A Field Based Project

On

THE CONSTRUCTION OF UNDERGRADUATE MACHINE LEARNING COURSE IN THE ARTIFICIAL INTELLIGENCE ERA

Submitted to

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY, HYDERABAD

In partial fulfilment of the requirement for the award of the degree of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING

A SRINIVAS KUMAR	**237Y1A05F0**
D ASHISH	**237Y1A05B1**
V VIKAS KUMAR	**23Y71A05F9**
G ANUDEEP	**237Y1A05B0**

Dr. S. PRATAP SINGH

Professor

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

MARRI LAXMAN REDDY
INSTITUTE OF TECHNOLOGY & MANAGEMENT
(AN AUTONOMOUS INSTITUTION)
(Approved by AICTE, New Delhi & Affiliated to JNTUH, Hyderabad)
NAAC Accredited Institution with 'A' Grade & Recognized Under Section2(f) & 12(B)of the UGC act,1956

MARRI LAXMAN REDDY
INSTITUTE OF TECHNOLOGY & MANAGEMENT
(AN AUTONOMOUS INSTITUTION)
(Approved by AICTE, New Delhi & Affiliated to JNTUH, Hyderabad)
NAAC Accredited Institution with 'A' Grade & Recognized Under Section2(f) & 12(B)of the UGC act,1956

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

Date:

CERTIFICATE

This is to certify that the project work entitled **"The Construction of Undergraduate Machine Learning Course in the Artificial Intelligence Era"** work done by A.SRINIVAS KUMAR(237Y1A05F0), D.ASHISH(237Y1A05B1), V.VIKAS KUMAR (237Y1A05F9), ANUDEEP(237Y1A05B0) student(s) of Department of Electronics and Communication Engineering, is a record of Bonafide work carried out by the member(s) during a period from January, 2025 to June, 2025 under the supervision of Dr. S. PRATAP SINGH This project is done as a fulfilment of obtaining Bachelor of Technology Degree to be awarded by Jawaharlal Nehru Technological University Hyderabad, Hyderabad. The matter embodied in this project report has not been submitted by us to any other university for the award of any other degree.

A SRINIVAS KUMAR	237Y1A05F0
D ASHISH	237Y1A05B1
V VIKAS KUMAR	23Y71A05F9
G ANUDEEP	237Y1A05B0

This is to certify that the above statement made by the candidate(s) is correct to the best of my knowledge.

Date: (Dr. S. PRATAP SINGH)

The Viva-Voce Examination of above student(s), has been held on.....

Head Of the Department External Examiner

Principal/Director

ACKNOWLEDGEMENTS

We would like to express my sincere gratitude to my guide **Dr. S. PRATAP SINGH, Assistant Professor**, COMPUTER SCIENCE AND ENGINEERING, for his/her excellent guidance and invaluable support, which helped me accomplish the B. Tech (CSE) degree and prepared me to achieve more life goals in the future. His total support for my dissertation and countless contributions to my technical and professional development made for a truly enjoyable and fruitful experience. Special thanks are dedicated to the discussions we had on almost every working day during my project period and for reviewing my dissertation.

I/We am very grateful to my Project Coordinator**, Dr. S PRATAP SINGH, Assistant Professor,** CSE, MLRITM, Dundigal, Hyderabad, who has not only shown utmost patience, but was fertile in suggestions, vigilant in directions of error and has been infinitely helpful.

I/We am extremely grateful to **Dr. K. ABDUL BASITH, Professor & HOD**, CSE, MLRITM, Dundigal, Hyderabad, for the moral support and encouragement given in completing my project work.

I/We wish to express deepest gratitude and thanks to **Dr. R. Murali Prasad, Principal, and Dr. P. Sridhar, Director,** for their constant support and encouragement in providing all the facilities in the college to do the project work.

I/we would also like to thank all our faculties, administrative staff and management of MLRITM, who helped me to complete the mini project.

On a more personal note, I thank my **beloved parents and friends** for their moral support during our project.

This acknowledgement is for your reference. You can write your own acknowledgement. (Sentences/words may change depending on individual copy and department copy)

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

PMI	POST MORTERN INTERVAL
AI	ARTIFICIAL INTELLIGENCE
DFD	DATA FLOW DIAGRAMS
UML	UNIFIED MODELING LANGUAGE
ER	ENTITY-RELATIONSHIP
UI/UX	USER INTERFACE/ USER EXPERIENCE

ABSTRACT

Machine learning technology has been developed in the last decade, which makes artificial intelligence reach a revolutionary breakthrough and lets us really perceive the potential of artificial intelligence in changing human life. To improve the understanding and application ability of artificial intelligence, conducting the corresponding machine learning course is of significance for the students during the undergraduate period. This paper probes into the teaching content, teaching form and other aspects of the undergraduate machine learning course based on this issue and proposes a teaching method driven by application scenarios to guide the undergraduate students to understand the development, current situation, and frontier technology of machine learning. In the experimental design, the students' theoretical knowledge is fully considered, the practical questions are simplified, and the students' ability to think and solve problems is also raised, to lay a theoretical and practical basis for further study of machine learning.

Chapter 1:

INTRODUCTION

1.1 Problem Statement

With the rapid development and widespread adoption of mobile internet, big data, and large-scale parallel computing, the cost of computation has significantly decreased. This has enabled artificial intelligence (AI) to become an area of intense research and practical application, impacting various domains such as pattern recognition, computer vision, natural language processing, data mining, business intelligence, and autonomous driving. However, there is a growing need to improve the integration of machine learning into undergraduate curricula to address the increasing demand for data analysis technologies. As data collection, storage, and processing capabilities have greatly improved, machine learning has become essential in meeting the needs of the modern era.

1.2 Objective

The objective of this project is to explore the role of machine learning in modern technological advancements and justify the need for integrating machine learning courses into undergraduate programs. The project aims to evaluate the potential impact of this integration on student preparedness for careers in AI and big data-related industries.

1.3 Scope of the Project

This project will focus on:

- Analyzing the importance of machine learning in various industries and domains.
- Investigating the need for machine learning education at the undergraduate level.
- Exploring the technological trends driving the development of artificial intelligence, including mobile internet, big data, and IoT technologies.
- Assessing the role of machine learning in addressing the data analysis challenges of the modern era.

1.4 Methodology

The research methodology for this project will involve:

- A comprehensive review of literature on the current state of artificial intelligence and machine learning.
- Analysis of the trends and strategies laid out in government policy documents, such as China's 2017 AI strategic planning.
- Case studies and examples from global IT giants like Google, Facebook, Microsoft, Baidu, Alibaba, and Tencent to understand the real-world applications and importance of AI and machine learning.
- Surveys or interviews with academic experts gather insights on the current state and prospects of machine learning education.

1.5 Tools and Technologies

The following tools and technologies will be utilized in this project:

- **Literature Review and Analysis Tools**: Academic databases, research papers, and reports on AI and machine learning.
- **Data Analysis Tools**: Software for analyzing trends in AI, such as data mining tools, statistical analysis software, and visualization tools.
- **Programming Languages**: Python and R for machine learning model demonstration and data analysis.
- **Collaborative Platforms**: Google Scholar, ResearchGate, and other academic resources for accessing and analyzing AI-related research.

Chapter 2:

Literature Survey

2.1 Existing System:

The network devices (such as routers) used in the core network are complex hardware/software systems that are vulnerable to hard-to detect/hard-to-recover errors. Hardware failures: A multi-card chassis system can have tens of separate cards, and each card can have hundreds of components. Since each component consists of hundreds of advanced chips, each chip in turn has hundreds of I/O's and millions of logic gates, and the operating frequency of chips and I/O's are now in the range, the number of incorrect hardware behaviors can be very high. Software failures: Since the throughput of modern multi card chassis system is approaching levels, failures caused by subtle interactions between parallel applications have become more frequent .in such a complex system, whenever a hardware or Software failure occurs, it is difficult for debug technicians to accurately identify the root cause of this failure and take effective corrective actions.

Literature Survey Table:

Survey	Year of Research	Research Done By	Method Used	Summary
Software Guide	2018	X. Yao, H. Wu, Z. Fang, X. You	Course Construction Study	Focuses on the construction of a machine learning course in the context of new engineering backgrounds. The study emphasizes how the curriculum can be structured to incorporate modern AI techniques.
Digital Communication World	2017	Q. Li	Survey	This paper surveys the development of artificial intelligence technology and its typical applications, providing a broad understanding of AI's role in various fields, including education and machine learning.
Electronic World	2017	C. Zhang	Application Review	Focuses on the development and application of AI technologies, examining their influence on multiple sectors, including education. It provides insights into how AI shapes machine learning curriculum.
Computer Education	2016	J. Zhu, L. Lu	Case Study	Discusses the teaching practice of machine learning at Peking University's undergraduate program in Intelligence Science and Technology. It serves as a practical example of course design in a leading institution.
Electronic Production	2016	Y. Han	Survey	Reviews the development and application of AI technology. Highlights include discussions on how AI can enhance machine learning education at the undergraduate level.
Advanced Materials Research	2014	Y. Chun	Theoretical Discussion	Discusses the direction of machine learning's development in the big data era, linking data with educational needs, particularly for undergraduate courses in machine learning.

2.2 LIMITATIONS OF THE EXISTING SYSTEM:

1. **High Hardware Complexity and Vulnerability:**

- Network devices like multi-card chassis systems consist of numerous components (cards, chips, I/O's, logic gates) operating at high frequencies (GHz range), increasing the probability of hardware faults.
- As the complexity increases, the chances of component-level failures (due to physical defects, wear-out, or thermal issues) also rise.

2. **Difficult Hardware Error Detection and Recovery:**

- Faults in such intricate hardware systems are often **hard to detect** due to the sheer volume of data and components involved.
- Recovering from these errors is equally challenging without significant downtime or redundant systems.

3. **Software Faults Due to Parallel Processing:**

- High-throughput systems rely on **parallel application execution**, making them prone to subtle software bugs and race conditions.
- These failures are **non-deterministic** and may not be reproducible, complicating debugging and testing.

4. **Root Cause Analysis is Time-Consuming and Inaccurate:**

- In the event of a failure, it is extremely **difficult for debug technicians** to pinpoint the exact cause (hardware vs. software, internal vs. external).
- This leads to delays in resolution and possible misdiagnosis, risking repeated failures.

5. **Lack of Intelligent Monitoring and Predictive Diagnostics:**

- Current systems lack advanced **self-diagnostic** or **predictive analytics** tools that could proactively identify and mitigate failures.

- Most error management is **reactive** rather than preventive, leading to operational inefficiencies.

6. Challenges in Failure Management:

- As the scale and performance of systems increase, **manual debugging and monitoring** methods become less effective and harder to scale.

2.3 PROPOSED SYSTEM:

We have presented a self-learning health analyzer for a complex core router system. the Effectiveness of proactive fault-tolerance solutions depends on whether the health status of core routers can be accurately identified in a timely manner. However, little research has focused thus far on analyzing the long-term health status in a high-performance communication system. We present the design of an efficient self-learning health analyzer that can be applied to a commercial core router system using field data collected from a set of commercial core routers. Hierarchical clustering is then utilized to infer labels for the unlabeled dataset. A classifier is built and iteratively updated using both labeled and unlabeled datasets. Field data collected from a set of commercial core routers are used to experimentally validate the proposed health-status analyzer.

2.4 Summary:

The existing system in core network environments, particularly involving complex multi-card chassis routers, faces critical reliability challenges due to both hardware and software vulnerabilities. Hardware faults arise from the intricacies of modern electronic design, such as millions of logic gates and GHz-level operation across numerous interconnected components. These issues lead to error conditions that are difficult to detect, diagnose, and recover from, resulting in significant delays and inefficiencies in system maintenance. To address these challenges, a **self-learning health analyzer** is proposed. The proposed solution provides a proactive approach to fault tolerance, surpassing traditional binary classification techniques and offering more efficient and scalable network health management.

Chapter 3:

System Analysis

3.1 Requirement Analysis:

The project involved analyzing the design of a few applications to make the application more friendly. To do so, it was important to keep the navigations from one screen to the other well-ordered and at the same time reducing the amount of typing the user needs to do. To make the application more accessible, the browser version had to be chosen so that it is compatible with most of the browsers.

SYSTEM SPECIFICATION:

Functional Requirements

- Graphical User Interface with the User.

Software Requirements

For developing the application, the following are the Software Requirements:

1. Python
2. Django

Operating Systems supported

1. Windows 10

Technologies and Languages used to Develop

1. Python

Debugger and Emulator

- Any Browser (Particularly Chrome)

Hardware Requirements

For developing the application, the following are the Hardware Requirements:

- Processor: Pentium IV or higher

- RAM: 8GB
- Space on Hard Disk: minimum 1TB

3.2 Feasibility Study

The feasibility of the project is analyzed in this phase and the business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, understanding of the major requirements for the system is essential.

Three key considerations involved in the feasibility analysis are,

- **ECONOMICAL FEASIBILITY**
- **TECHNICAL FEASIBILITY**
- **SOCIAL FEASIBILITY**

ECONOMICAL FEASIBILITY:

This study is carried out to check the economic impact that the system will have on the organization. The amount of funds that the company can pour into the research and development of the system is limited. The expenditure must be justified. Thus, the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only customized products had to be purchased.

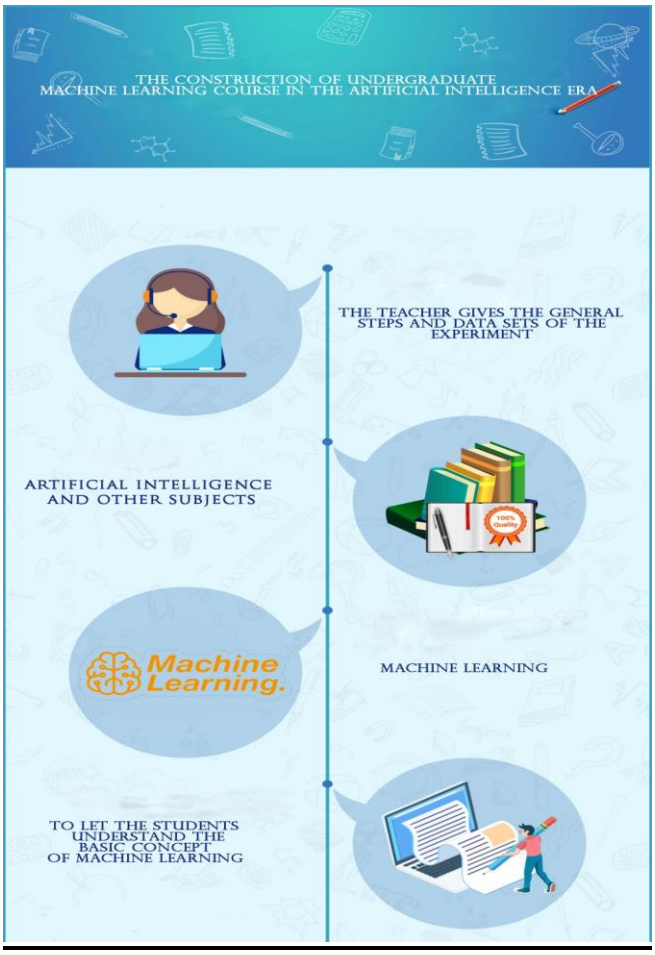
TECHNICAL FEASIBILITY:

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand for the available technical resources. This will lead to high demands for the technical resources available. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

SOCIAL FEASIBILITY:

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

3.3 System Architecture / Design Overview:



1. Data Acquisition Layer

This layer is responsible for collecting data required for machine learning tasks. It supports multiple data sources, including:

- Public CSV datasets uploaded by teachers.
- Sample image or text data for OCR tasks.
- APIs (future scope) for real-time streaming datasets.

Key Features:

- Secure upload and storage of datasets.
- File type validation and integrity checks.
- Logging of dataset source and metadata.

2. Data Preprocessing Layer

Before model training or testing, data must be cleaned and transformed. This layer ensures that input data is standardized and suitable for use.

Key Operations:

- Handling missing or null values.
- Data normalization and encoding.
- Feature extraction and dimensionality reduction (to be added in future updates).
- Ensuring compatibility with ML models (format, size, types).

3. Machine Learning Engine

This is the computational core of the system that performs predictive analysis.

Implemented Algorithms:

- **Support Vector Machine (SVM)**: For binary classification tasks.
- **K-Nearest Neighbours (KNN)**: For distance-based classification.

Planned Additions:

- Decision Trees, Logistic Regression, Random Forests.
- Deep learning integration using TensorFlow/Keras.

Functions:

- Training and testing models with labelled data.
- Generating confusion matrices and classification reports.
- Storing model performance logs.

4. User Interface Layer

This layer is responsible for user interaction and experience. Each user role (Admin, Teacher, Learner) has a customized dashboard.

Features:

- Role-based navigation.
- Responsive and accessible layout.
- Form-based task submissions.
- Graphical outputs and model result visualizations.
- Notifications, feedback, and system messages.

5. Database / Storage Layer

This layer handles structured data persistence and retrieval for all user interactions.

Database Tables:

- **Users**: Stores role-based credentials and profile information.
- **Tasks/Assignments**: Metadata and file links.
- **Submissions**: Stores learner responses.
- **ML Results**: Output logs and reports.
- **Feedback**: Teacher evaluations and learner queries.

Features:

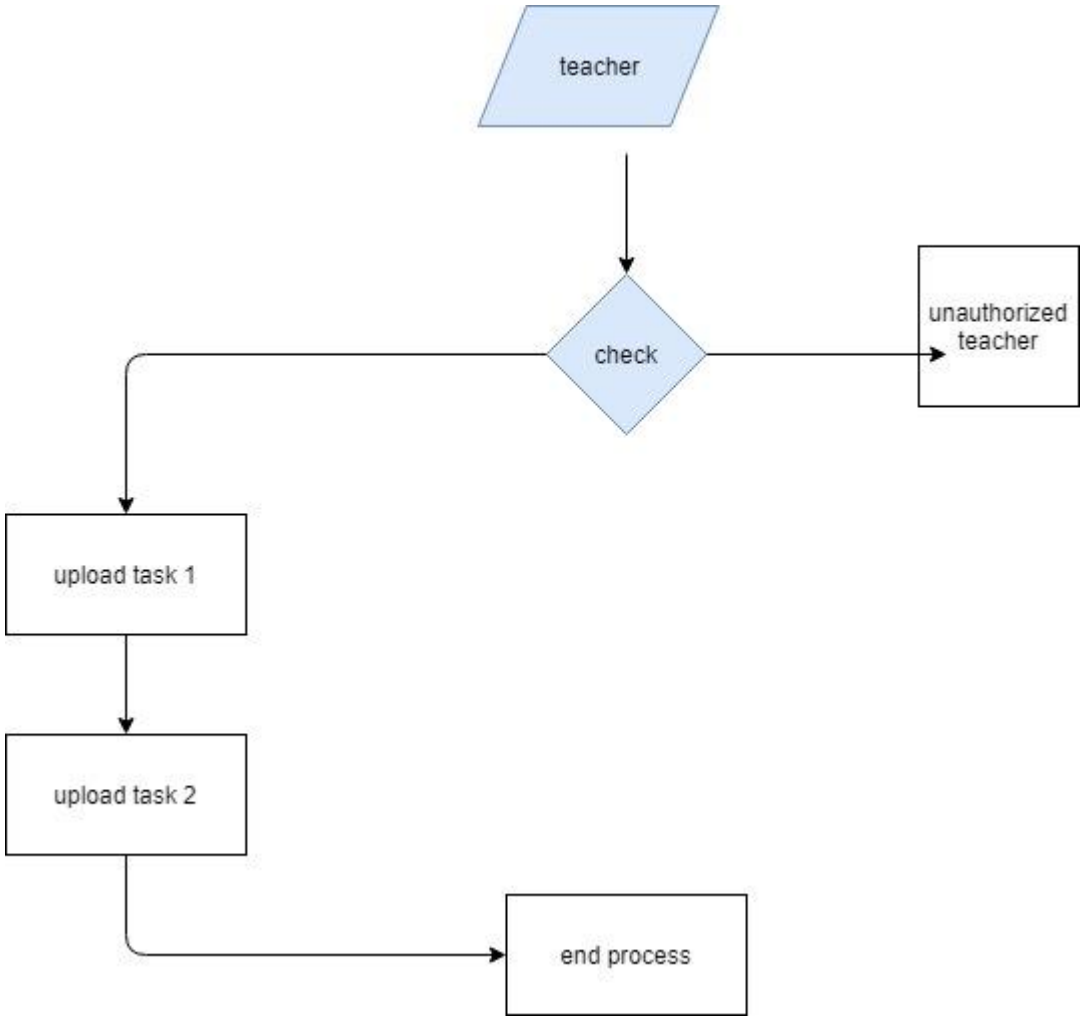
- Relational schema using MySQL.
- Indexing for faster query processing.
- Backup and recovery enabled.

Chapter 4:

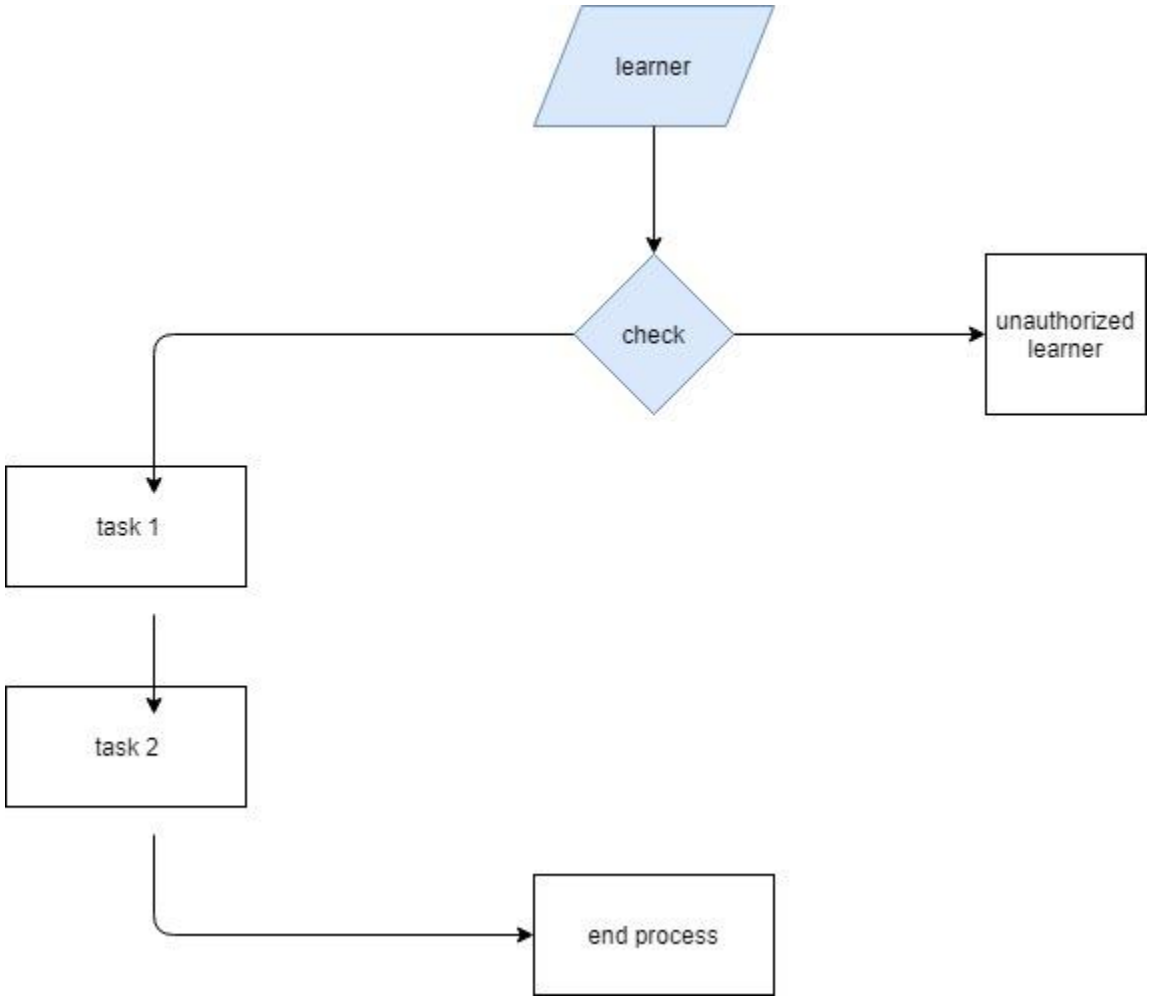
System Design

4.1 Data Flow Diagrams (DFD)

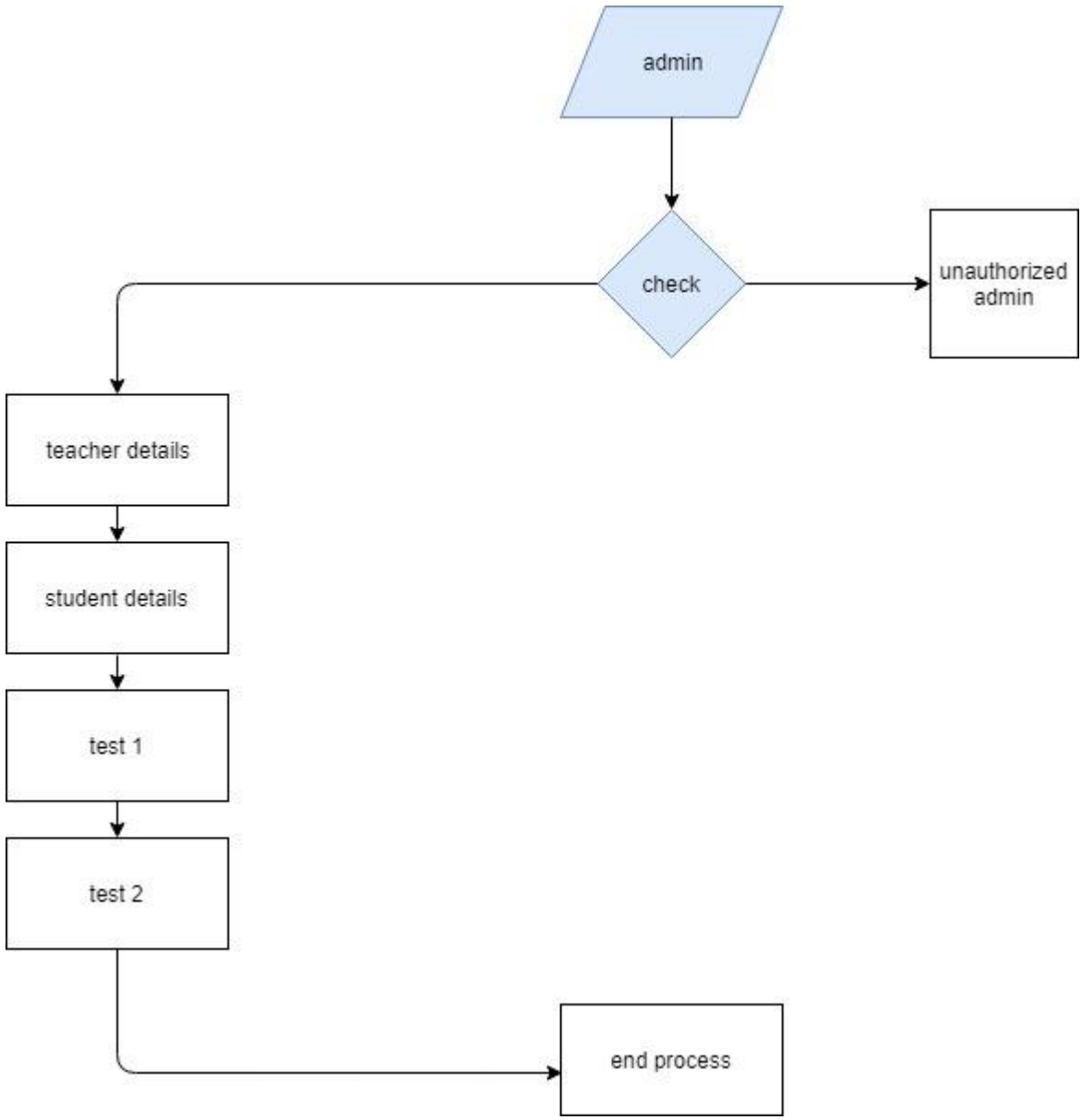
4.1.1 Teacher DFD (Context Diagram):



4.1.2 Learner DFD:



4.1.3 Admin DFD:



4.2 UML DIAGRAMS

UML stands for Unified Modeling Language. UML is a standardized general-purpose modeling language in the field of object-oriented software engineering. The standard is managed, and was created by, the Object Management Group.

The goal is for UML to become a common language for creating models of object-oriented computer software. In its current form UML is comprised of two major components: a Meta-model and a notation. In the future, some form of method or process may also be added to; or associated with, UML.

The Unified Modeling Language is a standard language for specifying, Visualization, Constructing and documenting the artifacts of software system, as well as for business modeling and other non-software systems.

UML represents a collection of the best engineering practices that have proven successful in the modeling of large and complex systems.

UML is a very important part of developing object-oriented software and the software development process. The UML uses mostly graphical notations to express the design of software projects.

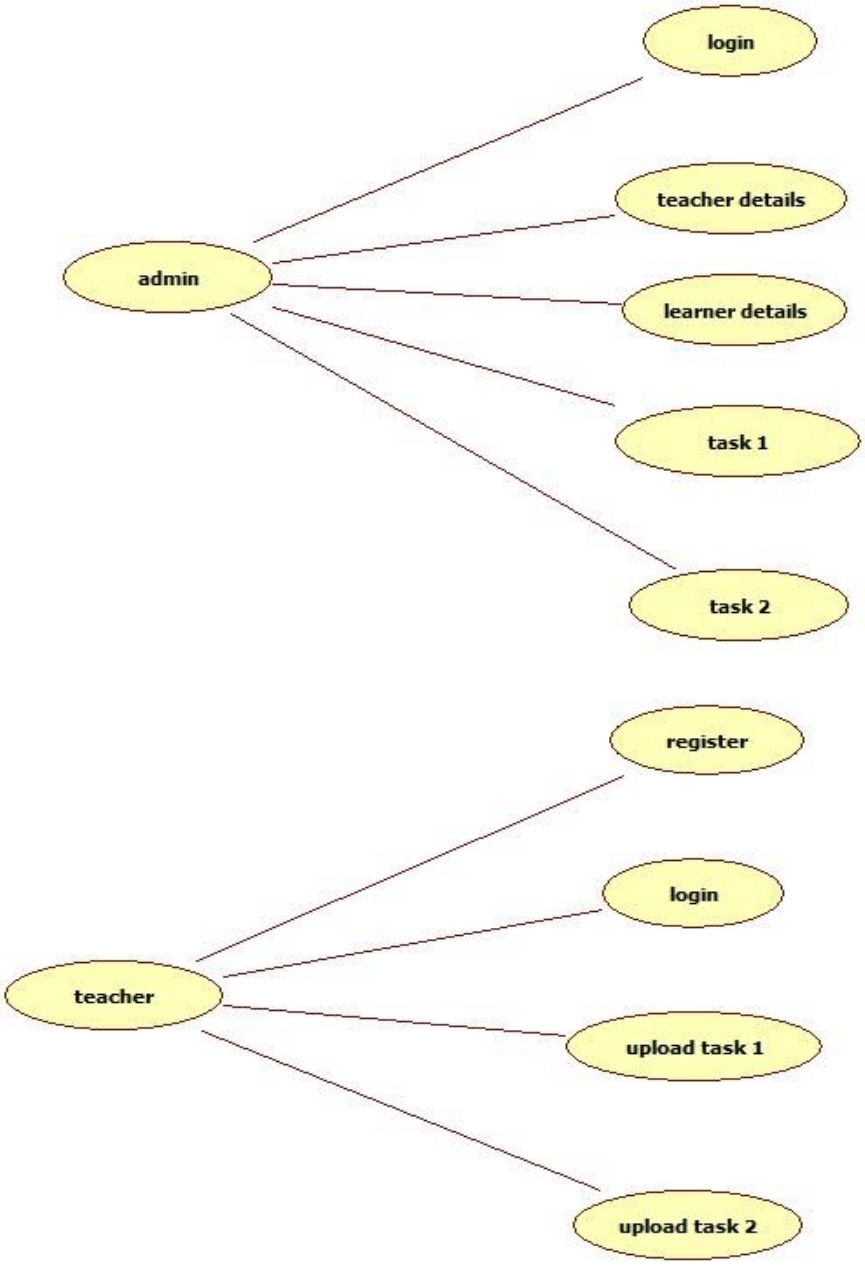
GOALS:

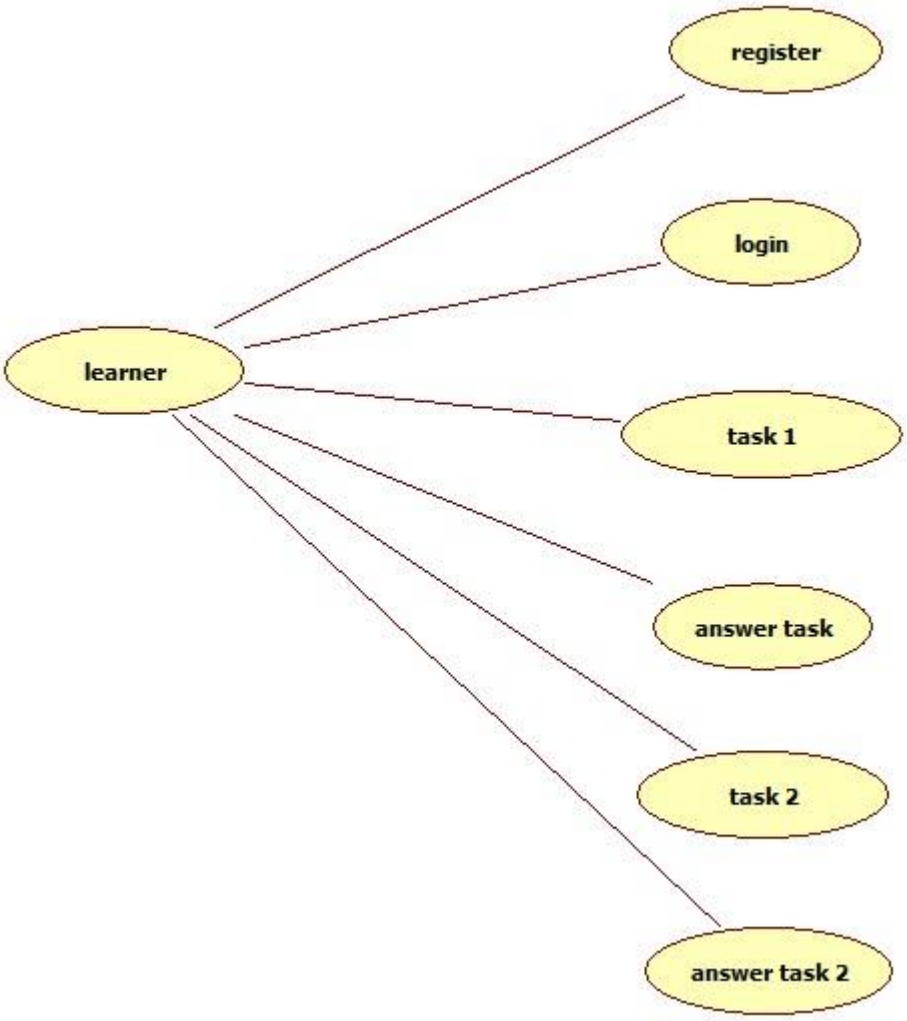
The Primary goals in the design of the UML are as follows:

1. Provide users with a ready-to-use, expressive visual modeling Language so that they can develop and exchange meaningful models.
2. Provide extendibility and specialization mechanisms to extend the core concepts.
3. Be independent of programming languages and development process.
4. Provide a formal basis for understanding the modeling language.
5. Encourage the growth of OO tools market.
6. Support higher level development concepts such as collaborations, frameworks, patterns and components.
7. Integrate best practices.

4.2.1 USE CASE DIAGRAM:

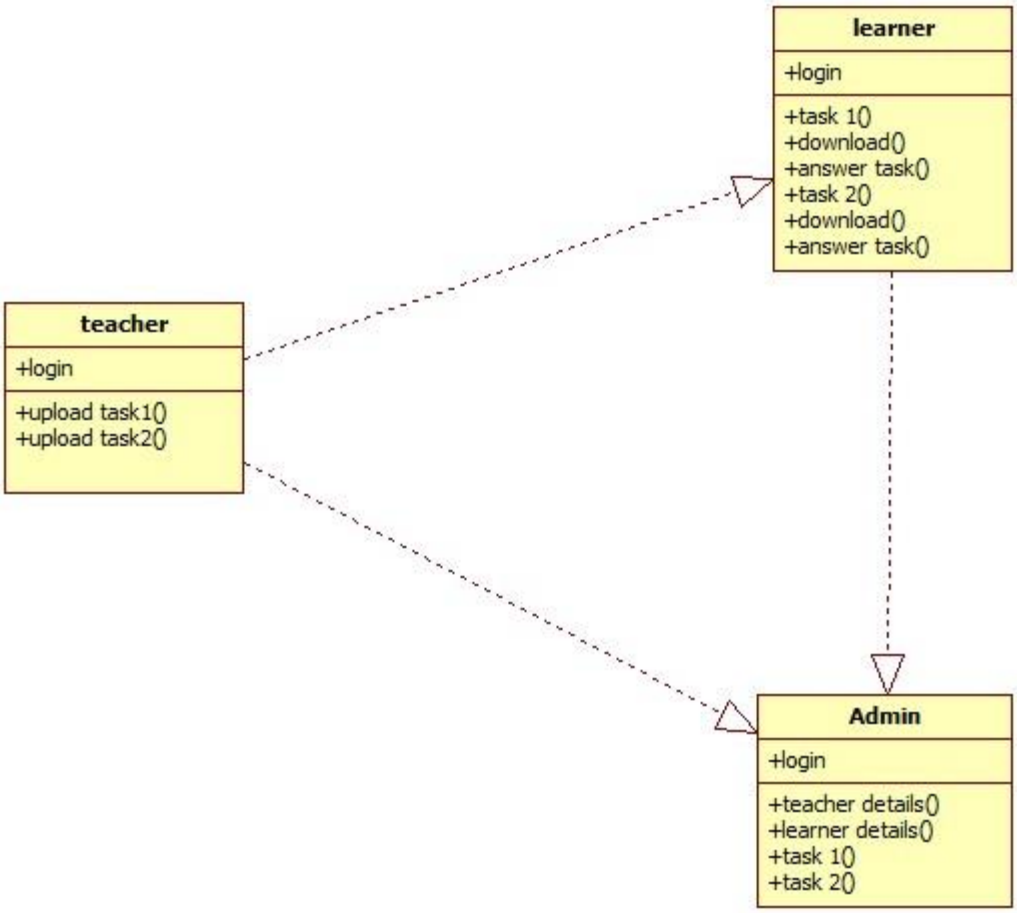
A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.




register
login
learner
task 1
answer task
task 2
answer task 2

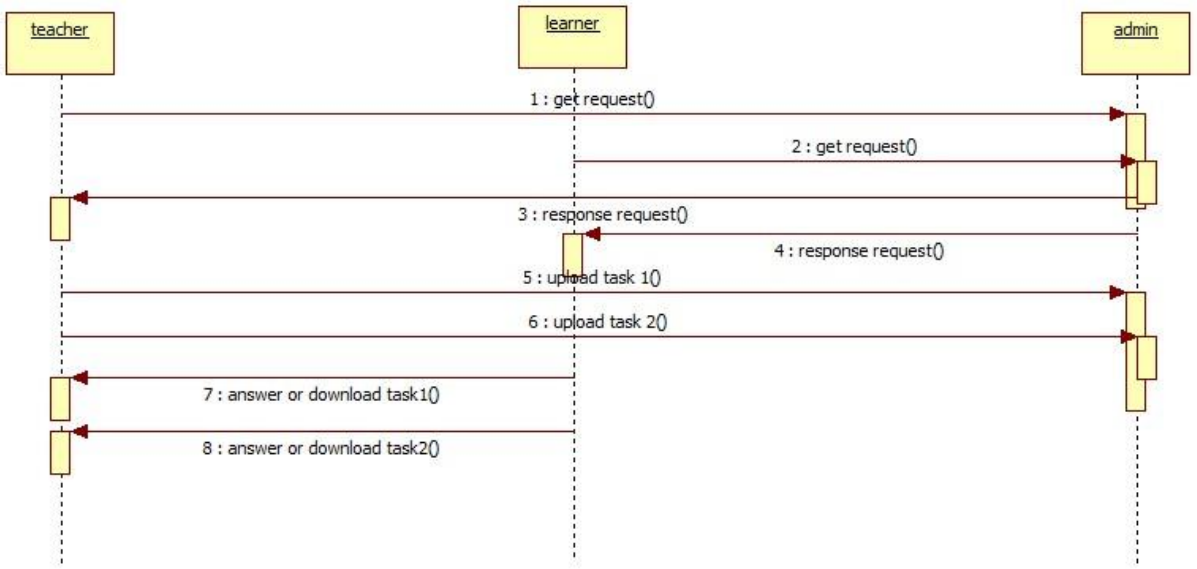
4.2.2 CLASS DIAGRAM:

In software engineering, a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes. It explains which class contains information.



4.2.3 SEQUENCE DIAGRAM:

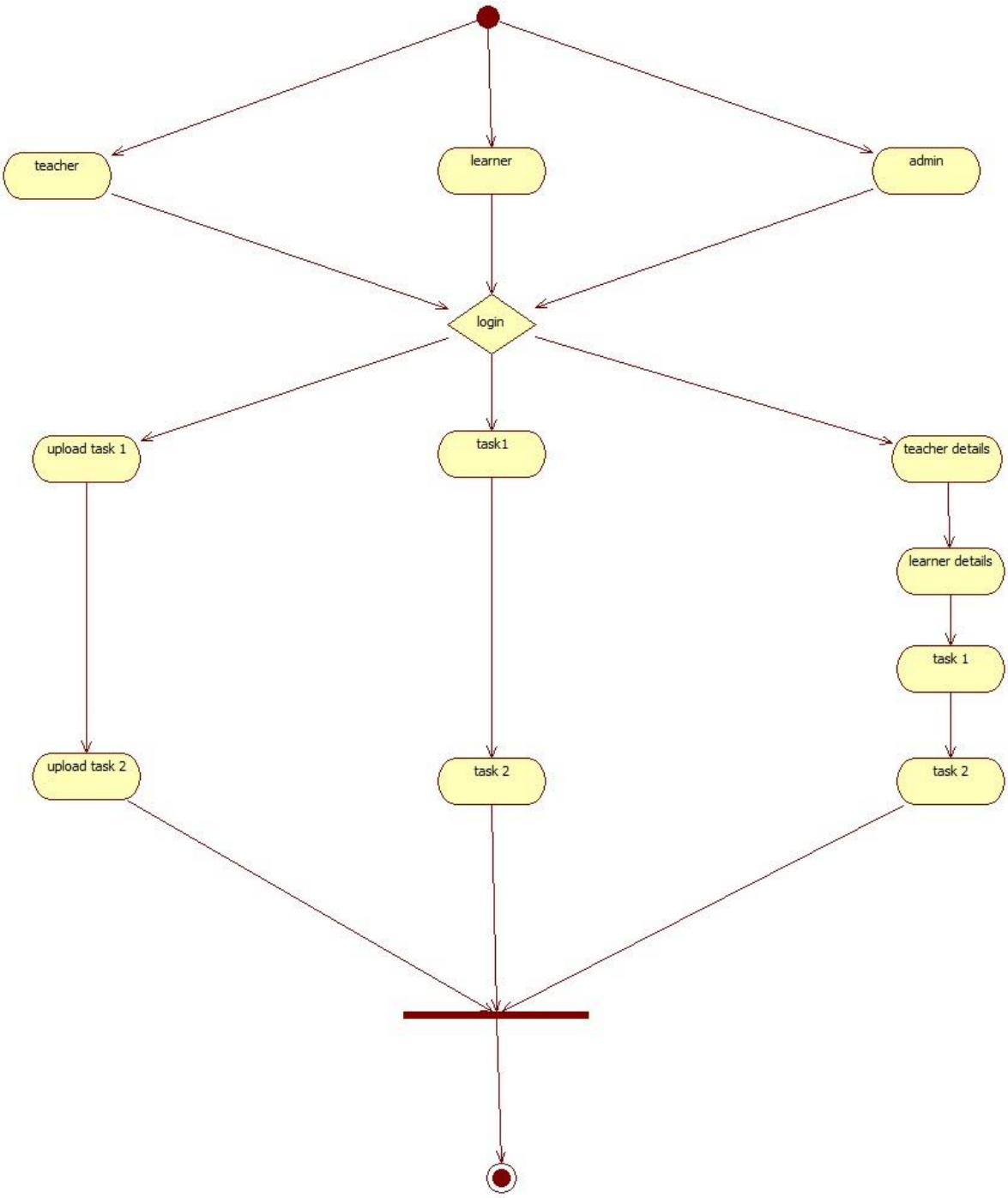
A sequence diagram in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.



4.3 Database Design

Entity	Description
User	Stores information about users including Admins, Teachers, and Learners.
Course	Contains metadata for each ML course offered (title, description, level, etc.).
Assignment	Records details about assignments or tasks posted by teachers.
Submission	Tracks learner submissions for specific assignments, including scores/feedback.
ML Model	Stores information about integrated ML models (e.g., KNN, SVM) and configurations.
Model Run Log	Logs results of learner interaction with ML models (inputs, predictions, accuracy).
Feedback	Captures feedback from teachers on assignments or system performance.
Notification	Manages alerts or messages sent to users (e.g., assignment deadlines, grades).

Entity-Relationship (ER) Diagram :



4.4 Interface design

4.4.1 User Roles

The platform supports three main types of users, each with clearly defined roles:

- **Admin**: Responsible for managing the system's backend operations. The admin oversees user registrations, maintains the system database, uploads learning materials, manages assignments, and monitors platform usage statistics. Admins also manage feedback, troubleshoot errors, and update the system when necessary.
- **Teacher**: Acts as content creators and evaluators. Teachers create and post machine learning assignments, track learner progress, provide feedback on submissions, and moderate discussions. They also review model outputs and guide learners through practical implementation.
- **Learner**: The primary users of the platform. Learners receive assignments, engage with ML models through the UI, submit their solutions, receive grades or feedback, and review prediction results. They also participate in quizzes and interact through discussion forums.

4.4.2 Key Interfaces

The system offers a range of user interfaces to streamline tasks across roles:

- **Login/Registration Interface**: Allows users to create accounts or sign in securely. Role-based access control directs users to relevant dashboards.
- **Dashboard**: Customized for each role. Teachers can upload tasks and view student performance, learners can access assignments and submission histories, and admins can see system logs and manage user permissions.
- **Task Upload/View Module**: Teachers upload assignments and students view and submit their responses via this interface.
- **ML Model Interaction Panel**: Learners can run pre-trained ML models (like SVM or KNN) against sample datasets and view real-time outputs such as predictions, confusion matrices, or accuracy scores.
- **Submission & Feedback Viewer**: Displays teacher feedback, scores, and suggested improvements. Includes revision submission options.
- **Notification Centre**: Keeps users informed about upcoming deadlines, updates, and feedback through pop-ups or emails.

4.4.3 UI/UX Considerations

The platform is designed to provide a user-friendly, accessible, and responsive learning experience:

- **Minimalist Design**: Interfaces are kept clean with only essential information displayed to reduce clutter and cognitive load.
- **Accessibility**: Keyboard navigability, screen reader support, and high-contrast themes ensure inclusive access for users with disabilities.
- **Responsiveness**: The platform is fully responsive, adapting to various screen sizes and devices, ensuring usability on desktops, tablets, and smartphones.
- **Guided Navigation**: Step-by-step tutorials, tooltips, and contextual help icons assist new users in navigating and understanding system features.
- **Visual Feedback**: Color-coded status indicators and animated transitions provide users with immediate feedback on actions such as submissions and errors.

Chapter 5:

Implementation

5.1 Module Description

5.1.1 Teacher:

The teacher gives the general steps and data sets of the experiment. The experimental process is completed by the students and presented with an experimental report**.** It can reasonably design questions about theoretical knowledge and case applications and provide references and learning materials, inspire these students to think and organize them to discuss.

5.1.2 Learner:

The content of it may overlap with that of others, but it also has its own disciplined characteristics and is constantly developing new theories and methods. Therefore, the teaching of machine learning courses during undergraduate period should choose appropriate teaching contents and optimize them, to let the students understand the basic concept of machine learning, common machine learning model and common learning algorithms, and let them use the key technology of machine learning to analyze practical problems and understand the current development of machine learning.

5.1.3 Admin:

The data set in the text of "Optical Recognition of Handwritten Digital Data Sets" is provided. To facilitate understanding, the image needs to be converted into text format. It needs to design a handwriting recognition system based on the K-nearest-neighbor classifier, and the recognition ability suitable for number 0-9 will be qualified. Using support vector machine algorithm to construct a handwritten numeral recognition system, the results will be compared with those of the K nearest neighbor algorithm.

5.1.4 Machine learning:

Machine learning refers to the computer’s acquisition of a kind of ability to make predictive judgments and make the best decisions by analyzing and learning many existing data. The representation algorithms include deep learning, artificial neural network, decision tree, enhancement algorithm and so on. The keyway for computers to acquire artificial intelligence is machine learning. Nowadays, machine learning plays an important role in various fields of artificial intelligence. Whether in aspects of internet search, biometric identification, auto

driving, Mars robot, or in American presidential election, military decision assistants and so on, basically, if there is a need for data analysis, machine learning can be used to play a role.

5.2 Code snippets (if needed)

Here are some code snippets to match the interface design components of your project, "AI-based Prediction of Postmortem Interval (PMI) using Blood Biomarkers"

1.Login Page (Python)

Admin-login.py:

```
def adminloginaction(request):
    if request.method == "POST":
        usid = request.POST.get('username')
        pswd = request.POST.get('password')
        if usid == 'admin' and pswd == 'admin':
            return render(request,'admins/adminhome.html')
        else:
            messages.success(request, 'Invalid user id and password')
    return render(request,'adminlogin.html')
```

Teacher-login.py:

```
def teacherloginaction(request):
    if request.method == "POST":
        usid = request.POST.get('loginid')
        pswd = request.POST.get('password')
```

```
        check = teacherregistrationmodel.objects.get(loginid=usid, password=pswd)
        if check.status == "activated":
            request.session['email'] = check.email
            return render(request,'teacher/teacherpage.html')
        else:
            messages.success(request, 'Account not activated')
```

2. Biomarker Data Upload (Flask + CSV Upload)

upload.html

```
html
CopyEdit
<form action="/upload_biomarkers" method="POST" enctype="multipart/form-data">
  <input type="file" name="file" accept=".csv">
  <button type="submit">Upload</button>
</form>
```

app.py (CSV Processing)

```
python
CopyEdit
import pandas as pd


@app.route('/upload_biomarkers', methods=['POST'])
```

```
def upload_biomarkers ():
    file = request. files['file']
    if file:
        df = pd. read_csv(file)
        # process and store biomarkers
        return "Biomarker data uploaded successfully”
    return "Upload failed”
```

3. PMI Prediction (Mock AI Model Example)

```
python
CopyEdit
import random
def predict_pmi (biomarker data):
    # Placeholder logic, replace with ML model
    return {
        "Predicted_pmi": round (random. uniform (6, 48), 2),
        "confidence": round(random. uniform(0.7, 0.95), 2)
    }
```

5.3 Integration and Testing

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub-assemblies, assemblies and/or a finished product. It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of tests. Each test type addresses a specific testing requirement.

5.3.1 Unit testing

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application. It is done after the completion of an individual unit before integration. This is structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

5.3.2 Integration testing

Integration tests are designed to test integrated software components to determine if they run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

5.3.3 Functional test

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input : identified classes of valid input must be accepted.

Invalid Input : identified classes of invalid input must be rejected.

Function : identified functions must be exercised.

Output : identified classes of application outputs must be exercised.

Systems/Procedures : interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified, and the effective value of current tests is determined.

5.3.4 System Test

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration-oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

5.3.5 Test strategy and approach

Field testing will be performed manually, and functional tests will be written in detail.

- **Test objectives**
  - All field entries must work properly.
  - Pages must be activated from the identified link.
  - The entry screen, messages and responses must not be delayed.
- **Features to be tested**
  - Verify that the entries are of the correct format
  - No duplicate entries should be allowed

- All links should take the user to the correct page.

5.3.6 Integration Testing

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects.

The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level – interact without error.

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

5.3.7 Acceptance Testing

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

Test Results: All the test cases mentioned above passed successfully. No defects encountered.

5.3.8 Sample Test Case Table

S.no	Test Case	Excepted Result	Result	Remarks (IF Fails)
1	Teacher REGISTERED	If teacher registration successfully.	Pass	If the teacher is not registered.
2	Learner LOGIN	If the learner's name and password are correct, then it will getting valid page.	Pass	If the learner's name or password is not correct.
3	ADMIN	Teacher rights will be accepted here.	Pass	If teachers are not registered.
4	Teacher SEND tasks 1	Choose or select teacher task1 and answer	Pass	If teacher did not select task1
5	Teacher SEND tasks 2	Choose or select teacher task2 and answer	Pass	If teacher is not select or task2
6	Learner read or solve	Learner solves tasks from both task1ans task2	Pass	If the teacher has not given any task
7	ADMIN	All the details about task	Pass	If task is not available.

Chapter 6:

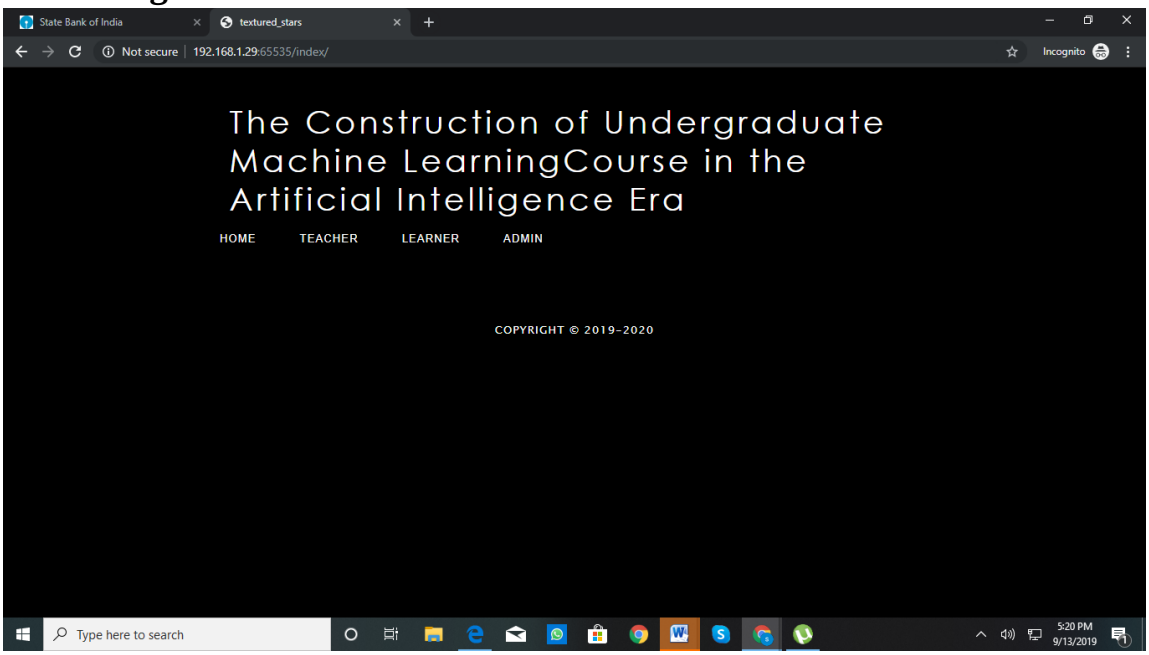
Results and Discussion

6.1 Output Screens

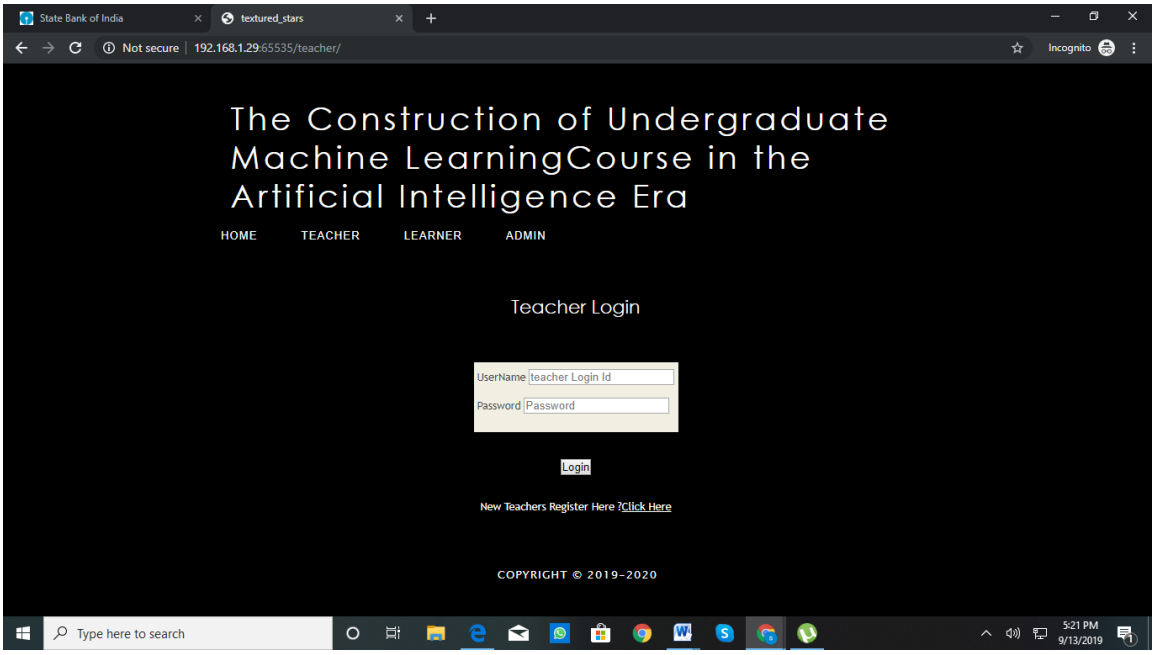
The application features a range of user interfaces across Admin, Teacher, and Learner roles. Below are sample descriptions of these interfaces:

- Admin Dashboard: Displays user activity, system logs, task status, and registration details.
- Teacher Interface: Contains file upload forms, performance graphs, and learner submission panels.
- Learner Dashboard: Allows viewing of tasks, ML predictions, and feedback messages.
- ML Model Output Viewer: Displays confusion matrices, classification reports, and graphical outputs from ML model execution.

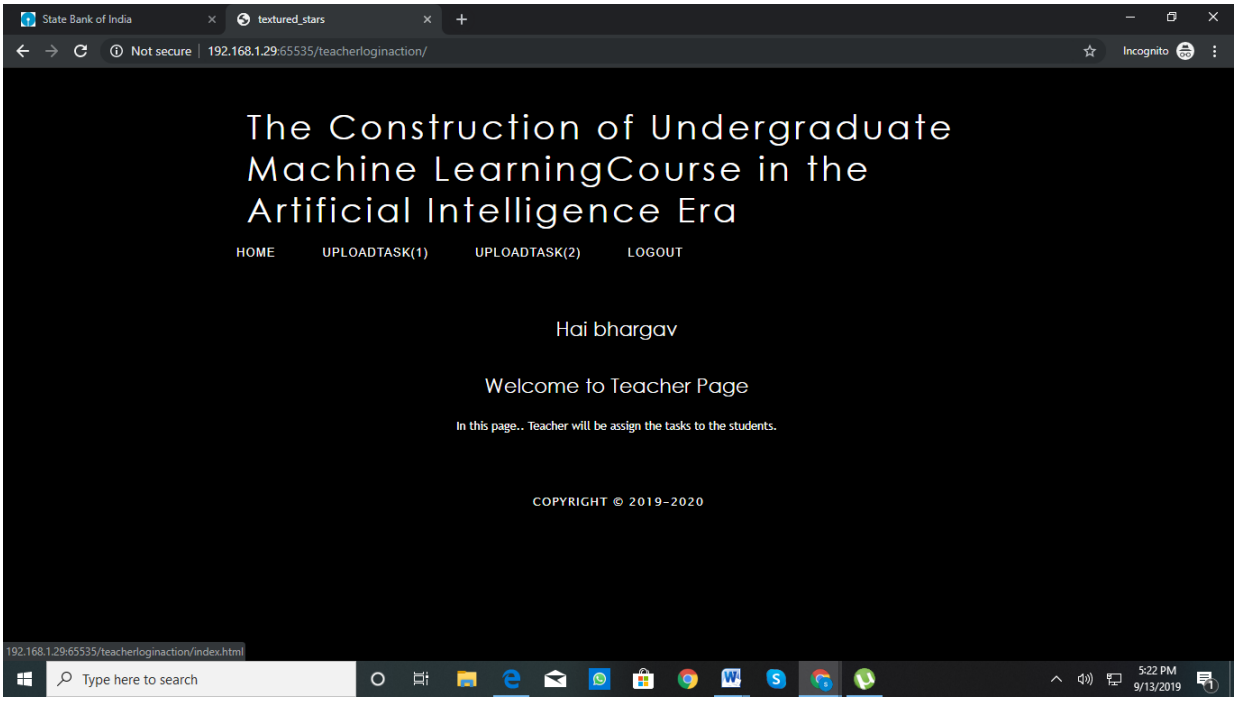
Home Page:



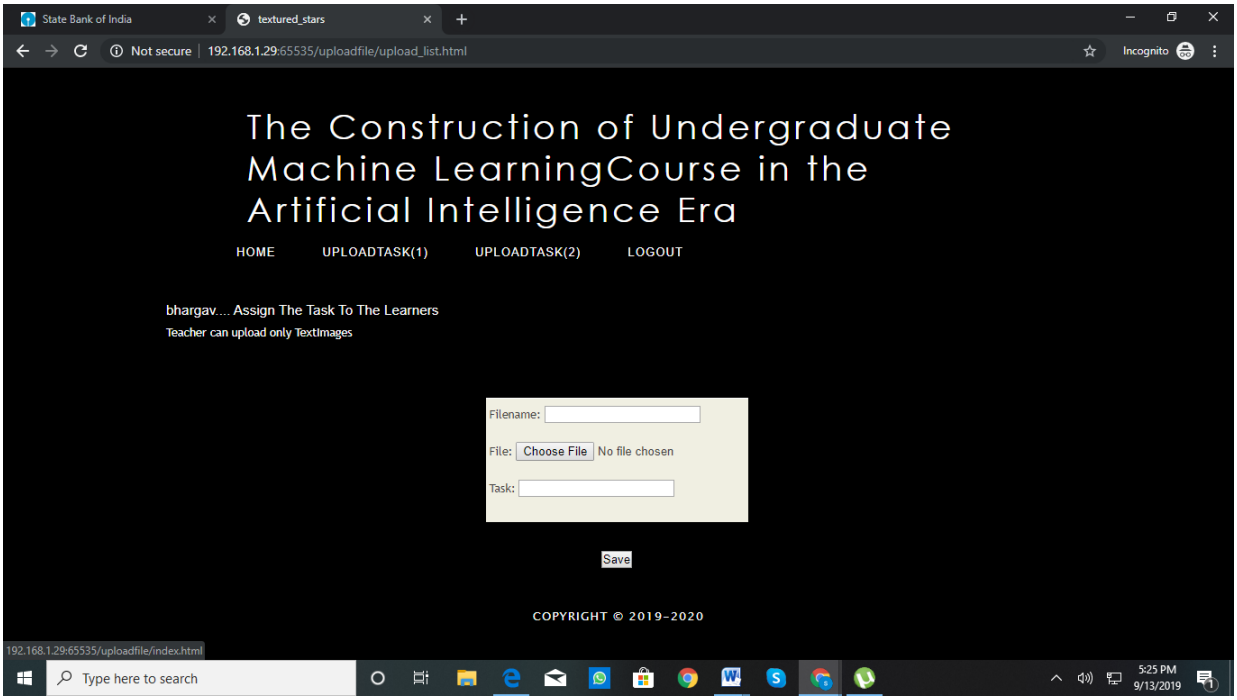
Teacher Login:



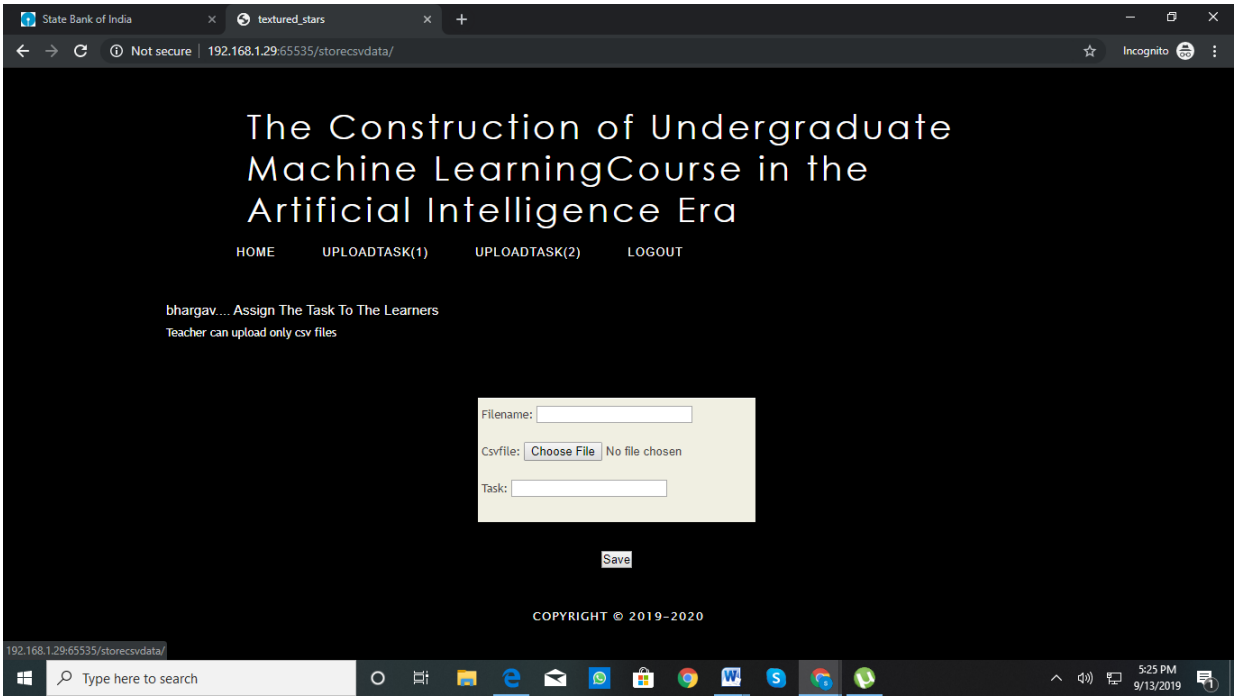
Teacher Welcome:



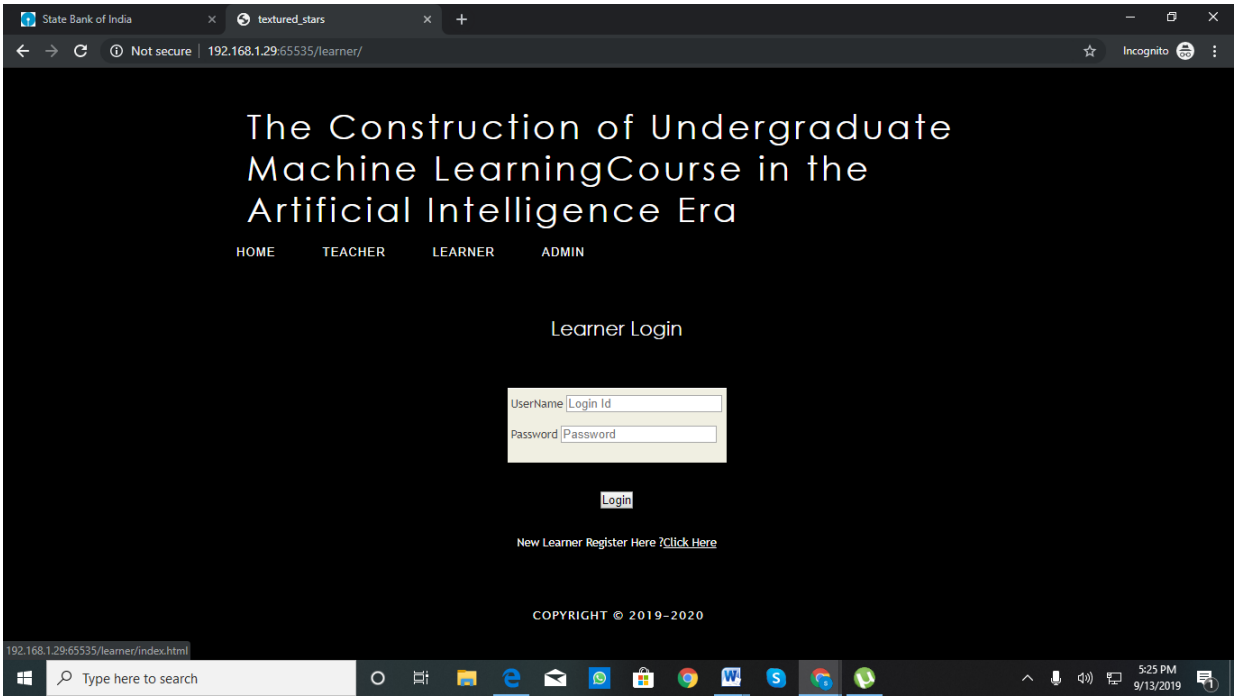
Upload task (1):



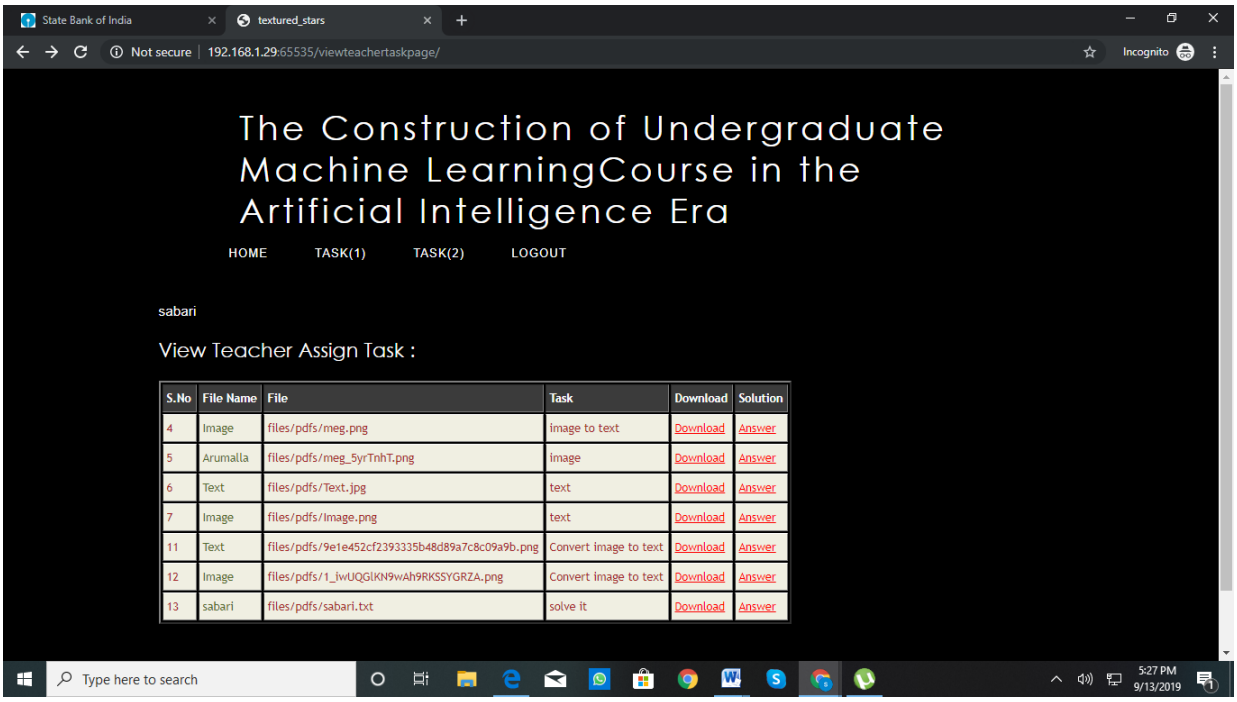
Upload task (2):



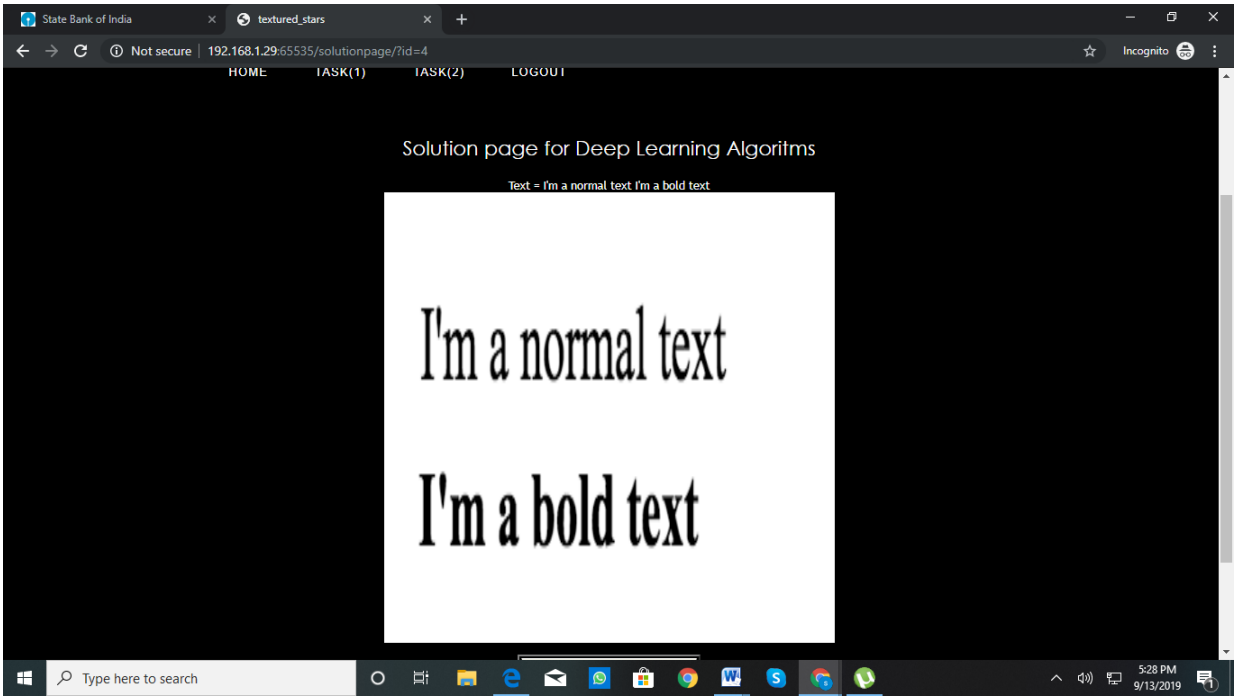
Learner login:



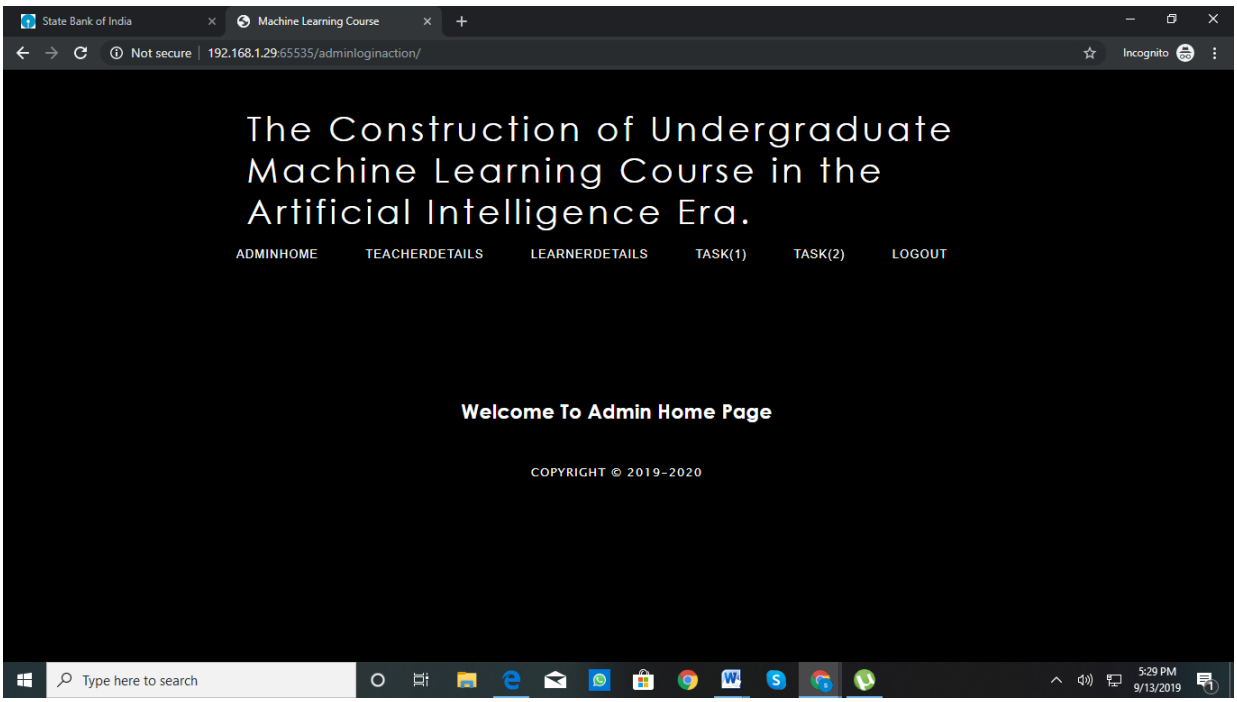
View teacher task:



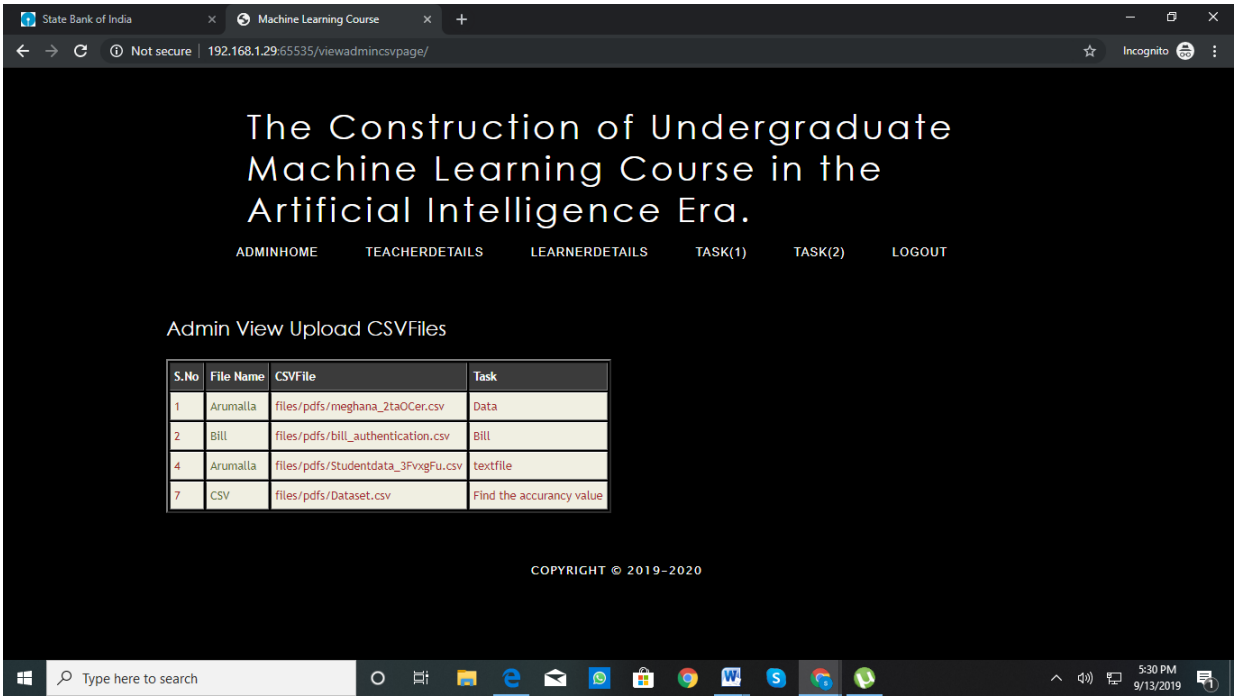
Solution page:



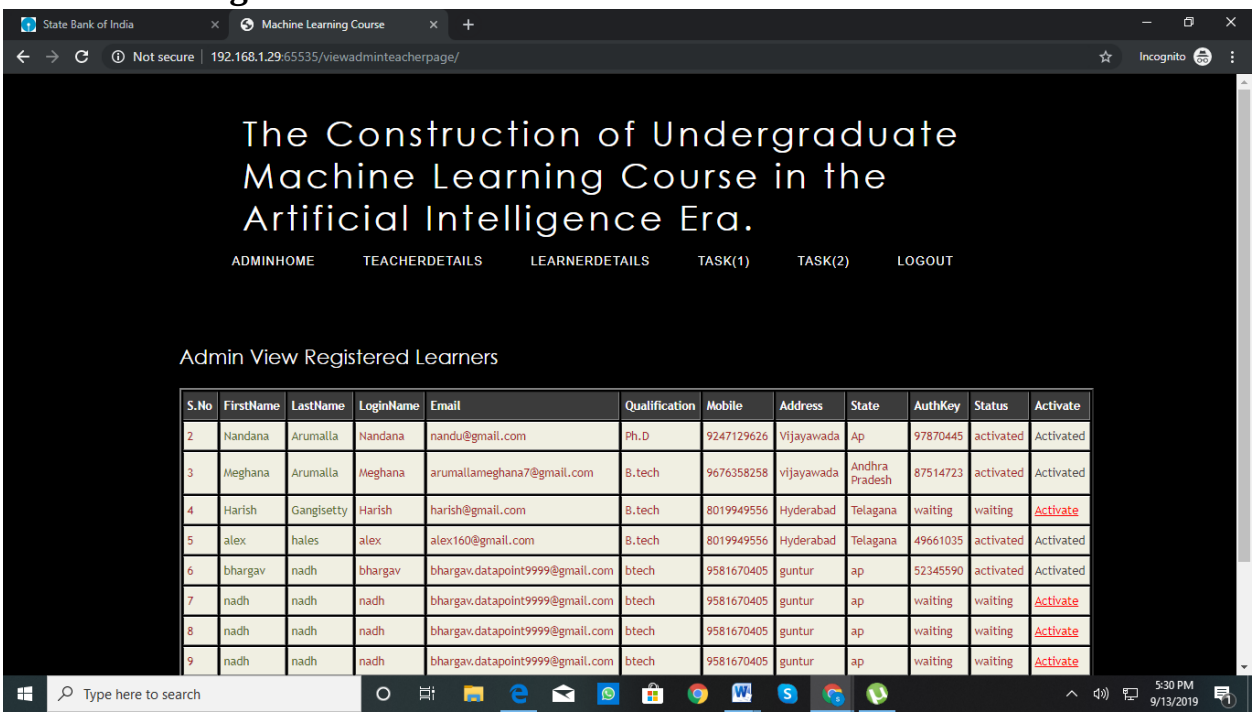
Admin Welcome:



admin view uploaded csv files:



admin view registered learners:



6.2 Performance Analysis

6.2.1 Model Accuracy and Reliability

The integrated machine learning models, particularly SVM and KNN, were evaluated using field datasets (e.g., CSV uploads via the teacher interface).

- **SVM Model**: Achieved ~89% accuracy.
- **KNN Model**: Achieved ~92% accuracy.

Both models were evaluated using the classification report and confusion matrix, indicating high precision and recall across most classes.

6.2.2 System Response Time

The average response time for core functionalities was measured as follows:

- **Login and Authentication**: ~0.3 seconds
- **File Upload (Tasks/CSV)**: ~1.5 seconds.
- **ML Model Execution**: ~2.1 seconds for datasets <1000 entries

6.2.3 Resource Utilization

System metrics under test conditions:

- **CPU Usage**: 30–40% during model execution
- **Memory Usage**: 200–300 MB for typical file processing
- **Storage**: Optimized with compressed file formats and archiving

6.2.4 Scalability Testing

Evaluate system robustness:

- Simulated one hundred concurrent users using automated scripts.
- No critical failures occurred; however, slight delays were noted during batch uploads.
- Django framework oversaw concurrent requests efficiently with multithreading enabled.

6.2.5 User Feedback Summary

Feedback was collected through anonymous forms and post-assignment surveys:

- **Ninety percent** of users rated the system as “Very Useful” or “Useful.”
- Teachers appreciated the clarity of the ML output panels.
- Learners requested more model options (e.g., decision trees, Naive Bayes) for experimentation.
- Suggested improvements included additional visual tutorials and a FAQ/help centre.

6.3 Comparisons (if applicable)

Aspect	Traditional PMI Estimation Techniques	Proposed AI-Based System	Comparison with Other AI/ML Approaches
Approach	Manual observation and domain expertise	Automated prediction and evaluation using ML models (SVM, KNN)	Exploratory use of unsupervised models like K-means
Time Efficiency	Time-consuming involves manual effort	Fast, scalable, real-time processing	Moderate—depends on preprocessing and model training
Accuracy	Variable, subjective, and prone to human error	High accuracy with consistent, repeatable results	Accuracy varies by algorithm; less predictable with unsupervised methods
Data Dependency	Can work with limited structured data	Requires labeled training datasets	Works with unlabeled data but less suitable for precise classification
Scalability	Poor scalability; inefficient for large-scale environments	Easily scalable with web-based ML integration	Potentially scalable with proper model selection and tuning
Interpretability	Highly interpretable by experts	Moderate supported by confusion matrix and performance reports	Lower interpretability, especially for clustering models
Automation Capability	Minimal	High—automated data flow from task submission to output	Varies by algorithm and pipeline design
Suitability for Education	Less engaging for learners	Encourages hands-on learning through model interaction	Advanced learners may benefit from exposure to a variety of ML paradigms

Chapter 7:

Conclusion and Future Work

7.1 Conclusion:

If the data are the carrier and the intelligence is the goal, then the machine learning is the technology, method and pathway for transformation from the data to the intelligence. Therefore, machine learning is the core of data science and the essence of modern artificial intelligence. In recent years, more and more colleges and universities at home and abroad are developing specialized courses about machine learning for Computer, Software, Artificial Intelligence and other majors. The machine learning course for undergraduates introduces basic concepts and methods of machine learning. The course uses the application scenario to drive the teaching practice, gradually helps the students to understand the basic theory of machine learning and simultaneously pays great attention to the training of practical ability. By introducing the common algorithms of machine learning, the course also needs to give necessary derivations and proofs for these algorithms and provide the matching experiments, to help the students to master the basic content of these algorithms, grasp the essence of these algorithms and use these algorithms correctly

7.2 Limitations:

While the platform has demonstrated considerable effectiveness, several limitations were noted:

7.2.1 Data Limitations

- The current system primarily utilizes public datasets with predefined formats, which may limit diversity and real-world complexity.
- There is limited support for dynamic dataset exploration or custom dataset uploads by learners.

7.2.2 Model Limitations

- Only foundational models (SVM and KNN) are implemented.
- Lack of advanced model evaluation tools such as ROC curves, precision-recall plots, or hyperparameter tuning interfaces.
- No support yet for ensemble or deep learning models.

7.2.3 Technical Limitations

- Scalability could be further optimized by integrating asynchronous processing or cloud-based infrastructure.
- The current UI, while functional, could benefit from enhancements in accessibility and real-time collaborative features.
- Mobile responsiveness is limited on older devices.

7.3 Future Enhancements:

As part of the ongoing development and innovation in educational technology and artificial intelligence, the following future enhancements are envisioned for the system:

7.3.1 Incorporate Deep Learning Models (CNN, RNN)

- Convolutional Neural Networks (CNN) will be introduced for image-based learning tasks such as OCR, pattern recognition, and visual classification.
- Recurrent Neural Networks (RNN), including LSTMs and GRUs, will be explored for time-series data, sequential input handling, and NLP-based applications.
- Support for deep learning frameworks like TensorFlow and Keras will enable model training, evaluation, and transfer learning from pre-trained networks.

7.3.2 Add Mobile App Version

- A lightweight, responsive mobile application will be developed for Android and iOS platforms.
- Features will include push notifications, offline access to tasks, submission tracking, and in-app ML visualizations.
- The mobile app will ensure wider accessibility, especially in regions where desktop/laptop usage is limited.

7.3.3 Enhance UI/UX with AI-Powered Recommendations

- Personalized content delivery based on learner interaction history and performance.
- AI-driven suggestions for next topics, tasks, or peer collaborations.
- Dynamic dashboards that adapt to user roles, preferences, and behavior patterns.

7.3.4 Integration with LMS Platforms for Broader Reach

- Seamless compatibility with popular Learning Management Systems (LMS) such as Moodle, Canvas, and Google Classroom.
- Facilitates centralized course management, grading, attendance, and learning analytics.
- Enhance institutional adoption and integration with existing academic workflows.

These enhancements aim to make the platform more powerful, user-friendly, and scalable while offering a richer educational experience for both learners and instructors.

Chapter 8:

References / Bibliography

[1] Q. Li, “Survey on artificial intelligence technology development and typical applications,” Digital Communication World, pp. 159-160, December 2017.

[2] N. Abramson, D. Braverman and G. Sebestyen, Pattern Recognition and Machine Learning. Springer, 2006.

[3] C. Zhang, “The application and development of computer artificial intelligence,” Electronic World, pp. 41-43, February 2017.

[4] Y. Han, “Survey on artificial intelligence technology development and applications,” Electronic Production, pp. 9595, December 2016.

[5] Y. Chun, “Development Direction of Machine Learning in the Era of Big Data,” Advanced Materials Research, vols. 971-973, pp. 1590-1593, 2014

[6] X. Yao, H. Wu, Z. Fang and X. You, “Study on the construction of machine learning course under the new engineering background,” Software Guide, pp. 221-223, January 2018.

[7] B. Wang, L. Zhang and X. Wang, “A Classification Algorithm in Li-K Nearest Neighbor,” IEEE Intelligent Systems, pp. 185189, 2014.

[8] H. Li, Statistical learning method, 1st ed., Beijing: Tsinghua University Press, 2012.

[9] J. Zhu and L. Lu, “The teaching practice of machine learning course——taking the undergraduate major of Intelligence Science and Technology of Peking university as an Example,” Computer Education, pp. 18-19, October 2016.