A

MAJOR PROJECT REPORT

On

Students Performance Prediction In online courses using machine learning Submitted to Jawaharlal Nehru Technological University for the partial

Fulfilment of the requirement for the award of the degree of

BACHELOR OF TECHNOLOGY

In

COMPUTER SCIENCE AND ENGINEERING

BY

GAJJALA PRANAYNATH	:	20Q91A05F6
NERUDI NIKITHA	:	20Q91A05E2
CHOUTAPELLY VARUN	:	20Q95A05C5
CHOWDARY PREETHAM	:	20Q91A05H3

Under the guidance of

Dr. V. VIVEKANANDHAN

ASSOCIATE PROFESSOR

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

MALLAREDDY COLLEGE OF ENGINEERING

(Approved by AICTE-Permanently Affiliated to JNTU-Hyderabad) Accredited by NBA & NAAC, Recognized section 2(f) & 12(B) of UGC New Delhi ISO 9001:2015 certified Institution.

Maisammaguda, Dhulapally (Post via Kompally), Secunderabad- 500100.

2023-2024

MALLA REDDY COLLEGE OF ENGINEERING

(Approved by AICTE-Permanently Affiliated to JNTU-Hyderabad)

Accredited by NBA & NAAC, Recognized section 2(f) & 12(B) of UGC New Delhi ISO 9001:2015 certified Institution.

Maisammaguda, Dhulapally (Post via Kompally), Secunderabad -500100.

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CERTIFICATE

This is to certify that the Major Project report on "Students Performance Prediction In online courses using machine learning" is successfully done by the following students of Department of Computer Science & Engineering of our college in partial fulfilment of the requirement for the award of B.Tech degree in the year 2023-2024. The results embodied in this report have not been submitted to any other University for the award of any diploma or degree.

GAJJALA PRANAYNATH	:	20Q91A05F6
NERUDI NIKITHA	:	20Q91A05E2
CHOUTAPELLY VARUN	:	20Q95A05C5
CHOWDARY PREETHAM	:	20Q91A05H3

Submitted for the viva voce examination held on: ____________________________

INTERNAL GUIDE
Dr. V. VIVEKANANDHAN
Associate Professor

HOD
Dr. G. Radha Devi
Assistant Professor &
HEAD of the CSE Dept

Internal Examiner

External Examiner

DECLARATION

We, the final year students are hereby declaring that the minor project report entitled "Students Performance Prediction In online courses using machine learning " has done by us under the guidance of Dr. V. VIVEKANANDHAN Associate Professor, Department of CSE is submitted in the partial fulfilment of the requirements for the award of the degree of **BACHELOR OF TECHNOLOGY** in **COMPUTER SCIENCE AND ENGINEERING.**

The Results embedded in this project report have not been submitted to any other University or institute for the award of any degree or diploma.

SIGNATURE OF THE CANDIDATE

GAJJALA PRANAYNATH	-	20Q91A05F6
NEERUDI NIKITHA	-	20Q91A05E2
CHOUTAPELLY VARUN	-	20Q91A05C5
CHOWDARY PREETAM	-	20Q91A05H3

DATE:30/05/2024

PLACE: Maisammaguda

ACKNOWLEDGEMENT

First and foremost, we would like to express our immense gratitude towards our institution Malla Reddy College of Engineering, which helped us to attain profound technical skills in the field of Computer Science & Engineering, there by fulfilling our most cherished goal.

We are pleased to thank Sri Ch. Malla Reddy, our Founder, Chairman MRGI, Sri Ch. Mahender Reddy, Secretary, **MRGI** for providing this opportunity and support throughout the course.

It gives us immense pleasure to acknowledge the perennial inspiration of Dr. M. Ashok our beloved principal for his kind co-operation and encouragement in bringing out this task.

We would like to thank Dr. T. V. Reddy our vice principal, Dr. G. Radha Devi HOD, CSE Department for their inspiration adroit guidance and constructive criticism for successful completion of our degree.

We would like to thank DR. V. Vivekanandhan Associate Professor our internal guide, for her valuable suggestions and guidance during the exhibition and completion of this project.

Finally, we avail this opportunity to express our deep gratitude to all staff who have contribute their valuable assistance and support making our project success.

GAJJALA PRANAYNATH - 20Q91A05F6

NEERUDI NIKITHA -20Q91A05E2

CHOUTAPELLY VARUN -20Q91A05C5

CHOWDHARY PREETHAM -20Q91A05H3

ABSTRACT

Advances in Information and Communications Technology (ICT) have increased the growth of Massive open online courses (MOOCs) applied in distance learning environments. Various tools have been utilized to deliver interactive content including pictures, figures, and videos that can motivate the learners to build new cognitive skills. High ranking universities have adopted MOOCs as an efficient dashboard platform where learners from around the world can participate in such courses.The students learning progress is evaluated by using set computer marked assessments. In particular, the computer gives immediate feedback to the student once he or she completes the online assessments. The researchers claim that student success rate in an online course can be related to their performance at the previous session in addition to the level of engagement. Insufficient attention has been paid by literature to evaluate whether student performance and engagement in the prior assessments could affect student achievement in the next assessments. In this paper, two predictive models have been designed namely students' assessments grades and final students' performance. The models can be used to detect the factors that influence students' learning achievement in MOOCs. The result shows that both models gain feasible and accurate results.The lowest RSME gain by RF acquire a value of 8.131 for students assessments grades model while GBM yields the highest accuracy in final students' performance, an average value of 0.086 was achieved.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF SCREENSHOTS

LIST OF TABLES

LIST OF ABBREVATIONS

MOOCs	Massive Open Online Courses
cMOOCs	connectivist Massive Open Online Courses
XMOOCs	eXtended Massive Open Online Courses

CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION

Massive Open Online Courses (MOOCs) is one of the most widespread e-learning platforms. The MOOCs present the course using digital tool materials in various forms such as visual, audio, video and plain text. Most students prefer using video lectures to understand the contents of lessons over thoroughly reading plain text documents. The interactive video in the MOOCs could reduce students' stress, help them to feel relaxed and learn quickly.

MOOCs can be classified into two distinct types mainly, connectivist Massive Open Online Courses (cMOOCs) and eXtended Massive Open Online Courses (xMOOCs).The xMOOCs are learning paradigm based on the principles of cognitivist behaviorist theory.The structure of the courses is similar to the traditional course where the syllabus consists of a set of video lectures and a set of multiple choice quizzes in addition to the final exam. The video lectures featuring the course instructor reviewing the content of the previous online lesson are released weekly. The participants can watch and pause the video at their own pace. Moreover, the students can socially interact with other participants and the instructor through posting in discussion forums.The instructors usually post questions, provide task solutions and reply to student questions via these discussion forums; as a consequence the discussion forums play a vital role in enhancing the course quality and make online sessions collaborative and engaging.

The cMOOCs are a new learning model based on connectivist learning theory. With the connectivism approach, the instructor would not provide the actual learning material; the students get the course syllabus by asking the questions and sharing this information with other participants. References posit the learning strategy of cMOOCs focused on a collaborative approach in which learning material combined remix, repurposable and provided, forwarded to other students.

With cMOOCs, it is impossible to involve expertise to assess the students' knowledge whereas in xMOOCs, university lecturers can evaluate the students' knowledge through the use of computer-marked assessment feedback. In particular, the computer gives immediate feedback to the student when he completes the online assessment. The learner, upon successful completion, will be awarded their certification in xMOOCs.The cMOOCs do not include a formal assessment. Hence, universities are not considered cMOOCs as an official course.

With rapid advancements in technology, artificial intelligence has recently become an effective approach in the evaluation and testing of student performance in online courses. Many researchers applied machine learning to predict student performance in, however few works have been done to examine the trajectories performance. As a result, educators could not monitor the real-time students learning curve. Two sets of experiments are conducted in this work. In the first set of eperiements, regression analysis is implemented for estimation of students' assessment scores. The student past and current activities in addition to past performance are employed to predict student outcome. In the second set of experiments, supervised machine learning method has been utilized to predict long-term student performance. Three types of candidate predictors have been considered firstly behavioral features, followed by temporal and demographic features. The proposed models offer new insight into determining the most critical learning activity and assist the educators in keeping tracking of timely student performance. To the best of our knowledge, student performance has been evaluated in online course using only two targets: "success" and "fail". Our model predicts the performance with three-class labels "success", "fail" and "withdrew".

1.2 OBJECTIVE

The objective of predicting students' performance in online courses using machine learning is multifaceted:

Early Intervention: Identify students who are at risk of underperforming or dropping out early in the course so that educators can provide timely support and interventions to help them stay on track.

Personalized Learning: Tailor the learning experience for individual students based on their predicted performance, providing targeted resources, feedback, and guidance to address their specific needs and learning styles.

Resource Allocation: Optimize resource allocation by directing instructional support and resources towards students who are most likely to benefit from them, maximizing the impact of limited educational resources.

Course Improvement: Gain insights into factors influencing student performance, such as engagement levels, learning behaviors, and course design, to inform the continuous improvement of online courses and teaching practices.

Student Success: Ultimately, the overarching objective is to enhance student success and learning outcomes in online courses by leveraging predictive analytics to identify and address challenges proactively, thereby promoting academic achievement and retention.

1.3 METHODOLOGY

A) Data Description

The OULAD dataset was captured from the Open University Learning Analytics Dataset (OULAD) repository. The open university in the UK delivers the online course in various topic for undergraduate and postgraduate students in the period between 2013-2014. The main composite table called “studentInfo” is linked to all tables.The"studentInfo "table Includes information relevant to students’ demographic characteristics.The information related to students perfromnce are collected in “Assessments” and Student Assessment tables. The table“Assessments” contains information about the number, weight and the type of assessments required for each module. In general, each module involves a set of assessments,followed by the final exam. The assessments are Tutor Marked Assessment (TMA), Computer Marked Assessment (CMA). The final average grade is computed with the sum of all assessments (50%) and final exams (50%). The “Student Assessment” table involves information relating to student assessment results, such as the date of the submitted assessment and the assessment mark. The “Student Registration” table contains information about the date the students registered and unregistered in a particular module. The overall date is measured by counting numbers of unique days that students interact with courses until the course ends. In Open University online courses, students are able to access a module even before being a student of the course; however, it is not possible to access the course post-course closure date. The students' information related to their interaction with digital is store in learners Virtual Learning Environment dataset.

B) Feature Extraction

The features extraction was undertaken in our previous work. The VLE activities were used to generate behavioural features. For each student at a specific time t, two features are extracted: the number of sessions () and the number of clicks (). The behavioral features can be divided as either static or dynamic. * Static Behavioral features: These are a set of behavioral features that correspond to student activities since the first time they engaged in the course till last day they quit the course. Let us consider the set $\in\mathbb{R}$, in which , , represents the jth activity of the ith student at time t. S is a set of students denoted as an n-dimensional vector [S1 …., Sn], where n is the number of students. Furthermore, M is defined as an m-dimensional vector that represents VLE learning activity types, M = [M1 …., Mm], where m is the number of learning activities that the ith student is assigned. * Dynamic Behavioral Features: These are a set of behavioral features that vary over time. Let t be a sequence of disjointed time intervals, where t ∈ [1,6]. To represent all student activities at time t, we define the type of student’s activity records as the vector , =[, , , , , ,…, , , ,]. Here jth denotes learning activity that is undertaken at time t by student Si, such that j = 1, the, m; where m is given as the number of learning activities.

CHAPTER 2

LITERATURE SURVEY

2.1 LITERATURE SURVEY

Student performance is a key indicator of measuring learning progress in a virtual learning environment. Researchers have adopted various methods to monitor performance. In this section, the literature review in the relevant area is presented.

The Factor Analysis Model (FAM) was proposed to predict the student's performance in Intelligent Tutoring System (ITS) taking into consideration the difficulty level of assessments based on Item Response Theory concept. The difficulty level of tasks can infer measurement of the correlation between the student’s performances and assessment questions. To compute the probability of a student solving a task correctly, a set of predictor variables are defined in the FAM including the number of opportunities presented to the student at each task, the duration spent on each step and the difficulty level of each question or latent variable. The results reveal that incorporating the latent variables into the estimates of student performance can significantly enhance the model.

To measure how the activities of learners could impact their learning achievement in MOOCs, the researchers found that Learning Analytics (LAs) in conjunction with machine learning, are effective tools that offer the potential to trace student knowledge. The researchers demonstrated that machine learning could help the educator in providing cohort information about the learning process, furnishing researchers with the ability to both visualise and analyse the information obtained from each tier of the learner. Thus, an accurate predictive model can be acquired in such courses.

Students’ marks in the first assessment and quiz scores in conjunction with social factors are used to predict students’ final performance in online course. Two predictive models were introduced. In the first model, logistic regression was used to predict whether students gained a normal or distinction certificate. In the second predictive model, logistic regression was also used to predict if students achieved certification or not. The results indicated that the number of peer assessment is the most effective feature for acquiring a distinction. The average quiz scores were considered the most reliable predictor for earning a certificate. The accuracy of distinction and normal models were reported with the percentage of 92.6% for the first model and 79.6 % for the second model, respectively.

The association between the Virtual Learning Environment (VLE) data and student performance has been investigated at the University of Maryland, Baltimore County (UMBC) LA used through the implementation of the Check My Activity (CMA) tool. CMA can be defined as an LA tool, which compares students VLE activities with other activities and provides lecturers frequent feedback of students’ emotional states. The results showed the students who engage with the course frequently are more likely to earn mark C or higher than those who did not regularly engage.

CHAPTER 3

SYSTEM ANALYSIS

3.1 EXIXTING SYSTEM

The Factor Analysis Model (FAM) was proposed to predict the student's performance in Intelligent Tutoring System (ITS) taking into consideration the difficulty level of assessments based on Item Response Theory concept [9] [10]. The difficulty level of tasks can infer measurement of the correlation between the student's performances and assessment questions. To compute the probability of a student solving a task correctly, a set of predictor variables are defined in the FAM including the number of opportunities presented to the student at each task, the duration spent on each step and the difficulty level of each question or latent variable. The results reveal that incorporating the latent variables into the estimates of student performance can significantly enhance the model.

To measure how the activities of learners could impact their learning achievement in MOOCs, the researchers found that Learning Analytics (LAs) in conjunction with machine learning, are effective tools that offer the potential to trace student knowledge. The researchers demonstrated that machine learning could help the educator in providing cohort information about the learning process, furnishing researchers with the ability to both visualise and analyse the information obtained from each tier of the learner. Thus, an accurate predictive model can be acquired in such courses. Students' marks in the first assessment and quiz scores in conjunction with social factors are used to predict students' final performance in online course.

Two predictive models were introduced. In the first model, logistic regression was used to predict whether students gained a normal or distinction certificate. In the second predictive model, logistic regression was also used to predict if students achieved certification or not. The results indicated that the number of peer assessment is the most effective feature for acquiring a distinction. The average quiz scores were considered the most reliable predictor for earning a certificate. The accuracy of distinction and normal models were reported with the percentage of 92.6% for the first model and 79.6 % for the second model, respectively.

The association between the Virtual Learning Environment (VLE) data and student performance has been investigated at the University of Maryland, Baltimore County (UMBC) [12]. LA used through the implementation of the Check My Activity (CMA) tool. CMA can be defined as an LA tool, which compares students VLE activities with other activities and provides lecturers frequent feedback of students' emotional states. The results showed the students who engage with the course frequently are more likely to earn mark C or higher than those who did not regularly engage .

3.2DRAWBACKS OF EXISTING SYSTEM

In the existing work, the system is poor performance in which the assessments are not Tutor Marked Assessment (TMA), Computer Marked Assessment(CMA).This system is less performance due to Lack of Massive open online courses (MOOCs).

3.3 PROPOSED SYSTEM

Data Description

The OULAD dataset was captured from the Open University Learning Analytics Dataset (OULAD)repository.The open university in the UK delivers the online course in various topic for undergraduate and postgraduate students in the period between 2013-2014. The main composite table called “studentInfo” is linked to all tables.The"studentInfo "table includes information relevant to students’ demographi characteristics. The information related to students performance are collected In “Assessments” and Student Assessment tables. The table “Assessments” contains information about the number, weight and the type of assessments required for each module. In general, each module involves a set of assessments, followed by the final exam. The assessments are Tutor Marked Assessment (TMA), Computer Marked Assessment (CMA). The final average grade is computed with the sum of all assessments (50%) and final exams (50%). The “Student Assessment” table involves information relating to student and the assessment mark. The “Student Registration” table contains information about the date the students registered and unregistered in a particular module. The overall date is measured by counting numbers of unique days that students interact with courses until the course ends. In Open University online courses, students are able to access a module even before being a student of the course; however, it is not possible to access the course post-course closure date. The students' information related to their interaction with digital is store in learners Virtual Learning Environment dataset

Students’ performance Model

Two sets of experiments are conducted in this case study. In the first experiment, the dynamic behavioral features are considered to predict student performance, while the static behavioral attributes are employed in the second experiment. The problems are formulated as classification and regression. The regression setting is considered when we aim to predict students’ assessments grades, whereas classification setting is utilised when we seek to predict final student performance in the entire course. It is considered a multi-class problem where the target class is whether students pass, fail or withdraw from courses. Early grade prediction could help educators deliver timely intervention support and additional learning materials to help students who have low scores. As discussed previously, the student should participate in five CMA assessments and six TMA assessments, in addition to the final exam. The assessments should be handed in within a specific period. Due to the TMA assessment weighing 45% of the final result, while the CMA assessment weighs only 5%, our temporal analysis is based on the submission date of the TMA. In first set experiments, student performance is predicated in a timely manner, as can be seen, in Figure 1 the course is subsequently the into six-time intervals, corresponding with assessment submission dates. The student behavioral records are distributed according to the assessment date. The student performance in prior assessments with their interaction behavior is considered in this analysis.

In the second set of experiments, we evaluated the trajectories student performance by aggregate the student's behavioral activities across the six-time slices into a single time slice. The behavioral features, demographic features and temporal features are used as input variables. We did not account for past assessments grade, and final exam mark as target class is computed based on these features. The dataset contains 4004 records where the proportion of "fail", "withdrawn" and "pass" classes are 28%t,40% and 32% respectively.

Features selection

As our aim is to investigate the impact of student performance in the previous assessment into the following assessment, features selection is only consider for the first set of experiment. Recursive Feature Elimination (RFE) is utilized in this case study. Rrecursive feature elimination is one of the most popular wrapper feature selection approaches. RFE can be defined as an optimisation algorithm based on backwards selection and resampling techniques. It keeps recursively creating the model until it gets a small number of features. The data set is partitioned into train and bootstrap samples with different elements. At each iteration, the algorithms are chosen as the most important features. To assess the probability of ranking features, the new model that includes the most essential predictors is retained until all are exhausted.

3.4 PROPOSED SYSTEM ADVANTAGES

- Students' marks in the first assessment and quiz scores in conjunction with social factors are used to predict students' final performance in online course.

- Learning Analytics (LAs) in conjunction with machine learning, are effective tools that offer the potential to trace student knowledge.

3.5 SYSTEM REQUIREMENTS

3.5.1 HARDWARE SYSTEM CONFIGURATION:

- **Processor** - Pentium –IV
- **RAM** - 4 GB (min)
- **Hard Disk** - 20 GB
- **Key Board** - Standard Windows Keyboard
- **Mouse** - Two or Three Button Mouse
- **Monitor** - SVGA

3.5.2 SOFTWARE REQUIREMENTS:

- **Operating system** : Windows 7 Ultimate.
- **Coding Language** : Python.
- **Front-End** : Python.
- **Back-End** : Django-ORM
- **Designing** : Html, css, javascript.
- **Data Base** : MySQL (WAMP Server).

3.6 FEASIBILITY STUDY

An important outcome of preliminary investigation is the determination that the system request is feasible. This is possible only if it is feasible within limited resource and time. The different feasibilities that have to be analyzed are

. Operational Feasibility

. Economic Feasibility

. Technical Feasibility

Operational Feasibility

Operational Feasibility deals with the study of prospects of the system to be developed. This system operationally eliminates all the tensions of the admin and helps him in effectively tracking the project progress. This kind of automation will surely reduce the time and energy, which previously consumed in manual work. Based on the study, the system is proved to be operationally feasible.

Economic Feasibility

Economic Feasibility or Cost-benefit is an assessment of the economic justification for a computer-based project. As hardware was installed from the beginning & for lots of purposes thus the cost on project of hardware is low. Since the system is a network based, any number of employees connected to the LAN within that organization can use this tool from at any time. The Virtual Private Network is to be developed using the existing resources of the organization. Hence the project is economically feasible.

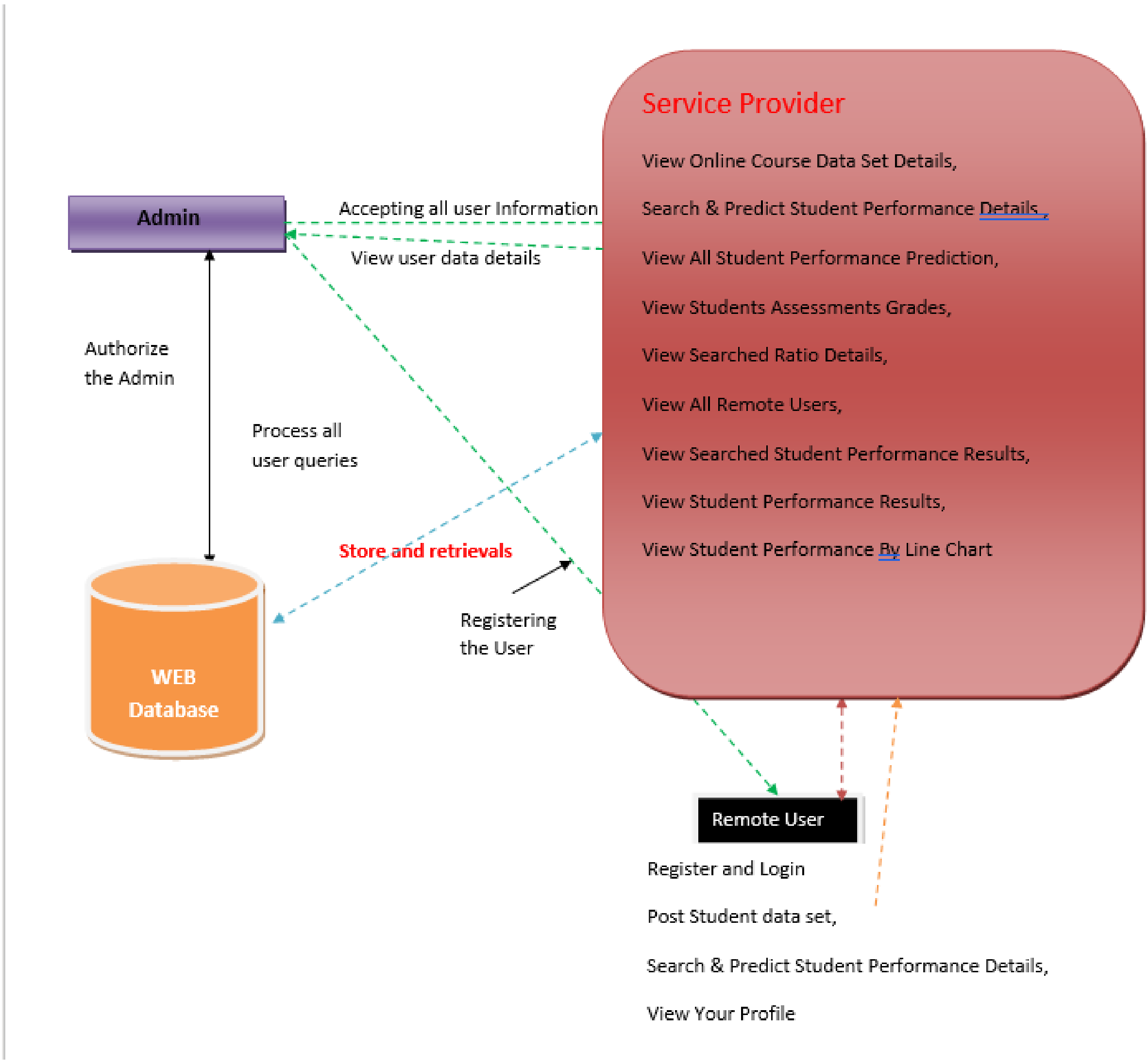
Technical Feasibility

According to Roger S. Pressman, Technical Feasibility is the assessment of the technical resources of the organization. The organization needs IBM compatible machines with a graphical web browser connected to the Internet and Intranet. The system is developed for platform independent environment. Java Server Pages, JavaScript, HTML, SQL server and WebLogic Server are used to develop the system. The technical feasibility has been carried out. The system is technically feasible for development and can be developed with the existing facility.

CHAPTER 4

SYSTEM DESIGN

4.1 SYSTEM ARCHITECTURE



4.2 MODULES

4.2.1 Service Provider

In this module, the Service Provider has to login by using valid user name and password. After login successful he can do some operations such as
View Online Course Data Set Details,Search & Predict Student Performance Details ,View All Student Performance Prediction, View Students Assessments Grades,View Searched Ratio Details,View All Remote Users,View Searched Student Performance Results,View Student Performance Results,View Student Performance By Line Chart.

4.2.2 View and Authorize Users

In this module, the admin can view the list of users who all registered. In this, the admin can view the user's details such as, user name, email, address and admin authorizes the users.

4.2.3 Remote User

In this module, there are n numbers of users are present. User should register before doing any operations. Once user registers, their details will be stored to the database. After registration successful, he has to login by using authorized user name and password. Once Login is successful user will do some operations like Post Student data set, Search & Predict Student Performance Details, View Your Profile.

4.3 UML DIAGRAMS

UML stands for Unified Modelling Language. UML is a standardized general-purpose modelling 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 objectoriented computer software. In its current form UML is comprised of two major components and a notation. In the future, some form of method or process may also be added to; or associated with, UML.

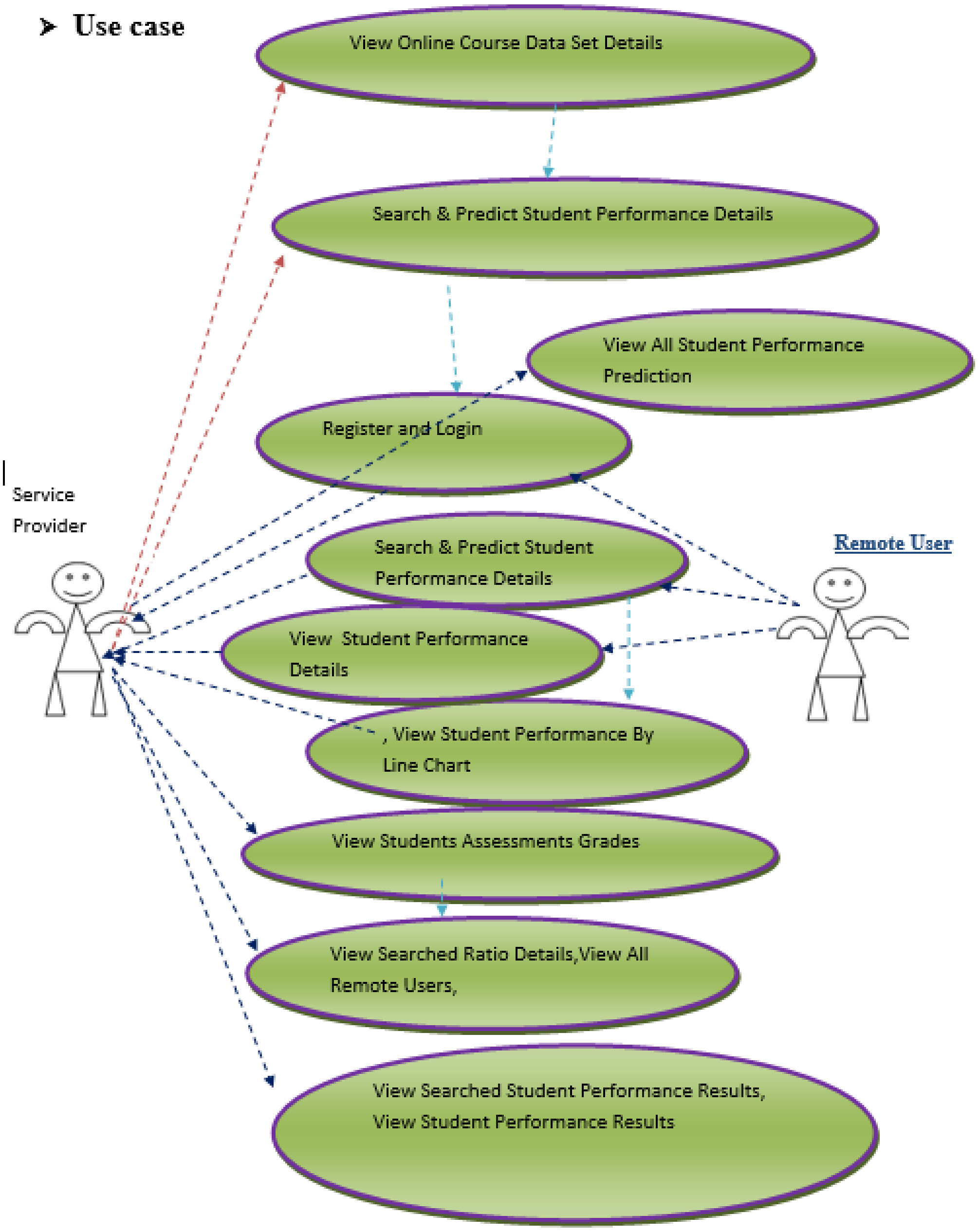
The Unified Modelling Language is a standard language for specifying, Visualization, Constructing and documenting the artifacts of software system, as well as for business modelling and other non-software systems.

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

1.Use Case Diagram

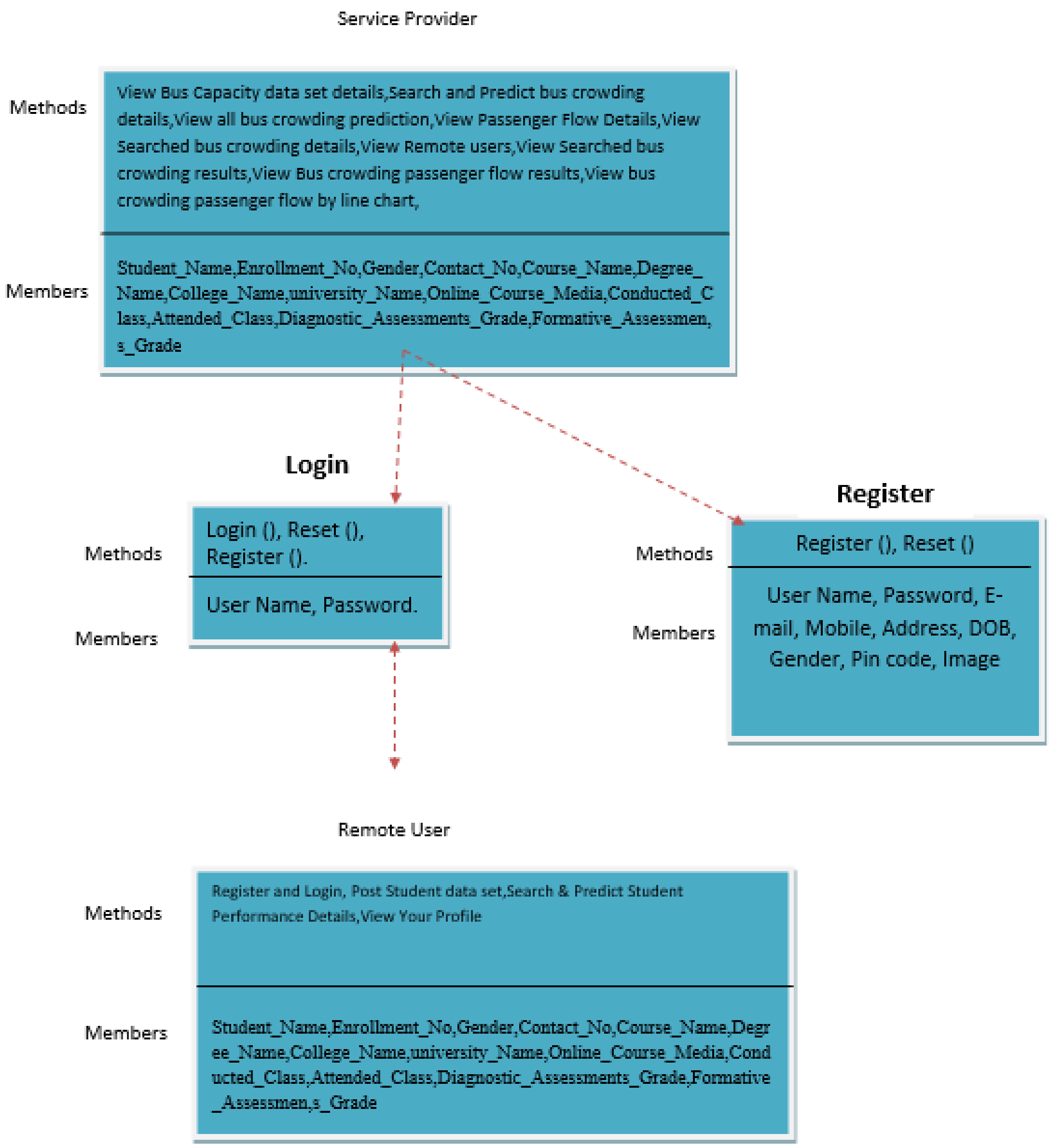
A use case diagram in the Unified Modelling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies 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.

- **Use case**


View Online Course Data Set Details
Search & Predict Student Performance Details
View All Student Performance Prediction
Register and Login
Service Provider
Search & Predict Student Performance Details
Remote User
View Student Performance Details
, View Student Performance By Line Chart
View Students Assessments Grades
View Searched Ratio Details,View All Remote Users,
View Searched Student Performance Results, View Student Performance Results

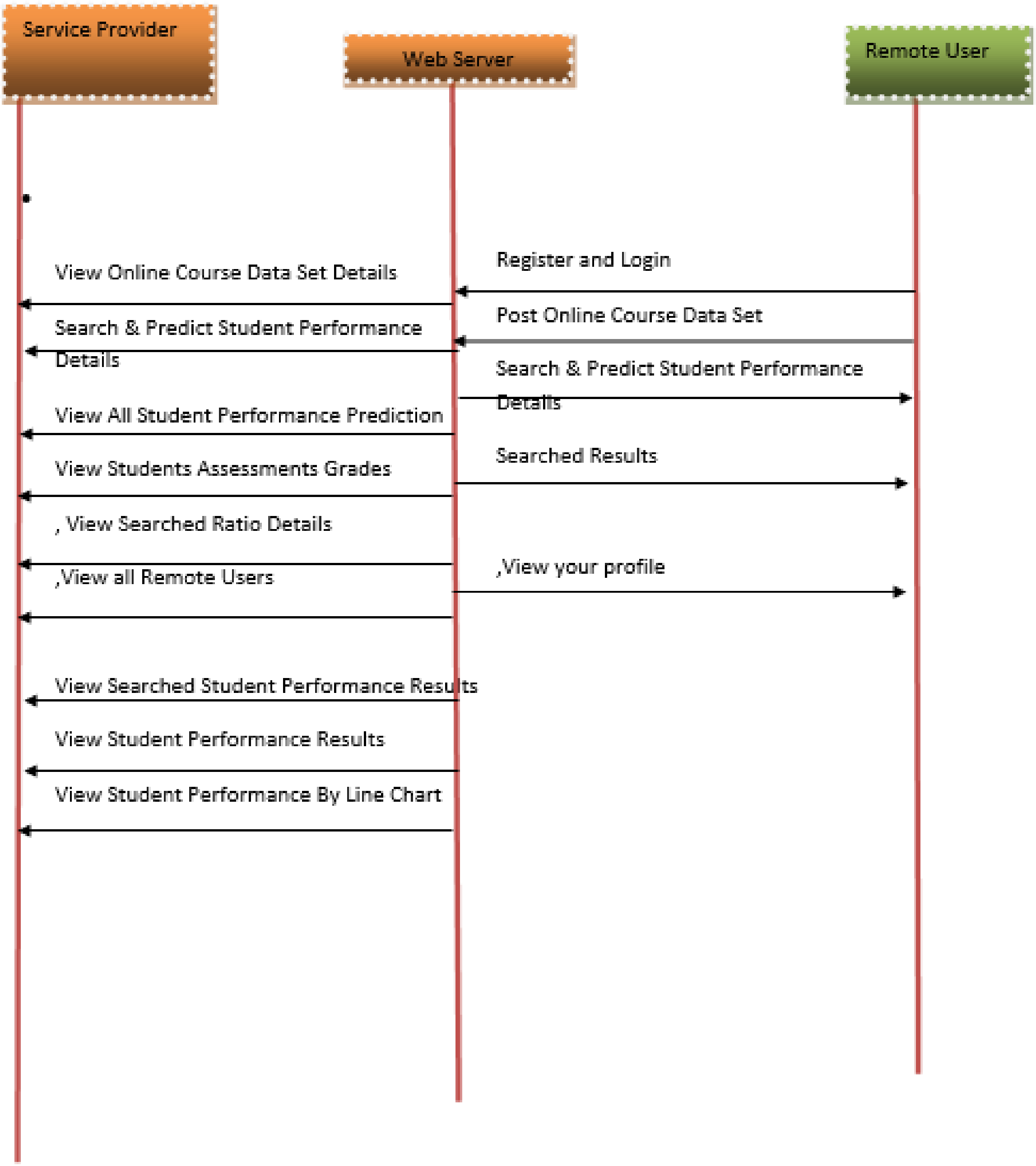
2. Class Diagram

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 upload histopathological image dataset preprocess dataset train AI with ADAM train AI with SCP train AI with min batch User comparison table classes, their attributes, operations (or methods), and the relationships among the classes. It explains which class contains information.



3. Sequence Diagram

A sequence diagram in Unified Modelling 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.Activity Diagram

An activity diagram is a type of UML (Unified Modelling Language) diagram that visually represents the workflow or flow of activities within a system, process, or business. It is particularly useful for modelling business processes and the sequence of activities that occur in a system. Activity diagrams are part of the UML diagram set and are often employed during the analysis and design phases of software development.

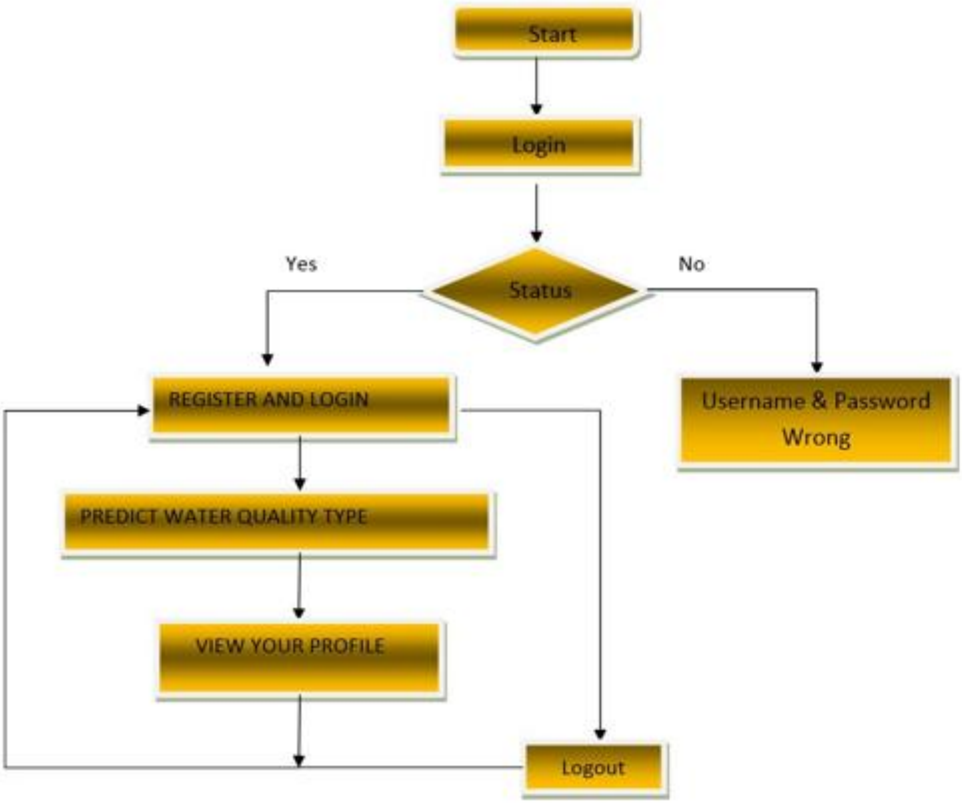


Fig 4.3.4 Activity Diagram

5.Data Flow Diagram

A Data Flow Diagram (DFD) is a visual representation that depicts the flow of data within a system and how it is processed. DFDs are a part of structured analysis and design methods and are commonly used in system analysis to model the data aspects of a system or process. The diagram provides a clear illustration of inputs, processes, data storage, and outputs in a system.

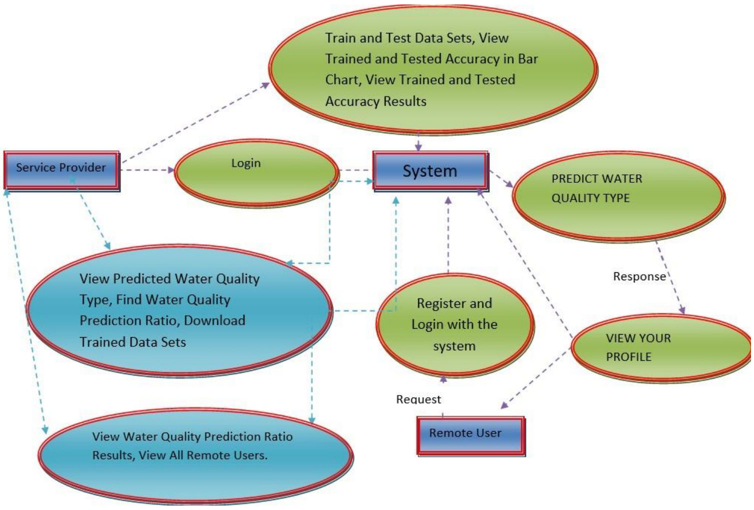


Fig 4.3.5 Data Flow Diagram

6.E-R Diagram

A Data Flow Diagram (DFD) is a visual representation that depicts the flow of data within a system and how it is processed. DFDs are a part of structured analysis and design methods and are commonly used in system analysis to model the data aspects of a system or process. The diagram provides a clear illustration of inputs, processes, data storage, and outputs in a system.

7.Component Diagram

A Component Diagram is a type of UML (Unified Modelling Language) diagram that illustrates the high-level structure of a system, showcasing the components that make up the system and their relationships. Components in a UML Component Diagram can represent physical entities like executable files, libraries, or database tables, as well as logical components such as classes, interfaces, or packages.

8. Collaboration Diagram

The collaboration diagram is used to show the relationship between the objects in a system. Both the sequence and the collaboration diagrams represent the same information but differently. Instead of showing the flow of messages, it depicts the architecture of the object residing in the system as it is based on object-oriented programming. An object consists of several features. Multiple objects present in the system are connected to each other. The collaboration diagram, which is also known as a communication diagram, is used to portray the object's architecture in the system.

CHAPTER5

SYSTEM IMPLEMENTATION

5.1 MACHINE LEARNING

Machine learning is a subfield of artificial intelligence (AI) that focuses on the development of algorithms and statistical models that enable computers to perform tasks without explicit programming. The primary goal of machine learning is to enable computers to learn and improve from experience, allowing them to make predictions or decisions based on data. In traditional programming, developers write explicit code to instruct a computer on how to perform a specific task. In contrast, machine learning involves training a model on a dataset to learn patterns, relationships, and rules from the data. The trained model can then make predictions or decisions when exposed to new, unseen data. There are various types of machine learning, including:

Supervised Learning

The model is trained on a labeled dataset, where the input data is paired with corresponding output labels. The goal is to learn a mapping from inputs to outputs.

Unsupervised Learning

The model is given unlabeled data and must find patterns and structures in the data without explicit guidance. Clustering and dimensionality reduction are common tasks in unsupervised learning.

Reinforcement Learning

The model learns through interaction with an environment. It receives feedback in the form of rewards or penalties based on its actions, allowing it to learn to make decisions to maximize cumulative rewards. Machine learning has applications in various domains, such as image and speech recognition, natural language processing, recommendation systems, and autonomous vehicles, among others.

5.1.1 Need For Machine Learning

Machine learning has become an indispensable tool in the realm of data analysis, driven by the escalating complexity and sheer volume of datasets. Traditional methods of data processing have proven inadequate in the face of big data, prompting the adoption of machine learning to efficiently handle large datasets, identify intricate patterns, and make predictions or decisions based on the extracted insights. The ability of machine learning algorithms to navigate and distill valuable information from vast datasets has positioned them as a cornerstone in modern data-driven decision-making processes.

Automation and efficiency are paramount factors propelling the adoption of machine learning. By automating tasks that are challenging or impractical to program using conventional methods, machine learning enhances efficiency and productivity. Moreover, its capacity to adapt and improve performance over time, learning from new data, further underscores its utility in automating processes and tasks that would otherwise require constant manual intervention.

Machine learning's adaptability to changing environments is a notable advantage. Traditional rule-based systems often struggle to cope with dynamic conditions, making machine learning models particularly suited to scenarios where adaptability and continual learning are essential. Industries such as finance, logistics, and healthcare benefit from machine learning's capability to evolve alongside changing circumstances, ensuring robust decision-making even in volatile environments.

5.1.2 Challenges In Machine Learning

Machine learning encounters a host of challenges that influence its efficacy and applicability across diverse domains. One of the primary hurdles is the quality and quantity of data. Successful machine learning models often rely on extensive and diverse datasets for training. However, obtaining a sufficient amount of relevant data, especially in niche or emerging fields, can be a formidable task. Additionally, the presence of biases in training data poses a significant challenge, as it can lead to biased models that perpetuate and amplify existing prejudices. Addressing data quality and mitigating biases are essential steps towards achieving ethical and unbiased machine learning outcomes.

Another critical challenge in machine learning is the delicate balance between overfitting and underfitting. Overfitting occurs when a model learns the training data too well, including noise and outliers, making it less effective on new, unseen data. On the contrary, underfitting arises when a model is too simplistic and fails to capture the underlying patterns in the data. Striking the right balance to avoid both overfitting and underfitting is an ongoing challenge that demands careful model tuning and selection.

5.1.3 Applications For Machine Learning

Machine learning finds applications across a wide range of industries and domains, demonstrating its versatility and transformative potential. Here are some notable applications of machine learning:

Healthcare

Machine learning is used for image analysis in medical diagnostics, including the interpretation of X-rays, MRIs, and CT scans.

Predictive models analyze patient data to identify individuals at risk of specific diseases, aiding in early diagnosis and intervention.

Finance

Machine learning algorithms analyze transaction patterns to detect unusual behavior and identify potential fraudulent activities.

Predictive models assess creditworthiness by analyzing historical financial data and other relevant variables.

Retail and E-Commerce

Machine learning powers personalized product recommendations based on user preferences and behavior.

Predictive modeling helps optimize inventory management by forecasting product demand.

Natural Language Processing

NLP algorithms enable the development of conversational agents for customer support and assistance.

Machine learning facilitates automatic language translation, breaking down language barriers.

Marketing and Advertising

Machine learning algorithms segment customers based on behavior, allowing for targeted marketing campaigns.

Predictive analytics is used to target advertisements to users who are more likely to engage or convert.

Autonomous Vehicles

Machine learning is used in computer vision systems to identify and classify objects on the road.

Manufacturing and Industry

Machine learning predicts equipment failures by analyzing sensor data, allowing for proactive maintenance. Computer vision systems inspect and identify defects in manufacturing processes.

Energy Management

Machine learning tailors educational content to individual student needs, adapting to different learning styles.

Predictive models identify students at risk of academic challenges, enabling timely intervention.

5.1.4 Working Of Machine Learning

Machine learning is the brain where all the learning takes place. The way the machine learns is similar to the human being. Humans learn from experience. The more we know, the more easily we can predict. By analogy, when we face an unknown situation, the likelihood of success is lower than the known situation. Machines are trained the same. To make an accurate prediction, the machine sees an example. When we give the machine a similar example, it can figure out the outcome. However, like a human, if its feeds a previously unseen example, the machine has difficulties predicting. The core objective of machine learning is the learning and inference. First of all, the machine learns through the discovery of patterns. This discovery is made thanks to the data. One crucial part of the data scientist is to choose carefully which data to provide to the machine.

The list of attributes used to solve a problem is called a feature vector. You can think of a feature vector as a subset of data that is used to tackle a problem. The machine uses some fancy algorithms to simplify the reality and transform this discovery into a model. Therefore, the learning stage is used to describe the data and summarize it into a model. For instance, the machine is trying to understand the relationship between the wage of an individual and the likelihood to go to a fancy restaurant. It turns out the machine finds a positive relationship between wage and going to a high- end restaurant. This is the model inferring. Fig 5.1.2: Machine Learning Inference from Model. When the model is built, it is possible to test how powerful it is on never-seen-before data. The new data are transformed into a features vector, go through the model and give a prediction. This is all the beautiful part of machine learning. There is no need to update the rules or train again the model. You can use the model previously trained to make inference on new data.

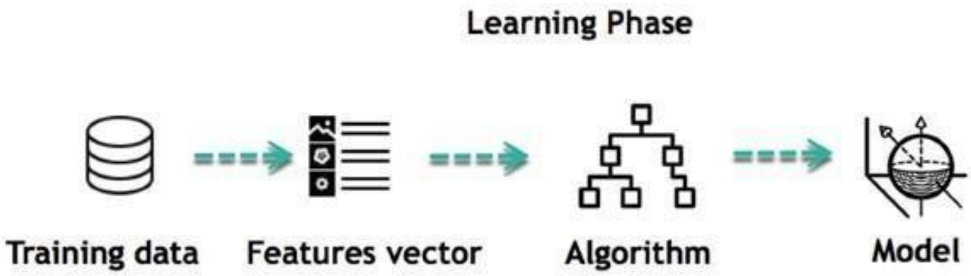


Fig 5.1.1 Learning phase of machine learning

The life of Machine Learning programs is straightforward and can be summarized in the following points

- Define a question
- Collect data
- Visualize data
- Train algorithm
- Test the algorithm
- Collect feedback
- Refine the algorithm
- Loop 4-7 until the results are satisfying
- Use the model to make a prediction

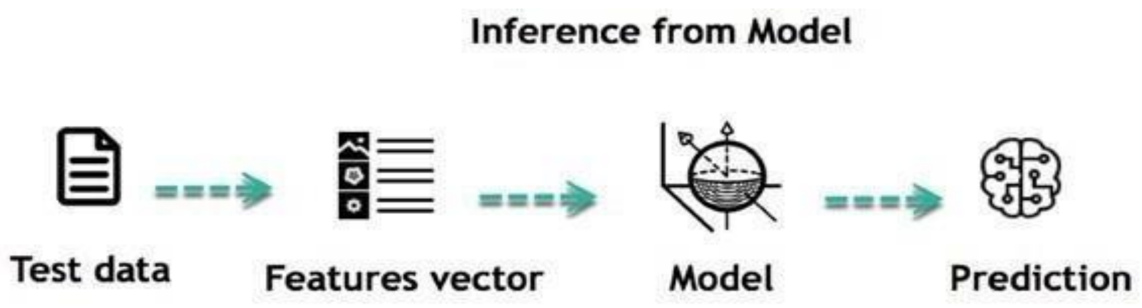


Fig 5.1.2 Machine Learning Inference from Model

5.2 PYTHON

Python programming language is used for building the machine learning model.

5.2.1 Introduction

Python is a high-level, general-purpose programming language renowned for its simplicity, readability, and versatility. Guido van Rossum created Python in the late 1980s, and it has since evolved into one of the most popular and widely used programming languages globally.

One of Python's defining features is its readability, emphasized by the use of indentation rather than braces for code blocks. This design choice makes Python code clean, concise, and easy to understand, fostering a collaborative and accessible programming environment. Python's syntax is clear and expressive, allowing developers to focus on problem-solving rather than deciphering complex code structures.

Python's versatility is evident in its applicability to various domains. From web development and data science to artificial intelligence and automation, Python is a go-to language for diverse applications. It supports multiple programming paradigms, including procedural, object-oriented, and functional programming, providing developers with flexibility in crafting solutions.

5.2.2 Python Features

1. Free and Open Source

Python language is freely available at the official website and you can download it from the given download link below click on the Download Python keyword. Download Python Since it is open-source, this means that source code is also available to the public. So you can download it, use it as well as share it.

2. Easy to Code

Python is a high-level programming language. Python is very easy to learn the language as compared to other languages like C, C#, Javascript, Java, etc. It is very easy to code in the Python language and anybody can learn Python basics in a few hours or days. It is also a developer-friendly language.

3. Interpreted Language

Python is an interpreted language; it means the Python program is executed one line at a time. The advantage of being interpreted language, it makes debugging easy and portable.

4. GUI Programming Support

Graphical User Interface is used for the developing Desktop application. PyQT5, Tkinter, Kivy are the libraries which are used for developing the web application

5.Object-Oriented Language

Python supports object-oriented language and concepts of classes and objects come into existence. It supports inheritance, polymorphism, and encapsulation, etc. The objectoriented procedure helps to programmer to write reusable code and develop applications in less code.

6.Extensible

It implies that other languages such as C/C++ can be used to compile the code and thus it can be used further in our Python code. It converts the program into byte code, and any platform can use that byte code.

5.2.3 Python History

Python laid its foundation in the late 1980s.The implementation of Python was started in the December 1989 by Guido Van Rossum at CWI in Netherland. ABC programming language is said to be the predecessor of Python language which was capable of Exception Handling and interfacing with Amoeba Operating System.

5.2.4 Python Version

Python programming language is being updated regularly with new features and support. There are a lot of updates in python versions, started from 1994 to current date. A list of python versions with its released date is given below

Python Version	Released Date
Python 1.0	January 20, 1994
Python 1.5	December 31, 1997
Python 1.6	September 5, 2000
Python 2.0	October 16, 2000
Python 2.1	April 17, 2001
Python 2.2	December 21, 2001

Python 2.3	July 29, 2003
Python 2.4	November 30, 2004
Python 2.5	September 19, 2006
Python 2.6	October 1, 2008
Python 2.7	July 3, 2010
Python 3.0	December 3, 2008
Python 3.1	June 27, 2009
Python 3.2	February 20, 2011
Python 3.3	September 29, 2012

Table 5.2.1 Python Version Table.

5.2.5 Python Applications

Python is a versatile programming language that finds applications across various domains. Here are some notable areas where Python is commonly used

1. Web Development

A high-level web framework that encourages rapid development and clean, pragmatic design. A lightweight web framework that is easy to use and extensible, making it suitable for small to medium-sized web applications.

2. Data Science and Machine Learning

Libraries for numerical and data analysis, respectively.

A machine learning library that provides simple and efficient tools for data mining and data analysis.

Frameworks for building and training machine learning models, particularly neural networks.

3. Artificial Intelligence
Python is widely used in AI development due to its simplicity and the availability of powerful libraries and frameworks.

4. Scientific Computing
Python is used in scientific computing for tasks such as simulations, data analysis, and visualization.

5.Automation and Scripting

Python's simplicity and readability make it an excellent choice for automating repetitive tasks and scripting.

6.Game Development

A set of Python modules designed for writing video games.

7.Desktop GUI Applications

The standard GUI (Graphical User Interface) package for Python. PyQt and wxPython Other popular options for creating desktop applications with graphical interfaces.

8.Network Programming

Python is used for writing network-related scripts and applications, leveraging libraries like socket and frameworks like Twisted.

9.Cybersecurity

Python is employed in various cybersecurity tasks, including penetration testing, network scanning, and developing security tools.

10.DevOps and System Administration

Python is widely used for automation in system administration and DevOps tasks. Tools like Ansible are written in Python.

11.Education

Python is a popular choice for teaching programming due to its readability and simplicity, making it accessible for beginners.

12. Financial and Trading Applications
Python is used in the financial industry for tasks like algorithmic trading, quantitative analysis, and financial modeling.

13. Embedded Systems
Python is sometimes used in embedded systems programming due to its simplicity and ease of integration.

5.2.6 Python Execution

1. Interactive Mode:
You can enter “python” in the command prompt and start working with Python by executing Python commands.

2. Script Mode:
Using Script Mode, Python code is written in a separate file using any editor of the Operating System. It is then saved using .py extension. In order to open use the command “python file_name.py” after setting the path of the file in command prompt.

NOTE: Path in the command prompt should be where you have saved your file. In the above case file should be saved at desktop.

3. Using IDE: (Integrated Development Environment)
Python code can be executed using a Graphical User Interface (GUI).It is done by following the below steps:

Click on Start button -> All Programs -> Python -> IDLE(Python GUI) In IDE both interactive and script mode can be used.

- Using Interactive mode:

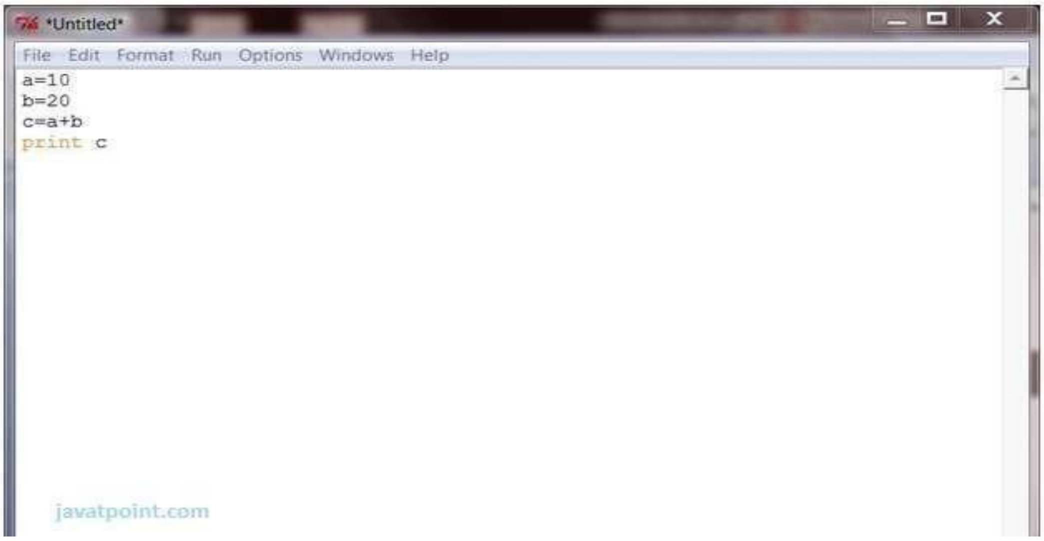
Execute your Python code on the Python prompt and it will display result simultaneously.

- Using Script Mode:

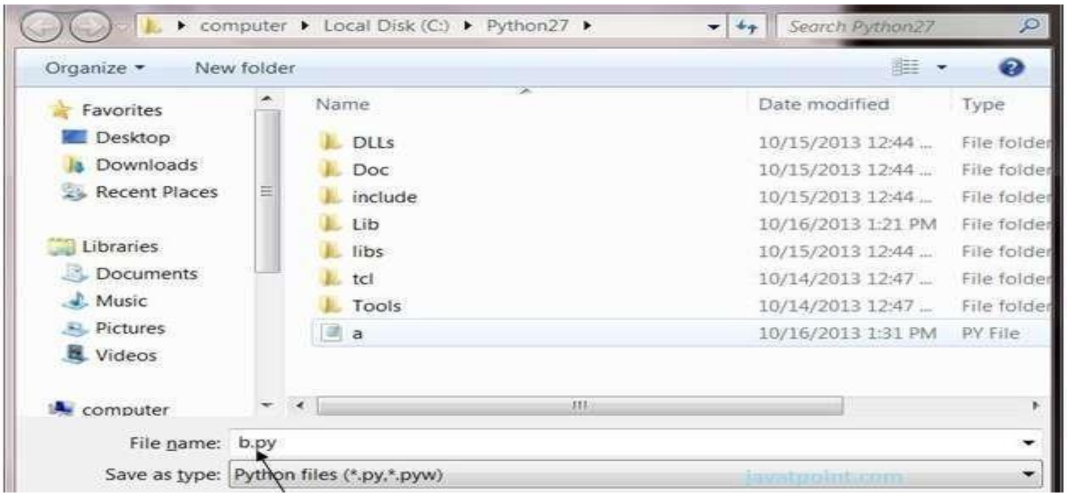
Click on Start button -> All Programs -> Python -> IDLE(Python GUI)

Python Shell will be opened. Now click on File -> New Window. A new Editor will be opened . Write your Python code here. Click on file -> save as Run then code by clicking on Run in the Menu bar. Run -> Run Module

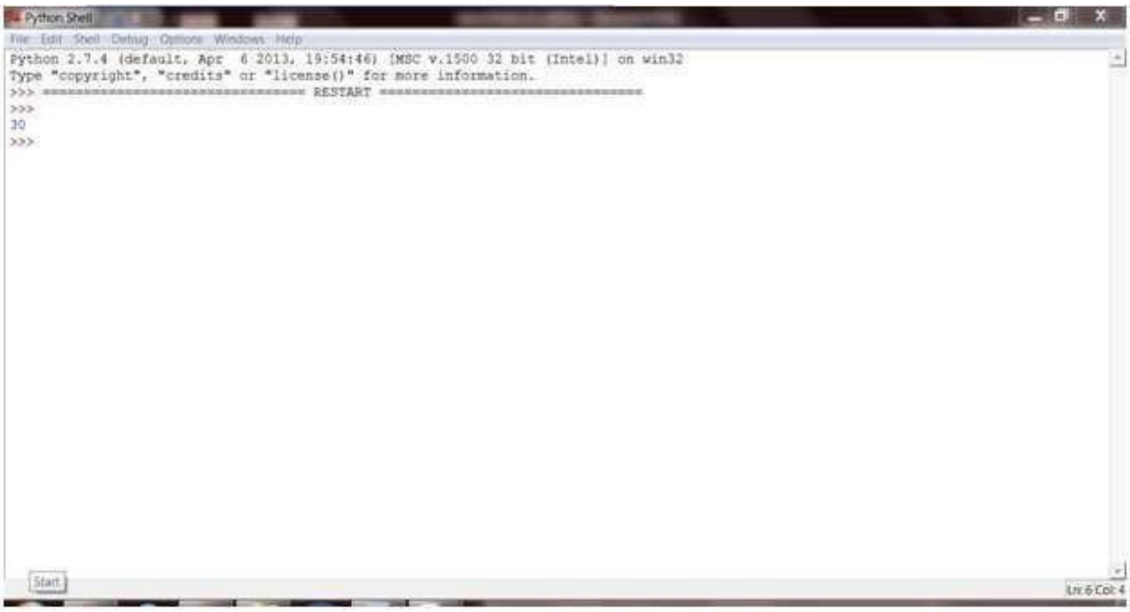
Result will be displayed on a new Python shell.



(a)



(b)



(c)

Fig 5.2.1 (a),(b),(c) Python program execution in IDE script mode

5.2.7 Python Variables

Variable is a name of the memory location where data is stored. Once a variable is stored that means a space is allocated in memory. For assigning values to variable we need not to declare explicitly variable in Python. When we assign any value to the variable that variable is declared automatically. The assignment is done using the equal (=) operator.

5.3 FUNDAMENTALS OF PYTHON

This section contains the basic fundamentals of Python.

5.3.1 Tokens

Tokens can be defined as a punctuator mark, reserved words and each individual word in a statement. Token is the smallest unit inside the given program. Tokens include Keywords, Identifiers, Literals, Operators.

5.3.2 Tuples

Tuple is another form of collection where different type of data can be stored. It is similar to list where data is separated by commas. Only the difference is that list uses square bracket and tuple uses parenthesis.

5.3.3 Dictionary

Dictionary is a collection which works on a key-value pair. It works like an associated array where no two keys can be same. Dictionaries are enclosed by curly braces ({}) and values can be retrieved by square bracket([])

Eg: dictionary={'name':'charlie','id':100,'dept':'it'}

5.4 PACKAGES

5.4.1 Django

Django is a Python framework that makes it easier to create web sites using Python.

Django takes care of the difficult stuff so that you can concentrate on building your web applications.

Django emphasizes reusability of components, also referred to as DRY (Don't Repeat Yourself), and comes with ready-to-use features like login system, database connection and CRUD operations (Create Read Update Delete).

Django Working

Django follows the MVT design pattern (Model View Template).

Model - The data you want to present, usually data from a database.

View - A request handler that returns the relevant template and content - based on the request from the user.

Template - A text file (like an HTML file) containing the layout of the web page, with logic on how to display the data.

Model

The model provides data from the database.

In Django, the data is delivered as an Object Relational Mapping (ORM), which is a technique designed to make it easier to work with databases.

The most common way to extract data from a database is SQL. One problem with SQL is that you have to have a pretty good understanding of the database structure to be able to work with it.

Django, with ORM, makes it easier to communicate with the database, without having to write complex SQL statements.

The models are usually located in a file called models.py.

View

A view is a function or method that takes http requests as arguments, imports the relevant model(s), and finds out what data to send to the template, and returns the final result.

The views are usually located in a file called views.py.

Template

A template is a file where you describe how the result should be represented.

Templates are often .html files, with HTML code describing the layout of a web page, but it can also be in other file formats to present other results, but we will concentrate on .html files.

5.5 MYSQL

MySQL is the world's most popular open source database. According to DB-Engines, MySQL ranks as the second-most-popular database, behind Oracle Database. MySQL powers many of the most accessed applications, including Facebook, Twitter, Netflix, Uber, Airbnb, Shopify, and Booking.com.

Since MySQL is open source, it includes numerous features developed in close cooperation with users over more than 25 years. So it’s very likely that your favourite application or programming language is supported by MySQL Database.

Databases are the essential data repository for all software applications. For example, whenever someone conducts a web search, logs in to an account, or completes a transaction, a database system is storing the information so it can be accessed in the future.

A relational database stores data in separate tables rather than putting all the data in one big storeroom. The database structure is organized into physical files optimized for speed. The logical data model, with objects such as data tables, views, rows, and columns, offers a flexible programming environment. You set up rules governing the relationships between different data fields, such as one to one, one to many, unique, required, or optional, and “pointers” between different tables. The database enforces these rules so that with a well-designed database your application never sees data that’s inconsistent, duplicated, orphaned, out of date, or missing.

The “SQL” part of “MySQL” stands for “Structured Query Language.” SQL is the most common standardized language used to access databases. Depending on your programming environment, you might enter SQL directly (for example, to generate reports), embed SQL statements into code written in another language, or use a languagespecific API that hides the SQL syntax.

Source code

```
# Import necessary libraries

import pandas as pd

from sklearn.model_selection import train_test_split

from sklearn.preprocessing import StandardScaler

from sklearn.ensemble import RandomForestRegressor

from sklearn.metrics import mean_squared_error


# Load the dataset

data = pd.read_csv('student_data.csv')
```

```
# Split data into features and target variable
X = data.drop('performance_grade', axis=1) # Features
y = data['performance_grade'] # Target variable

# Split data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Feature scaling
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)

# Initialize and train the model
model = RandomForestRegressor(n_estimators=100, random_state=42)
model.fit(X_train_scaled, y_train)

# Make predictions
y_pred = model.predict(X_test_scaled)

# Evaluate the model
mse = mean_squared_error(y_test, y_pred)
print('Mean Squared Error:', mse)
# Load the dataset
data = pd.read_csv('student_data.csv')

# Split data into features and target variable
```

```
X = data.drop('performance_grade', axis=1)  # Features

y = data['performance_grade']  # Target variable


# Split data into training and testing sets

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)


# Feature scaling

scaler = StandardScaler()

X_train_scaled = scaler.fit_transform(X_train)

X_test_scaled = scaler.transform(X_test)


# Initialize and train the model

model = RandomForestRegressor(n_estimators=100, random_state=42)

model.fit(X_train_scaled, y_train)


# Make predictions

y_pred = model.predict(X_test_scaled)


# Evaluate the model

mse = mean_squared_error(y_test, y_pred)

print('Mean Squared Error:', mse)

# Import necessary libraries

import pandas as pd

from sklearn.model_selection import train_test_split

from sklearn.preprocessing import StandardScaler

from sklearn.ensemble import RandomForestRegressor
```

```
from sklearn.metrics import mean_squared_error

# Load the dataset
data = pd.read_csv('student_data.csv')

# Split data into features and target variable
X = data.drop('performance_grade', axis=1) # Features
y = data['performance_grade'] # Target variable

# Split data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Feature scaling
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)

# Initialize and train the model
model = RandomForestRegressor(n_estimators=100, random_state=42)
model.fit(X_train_scaled, y_train)

# Make predictions
y_pred = model.predict(X_test_scaled)

# Evaluate the model
mse = mean_squared_error(y_test, y_pred)
```

```
print('Mean Squared Error:', mse)

# Load the dataset
data = pd.read_csv('student_data.csv')

# Split data into features and target variable
X = data.drop('performance_grade', axis=1) # Features
y = data['performance_grade'] # Target variable

# Split data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Feature scaling
scaler = StandardScaler()
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)

# Initialize and train the model
model = RandomForestRegressor(n_estimators=100, random_state=42)
model.fit(X_train_scaled, y_train)

# Make predictions
y_pred = model.predict(X_test_scaled)

# Evaluate the model
```

```
mse = mean_squared_error(y_test, y_pred)

print('Mean Squared Error:', mse)
```

CHAPTER 6
TESTING

6.1 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 test. Each test type addresses a specific testing requirement.

6.2 TYPES OF TESTING

6.2.1 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 a 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.

6.2.2 Integration Testing

Integration tests are designed to test integrated software components to determine if they actually 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.

6.2.3 Functional Testing

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.

Functions: 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.

6.2.4 System Testing

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.

6.25 White Box Testing

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

6.26 Black Box Testing

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box, you cannot “see” into it. The test provides inputs and responds to outputs without considering how the software works.

6.27 Unit Testing

Unit testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases.

6.3 TEST STRATEGY AND APPROACH

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

6.31 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.

6.32 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.

6.33 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.

6.34 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.

CHAPTER 7

RESULTS

7.1 SCREENSHOTS

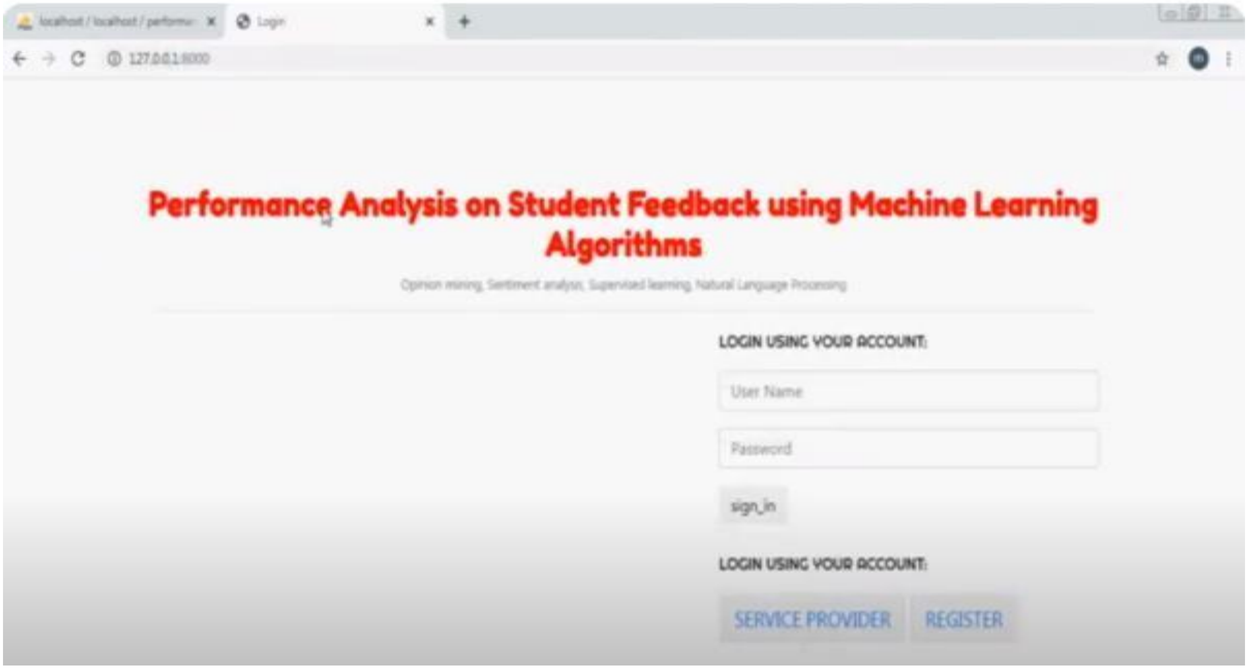


FIG 1:HOME PAGE

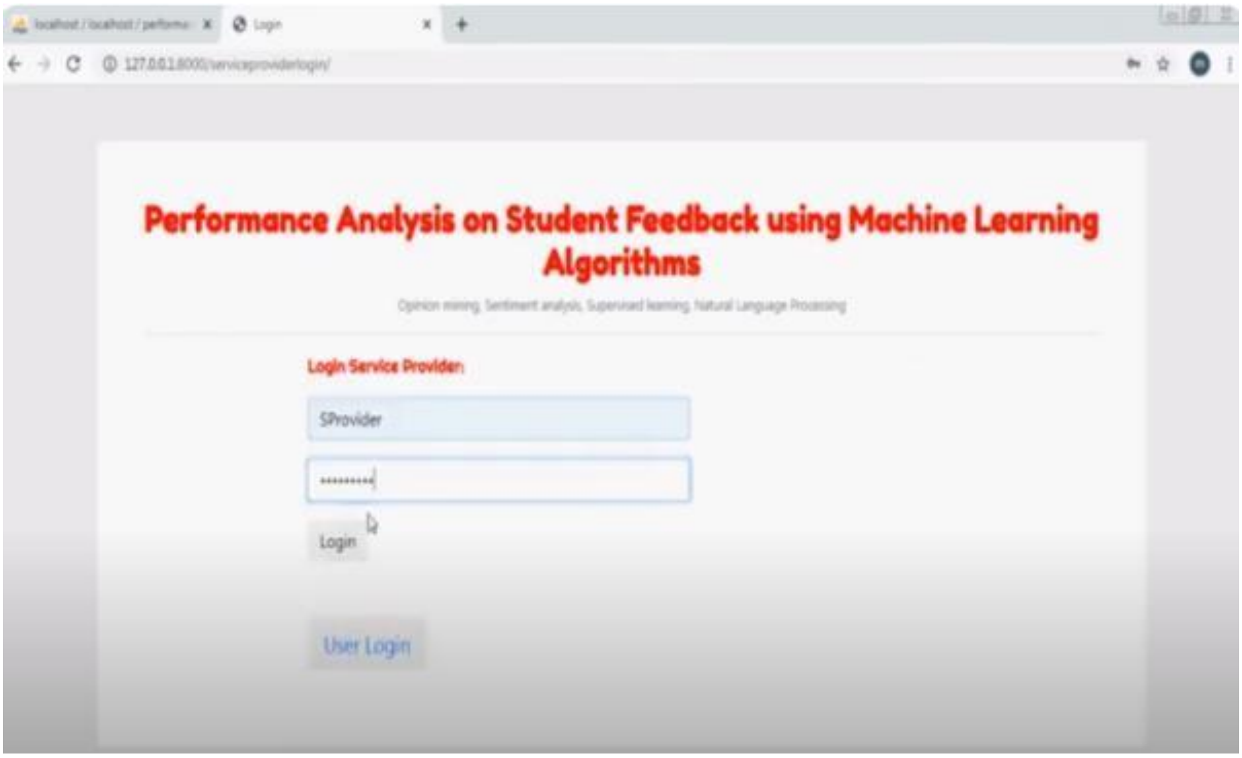


FIG 2: ADMIN LOGIN PAGE

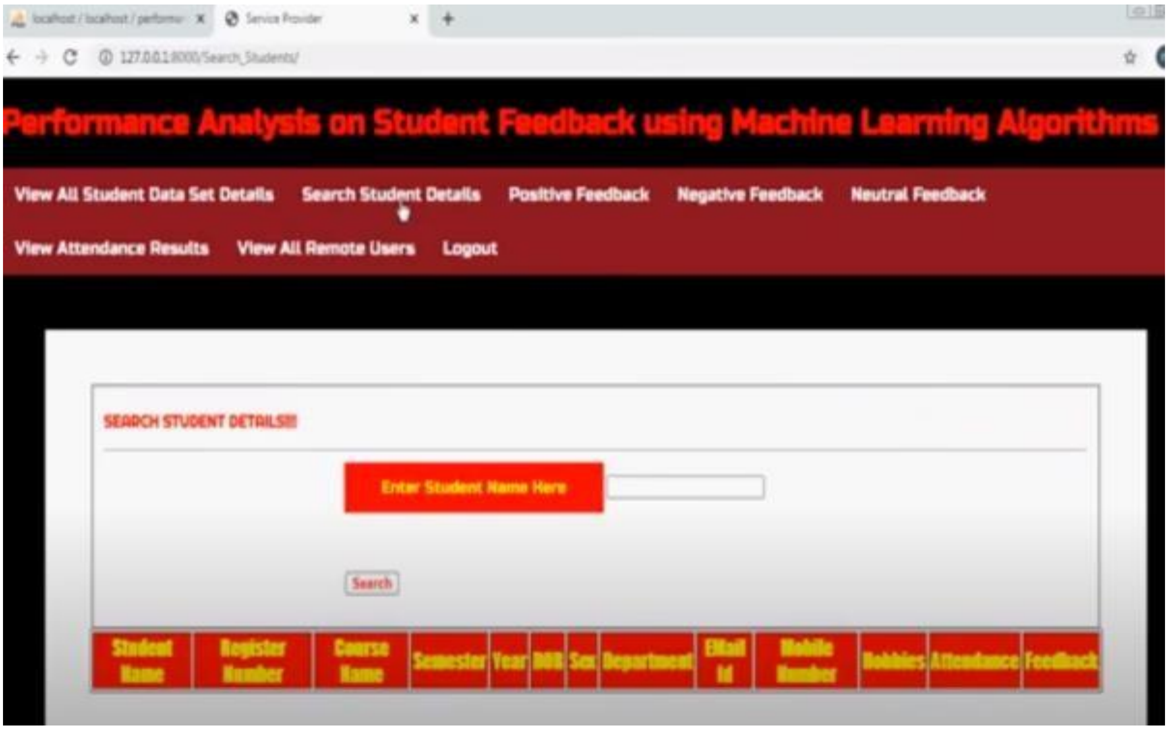


FIG 3: SEARCH STUDENT DETAILS

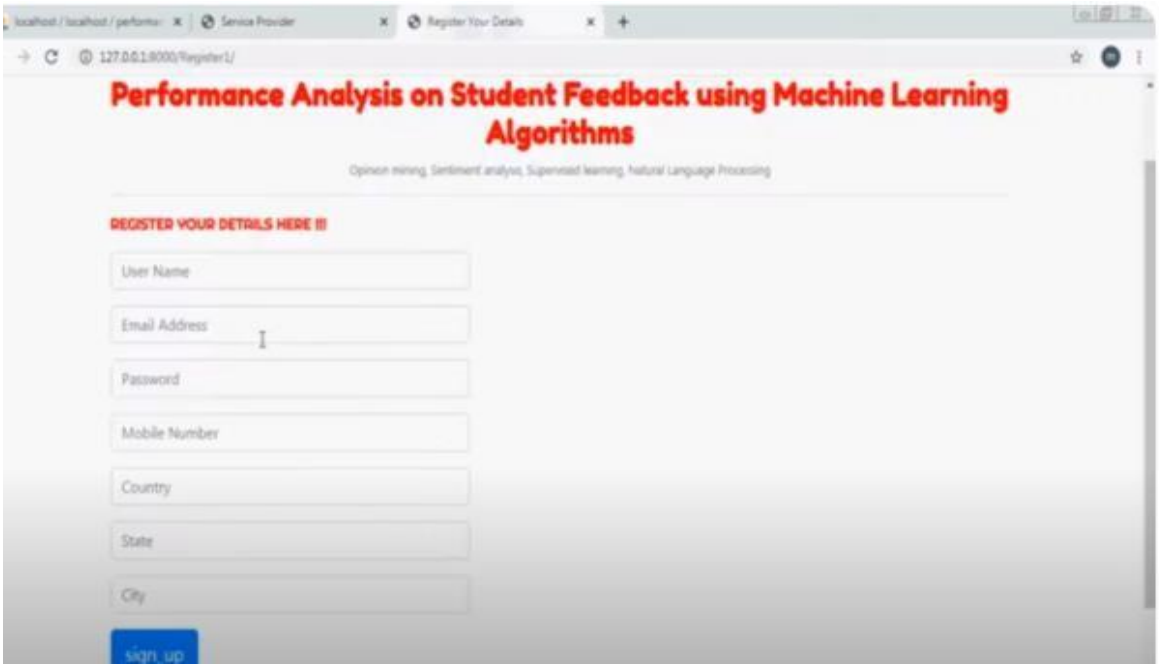


FIG 4: REGISTRATION DETAILS

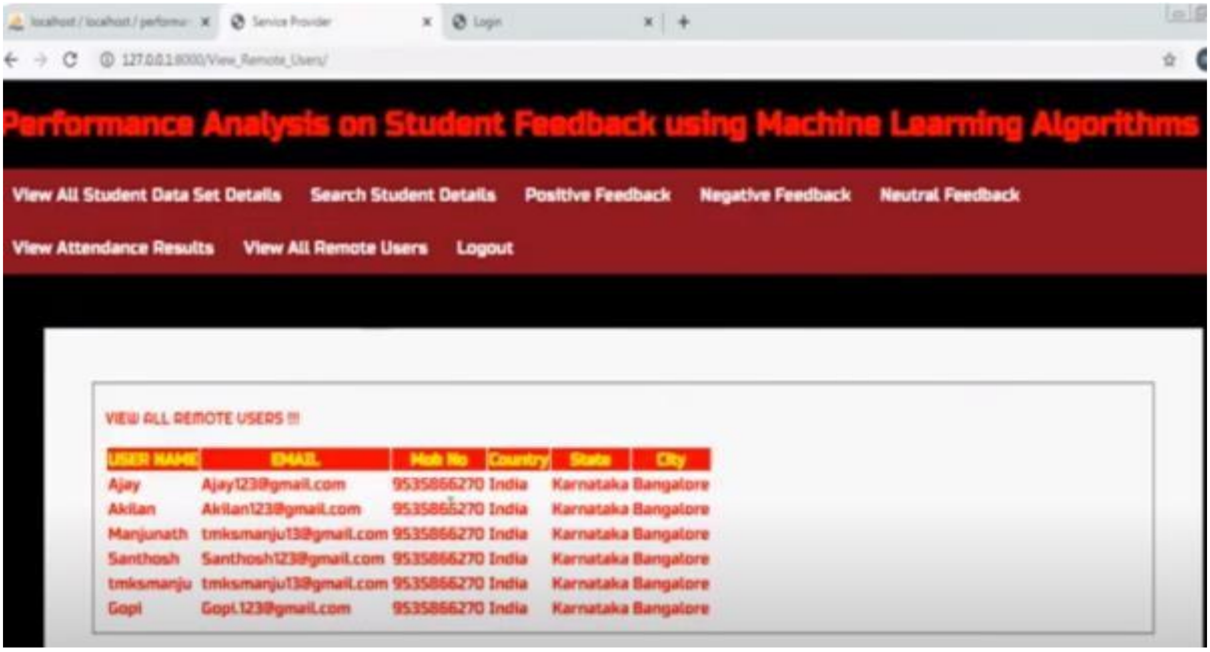


FIG 5: USER DETAILS

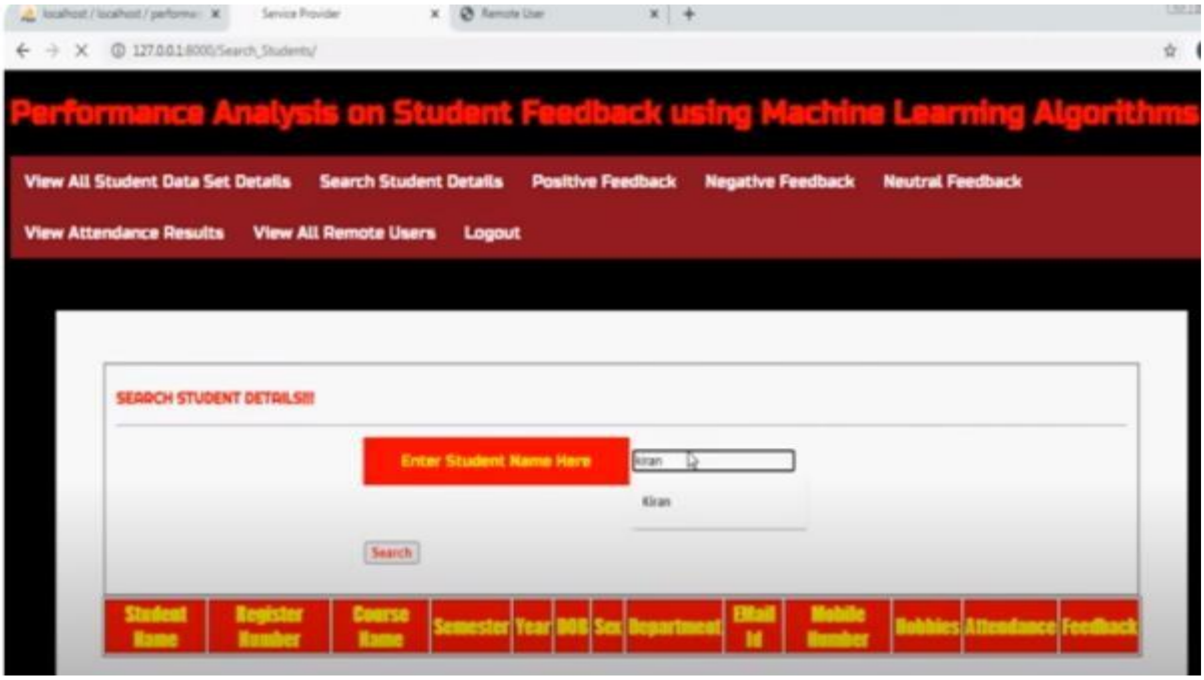


FIG 6: OUTPUT WITH STUDENTS DETAILS

CHAPTER 8

CONCLUSION

8.1 CONCLUSION

Two sets of exterminates have been carried out in this study using regression and classification analysis. The results of predicting students' assessments grades model show that the students' performance in a particular assignment relies on students' mark in the previous assignment within single Courses. The researchers conclude that students' prior grade point average (GPA) with a low mark is considered as a significant factor of withdrawal from the next course in the traditional classroom setting, both conventional classroom setting and virtual class share similar characteristic in term of the effective of pervious performance into student learning achievement in the future.

The final student performance predictive model revealed that student engagement with digital material has a significant impact on their success in the entire course. The findings' results also demonstrate that long-term students' performance achieves better accuracy than students' assessments grades prediction model, due to the exclusion of temporal features in regression analysis. The date of student deregistration from the course is a valuable predictor that is significantly correlated with student performance. With the regression analysis, the data does not provide the last date of students' activity prior to undertaken assessments. The findings' results have been recommended to take into account the temporal features on predicting of subsequent assessments grades.

Future research direction involves the use of temporal features for predicting students' assessments grades model. With temporal feature time series analysis will be untaken, might be more advanced machine leering will be utilized.

CHAPTER 9

FUTURE ENHANCEMENTS

9.1 FUTURE ENHANCEMENTS

Here are some potential future enhancements to further improve the accuracy and effectiveness of the model for predicting students' performance in online courses:

Incorporating Natural Language Processing (NLP): Analyze the textual data from discussion forums, assignment submissions, and feedback to extract insights about students' sentiments, comprehension levels, and engagement. NLP techniques can help identify common challenges, misconceptions, or areas where students need additional support.

Dynamic Feature Updating: Implement mechanisms to continuously update the model with real-time data on student interactions, performance, and behavior. This could involve streaming data processing techniques to adapt the model to evolving learning patterns and trends.

Personalized Learning Pathways: Develop algorithms that recommend personalized learning pathways based on individual student profiles, preferences, and learning styles. By leveraging machine learning to analyze historical data and predict future performance, the platform can dynamically adjust course content, difficulty levels, and supplementary resources to cater to each student's needs.

Integration with Learning Analytics Platforms: Integrate the predictive model with learning analytics platforms to provide educators and administrators with actionable insights and recommendations. This could include early warning systems to identify at-risk students, adaptive learning interventions, and optimization of course design based on predictive analytics.

Multi-modal Data Fusion: Combine data from multiple sources such as video interactions, mouse movements, and eye-tracking to gain deeper insights into students' learning behaviors and cognitive processes. By fusing different modalities of data, the model can capture richer contextual information and improve the accuracy of performance predictions.

Transfer Learning and Domain Adaptation: Explore techniques such as transfer learning and domain adaptation to leverage pre-trained models or knowledge from related domains (e.g., traditional education) to enhance the performance of the predictive model in the online learning context. This can help overcome data scarcity and improve generalization across diverse student populations.

REFERENCES

[1] K. F. Hew and W. S. Cheung, “Students’ and instructors’ use of massive open online courses (MOOCs): Motivations and challenges,” Educ. Res.Rev., vol. 12, pp. 45–58, 2014.

[2] H. B. Shapiro, C. H. Lee, N. E. Wyman Roth, K. Li,M. Çetinkaya-Rundel, and D. A. Canelas,“Understanding the massive open online course(MOOC) student experience: An examination of attitudes, motivations, and barriers,” Comput. Educ.,vol. 110, pp. 35–50, 2017.

[3] J. Renz, F. Schwerer, and C. Meinel, “openSAP: Evaluating xMOOC Usage and Challenges for Scalable and Open Enterprise Education.,” Int. J. Adv.Corp. Learn., vol. 9, no. 2, pp. 34–39, 2016.

[4] S. Li, Q. Tang, and Y. Zhang, “A Case Study on Learning Difficulties and Corresponding Supports for Learning in cMOOCs| Une étude de cas sur les difficultés d’apprentissage et le soutien correspondantpour l’apprentissage dans les cMOOC,” Can. J. Learn.Technol. Rev. Can. l’apprentissage la Technol., vol.42, no. 2, 2016.

[5] S. Zutshi, S. O’Hare, and A. Rodafinos, “Experiences in MOOCs: The Perspective of Students,” Am. J.Distance Educ., vol. 27, no. 4, pp. 218–227,2013.

[6] Z. Wang, T. Anderson, L. Chen, and E. Barbera, “Interaction pattern analysis in cMOOCs based on the connectivist interaction and engagement framework,”Br. J. Educ. Technol., vol. 48, no. 2, pp. 683–699,2017.

[7] W. Xing and D. Du, “Dropout Prediction in MOOCs:Using Deep Learning for Personalized Intervention,”J. Educ. Comput. Res. p.0735633118757015., 2018.

[8] M. J. Gallego Arrufat, V. Gamiz Sanchez, and E.Gutierrez Santiuste, “Trends in Assessment in Massive Open Online Courses,” Educ. Xx1, vol. 18, no. 2, pp. 77–96, 2015.

[9] B. J. Hao Cen, Kenneth Koedinger and Carnegie,“Learning Factors Analysis – A General Method for Cognitive Model Evaluation and Improvement,” in In International Conference on Intelligent Tutoring Systems, 2006, vol. 8, pp. 164–175.

[10] K. R. Koedinger, E. A. Mclaughlin, and J. Stamper,“Automated Student Model Improvement,” in Educational Data Mining, proceedings of the 5thInternational Conference on, 2012, pp. 17–24.

[11] J. Sinclair and S. Kalvala, "Student engagement in massive open online courses," Int. J. Learn. Technol.,vol. 11, no. 3, pp. 218–237, 2016.

[12] J. Mullan, "Learning Analytics in Higher Education,"London, 2016.

[13] P. and K. Al-Shabandar, R., Hussain, A.J., Liatsis, "Detecting At-Risk Students With Early Interventions Using Machine Learning Techniques," IEEE Access,vol. 7, pp. 149464–149478, 2019.

[14] S. Jiang, A. E. Williams, K. Schenke, M. Warschauer,and D. O. Dowd, "Predicting MOOC Performance with Week 1 Behavior," in Proceedings of the 7th International Conference on Educational Data Mining (EDM), 2014, pp. 273–275.

[15] L. Analytics and C. Exchange, "OU Analyse: Analysing at - risk students at The Open University,"in in Conference, 5th International Learning Analytics and Knowledge (LAK) (ed.), 2015, no. October 2014.