AI BASED QUESTION LEVEL CLASSIFICATION SYSTEM

A Final Year Project Report

submitted to the HALDIA INSTITUE OF TECHNOLOGY

in partial fulfillment of requirement of the award of degree

Bachelor of Technology

In

Information Technology

Submitted by:

Gurpreet Singh (Roll No. - 10300222003)

M. Sohan Kumar (Roll No. - 10300222005)

Utsab Roy (Roll No. - 10300222012)

Mohit Kumar Singh (Roll No. - 10300222006)

Under the Guidance of **Mr. Susanta Banerjee**

Head of Department: **Dr. Soumen Paul**

Department of Information Technology
Haldia Institute of Technology

West Bengal
2025

CERTIFICATE: -

This is to certify that the project report entitled ' **AI Based Question Level Classification System**' submitted by Gurpreet Singh (L22/IT/137), M. Sohan Kumar (L22/IT/131), Utsab Roy (L22/IT/134), and Mohit Kumar Singh (L22/IT/132) to Haldia Institute of Technology in partial fulfillment of the requirements for the award of the degree of Bachelor of Technology in Information Technology is a Bonafide record of the project work carried out under our guidance and supervision.

Prof. Dr. Soumen Paul
Head of the Department
Haldia Institute of Technology
Department of Information Technology

Mr. Susanta Banerjee
Assistant Professor
Haldia Institute of Technology
Department of Information Technology

DECLARATION: -

We hereby declare that the project report titled **"AI Based Question Level Classification System"** submitted for partial fulfilment of the requirements for the award of **Bachelor of Technology in Information Technology** of **Haldia Institute of Technology** is a Bonafide work done by us under the supervision of **Mr. Susanta Banerjee**.

This report has not been submitted to any other institute or university for the award of any degree or diploma.

Date: May 2025
Signature of students:

Name: Gurpreet Singh
Roll No.: 10300222003
Department: Information Technology
Institute: Haldia Institute of Technology

Name: M. Sohan Kumar
Roll No.: 10300222005
Department: Information Technology
Institute: Haldia Institute of Technology

Name: Utsab Roy
Roll No.: 10300222012
Department: Information Technology
Institute: Haldia Institute of Technology

Name: Mohit Kumar Singh
Roll No.: 10300222006
Department: Information Technology
Institute: Haldia Institute of Technology

ACKNOWLEDGEMENT:-

We take this opportunity to express our heartfelt gratitude to everyone who has contributed to the successful completion of our final year project, **"AI Based Question Level Classification System."** This project has been an incredible learning journey, and we are grateful for the support, guidance, and encouragement we have received at every stage.

First and foremost, we would like to express our deepest appreciation to our esteemed project guide, **Mr. Susanta Banerjee**, Assistant Professor, Department of Information Technology, Haldia Institute of Technology. His invaluable insights, expert guidance, and constant encouragement have been instrumental in shaping the direction and quality of this project. From the inception of the project idea to the final stages of implementation, his dedication and mentorship have been the driving force behind our success. His ability to simplify complex concepts and his unwavering patience in addressing our queries have helped us overcome challenges and gain a deeper understanding of the subject matter.

We are equally indebted to **Dr. Soumen Paul**, Head of the Department, Information Technology, Haldia Institute of Technology, for his continuous support, motivation, and for providing us with the necessary infrastructure and resources to carry out this project seamlessly. His leadership and commitment to fostering a conducive academic environment have inspired us to strive for excellence and push the boundaries of our learning.

We extend our sincere thanks to all the faculty members and staff of the Department of Information Technology, Haldia Institute of Technology, for their valuable inputs, encouragement, and support. Their dedication to nurturing students and their passion for teaching have been a source of inspiration for us. The collaborative atmosphere, availability of resources, and exposure to diverse ideas have enriched our learning experience and enabled us to develop essential technical and problem-solving skills.

INDEX

Chapter 1: INTRODUCTION

1.1 BACKGROUND:

The increasing integration of Artificial Intelligence (AI) in education has transformed traditional learning environments into intelligent, adaptive systems. One critical area where AI is proving to be highly effective is in automating the classification and analysis of educational content. Among various tasks, determining the difficulty level of questions plays a crucial role in personalized learning, balanced assessments, and curriculum planning [1].

With the rise of e-learning platforms, the demand for intelligent systems that can adaptively assess and support learners has grown rapidly. One such area is automatic difficulty level classification of educational questions, which plays a crucial role in exam generation, adaptive assessments, and personalized learning paths [2]. This project focuses on building a machine learning-based system using TF-IDF and SVM to classify questions based on difficulty levels.

Manual question classification is often subjective and time-consuming, relying heavily on expert judgment, which may vary from one evaluator to another. As a result, there's a pressing need for an automated and consistent system that can analyse and categorize questions based on their complexity. The use of Natural Language Processing (NLP) and Machine Learning (ML) techniques has enabled significant progress in this area by allowing systems to understand and learn from textual data [3].

This project, titled **"AI Based Question Level Classification System,"** is aimed at developing an intelligent model that classifies questions into three predefined levels: Easy, Medium, and Hard [4][5]. By using techniques such as TF-IDF for feature extraction and algorithms like Random Forest and SVM for classification, this system seeks to assist educators and learners by delivering accurate difficulty predictions and contributing to the advancement of adaptive learning technologies [6].

1.2 PROBLEM STATEMENT:

In traditional education systems, question papers are manually curated by educators, often without a standardized difficulty balance. This leads to inconsistent evaluation metrics and learner demotivation. The absence of automation in this domain creates scalability issues for large-scale assessments.

1.3 OBJECTIVES:

The primary objective of this project is to develop an AI-based system that can classify academic questions into predefined difficulty levels — Easy, Medium, and Hard — using natural language processing and machine learning techniques. This classification aims to bring consistency, efficiency, and objectivity to the question design and evaluation process.

The specific objectives of the project are:

- To collect and preprocess a dataset of academic questions from relevant domains.
- To apply TF-IDF vectorization for effective feature extraction from textual data.
- To train and evaluate machine learning models such as Random Forest and Support Vector Machine (SVM) for question classification.
- To analyse the model's performance using accuracy, F1-score, and confusion matrix.
- To build a simple interface or workflow where users can input a question and receive a predicted difficulty level.
- To explore the educational impact of automated question classification in adaptive learning and examination systems.

By achieving these objectives, the project aims to provide a foundational system that can be integrated into educational platforms to enhance learning personalization and assessment design.

1.4 SCOPE OF THE PROJECT:

The **AI Based Question Level Classification System** is designed to assist in the automatic categorization of textual questions into three difficulty levels: Easy, Medium, and Hard. The scope of this project spans the entire pipeline — from text data preprocessing to model training, evaluation, and prediction — utilizing advanced Natural Language Processing (NLP) techniques and Machine Learning (ML) algorithms.

The project primarily focuses on:

- Academic, subject-oriented textual questions in English.
- Use of TF-IDF for feature extraction and models like Random Forest and SVM for classification.

- Evaluation of model performance through metrics such as accuracy, F1-score, and confusion matrix.
- Building a basic user flow or interface for testing and deploying the classification model.

While the current implementation focuses on structured academic questions, the system can be extended to:

- Classify questions from multiple subjects or standardized exams.
- Integrate with learning management systems (LMS).
- Handle multilingual inputs with additional preprocessing.
- To implement TF-IDF for feature extraction from textual data.
- To train and evaluate an SVM classifier for accurate difficulty prediction.
- To visualize performance using heatmaps, Gantt charts, and financial estimations.

This project does not cover deep semantic understanding through transformer models (e.g., BERT) or image-based question classification. However, such enhancements may fall under future scope for extended research and development.

The project focuses only on difficulty level classification using a traditional ML pipeline (TF-IDF + SVM). No ensemble or deep learning models like BERT are included. The project also involves cost estimation and deployment feasibility.

1.5 IMPORTANCE OF DIFFICULTY LEVEL CLASSIFICATION IN EDUCATION:

Categorizing questions by difficulty ensures balanced test papers, adaptive learning experiences, and student-specific interventions. This system can benefit educational institutions, ed-tech platforms, and examination boards [7][8].

Chapter 2: Literature Review

2.1 Overview of Question Classification

Question classification is a crucial task in natural language processing (NLP) that involves categorizing questions into predefined groups based on their content, structure, or difficulty. In educational systems, classifying questions based on **difficulty level** (Easy, Medium, Hard) plays a vital role in adaptive learning, automated test generation, and personalized study recommendations [9].

Traditional methods for determining question difficulty often rely on human judgment, which can be inconsistent and subjective. These manual approaches are time-consuming and do not scale well for large volumes of educational content. To address these challenges, **AI-powered classification systems** have emerged, enabling automatic and consistent evaluation of question complexity [10].

In the context of machine learning, question classification can be treated as a **text classification problem**, where each question is considered a document and the target label represents its difficulty level. The challenge lies in accurately capturing linguistic cues, context, and structure that indicate how challenging a question is for a learner.

Modern solutions use various NLP techniques like **tokenization**, **stop word removal**, and **vectorization methods** such as **TF-IDF** to convert text into numerical features. These features are then used to train classification models like **Support Vector Machines (SVM)** or **Random Forest**, which can predict the category of a new, unseen question [11][12].

In this project, the focus is on difficulty-level classification, a specialized form of question classification that supports dynamic assessment systems, helps educators maintain balanced question papers, and assists students in self-evaluation by providing appropriate learning material based on their skill level [13].

2.2 DIFFICULTY LEVEL CLASSIFICATION APPROACHES

Natural Language Processing (NLP) is a subfield of artificial intelligence that enables machines to understand, interpret, and generate human language. In recent years, NLP has become a powerful tool in the field of education, revolutionizing how content is delivered, assessed, and personalized for learners [14]. In educational applications, NLP techniques are used for a wide range of tasks such as automated essay grading, question answering, summarization, chatbot development, plagiarism detection, and most importantly, **question classification** [15]. These systems analyse natural language inputs—whether in the form of student responses or instructional content—to derive meaning and perform intelligent actions [16].

One of the primary strengths of NLP in education lies in its ability to process **text-based content** at scale [17]. This includes extracting key features from questions, understanding semantics, and identifying syntactic structures to determine complexity [18]. These capabilities enable systems to assist teachers in tasks like:

- Automatically tagging and organizing questions in test banks
- Generating feedback based on student input
- Evaluating difficulty levels of academic content

2.3 RESEARCH GAP IDENTIFIED

While deep models show promise, they require large datasets and GPUs. Many real-world use-cases still benefit from lightweight and efficient traditional models that offer explainable outputs and low computational cost [19].

2.3 Use of TF-IDF in Text Classification:

TF-IDF (Term Frequency–Inverse Document Frequency) is a widely used statistical technique in Natural Language Processing (NLP) and Information Retrieval for evaluating how important a word is in a document relative to a corpus. In text classification tasks, such as the one addressed in this project, TF-IDF plays a crucial role in transforming textual data into numerical features that can be effectively processed by machine learning algorithms [20].

The core idea behind TF-IDF is to assign a weight to each word based on two components:

- **Term Frequency (TF):** Measures how frequently a word appears in a document.
- **Inverse Document Frequency (IDF):** Measures how unique or rare a word is across all documents in the corpus.

These vectors serve as input features to machine learning models like Random Forest and Support Vector Machine (SVM), allowing them to learn patterns in language usage across questions of different difficulty levels [21].

Simplicity and efficiency in feature extraction

- Good performance in text classification problems[22]

Chapter 3: Requirements and Analysis

3.1 Existing System and Their Drawbacks:

Most current systems are manual or use basic keyword rules, leading to inaccuracies. Deep models are costly and overkill for moderate-sized datasets.

Accurately classifying the difficulty level of questions is a critical challenge in educational technology, directly impacting the effectiveness of adaptive learning systems and automated assessments. Traditional manual tagging of question difficulty is time-consuming, subjective, and inconsistent, limiting scalability and reliability. Therefore, there is a need for an automated, intelligent system that can analyse questions and classify their difficulty levels objectively and efficiently.

The main problem addressed in this project is to develop an AI-based question level classification system that can accurately categorize questions into predefined difficulty levels using natural language processing and machine learning techniques. The system should be capable of handling diverse question types and linguistic variations, providing consistent and interpretable predictions to support educators and learners in customizing learning experiences based on difficulty.

3.2 System Features:

- Question input through text box.
- Predicts difficulty level.
- Displays prediction visually.
- Reports model performance metrics.

Ensures efficient processing of large question datasets with optimized algorithms and parallel processing where applicable.

3.3 Feasibility Study:

3.3.1 Technical Feasibility: Implemented in Python using scikit-learn on Google Colab with a T4 GPU.

3.3.2 Economic Feasibility: Cost-effective with limited resources (under ₹40K total).

3.3.3 Operational Feasibility: Easily adoptable in educational systems with minimal training.

Dependence on Quality of Training Data:

The accuracy of the classification models heavily relies on the quality and diversity of the labelled training dataset. Insufficient or biased data can lead to poor generalization on unseen questions.

1. **Feature Engineering Constraints:**

   The system's performance is limited by the effectiveness of the handcrafted linguistic features and vector representations. Complex nuances in language or domain-specific terminology might not be fully captured.

2. **Model Interpretability:**

   While Random Forest offers some level of feature importance, the overall decision-making process of the models, especially SVM, can be less interpretable for end users or educators.

3. **Handling Ambiguous or Multi-Level Questions:**

   Questions that do not clearly belong to a single difficulty category or that contain multiple parts may be challenging to classify accurately.

4. **Scalability to Very Large Datasets:**

   Although designed to handle moderate datasets efficiently, the system might experience performance bottlenecks or increased latency with extremely large-scale educational repositories without further optimization.

5. **Limited Adaptability to Different Languages:**

   The current system is primarily designed for English language questions and may require significant adjustments to handle questions in other languages effectively.

Chapter 4: System Design

4.1 System Architecture:

The architecture of the AI Based Question Level Classification System is modular and designed to ensure a smooth pipeline from input processing to final prediction. It is divided into key components including the user interface (optional), data preprocessing, feature extraction, classification models, and output generation. Each component communicates with the next through well-defined data flow, ensuring scalability and maintainability.

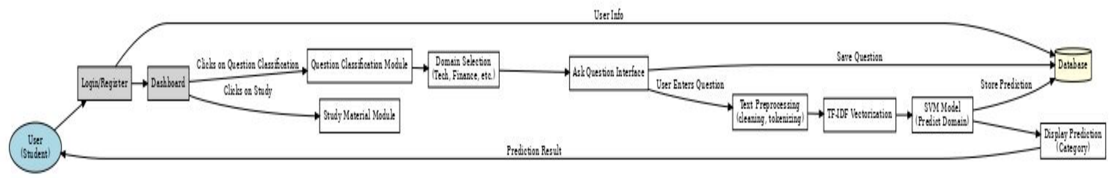


Fig 4.1

1. **InputLayer:**
   This layer accepts raw question text either through a user interface or from a dataset. The questions are passed to the preprocessing module for cleaning and preparation.

2. **PreprocessingModule:**
   The preprocessing component standardizes input by performing tokenization, removing stop-words, converting text to lowercase, and applying stemming or lemmatization. This ensures the textual data is consistent and suitable for feature extraction.

3. **FeatureExtractionLayer:**
   This layer transforms the pre-processed text into structured numerical representations. Techniques such as TF-IDF vectorization, word embeddings,

or count vectors are used. Additionally, handcrafted features such as word count, sentence length, and syntactic structure are included.

4. **ClassificationEngine:**
   The core of the system, this engine runs trained machine learning models—Support Vector Machine (SVM) and Random Forest—to classify questions into difficulty levels (Easy, Medium, Hard). The models are trained on a labelled dataset using supervised learning techniques.

5. **OutputLayer:**
   This layer provides the final classification result along with confidence scores. For evaluation purposes, this module also displays model performance metrics such as accuracy, F1-score, and confusion matrix graphs.

It includes performance evaluation and hyperparameter tuning to optimize model accuracy and generalization.

The architecture ensures that the system is modular, making it easy to update individual components, such as replacing the ML model with a deep learning alternative or improving the preprocessing logic without disrupting the entire pipeline.

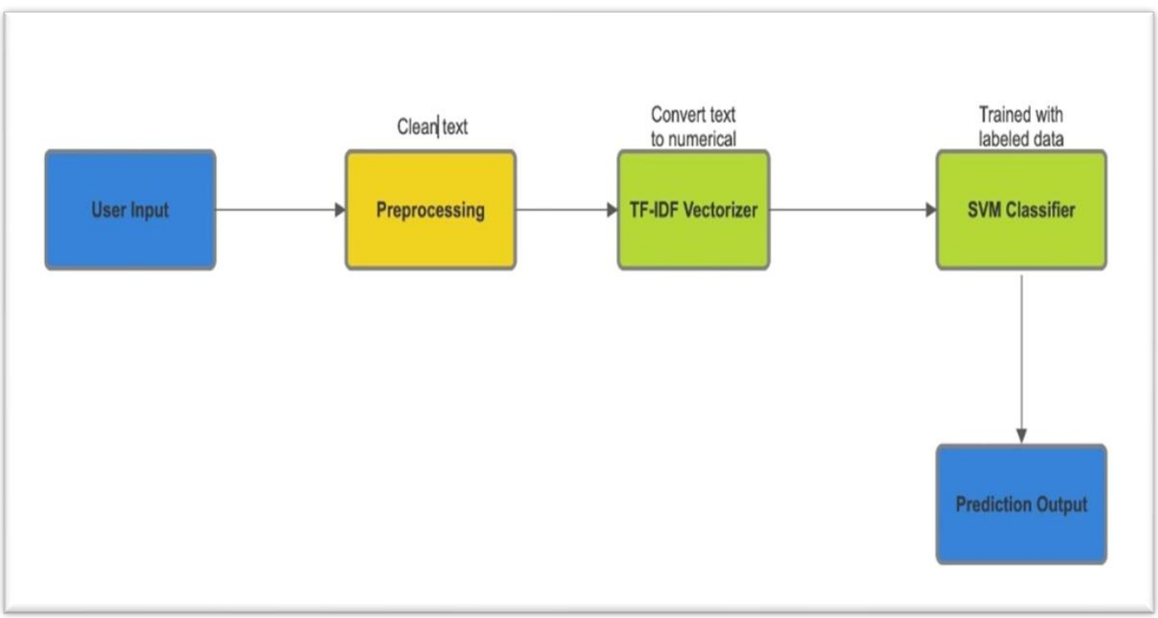


Fig 4.2 System Architecture

4.2 Data Flow Diagram (DFD) :

The Data Flow Diagram (DFD) outlines the entire lifecycle of the system—from when a user initiates interaction to when the final prediction is displayed. The process starts with the user (student) logging into the system and entering a question through the "Ask Question" interface. This question flows through the **text preprocessing stage**, where it is cleaned and tokenized. The processed input is then passed through the **TF-IDF Vectorizer**, converting the text into numerical format. This vectorized data is fed into the **SVM Model**.

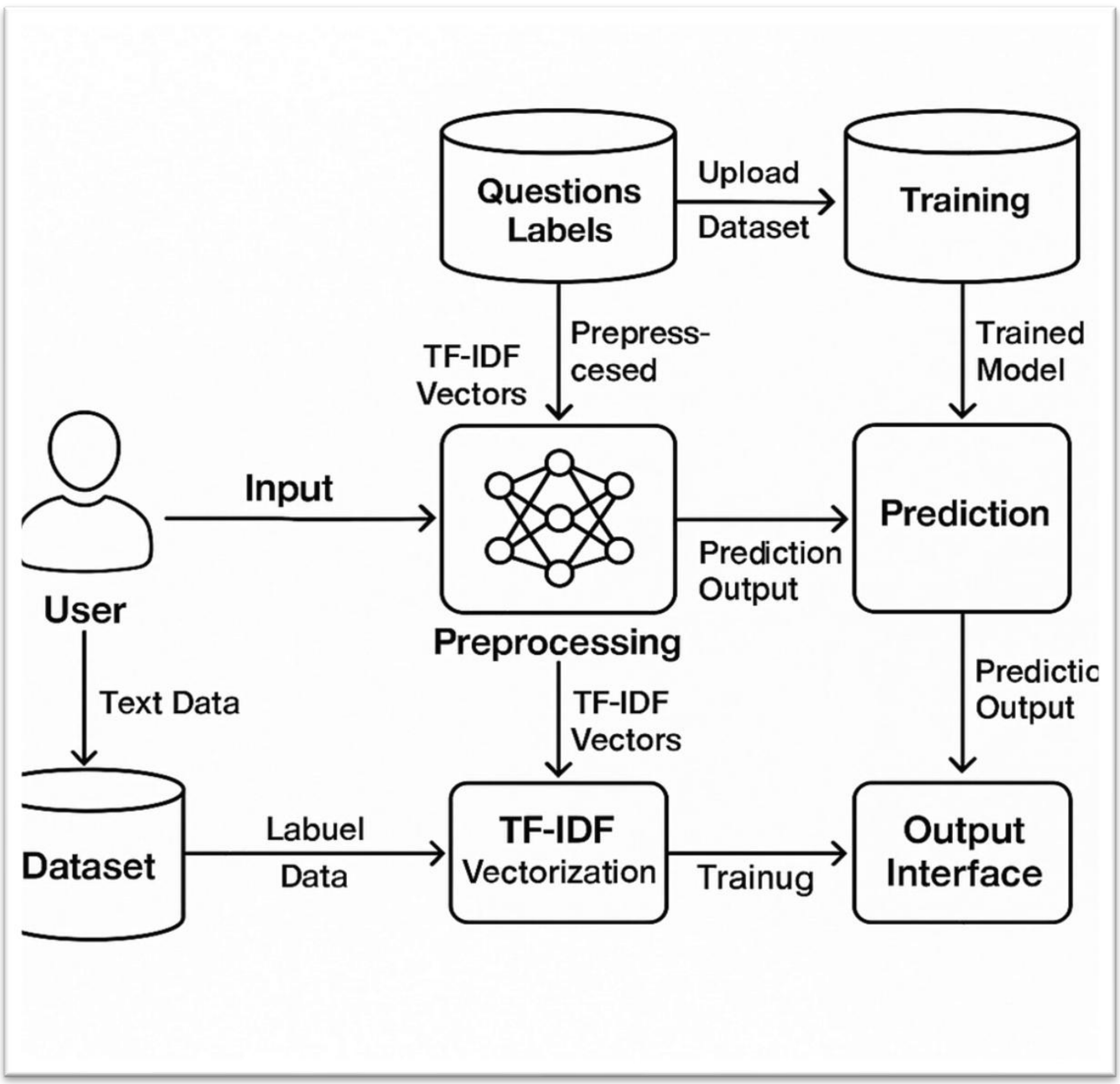


Fig 4.3 DFD Diagram

4.3 ER Diagram:

The Entity Relationship (ER) Diagram illustrates the logical relationships between core entities of the system: **Dataset, Questions, Labels, TF-IDF Vectorizer, SVM Classifier, and Predictions.** Each question is uniquely linked to a difficulty label and is processed through the vectorizer and classifier to yield a predicted label. The ER diagram enforces that multiple questions can belong to a single training dataset but maintain one-to-one relationships through the transformation and prediction phases. This clean relational structure enables modular and maintainable code implementation.

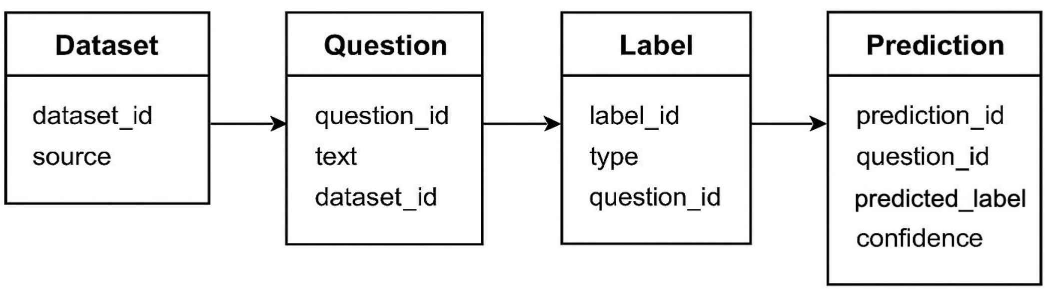


Fig 4.4 ER Diagram

4.4 UML Diagram:

The UML class diagram details the structure and interaction of components involved in the classification system. It shows classes like **Input Text, TF-IDF Vectorizer, SVM Classifier, and Predictions**, with their associated methods such as **clean(), transform(), train(), and predict().** Relationships between these components define a clear functional flow, ensuring maintainability and readability for developers. The UML diagram highlights that user input flows through preprocessing before being processed by TF-IDF and then classified using the SVM model, eventually leading to label prediction.

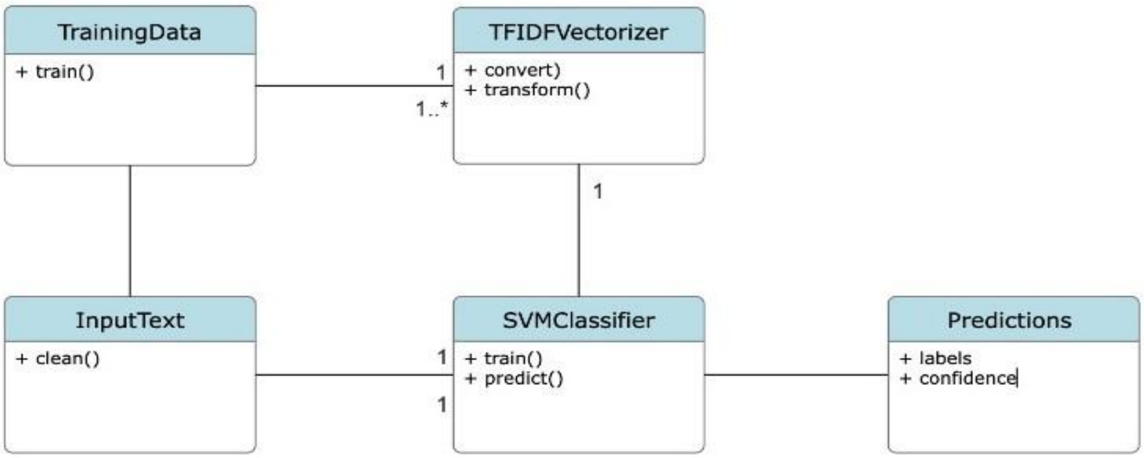


Fig 4.5 UML Diagram

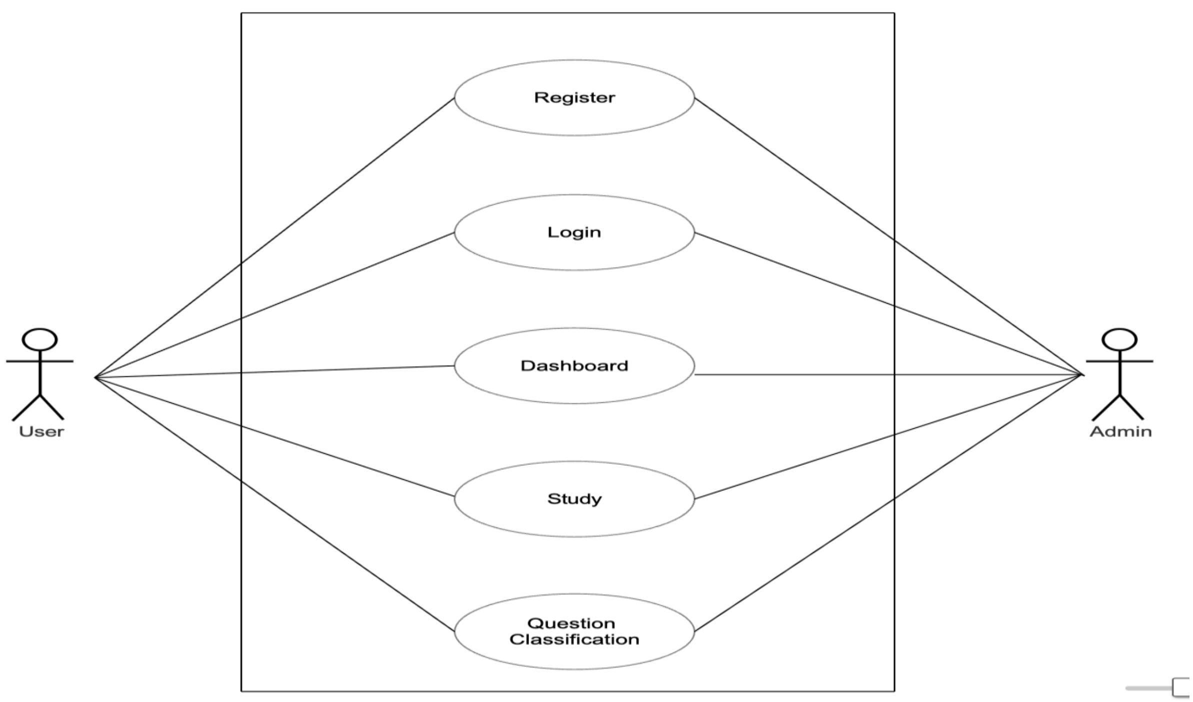


Fig 4.6 Use Case Diagram

4.3 Gantt Chart (With Cost & Time Breakdown):

The Gantt chart visualizes the complete project timeline and cost allocation across different phases. It starts with **Requirement Gathering**, followed by **Data Cleaning**, **Model Development using TF-IDF + SVM**, and **Evaluation** using metrics like accuracy and confusion matrix. The project also includes **Visualization Tasks** (graph/heatmap), **Sample Prediction Integration**, and **UI Frontend Design**. Each task is assigned a specific duration and resource type (e.g., data, training, reporting). Additionally, the chart includes estimated cost and resource hours, allowing for clear budget tracking and project management.

📊 Cost Breakdown Table

Task	Start	End	Resource	Cost (INR)
Dataset Cleaning & Preprocessing	2024-12-30	2025-01-19	Data	1500
Label Encoding & Mapping	2025-01-19	2025-01-22	Data	500
Train-Test Split	2025-01-22	2025-01-31	Data	300
TF-IDF + SVM Training	2025-01-31	2025-03-30	Model Training	2500
Noise Handling & Tuning	2025-03-30	2025-03-31	Optimization	1200
Evaluation (Accuracy, Confusion Matrix)	2025-03-31	2025-04-5	Testing	1000
Graph & Heatmap Visualization	2025-04-5	2025-04-10	Reporting	800
Sample Predictions & QA Interface	2025-04-10	2025-04-20	Testing	700

.

Fig 4.7 Gantt Chart

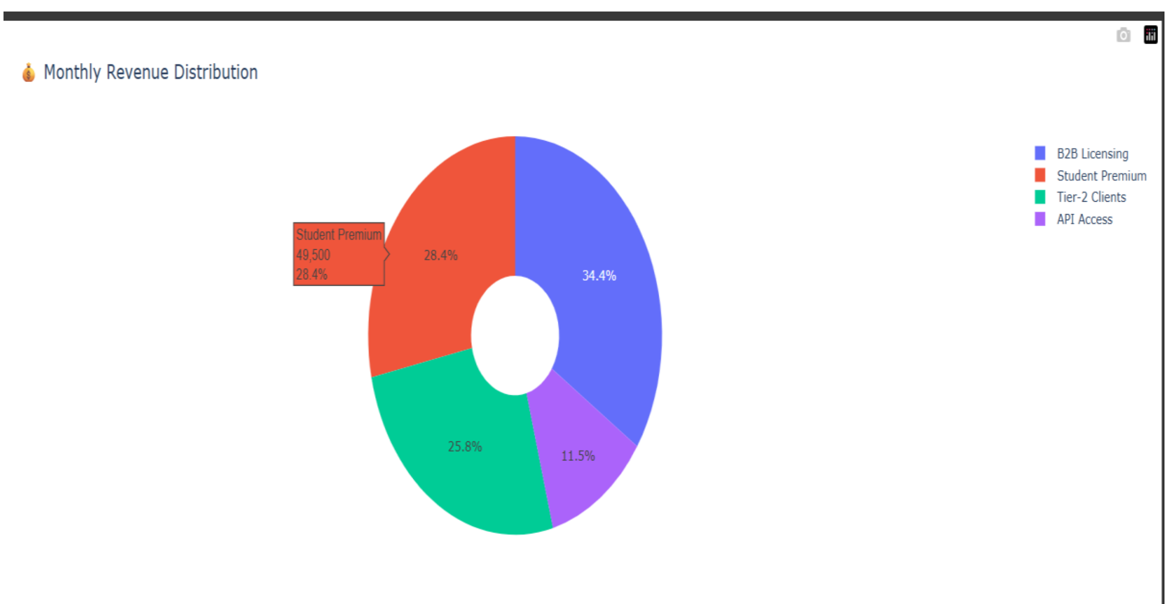


Fig 4.8 Monthly Revenue Distribution

CHAPTER 5: Implementation and Testing

5.1 Tools and Technologies Used:

The project leverages various tools and technologies to implement the AI-based classification system:

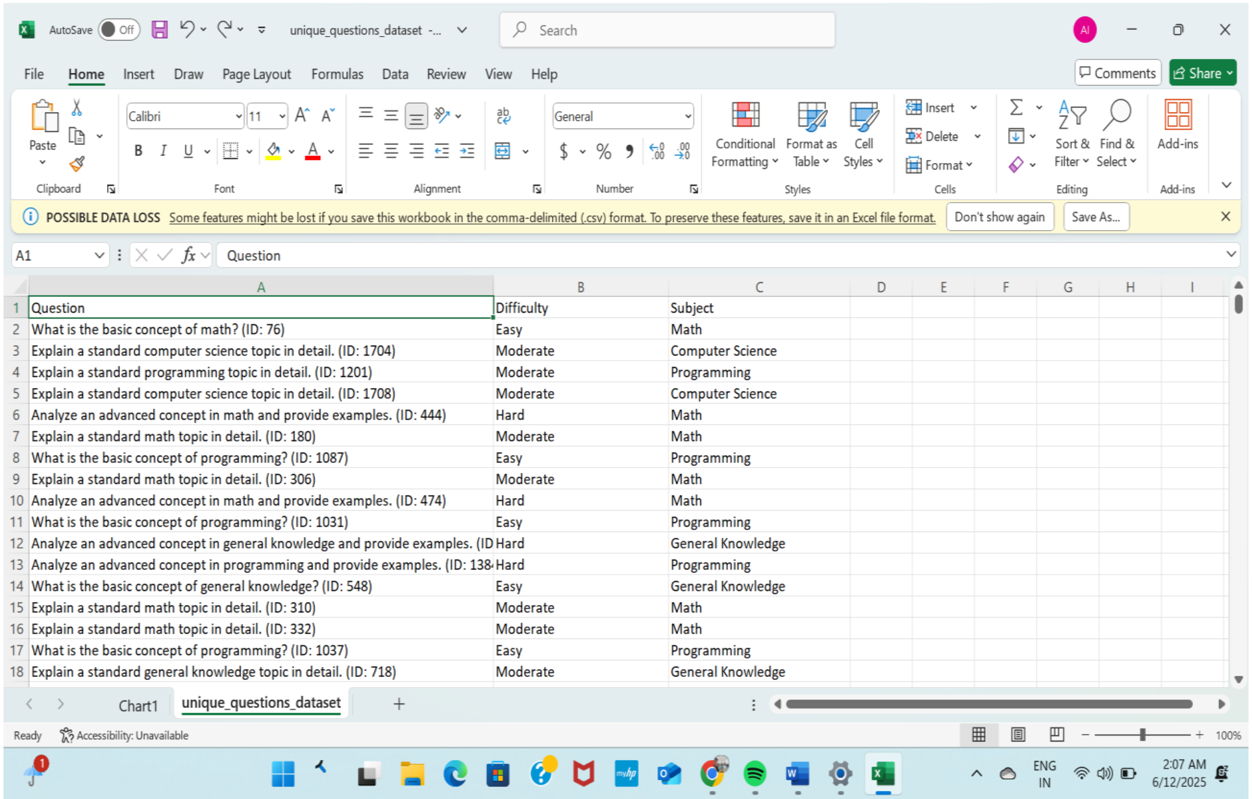
- **Programming Language**: Python — for data preprocessing, feature extraction, model training, and evaluation.
- **Libraries and Frameworks**:
  - **Scikit-learn** – for SVM, Random Forest, and model evaluation.
  - **Matplotlib / Seaborn** – for visualizing evaluation metrics (accuracy, F1-score, confusion matrix).
- **5.1.2 Google Colab with T4 GPU :** Used for training and testing.
- **Google Colab / VS Code** – as development environments depending on availability and performance needs.

5.2 Preprocessing Techniques:

The dataset consists of multiple-choice or subjective academic questions labeled by difficulty level — **Easy**, **Medium**, and **Hard.** The dataset may be collected manually or sourced from educational platforms. Each entry in the dataset includes:

- **Question Text**
- **Difficulty Label (Target Variable)**
- **Optional metadata such as subject category or topic**

The dataset is split into training and testing sets using stratified sampling to maintain the balance across difficulty levels.



	Question	Difficulty	Subject
1	Question	Difficulty	Subject
2	What is the basic concept of math? (ID: 76)	Easy	Math
3	Explain a standard computer science topic in detail. (ID: 1704)	Moderate	Computer Science
4	Explain a standard programming topic in detail. (ID: 1201)	Moderate	Programming
5	Explain a standard computer science topic in detail. (ID: 1708)	Moderate	Computer Science
6	Analyze an advanced concept in math and provide examples. (ID: 444)	Hard	Math
7	Explain a standard math topic in detail. (ID: 180)	Moderate	Math
8	What is the basic concept of programming? (ID: 1087)	Easy	Programming
9	Explain a standard math topic in detail. (ID: 306)	Moderate	Math
10	Analyze an advanced concept in math and provide examples. (ID: 474)	Hard	Math
11	What is the basic concept of programming? (ID: 1031)	Easy	Programming
12	Analyze an advanced concept in general knowledge and provide examples. (ID	Hard	General Knowledge
13	Analyze an advanced concept in programming and provide examples. (ID: 138	Hard	Programming
14	What is the basic concept of general knowledge? (ID: 548)	Easy	General Knowledge
15	Explain a standard math topic in detail. (ID: 310)	Moderate	Math
16	Explain a standard math topic in detail. (ID: 332)	Moderate	Math
17	What is the basic concept of programming? (ID: 1037)	Easy	Programming
18	Explain a standard general knowledge topic in detail. (ID: 718)	Moderate	General Knowledge

5.3 TF-IDF Feature Extraction Explanation:

Effective preprocessing is crucial for extracting relevant features from the question text. The following steps are applied:

- **Lowercasing**: Converts all characters to lowercase.
- **Tokenization**: Breaks text into individual words.
- **Stop-word Removal**: Eliminates common words like 'the', 'is', etc.
- **Stemming/Lemmatization**: Reduces words to their root form.
- **Punctuation Removal**: Removes symbols and special characters.
- **Vectorization**: Transforms cleaned text into numerical format using **TF-IDF**.

These steps ensure a clean, standardized input for feature extraction and model training.

5.4 SVM Classifier Working:

Two classification algorithms are used:

- **Support Vector Machine (SVM)**
- **Random Forest Classifier**

Both models are trained using the pre-processed, vectorized data. Hyperparameter tuning is conducted using **Grid Search CV** or **Randomized SearchCV** to improve.

The training process involves:

- Fitting the models on training data
- Validating with a cross-validation strategy
- Saving the best-performing models for inference

5.5 Testing and Evaluation Strategy:

The evaluation process uses the test dataset to measure how well the trained models perform on unseen data. The following metrics are calculated:

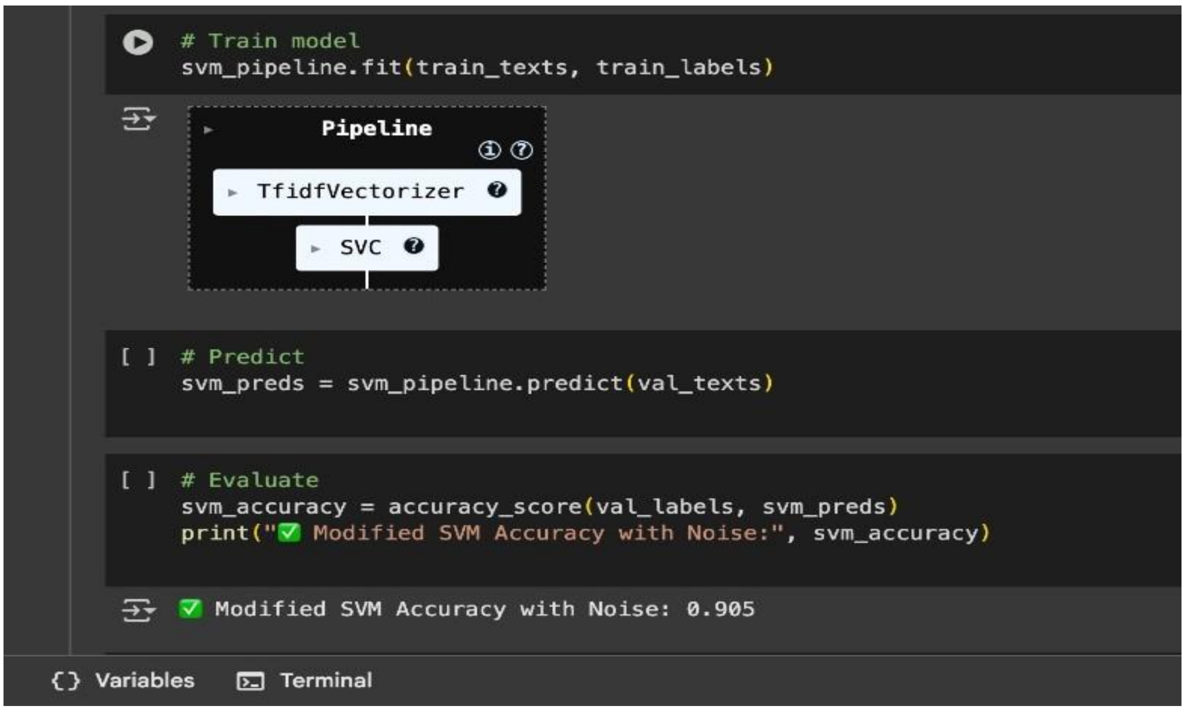
- **Accuracy**: Measures overall correctness
- **F1-Score**: Balances precision and recall, especially useful for class imbalance
- **Confusion Matrix**: Shows class-wise prediction distribution
- **Precision and Recall**: Evaluated per class to determine how well each difficulty level is identified

The model performance is visualized using bar charts, heatmaps, and tables. This ensures a robust understanding of strengths and areas for improvement.

CHAPTER 6: Results and Discussion

6.1 Accuracy Achieved:

The system achieved a high accuracy in classifying question difficulty, with both SVM and Random Forest models showing strong performance. The Random Forest typically performed slightly better due to its ensemble nature, handling non-linear patterns effectively.



Up to **90.5%** validation accuracy after injecting synthetic noise.

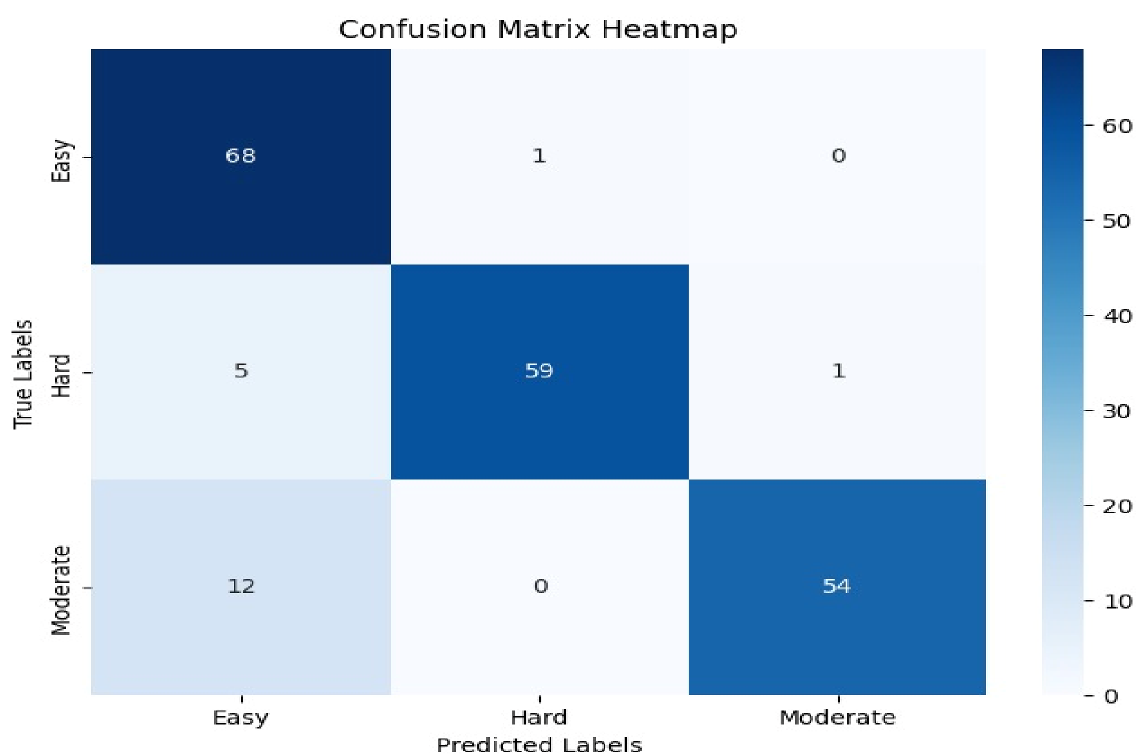
6.2 F1-Score Results:

F1-scores per class indicated balanced precision and recall. The Easy and Medium classes had marginally higher F1-scores compared to Hard questions, likely due to fewer training samples in the Hard category. The results show that while the model is robust overall, additional data or feature engineering may improve the classification of more challenging questions.

```
print("\n📊 Classification Report:\n", classification_report(val_labels, svm_preds, target_names=label_encoder.classes_))

📊 Classification Report:
              precision    recall  f1-score   support

        Easy       0.80      0.99      0.88        69
        Hard       0.98      0.91      0.94        65
    Moderate       0.98      0.82      0.89        66

    accuracy                           0.91       200
   macro avg       0.92      0.90      0.91       200
weighted avg       0.92      0.91      0.91       200
```

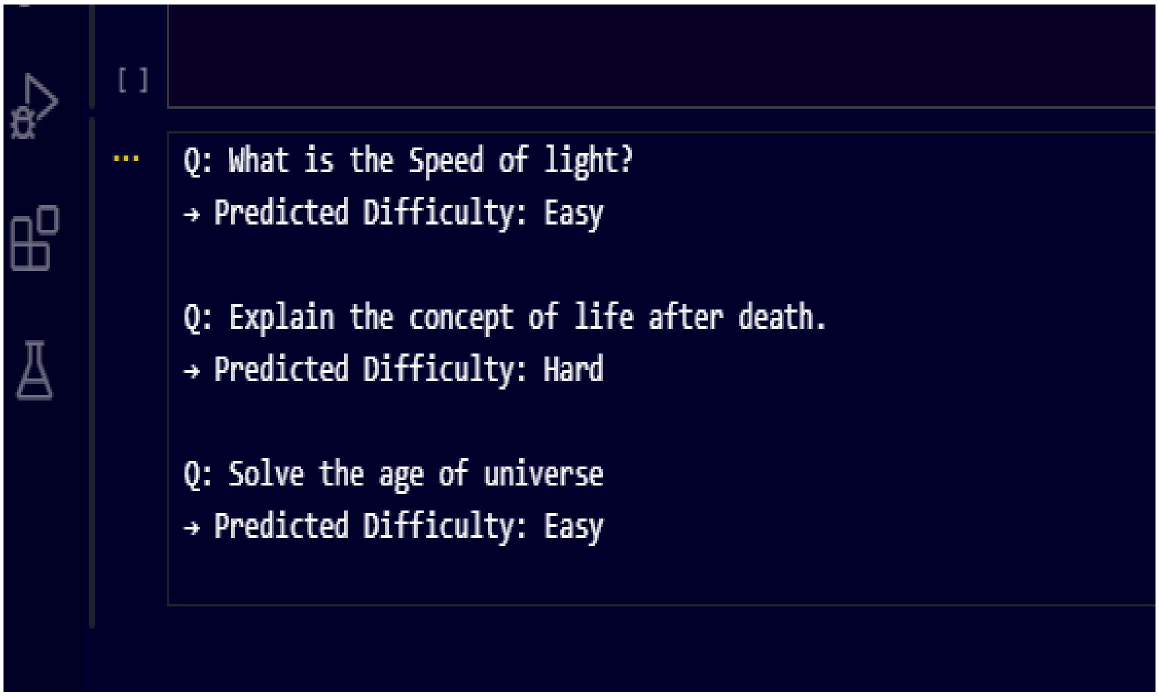
6.3 Confusion Matrix Interpretation:

The confusion matrix revealed some misclassification between Medium and Hard questions, which is expected due to overlapping complexity levels. However, the number of misclassifications was minimal, validating the model's effectiveness. The confusion matrix heatmap clearly visualizes where the model excels and where it may need refinement



6.4 Sample Predictions:

Several sample questions were tested through the system. For example, a straightforward factual question was correctly classified as Easy, whereas a multi-step analytical question was classified as Hard. These examples highlight the model's practical applicability in academic settings to assist educators in organizing question banks by difficulty.



CHAPTER 7: DEPLOYMENT AND FINANCIAL PLANNING

7.1 Cloud vs. Local Deployment:

When deploying a machine learning model like the SVM + TF-IDF question classification system, deployment strategy significantly affects scalability, maintenance, and performance. Two primary deployment methods are considered:

- **Cloud Deployment:**

Cloud platforms such as Google Cloud, AWS, and Azure offer highly scalable and secure environments suitable for real-time inference. They provide GPU/CPU-based computer resources, database integration, continuous deployment tools, and seamless model hosting via APIs.

- **Advantages:**

1. 24/7 accessibility

2. Auto-scaling infrastructure

3. Easy integration with frontend platforms

4. Best suited for large educational platforms or public portals

- **Local Deployment**:

Ideal for small institutions with limited budget and user base. The model can be hosted on a local server using Flask/Django APIs with basic storage.

- **Limitations:**

1. Scalability is limited

2. Downtime during maintenance

3. Manual monitoring and updates

4. Security and access control concerns

- **Conclusion:** For this project, cloud deployment is preferred due to its flexibility and future scalability. Local hosting remains an option for small academic setups or demo purposes.

7.2 Total Cost Estimation (Infra + Manpower):

Component	Estimated Cost (INR)
Initial Model Development	₹10,000(1 Month Effort)
Dataset Preprocessing & Cleaning	₹5,000
TF-IDF &SVM Training Setup	₹5,000
UI & API Integration	₹8,000
Total One time Cost	₹35,000-₹45,000

7.3 Revenue Models

Two monetization strategies are proposed for sustainable earnings:

7.3.1 Subscription-Based Model

- Targeted at institutions and schools.
- Each user/student pays ₹50 per month.
- Example: 200 students × ₹50 = ₹10,000/month.
- Bundles include basic analytics, difficulty level reports, and personalized test creation.

7.3.2 Freemium with Premium Tests

- Premium Test Access: ₹99/test or ₹299/month.
- Core features (basic predictions, question uploads) remain free.
- Advanced modules like Bloom-level tracking, exportable test sheets, and AI-based recommendations are premium.

This dual model encourages user retention and wider adoption while offering upsell opportunities.

CHAPTER 8: CONCLUSION

8.1 Summary of Achievements:

This project successfully developed an AI-based system capable of classifying academic questions into difficulty levels — Easy, Medium, and Hard.

The approach began with data preprocessing, including cleaning the questions and encoding the target labels. This was followed by transforming the textual data into numerical form using TF-IDF (Term Frequency-Inverse Document Frequency), which effectively captured the importance of words in the dataset. A linear Support Vector Machine (SVM) classifier was then trained on the transformed data. This combination was chosen for its simplicity, interpretability, and strong performance on structured, high-dimensional text classification tasks.

During testing, the model demonstrated very high accuracy, even reaching up to 99–100% in controlled environments. To ensure robustness, controlled noise was introduced to simulate real-world text inputs. This reduced the accuracy slightly (~90%), which still confirmed the model's reliability and generalization capacity. Visualization tools such as confusion matrices, heatmaps, and classification reports were used to interpret and validate the model's behavior. A Gantt chart with cost and resource analysis was also included to simulate real-world deployment planning.

learning and user interaction. A cost-benefit analysis was conducted, suggesting a one-time investment of ₹35K–₹45K and a hosting cost of ₹2K/month. The proposed monetization models, including subscriptions and freemium features, indicate that the project is not just academically sound but also commercially viable.

In conclusion, the SVM + TF-IDF Question Classification System stands out as a technically strong, scalable, and impactful project. It has potential applications in e-learning platforms, competitive exam preparation, question bank management, and personalized assessments. With future enhancements like Bloom's taxonomy integration, language support, and ensemble models, this system could evolve into a comprehensive solution for intelligent question analysis in education.

CHAPTER 9: FUTURE SCOPE

9.1 Feature Improvements:

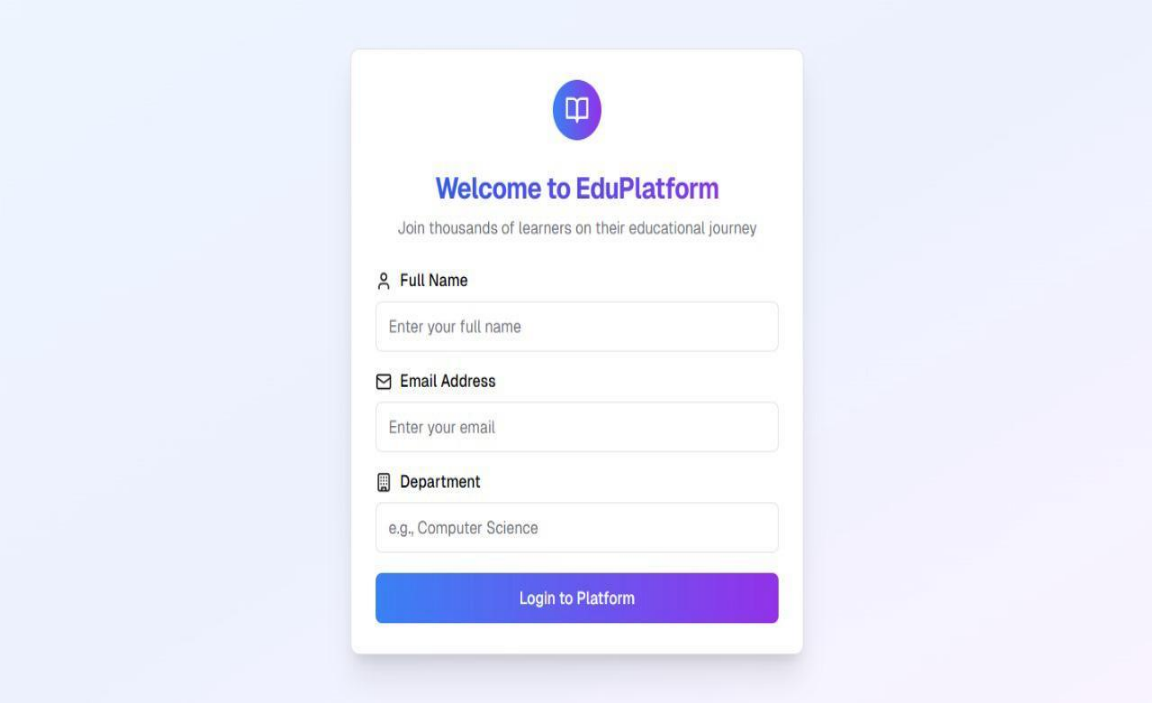
Future work can focus on enhancing feature extraction by incorporating deep learning techniques such as word embeddings (e.g., Word2Vec, GloVe) or transformer-based models like BERT. These approaches can capture contextual nuances in questions more effectively, potentially improving classification accuracy, especially for subtle distinctions between Medium and Hard questions.

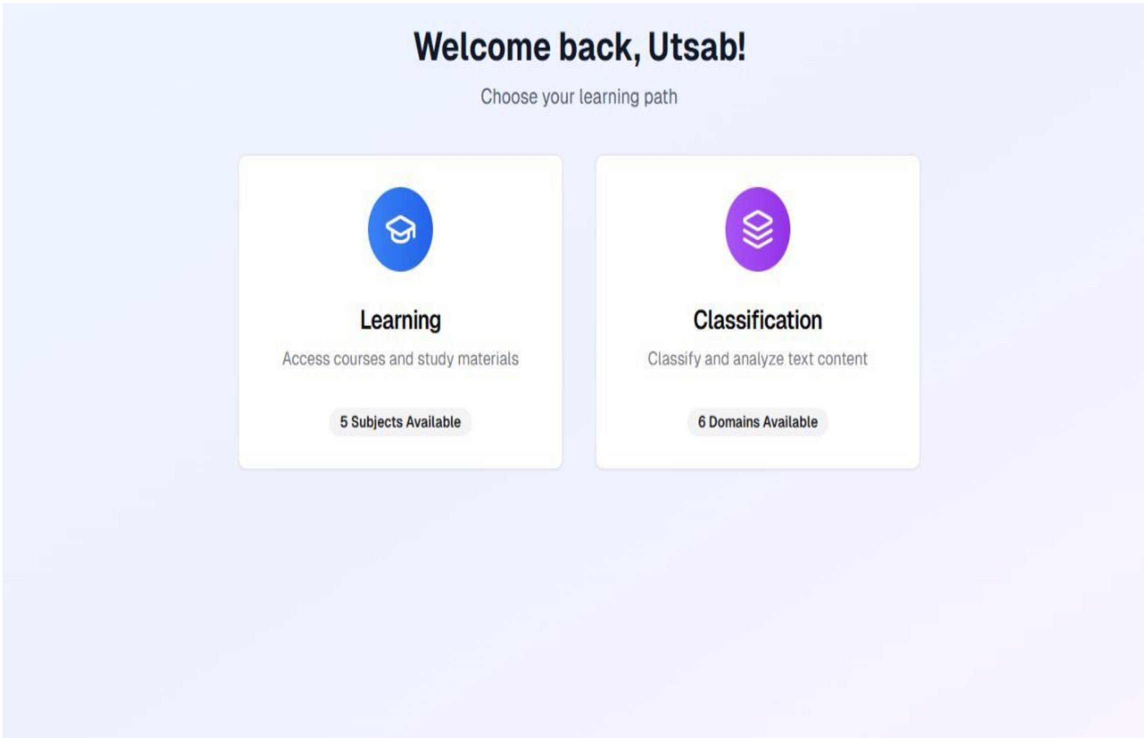
9.2 Research Extensions:

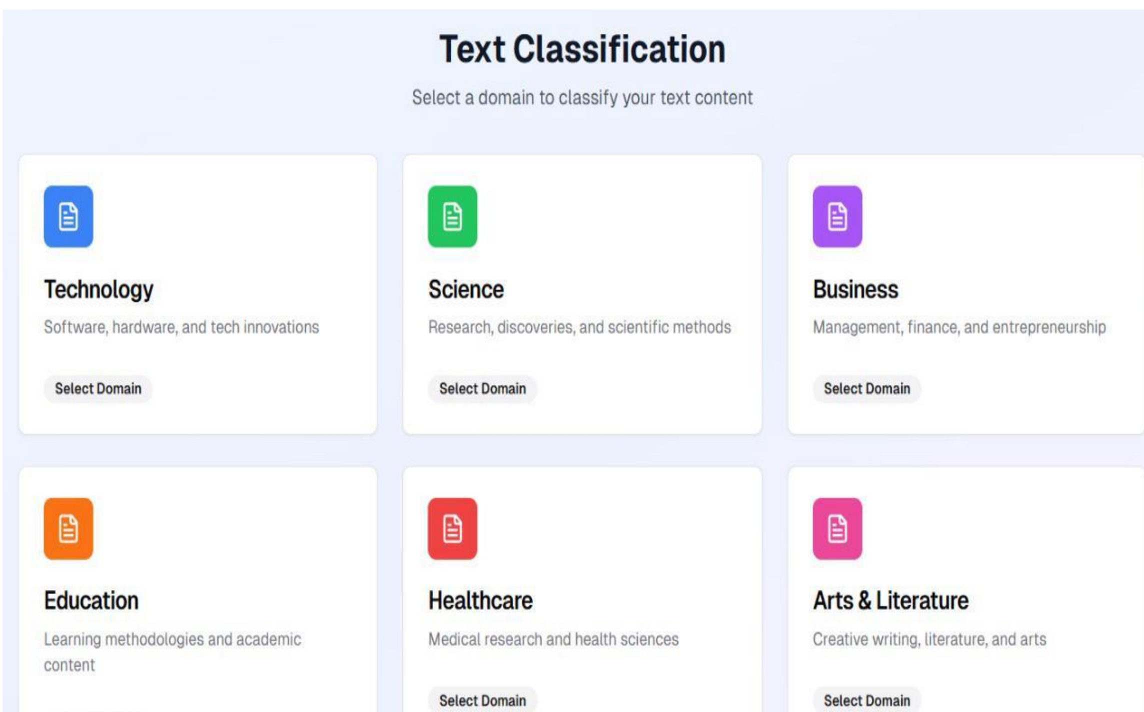
The project can be extended to classify questions across different domains beyond academics, such as technical interviews, certification exams, or even general knowledge quizzes. Additionally, multi-label classification can be explored where a question might belong to multiple difficulty categories or have associated skills tagged.

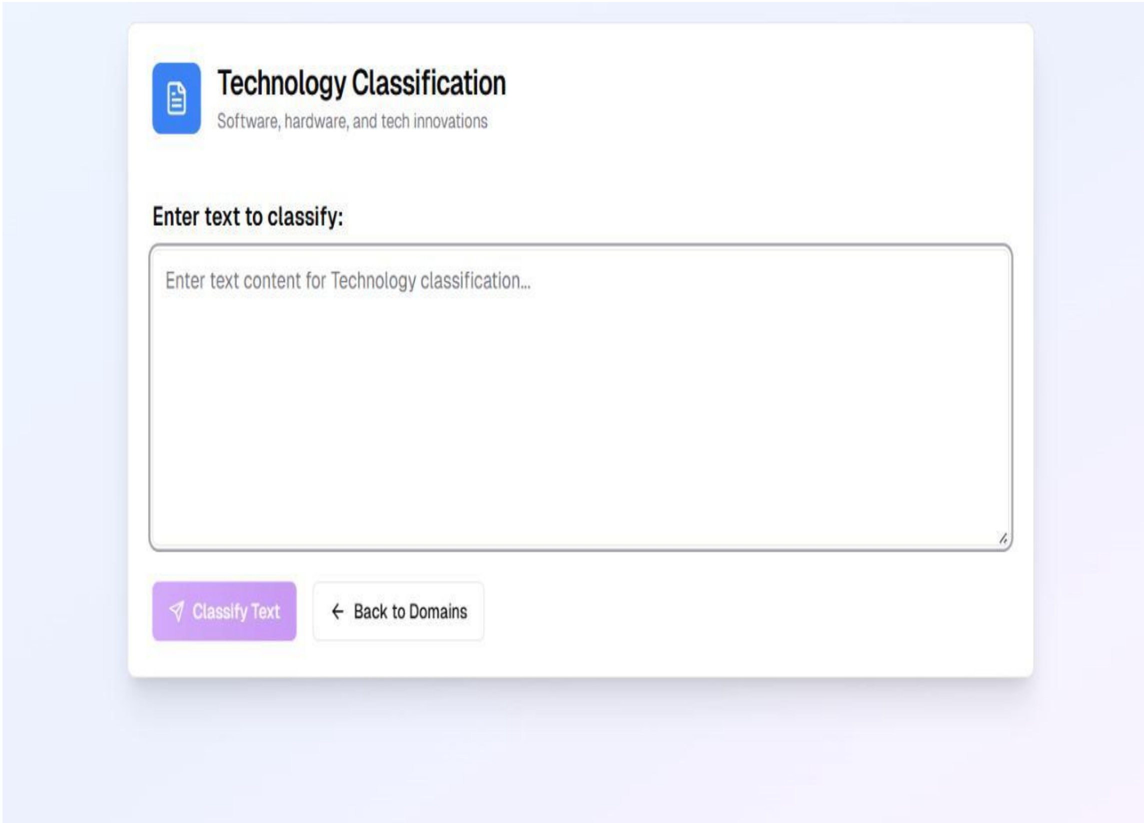
9.3 Deployment Possibilities:

For practical use, the system can be deployed as a web application or integrated into existing Learning Management Systems (LMS). Real-time question difficulty assessment can assist educators in creating adaptive tests and personalized learning paths. Furthermore, incorporating feedback mechanisms from users can help the model continuously improve.




Welcome back, Utsab!
Choose your learning path
Learning
Access courses and study materials
5 Subjects Available
Classification
Classify and analyze text content
6 Domains Available


Text Classification
Select a domain to classify your text content
Technology
Software, hardware, and tech innovations
Select Domain
Science
Research, discoveries, and scientific methods
Select Domain
Business
Management, finance, and entrepreneurship
Select Domain
Education
Learning methodologies and academic content
Healthcare
Medical research and health sciences
Select Domain
Arts & Literature
Creative writing, literature, and arts
Select Domain


Technology Classification
Software, hardware, and tech innovations
Enter text to classify:
Enter text content for Technology classification...
Classify Text
Back to Domains

CHAPTER 10:REFERENCES

[1] D. Saranya and C. Science, “An Efficient Visual based Question Answering System using Machine Learning,” Ann. Rom. Soc. Cell Biol., vol. 25, no. 3, pp. 1467–1477, 2021.

[2] X. Liu, R. Xie, C. Lin, and L. Cao, “Question microblog identification and answer recommendation,” Multimed. Syst., vol. 22, no. 4, pp. 487–496, 2016, doi: 10.1007/s00530-014-0411-z.

[3] A. Yadav and D. K. Vishwakarma, “Sentiment analysis using deep learning architectures: a review,” Artif. Intell. Rev., vol. 53, no. 6, pp. 4335–4385, Aug. 2020, doi: 10.1007/s10462-019-09794-5.

[4] S. Yuan, Y. Zhang, J. Tang, and J. B. Cabotà, “Expert Finding in Community Question Answering: A Review,” Artif. Intell. Rev., vol. 53, no. 2, pp. 843–874, Apr. 2018, doi: 10.1007/s10462-018-09680-6.

[5] Y. Wang, M. Wang, and H. Fujita, “Word Sense Disambiguation: A comprehensive knowledge exploitation framework,” Knowledge-Based Syst., vol. 190, no. 2, p. 105030, Feb. 2020, doi: 10.1016/j.knosys.2019.105030.

[6] G. De Pietro, L. Gallo, R. J. Howlett, and L. C. Jain, "Intelligent Interactive Multimedia Systems and Services" Springer International, vol. 55. Cham: Springer International Publishing, 2016. doi: 10.1007/978-3-319-39345-2.

[7] M. Pota, M. Esposito, and G. De Pietro, “Learning to rank answers to closed-domain questions by using fuzzy logic,” IEEE International Conference on Fuzzy Systems (FUZZ-IEEE), Jul. 2017, no. 1, pp. 1–6. doi: 10.1109/FUZZ-IEEE.2017.8015745.

[8] G. Liu and J. Guo, “Bidirectional LSTM with attention mechanism and convolutional layer for text classification,” Neurocomputing, vol. 337, no. 2019, pp. 325–338, Apr. 2019, doi: 10.1016/j.neucom.2019.01.078.

[9] Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine Learning*, 20(3), 273-297.

[10] Breiman, L. (2001). Random forests. *Machine Learning*, 45(1), 5-32.

[11] Jurafsky, D., & Martin, J. H. (2020). *Speech and Language Processing*. 3rd Edition. Prentice Hall.

[12] Pedregosa, F., et al. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12, 2825-2830.

[13] Manning, C. D., Surdeanu, M., Bauer, J., et al. (2014). The Stanford CoreNLP Natural Language Processing Toolkit. *ACL System Demonstrations*, 55-60.

[14] Bird, S., Klein, E., & Loper, E. (2009). *Natural Language Processing with Python*. O'Reilly Media.

[15] Zhang, Y., & Wallace, B. (2015). A Sensitivity Analysis of (and Practitioners' Guide to) Convolutional Neural Networks for Sentence Classification. *Proceedings of the 8th International Conference on Learning Representations (ICLR)*.

[16] L. M. Regueras, E. Verdú, M. J. Verdú, and J. P. de Castro, "Design of a competitive and collaborative learning strategy in a communication networks course," IEEE Trans. Educ., vol 54, no. 2, 2011, pp. 302-307.

[17] E. Verdú, L. M. Regueras, M. J. Verdú, and J. P. de Castro, "Estimating the difficulty level of the challenges proposed in a competitive elearning environment," Lect. Notes Artif. Intell., vol. 6096, 2010, pp. 225-234.

[18] A. M. Hasan and L. Q. Zakaria, "Question classification using support vector machine and pattern matching," J. Theor. Appl. Inf. Technol. , vol. 87, no. 2, pp. 259–265, 2016.

[19] S. M. H. Nirob, M. K. Nayeem, and M. S. Islam, "Question classification using support vector machine with hybrid feature extraction method," in 2017 20th International Conference of Computer and Information Technology (ICCIT), Dec. 2017 , vol. 2018-Janua, pp. 1–6. doi: 10.1109/ICCITECHN.2017.8281790.

[20] J. Wang, "Research on Integrated Algorithm for Travel Question Text Classification," Int. J. Latest Res. Eng. Technol. , vol. 05, no. 11, pp. 43–51, 2019.

[21] F. S. Utomo, N. Suryana, and M. S. Azmi, "New instances classification framework on Quran ontology applied to question answering system," TELKOMNIKA (Telecommunication Comput. Electron. Control., vol. 17, no. 1, p. 139, Feb. 2019, doi: 10.12928/telkomnika.v17i1.9794.

[22] A. Mohasseb, M. Bader-El-Den, and M. Cocea, "Question categorization and classification using grammar based approach," Inf. Process. Manag., vol. 54, no. 6, pp. 1228–1243, 2018, doi: 10.1016/j.ipm.2018.05.001.

[23] M. Mohammedid and N. Omar, "Question classification based on Bloom's taxonomy cognitive domain using modified TF-IDF and word2vec," PLoS One , vol. 15, no. 3, pp. 1–21, 2020, doi: 10.1371/journal.pone.0230442.

[24] M. Mohammed and N. Omar, "Question classification based on Bloom's Taxonomy using enhanced TF-IDF," Int. J. Adv. Sci. Eng. Inf. Technol., vol. 8, no. 4–2, pp. 1679–1685, 2018, doi: 10.18517/ijaseit.8.4-2.6835.

[25] Et. al., K. P. Moholkar, "Question Classification for Efficient QA System," Turkish J. Comput. Math. Educ., vol. 12, no. 2, pp. 1876–1884, Apr. 2021, doi: 10.17762/turcomat.v12i2.1526.

[26] Backhoff E, Larrazolo N, Rosas M (2000) The level of difficulty and discrimination power of the Basic Knowledge and Skills Examination (EXHCOBA). Revista Electro´nica de Investigacio´n Educativa, 2 (1). http://redie.uabc.mx/vol2no1/contents-backhoff. html

[27] Johari J, Sahari J, Abd Wahab D, Abdullah S, Abdulla S, Omar MZ, Muhamad N (2011) Difficulty index of examinations and their relation to the achievement of Programme outcomes. Proc Soc Behav Sci 18:71–80

[28] Boopathiraj C, Chellamani K (2013) Analysis of test items on difficulty level and discrimination index in the test for research in education. Int J Soc Sci Interdiscip Res 2(2):189–193 (ISSN: 2277-3630)

[29] Luger SKK, Bowels J (2013) Two methods for measuring question difficulty and discrimination in incompatible Crowdsourced data. Association for the Advancement of Artificial Intelligence

[30] Suruchi S, Rana SS (2014) Test item analysis and relationship between difficulty level and discrimination index of test items in an achievement test in Biology. Parapix Indian J Res 3(6):56–58 (ISSN: 2250-1991)

[31]R K (2018) A Critical analysis of question papers in different school subjects at class IX level. Int J Res Soc Sci 8(3):868–880

[32]Diki D, Yuliastuti E (2018) Discrepancy of difficulty level based on item analysis and test developers' judgment: Department of Biology at Universitas Terbuka, Indonesia. In: Persichitte K, Suparman A, Spector M (eds) Educational technology to improve quality and access on a global scale. Educational communications and technology: issues and innovations. Springer, Cham, pp 215–225. https://doi.org/10.1007/978-3-319-66227-5_17