Taxonomy Classification using Machine Learning Based Model

Anup Majumder
Department of Computer Science
Boise State University
anupmajumder@u.boisestate.edu

Abstract

Large language model (LLM) trends and taxonomy have changed rapidly in the last few years, primarily due to the advancement of data sciences like natural language processing (NLP), deep learning, and the ever-growing size of computational resources. These models aim to enhance logical and mathematical reasoning beyond pattern recognition. This work aims to explore trends in survey papers over time and analyze their associated taxonomies through data exploration, visualization, and machine learning modeling. Initially, the dataset of survey papers is preprocessed by grouping the number of surveys by year and month, revealing publication trends across time. A detailed analysis of taxonomy distributions is performed to identify the prevalence of various survey categories. Using the TF-IDF method, the titles and summaries of papers are vectorized, transforming textual information into numerical features. A one-hot encoding approach is applied to the survey categories to enable better feature representation for machine learning models. The results show that the Random Forest Classifier and Support Vector Machine achieved the highest accuracies in classifying survey papers based on their taxonomy. This research not only highlights trends in the publication of surveys but also offers an automated approach for classifying them, potentially aiding future research in organizing and categorizing survey literature efficiently.

Keywords: *Data Exploration*, *Data Visualization*, *Classifiers*, *LLM Survey*, *TF-IDF*

1 Introduction

The advent of large-scale digital datasets has dramatically reshaped the landscape of data-driven research, particularly in fields such as computer science, social science, and healthcare. Data-driven methods have become integral to uncovering meaningful insights, patterns, and trends from vast amounts of unstructured information. This paper presents an analysis of survey data using various machine-learning techniques and provides a framework for understanding trends in research surveys. Specifically, this analysis focuses on extracting valuable insights from survey datasets, including trend analysis, taxonomy distribution, and classification of papers using machine learning algorithms. The dataset used in this study includes survey papers characterized by fields such as the release date, paper title, taxonomy categories, and textual summaries. This analysis's main goals are to identify temporal trends, categorize papers according to their taxonomy, and evaluate how well different machine learning models perform in the classification of survey papers. We implement a multi-step data analysis pipeline that includes data exploration, visualization, and machine learning-based classification.

The first phase of this analysis focuses on exploring the dataset to identify trends over time. Using Python's pandas library, we preprocess the dataset by extracting temporal features such as the year and month from the release dates of the survey papers. We then group the dataset by year and month to observe the frequency of surveys released over time. The ggplot-style visualizations help to identify any cyclic patterns, growth, or declines in survey releases. These visualizations serve as the foundation for identifying how interest in particular research areas evolve over time, providing researchers with a clear understanding of temporal trends.

In the second phase, we analyze the distribution of taxonomy categories proposed in the dataset. The taxonomy categories reflect the thematic areas of the survey papers. We use bar charts to visualize the distribution of these categories, allowing us to identify which taxonomies are dominant and which are underrepresented. For example, the "Trustworthy" category comprises 26 papers, representing a significant portion of the dataset. Next, we employ machine learning techniques to classify papers based on their titles, summaries, and taxonomy categories. We vectorize the textual aspects of the dataset, including the article titles and summaries, by using the TF-IDF (Term Frequency-Inverse Document Frequency) technique for the creation of a feature matrix. Additionally, we use one-hot encoding for categorical fields, such as the taxonomy categories. This feature engineering step allows us to build a comprehensive feature matrix representing the dataset in a numerical format that is suitable for machine learning models.

Lastly, we use a several machine learning classifiers, like Naive Bayes, Random Forests Classifier, Decision Tree, then Support Vector Machines (SVM), and Logistic Regression, to categorize the survey papers according to their taxonomy. Using common classification metrics, including accuracy, recall, precision, and confusion matrices, we evaluate the performance of each model. Our results demonstrate varying degrees of effectiveness across these models, with Random Forests and SVM achieving higher classification accuracy compared to others. This analysis contributes to the field of survey paper categorization and data-driven research by offering a comprehensive methodology for extracting trends and insights from research datasets. The findings have the potential to guide future research efforts, highlighting emerging fields and presenting an accurate picture of the changing field of research interests.

2 Related Work

A thorough overview of the integration of the Large Language Models (LLMs) with graph learning techniques is provided on LLMs for graphs X. Ren. et al. [1]. It categorizes existing methods into four unique designs, highlighting their strengths and limitations. The authors discuss the impressive performance of LLMs in natural language processing and their potential to enhance graph-centric tasks, such as link prediction and node classification. However, challenges like data sparsity, scalability, and limited interaction between agents and graph data persist. The survey aims to inspire future research and innovation in leveraging LLMs for more effective graph-related applications. X. Wang et al. [2] describes a thorough rundown of how Large Language Models (LLMs) and Knowledge Representation Learning (KRL) are integrated, yet it lacks depth in critical analysis. While it lists various models and datasets, it does not sufficiently address the limitations and challenges associated with these approaches, such as computational demands and data requirements. Additionally, the evaluation metrics discussed, while important, could benefit from a more nuanced discussion on their applicability across different contexts.

The integration of Graph Representation Learning (GRL) with Large Language Models (LLMs) is investigated in this survey Q Mao et al. [3], enhancing contextual understanding. It categorizes components and techniques, discusses training strategies, and identifies future research directions, particularly in adapting LLMs to non-textual graph data and improving knowledge transfer across diverse graph structures. There remains a significant gap in systematically understanding how to effectively design and train models that leverage LLMs for non-textual graph data interpretation and knowledge transfer across diverse graph structures. R. Liu et al. [4] investigates the performance of Graph Neural Networks (GNNs) across various benchmark datasets, emphasizing the importance of both graph structures and node features. It reveals that excluding node features significantly impacts GNN prediction performance, particularly in transductive tasks. The study highlights the necessity for comprehensive benchmarking frameworks, as synthetic datasets may not accurately reflect real-world complexities. Additionally, it discusses the potential of new datasets, like DeezerEurope, and suggests that understanding dataset characteristics can guide advancements in self-supervised learning and data augmentation for GNNs. M. A. K. Raiaan et al. [5] explores the architecture, evolution, and transformer models of LLMs like GPT and BERT, along with their practical applications in various sectors like healthcare, education, and business. The paper also discusses challenges related to LLM deployment, such as security, ethical concerns, privacy, and future research directions. The paper highlights key LLMs, such as GPT, BERT, PaLM, and LaMDA, explaining their architectures, training methods, and diverse applications.

Y. Li et al. [6] presents a novel approach for learning effective representations from text-attributed graphs (TAGs) using a combination of the pre-trained language models (PLMs) and graph neural networks (GNNs). Existing methods for the self-supervised learning on TAGs often struggle to fully capture structural context or rely on labeled data. Grenade introduces two self-supervised learning techniques, such as Graph-Centric Knowledge Alignment and Graph-Centric Contrastive Learning, to address this. These techniques help capture both the textual semantics and structural relationships in the graph, improving performance in various downstream tasks such as node classification, link prediction and clustering. Extensive trials show that Grenade produces more generalizable representations than state-of-the-art techniques. A model centered on the integration of graph-based tasks with Large Language Models (LLMs) was proposed by J. Li et al. [7]. Although LLMs have shown success in natural language processing, their full potential when applied to graph data remains underexplored. To be more precise, difficulties include the fact that creating graph prompts is more difficult than creating language prompts, that matching downstream graph tasks to pre-training activities is more complicated, and that theoretical study is required to fully comprehend the function of graph prompts. Additionally, there is a lack of clear evaluation criteria for assessing the effectiveness of these prompts. T. Sen et al. [8] presents a novel approach to enhancing pre-trained language models (LMs) by integrating structured knowledge from knowledge graphs (KGs). The authors propose two self-supervised learning tasks to enrich LMs: an entity-level masking scheme that uses KGs to select and mask key entities and a distractor-suppressed ranking task to improve learning by using negative entity samples from KGs. This method incorporates structured knowledge implicitly during pre-training, making it more efficient and generalizable compared to existing approaches that rely on explicit KG integration during fine-tuning. Tests reveal that the model outperforms the benchmarks in a number of domains, such as knowledge base completion and question answering.

D. C. Zhang et al. [9] presents an innovative method to integrate knowledge graphs with pre-trained language models, but it also has notable limitations. One criticism is that while the model claims efficiency and generalizability, its reliance on knowledge graphs during pre-training could limit applicability to domains lacking rich, well-structured knowledge graphs. Additionally, the proposed approach's focus on entity-level masking and distractor-based ranking might oversimplify complex relational information, potentially overlooking subtler semantic nuances. Moreover, the performance improvements, though demonstrated on a few benchmarks, may not generalize well across a broader range of tasks, particularly in real-world settings where knowledge graph quality varies widely. Finally, the method's computational requirements for pre-training may hinder its scalability for larger and more diverse corpora. W. Ju et al. [10] introduces a method to integrate knowledge graphs with the pre-trained language models (PLMs) through two novel self-supervised tasks: entity-level masking and distractor-suppressed ranking. This aims to improve models' ability to handle structured knowledge without needing explicit KG usage during fine-tuning, showing improved results in question answering and knowledge base tasks. However, the paper's focus on pre-training with knowledge graphs might limit its generalizability to domains where structured knowledge is scarce or less organized. Additionally, while the proposed tasks enhance entity representation, they might not fully capture more complex relationships or semantic nuances present in natural language, especially in less structured or informal text settings. Future work could explore incorporating more dynamic and heterogeneous data sources beyond knowledge graphs to address these limitations.

3 Methodology

In this research work dataset has been loaded using the panda's library. The dataset contains several information's on various types of surveys such as release dates, titles, summaries, and assigned categories. The following figure. 1 shows the methodology of this research work.

3.1 Data Exploration

The dataset is explored in two ways-first explore the dataset by analyzing the number of surveys grouped by their release date (Year-Month). Next, explore the dataset by analyzing the number of surveys grouped by their Paper ID (first part).

3.1.1 Explore the dataset by analyzing the number of surveys grouped by their release date (Year-Month)

Step-1 Conversion of Release Date to DateTime Format The Release Date column in the dataset is likely stored as text or a non-datetime format. For

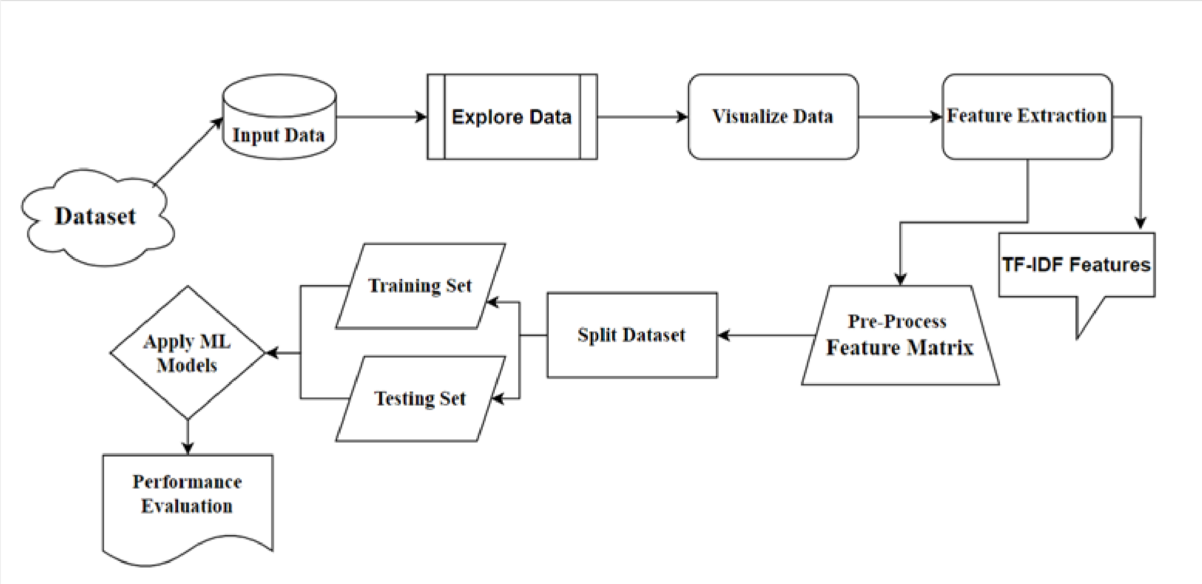

Figure 1: Methodology of the Taxonomy Classification

accurate grouping and manipulation, this column has to convert to a datetime object.
Step-2 Extracting Year and Month After the Release Date column is in datetime format, it's necessary to extract both the year and month from each release date for grouping purposes.
Step-3 Grouping the Data by Year and Month: Then, the number of survey papers are grouped by both year and month for the analysis of trends over time.

3.1.2 Explore the dataset by analyzing the number of surveys grouped by their Paper ID((first-part)

Step-1 Splitting Paper ID The Paper ID column contains unique identifiers for each survey, formatted with periods (e.g. 2023.0014). I extract the first part of this ID, which may represent a higher-level grouping of surveys.
Step-2 Grouping by Paper ID (First Part) After extracting the first part of the Paper ID, the dataset is grouped by this column to count how many surveys share the same Paper ID of its first part.

3.2 Data Visualization

Visualizations provide critical insights into the survey dataset, enabling the identification of trends, category distributions, and patterns across different dimensions. This Dataset is visualized into 3 ways with the help of "Matplotlib" Library.

i. Visualize the data by survey trends over time (Year-Month)

ii. Visualize the data by distribution of taxonomy categories

iii. Visualize the data by survey trends by Paper ID (First Part)

3.3 Feature Extraction

Feature extraction is a crucial step in the machine learning process, where raw text data is transformed into a format that can be processed by machine learning algorithms. In this work feature extraction has been done in the following way.

3.3.1 Extract features from the dataset by transforming the Title and Summary text fields using TF-IDF vectorization

Term Frequency-Inverse Document Frequency (TF-IDF) is a popular method used to convert text data into numerical representations by capturing the importance of words within a document relative to the entire corpus.
TF-IDF Process:
Term Frequency (TF) measures how frequently a word appears in a document. Words that appear more frequently are assigned higher values.

$$\mathrm{tf}_{t,d} = \frac{n_{t,d}}{\text{Number of terms in the document}} \tag{1}$$

Here, in the numerator, n is the number of times the term "t" appears in the document "d". Thus, each document and term would have its own TF value.
Inverse Document Frequency (IDF) reduces the weight of commonly occurring words (such as

"the" or "is") across all documents, giving more importance to unique or rare words in the corpus.

$$\text{idf}_t = \log\left(\frac{\text{Number of documents}}{\text{Number of documents with term } t}\right) \tag{2}$$

3.3.2 Encoding the Categories using one-hot encoding

One-hot encoding is a process of converting categorical data into a binary matrix representation. In this dataset, the Categories column contains multiple comma-separated categories (e.g., "Cs", "AI"). The goal is to transform each category into its own binary feature, where the value is 1 if the survey belongs to that category and 0 otherwise.

3.3.3 Combining All Features

After extracting features from the Title, Summary, and Categories, the next step is to combine them into a single feature matrix. This matrix will serve as the input for machine learning models.

3.4 Pre-Process Feature Matrix

In this part, the feature matrix is prepared for machine learning by applying normalization and label encoding. These preprocessing steps are crucial for ensuring that the features and labels are in a suitable format for machine learning models.

3.4.1 Normalization of the Feature Matrix

Normalization is a common preprocessing step where features are scaled to a specific range (usually between 0 and 1). This ensures that all features contribute equally to the model, and it prevents features with larger numerical values from dominating those with smaller values. For this task, the MinMaxScaler is used which scales each feature to a range between 0 and 1.

3.4.2 Label Encoding of the Taxonomy

Label encoding is used to convert the target labels (in this case, taxonomy categories) into a numerical format that machine learning models can process. Since the target labels are categorical, they must be transformed into numerical values.

3.5 Split Dataset

The dataset is split into two parts—train and test. I keep 70% of the data for training and 30% for testing.

3.6 Employing Classifiers/Models

I use five classifiers including logistic regression, decision tree, random forest, SVM, Naïve bayes to classify the taxonomy of the papers.

3.7 Performance Evaluation

I have used confusion matrix for measuring performance of our proposed model that is a tool which evaluates how well a classifier model performs. It provides the proportion of the model's accurate and inaccurate predictions in comparison with the actual results. A standard performance indicator generated from the confusion matrix is accuracy, which is the proportion of correct predictions to all predictions. However, accuracy alone may not provide a complete picture of the model's performance, as it can be biased by the distribution of the classes in the dataset. As a result, in addition to accuracy, other measures including F1 score, precision and sensitivity can be employed. I must define the following to evaluate the system's performance:

3.7.1 Accuracy: It refers to its ability to correctly distinguish between negative and positive categories. This can be expressed as:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \times 100\% \tag{3}$$

3.7.2 Sensitivity (Recall): The recall of a classification system refers to its ability to correctly identify positive cases. Mathematically, recall can be calculated by taking the total correctly identified positive instances and it divides by the total positive cases that actually exist. This can be expressed as:

$$\text{Sensitivity} = \frac{TP}{TP + FN} \times 100\% \tag{4}$$

3.7.3 Precision: Precision is a performance metric that computes the proportion of correctly classified instances that are positive out of every positive instance predicted by the categorization classifier. Mathematically, it is the fraction of true positive instances out of all cases classified as positive by the system or test, regardless of whether they are actually positive or negative. This can be expressed as:

$$\text{Precision} = \frac{TP}{TP + FP} \times 100\% \tag{5}$$

3.7.4 F1-score: A model's accuracy can be evaluated using the F1 score, which takes both precision and recall into account. The F1 score is defined as the harmonic mean of recall and precision. A higher F1 score, ranging from 0 to 1, signifies better performance.This can be expressed as follows:

$$\text{F1 Score} = \frac{2 \times \text{recall} \times \text{precision}}{\text{precision} + \text{recall}} \quad (6)$$

4 Experimental Results and Evaluation

4.1 Results of Data Exploration and Visualization

In this part it has been described about the data exploration and visualization

4.1.1 Survey Trends

In Fig. 2, a line chart is created to visualize the number of surveys released each month. The x-axis represents the Year-Month while the y-axis represents the number of surveys. This exploration reveals seasonal patterns in survey releases. Peaks are observed at regular intervals, indicating that certain periods are more active for research and publication.

Table 1: Survey Data Summary

	Year	Month	No. of Surv
count	15.000000	15.000000	15.000000
mean	2022.800000	7.000000	9.600000
std	0.676123	3.505098	6.231258
min	2021.000000	1.000000	1.000000
25%	2023.000000	4.500000	4.000000
50%	2023.000000	7.000000	11.000000
75%	2023.000000	9.500000	14.500000
max	2024.000000	12.000000	21.000000

Table1 provides descriptive statistics for a dataset containing "Year," "Month," and "Number of Surveys" for 15 observations. The average year is 2022.8, with surveys mainly conducted around July (mean month 7), and an average of 9.6 surveys per entry. The highest number of surveys recorded was 21, while the lowest was 1, and most of the surveys were concentrated between 4 and 14.5 surveys (25th to 75th percentiles). The data spans from 2021 to 2024.

4.1.2 Taxonomy Distribution:

The Fig. 3 shows the taxonomy distribution In Fig. 3, a bar chart is plotted to show the distribution of surveys across different taxonomy categories. The category "Trustworthy" has the highest number of papers, followed by other categories. This suggests a major research focus on building trustworthy category in the dataset.

Table 2: Taxonomy Counts

Taxonomy	Count
Trustworthy	26
Comprehensive	17
Prompting	17
Science	13
RecSys & IR	10
Multi-modal & Pre-training	9
Evaluation	9
Software Engineering	9
Adaptation Tuning	8
Robotics	8
Graphs	8
Others	5
Law	2
Finance	1
Education	1
Hardware Architecture	1

Table 2. lists topics with their respective frequencies under the "Taxonomy" category. "Trustworthy" is the most frequent topic with 26 entries, followed by "Comprehensive" and "Prompting" with 17 each. Less common topics include "Law" (2) and "Finance," "Education," and "Hardware Architecture," each with 1 entry.

4.1.3 Paper ID Part Trends

In Fig. 4, a line chart is used to visualize the trends based on the first part of the Paper ID.
Certain Paper ID (First-Part) values has a significantly higher number of surveys, indicating that some groups or institutions are more active in publishing research.

Table 3: Summary Statistics for Number of Surveys

Statistic	Value
Count	19.000000
Mean	7.578947
Std	5.965789
Min	1.000000
25%	1.500000
50%	7.000000
75%	12.500000
Max	20.000000

Table 3 summarizes statistics for "Number of Surveys" based on Paper ID (First-Part) across 19 observations. The average number of surveys is

Figure 2: Survey Trends Over Time (Year-Month)

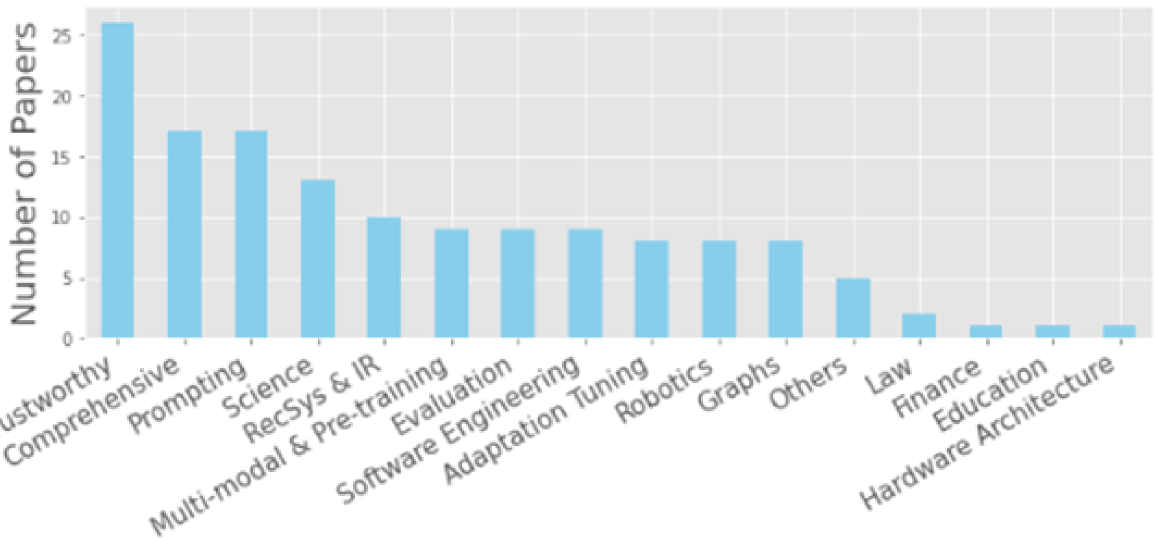

Figure 3: Distribution of taxonomy categories)

7.58, with a standard deviation of 5.97, indicating some variation. The range of surveys goes from a minimum of 1 to a maximum of 20, with the middle 50

4.2 Results of feature matrix

The Fig. 5 shows the feature matrix of the dataset.

4.3 Results of Classifiers

This research applies five distinct classifier models—Logistic Regression, Decision Tree, Random Forest, SVM, and Naïve Bayes—to a test dataset for classifying the taxonomy of research papers. Each classifier is evaluated individually on the dataset, and performance results are collected separately for comparison. Detailed analysis of the results is conducted, with confusion matrices constructed for each model to assess accuracy, precision, recall, and other performance metrics. The use of multiple classifiers provides a comprehensive understanding of how each model performs in classifying the dataset and give some insights into their strengths and weaknesses in handling the given task.
The following Figures (6, 7, 8, 9, 10) shows the confusion matrix of the classifiers applied to the dataset of this research work.

Table 4 presents the accuracy of different classifiers. Random Forest has the highest accuracy at 50%, followed by Logistic Regression at 48%. Decision Tree and Naïve Bayes both have an accuracy of 41%, while SVM performs the lowest at 39%. The performance difference highlights that Random Forest and Logistic Regression are

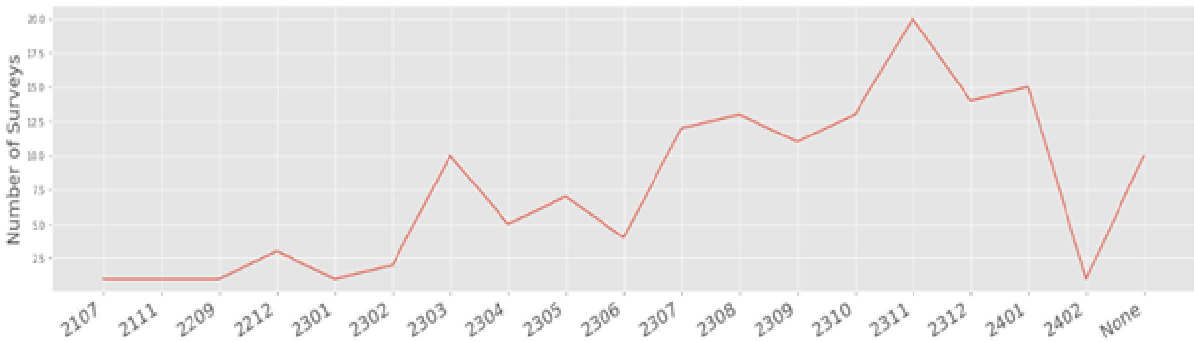

Figure 4: Distribution of taxonomy categories)

	0	1	2	3	4	5	6	7	8	9	...	_cs.CV	_cs.HC	_cs.LG	_cs.MA	_cs.MM	_cs.PF	_cs.PL	_cs.SI	_cs.LG	_cs.MA
0	0	0	0	0	0	0	0	0	0	0	...	False	False	False	False	False	False	False	False	False	False
1	0	0	0	0	0	0	0	0	0	0	...	False	False	False	False	False	False	False	False	False	False
2	0	0	0	0	0	0	0	0	0	0	...	False	False	False	False	False	False	False	False	False	False
3	0	0	0	0	0	0	0	0	0	0	...	True	False	False	False	False	False	False	False	True	False
4	0	0	0	0	0	0	0	0	0	0	...	False	False	False	False	False	False	False	False	False	False
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
139	0	0	0	0	0	0	0	0	0	0	...	False	False	False	False	False	False	False	False	False	False
140	0.353135	0	0	0	0	0	0	0	0	0	...	False	False	False	False	False	False	False	False	False	False
141	0	0	0	0	0	0	0	0	0	0	...	False	False	False	False	False	False	False	False	False	False
142	0	0	0	0	0	0	0	0	0	0	...	False	False	False	False	False	False	False	False	False	False
143	0	0	0	0	0	0	0	0	0	0	...	False	False	False	False	False	False	False	False	False	False

Figure 5: Output of Feature Matrix)

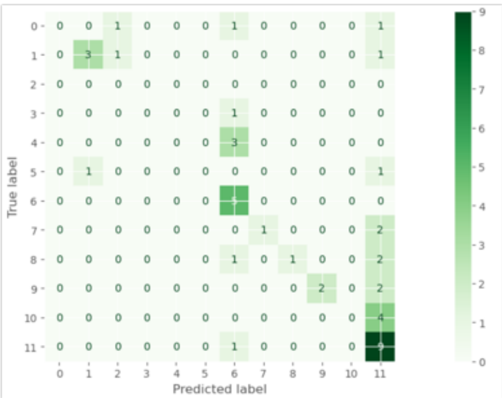

Figure 6: Confusion Matrix of Logistic Regression

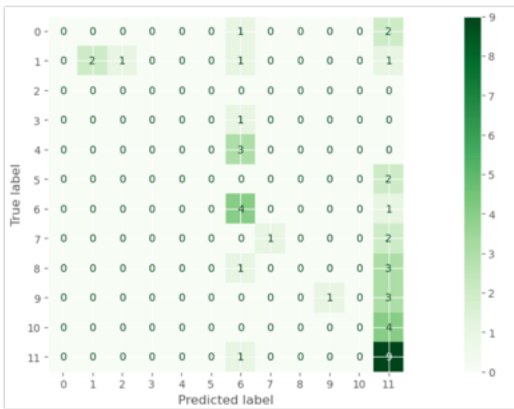

Figure 7: Confusion Matrix of Support Vector Machine

more effective for this task, possibly because Random Forest's ensemble method reduces overfitting, while Logistic Regression may capture linear patterns well.

SVM, despite being a powerful method, performs poorly here, likely due to the characteristics of the dataset. Decision Tree and Naïve Bayes perform moderately. This suggests that Random Forest and Logistic Regression may be more effective for this specific dataset or classification task.

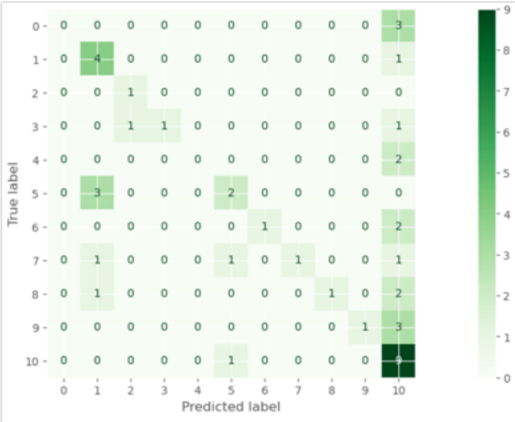

Figure 8: Confusion Matrix of Random Forest Classifier

Figure 9: Confusion Matrix of Decision Tree

5 Conclusion

In this project, we explored the "LLM Survey" dataset using various data exploration, manipulation, and evaluation techniques. We applied five different classifiers like Logistic Regression, Support Vector Machine (SVM), Random Forest, Decision Tree, and Naïve Bayes to classify the taxonomy of the papers. Each classifier was trained and evaluated using metrics like accuracy, precision, recall, and confusion matrices. The models demonstrated varying levels of performance, with Random Forest and SVM showing relatively higher accuracy in comparison to others. By analyzing trends over time and understanding the distribution of taxonomies, this work provided valuable insights into the dataset.

Table 4: Accuracy of Five classifiers

Classifier	**Accuracy (%)**
Logistic Regression	48
SVM	39
Random Forest	50
Decision Tree	41
Naïve Bayes	41

Figure 10: Confusion Matrix of Gaussian Naive Bayes

References

[1] Xubin Ren, Jiabin Tang, Dawei Yin, Nitesh Chawla, and Chao Huang. A survey of large language models for graphs. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 6616–6626, 2024.

[2] Xin Wang, Zirui Chen, Haofen Wang, Zhao Li, Wenbin Guo, et al. Large language model enhanced knowledge representation learning: A survey. *arXiv preprint arXiv:2407.00936*, 2024.

[3] Qiheng Mao, Zemin Liu, Chenghao Liu, Zhuo Li, and Jianling Sun. Advancing graph representation learning with large language models: A comprehensive survey of techniques. *arXiv preprint arXiv:2402.05952*, 2024.

[4] Renming Liu, Semih Cantürk, Frederik Wenkel, Sarah McGuire, Xinyi Wang, Anna Little, Leslie O'Bray, Michael Perlmutter, Bastian Rieck, Matthew Hirn, et al. Taxonomy of benchmarks in graph representation learning. In *Learning on Graphs Conference*, pages 6–1. PMLR, 2022.

[5] Mohaimenul Azam Khan Raiaan, Md Saddam Hossain Mukta, Kaniz Fatema, Nur Mohammad Fahad, Sadman Sakib, Most Marufatul Jannat Mim, Jubaer Ahmad, Mohammed Eunus Ali, and Sami Azam. A review on large language models: Architectures, applications, taxonomies, open issues and challenges. *IEEE Access*, 2024.

[6] Yichuan Li, Kaize Ding, and Kyumin Lee. Grenade: Graph-centric language model for self-supervised representation learning on text-attributed graphs. *arXiv preprint arXiv:2310.15109*, 2023.

[7] Jia Li, Xiangguo Sun, Yuhan Li, Zhixun Li, Hong Cheng, and Jeffrey Xu Yu. Graph intelligence with

large language models and prompt learning. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 6545–6554, 2024.

[8] Tao Shen, Yi Mao, Pengcheng He, Guodong Long, Adam Trischler, and Weizhu Chen. Exploiting structured knowledge in text via graph-guided representation learning. *arXiv preprint arXiv:2004.14224*, 2020.

[9] Delvin Ce Zhang, Menglin Yang, Rex Ying, and Hady W Lauw. Text-attributed graph representation learning: Methods, applications, and challenges. In *Companion Proceedings of the ACM on Web Conference 2024*, pages 1298–1301, 2024.

[10] Wei Ju, Zheng Fang, Yiyang Gu, Zequn Liu, Qingqing Long, Ziyue Qiao, Yifang Qin, Jianhao Shen, Fang Sun, Zhiping Xiao, et al. A comprehensive survey on deep graph representation learning. *Neural Networks*, page 106207, 2024.