceed with Option 3 (Hybrid approach). Our current model already meets our objectives and can provide immediate value for the business stakeholders, while we can proceed with full-dataset processing. It can also be easily updated if improved model becomes available.

This approach delivers a working solution that addresses the business need for identifying research gaps, while leaving room for improvements.

6 Repository

We have created a pipeline to run our code semi-automatically and get similar results to the ones we got. All of the data and insights we gathered have been uploaded to the `outputs` folder which can be reproduced by following the steps in the repo linked below.

- Dev branch repo with necessary steps included.

7 Distribution of the work

Roles description:

- PM (Project Manager) - Uniform among all stages
- BU (Business Unit) - 0.6 Business Understanding, 0.2 Data Understanding, 0.2 Evaluation
- DS (Data Scientists) - 1 Data Understanding

- ML (machine Learner) - 0.7 Modeling, 0.3 Evaluation
- BA (Business Analyst) - 0.6 Business Understanding, 0.2 Data Understanding, 0.1 Modeling, 0.1 Evaluation

Name	**PM**	**BU**	**DS**	**ML**	**BA**	**Total**
Anastasia Pichugina	0.25	0.6	0	0.05	0.35	1.25
Arthur Gubaidullin	0.25	0.1	0.8	0	0.1	1.25
Israel Adewuyi	0.25	0	0.2	0.7	0.1	1.25
Anna Gromova	0.25	0.3	0	0.25	0.45	1.25
Khush Patel	0	0	0	0	0	0
Total	1	1	1	1	1	-

Table 1: Work Distribution