AutoJudge:
Programming Problem Difficulty Prediction

Project Report:

Name: Devansh Pandey

Enrollment Number: 23113049

Branch: Civil Engineering, 3rd Year

This report presents the design and implementation of an automated system to predict the difficulty level of competitive programming problems using machine learning techniques.

1. Introduction and Problem Statement

Competitive programming platforms host thousands of problems with varying difficulty levels. Manually assigning difficulty labels is subjective and time-consuming. This project aims to automate difficulty prediction using machine learning by analyzing problem statements and associated metadata.

2. Dataset Description

The dataset used in this project is provided in JSON Lines (JSONL) format and consists of programming problems scraped from online judges. Each record includes the problem title, description, input/output specifications, sample input-output examples, difficulty class (Easy, Medium, Hard), and a numerical difficulty score.

3. Data Preprocessing

Textual fields are cleaned and combined for feature extraction. Numerical features are scaled using standard normalization. The dataset is split into training and testing subsets to evaluate model generalization.

4. Feature Engineering

Feature engineering is implemented in feature_utils.py. TF-IDF vectorization is applied to textual data to capture keyword importance, while handcrafted numerical features represent structural characteristics such as text length and numeric density.

5. Machine Learning Models

Two Random Forest models are employed. The classification model predicts discrete difficulty classes, while the regression model predicts a continuous difficulty score. Both models are implemented using scikit-learn pipelines to ensure consistent preprocessing during inference.

6. Experimental Setup

The dataset is divided into training and testing sets with an 80-20 split. Models are trained on the training set and evaluated on the test set using standard performance metrics.

7. Evaluation Metrics and Results

Metric	Exact Value
Classification Accuracy	54.19%
Mean Absolute Error (MAE)	1.696
Root Mean Square Error (RMSE)	2.043

The classification accuracy reflects the challenging and subjective nature of difficulty prediction. The confusion matrix analysis shows that the model performs best on Medium difficulty problems, which dominate the dataset. Regression metrics indicate acceptable error margins for predicting difficulty scores.

8. Web Interface and Sample Predictions

A Flask-based web interface allows users to enter problem details and obtain instant difficulty predictions. The screenshots below demonstrate the system behavior for Easy and Hard problems during local execution.

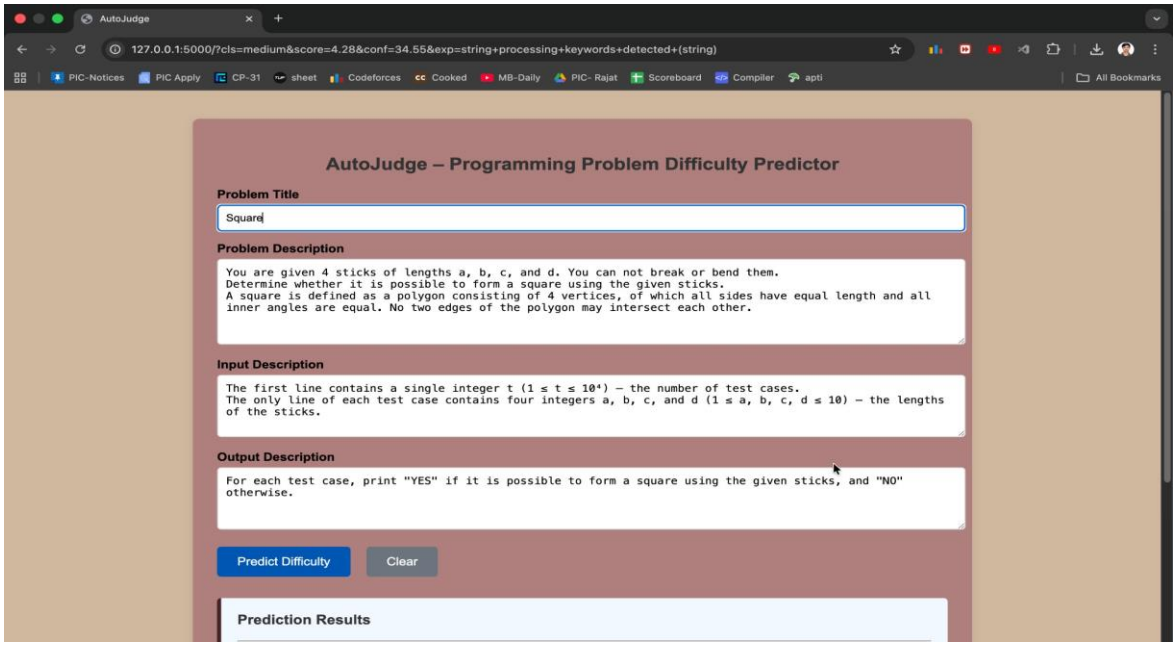

Figure 1: Easy problem input (Square problem)

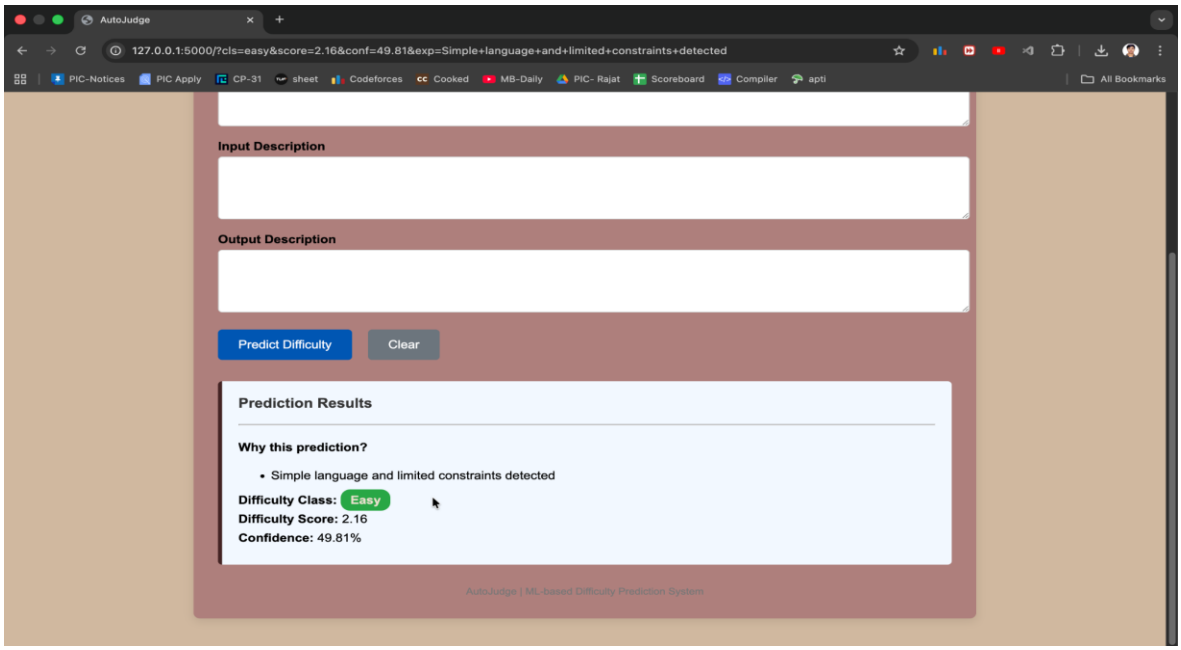

Figure 2: Easy problem prediction result

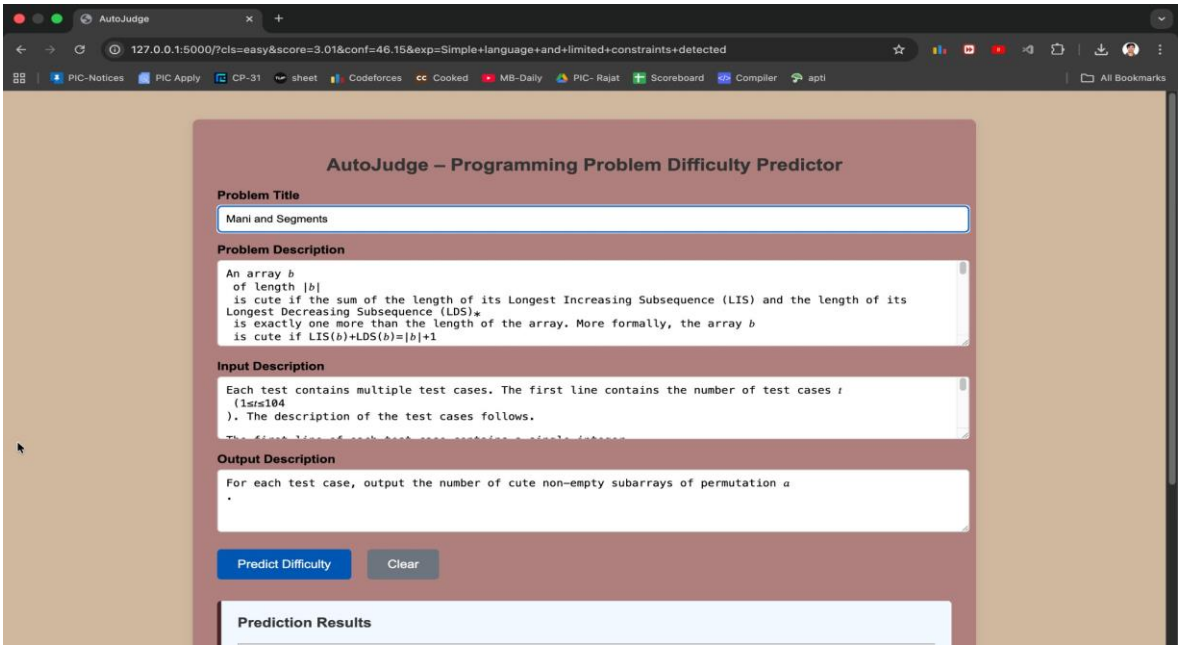

Figure 3: Hard problem input (Mani and Segments)

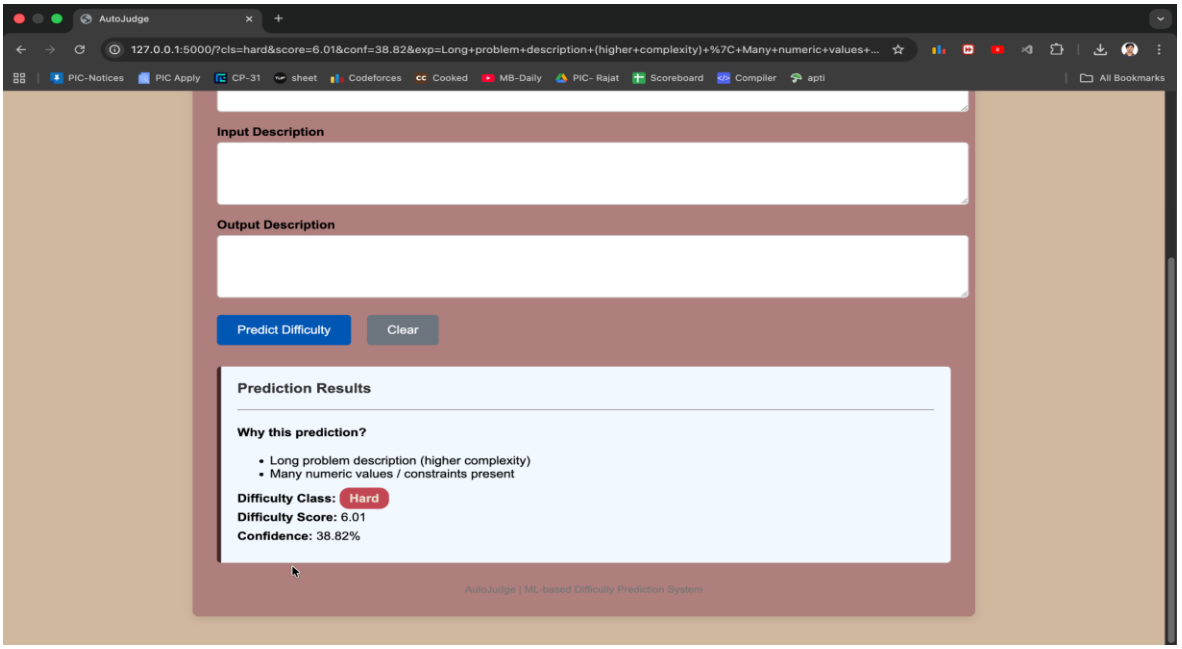

Figure 4: Hard problem prediction result

9. Discussion

The results indicate that textual complexity and numeric constraints significantly influence difficulty prediction. Confusion between Medium and Hard categories arises due to overlapping problem characteristics. Despite moderate accuracy, the system demonstrates reliable behavior for practical use cases.

10. Conclusion and Future Work

This project successfully implements an end-to-end machine learning pipeline for predicting programming problem difficulty. Future work may explore dataset balancing, transformer-based language models, and advanced feature representations to improve performance.