RAJALAKSHMI ENGINEERING COLLEGE
An AUTONOMOUS Institution
Affiliated to ANNA UNIVERSITY, Chennai

25 years OF ACADEMIC EXCELLENCE

Department of Computer Science and Engineering

SKILLFORGE: YOUR CODING COMPANION

Mrs. M. Divya M.E.
Project Mentor

NANDITHA N
220701182

Problem Statement and Motivation

Learners on coding platforms often struggle to assess whether a problem matches their skill level, while educators face difficulties in consistently tagging and categorizing problems. Manual classification is time-consuming and subjective. To address this, we propose an automated system that:

1. Predicts the required skill level for a problem,
2. Auto-tags it with relevant algorithmic concepts, and
3. Recommends a personalized learning path.

This solution aims to support adaptive learning and streamline content delivery on coding platforms.

Existing System

- Most coding platforms (e.g., LeetCode, HackerRank) rely on **manual tagging** and **subjective difficulty levels** assigned by content creators or user votes.
- **Skill recommendations are generic**, often not tailored to the learner's journey.
- No unified mechanism to **auto-analyze problem statements** to extract underlying concepts or difficulty.
- Learners spend extra time figuring out which problems suit their current skill set.
- Educators face challenges in **batch-tagging and organizing problems** for targeted learning.

Objectives

- **Predict the skill level** required to solve a coding problem using machine learning.
- **Automatically tag problems** with relevant data structures and algorithms based on the problem statement.
- **Explain the logic** behind skill level prediction for transparency and trust.
- **Suggest personalized learning paths** based on predicted skill level and identified tags.
- Assist educators and learners in **curating structured problem sets** for different learning stages.
- Reduce time spent manually classifying or selecting problems for skill development.

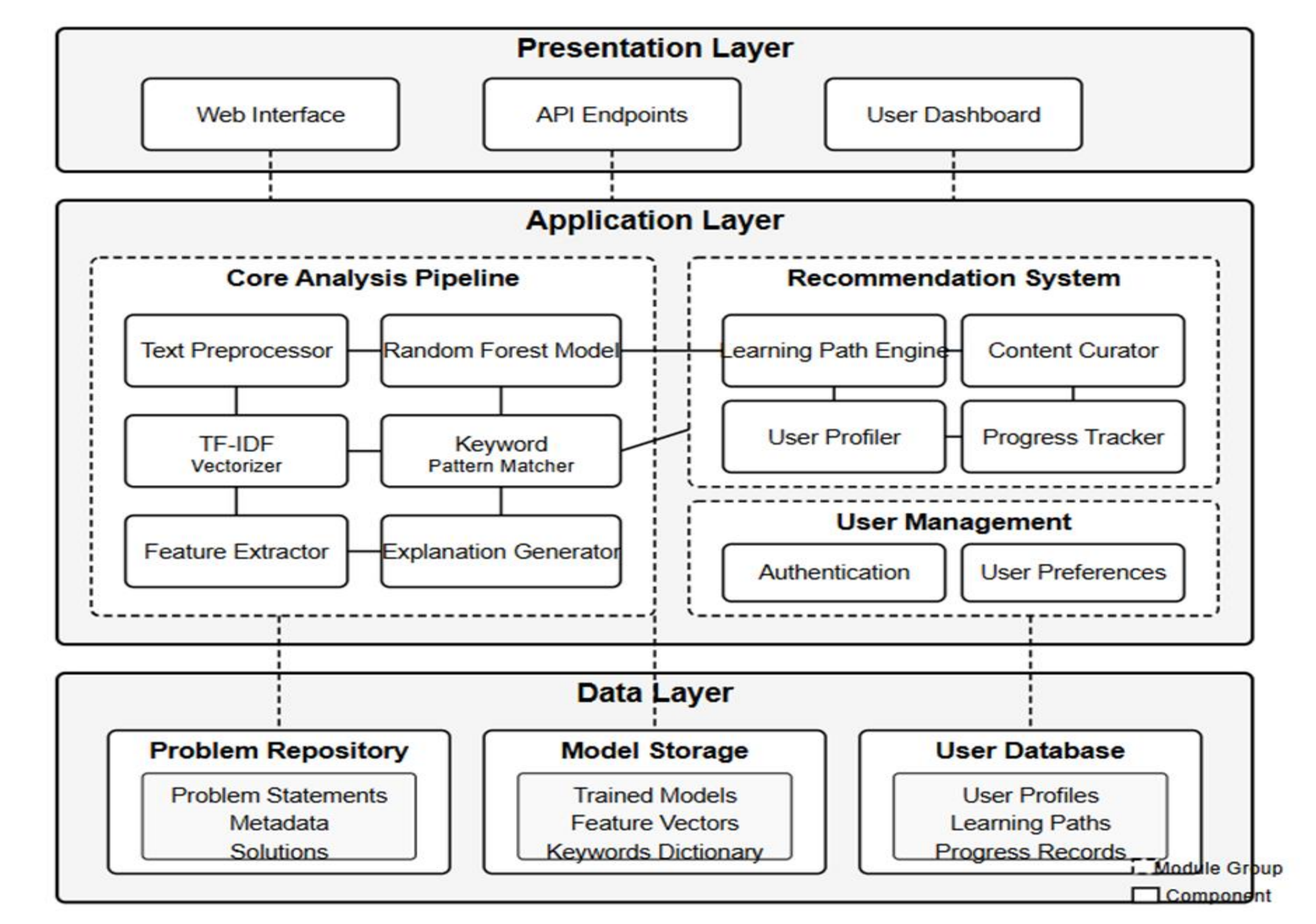
Abstract

This project presents a machine learning-based system that predicts the skill level required to solve a coding problem by analyzing its title, description, difficulty, and score. It leverages TF-IDF for text vectorization and a Random Forest Classifier for classification. In addition to skill prediction, the system auto-tags each problem with relevant algorithmic concepts such as Dynamic Programming, Graphs, or Trees. It also explains the rationale behind predictions and recommends customized learning paths based on the tags and predicted difficulty. The overall aim is to assist learners, educators, and online platforms in organizing, understanding, and navigating coding problems more efficiently, thereby enhancing the learning experience and supporting adaptive skill development.

Proposed System

- The proposed system uses a combination of Natural Language Processing (NLP) and Machine Learning techniques to automatically analyze and classify coding problems. The process begins by cleaning and combining the problem name and description, followed by vectorization using TF-IDF to extract meaningful features. Categorical fields like difficulty and skill level are label-encoded, and a Random Forest Classifier is trained on this processed data to predict the required skill level for new problems.
- Additionally, the system performs auto-tagging by scanning the problem text for keywords linked to specific algorithms or data structures. It provides human-readable explanations for predictions and suggests learning paths tailored to the problem's content and predicted skill level. This makes the platform intelligent, personalized, and effective for both learners and educators.

System Architecture



Presentation Layer

Web Interface

API Endpoints

User Dashboard

Application Layer

Core Analysis Pipeline

Text Preprocessor

Random Forest Model

TF-IDF Vectorizer

Keyword Pattern Matcher

Feature Extractor

Explanation Generator

Recommendation System

Learning Path Engine

Content Curator

User Profiler

Progress Tracker

User Management

Authentication

User Preferences

Data Layer

Problem Repository

Problem Statements
Metadata
Solutions

Model Storage

Trained Models
Feature Vectors
Keywords Dictionary

User Database

User Profiles
Learning Paths
Progress Records

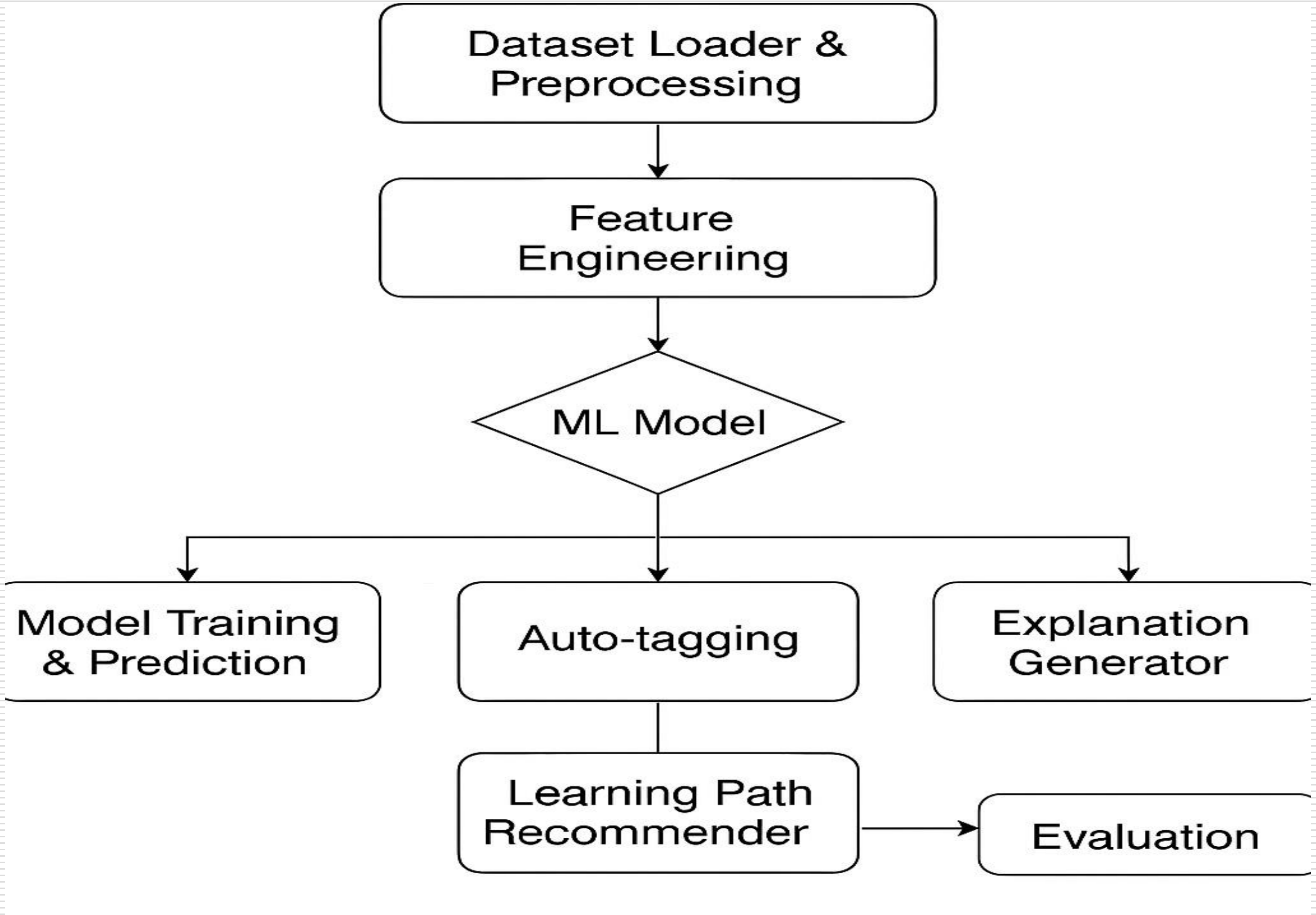
Module Group

Component

List of Modules

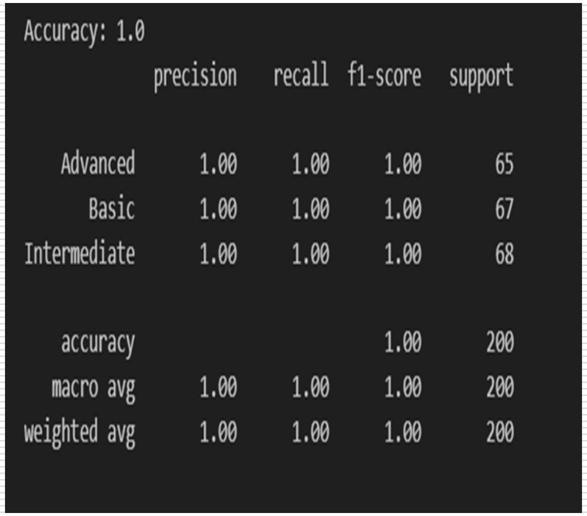
- □ Dataset Loader & Preprocessing Module
- □ Feature Engineering Module
- □ Model Training & Prediction Module
- □ Auto-tagging Module
- □ Explanation Generator
- □ Learning Path Recommender
- □ Performance Evaluation Module

Functional Description for each modules with DFD and Activity Diagram


Dataset Loader & Preprocessing
Feature Engineeriing
ML Model
Model Training & Prediction
Auto-tagging
Explanation Generator
Learning Path Recommender
Evaluation

Implementation & Results of Module

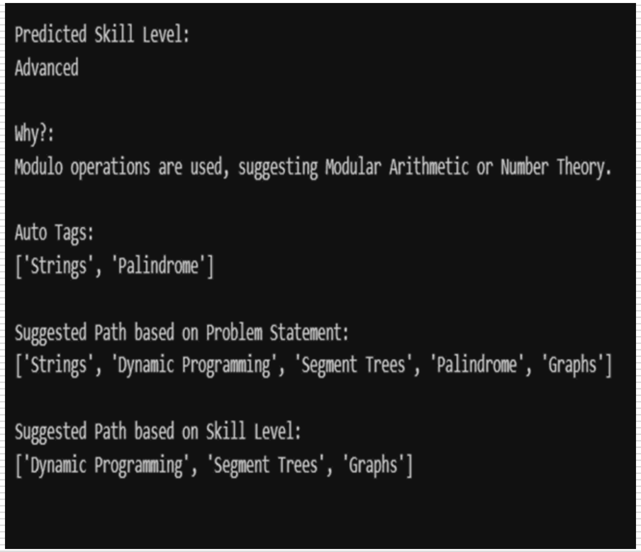
- The dataset was cleaned and preprocessed by removing null values.
- Text data was vectorized using TF-IDF to extract meaningful features.
- Label encoding was used to convert categorical values to numerical format.
- A Random Forest Classifier was trained to predict skill levels of problems.
- The model achieved good accuracy and generated tags with suggested learning paths.



```
Accuracy: 1.0
              precision    recall  f1-score   support

    Advanced       1.00      1.00      1.00        65
       Basic       1.00      1.00      1.00        67
Intermediate       1.00      1.00      1.00        68

    accuracy                           1.00       200
   macro avg       1.00      1.00      1.00       200
weighted avg       1.00      1.00      1.00       200
```



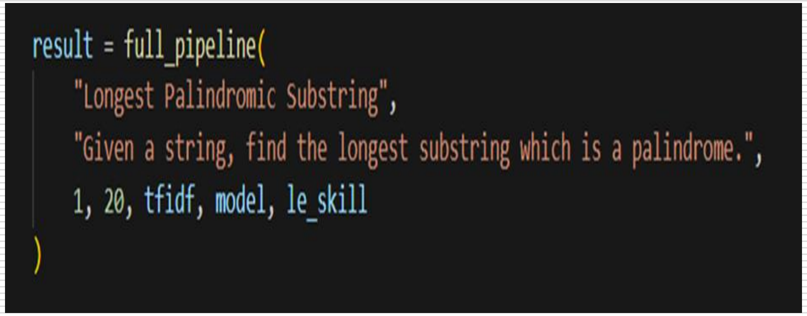
```
Predicted Skill Level:
Advanced

Why?:
Modulo operations are used, suggesting Modular Arithmetic or Number Theory.

Auto Tags:
['Strings', 'Palindrome']

Suggested Path based on Problem Statement:
['Strings', 'Dynamic Programming', 'Segment Trees', 'Palindrome', 'Graphs']

Suggested Path based on Skill Level:
['Dynamic Programming', 'Segment Trees', 'Graphs']
```



```
result = full_pipeline(
    "Longest Palindromic Substring",
    "Given a string, find the longest substring which is a palindrome.",
    1, 20, tfidf, model, le_skill
)
```

Conclusion & Future Work

Conclusion:

The system effectively predicts the skill level required to solve programming problems using machine learning. It also auto-tags the problem with relevant topics and suggests a personalized learning path, aiding both learners and educators.

Future Work:

- Integrate code analysis for deeper context understanding.
- Enhance tagging using advanced NLP models like BERT.
- Support multilingual problem statements.
- Incorporate user feedback for continual model improvement.
- Deploy as a web-based platform for wider accessibility.

References

•Aggarwal, C. C. (2018). *Machine Learning for Text*. Springer.
•Vaswani, A. et al. (2017). “Attention is All You Need.” *Advances in Neural Information Processing Systems*.
•Mikolov, T. et al. (2013). “Efficient Estimation of Word Representations in Vector Space.” *arXiv preprint arXiv:1301.3781*.
•Le, Q. V., & Mikolov, T. (2014). “Distributed Representations of Sentences and Documents.” *Proceedings of the 31st International Conference on Machine Learning*.
•Singh, A. et al. (2020). “Automated Difficulty Level Prediction of Programming Problems.” *International Journal of Computer Applications*, 176(8), 1-5.

Thank You