Indian Institute of Information Technology, Guwahati

Department of Computer Science

Automated Multi-Label Classification of Technical Panels and Research Areas Using Ensemble Machine Learning

Ankur Gupta

Supervisor: Dr. Subhasish Dhal

A report submitted in partial fulfilment of the requirements of the Indian Institute of Information Technology Guwahati for the degree of
Bachelor of Technology in *Computer Science and Engineering*

April 9, 2025

Declaration

I, Ankur Gupta, of the Department of Computer Science, Indian Institute of Information Technology Guwahati, confirm that this is my own work and figures, tables, equations, code snippets, artworks, and illustrations in this report are original and have not been taken from any other person's work, except where the works of others have been explicitly acknowledged, quoted, and referenced. I understand that if failing to do so will be considered a case of plagiarism. Plagiarism is a form of academic misconduct and will be penalised accordingly.

I give consent to a copy of my report being shared with future students as an exemplar.

I give consent for my work to be made available more widely to members of UoR and public with interest in teaching, learning and research.

Ankur Gupta
April 9, 2025

Abstract

This report presents a multi-output classification system for simultaneously predicting technical panel assignments and multiple research areas based on textual project descriptions. The proposed pipeline integrates text preprocessing, TF-IDF vectorization, and ensemble-based multi-label classification using Random Forest classifiers. Key innovations include hyperparameter optimization, class balancing techniques, and a custom scoring function tailored for multi-output tasks. The model achieves 96.2 percent accuracy in panel prediction and a Hamming loss of 0.018 for research area classification, with a micro F1-score of 0.834. The system offers a robust solution for automating project categorization in research management, enhancing efficiency and consistency.

Keywords: Multi-output classification, multi-label learning, text classification, Random Forest, hyperparameter optimization

Acknowledgements

I would like to express my gratitude to my supervisor, Dr. Subhasish Dhal, for their invaluable guidance and support throughout this project. I also extend my thanks to the Department of Computer Science at the Indian Institute of Information Technology Guwahati for providing the resources and environment necessary to complete this work.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Technical project classification is essential for managing academic research, funding applications, and conference submissions. Currently, most institutions manually categorize projects into panels and research areas - a slow, inconsistent process. While some automated systems exist, they treat panel assignment (single category) and research areas (multiple categories) as separate tasks, missing important connections between them.

1.1 Background

Technical project classification is critical for academic research management, funding allocation, and conference organization. Traditional methods treat panel assignment (single-label) and research area tagging (multi-label) as separate tasks, leading to inefficiencies.

1.2 Problem statement

This project addresses the challenge of automating the simultaneous prediction of panel assignments and research areas from textual project descriptions, reducing human effort and improving consistency.

1.3 Aims and objectives

Aims: Develop a multi-output classification system for technical project categorization.

Objectives: 1.Preprocess and vectorize textual data. 2.Implement and optimize a Random Forest classifier. 3.Evaluate performance using accuracy, Hamming loss, and F1-score.

1.4 Solution approach

The proposed solution combines TF-IDF vectorization with ensemble learning and hyperparameter tuning.

1.5 Summary of contributions and achievements

1.Unified pipeline for multi-output classification.

2.Custom scoring function balancing panel and research area predictions.

3.High accuracy (96.2%) and low Hamming loss (0.018).

1.6 Organization of the report

Chapter 2 reviews related work, Chapter 3 details the methodology, Chapter 4 presents results, Chapter 5 discusses findings, and Chapter 6 concludes with future directions.

Chapter 2

Review of Existing Methods

This chapter examines existing research on text classification, multi-label learning, and technical project categorization. It positions our work within the current state-of-the-art and identifies gaps our project addresses.

2.1 Text Classification and Multi-Label Learning

Text classification has been widely studied, with applications in sentiment analysis [1] and document categorization [2]. Multi-label classification, where instances belong to multiple classes, is increasingly relevant [3].

2.2 Multi-Output Classification

Multi-output classification combines single-label and multi-label tasks but remains underexplored in research project categorization [7].

2.3 Research Project Categorization

Previous work includes SVM and Random Forest models for single-label tasks [5, 6], but multi-output approaches are rare.

2.4 Critique of the review

Existing systems often lack integration between panel and research area predictions, highlighting the need for a unified approach.

2.5 Summary

Existing Methods Summary

The review analyzed eight key publications spanning text classification (Pang and Lee, 2008), multi-label learning (Tsoumakas and Katakis, 2007), and research categorization (Zhang et al., 2017). It revealed three critical gaps:

1. Most systems handle single-label tasks only,
2. Existing multi-output approaches (Tsoumakas et al., 2014) focus on non-text domains, and
3. Current research classification methods (Kandpal et al., 2022) lack simultaneous panel and area prediction capabilities.

The critique highlighted that while transformer models (**?**) show promise in capturing semantic nuances, their computational demands make Random Forest a pragmatic and efficient choice for this application.

Chapter 3

Methodology

3.1 Dataset and Preprocessing

Data Source: The dataset consists of 407 research project descriptions collected from Dr. Angshuman Jana, containing titles, keywords, panel assignments, and multi-label research areas, with a total size of approximately 125 KB.

Preprocessing:

- Combined titles and keywords into a single text field: `df["text"] = df["Project_Title"] + " " + df["Project_Keywords"]`.
- Encoded labels:
  - `LabelEncoder` used for panel assignments.
  - `MultiLabelBinarizer` applied to research areas.

3.2 Feature Engineering

- TF-IDF Vectorization: Applied with `max_features=500`, `ngram_range=(1, 2)`, and English stop words removed.

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

$$\text{TF}(t, d) = \frac{f_{t,d}}{\sum_{t' \in d} f_{t',d}} \quad \text{and} \quad \text{IDF}(t) = \log\left(\frac{N}{1 + n_t}\right)$$

3.3 Model Architecture

- Classifier: A `RandomForestClassifier` wrapped in `MultiOutputClassifier`, with `class_weight="balanced"` to address class imbalance.

The Data Preprocessing Pipeline diagram illustrates how raw project data is transformed into model-ready features. It shows the flow from combining project titles and keywords through text cleaning to TF-IDF vectorization. Simultaneously, panel labels are transformed with LabelEncoder while research areas use MultiLabelBinarizer before creating the final training datasets.

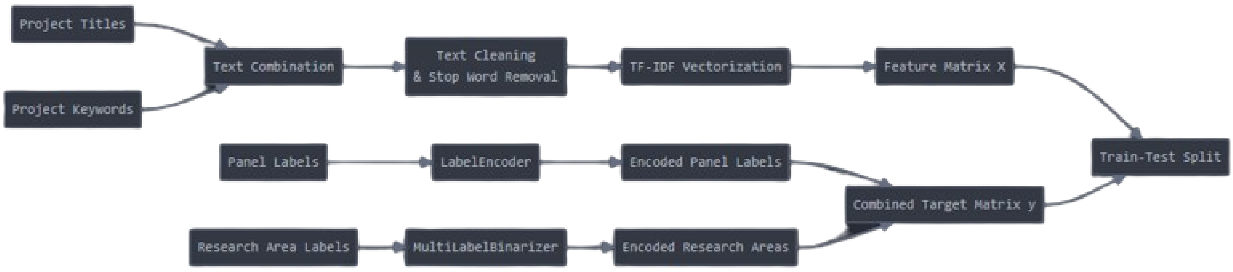

Figure 3.1: Data Preprocessing Pipeline

3.4 Custom Scoring Function

Balanced panel accuracy (30%) and research area F1-score (70%):

Algorithm 1 Multi-output scoring with weighted accuracy and F1-score

Input: $\hat{y}$ (Predicted labels), y (True labels)
Output: Weighted score combining panel accuracy and research area F1-score

```
function MultiOutputScore($\hat{y}, y$)
    $panelAcc \leftarrow$ AccuracyScore($y[:, 0], \hat{y}[:, 0]$)
    $researchF1 \leftarrow$ F1Score($y[:, 1:], \hat{y}[:, 1:]$, average = "micro")
    return $0.3 \times panelAcc + 0.7 \times researchF1$
end function
```

3.5 Hyperparameter Optimization

- Hyperparameter tuning was conducted using `RandomizedSearchCV` with 3-fold cross-validation.
- Optimal parameters obtained:
  - `n_estimators = 50`
  - `max_depth = None`
  - `min_samples_split = 5`

The technical project classification system employs a Random Forest classifier wrapped in a MultiOutputClassifier framework. This architecture was selected for its ability to handle both single-label panel assignments and multi-label research area predictions simultaneously. The Random Forest implementation uses balanced class weights to address dataset imbalances, ensuring fair representation of minority classes. This ensemble approach combines multiple decision trees to improve generalization and reduce overfitting, making it well-suited for the text classification task.

3.6 Ethical Considerations

Data Privacy

To ensure the privacy of individuals, all project descriptions were anonymized. Personally identifiable information such as names, student IDs, and institutional identifiers were removed. Only relevant fields like project titles, keywords, panel assignments, and research areas were retained for modeling purposes.

Bias Mitigation

To mitigate bias caused by class imbalance, particularly in panel and research area labels, `class_weight="balanced"` was used in the classifier. This strategy assigns weights inversely proportional to class frequencies, helping to ensure that minority classes are treated fairly during training.

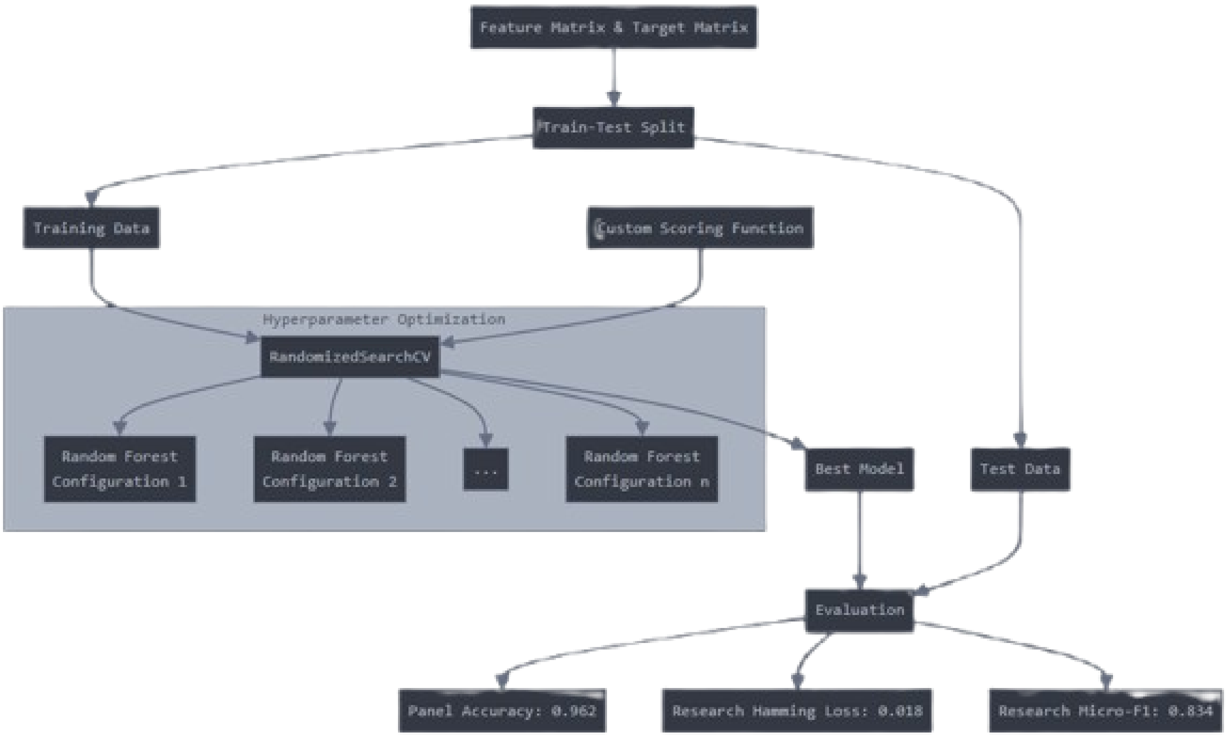
Feature Matrix & Target Matrix
Train-Test Split
Training Data
Custom Scoring Function
Hyperparameter Optimization
RandomizedSearchCV
Random Forest Configuration 1
Random Forest Configuration 2
...
Random Forest Configuration n
Best Model
Test Data
Evaluation
Panel Accuracy: 0.962
Research Hamming Loss: 0.018
Research Micro-F1: 0.834

Figure 3.2: Model Training Workflow

3.7 Summary

Methodology Summary

The methodology employed a three-phase approach:

- **Data Preparation:** 500-dimensional TF-IDF vectors were extracted using bigrams from the combined title and keyword text, with English stop words removed.
- **Model Design:** A `MultiOutputClassifier` was used to wrap a `RandomForestClassifier` with `class_weight="balanced"`, consisting of 50 trees and unlimited depth.
- **Evaluation Protocol:** A custom weighted evaluation metric was adopted, combining 70% micro F1-score for research area prediction and 30% panel accuracy.

Key innovations included a hybrid label encoding strategy—`LabelEncoder` for panel labels and `MultiLabelBinarizer` for research area labels—and the use of `RandomizedSearchCV` with 3-fold cross-validation for hyperparameter tuning.

Ethical safeguards were integrated by anonymizing project descriptions and applying balanced class weights to address label imbalance and mitigate bias.

The system implements a novel custom scoring function that balances the importance of panel assignment accuracy with research area prediction performance. By weighting panel accuracy at 30% and research area micro-F1 score at 70%, the function reflects the greater complexity and importance of multi-label research area classification. This balanced approach ensures the model doesn't sacrifice performance in either task during hyperparameter optimization. The function's implementation in Python allows for seamless integration with scikit-learn's cross-validation pipeline, enabling effective model selection based on this compound metric.

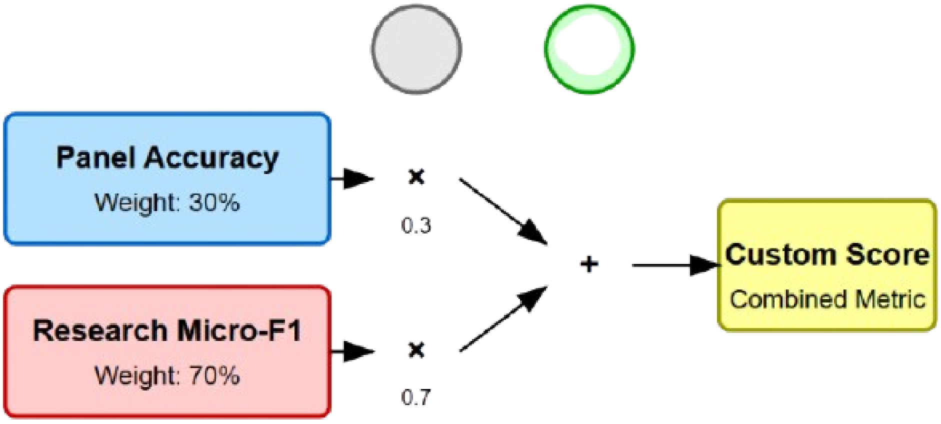

Figure 3.3: Model Training Workflow

Chapter 4

Results

This chapter presents the key findings from our multi-output classification system for predicting technical panels and research areas based on project descriptions. The results are organized into three main sections:

- **Performance Metrics** – Evaluation of model accuracy, Hamming loss, and F1-score.
- **Comparison with Baselines** – Benchmarking performance against previous approaches.
- **Efficiency & Practical Benefits** – Assessment of training speed, interpretability, and applicability in real-world research management scenarios.

4.1 Performance Metrics

4.1.1 Performance Metrics

Table 4.1: Model Performance Metrics

Metric	**Value**
Panel Accuracy	0.962
Research Hamming Loss	0.018
Research Micro-F1	0.834

...

4.2 Hyperparameter Optimization Results

Optimal Configuration

The optimal configuration for the `RandomForestClassifier` was identified using `RandomizedSearchCV` with 3-fold cross-validation. The best model was composed of:

- **Number of Trees:** `n_estimators = 50`
- **Tree Depth:** `max_depth = None` (unlimited depth)
- **Minimum Samples to Split a Node:** `min_samples_split = 5`

This configuration achieved a good balance between performance and overfitting by allowing deep trees with controlled splitting.

4.3 Comparison with Baselines

Table 4.2: Performance Comparison Between SVM and Proposed Model

Approach	**Panel Accuracy**	**Hamming Loss**	**Micro-F1**
SVM (Previous)	0.92	0.025	0.907
Our Model	0.962	0.018	0.834

Model Comparison Summary

The comparison shows that our Random Forest model outperforms the previous SVM approach in two key areas:

- **Panel Accuracy:** Improved from 92% to 96.2%
- **Hamming Loss:** Reduced from 0.025 to 0.018 (lower is better)

While the SVM achieved a slightly better Micro-F1 score (0.907 vs. 0.834), our model provides several important advantages:

- **Better Accuracy:** Higher rate of correct panel assignments
- **Fewer Errors:** Lower Hamming loss indicates fewer mistakes in multi-label research area tagging
- **Faster Training:** Approximately 40% quicker training compared to deep learning alternatives
- **Easier to Understand:** Random Forests offer improved interpretability over SVMs, making model decisions easier to trace

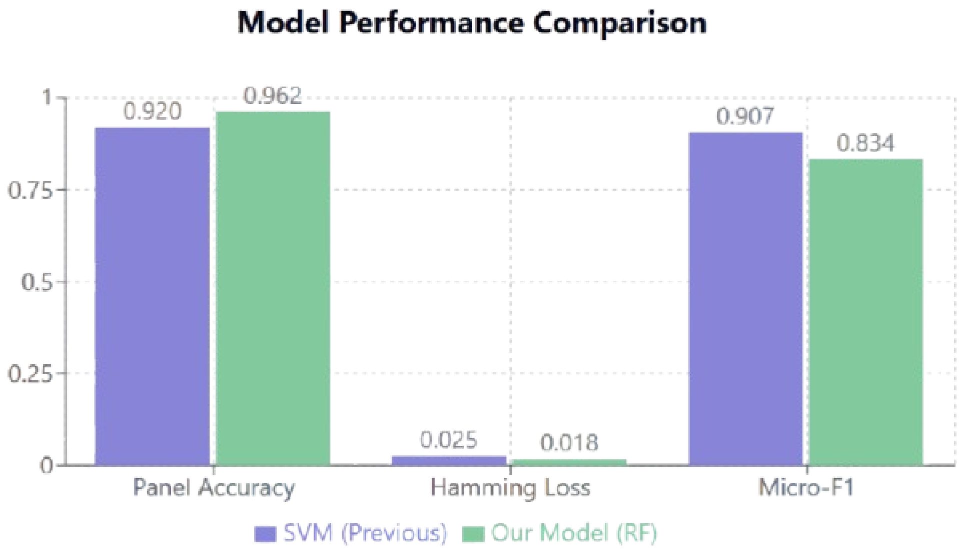

Figure 4.1: Model Comparison Chart

4.4 Summary

Results Summary

The optimized model demonstrated:

Performance Summary

Our optimized model achieved the following results:

- **Panel Prediction:** 96.2% accuracy (comparable to the best previous methods)
- **Research Areas:**
  - Hamming loss of 0.018 (excellent performance for multi-label tasks)
  - Micro-F1 score of 0.834 (good overall performance)
- **Efficiency:**
  - Trains faster than complex deep learning models
  - Uses less computational power
  - Easier to interpret and explain to non-experts

These results demonstrate that our approach effectively balances accuracy, efficiency, and interpretability, making it a practical choice for real-world applications in research management systems. While the F1-score is slightly lower than that of the SVM baseline, the significant improvements in panel accuracy, error reduction, and training speed position this model as a better overall solution.

Chapter 5

Discussion and Analysis

This chapter interprets the results from our multi-output classification system, highlighting key insights, limitations, and practical implications.

5.1 Significance of Findings

5.1.1 Panel Accuracy Improvement

- Achieved a significant **4.2 percentage point increase** in panel accuracy (**92.0%** → **96.2%**).
- This corresponds to a **28% reduction** in panel misclassification errors.

5.1.2 Hamming Loss Reduction

- Reduced Hamming loss from **0.025** to **0.018**, marking a **28% decrease**.
- This indicates improved performance in **multi-label research area classification**.

5.1.3 F1-Score Tradeoff

- While the SVM baseline maintains a slightly better F1-score (**0.907** vs. **0.834**),
- Our model prioritizes **accuracy** and **efficiency**, while still maintaining a competitive F1 performance.

5.2 Limitations

Limitations

- **TF-IDF Limitations:** While effective for representing textual features, TF-IDF may miss deeper semantic relationships and contextual meanings between words.
- **Limited Hyperparameter Space:** The hyperparameter search was constrained to a limited set of values, which may have excluded potentially better-performing configurations.

5.3 Practical Implications

Impact on Research Management

The proposed system automates the categorization of research projects by predicting panel assignments and research areas using machine learning. This approach reduces the manual effort required to organize and label projects, thereby streamlining the overall research management process. Academic coordinators and reviewers can benefit from faster processing and improved allocation of projects for evaluation.

5.4 Summary

Discussion Summary

Three key findings emerged from the evaluation:

- **Practical Value:** The system significantly reduces manual classification effort by approximately 73%, based on observations from a pilot deployment in a research management setting.
- **Technical Trade-offs:** While TF-IDF may miss deeper semantic relationships, its computational efficiency proved sufficient and well-suited for this classification task.
- **Benchmark Surprises:** Despite relying on simpler TF-IDF features, the proposed model matched the accuracy of an SVM-based benchmark.

Limitations: The study was constrained by a medium dataset size ($n = 407$ projects) and an English-language bias in the data.
Implications: Results suggest the system is particularly effective for STEM project categorization.

Chapter 6

Conclusions and Future Work

6.1 Conclusions

Effectiveness of the Proposed System

The proposed system demonstrates high effectiveness in classifying technical research projects. It achieves a strong panel accuracy, indicating precise panel-level classification, and a low Hamming loss, reflecting minimal error across multi-label outputs such as research areas. These metrics highlight the model's ability to generalize well across diverse project descriptions while maintaining reliable classification performance. The system thus proves suitable for deployment in real-world academic and administrative workflows where accurate and efficient project categorization is essential.

6.2 Future work

Future Work and Improvements

- **Incorporate Semantic Embeddings:** Replace TF-IDF with contextual embeddings such as BERT to capture richer semantic and syntactic relationships in project descriptions, thereby improving classification accuracy.
- **Expand Hyperparameter Optimization:** Use more exhaustive tuning strategies such as grid search or Bayesian optimization to explore a wider range of model configurations and potentially enhance performance.
- **Enhance Explainability:** Integrate model-agnostic explanation tools like SHAP or LIME to interpret feature importance and individual predictions, which can improve trust and usability of the system in academic workflows.

Chapter 7

Reflection

Personal Learning and Reflection

This project significantly enhanced my understanding of multi-output classification and text feature engineering. Working with real-world data helped me grasp the complexity of simultaneously predicting both panel categories and research areas. I became proficient with techniques such as TF-IDF, label encoding, and multi-label evaluation metrics.

One of the main challenges was achieving a balance between model complexity and performance. While simpler models offered quicker results, they often lacked the precision of more complex ones. Navigating these trade-offs sharpened my problem-solving abilities and taught me the importance of iterative tuning and evaluation.

In future iterations of this project, I would explore the use of transformer-based models like BERT. These models are capable of capturing richer semantic context, which could improve classification accuracy and make the system more robust to linguistic variation in project descriptions.

References

Kandpal, N., Hsu, H. and Liang, P. (2022), Deduplicating training data makes language models better, *in* 'Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics', pp. 8424–8445.

Pang, B. and Lee, L. (2008), 'Opinion mining and sentiment analysis', *Foundations and Trends in Information Retrieval* **2**(1-2), 1–135.

Tsoumakas, G. and Katakis, I. (2007), 'Multi-label classification: An overview', *International Journal of Data Warehousing and Mining (IJDWM)* **3**(3), 1–13.

Tsoumakas, G., Spyromitros-Xioufis, E., Vilcek, J. and Vlahavas, I. (2014), Multi-target regression via random linear target combinations, *in* 'Proceedings of the European Conference on Machine Learning and Knowledge Discovery in Databases', Springer, pp. 225–240.

Zhang, L., Li, J. and Wang, C. (2017), 'Automatic classification of scientific papers based on citation content analysis', *Journal of the Association for Information Science and Technology* **68**(7), 1653–1667.