Pattern Recognition and Neural Networks Final Project: Classification of Academic Papers Using Textual Analysis

De Angelis Matteo
University of Miami
mld163@miami.edu

***Abstract* — This paper explores the development of a classification system for academic papers based on their textual content. Using version 177 of the arXiv dataset repository comprising over 2.4 million STEM papers, we apply classification methods and data analysis techniques to categorize documents into a simplified set of categories. This study utilizes the abstracts of these papers, applying preprocessing techniques such as lemmatization and employing methods like TF-IDF and Singular Value Decomposition for feature extraction. The effectiveness of different classifiers, including Multinomial Naïve Bayes, K-Nearest Neighbors, and Support Vector Machines, is evaluated in terms of computational efficiency and accuracy.**

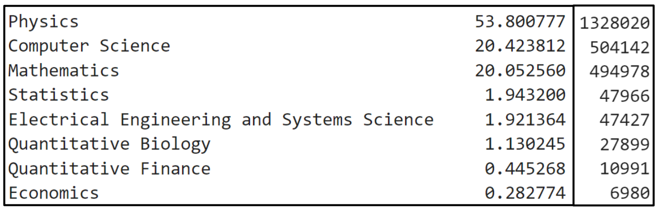
Keywords* — *TF-IDF, SVD, MultinomialNB, KNN, SVM

I. Introduction

The classification of academic papers by their textual content involves significant challenges due to the vast amount of data and the diversity of topics. This study leverages a large dataset from the arXiv repository [1], which includes papers categorized into 155 different categories [2]. By focusing on the abstracts and applying various preprocessing and feature extraction techniques, this project aims to develop an effective classification system.

II. Data Description

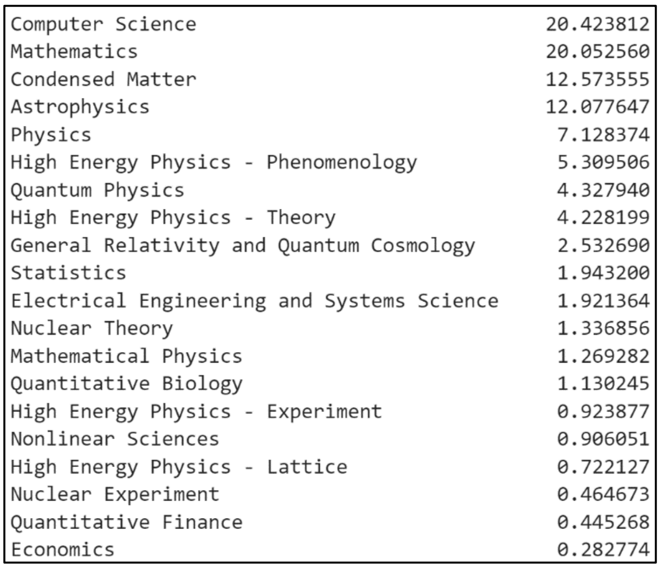
The dataset includes 2,468,403 STEM papers from arXiv, each labeled with one or more categories. Only the primary category, as per arXiv's policy, was used for classification. The data was imbalanced with a significant dominance of Physics papers, prompting the creation of a balanced subset including underrepresented categories such as Economics. Thus, a new smaller dataset was created comprised of all the papers with Economics as primary category and the same amount of papers chosen randomly from each other main category. Hence of size 55840 papers.



Physics	53.800777	1328020
Computer Science	20.423812	504142
Mathematics	20.052560	494978
Statistics	1.943200	47966
Electrical Engineering and Systems Science	1.921364	47427
Quantitative Biology	1.130245	27899
Quantitative Finance	0.445268	10991
Economics	0.282774	6980

Figure 1. Three columns, from left to right: name of the category, percentage of category papers in the dataset, number of papers of the category in the dataset.

Furthermore, another group of categories was created by expanding Physics into other main subcategories totaling 20 categories.



Computer Science	20.423812
Mathematics	20.052560
Condensed Matter	12.573555
Astrophysics	12.077647
Physics	7.128374
High Energy Physics - Phenomenology	5.309506
Quantum Physics	4.327940
High Energy Physics - Theory	4.228199
General Relativity and Quantum Cosmology	2.532690
Statistics	1.943200
Electrical Engineering and Systems Science	1.921364
Nuclear Theory	1.336856
Mathematical Physics	1.269282
Quantitative Biology	1.130245
High Energy Physics - Experiment	0.923877
Nonlinear Sciences	0.906051
High Energy Physics - Lattice	0.722127
Nuclear Experiment	0.464673
Quantitative Finance	0.445268
Economics	0.282774

Figure 2. Two columns, from left to right: name of the category, percentage of category papers in the dataset.

III. Methodology

A. Data Preprocessing

Data preprocessing involved extracting and tokenizing the words from the abstracts and removing non-alphanumeric characters, stopwords, and applying lemmatization to retain valuable linguistic stems.

B. Feature Extraction

We utilized the Term Frequency-Inverse Document Frequency (TF-IDF) algorithm to develop a feature sparse matrix, highlighting the uniqueness of words in documents [3].

First we calculate the TF (Term Frequency):

$$TF(t,d) = \frac{Number\ of\ times\ term\ t\ appears\ in\ document\ d}{Total\ number\ of\ terms\ in\ document\ d}$$

Then the IDF (Inverse Document Frequency):

$$IDF(t,D) = \log\left(\frac{Total\ number\ of\ documents\ in\ database\ D}{Number\ of\ documents\ containing\ term\ t}\right)$$

Finally, apply a score (the higher the score the more important term t is to a specific document):

$$TF\text{-}IDF(t,d,D) = TF(t,d) \times IDF(t,D)$$

Given the large size of the feature matrix, we applied Singular Value Decomposition (SVD) as part of Latent Semantic Analysis to reduce dimensionality [4]. However, space complexity turned out to be a big issue. A maximum of 85% explained variance was obtained from the smaller dataset.

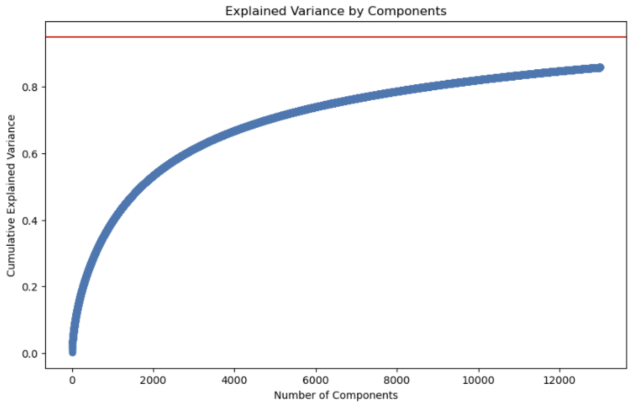


Figure 3. Plot of Explained Variance by Components.

C. Model Implementation

The same method was used to implement all three models. The data was split into 70% training papers and 30% testing papers. A 5-fold cross-validation was done on each model with one hyperparameter being fine-tuned per model. The model we experimented with were:

1. **Multinomial Naïve Bayes** — Used with Laplace smoothing for its computational efficiency, though its performance varied across different categorizations. The fastest by a big margin of the three models. [5]
2. **K-Nearest Neighbors** — Achieved high accuracy but was computationally demanding being the slowest of the three models and performed poorly in classifying Economics papers. We chose K-Nearest Neighbors as the hyperparameter. [6]
3. **Support Vector Machines** — Employed with a Radial Basis Function kernel as SVM rely on preprocessing the data to represent patterns in a high dimension [7]; showed robust performance across datasets, particularly when using reduced dimensionality features. C was chosen as the hyperparameter. [8]

D. Results and Discussion

The results highlighted the strengths and limitations of each model. While the SVM and KNN showed high accuracy, Naïve Bayes was faster but less accurate in complex categorizations. The cross-validation tests always show very good generalization across the evaluations. For reference, all the accuracies that are going to be presented in the following paragraphs relate to the testing accuracies unless otherwise specified.

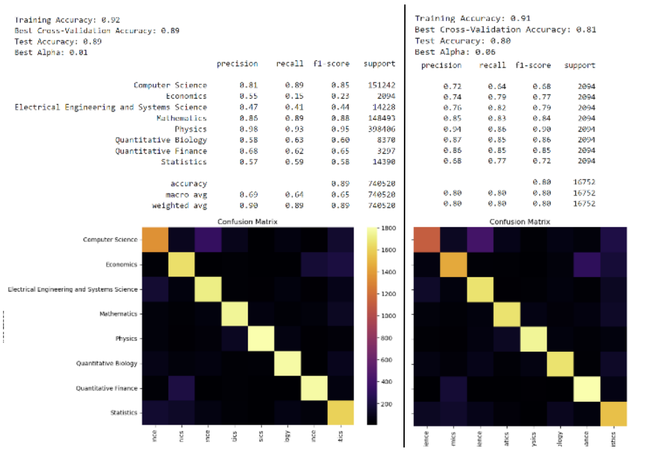
On the big dataset with the main categories, KNN performed best with a 95% testing accuracy, SVM performed with a 92% accuracy, and NB obtained a 89% accuracy.

On the small dataset, using the main categories, SVM performed the best with 81% accuracy, NB achieved 80% accuracy, and KNN obtained 76% accuracy.

Using the small dataset on the other group categories was not efficient as the data wasn't enough for accurate classification. This was also true using the bigger dataset, in particular when trying to classify all 155 categories. For example, the NB would be able to classify all 155 categories with a 30% accuracy, and the main categories with Physics expanded into main subcategories with 81% accuracy.

Although the smaller dataset always underperformed the bigger dataset in terms of overall accuracy, it was more precise into recognizing all the categories evenly. For example, using KNN with the bigger dataset on the main categories resulted in a 95% accuracy, but a 0% precision, recall, and f1-score for the Economics papers.

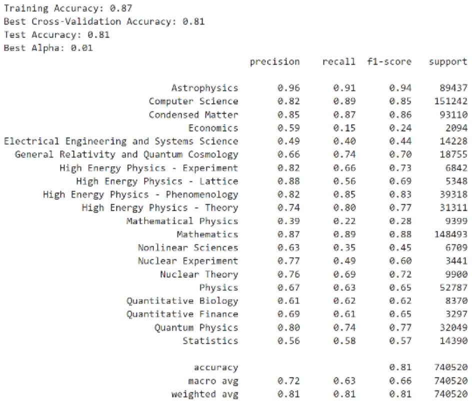
SVD helped in managing the high-dimensional data but was limited by computational resources. 13000 components and 85% explained variance wasn't enough to achieve a high level accuracy on the KNN. Nevertheless, very surprisingly, it outperformed the SVM model on the smaller dataset counterpart trained on TF-IDF feature matrix.



```
Training Accuracy: 0.92
Best Cross-Validation Accuracy: 0.89
Test Accuracy: 0.89
Best Alpha: 0.01
                                        precision    recall  f1-score   support

                      Computer Science       0.81      0.89      0.85    151242
                             Economics       0.55      0.15      0.23      2094
Electrical Engineering and Systems Science       0.47      0.41      0.44     14228
                           Mathematics       0.86      0.89      0.88    148493
                               Physics       0.98      0.93      0.95    398406
                  Quantitative Biology       0.58      0.63      0.60      8370
                  Quantitative Finance       0.68      0.62      0.65      3297
                            Statistics       0.57      0.59      0.58     14390

                              accuracy                           0.89    740520
                             macro avg       0.69      0.64      0.65    740520
                          weighted avg       0.90      0.89      0.89    740520
```

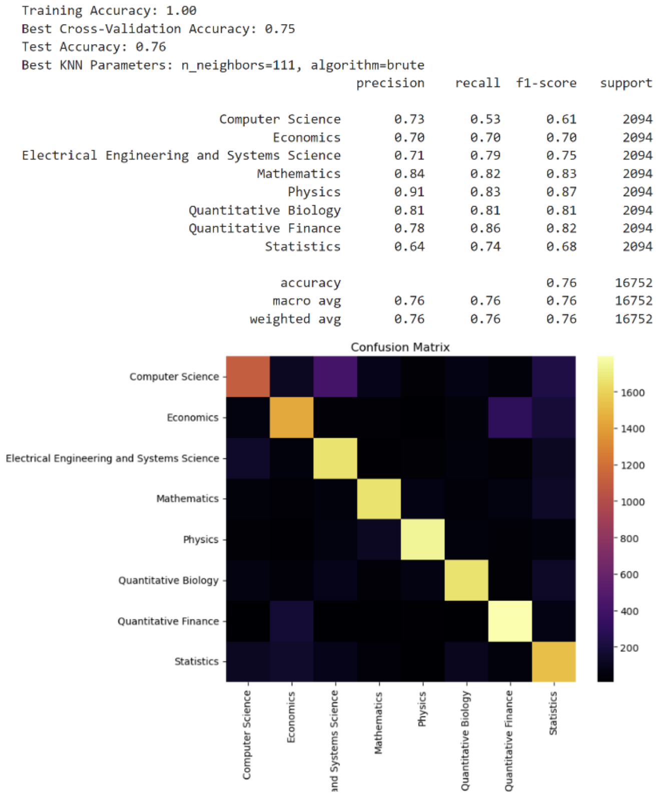
```
Training Accuracy: 0.91
Best Cross-Validation Accuracy: 0.81
Test Accuracy: 0.80
Best Alpha: 0.06
 precision    recall  f1-score   support

      0.72      0.64      0.68      2094
      0.74      0.79      0.77      2094
      0.76      0.82      0.79      2094
      0.85      0.83      0.84      2094
      0.94      0.86      0.90      2094
      0.87      0.85      0.86      2094
      0.86      0.85      0.85      2094
      0.68      0.77      0.72      2094

                          0.80     16752
      0.80      0.80      0.80     16752
      0.80      0.80      0.80     16752
```

Figure 4. Multinomial Naïve Bayes Model. Two columns, from left to right: accuracy, classification report and confusion matrix of the main categories in the big dataset, and accuracy, classification report and confusion matrix of the main categories in the small dataset.



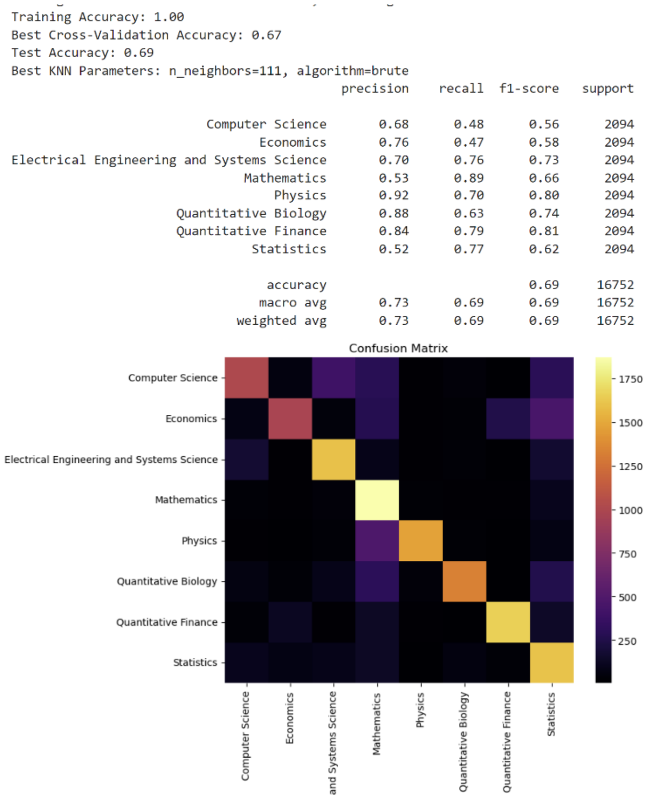
```
Training Accuracy: 0.87
Best Cross-Validation Accuracy: 0.81
Test Accuracy: 0.81
Best Alpha: 0.01
                                        precision    recall  f1-score   support

                         Astrophysics       0.96      0.91      0.94     89437
                     Computer Science       0.82      0.89      0.85    151242
                     Condensed Matter       0.85      0.87      0.86     93110
                            Economics       0.59      0.15      0.24      2094
Electrical Engineering and Systems Science       0.49      0.40      0.44     14228
 General Relativity and Quantum Cosmology       0.66      0.74      0.70     18755
        High Energy Physics - Experiment       0.82      0.66      0.73      6842
           High Energy Physics - Lattice       0.88      0.56      0.69      5348
     High Energy Physics - Phenomenology       0.82      0.85      0.83     39318
            High Energy Physics - Theory       0.74      0.80      0.77     31311
                   Mathematical Physics       0.39      0.22      0.28      9399
                            Mathematics       0.87      0.89      0.88    148493
                      Nonlinear Sciences       0.63      0.35      0.45      6709
                      Nuclear Experiment       0.77      0.49      0.60      3441
                          Nuclear Theory       0.76      0.69      0.72      9900
                                Physics       0.67      0.63      0.65     52787
                   Quantitative Biology       0.61      0.62      0.62      8370
                   Quantitative Finance       0.69      0.61      0.65      3297
                        Quantum Physics       0.80      0.74      0.77     32049
                             Statistics       0.56      0.58      0.57     14390

                               accuracy                           0.81    740520
                              macro avg       0.72      0.63      0.66    740520
                           weighted avg       0.81      0.81      0.81    740520
```

Figure 5. Multinomial Naïve Bayes Model. Accuracy and classification report of the big dataset on the main categories with Physics expanded into main subcategories.



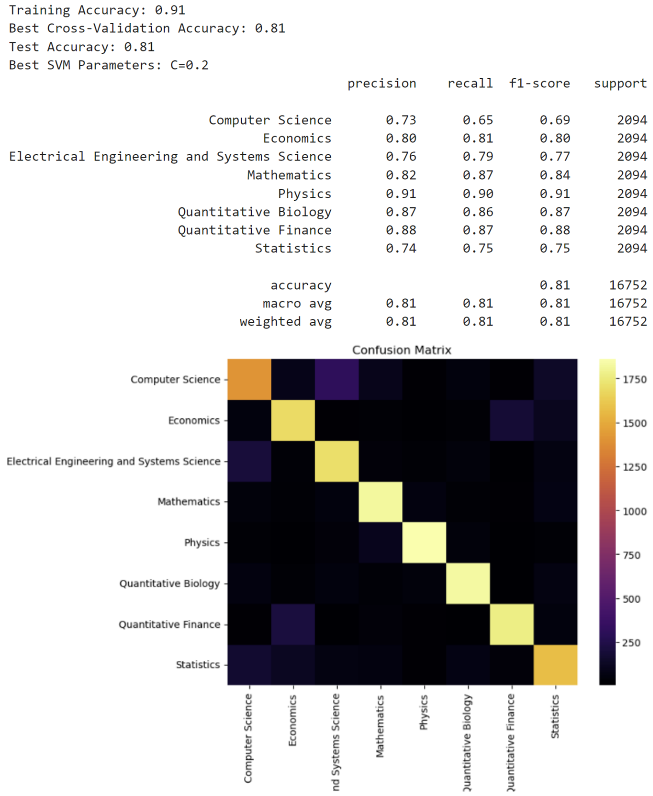
```
Training Accuracy: 1.00
Best Cross-Validation Accuracy: 0.75
Test Accuracy: 0.76
Best KNN Parameters: n_neighbors=111, algorithm=brute
                                        precision    recall  f1-score   support

                      Computer Science       0.73      0.53      0.61      2094
                             Economics       0.70      0.70      0.70      2094
Electrical Engineering and Systems Science       0.71      0.79      0.75      2094
                           Mathematics       0.84      0.82      0.83      2094
                               Physics       0.91      0.83      0.87      2094
                  Quantitative Biology       0.81      0.81      0.81      2094
                   Quantitative Finance       0.78      0.86      0.82      2094
                            Statistics       0.64      0.74      0.68      2094

                              accuracy                           0.76     16752
                             macro avg       0.76      0.76      0.76     16752
                          weighted avg       0.76      0.76      0.76     16752
```

Figure 6. K-Nearest Neighbors Model. Accuracy, classification report and confusion matrix of the small dataset with the main categories.



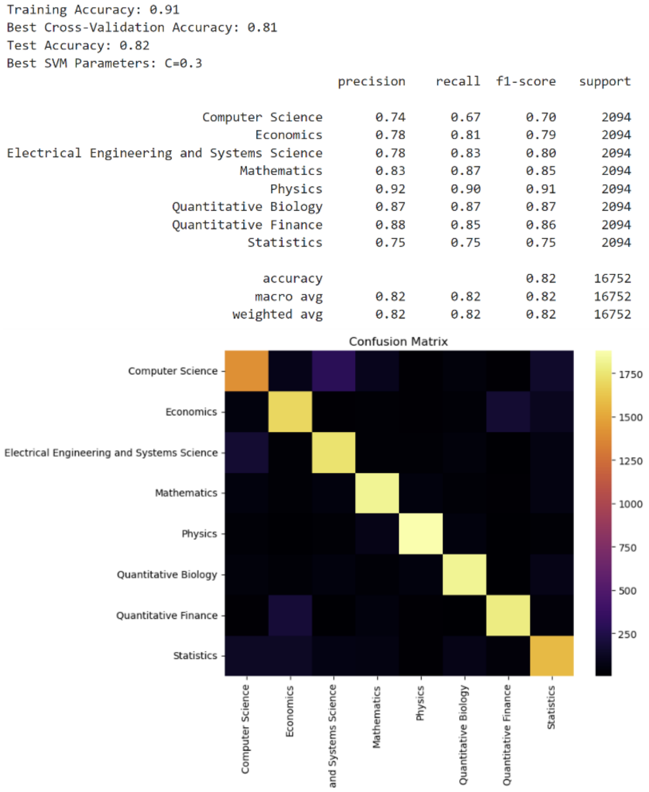
```
Training Accuracy: 1.00
Best Cross-Validation Accuracy: 0.67
Test Accuracy: 0.69
Best KNN Parameters: n_neighbors=111, algorithm=brute
                                        precision    recall  f1-score   support

                      Computer Science       0.68      0.48      0.56      2094
                             Economics       0.76      0.47      0.58      2094
Electrical Engineering and Systems Science       0.70      0.76      0.73      2094
                           Mathematics       0.53      0.89      0.66      2094
                               Physics       0.92      0.70      0.80      2094
                  Quantitative Biology       0.88      0.63      0.74      2094
                   Quantitative Finance       0.84      0.79      0.81      2094
                            Statistics       0.52      0.77      0.62      2094

                              accuracy                           0.69     16752
                             macro avg       0.73      0.69      0.69     16752
                          weighted avg       0.73      0.69      0.69     16752
```

Figure 7. K-Nearest Neighbors Model. Accuracy, classification report and confusion matrix of the small dataset with the main categories using Singular Value Decomposition features.



```
Training Accuracy: 0.91
Best Cross-Validation Accuracy: 0.81
Test Accuracy: 0.81
Best SVM Parameters: C=0.2
                                        precision    recall  f1-score   support

                      Computer Science       0.73      0.65      0.69      2094
                             Economics       0.80      0.81      0.80      2094
Electrical Engineering and Systems Science       0.76      0.79      0.77      2094
                           Mathematics       0.82      0.87      0.84      2094
                               Physics       0.91      0.90      0.91      2094
                  Quantitative Biology       0.87      0.86      0.87      2094
                   Quantitative Finance       0.88      0.87      0.88      2094
                            Statistics       0.74      0.75      0.75      2094

                              accuracy                           0.81     16752
                             macro avg       0.81      0.81      0.81     16752
                          weighted avg       0.81      0.81      0.81     16752
```

Figure 8. Support Vector Machines Model. Accuracy, classification report and confusion matrix of the small dataset with the main categories.



```
Training Accuracy: 0.91
Best Cross-Validation Accuracy: 0.81
Test Accuracy: 0.82
Best SVM Parameters: C=0.3
                                        precision    recall  f1-score   support

                      Computer Science       0.74      0.67      0.70      2094
                             Economics       0.78      0.81      0.79      2094
Electrical Engineering and Systems Science       0.78      0.83      0.80      2094
                           Mathematics       0.83      0.87      0.85      2094
                               Physics       0.92      0.90      0.91      2094
                  Quantitative Biology       0.87      0.87      0.87      2094
                   Quantitative Finance       0.88      0.85      0.86      2094
                            Statistics       0.75      0.75      0.75      2094

                              accuracy                           0.82     16752
                             macro avg       0.82      0.82      0.82     16752
                          weighted avg       0.82      0.82      0.82     16752
```

Figure 9. Support Vector Machines Model. Accuracy, classification report and confusion matrix of the small dataset with the main categories using Singular Value Decomposition features.

IV. Challenges and technical issues

The project faced significant computational challenges, particularly with SVD where retaining more than 13000 components was not feasible without encountering memory errors. Optimization attempts included reducing data precision and increasing available RAM, suggesting future exploration into High-Performance Computing solutions.

This study underscores the complexities of text classification at scale and the potential of advanced preprocessing and machine learning techniques to improve classification accuracy. Future work will focus on enhancing computational efficiencies and exploring more sophisticated Natural Language Processing tools.

References

[1] Cornell University, "arXiv Dataset," Kaggle, Version 177, Available: https://www.kaggle.com/datasets/Cornell-University/arxiv/data?select=arxiv-metadata-oai-snapshot.json.

[2] Arxiv, "Category taxonomy", Available: https://arxiv.org/category_taxonomy.

[3] Scikit-learn developers, "sklearn.feature_extraction.text.TfidfVectorizer — scikit-learn 0.24.1 documentation," Available: https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html.

[4] Scikit-learn developers, "sklearn.decomposition.TruncatedSVD — scikit-learn 0.24.1 documentation," Available: https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.TruncatedSVD.html.

[5] Scikit-learn developers, "sklearn.naive_bayes.MultinomialNB — scikit-learn 0.24.1 documentation," Available: https://scikit-learn.org/stable/modules/generated/sklearn.naive_bayes.MultinomialNB.html

[6] Scikit-learn developers, "sklearn.neighbors.KNeighborsClassifier — scikit-learn 0.24.1 documentation," Available: https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html

[7] R. O. Duda, P. E. Hart, and D. G. Stork, Pattern Classification, 2nd ed. New York, NY, USA: Wiley, 2001.

[8] Scikit-learn developers, "sklearn.svm.LinearSVC — scikit-learn 0.24.1 documentation," Available: https://scikit-learn.org/stable/modules/generated/sklearn.svm.LinearSVC.html