Project Sentry: Project Manager With AI Based Duplicate Topic Detection

Prof. Monali Bansode[1], Prathamesh Tondilkar[2], Arjun Joshi[3], Hemant Singh[4], Nishant Poojari[5]
[1, 2, 3, 4, 5] Dept of Computer Engineering
[1, 2, 3, 4, 5] Sinhgad Institute of Technology Lonavala, India

Abstract- *With the increasing volume of final-year project sub- missions in educational institutions, ensuring the uniqueness and originality of student work has become a pressing issue. Duplicate project topics compromise innovation and create challenges for faculty during the evaluation process. Traditional methods of identifying duplicates are manual, time-consuming, and prone to human error. To address this issue effectively, there is a need for an intelligent system that not only detects duplicate topics but also supports streamlined project management.*

This research introduces an AI-driven approach that leverages natural language processing and machine learning to detect duplicate project submissions. Key textual components such as project title, description, domain, and technologies are combined, preprocessed, and transformed into numerical features using TF- IDF and dimensionality reduction techniques like SVD. A robust classifier, specifically XGBoost, is trained to distinguish between unique and duplicate projects. In addition to classification, the system uses cosine similarity to provide a duplication score and prediction confidence, ensuring transparency and reliability in decision-making.

Beyond prediction, the system also contributes to efficient project management. By organizing project data, tracking sub- mission originality, and integrating a duplicate detection mecha- nism, the system aids faculty in supervising projects effectively. It empowers academic institutions to uphold integrity while managing large datasets of student work. The solution is scalable, interpretable, and practical for integration into college-level project portals, fostering a culture of innovation and account- ability in academic environments.

Keywords- Project Management System, Duplicate Project Detection, Natural Language Processing (NLP), Machine Learn- ing (ML), Deep Learning, Text Similarity Analysis, TF-IDF Vectorization, Truncated SVD, Cosine Similarity, XGBoost Classifier, Academic Integrity

I. INTRODUCTION

In every academic institution, final-year projects are an essential component of undergraduate education. These projects are designed to help students apply what they've learned over the years to solve real-world problems or explore emerging technologies. However, as the number of students and submissions grows, colleges face a major challenge: duplicate project topics.

Duplicate Projects — where different students submit sim- ilar ideas, often unknowingly — dilute innovation, reduce academic value, and make evaluation harder for faculty. Some- times, students intentionally reuse ideas from previous years or other colleges, leading to issues of academic dishonesty. Other times, the duplication is accidental, simply because many students gravitate toward popular technologies or problem statements.

Currently, most colleges rely on manual checking to find such overlaps, which is both time-consuming and subjective. Faculty members compare titles and descriptions by eye, which is not only inefficient but also inconsistent. With hun- dreds of submissions, this becomes practically unmanageable. There is a growing need for an automated, scalable, and intelligent system that can assist in identifying duplicate or similar project ideas early in the submission process.

This research addresses this problem by proposing a ma- chine learning-based system that can automatically detect whether a new project topic is unique or potentially a du- plicate. Our approach uses natural language processing (NLP) to understand and process the content of project titles, de- scriptions, domains, and technologies. These text inputs are then converted into numerical features using TF-IDF (Term Frequency-Inverse Document Frequency), a technique that helps the model understand the importance of words in a document.

To make the system more efficient, we apply dimensionality reduction using Truncated Singular Value Decomposition (SVD), and then train a robust XGBoost

classification model. The system learns patterns from past project data and determines whether a new submission is similar enough to be flagged as a duplicate. The model is also capable of providing a confidence score and a similarity percentage, giving faculty more insight into each prediction.

The goal is not just to catch plagiarism but to foster origi- nality and creativity among students. By automating duplicate detection, this system can save valuable time for faculty, ensure fair evaluation, and encourage students to explore novel and diverse project ideas.

II. EASE OF USE

The Project Management System with integrated Duplicate Topic Detection has been carefully developed with a primary focus on user-friendliness and accessibility. The system en- sures that users, whether students or faculty members, can in- teract with the platform seamlessly without needing advanced technical knowledge. The submission process is streamlined, requiring only the project title, domain, and description (in- cluding technologies used), minimizing the effort needed to provide input.

The duplicate detection feature operates automatically once the details are submitted, instantly analyzing the information and displaying a clear plagiarism percentage through the integrated plagiarism meter. This eliminates any complex con- figurations or manual interventions typically associated with similarity checking tools. Furthermore, the system's clean and intuitive interface guides users effortlessly through the project submission, detection, and management processes, ensuring that even first-time users can navigate and utilize the platform effectively.

By minimizing unnecessary complexity and focusing on a smooth user experience, the system promotes wider adoption across academic environments. The intuitive design, coupled with real-time feedback mechanisms, significantly enhances the efficiency and satisfaction of users engaging with project registration and validation tasks.

III. LITERATURE SURVEY

The task of duplicate detection in academic projects, papers, and documents has been explored in various studies. Several approaches have been proposed to identify duplicate content, particularly in text mining and natural language processing (NLP) domains. This section highlights key works that are relevant to the current research, focusing on methods for dupli- cate detection, project topic classification, and text similarity measurement.

A. Duplicate Detection Methods

Duplicate detection in text is primarily based on text sim- ilarity analysis. Various algorithms, such as cosine similar- ity, Jaccard similarity, and Levenshtein distance, are used to measure how similar two texts are. Salton et al. proposed the vector space model for automatic indexing and retrieval of documents, which laid the groundwork for modern duplicate detection [18]. The vector space model relies on representing documents as vectors and computing similarity based on term frequency.

In recent years, machine learning techniques have been widely adopted for duplicate detection. Aggarwal and Zhai surveyed different clustering algorithms that can be used for text classification and duplicate detection [12]. These algo- rithms group similar documents into clusters, making it easier to detect duplicates by comparing cluster labels. The intro- duction of deep learning models, particularly the Transformer architecture [6], revolutionized text similarity tasks. Vaswani et al. presented a self-attention mechanism that has become a cornerstone of NLP research, allowing for more accurate text matching and semantic understanding.

B. Topic Modeling and Classification

To address duplicate topic detection in project submissions, topic modeling approaches can be highly effective. Latent Dirichlet Allocation (LDA) is one of the most commonly used models for extracting topics from text [21]. LDA generates a distribution over topics for each document and a distribution over words for each topic, making it ideal for discovering underlying themes in project descriptions. Blei et al. demon- strated the power of LDA in analyzing large collections of text and extracting meaningful topics, which can be applied to academic project datasets to detect duplicate topics.

Recent advancements in pre-trained language models, such as BERT (Bidirectional Encoder Representations from Trans- formers) [7], have further enhanced the accuracy of text classi- fication and similarity detection. BERT's ability to understand context and capture semantic meaning has led to significant improvements in various NLP tasks, including duplicate de- tection. By fine-tuning BERT on project-related data, we can achieve high precision in identifying duplicate project topics.

C. Applications in Project Management Systems

Project Management Systems (PMS) in academic settings have seen a rise in adoption for managing and

tracking student projects. Sulaiman et al. proposed a web-based PMS that assists students in managing their final-year projects, including features like project submission, tracking, and feedback [16]. However, none of the existing systems specifically address the issue of duplicate project topics, which often leads to a lack of originality in project proposals.

To address this gap, our proposed system integrates dupli- cate topic detection into the PMS. The system leverages NLP and machine learning techniques to analyze project descrip- tions and detect duplicates. Our approach combines TF-IDF vectorization, XGBoost classifiers, and cosine similarity to determine the uniqueness of project topics, providing students with valuable feedback to help them choose novel and original project ideas.

D. Plagiarism Detection and Similarity Scoring

Plagiarism detection tools are often employed to ensure that the content submitted by students is original. While plagiarism detection is crucial for academic integrity, it is distinct from the task of detecting duplicate topics, which requires understanding the underlying semantic meaning of text rather than simple word matching. Mikolov et al. ex-plored word embeddings, which capture semantic relationships between words, and are widely used in various NLP tasks, including plagiarism detection [11]. Word embeddings can be integrated into duplicate topic detection systems to improve the detection of semantically similar but lexically distinct project descriptions.

Our system builds on these concepts by incorporating semantic understanding into the duplicate detection process, enabling it to identify duplicates even when there are variations in the wording or phrasing of project descriptions.

E. Project Management System

Project Management Systems have been developed in aca- demic environments to streamline the monitoring and eval- uation of final year student projects. Isa et al. proposed a system that enables supervisors to track student progress and performance in real time, aiming to improve coordination and reduce communication gaps between stakeholders [1]. Rajesh et al. designed a comprehensive project management system tailored to handle project proposal submission, ap- proval workflows, and documentation management within en-gineering institutions [2]. Soms et al. introduced a system that facilitates student-supervisor interaction, automates deadline tracking, and maintains evaluation records to ensure timely completion of project milestones [3]. Jain et al. implemented a lightweight PMS focused on simplifying the submission and review process of project documentation, offering a user-friendly interface for both students and faculty [4]. These systems primarily address the administrative aspects of project tracking, while future work can focus on integrating intelligent features such as topic duplication detection and semantic analysis.

F. Summary Table

TABLE I
LITERATURE SURVEY ON DUPLICATE TOPIC DETECTION AND PROJECT MANAGEMENT SYSTEMS

Reference	Key Contribution	Techniques/Methods Used
[18]	Introduced vector space model for automatic indexing.	Cosine similarity, TF-IDF
[12]	Survey of text clustering algo-rithms.	Clustering, Similarity-based tech-niques
[6]	Transformer architecture for NLP tasks.	Self-attention mechanism
[7]	BERT for deep language under-standing.	Pre-trained transformers
[21]	LDA for topic modeling.	Topic modeling, Latent variables
[11]	Word2Vec for semantic analysis.	Word embeddings
[16]	Web-based student project man-agement system.	Web application, Project tracking
[17]	Hybrid knowledge-based system for software selection.	Decision-making systems
[8]	Scikit-learn machine learning li-brary.	ML algorithms in Python
[9]	XGBoost scalable tree boosting method.	Gradient boosting
[19]	Models for information retrieval.	Text retrieval, Search models
[20]	Principles of speech and language processing.	NLP, Speech processing

IV. METHODOLOGY

The proposed system is a Project Management System (PMS) for academic institutions, featuring a module for detect- ing duplicate project topics. It streamlines project supervision, student submissions, and ensures originality by integrating software engineering practices with modern NLP techniques. Students submit project proposals with details such as title, description (including technology used), domain. Faculty can review submissions, track progress, and manage student groups. The Duplicate Topic Detection engine is manually triggered by the coordinator to check for repeated ideas and foster innovation.

The proposed system for detecting duplicate final-year project topics involves a sequence of structured steps that transform raw project submissions into meaningful features, apply machine learning for classification, and generate insights for decision-making. The methodology consists of the follow- ing key stages:

A. Data Collection and Feature Construction

A dataset comprising final-year project records was col- lected from academic archives spanning the last three years. Each record included the following attributes: Project Title, Project Description, Domain, and Technologies Used.
To increase the semantic richness and give more weight to the project title, it was repeated twice and concatenated with the other fields to form a single combined text input. This unified textual input allowed the model to better understand the context and intent behind each project.

B. Text Preprocessing

Text preprocessing was performed with the following steps:

- Converting all text to lowercase
- Removing URLs, special characters, and numeric values
- Tokenizing the text into individual words
- Removing common English stopwords (e.g., ”the”, ”is”, ”and”)
- Applying lemmatization to reduce words to their root form (e.g., ”running” → ”run”)

This cleaned and normalized text helped reduce noise and improved the quality of feature extraction.

C. Feature Extraction with TF-IDF and Dimensionality Reduction

The processed text was converted into numerical vectors using the Term Frequency-Inverse Document Frequency (TF- IDF) technique. The TF-IDF value for each term t in a document d was calculated as:

$$\text{TF-IDF}(t, d) = \text{TF}(t, d) \times \text{IDF}(t) \quad (1)$$

where:

- TF(t, d) is the term frequency of term t in document d,
- IDF(t) is the inverse document frequency of term t across the entire corpus.

To capture more contextual meaning, n-grams (up to tri- grams) were included, and a maximum of 7750 features was set for optimal performance.

Since high-dimensional data can lead to overfitting and increased computation time, dimensionality reduction was ap- plied using Truncated Singular Value Decomposition (SVD). The SVD decomposes the TF-IDF matrix A into three matri- ces:

$$A = U\Sigma V^{T} \quad (2)$$

where U and V are orthogonal matrices and Σ is a diagonal matrix containing singular values.

By retaining only the top 350 components, the most infor- mative features were preserved.

D. Generating Labels with Cosine Similarity

To generate labels for supervised learning, a similarity matrix was computed using cosine similarity over the TF-IDF vectors. The cosine similarity between two vectors $\vec{A}$ and $\vec{B}$ is given by:

$$\text{cosine\_similarity}(\vec{A}, \vec{B}) = \frac{A \cdot B}{||\vec{A}|| \times ||\vec{B}||} \quad (3)$$

If the maximum similarity between a project and any other project exceeded a predefined threshold (0.67), it was labelled as a duplicate (1); otherwise, it was considered unique (0).

E. Model Training and Evaluation

The labelled dataset was split into training and testing sets using an 80:20 stratified split to maintain class balance. The XGBoost classifier was chosen for model training due to its robustness and speed.

The XGBoost model optimizes the following objective function:

$$\text{Obj}(\theta) = \sum_{i=1}^{n} l(y_i, \hat{y}_i) + \sum_{k=1}^{K} \Omega(f_k) \quad (4)$$

where:

- l is the loss function (e.g., logistic loss for classification),
- Ω(f) is the regularization term controlling model com- plexity,
- K is the number of trees.

The model was evaluated using the following metrics:

- **Accuracy**:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \quad (5)$$

- **Precision**:

$$\text{Precision} = \frac{TP}{TP + FP} \quad (6)$$

- **Recall**:

$$\text{Recall} = \frac{TP}{TP + FN} \quad (7)$$

- **F1-Score**:

$$F1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (8)$$

where TP, TN, FP, and FN represent true positives, true negatives, false positives, and false negatives respectively.

Additionally, ROC AUC Score and Confusion Matrix visu- alization were used to further assess model performance.

F. Prediction and Similarity Scoring

A custom function was created to take a new project input (title, description, domain, technologies) and:

- Preprocess the input
- Transform it using the same TF-IDF and SVD pipeline
- Use the trained model to predict whether it is a duplicate
- Calculate its maximum cosine similarity with existing projects

This dual output (classification + similarity score) offers both a binary prediction and a transparent ”plagiarism meter” style score for users and evaluators.

V. SYSTEM ARCHITECTURE

The proposed Project Management System (PMS) with Duplicate Topic Detection is designed to be modular, scalable, and user-centric. It comprises three main layers: frontend, backend, and machine learning module, ensuring separation of concerns, ease of maintenance, and seamless integration.

1. Frontend:

Developed using React.js, the frontend offers a user-friendly interface for students and faculty. It supports features like project submission, tracking, feedback ex- change, and real-time status updates. Upon project sub- mission, the frontend sends data to the backend via RESTful APIs.

2. Backend:

The backend is built with Node.js and Express.js, han- dling data validation, authentication, routing, and inter- action with the machine learning module. MongoDB stores user profiles, project submissions, group infor- mation, and project history. Role-based access control ensures data segregation and security for students, fac- ulty, and administrators.

3. Machine Learning Module:

The ML module, integrated with Python and accessed via a REST API, performs the following tasks:

- **Preprocessing:** Cleans and lemmatizes the text data (title, description, domain, technologies).
- **Feature Extraction:** Converts text into numerical vectors using TF-IDF and n-grams.
- **Dimensionality Reduction:** Applies SVD for com-
- **Duplicate Detection:** The XGBoost classifier de- termines if a project is a duplicate.
- **Similarity Scoring:** Calculates cosine similarity to generate a percentage score (plagiarism meter).
- **Response Output:** Provides a categorical label (”Duplicate” or ”Unique”) and similarity percentage to the backend.

4. Integration and Workflow:

When a student submits a project, the backend sends the data to the ML module, which processes it and returns the prediction and similarity score. This result is displayed to the student and guide through the frontend, helping them decide whether to proceed or modify the topic.

This architecture ensures the system is robust, responsive, and effective in managing projects while maintaining original- ity through intelligent duplicate detection.

VI. WORKFLOW

The end-to-end workflow of the proposed Project Manage- ment System (PMS) with Duplicate Topic Detection is as follows:

- **User Registration and Authentication:**
  Students and faculty members register and log in through the frontend application. The backend authenticates users and manages role-based access control.
- **Project Submission:**
  A student group submits a project proposal, including the project title, description, domain, and technologies used, through the frontend interface.
- **Data Transmission to Backend:**
  The submitted project data is securely transmitted to the backend server via RESTful APIs. The backend validates the input and formats it for analysis.
- **Preprocessing and Feature Engineering:**
  The backend forwards the project data to the Machine Learning (ML) module. Text preprocessing (tokenization, lemmatization) is performed, followed by feature extraction using TF-IDF vectorization and n-gram generation.
- **Duplicate Detection and Similarity Computation:**
  The ML module applies dimensionality reduction (SVD) to the feature set for efficiency. An XGBoost classifier predicts whether the project is a duplicate. Cosine similarity scoring is also calculated to quantify the degree of similarity.
- **Result Transmission to Backend:**
  The ML module sends back the duplicate prediction (Duplicate/Unique) along with a similarity percentage score.
- **Result Visualization:**
  The backend communicates the results to the frontend, where students and faculty can view the prediction and similarity score through a visual plagiarism meter.
- **Feedback and Decision:**
  Based on the duplicate detection result, students can either proceed with the project submission or revise and resubmit the proposal.

This workflow ensures smooth interaction between users and the system, providing timely and accurate duplicate topic detection while maintaining an intuitive user experience.

VII. SYSTEM WORKFLOW DESCRIPTION

The overall workflow of the proposed Project Management System (PMS) with Duplicate Topic Detection is explained as follows:

- **User Registration & Authentication**: Students, faculty, and administrators register and log in to the system us- ing secure authentication mechanisms. Role-based access control ensures appropriate access privileges.
- **Project Submission**: Students submit their project title, abstract/description, domain, and technologies via the frontend interface.
- **Data Validation & Transmission**: The backend validates the input data for completeness and correctness before transmitting it to the machine learning module through RESTful APIs.
- **Preprocessing & Feature Extraction**: The machine learning module preprocesses the text data by removing stopwords, performing lemmatization, and converting the cleaned text into numerical vectors using TF-IDF and n- gram modeling.
- **Duplicate Detection (XGBoost + Cosine Similarity)**: The processed vectors are passed to a trained XGBoost classifier which predicts whether the project submission is a duplicate or unique. Cosine similarity is also computed to provide a similarity percentage score.
- **Prediction & Similarity Score**: The machine learning module sends back the prediction (Duplicate/Unique) along with the similarity score to the backend server.
- **Result Visualization**: The frontend displays the predic- tion outcome and similarity score to the student and guide, aiding them in decision-making.
- **Feedback & Decision**: Based on the system's feedback, students can either proceed with their submission or modify their project details to ensure uniqueness.

VIII. EXPERIMENTAL SETUP

The experimental setup for evaluating the proposed Project Management System (PMS) with Duplicate Topic Detection is detailed below:

A. Software and Tools

- **Frontend**: Developed using React.js with Axios for API integration and Material-UI for UI components.
- **Backend**: Implemented using Node.js and Express.js, with MongoDB Atlas as the cloud-based database.
- **Machine Learning Module**: Built using Python 3.10, employing libraries such as Scikit-learn, XGBoost, NLTK, SVM and RandomForest.

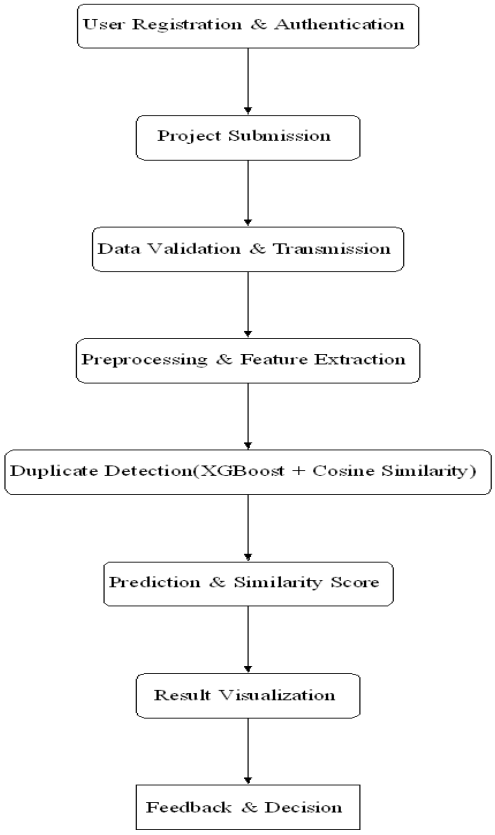

Fig. 1. Workflow of Project Management System with Duplicate Topic Detection

B. Dataset

A custom dataset comprising 500+ final-year project sub- missions was created, including fields such as Project Title, Description, Domain, Technologies, and a Duplicate Label (0 for unique, 1 for duplicate). Additional augmentation was performed to simulate real-world variations in project topics.

C. Preprocessing Techniques

- Removal of stopwords and punctuation.
- Lowercasing and lemmatization using NLTK.
- Feature extraction using TF-IDF vectors with unigrams, bigrams, and trigrams.
- Dimensionality reduction using Truncated Singular Value Decomposition (SVD).

D. Model Training

To classify whether a project topic is unique or duplicate, three machine learning models were trained: Support Vector Machine (SVM), Random Forest, and XGBoost. The input data was pre-processed using techniques such as tokenization, stopword removal, lemmatization, and enhanced with domain- specific alternative terms.

After this, TF-IDF was used for vectorization, followed by Truncated SVD for dimensionality reduction.

Due to class imbalance in the dataset (fewer duplicate topics compared to unique ones), class weights were calculated and applied during training, especially in the XGBoost model. This helped the model learn from the minority class more effectively.

Basic hyperparameter tuning was performed on XGBoost with 250 estimators, a learning rate of 0.06, and a maximum depth of 6. Among all models, XGBoost achieved the highest accuracy (90) along with balanced precision and recall scores. Hence, XGBoost was selected as the final model for deploy- ment in the project topic detection system.

E. Hardware Environment

The experiments were carried out on a system with the following specifications:

- Processor: Intel Core i5, 10th Gen
- RAM: 8 GB
- Storage: 256 GB SSD
- Operating System: Windows 10

IX. RESULTS AND EVALUATION

To evaluate the effectiveness of the proposed Duplicate Project Topic Detection system, three popular machine learn- ing models were implemented and tested: **Support Vector Machine (SVM)**, **Random Forest**, and **Extreme Gradient Boosting (XGBoost)**. The dataset was split using an 80:20 ratio for training and testing, and the performance of each model was assessed using standard classification metrics: *Accuracy*, *Precision*, *Recall*, and *F1-Score*.

1. *Evaluation Metrics*

- **Accuracy**: The proportion of correctly predicted in- stances out of the total samples.
- **Precision**: The proportion of true positive predictions among all predicted positives.
- **Recall**: The proportion of true positive predictions among all actual positives.

- **F1-Score**: The harmonic mean of precision and recall, providing a balanced measure.

2. *Performance Comparison*

Model	Accuracy	Precision	Recall	F1-Score
SVM	0.89	0.89	0.89	0.89
Random Forest	0.87	0.87	0.87	0.87
XGBoost	**0.90**	**0.90**	**0.90**	**0.90**

TABLE II
Performance Comparison of Models

The experimental results indicate that XGBoost is the most effective model for detecting duplicate project topics, owing to its robust handling of feature interactions and class imbalance.

3. *Detailed Model Evaluation*

Support Vector Machine (SVM):

- Precision: 0.92 (Class 0), 0.86 (Class 1)
- Recall: 0.87 (Class 0), 0.92 (Class 1)
- F1-Score: 0.89 (both classes)
- Accuracy: 89%

SVM performed consistently across classes and showed good recall in detecting duplicate topics.

Random Forest:

- Precision: 0.85 (Class 0), 0.89 (Class 1)
- Recall: 0.90 (Class 0), 0.83 (Class 1)
- F1-Score: 0.88 (Class 0), 0.86 (Class 1)
- Accuracy: 87%

Random Forest performed slightly lower, especially in re-calling duplicate topics.

XGBoost:

- Precision: 0.93 (Class 0), 0.87 (Class 1)
- Recall: 0.89 (Class 0), 0.91 (Class 1)
- F1-Score: 0.91 (Class 0), 0.89 (Class 1)
- Accuracy: 90%

XGBoost demonstrated the highest overall performance across all metrics.

Based on the evaluation results, **XGBoost was selected as the final model** for deployment. It achieved the highest accuracy and exhibited balanced performance between both classes. The model's ability to handle class imbalance and capture complex feature interactions made it the most suitable choice for the Duplicate Project Topic Detection system.

X. CONCLUSION

In this research, we presented a comprehensive Project Man- agement System that integrates an intelligent Duplicate Project Topic Detection mechanism using natural language processing and machine learning. The system addresses a prevalent issue in academic institutions — the repetition of project ideas — and offers a scalable, user-friendly, and efficient solution for maintaining originality in student submissions.

The system successfully combines TF-IDF vectorization, di- mensionality reduction using SVD, and an XGBoost classifier to identify topic duplications with high accuracy and confi- dence. Inclusion of cosine similarity scores further enhances transparency, enabling users and evaluators to understand the reasoning behind duplicate or unique classifications. This makes the system not only a decision maker but also an assistant in guiding project originality.

Furthermore, by embedding this detection mechanism within a broader project management framework, the tool becomes more than a classifier. It evolves into a comprehensive platform that facilitates project allocation, status tracking, domain tagging, and historical project archiving, which are crucial for streamlined academic administration.

XI. FUTURE SCOPE

Future Work will focus on several key enhancements:

- Semantic Deep Learning Integration: Incorporating pre- trained models like BERT can provide deeper contextual understanding and improve duplicate detection, especially in paraphrased scenarios.
- Multilingual Support: Expanding the system to understand and classify projects in regional languages can broaden its reach in diverse academic environments.
- Real-Time Feedback: Enhancing the UI/UX to allow students to receive immediate suggestions or alerts about possible duplications at the time of topic submission.

- Faculty Dashboard: Providing faculty with insights into topic trends, originality scores, and domain-specific workload distribution.

The proposed system demonstrates that merging project management tools with intelligent text analysis not only im- proves administrative efficiency but also cultivates a culture of innovation and intellectual integrity in academia. This blend of automation and insight paves the way for more robust, fair, and impactful project evaluation ecosystems.

XII. ACKNOWLEDGMENT

The authors would like to thank Dr. Amruta Surana, Prof. Karim Mulani, and Prof. Santosh Dabade for their valuable guidance and continuous support throughout the course of this project. The authors also acknowledge the Department of Computer Engineering, Sinhgad Institute of Technology, for providing the necessary facilities and environment for the successful completion of this work.

REFERENCES

[1] R. Isa, S. Othman, A. S. Ali, N. Azizan, and J. Ferguson, "Prototype development of final year project management system to monitor stu- dents' performance," *Journal of Advanced Research in Applied Sciences and Engineering Technology*, 2024.

[2] K. Rajesh, C. S. Kumar, B. P. Kalyan, and G. Vani, "Another college project management system," *International Research Journal of Engi- neering and Technology*, 2023.

[3] N. Soms, S. Prashanth, P. Preethika, D. D. Kumar, and G. Vani, "Student project management system," *International Journal of All Research Education and Scientific Methods*, 2021.

[4] A. Jain, S. Kolabkar, K. Netke, and P. Mate, "Project management system (PMS)," *International Research Journal of Engineering and Technology*, 2020.

[5] U. C. Chikwendu and M. A. Eziechina, "Web-based student project management system: A Tetfund institution based research report," *In- ternational Journal of Current Science Research and Review*, 2021.

[6] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, *et al.*, "Attention is all you need," in *Advances in Neural Information Processing Systems*, pp. 5998-6008, 2017.

[7] J. Devlin, M. W. Chang, K. Lee, and K. Toutanova, "BERT: Pre- training of deep bidirectional transformers for language understanding," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics*, 2019.

[8] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, *et al.*, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825-2830, 2011.

[9] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2016.

[10] S. Bird, E. Klein, and E. Loper, *Natural Language Processing with Python: Analyzing Text with the Natural Language Toolkit*, O'Reilly Media, 2009.

[11] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their composi- tionality," in *Advances in Neural Information Processing Systems*, 2013.

[12] C. C. Aggarwal and C. Zhai, "A survey of text clustering algorithms," in *Mining Text Data*, Springer, pp. 77–128, 2012.

[13] Scikit-learn documentation, [Online]. Available: https://scikit-learn.org/.

[14] OpenAI, "GPT models and language understanding," [Online]. Avail- able: https://openai.com/research/.

[15] IEEE Xplore Digital Library, "Research papers on duplicate detection in document repositories and academic datasets," [Online]. Available: https://ieeexplore.ieee.org/.

[16] N. Sulaiman, F. Karim, and M. H. Ismail, "A web-based final year project management system," in *2014 International Conference on Computer, Communications, and Control Technology (I4CT)*, IEEE, 2014.

[17] A. Jadhav and R. Sonar, "Framework for evaluating and selecting software packages: A hybrid knowledge-based system," *Information and Software Technology*, vol. 51, no. 3, pp. 639-648, 2011.

[18] G. Salton, A. Wong, and C. S. Yang, "A vector space model for automatic indexing," *Communications of the ACM*, vol. 18, no. 11, pp. 613–620, 1975.

[19] C. D. Manning, P. Raghavan, and H. Schu¨tze, *Introduction to Informa- tion Retrieval*, Cambridge University Press, 2008.

[20] D. Jurafsky and J. H. Martin, *Speech and Language Processing* (3rd ed.), Draft version, Stanford University, 2020.

[21] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent Dirichlet Allocation,"*Journal of Machine Learning Research*, vol. 3, pp. 993–1022, 2003.

[22] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[23] Y. Zhang, R. Jin, and Z. H. Zhou, "Understanding bag-of-words model: A statistical framework," *International*

Journal of Machine Learning and Cybernetics, vol. 1, no. 1-4, pp. 43–52, 2010.

[24] R. L. Figueroa, Q. Zeng-Treitler, S. Kandula, and L. H. Ngo, "Predicting sample size required for classification performance," *BMC Medical Informatics and Decision Making*, vol. 12, no. 1, pp. 1–10, 2012.

[25] Kaggle, "Project Topic Detection datasets and competitions," [Online]. Available: https://www.kaggle.com/.

[26] UCI Machine Learning Repository, "Text classification datasets," [On- line]. Available: https://archive.ics.uci.edu/ml/index.php.

[27] Gensim Documentation, "Topic Modeling for Humans," [Online]. Avail- able: https://radimrehurek.com/gensim/.

[28] P. N. Tan, M. Steinbach, and V. Kumar, *Introduction to Data Mining*(2nd ed.), Pearson, 2018.