A Comprehensive Analysis of Survey Papers on Large Language Models

Upama Thapa Magar
Department of Computer Science
Boise State University
UpamaThapamagar@u.boisestate.edu

Abstract

As Large Language Models (LLMs) have become more popular in recent years, the number of survey papers in this field has greatly increased. This large amount of research can be difficult for new researchers to navigate. In this paper, I analyze the metadata of these LLM survey papers, looking at publication trends and grouping them by research focus areas. I build a feature matrix using simple methods like TF-IDF vectorization and one-hot encoding. I also use a logistic regression model to predict the category of each paper based on its metadata. To address class imbalance in the data, I apply class weighting to improve performance for less common categories. The results show an improvement in accuracy, from 43% to 47%, showing that class weighting helps. This paper provides insights into research trends in the LLM field and helps new researchers explore the literature and find areas for future work.

1 Introduction

AI techniques have been widely applied to various domains, such as images [He et al., 2016, Dosovitskiy, 2020], texts [Vaswani et al., 2017, Devlin et al., 2018], and graphs [Kipf and Welling, 2016, Zhuang and Al Hasan, 2022]. As a critical subset of AI techniques, Large Language Models (LLMs) have gained significant attention in recent years [Radford et al., 2018, 2019, Brown et al., 2020, Achiam et al., 2023, Bai et al., 2022, Team et al., 2023]. Especially, more and more new beginners are interested in the research topics about LLMs. To learn the recent progress in this field, new beginners commonly will read survey papers about LLMs. Therefore, to facilitate their learning, numerous survey papers on LLMs have been published in the last two years. However, a large amount of these survey papers can be overwhelming, making it challenging for new beginners to read them efficiently. To embrace this challenge, in this project, I aim to explore and analyze the metadata of LLMs survey papers, providing insights to enhance their accessibility and understanding [Zhuang and Kennington, 2024]. Specifically, I aim to systematically analyze the metadata of these LLM survey papers by examining publication trends, categorizing research topics, and identifying gaps in the existing literature.

Overall, my contributions can be summarized as follows:

- I conducted a thorough analysis of the metadata from LLM survey papers, identifying key trends in publication dates and categorizing research focus areas.
- I built a feature matrix by applying Term Frequency-Inverse Document Frequency (TF-IDF) vectorization to the textual data and one-hot encoding to categorical data, enabling machine learning analysis.
- I implemented a logistic regression model to classify survey papers based on their metadata, addressing class imbalance through class-weighted techniques.
- I evaluated the performance of the logistic regression model using metrics such as accuracy, precision, recall, and F1-score, and visualized classification performance through a confusion matrix.
- My analysis provides insights into the distribution of research areas within LLM-related studies, helping newcomers navigate the literature more effectively and identify underexplored topics for future research.

2 Related Work

Large Language Models (LLMs) have seen rapid advancements in recent years. GPT-4 by [Achiam et al., 2023] exemplifies this progress, showcasing improvements in text generation and handling complex tasks. [Bai et al., 2022] introduced Constitutional AI, focusing on making AI systems safer through human feedback. Language models like

GPT-3, as explored by [Brown et al., 2020], revolutionized few-shot learning, enabling models to generalize from minimal data. The introduction of BERT by [Devlin et al., 2018] has also become foundational for many NLP tasks.

While these papers highlight advancements in LLM development, relatively few studies have focused specifically on categorizing or analyzing survey papers within the LLM field. Survey papers typically provide overviews of advancements in LLMs, addressing technical challenges, ethical concerns, and application domains. However, research explicitly aiming to systematically categorize and analyze the landscape of LLM-related surveys is scarce.

In contrast to existing works that focus on technical contributions or ethical frameworks, this paper addresses the lack of structured analysis of LLM survey papers. Existing reviews tend to focus on the functionality and performance of specific models, while this work provides a meta-analysis of the survey literature itself. By focusing on publication trends, taxonomy categorization, and identifying underexplored research areas, my analysis provides a new dimension that complements the body of LLM research.

3 Methodology

I began the analyses with importing the Pandas library, a powerful tool for data manipulation and analysis in Python. I loaded the dataset, stored as a CSV file, into a Pandas DataFrame for easier handling and manipulation of the structured data. This initial step helped verify that the data had been correctly loaded, providing an understanding of the dataset's structure, which included columns like "Taxonomy," "Title," "Authors," "Release Date," and "Summary." This analysis is illustrated in Figure 1. This preparation laid the foundation for the subsequent stages of data exploration and reporting. Later on, the exploration phase involved examining trends in survey paper publications over time, with a focus on the "Release Date" field, and analyzing the distribution of research across various taxonomy categories. To better understand the data, descriptive statistics were computed for key fields like "Release Year" and "Taxonomy." For the data manipulation phase, textual data from the "Title" and "Summary" fields were transformed into numerical representations using Term Frequency-Inverse Document Frequency (TF-IDF) vectorization, while the categorical "Categories" field was processed using one-hot encoding. These transformations were combined to build a feature matrix, which was then normalized to ensure that all features were on a consistent scale. In the final phase, a logistic regression model was implemented to predict the taxonomy category of each paper, and its performance was evaluated using metrics such as accuracy, precision, recall, and F1-score. To address class imbalance, class-weighting techniques were applied to the model, improving its ability to classify underrepresented categories. This approach provided a comprehensive understanding of the dataset and facilitated the development of a predictive model for classifying survey papers on Large Language Models (LLMs).

3.1 Data Exploration

3.1.1 Survey Paper Trends Over Time

To analyze trends in survey paper publications over time, I examined the "Release Date" column in the dataset to track how the volume of published papers changed.

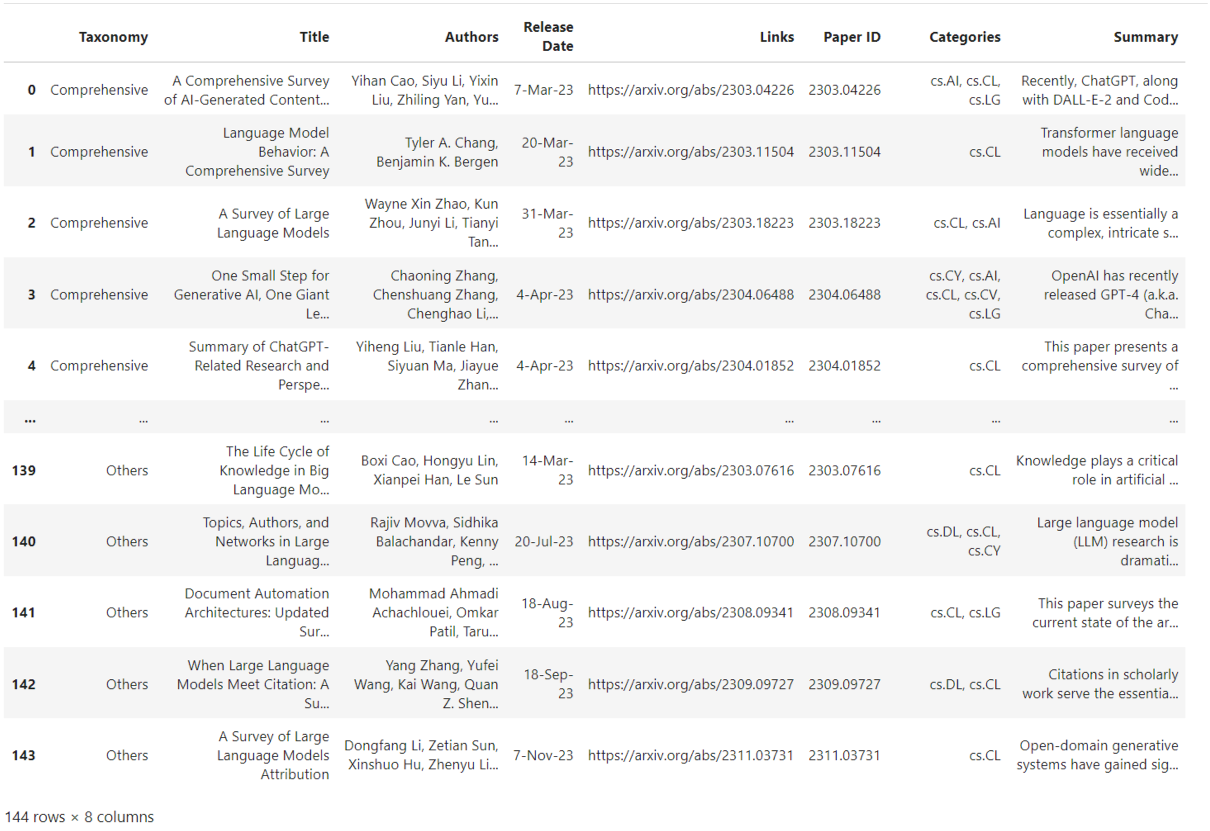
Methodology: The "Release Date" column was first converted into a datetime format using Pandas, making it easier to manipulate time-based data. From the converted dates, I extracted the "YearMonth" component, allowing the data to be grouped by month and the number of survey papers published each month to be counted. This grouping method was essential for visualizing trends in publication rates.

I used Matplotlib to generate a plot, where the X-axis represented the publication year and month, and the Y-axis displayed the number of survey papers published in that time frame. This analysis is illustrated in Figure 2.

Results: The plot revealed that very few survey papers were published in 2021, with a gradual increase starting in early 2022. A significant spike occurred in late 2023, peaking in November 2023, where nearly 18 survey papers were published. This sharp increase reflects the growing interest and research activity in LLMs, likely driven by advancements in AI technologies. The trend indicates a clear rise in publication numbers, showing a surge in research focus on LLMs, especially in the second half of 2023.

3.1.2 Taxonomy Distribution

I analyzed the distribution of survey papers across the various categories within the proposed taxon-



	Taxonomy	Title	Authors	Release Date	Links	Paper ID	Categories	Summary
0	Comprehensive	A Comprehensive Survey of AI-Generated Content...	Yihan Cao, Siyu Li, Yixin Liu, Zhiling Yan, Yu...	7-Mar-23	https://arxiv.org/abs/2303.04226	2303.04226	cs.AI, cs.CL, cs.LG	Recently, ChatGPT, along with DALL-E-2 and Cod...
1	Comprehensive	Language Model Behavior: A Comprehensive Survey	Tyler A. Chang, Benjamin K. Bergen	20-Mar-23	https://arxiv.org/abs/2303.11504	2303.11504	cs.CL	Transformer language models have received wide...
2	Comprehensive	A Survey of Large Language Models	Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tan...	31-Mar-23	https://arxiv.org/abs/2303.18223	2303.18223	cs.CL, cs.AI	Language is essentially a complex, intricate s...
3	Comprehensive	One Small Step for Generative AI, One Giant Le...	Chaoning Zhang, Chenshuang Zhang, Chenghao Li,...	4-Apr-23	https://arxiv.org/abs/2304.06488	2304.06488	cs.CY, cs.AI, cs.CL, cs.CV, cs.LG	OpenAI has recently released GPT-4 (a.k.a. Cha...
4	Comprehensive	Summary of ChatGPT-Related Research and Perspe...	Yiheng Liu, Tianle Han, Siyuan Ma, Jiayue Zhan...	4-Apr-23	https://arxiv.org/abs/2304.01852	2304.01852	cs.CL	This paper presents a comprehensive survey of ...
...	...	...	...	...	...	...	...	...
139	Others	The Life Cycle of Knowledge in Big Language Mo...	Boxi Cao, Hongyu Lin, Xianpei Han, Le Sun	14-Mar-23	https://arxiv.org/abs/2303.07616	2303.07616	cs.CL	Knowledge plays a critical role in artificial ...
140	Others	Topics, Authors, and Networks in Large Languag...	Rajiv Movva, Sidhika Balachandar, Kenny Peng, ...	20-Jul-23	https://arxiv.org/abs/2307.10700	2307.10700	cs.DL, cs.CL, cs.CY	Large language model (LLM) research is dramati...
141	Others	Document Automation Architectures: Updated Sur...	Mohammad Ahmadi Achachlouei, Omkar Patil, Taru...	18-Aug-23	https://arxiv.org/abs/2308.09341	2308.09341	cs.CL, cs.LG	This paper surveys the current state of the ar...
142	Others	When Large Language Models Meet Citation: A Su...	Yang Zhang, Yufei Wang, Kai Wang, Quan Z. Shen...	18-Sep-23	https://arxiv.org/abs/2309.09727	2309.09727	cs.DL, cs.CL	Citations in scholarly work serve the essentia...
143	Others	A Survey of Large Language Models Attribution	Dongfang Li, Zetian Sun, Xinshuo Hu, Zhenyu Li...	7-Nov-23	https://arxiv.org/abs/2311.03731	2311.03731	cs.CL	Open-domain generative systems have gained sig...

144 rows × 8 columns

Figure 1: Dataset in a tabular format

omy to identify which areas of research were dominant and which were underrepresented.

Methodology: To calculate the frequency of papers assigned to each category, I used the value_counts() function on the "Taxonomy" column. The resulting counts gave a clear overview of how many papers were categorized under each taxonomy label.

A bar chart was created using Matplotlib to visualize the distribution. The X-axis represented taxonomy categories, while the Y-axis showed the number of papers in each category. This analysis is illustrated in Figure 3.

Results: The bar chart revealed that the "Trustworthy" category was the most common, with 26 papers. Other frequently represented categories included "Comprehensive" and "Prompting," both showing a relatively high number of papers. However, categories such as "Law," "Finance," and "Education" were underrepresented, suggesting that fewer survey papers focused on these areas. The results highlighted the research focus on trustworthiness and prompting in LLMs while identifying potential gaps in other fields like finance and education, where fewer studies were published.

3.1.3 Descriptive Statistics on Release Years

To better understand the temporal distribution of survey papers, I computed descriptive statistics on the "Release Year" column. This analysis provided insights into the central tendencies and spread of the data.

Methodology: The "Release Date" column was first converted from string format into a datetime format using Pandas, and the "Release Year" was extracted for each paper. I computed several key statistical measures for the "Release Year" data:
Mean: The average year of publication, giving insight into the central tendency.
Median: The middle year of publication, which helps clarify the concentration of the data.
Variance and Standard Deviation: These metrics describe the spread of publication years around the mean.
Skewness: The asymmetry of the distribution, indicating whether papers were concentrated in earlier or more recent years.
Range and Interquartile Range (IQR): These values capture the earliest and latest years of publication, as well as the spread of the middle 50% of the data.
Quantiles: The 25th, 50th, and 75th percentiles

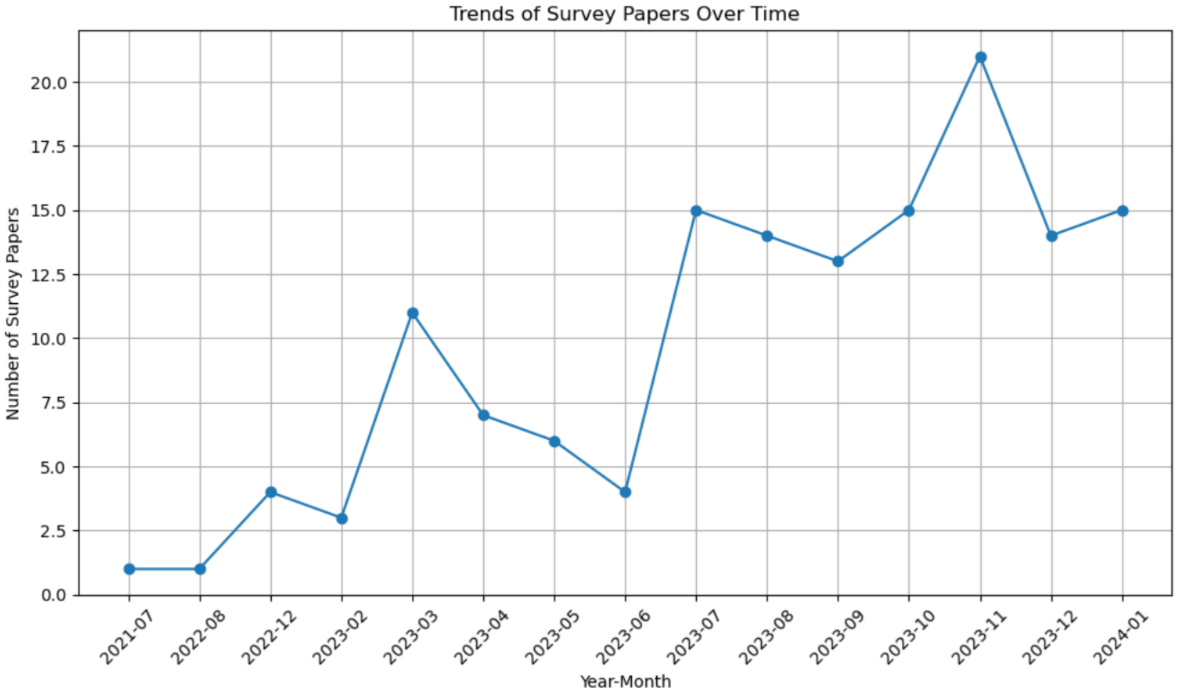


Figure 2: Trends in Survey Paper Publications Over Time

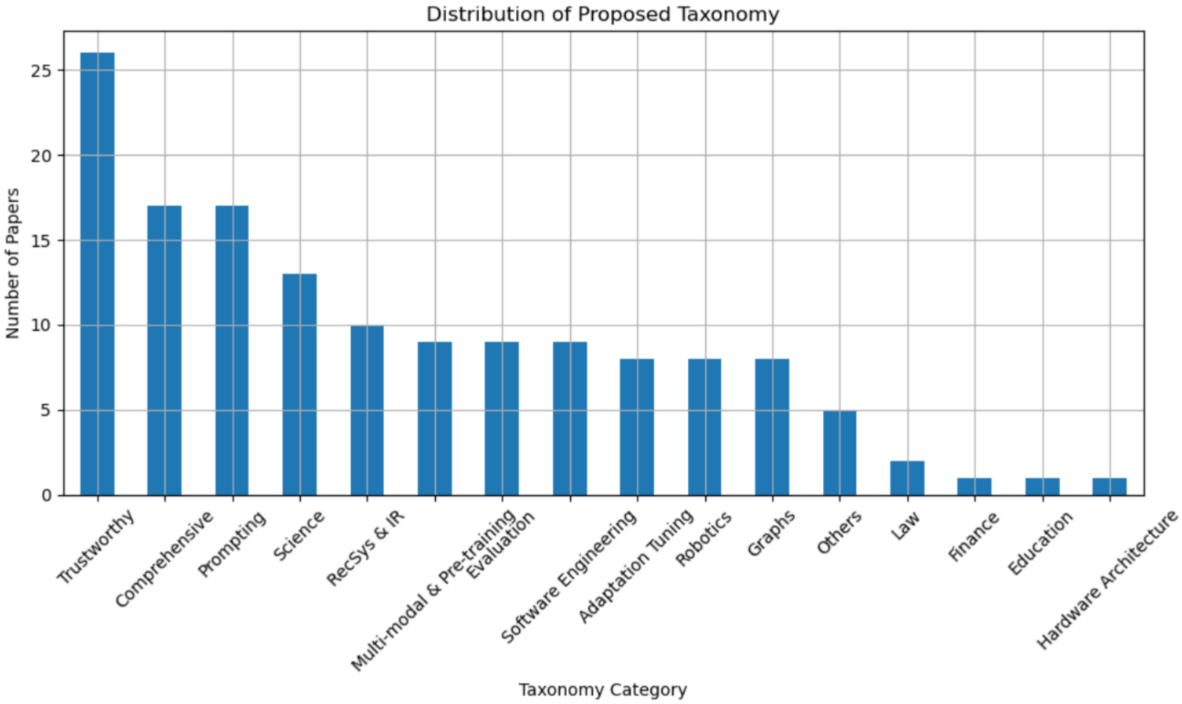
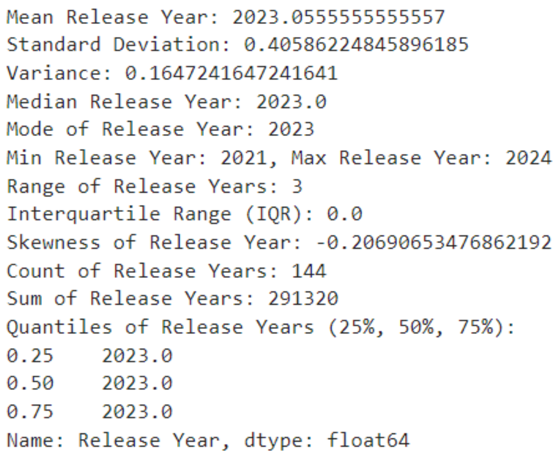


Figure 3: Distribution of Survey Papers by Taxonomy Category

were calculated to understand how the data is distributed.

Results: This analysis is illustrated in Figure 4. The computed descriptive statistics showed a mean release year of 2023, confirming that the majority of survey papers were published recently. The skewness value suggested a slight rightward skew, indicating that most publications were concentrated in the last few years, especially in 2023 and 2024. This analysis provided a clearer picture of the publication trends over time and helped interpret the temporal aspects of LLM-related research.



```
Mean Release Year: 2023.0555555555557
Standard Deviation: 0.40586224845896185
Variance: 0.1647241647241641
Median Release Year: 2023.0
Mode of Release Year: 2023
Min Release Year: 2021, Max Release Year: 2024
Range of Release Years: 3
Interquartile Range (IQR): 0.0
Skewness of Release Year: -0.20690653476862192
Count of Release Years: 144
Sum of Release Years: 291320
Quantiles of Release Years (25%, 50%, 75%):
0.25    2023.0
0.50    2023.0
0.75    2023.0
Name: Release Year, dtype: float64
```

Figure 4: Descriptive Statistics of Survey Paper Release Years

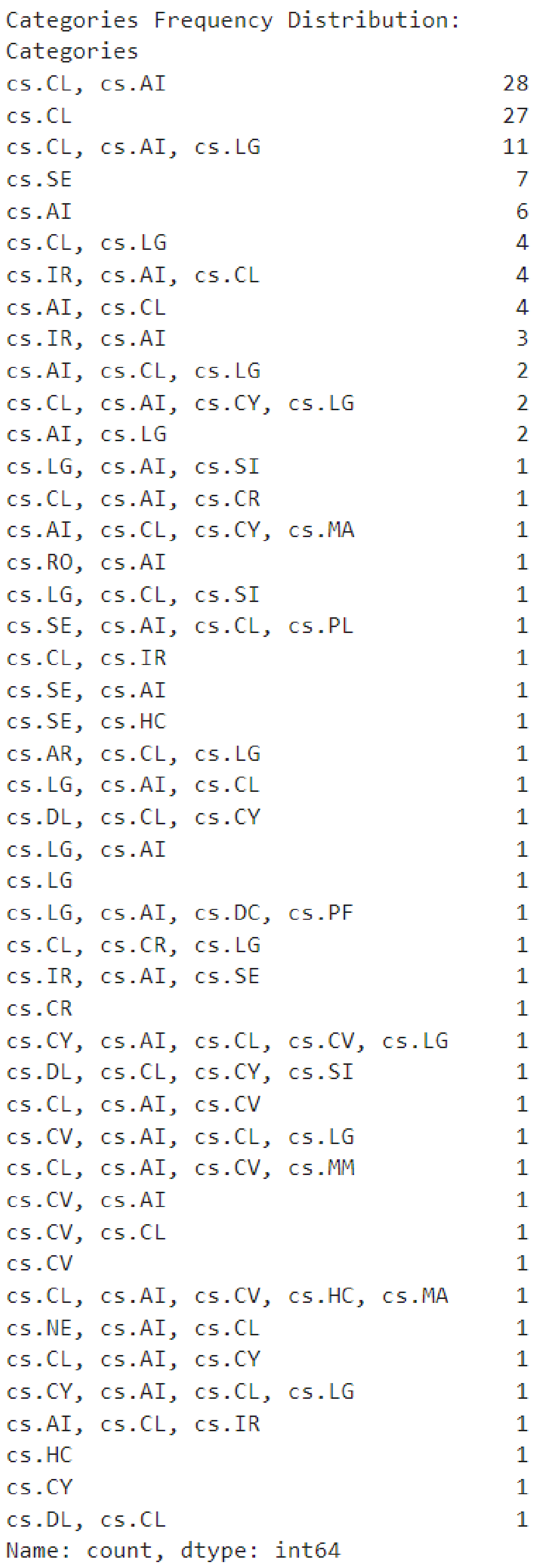
3.1.4 Visualizing Categorical and Temporal Data

In the final part of data exploration, I examined the frequency distribution of categories and the temporal distribution of the release years for the survey papers. This analysis is illustrated in Figure 6, 7 and 8.

Methodology: For the categorical data, I computed frequency counts for the "Categories" column to understand which categories were most represented. This count revealed that categories like "cs.CL" and "cs.AI" were highly represented, with 28 and 27 papers respectively.

To visualize the results, I created a bar chart to display the distribution of categories. The chart showed that only a few categories accounted for most of the papers, while several others were sparsely represented.

To analyze the temporal data, I plotted a histogram of release years and a cumulative distribution plot. The histogram depicted the frequency of papers published in each year, while the cumulative



```
Categories Frequency Distribution:
Categories
cs.CL, cs.AI                            28
cs.CL                                   27
cs.CL, cs.AI, cs.LG                     11
cs.SE                                    7
cs.AI                                    6
cs.CL, cs.LG                             4
cs.IR, cs.AI, cs.CL                      4
cs.AI, cs.CL                             4
cs.IR, cs.AI                             3
cs.AI, cs.CL, cs.LG                      2
cs.CL, cs.AI, cs.CY, cs.LG               2
cs.AI, cs.LG                             2
cs.LG, cs.AI, cs.SI                      1
cs.CL, cs.AI, cs.CR                      1
cs.AI, cs.CL, cs.CY, cs.MA               1
cs.RO, cs.AI                             1
cs.LG, cs.CL, cs.SI                      1
cs.SE, cs.AI, cs.CL, cs.PL               1
cs.CL, cs.IR                             1
cs.SE, cs.AI                             1
cs.SE, cs.HC                             1
cs.AR, cs.CL, cs.LG                      1
cs.LG, cs.AI, cs.CL                      1
cs.DL, cs.CL, cs.CY                      1
cs.LG, cs.AI                             1
cs.LG                                    1
cs.LG, cs.AI, cs.DC, cs.PF               1
cs.CL, cs.CR, cs.LG                      1
cs.IR, cs.AI, cs.SE                      1
cs.CR                                    1
cs.CY, cs.AI, cs.CL, cs.CV, cs.LG        1
cs.DL, cs.CL, cs.CY, cs.SI               1
cs.CL, cs.AI, cs.CV                      1
cs.CV, cs.AI, cs.CL, cs.LG               1
cs.CL, cs.AI, cs.CV, cs.MM               1
cs.CV, cs.AI                             1
cs.CV, cs.CL                             1
cs.CV                                    1
cs.CL, cs.AI, cs.CV, cs.HC, cs.MA        1
cs.NE, cs.AI, cs.CL                      1
cs.CL, cs.AI, cs.CY                      1
cs.CY, cs.AI, cs.CL, cs.LG               1
cs.AI, cs.CL, cs.IR                      1
cs.HC                                    1
cs.CY                                    1
cs.DL, cs.CL                             1
Name: count, dtype: int64
```

Figure 5: Categories Frequency Distribution

distribution plot provided an overall view of how the number of publications increased over time.

Results: The histogram and cumulative distribution plot confirmed that most survey papers were published in 2023, with few papers published in earlier years such as 2021 and 2022. The rapid increase in publications starting in 2023 underscored a significant rise in research efforts within the field. The combination of categorical and temporal visualizations gave valuable insights into both the areas of focus and the timing of research in LLM-related studies.

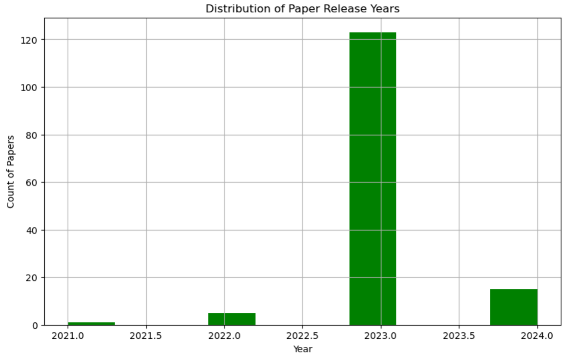


Figure 6: Cumulative Distribution of Survey Paper Publications Over Time

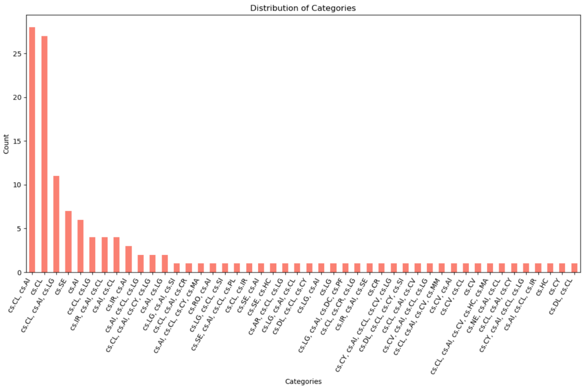


Figure 7: Bar Chart of Categories Frequency Distribution

3.2 Data Manipulation

3.2.1 Building a Feature Matrix

In this stage of the analysis, I created a feature matrix for the dataset by transforming categorical and text data into numerical representations that could be used for further analysis or modeling. This process involved the use of techniques such as Term Frequency-Inverse Document Frequency (TF-IDF) vectorization and one-hot encoding.

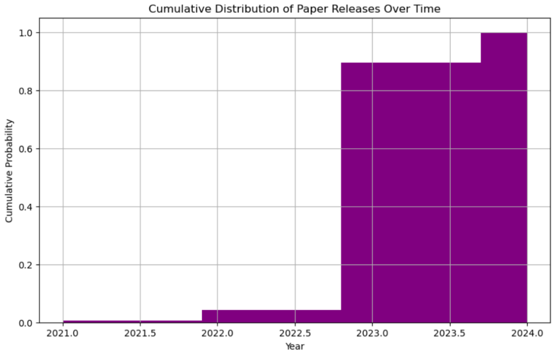


Figure 8: Cumulative Distribution of Paper Release Years Over Time

Methodology:

Loading the Dataset: I first loaded the dataset from the CSV file into a Pandas DataFrame for manipulation and analysis.

Building the Feature Matrix: To create a robust feature matrix, I applied the following transformations:

TF-IDF Vectorization of Text Data: The "Title" and "Summary" columns, which contain textual data, were transformed using the TF-IDF vectorizer. This method converts the textual data into numerical vectors that capture the importance of each word relative to the rest of the dataset. The TF-IDF scores were then stored in a dense matrix form for each paper title and summary.

One-Hot Encoding of Categorical Data: The "Categories" column, which contains categorical data, was processed using the one-hot encoding technique. This method transforms each unique category into a binary feature, where a value of "1" indicates the presence of a category for a particular paper and "0" indicates its absence. The resulting one-hot encoded matrix was then combined with the TF-IDF matrices.

Combining Features: Once the text and categorical data had been vectorized, I concatenated the resulting matrices into a single feature matrix. This final matrix included the TF-IDF representations of the "Title" and "Summary" columns as well as the one-hot encoded "Categories" column, creating a comprehensive numerical representation of the dataset.

Displaying the Feature Matrix: Finally, I displayed the combined feature matrix (figure 9), which had 3,748 columns representing all the features extracted from the textual and categorical data. The large number of columns reflects the complexity of the dataset, with each column corresponding to either a unique word in the text or a binary category label. This data manipulation step was crucial in preparing the dataset for further analysis or machine learning tasks by converting all text and categorical data into a format that can be processed by algorithms. The feature matrix served as the foundation for modeling and deeper exploration of the dataset's properties.

3.2.2 Normalization of Feature Matrix

In the subsequent step of data manipulation, I normalized the feature matrix to ensure all features were on the same scale and prepared for further machine learning algorithms. Normalization is essential because many machine learning models are sensitive to the magnitude of different features, and this process ensures that no feature dominates the others due to its scale.

Methodology:

Converting Boolean Columns: Initially, I converted the boolean columns within the feature matrix into numeric values (1s and 0s). This was achieved by applying the `pd.to_numeric()` function to ensure that all columns in the matrix were numeric and ready for scaling. Any errors during conversion were handled by filling in missing values with zeros, ensuring a clean dataset.

Normalization with MinMaxScaler: To normalize the numeric data, I applied the MinMaxScaler from Scikit-learn, which scales all features to a range between 0 and 1. The MinMaxScaler was applied only to the numeric values of the DataFrame to preserve the feature matrix's structure. This scaling ensures that features are comparable, regardless of their original scale.

Reconstructing the DataFrame: After normalization, the scaled values were converted back into a Pandas DataFrame, preserving the column names and structure of the original feature matrix. This made it easier to interpret the scaled values and ensured compatibility with future operations.

Displaying the Normalized Matrix: The resulting normalized matrix was displayed (figure 10), showing that all values across the 3,748 columns (features) were now scaled between 0 and 1. This step confirmed that the normalization process had successfully standardized the dataset, ensuring that each feature contributed equally to any further analysis or modeling.

This normalization step was crucial for ensuring that no individual feature would disproportionately affect the results of machine learning models. By scaling all features to the same range, the data was made ready for a variety of machine learning techniques, which often assume normalized input features for optimal performance.

3.2.3 Encoding Labels Using LabelEncoder

In the next stage of data manipulation, I used the LabelEncoder from Scikit-learn to transform the categorical "Taxonomy" labels into numerical values. This transformation is essential for converting non-numeric data into a format that can be fed into machine learning algorithms, which typically require numeric input. This analysis is illustrated in Figure 11.

Methodology:

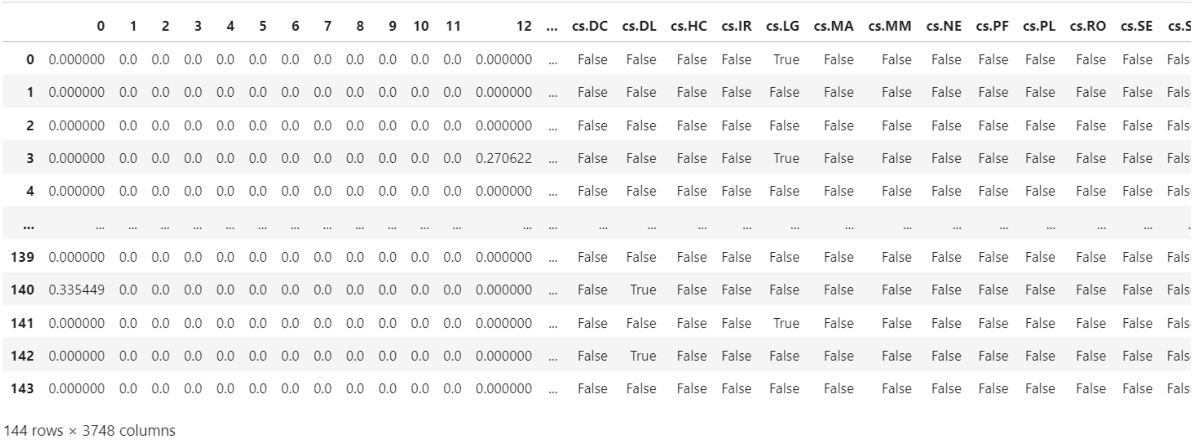
Initializing the LabelEncoder: I first initialized the LabelEncoder to convert the categorical labels in the "Taxonomy" column. The "Taxonomy" column contained the different categories to which each paper was assigned, and I assigned this column to the variable `y`.

Fitting and Transforming Labels: The LabelEncoder was then used to fit and transform the taxonomy labels. This process encoded each unique taxonomy category as an integer. For example, the label "Comprehensive" was encoded as "1," and "Trustworthy" was encoded as "7." This transformation allowed me to represent the categorical labels numerically, making the dataset more suitable for machine learning models.

Checking Encoded Labels: After the transformation, I printed both the encoded labels and the original labels to verify the correctness of the encoding process. This step ensured that the mapping between the original taxonomy categories and the corresponding integers was accurate.

Inverse Transforming Labels: To validate the encoding, I also used the `inverse_transform` method of the LabelEncoder to map the encoded integers back to their original categorical labels. This allowed me to confirm that the encoded labels could be correctly reverted to their original taxonomy categories if necessary.

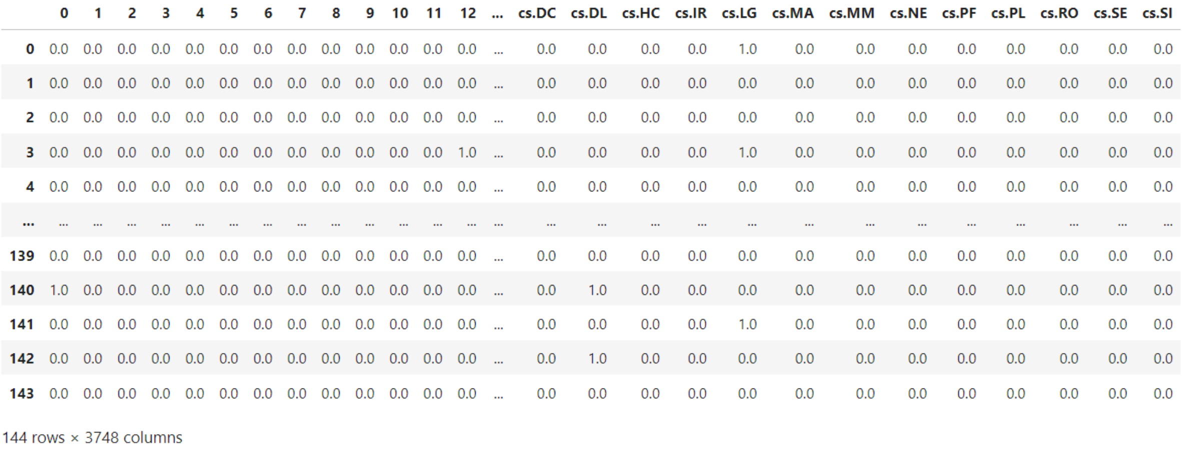
This encoding step was crucial for transforming the categorical taxonomy data into a numerical format, which is required for many machine learning algorithms. By using LabelEncoder, I was able to retain the essential categorical information while making the data more computationally manageable for future analysis or modeling tasks.



	0	1	2	3	4	5	6	7	8	9	10	11	12	...	cs.DC	cs.DL	cs.HC	cs.IR	cs.LG	cs.MA	cs.MM	cs.NE	cs.PF	cs.PL	cs.RO	cs.SE	cs.S
0	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.000000	...	False	False	False	False	True	False	False	False	False	False	False	False	Fals
1	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.000000	...	False	False	False	False	False	False	False	False	False	False	False	False	Fals
2	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.000000	...	False	False	False	False	False	False	False	False	False	False	False	False	Fals
3	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.270622	...	False	False	False	False	True	False	False	False	False	False	False	False	Fals
4	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.000000	...	False	False	False	False	False	False	False	False	False	False	False	False	Fals
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	.
139	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.000000	...	False	False	False	False	False	False	False	False	False	False	False	False	Fals
140	0.335449	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.000000	...	False	True	False	False	False	False	False	False	False	False	False	False	Fals
141	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.000000	...	False	False	False	False	True	False	False	False	False	False	False	False	Fals
142	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.000000	...	False	True	False	False	False	False	False	False	False	False	False	False	Fals
143	0.000000	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.000000	...	False	False	False	False	False	False	False	False	False	False	False	False	Fals

144 rows × 3748 columns

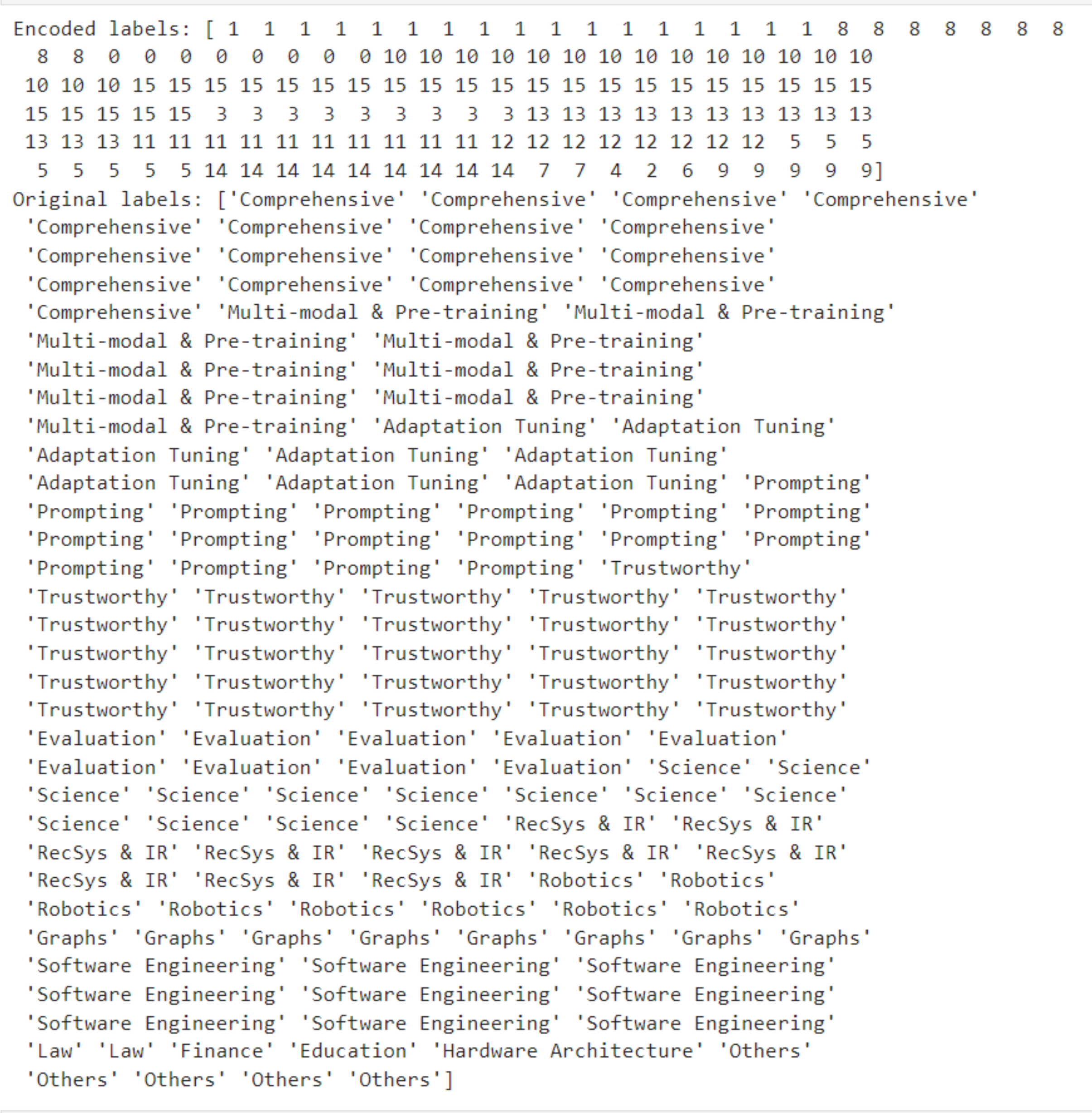
Figure 9: Featured Matrix



	0	1	2	3	4	5	6	7	8	9	10	11	12	...	cs.DC	cs.DL	cs.HC	cs.IR	cs.LG	cs.MA	cs.MM	cs.NE	cs.PF	cs.PL	cs.RO	cs.SE	cs.SI
0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	...	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
139	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
140	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
141	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
142	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
143	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

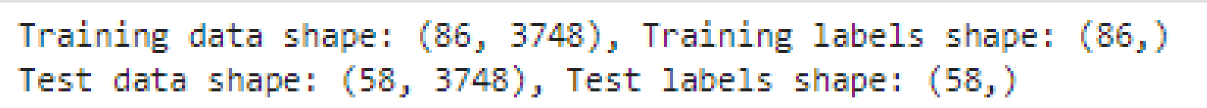
144 rows × 3748 columns

Figure 10: Normalized matrix



```
Encoded labels: [ 1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  1  8  8  8  8  8  8  8
  8  8  0  0  0  0  0  0  0  0 10 10 10 10 10 10 10 10 10 10 10 10 10 10
 10 10 10 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15 15
 15 15 15 15 15  3  3  3  3  3  3  3  3  3 13 13 13 13 13 13 13 13 13 13
 13 13 13 11 11 11 11 11 11 11 11 11 11 12 12 12 12 12 12 12 12  5  5  5
  5  5  5  5  5 14 14 14 14 14 14 14 14 14  7  7  4  2  6  9  9  9  9  9]
Original labels: ['Comprehensive' 'Comprehensive' 'Comprehensive' 'Comprehensive'
 'Comprehensive' 'Comprehensive' 'Comprehensive' 'Comprehensive'
 'Comprehensive' 'Comprehensive' 'Comprehensive' 'Comprehensive'
 'Comprehensive' 'Comprehensive' 'Comprehensive' 'Comprehensive'
 'Comprehensive' 'Multi-modal & Pre-training' 'Multi-modal & Pre-training'
 'Multi-modal & Pre-training' 'Multi-modal & Pre-training'
 'Multi-modal & Pre-training' 'Multi-modal & Pre-training'
 'Multi-modal & Pre-training' 'Multi-modal & Pre-training'
 'Multi-modal & Pre-training' 'Adaptation Tuning' 'Adaptation Tuning'
 'Adaptation Tuning' 'Adaptation Tuning' 'Adaptation Tuning'
 'Adaptation Tuning' 'Adaptation Tuning' 'Adaptation Tuning' 'Prompting'
 'Prompting' 'Prompting' 'Prompting' 'Prompting' 'Prompting' 'Prompting'
 'Prompting' 'Prompting' 'Prompting' 'Prompting' 'Prompting' 'Prompting'
 'Prompting' 'Prompting' 'Prompting' 'Prompting' 'Trustworthy'
 'Trustworthy' 'Trustworthy' 'Trustworthy' 'Trustworthy' 'Trustworthy'
 'Trustworthy' 'Trustworthy' 'Trustworthy' 'Trustworthy' 'Trustworthy'
 'Trustworthy' 'Trustworthy' 'Trustworthy' 'Trustworthy' 'Trustworthy'
 'Trustworthy' 'Trustworthy' 'Trustworthy' 'Trustworthy' 'Trustworthy'
 'Trustworthy' 'Trustworthy' 'Trustworthy' 'Trustworthy' 'Trustworthy'
 'Evaluation' 'Evaluation' 'Evaluation' 'Evaluation' 'Evaluation'
 'Evaluation' 'Evaluation' 'Evaluation' 'Evaluation' 'Science' 'Science'
 'Science' 'Science' 'Science' 'Science' 'Science' 'Science' 'Science'
 'Science' 'Science' 'Science' 'Science' 'RecSys & IR' 'RecSys & IR'
 'RecSys & IR' 'RecSys & IR' 'RecSys & IR' 'RecSys & IR' 'RecSys & IR'
 'RecSys & IR' 'RecSys & IR' 'RecSys & IR' 'Robotics' 'Robotics'
 'Robotics' 'Robotics' 'Robotics' 'Robotics' 'Robotics' 'Robotics'
 'Graphs' 'Graphs' 'Graphs' 'Graphs' 'Graphs' 'Graphs' 'Graphs' 'Graphs'
 'Software Engineering' 'Software Engineering' 'Software Engineering'
 'Software Engineering' 'Software Engineering' 'Software Engineering'
 'Software Engineering' 'Software Engineering' 'Software Engineering'
 'Law' 'Law' 'Finance' 'Education' 'Hardware Architecture' 'Others'
 'Others' 'Others' 'Others' 'Others']
```

Figure 11: Visualization of Encoded and Original Taxonomy Labels for Survey Papers



```
Training data shape: (86, 3748), Training labels shape: (86,)
Test data shape: (58, 3748), Test labels shape: (58,)
```

Figure 12: Dataset Overview: Training and Testing Data Shapes

3.2.4 Splitting the Data into Training and Testing Sets

In this step, I prepared the dataset for model training by splitting it into training and testing sets. This process is crucial for evaluating how well a machine learning model generalizes to unseen data by using one portion of the data for training and another for testing. I used the `train_test_split` function from Scikit-learn to carry out this operation.This analysis is illustrated in Figure 12.

Methodology:

Defining the Test Ratio: I set the test ratio to 0.4, meaning that 40% of the data would be reserved for testing, and the remaining 60% would be used for training. This split ratio ensures that a significant portion of the data is set aside to evaluate the model's performance on unseen data.

Splitting the Data: The `train_test_split` function was used to split the normalized feature matrix (`X`) and the encoded labels (`y_encoded`). The function randomly split the data according to the specified ratio while maintaining the structure and shape of the data. A random seed (`random_state=42`) was set to ensure reproducibility of the split.

- `X_train` and `y_train`: These represent the training set features and labels. - `X_test` and `y_test`: These represent the testing set features and labels.

Verifying the Split: After splitting the data, I checked the shapes of both the training and testing sets to ensure that the split was done correctly. The training data had 86 samples, while the testing data had 58 samples, consistent with the 60/40 split.

- **Training Data Shape**: (86, 3748), meaning 86 samples with 3,748 features each. - **Testing Data Shape**: (58, 3748), meaning 58 samples with 3,748 features each.

This step was essential for preparing the data for machine learning, ensuring that the model would be trained on one portion of the data and validated on another to avoid overfitting and assess its generalization performance.

3.3 Data Evaluation

3.3.1 Logistic Regression Model

To evaluate the model's performance, a logistic regression algorithm was implemented to predict the taxonomy category for each paper in the dataset. Logistic regression is suitable for multi-class classification problems and provides insight into the relationships between features and labels. This analysis is illustrated in Figure 13.

Methodology:

- *Converting Column Names:* The column names in the feature matrix were converted to strings to ensure compatibility with Scikit-learn.

- *Model Initialization:* The logistic regression model was initialized using Scikit-learn's `LogisticRegression` function. The max parameter was set to 1000 to allow sufficient iterations for model convergence.

- *Training the Model:* The model was trained on the training dataset using the `fit()` method to establish patterns between features and taxonomy labels.

- *Making Predictions:* After training, predictions were made on the test dataset using the `predict()` method, aiming to classify each paper into its respective taxonomy category.

- *Evaluating Performance:* Model performance was evaluated using the `accuracy_score` function, with the overall accuracy measured at 43%. A classification report was generated, providing precision, recall, and F1-score for each category.

- *Addressing Class Imbalance:* The label distribution in both the training and testing datasets was reviewed to assess class imbalance, which can negatively impact model performance, especially for underrepresented categories.

The model's performance highlighted areas for improvement, particularly regarding class imbalance and the need for more complex models to enhance predictive accuracy.

3.3.2 Confusion Matrix and Model Visualization

To further assess the logistic regression model, a confusion matrix was generated. This matrix provided detailed insights into the number of correct and incorrect classifications for each taxonomy category. This is illustrated in Figure 14 and Figure 15.

Methodology:

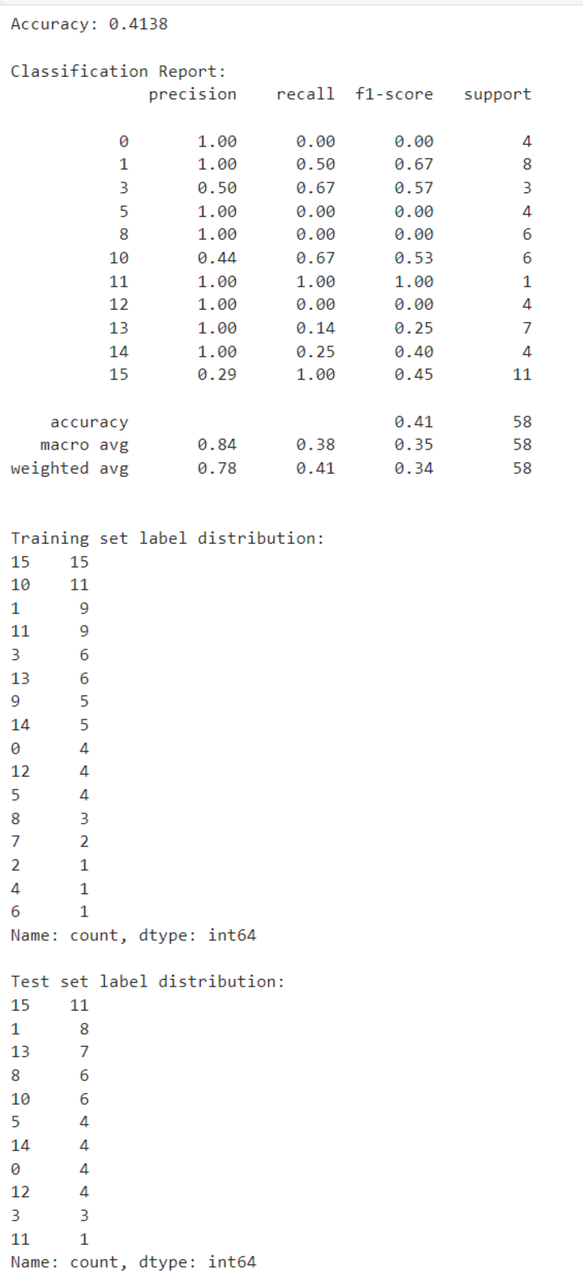
- *Classification Report:* A classification report was created to summarize precision, recall, and F1-scores for each class, offering a more comprehensive view of the model's performance.

- *Confusion Matrix:* A confusion matrix was computed using the confusion matrix function to compare true labels with predicted labels across all classes.

- *Confusion Matrix Visualization:* The confusion matrix was visualized as a heatmap using the Seaborn library. The darker colors along the diagonal of the heatmap indicated correctly predicted classes, while off-diagonal elements reflected misclassifications.

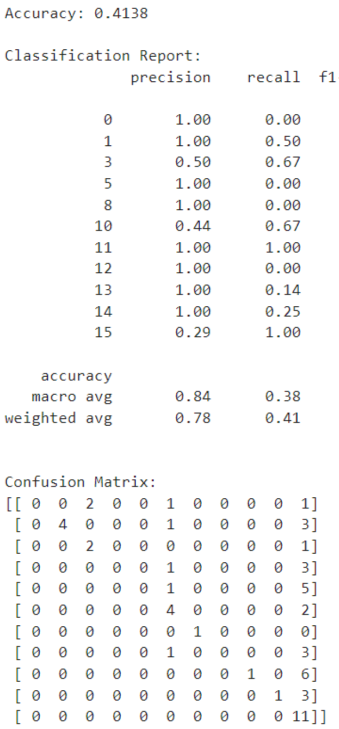
- *Model Accuracy:* The overall model accuracy remained consistent at 43

  The confusion matrix and visual representation helped identify which classes were misclassified more frequently, highlighting potential areas where the model could be improved.



```
Accuracy: 0.4138

Classification Report:
              precision    recall  f1-score   support

           0       1.00      0.00      0.00         4
           1       1.00      0.50      0.67         8
           3       0.50      0.67      0.57         3
           5       1.00      0.00      0.00         4
           8       1.00      0.00      0.00         6
          10       0.44      0.67      0.53         6
          11       1.00      1.00      1.00         1
          12       1.00      0.00      0.00         4
          13       1.00      0.14      0.25         7
          14       1.00      0.25      0.40         4
          15       0.29      1.00      0.45        11

    accuracy                           0.41        58
   macro avg       0.84      0.38      0.35        58
weighted avg       0.78      0.41      0.34        58


Training set label distribution:
15    15
10    11
1      9
11     9
3      6
13     6
9      5
14     5
0      4
12     4
5      4
8      3
7      2
2      1
4      1
6      1
Name: count, dtype: int64

Test set label distribution:
15    11
1      8
13     7
8      6
10     6
5      4
14     4
0      4
12     4
3      3
11     1
Name: count, dtype: int64
```

Figure 13: Classification Report and Label Distribution



```
Accuracy: 0.4138

Classification Report:
              precision    recall  f1-score   support

           0       1.00      0.00      0.00         4
           1       1.00      0.50      0.67         8
           3       0.50      0.67      0.57         3
           5       1.00      0.00      0.00         4
           8       1.00      0.00      0.00         6
          10       0.44      0.67      0.53         6
          11       1.00      1.00      1.00         1
          12       1.00      0.00      0.00         4
          13       1.00      0.14      0.25         7
          14       1.00      0.25      0.40         4
          15       0.29      1.00      0.45        11

    accuracy                           0.41        58
   macro avg       0.84      0.38      0.35        58
weighted avg       0.78      0.41      0.34        58


Confusion Matrix:
[[ 0  0  2  0  0  1  0  0  0  0  1]
 [ 0  4  0  0  0  1  0  0  0  0  3]
 [ 0  0  2  0  0  0  0  0  0  0  1]
 [ 0  0  0  0  0  1  0  0  0  0  3]
 [ 0  0  0  0  0  1  0  0  0  0  5]
 [ 0  0  0  0  0  4  0  0  0  0  2]
 [ 0  0  0  0  0  0  1  0  0  0  0]
 [ 0  0  0  0  0  1  0  0  0  0  3]
 [ 0  0  0  0  0  0  0  0  1  0  6]
 [ 0  0  0  0  0  0  0  0  0  1  3]
 [ 0  0  0  0  0  0  0  0  0  0 11]]
```

Figure 14: Classification Report with Confusion Matrix

3.3.3 Bonus Question: Logistic Regression with Class Weighting

Given the class imbalance in the dataset, class weighting was applied to the logistic regres-

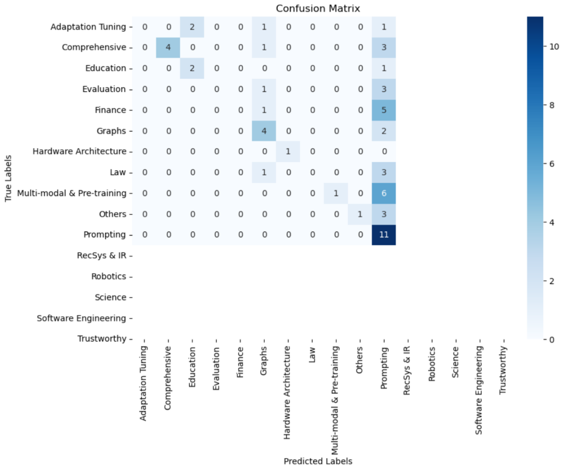


Figure 15: Confusion Matrix Heatmap

sion model to improve performance, particularly for underrepresented categories. This is illustrated in Figure 16.

Methodology:

- *Installing Imbalanced-Learn:* The `imbalanced-learn` library was installed to manage class imbalances during training.
- *Initializing Logistic Regression with Class Weighting:* The logistic regression model was re-initialized with the `class_weight='balanced'` parameter, ensuring that classes were weighted inversely proportional to their frequency in the dataset.
- *Training the Model:* The model was retrained on the dataset with the adjusted class weights to address the imbalance and improve classification of underrepresented categories.
- *Making Predictions and Evaluating Performance:* Predictions were made on the test dataset, and the accuracy improved to 47%. Another classification report was generated to evaluate improvements in the precision, recall, and F1-score for underrepresented classes.

Results: The adjusted logistic regression model showed a modest improvement in overall accuracy, from 43% to 47%, and improved performance on underrepresented classes, although certain categories still exhibited lower performance.

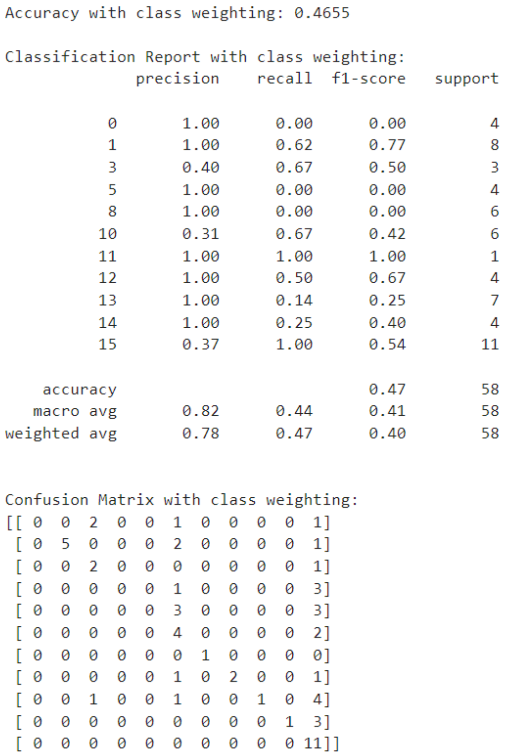

```
Accuracy with class weighting: 0.4655

Classification Report with class weighting:
              precision    recall  f1-score   support

           0       1.00      0.00      0.00         4
           1       1.00      0.62      0.77         8
           3       0.40      0.67      0.50         3
           5       1.00      0.00      0.00         4
           8       1.00      0.00      0.00         6
          10       0.31      0.67      0.42         6
          11       1.00      1.00      1.00         1
          12       1.00      0.50      0.67         4
          13       1.00      0.14      0.25         7
          14       1.00      0.25      0.40         4
          15       0.37      1.00      0.54        11

    accuracy                           0.47        58
   macro avg       0.82      0.44      0.41        58
weighted avg       0.78      0.47      0.40        58

Confusion Matrix with class weighting:
[[ 0  0  2  0  0  1  0  0  0  0  1]
 [ 0  5  0  0  0  2  0  0  0  0  1]
 [ 0  0  2  0  0  0  0  0  0  0  1]
 [ 0  0  0  0  0  1  0  0  0  0  3]
 [ 0  0  0  0  0  3  0  0  0  0  3]
 [ 0  0  0  0  0  4  0  0  0  0  2]
 [ 0  0  0  0  0  0  1  0  0  0  0]
 [ 0  0  0  0  0  1  0  2  0  0  1]
 [ 0  0  1  0  0  1  0  0  1  0  4]
 [ 0  0  0  0  0  0  0  0  0  1  3]
 [ 0  0  0  0  0  0  0  0  0  0 11]]
```

Figure 16: Class weighing

4 Conclusion

In this report, I analyzed survey papers on Large Language Models (LLMs) by exploring metadata, categorizing research areas, and using machine learning to predict taxonomy categories. The analysis highlighted publication trends, with significant research growth in 2023, particularly in "Trustworthy" and "Prompting" areas.

I implemented a logistic regression model, initially achieving 43% accuracy, which improved to 47% after applying class-weighting techniques to address class imbalance. Although this improved classification for underrepresented categories, further refinement is needed to enhance performance in some areas.

This study provides a foundation for understanding LLM survey papers and offers a machine learning-based approach to help researchers identify key trends and gaps in the field. Future work could involve more advanced models or additional metadata to improve classification and insight into LLM research.

A APPENDIX

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, et al. Constitutional ai: Harmlessness from ai feedback. *arXiv preprint arXiv:2212.08073*, 2022.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Alexey Dosovitskiy. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016.

Alec Radford, Karthik Narasimhan, Tim Salimans, Ilya Sutskever, et al. Improving language understanding by generative pre-training. *OpenAI blog*, 2018.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

Gemini Team, Rohan Anil, Sebastian Borgeaud, Yonghui Wu, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, et al. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*, 2023.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Jun Zhuang and Mohammad Al Hasan. Defending graph convolutional networks against dynamic graph perturbations via bayesian self-supervision. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 4405–4413, 2022.

Jun Zhuang and Casey Kennington. Understanding survey paper taxonomy about large language models via graph representation learning. *arXiv preprint arXiv:2402.10409*, 2024.