# **CHAPTER – 1**

**INTRODUCTION**

**1.1 About the project:**

Student performance prediction is a critical aspect of educational data mining that helps institutions identify at-risk students, improve learning outcomes, and enhance academic planning, this study presents a machine learning-based approach to track and predict student performance in undergraduate programs. Various supervised learning models, including decision trees, random forests, support vector machines, and neural networks, are explored to analyse academic and behavioural data. Key performance indicators such as attendance, grades, participation, and demographic factors are considered to develop predictive models. The proposed system provides early intervention strategies by classifying students based on their risk levels and suggesting personalized academic support. The results demonstrate that machine learning models can significantly improve prediction accuracy

This project presents a machine learning approach for tracking and predicting student performance in undergraduate programs using **supervised machine learning techniques.** The proposed system leverages historical academic records, attendance, and behavioural data to train predictive models. **Decision Tree algorithms** are used for their interpretability, allowing educators to understand key factors influencing performance. **Support Vector Machine (SVM)** is employed to classify students into different performance categories with high accuracy, particularly for complex datasets. Additionally, **Naïve Bayes,** known for its simplicity and efficiency, is utilized to model probabilistic relationships between various academic and personal factors.

# **CHAPTER – 2**

**SYSTEM ANALYSIS**

**2.1. Existing System:**

Before the introduction of machine learning (ML)-based tracking and prediction systems, student performance evaluation relied on traditional methods. These conventional approaches had several limitations in terms of accuracy, efficiency, and adaptability.

The existing system for tracking and predicting student performance in an undergraduate program primarily relies on traditional assessment methods, such as exams, assignments, and attendance records. Universities and institutions use Learning Management Systems (LMS) to store and manage student data, but these systems often lack advanced predictive capabilities. Faculty members manually analyze student grades and performance trends, making it difficult to identify at-risk students early. Additionally, feedback mechanisms are often delayed, limiting the ability to provide timely interventions. As a result, the current system is reactive rather than proactive, leading to challenges in improving student outcomes efficiently.

**2.1.1 Drawbacks:**

* **Manual and Time-Consuming Analysis** – The current system relies on faculty members manually analysing student performance, which is inefficient and prone to human error.
* **Delayed Feedback and Intervention** – Traditional methods identify struggling students attendance, neglecting other important factors such as student engagement, learning patterns, and behavioural data.
* **Lack of Personalized Learning Support** – Without predictive analytics, institutions struggle to provide tailored guidance to students based on their individual learning needs.
* **Inefficiency in Identifying At-Risk Students** – The absence of advanced analytics makes it difficult to detect early signs of academic struggle, leading to higher dropout rates.

**2.2 Proposed System:**

The proposed system leverages machine learning techniques to track and predict student performance in undergraduate programs. By analysing key academic and behavioural data, such as attendance, grades, Participation the system provides real-time insights and personalized recommendations. Unlike traditional rule-based methods, this approach offers higher accuracy, dynamic updates, Live monitoring Advanced algorithms early identification of at-risk students.

A proposed system for tracking and predicting student performance in an undergraduate program using a machine learning approach would involve collecting and analysing various data points related to student progress. This data could include demographics, academic history, grades in relevant courses, attendance records, participation in extracurricular activities, and even data from learning management systems. Machine learning algorithms, such as regression models, decision trees, or neural networks, would then be trained on this data to identify patterns and predict future performance. The system could also incorporate natural language processing to analyse student feedback and identify potential issues.

This system could provide valuable insights to educators, allowing them to identify at-risk students early on and provide targeted interventions. It could also help personalize learning experiences by identifying individual strengths and weaknesses. Furthermore, the system could be used to evaluate the effectiveness of different teaching methods and curriculum design. By continuously tracking and predicting student performance, the proposed system could contribute to improved learning outcomes and a more personalized educational experience for undergraduate students.

**2.2.1 Advantages:**

* **Early identification of at-risk students:** By analysing patterns in student data, the system can identify students who are likely to struggle academically. This allows educators to intervene early and provide personalized support, such as tutoring, counselling, or adjusted learning plans, potentially preventing students from falling behind or dropping out.
* **Personalized learning experiences:** The system can identify individual student strengths and weaknesses, enabling educators to tailor learning experiences to meet specific needs. This could involve recommending relevant resources, adjusting the pace of learning, or providing targeted feedback. Personalized learning can lead to increased student engagement, motivation, and ultimately, better learning outcomes.
* **Improved teaching methods and curriculum design:** By analysing student performance data, educators can gain insights into the effectiveness of different teaching methods and curriculum design. This data-driven approach can inform decisions about how to improve instruction and optimize the learning environment for all students.
* **Enhanced operational efficiency:** The system can automate administrative tasks, such as tracking attendance and grading assignments, freeing up educators' time to focus on teaching and student support. This can lead to increased efficiency and better allocation of resources within the undergraduate program.

**2.3 Module Description:**

My project contains three Modules:

1. Home
2. User login
3. About us

* **Use login :** Our project main priority is user module, that’s the reason we are showing the user module performances. To login the user, we have to use the credentials are user name and password.

**Username:** admin

**Password:** admin

### User modules contains sub modules those are

### **Load &Process Dataset:** In this module we are loading the data and processing the inputs to the output values with the help of two contained datasets. Those are

**1.Train dataset:** Train dataset is nothing but training the machine based on our requirement input to generate specified output to the given required input.

**2. Test dataset:** The main purpose of test dataset is to rectifying and passing the testcases of trained data failed testcases.

* **Train ML Model:** In this module we are training our system labelled dataset to generate the better accuracy,

Now we are using ML techniques to train the machine for the purpose of generating the better accuracy output.

* **Predict performance:** This module is used to predict the performance of the students based on the certain requirement fields. Those are
* Study time
* School support
* Paid classes
* Failures
* Period 1 score
* Period 2 score
* Period 3 score

**2.4 Feasibility Study:**

**Model Testing:**

After the model training is complete, and it is understood that the model shows the right result, it can be saved by: model. save(“name\_of\_file.h5”).

**Model Evaluation**:

Finally, the saved model can be used in the real world. The name of this phase

The feasibility of the project is analyzed in this phase and 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, some 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
  + 1. **Economical Feasibility:**

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures 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 the customized products had to be purchased.

* + 1. **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 on the available technical resources. This will lead to high demands on the available technical resources.

* + 1. **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.

CHAPTER – 3

**REQUIREMENT ANALYSIS**

**HARDWARE SYSTEM REQUIREMENTS:**

**➢** Processor – intel core i3 or Ryzen3

➢ RAM - 4 GB or above

➢ Hard Disk - 500 GB

➢ Solid State Drive – 256 GB

**SOFTWARE SYSTEM REQUIREMENTS:**

* Operating system : Windows 7 or above
* Coding Language : Python.
* Front-End : HTML,CSS,JAVASCRIPT
* Back-End: SQL Server
* Framework :Web-based application(Django)

**3.3 functional Requirements:**

* Data owner can maintain the details and he can add, delete and update the files.
* Data owner uploads the files and makes them available to user.
* Admin generates a schedule and sends it to the user
* Admin can maintain the details of number of users and the downloaded data.
* User can register in the site and get a login-id and password.
* With the login details the customer login into the cloud and check for available files.
* User will enter the details of filename, and may download the file after registering in the site.

**3.4 Non-functional Requirements:**

* **Reliability**

Reliability is improved if multiple redundant users register into database , which makes well-designed cloud computing suitable for business continuity and disaster recovery.

* **Performance**:

Performance is monitored and consistent and loosely coupled architectures are constructed using cloud services as the system interface.

* **Security :**

Security could improve due to centralization of data, increased security-focused resources, etc., but concerns can persist about loss of control over certain sensitive data, and the lack of security for stored kernels. Security is often as good as or better than other traditional systems, in part because providers are able to devote resources to solving security issues that many customers cannot afford. However, the complexity of security is greatly increased when data is distributed over a wider area or greater number of devices and in multi-tenant systems that are being shared by unrelated users. In addition, user access to security audit logs may be difficult or impossible. Private cloud installations are in part motivated by users' desire to retain control over the infrastructure and avoid losing control of information security.

* **Maintenance**:

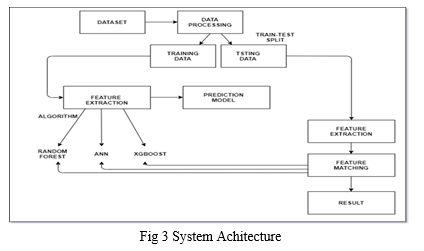
Cloud computing applications is easier, because they do not need to be installed on each user's computer and can be accessed from different places.

# 

# **CHAPTER – 4**

#### SYSTEM DESIGN

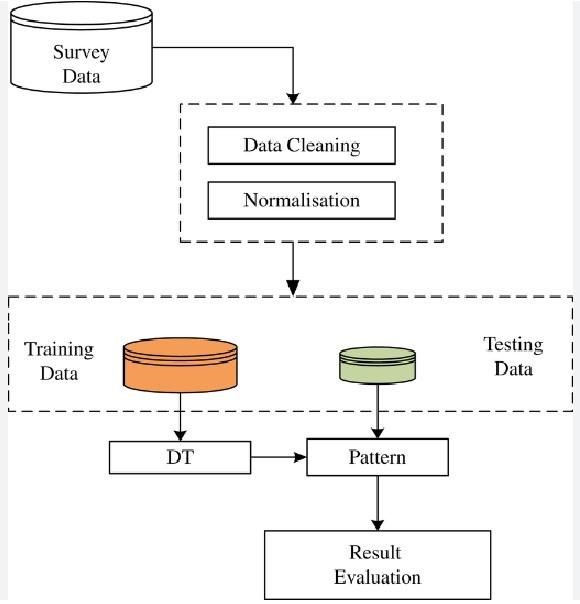
4.1 **System Architecture:**

****

**Fig 4.1 System Architecture**

**4.2 Data Flow Diagram:**

* The DFD is also called as bubble chart. It is a simple graphical formalism that can be used to represent a system in terms of input data to the system, various processing carried out on this data, and the output data is generated by this system.
* The data flow diagram (DFD) is one of the most important modeling tools. It is used to model the system components. These components are the system process, the data used by the process, an external entity that interacts with the system and the information flows in the system.
* DFD shows how the information moves through the system and how it is modified by a series of transformations. It is a graphical technique that depict sin formation flow and the transformations that are applied as data moves from input to output.
* DFD is also known as bubble chart. A DFD may be used to represent a system at any level of abstraction. DFD may be partitioned into levels that represent increasing information flow and functional detail.



**Fig:4.2 Data Flow Diagram**

**4.3 Unified Modeling Language Design:**

UML stands for Unified Modeling Language. UML is a standardized generalpurpose 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.

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

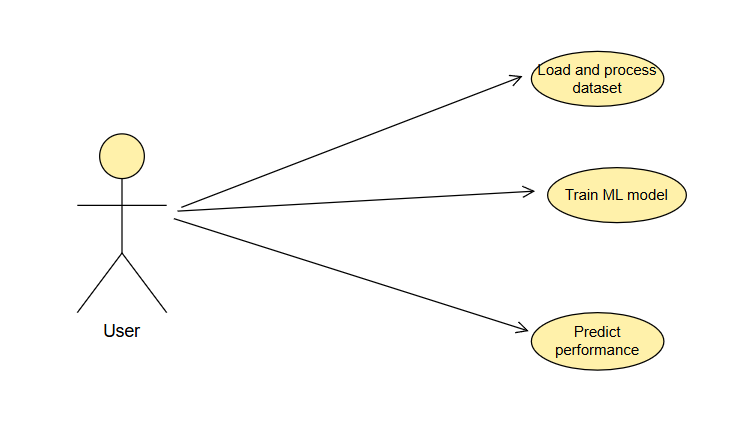
**GOALS:**

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

1. Provide users 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 particular 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.

**4.3.1 Use Case Diagram:**

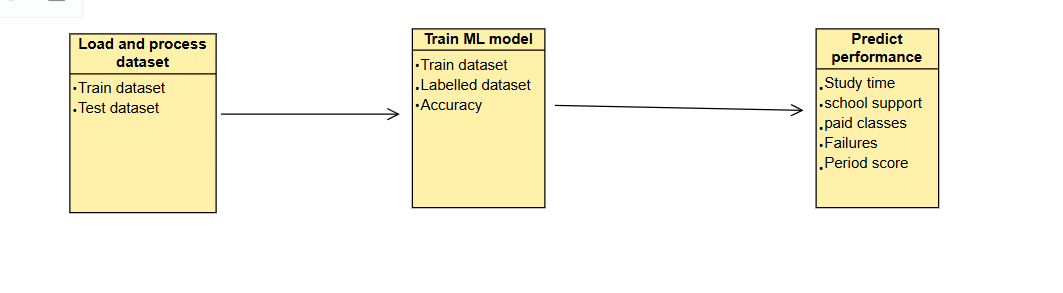
A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies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



**Fig 4.3 Use case Diagram for user**

**4.3.2 Class Diagram:**

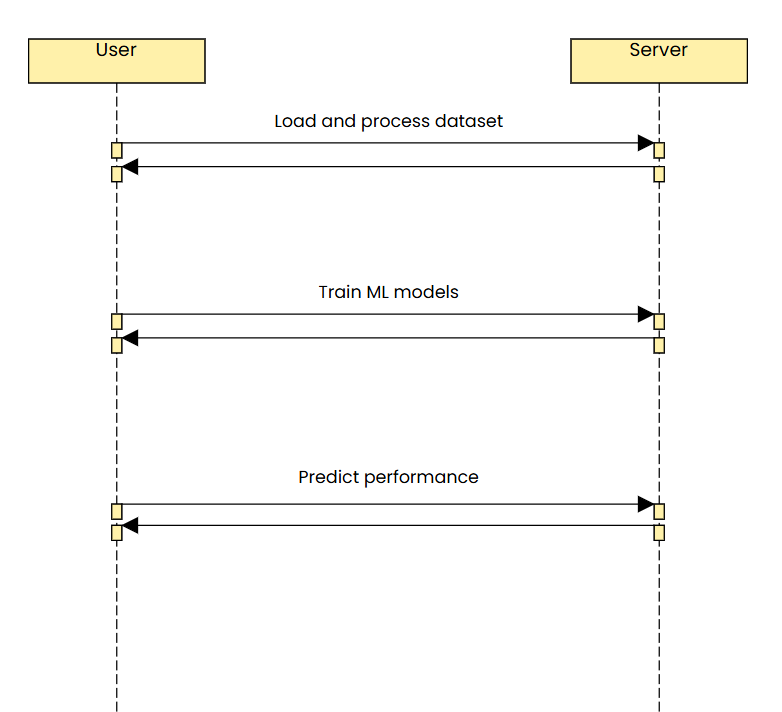
A class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes. It explains which class contains information.



###### Fig.4.4 Class Diagram for user

**4.3.3 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.



**Fig 4.5 Sequence Diagram for user**

**4.4 Input Design:**

The input design is the link between the information system and the user. It comprises the developing specification and procedures for data preparation and those steps are necessary to put transaction data in to a usable form for processing can be achieved by inspecting the computer to read data from a written or printed document or it can occur by having people keying the data directly into the system. The design of input focuses on controlling the amount of input required, controlling the errors, avoiding delay, avoiding extra steps and keeping the process simple. The input is designed in such a way so that it provides security and ease of use with retaining the privacy. Input Design considered the following things:

* What data should be given as input?
* How the data should be arranged or coded?
* The dialog to guide the operating personnel in providing input.
* Methods for preparing input validations and steps to follow when error occur.

**Objectives:**

1. Input Design is the process of converting a user-oriented description of the input into a computer-based system. This design is important to avoid errors in the data input process and show the correct direction to the management for getting correct information from the computerized system.
2. It is achieved by creating user-friendly screens for the data entry to handle large volume of data. The goal of designing input is to make data entry easier and to be free from errors. The data entry screen is designed in such a way that all the data manipulates can be performed. It also provides record viewing facilities.
3. When the data is entered it will check for its validity. Data can be entered with the help of screens. Appropriate messages are provided as when needed so that the user will not be in maize of instant. Thus the objective of input design is to create an input layout that is easy to follow.

**4.5 Output Design:**

A quality output is one, which meets the requirements of the end user and presents the information clearly. In any system results of processing are communicated to the users and to other system through outputs. In output design it is determined how the information is to be displaced for immediate need and also the hard copy output. It is the most important and direct source information to the user. Efficient and intelligent output design improves the system’s relationship to help user decision-making.

1. Designing computer output should proceed in an organized, well thought out manner; the right output must be developed while ensuring that each output element is designed so that people will find the system can use easily and effectively. When analysis design computer output, they should Identify the specific output that is needed to meet the requirements. 2. Select methods for presenting information.

3. Create document, report, or other formats that contain information produced by the system.

The output form of an information system should accomplish one or more of the following objectives.

* + Convey information about past activities, current status or projections of the
  + Future.
  + Signal important events, opportunities, problems, or warnings.
  + Trigger an action.
  + Confirm an action.

**CHAPTER - 5**

**IMPLEMENTATION**

**5.1 Machine Learning Technology:**

Machine learning is a discipline that deals with programming the systems so as to make them automatically learn and improve with experience. Here, learning implies recognizing and understanding the input data and taking informed decisions based on the supplied data. It is very difficult to consider all the decisions based on all possible inputs. To solve this problem, algorithms are developed that build knowledge from a specific data and past experience by applying the principles of statistical science, probability, logic, mathematical optimization, reinforcement learning, and control theory.

**5..2 Steps Involved in Machine Learning:**

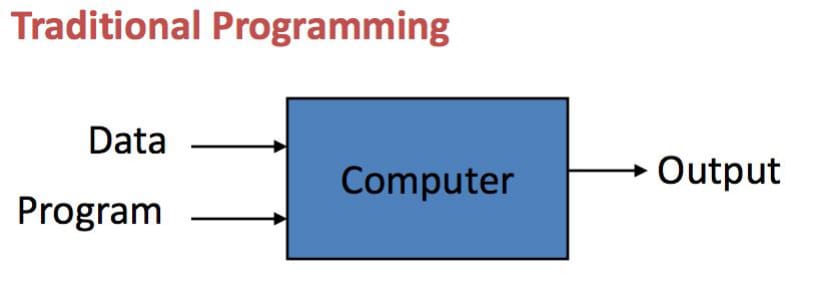
A machine learning project involves the following steps

* + Defining a Problem
  + Preparing Data
  + Evaluating Algorithms
  + Improving Results • Presenting Results

Machine Learning (ML) is an automated learning with little or no human intervention. It involves programming computers so that they learn from the available inputs. The main purpose of machine learning is to explore and construct algorithms that can learn from the previous data and make predictions on new input data.The input to a learning algorithm is training data, representing experience, and the output is any expertise, which usually takes the form of another algorithm that can perform a task. The input data to a machine learning system can be numerical, textual, audio, visual, or multimedia.

**5.3 Machine Learning vs. Traditional Programming:**

Traditional programming differs significantly from machine learning. In traditional programming, programmers code all the rules in consultation with an expert in the industry for which software is being developed. Each rule is based on a logical foundation; the machine will execute an output following the logical statement. When the system grows complex, more rules need to be written. It can quickly become sustainable to maintain.



**Fig 5.1: Traditional Programming**

Machine learning is supposed to overcome this issue. The machine learns how the input and output data are correlated and it writes a rule. The programmers do not need to write new rules each time there is new data. The algorithms adapt in response to new data and experiences to improve efficacy over time.

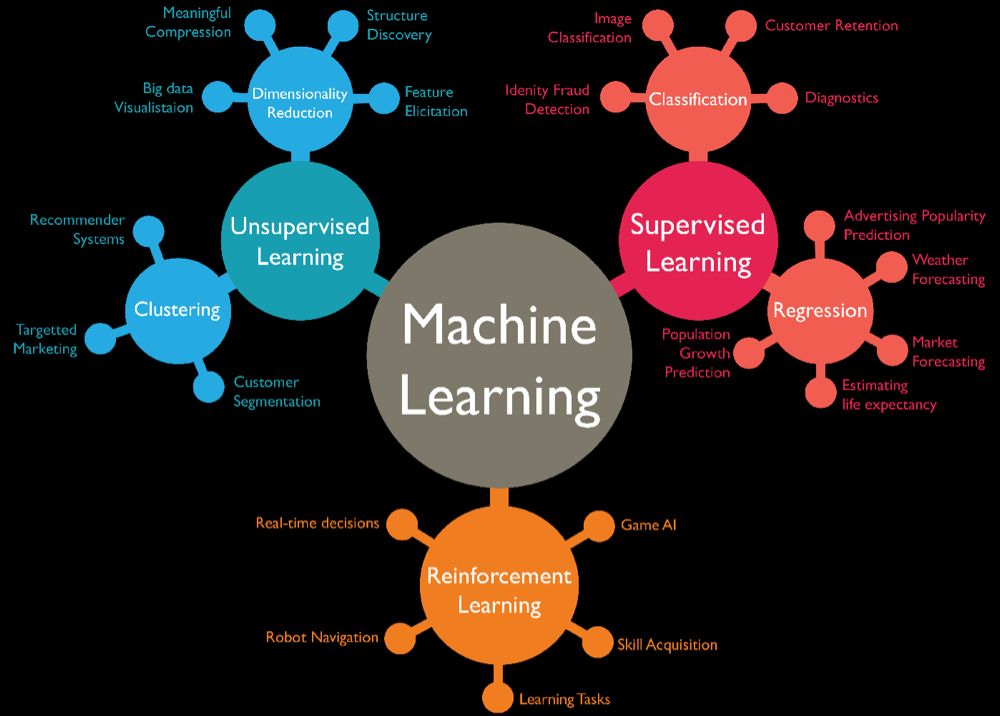
## 

**5.2: Machine Learning**

**5.4 How does Machine learning work?**

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 it’s feed a previously unseen example, the machine has difficulties to predict.

5.5 Machine learning Algorithms and where they are used?



**Fig 5.3: Machine Learning Algorithms**

**5.6 Types of Machine Learning:**

There are four categories of machine learning algorithms as shown below

* Supervised learning algorithm
* Unsupervised learning algorithm
* Semi-supervised learning algorithm
* Reinforcement learning algorithm

**5.6.1 Supervised Learning:**

Supervised learning is commonly used in real world applications, such as face and speech recognition, products or movie recommendations, and sales forecasting.

Supervised learning can be further classified into two types - Regression and Classification. Regression trains on and predicts a continuous-valued response, for example predicting real estate prices. Classification attempts to find the appropriate class label, such as analyzing positive/negative sentiment, male and female persons, benign and malignant tumors, secure and unsecure loans etc. In supervised learning, learning data comes with description, labels, targets or desired outputs and the objective is to find a general rule that maps inputs to outputs. This kind of learning data is called labeled data. The learned rule is then used to label new data with unknown outputs. Supervised learning involves building a machine learning model that is based on labeled samples.

**5.6.2 Unsupervised Learning:**

Unsupervised learning is used to detect anomalies, outliers, such as fraud or defective equipment, or to group customers with similar behaviors for a sales campaign. It is the opposite of supervised learning. When learning data contains only some indications without any description or labels, it is up to the coder or to the algorithm to find the structure of the underlying data, to discover hidden patterns, or to determine how to describe the data. We may not exactly know what the criteria of classification would be. So, an unsupervised learning algorithm tries to classify the given dataset into a certain number of groups in an optimum way.

**5.6.3 Semi-supervised Learning:**

If some learning samples are labeled, but some other are not labeled, then it is semi-supervised learning. It makes use of a large amount of unlabeled data for training and a small amount of labeled data for testing. Semi-supervised learning is applied in cases where it is expensive to acquire a fully labeled dataset while more practical to label a small subset.

**5.6.4 Reinforcement Learning:**

Here learning data gives feedback so that the system adjusts to dynamic conditions in order to achieve a certain objective. The system evaluates its performance based on the feedback responses and reacts accordingly.

**5.7 Challenges and Limitations of Machine learning:**

The primary challenge of machine learning is the lack of data or the diversity in the dataset. A machine cannot learn if there is no data available. Besides, a dataset with a lack of diversity gives the machine a hard time. A machine needs to have heterogeneity to learn meaningful insight. It is rare that an algorithm can extract information when there are no or few variations. It is recommended to have at least 20 observations per group to help the machine learn. This constraint leads to poor evaluation and prediction.

* 1. **Application of Machine learning :**

**1 Augmentation:**

Machine learning, which assists humans with their day-to-day tasks, personally or commercially without having complete control of the output. Such machine learning is used in different ways such as Virtual Assistant, Data analysis, software solutions.

**2.Automation:**

Machine learning, which works entirely autonomously in any field without the need for any human intervention. For example, robots performing the essential process steps in manufacturing plants.

**3.Finance Industry**:

Machine learning is growing in popularity in the finance industry. Banks are mainly using ML to find patterns inside the data but also to prevent fraud.

1. Government organization

The The government makes use of ML to manage public safety and utilities.Take the example of China with the massive face recognition.

**4.Marketing**

Broad use of AI is done in marketing thanks to abundant access to data. Before the age of mass data, researchers develop advanced mathematical tools like Bayesian analysis to estimate the value of a customer. With the boom of data, marketing department relies on AI to optimize the customer relationship and marketing campaign.

* 1. **Why is Machine Learning important?**

Machine learning is the best tool so far to analyze, understand and identify a pattern in the data. One of the main ideas behind machine learning is that the computer can be trained to automate tasks that would be exhaustive or impossible for a human being. The clear breach from the traditional analysis is that machine learning can take decisions with minimal human intervention.

* 1. **Python:**

Python is a general-purpose interpreted, interactive, object-oriented, and high-level programming language. An interpreted language, Python has a design philosophy that emphasizes code readability (notably using whitespace indentation to delimit code blocks rather than curly brackets or keywords), and a syntax that allows programmers to express concepts in fewer lines of code than might be used in languages such as C++or Java. It provides constructs that enable clear programming on both small and large scales. Python interpreters are available for many operating systems. Python, the reference implementation of Python, is open source software and has a community-based development model, as do nearly all of its variant implementations. CPython is managed by the non-profit Python Software Foundation. Python features a dynamic type system and automatic memory management.

It supports multiple programming paradigms, including object oriented, imperative, and functional and has a large and comprehensive standard library.

* 1. **python libraries for machine learning:** 
     1. **NumPy**:

NumPy is a library for the Python programming language, adding support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays. The core functionality of NumPy is its

"ndarray", for n-dimensional array, data structure. These arrays are strided views on memory. In contrast to Python's built-in list data structure, these arrays are homogeneously typed: all elements of a single array must be of the same type.

* + 1. **Pandas:**

Pandas is the most popular python library that is used for data analysis. It provides highly optimized performance with back-end source code is purely written in C or Python.

We can analyze data in pandas with Series and Data Frames.

* + 1. **Matplotlib:**

Matplotlib is a python library used to create 2D graphs and plots by using python scripts. It has a module named pyplot which makes things easy for plotting by providing feature to control line styles, font properties, formatting axes etc. It supports a very wide variety of graphs and plots namely histogram, bar charts, power spectra, error charts etc. It is used along with NumPy to provide an environment that is an effective open source alternative for MatLab. It can also be used with graphics toolkits like PyQt and wxPython.

* + 1. **Scikit-learn:**

Scikit-learn is a machine learning library for Python. It features several regression, classification and clustering algorithms including SVMs, gradient boosting, k-means, random forests and DBSCAN. Scikit is written in Python (most of it) and some of its core algorithms are written in Cython for even better performance. Scikit-learn is used to build models and it is not recommended to use it for reading, manipulating and summarizing data as there are better frameworks available for the purpose. It is open source and released under BSD license.

* + 1. **Keras:**

Keras is a powerful and user-friendly open-source Python library that provides a high-level API for building and training neural networks. It acts as an interface to several lower-level deep learning frameworks like TensorFlow, CNTK, and Theano (though TensorFlow is now the primary backend). Keras simplifies the process of designing, compiling, training, and evaluating various types of neural network architectures, from simple feedforward networks to complex recurrent and convolutional networks. Its focus on user experience makes it particularly well-suited for rapid prototyping and experimentation, allowing developers to quickly build and test different models. Keras emphasizes modularity, allowing users to combine pre-built layers and components to create custom models, and its integration with Python makes it easy to use alongside other scientific computing libraries like NumPy and Pandas.

**Uses of Python:**

* + 1. **Applications:**

Python can be used to develop different applications like web applications, graphic user interface based applications, software development application, scientific and numeric applications, network programming, Games and 3D applications and other business applications. It makes an interactive interface and easy development of applications.

* + 1. **Multiple Programming paradigms**:

Python is also used because of its providing continuous support to several programming paradigms. As it supports object-oriented programming and structured programming. Python has features, which also support various concepts of functional programming language. It is used for dynamic type system and automatic memory management. Python language features and programming paradigms allow you for developing the small as well as large applications. It can be used for complex software applications.

* + 1. **Robust Standard Library**:

Python has a large and robust standard library to use for developing the applications. It also makes the developers use Python over other languages. The standard library helps in using the different range of modules available for Python. As this module helps you in adding the functionality without writing any more code. To get the information about various modules, documentation on python standard library can be referred. While developing any web application, implementing web services, performing string operations and other usages like interface protocol, the standard library documentation helps.

* + 1. **Compatible with Major Platforms and Systems:**

Python is mainly compatible with major platforms and systems because of which it is used mainly for developing applications. With help of python interpreters, python code can be run on specific platforms and tools as it supports many operating systems. As python is an interpreted high-level programming language and it allows you to run the code on multiple platforms.

The new and modified code can be executed without recompiling and its impact can be monitored or checked. It means it’s not required to recompile the code after every change. This feature helps in saving the development time of the developers. 5 Access of Database:

Uses of Python also helps in accessing the database easily. Python helps in customizing the interfaces of different databases like MySQL, Oracle, Microsoft SQL Server, PostgreSQL, and other databases. It has an object database like Durus and ZODB.

It is used for standard database API and freely available for download.

1. **Code Readability:**

Python code is easy to read and maintained. It is easily reusable as well wherever it is required. Python’s having simple syntax, which allows the different concepts to develop without writing any additional code. The code should be of good quality and easy to maintain the source code and simplify the maintenance, which is required to develop the software application. It also emphasizes code readability, which is the great feature, unlike other programming languages. It helps in building custom applications and clean code helps in maintaining and updating the software applications without putting extra effort on the same code.

1. **Simplify Complex Software Development:**

Applications of Python is used to simplifying the complex software development process as it is a general-purpose programming language. It is used for developing the complex application like scientific and numeric application, and for both desktop and web applications. Python has features like analyzing data and visualization, which helps in creating custom solutions without putting extra effort and time. It helps you to visualize and present data in an effective way.

1. **Many Open Source Frameworks and Tools:**

Python is open source and easily available. This also helps in costing the software development significantly. There are many open source applications of python frameworks, libraries, and development tools for developing the application without putting extra cost.

Python frameworks simplify and make the process faster for web application development and the frameworks are Django, Flask, pyramid etc. Python GUI frameworks are available for developing the GUI based application.

1. **Adopt Test Driven Development:**

Python makes coding easier as well as testing with help of adopting Test Driven Development approach. The test cases can be easily written before any code development. Whenever the code development started, the written test cases can start testing the code simultaneously and provides the result. These can also be used for checking or testing the pre-requirements based on the source code.

1. **Other applications for which python is used:**

There are other applications for which python is used that are Robotics, web scraping, scripting, artificial intelligence, data analysis, machine learning, face detection, color detection, 3D CAD applications, console-based applications, audio-based applications, video-based applications, enterprise applications, and applications for Images etc. These are some major applications used.

**5.13 Neural network:**

Is a machine learning algorithm, which is built on the principle of the organization and functioning of biological neural networks. This concept arose in an attempt to simulate the processes occurring in the brain by Warren McCulloch and Walter Pitts in 1943.

Neural networks consist of individual units called neurons. Neurons are located in a series of groups — layers (see figure allow).

Neurons in each layer are connected to neurons of the next layer. Data comes from the input layer to the output layer along these compounds.

Each individual node performs a simple mathematical calculation. Тhen it transmits its data to all the nodes it is connected to.

The last wave of neural networks came in connection with the increase in computing power and the accumulation of experience.

That brought Deep learning, where technological structures of neural networks have become more complex and able to solve a wide range of tasks that could not be effectively solved before. Image classification is a prominent.

**5.15 Sample code:**

from django.shortcuts import render

from django.template import RequestContext

from django.contrib import messages

from django.http import HttpResponse

from django.core.files.storage import FileSystemStorage

import os

import pandas as pd

import numpy as np

from sklearn.preprocessing import LabelEncoder

from sklearn.model\_selection import train\_test\_split

from sklearn.preprocessing import StandardScaler

from sklearn.ensemble import GradientBoostingClassifier

from xgboost import XGBClassifier

from sklearn.neighbors import KNeighborsClassifier

from sklearn import svm

from sklearn.ensemble import RandomForestClassifier

from sklearn.linear\_model import LogisticRegression

from sklearn.metrics import confusion\_matrix

import seaborn as sns

from sklearn.metrics import precision\_score

from sklearn.metrics import recall\_score

from sklearn.metrics import f1\_score

from sklearn.metrics import accuracy\_score

import matplotlib.pyplot as plt

import io

import base64

global uname

global X\_train, X\_test, y\_train, y\_test

accuracy, precision, recall, fscore = [], [], [], []

graph\_data = []

dataset = pd.read\_csv("Dataset/student-por.csv")

labels = np.unique(dataset['FinalResult'])

label\_encoder = []

columns = dataset.columns

types = dataset.dtypes.values

for i in range(len(types)):

name = types[i]

if name == 'object': #finding column with object type

le = LabelEncoder()

dataset[columns[i]] = pd.Series(le.fit\_transform(dataset[columns[i]].astype(str)))#encode all str columns to numeric

label\_encoder.append([columns[i], le])

dataset.fillna(0, inplace = True)

Y = dataset['FinalResult'].ravel()

dataset.drop(['FinalResult'], axis = 1,inplace=True)

X = dataset.values

sc = StandardScaler()

X = sc.fit\_transform(X)

indices = np.arange(X.shape[0])

np.random.shuffle(indices)

X = X[indices]

Y = Y[indices]

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, Y, test\_size=0.5) #split dataset into train and test

def calculateMetrics(algorithm, predict, y\_test):

global accuracy, precision, recall, fscore

a = accuracy\_score(y\_test,predict)\*100

p = precision\_score(y\_test, predict,average='macro') \* 100

r = recall\_score(y\_test, predict,average='macro') \* 100

f = f1\_score(y\_test, predict,average='macro') \* 100

accuracy.append(float(round(a, 2)))

precision.append(float(round(p, 2)))

recall.append(float(round(r, 2)))

fscore.append(float(round(f, 2)))

knn = KNeighborsClassifier(n\_neighbors=2)

knn.fit(X\_train, y\_train)

predict = knn.predict(X\_test)

calculateMetrics("KNN", predict, y\_test)

rf = RandomForestClassifier(n\_estimators=1)

rf.fit(X\_train, y\_train)

predict = rf.predict(X\_test)

calculateMetrics("Random Forest", predict, y\_test)

svm\_cls = svm.SVC(C=1.0, kernel="linear")

svm\_cls.fit(X\_train, y\_train)

predict = svm\_cls.predict(X\_test)

calculateMetrics("SVM", predict, y\_test)

gb\_cls = GradientBoostingClassifier(n\_estimators=10)

gb\_cls.fit(X\_train, y\_train)

predict = gb\_cls.predict(X\_test)

calculateMetrics("Gradient Boosting", predict, y\_test)

lr\_cls = LogisticRegression()

lr\_cls.fit(X\_train, y\_train)

predict = lr\_cls.predict(X\_test)

calculateMetrics("Logistic Regression", predict, y\_test)

xg\_cls = XGBClassifier(n\_estimators=1)

xg\_cls.fit(X\_train, y\_train)

predict = xg\_cls.predict(X\_test)

calculateMetrics("XGBoost", predict, y\_test)

xg\_cls = RandomForestClassifier()

xg\_cls.fit(X, Y)

predict = xg\_cls.predict(X\_test)

def PredictPerformance(request):

if request.method == 'GET':

return render(request, 'PredictPerformance.html', {})

def PredictPerformanceAction(request):

if request.method == 'POST':

global label\_encoder, rf, sc, labels, graph\_data, gb\_cls, xg\_cls

roll\_no = request.POST.get('rollno', False)

gender = request.POST.get('t1', False)

age = float(request.POST.get('t2', False).strip())

mother = request.POST.get('t3', False)

father = request.POST.get('t4', False)

reason = request.POST.get('t5', False)

guardian = request.POST.get('t6', False)

study = float(request.POST.get('t7', False).strip())

failure = float(request.POST.get('t8', False).strip())

school = request.POST.get('t9', False)

family = request.POST.get('t10', False)

paid = request.POST.get('t11', False)

activity = request.POST.get('t12', False)

internet = request.POST.get('t13', False)

free = float(request.POST.get('t14', False).strip())

out = float(request.POST.get('t15', False).strip())

health = float(request.POST.get('t16', False).strip())

absent = float(request.POST.get('t17', False).strip())

score1 = float(request.POST.get('t18', False).strip())

score2 = float(request.POST.get('t19', False).strip())

score3 = float(request.POST.get('t20', False).strip())

data = []

data.append([gender,age,mother,father,reason,guardian,study,failure,school,family,paid,activity,internet,free,out,health,absent,score1,score2,score3])

data = pd.DataFrame(data, columns=['sex','age','mother\_job','father\_job','reason','guardian','studytime','failures','schoolsup','famsup','paid','activities','internet','freetime','goout','health','absences','G1','G2','G3'])

testData = data.values

for i in range(len(label\_encoder)-1):

temp = label\_encoder[i]

name = temp[0]

le = temp[1]

data[name] = pd.Series(le.transform(data[name].astype(str)))#encode all str columns to numeric

data.fillna(0, inplace = True)

data = data.values

data = sc.transform(data)

predict = xg\_cls.predict(data)[0]

print(predict)

print(labels)

predict = int(predict)

predict = labels[predict]

graph\_data.append(predict)

status = ""

if predict == "Poor":

status = "Warning! Need more focus & hardwork"

output = "Roll No : "+roll\_no+"<br/>Overall Predicted Performance ===> "+predict+"<br/>"+status

context= {'data':output}

return render(request, 'PredictPerformance.html', context)

def Graphs(request):

if request.method == 'GET':

global graph\_data

output = "All Students Performance Graph"

gd = np.asarray(graph\_data)

unique, count = np.unique(gd, return\_counts=True)

plt.pie(count,labels=unique,autopct='%1.1f%%')

plt.title('Performance Graph')

plt.axis('equal')

buf = io.BytesIO()

plt.savefig(buf, format='png', bbox\_inches='tight')

plt.close()

img\_b64 = base64.b64encode(buf.getvalue()).decode()

context= {'data':output, 'img': img\_b64}

return render(request, 'ViewResult.html', context)

def TrainML(request):

if request.method == 'GET':

output = ''

output+='<table border=1 align=center width=100%><tr><th><font size="" color="black">Algorithm Name</th><th><font size="" color="black">Accuracy</th><th><font size="" color="black">Precision</th>'

output+='<th><font size="" color="black">Recall</th><th><font size="" color="black">FSCORE</th></tr>'

global accuracy, precision, recall, fscore

algorithms = ['KNN', 'Random Forest', 'SVM', 'Gradient Boosting', 'Logistic Regression', 'XGBoost']

for i in range(len(algorithms)):

output+='<td><font size="" color="black">'+algorithms[i]+'</td><td><font size="" color="black">'+str(accuracy[i])+'</td><td><font size="" color="black">'+str(precision[i])+'</td><td><font size="" color="black">'+str(recall[i])+'</td><td><font size="" color="black">'+str(fscore[i])+'</td></tr>'

output+= "</table></br>"

df = pd.DataFrame([['KNN','Precision',precision[0]],['KNN','Recall',recall[0]],['KNN','F1 Score',fscore[0]],['KNN','Accuracy',accuracy[0]],

['Random Forest','Precision',precision[1]],['Random Forest','Recall',recall[1]],['Random Forest','F1 Score',fscore[1]],['Random Forest','Accuracy',accuracy[1]],

['SVM','Precision',precision[2]],['SVM','Recall',recall[2]],['SVM','F1 Score',fscore[2]],['SVM','Accuracy',accuracy[2]],

['Gradient Boosting','Precision',precision[3]],['Gradient Boosting','Recall',recall[3]],['Gradient Boosting','F1 Score',fscore[3]],['Gradient Boosting','Accuracy',accuracy[3]],

['Logistic Regression','Precision',precision[3]],['Logistic Regression','Recall',recall[3]],['Logistic Regression','F1 Score',fscore[3]],['Logistic Regression','Accuracy',accuracy[3]],

['XGBoost','Precision',precision[3]],['XGBoost','Recall',recall[3]],['XGBoost','F1 Score',fscore[3]],['XGBoost','Accuracy',accuracy[3]],

],columns=['Algorithms','Metrics','Value'])

df.pivot\_table(index="Algorithms", columns="Metrics", values="Value").plot(kind='bar', figsize=(8, 4))

plt.title("All Algorithms Performance Graph")

plt.tight\_layout()

buf = io.BytesIO()

plt.savefig(buf, format='png', bbox\_inches='tight')

plt.close()

img\_b64 = base64.b64encode(buf.getvalue()).decode()

context= {'data':output, 'img': img\_b64}

return render(request, 'ViewResult.html', context)

def AdminLogin(request):

if request.method == 'GET':

return render(request, 'AdminLogin.html', {})

def index(request):

if request.method == 'GET':

return render(request, 'index.html', {})

def Aboutus(request):

if request.method == 'GET':

return render(request, 'Aboutus.html', {})

def LoadDataset(request):

if request.method == 'GET':

return render(request, 'LoadDataset.html', {})

def AdminLoginAction(request):

if request.method == 'POST':

global uname

username = request.POST.get('username', False)

password = request.POST.get('password', False)

if username == "admin" and password == "admin":

context= {'data':'welcome '+username}

return render(request, 'AdminScreen.html', context)

else:

context= {'data':'Invalid login details'}

return render(request, 'AdminLogin.html', context)

def LoadDatasetAction(request):

if request.method == 'POST':

myfile = request.FILES['t1'].read()

fname = request.FILES['t1'].name

if os.path.exists("StudentApp/static/"+fname):

os.remove("StudentApp/static/"+fname)

with open("StudentApp/static/"+fname, "wb") as file:

file.write(myfile)

file.close()

dataset = pd.read\_csv("StudentApp/static/"+fname)

columns = dataset.columns

dataset = dataset.values

output='<table border=1 align=center width=100%><tr>'

for i in range(len(columns)):

output += '<th><font size="" color="black">'+columns[i]+'</th>'

output += '</tr>'

for i in range(len(dataset)):

output += '<tr>'

for j in range(len(dataset[i])):

output += '<td><font size="" color="black">'+str(dataset[i,j])+'</td>'

output += '</tr>'

output+= "</table></br></br></br></br>"

#print(output)

context= {'data':output}

return render(request, 'ViewResult.html', context)

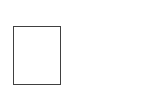
**CHAPTER-6**

**TESTING**

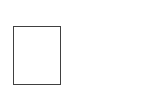
**6.1 Introduction**

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 tests**

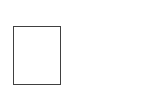
 **Unit testing:**

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 sic 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.

 **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.

 **Functional test:**

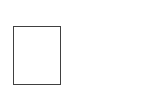
Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

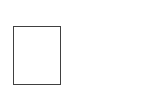
|  |  |  |
| --- | --- | --- |
| Valid Input | : | identified classes of valid input must be accepted. |
| Invalid Input | : | identified classes of invalid input must be rejected. |
| Functions output excercised | : | identified functions must be exercised. |
|  |  | identified classes of application outputs must be |

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

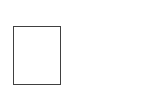
Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional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

 **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.

** 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.

** BlackBox 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.

**6.3 Test Cases:**

All the test cases mentioned above passed successfully.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test id | Test case name | Input | Expected output | Obtained output | Result |
| T1 | Student data Upload Dataset | CSV file with student data (ID, name, grades, etc.) | Dataset loaded successfully | Dataset loaded successfully | Success |
| T2 | Data Preprocessing | Uploaded student dataset | Data cleaned, transformed, and ready | Data cleaned, transformed, and ready | Success |
| T3 | Feature Engineering | Pre-processed student data | Engineered features (e.g., GPA, avg. grade) | Engineered features (e.g., GPA, avg. grade) | Success |
| T4 | Data Splitting | Engineered features | Training and testing datasets split | Training and testing datasets split | success |
| T5 | Prediction - GPA | Student data (new or existing) | Predicted GPA | Predicted GPA | Success |

**Tab: 6.1 Test case Specifications**

**6.4 Test Results:**

All the test cases mentioned above passed successfully. No defect encountered. 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.

**CHAPTER - 7**

**SAMPLE SCREENS**

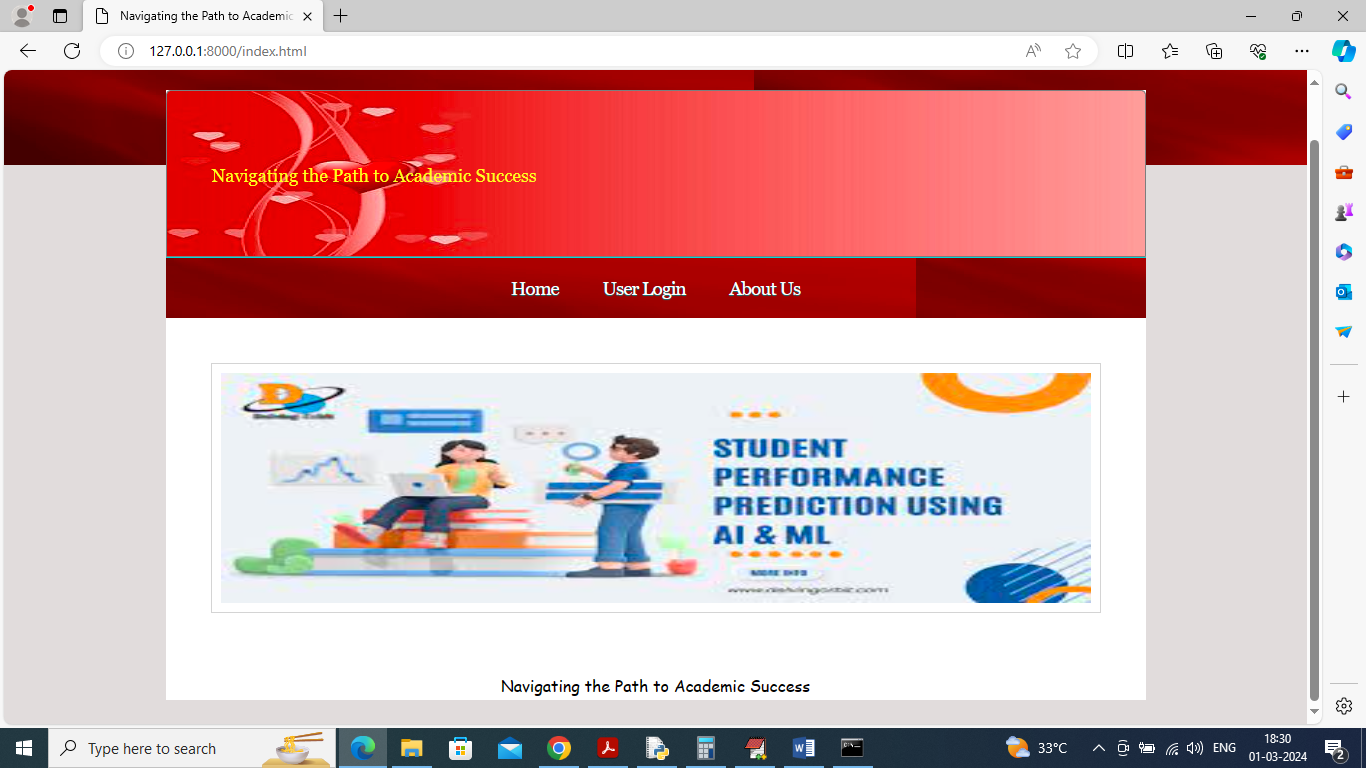


Fig 7.1:Home page of student performance prediction using ML

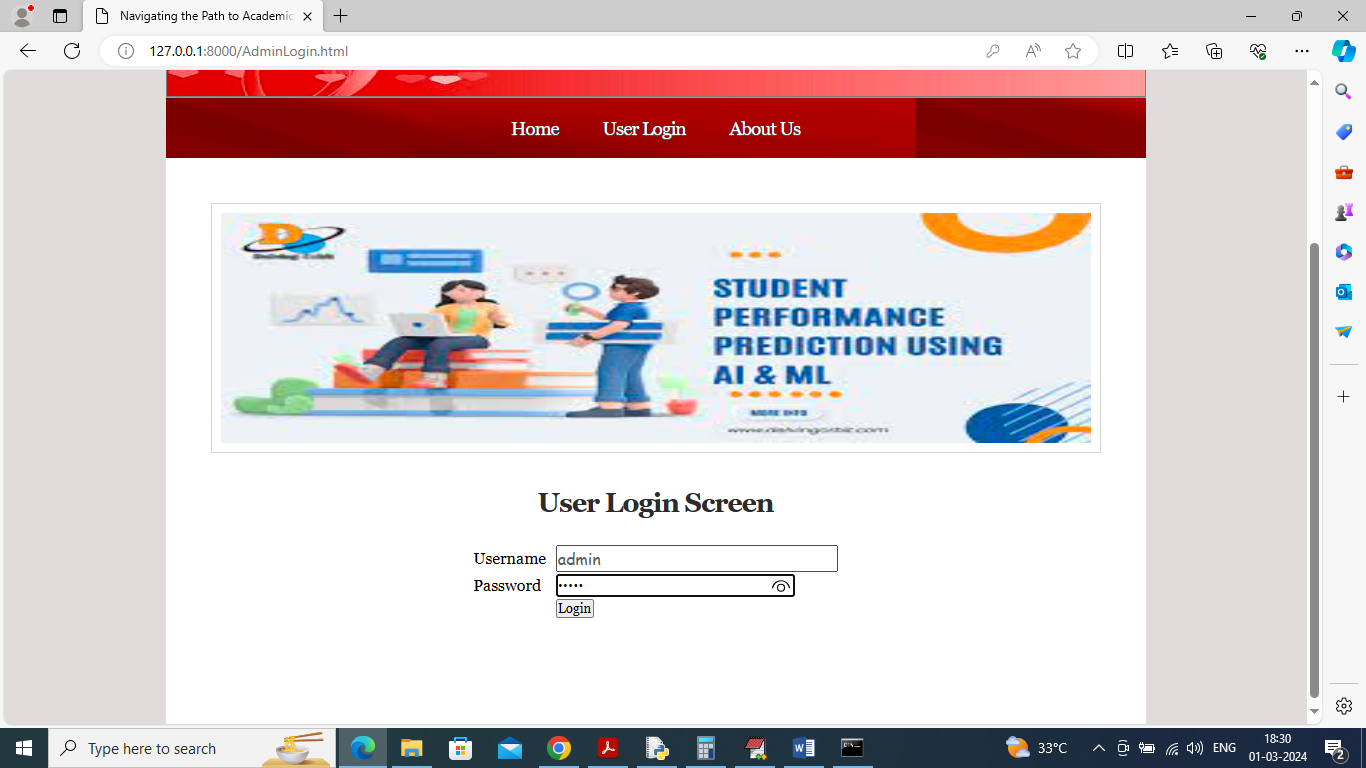


Fig7.2: In above screen user is login and after login will get below page

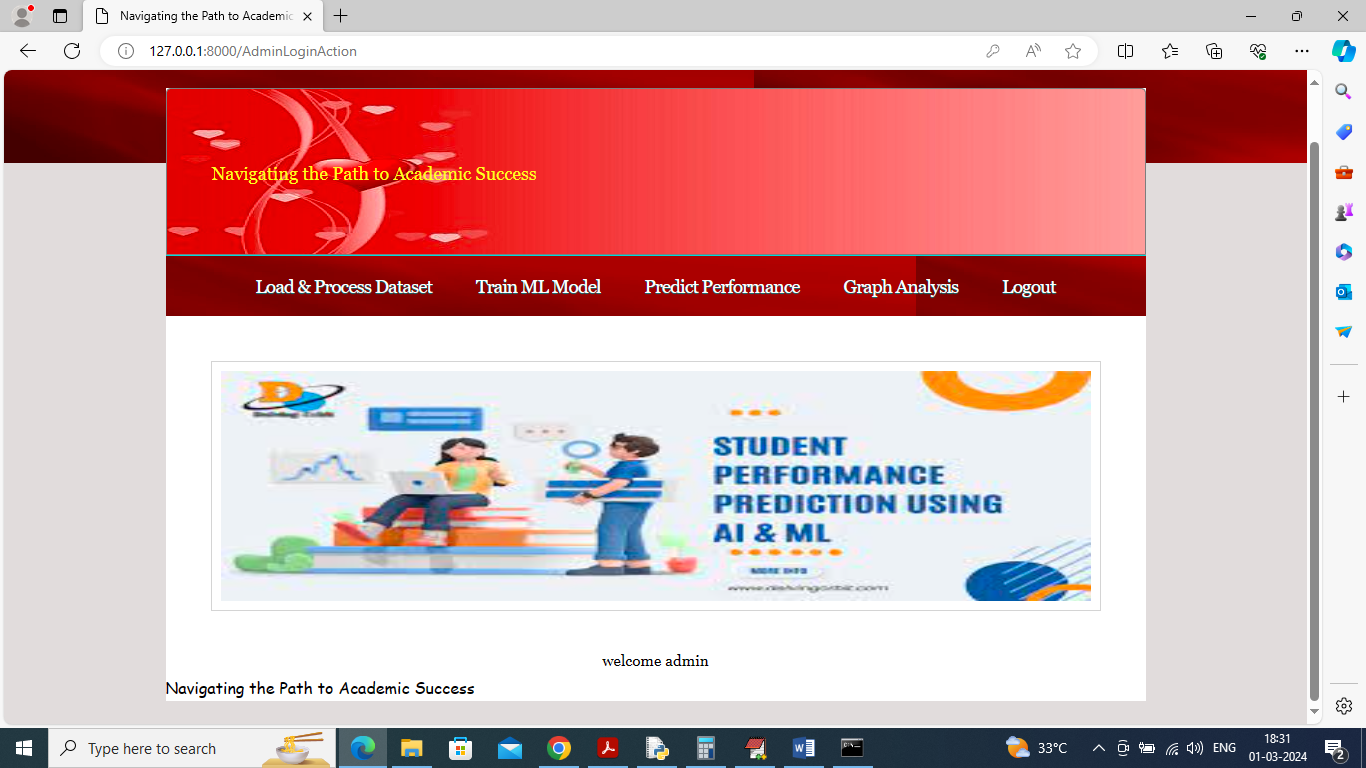


Fig 7.3: In above screen user can click on ‘Load & Process Dataset’ link to get below page

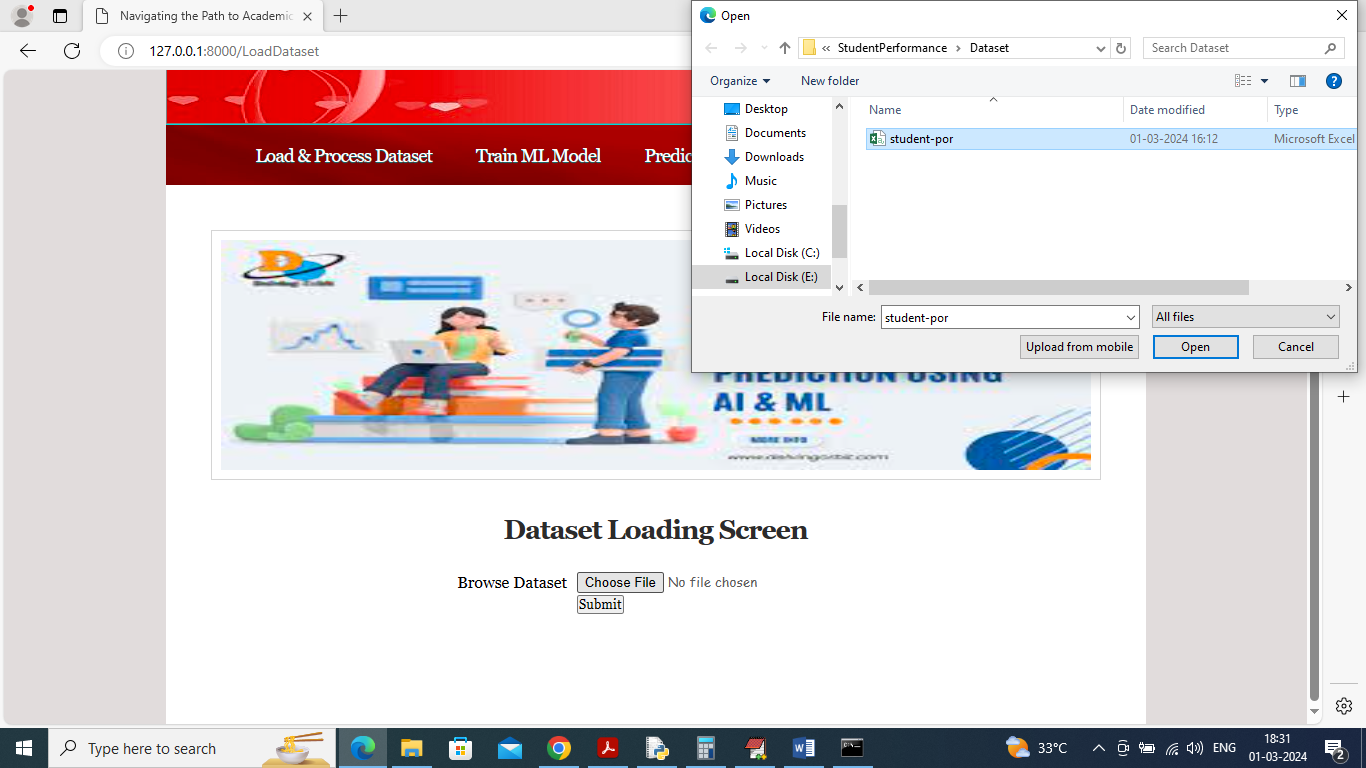


Fig 7.4: In above screen select and load dataset file and this dataset file available inside ‘Dataset’ folder and then click on ‘Open’ and ‘Submit’ button to get below page

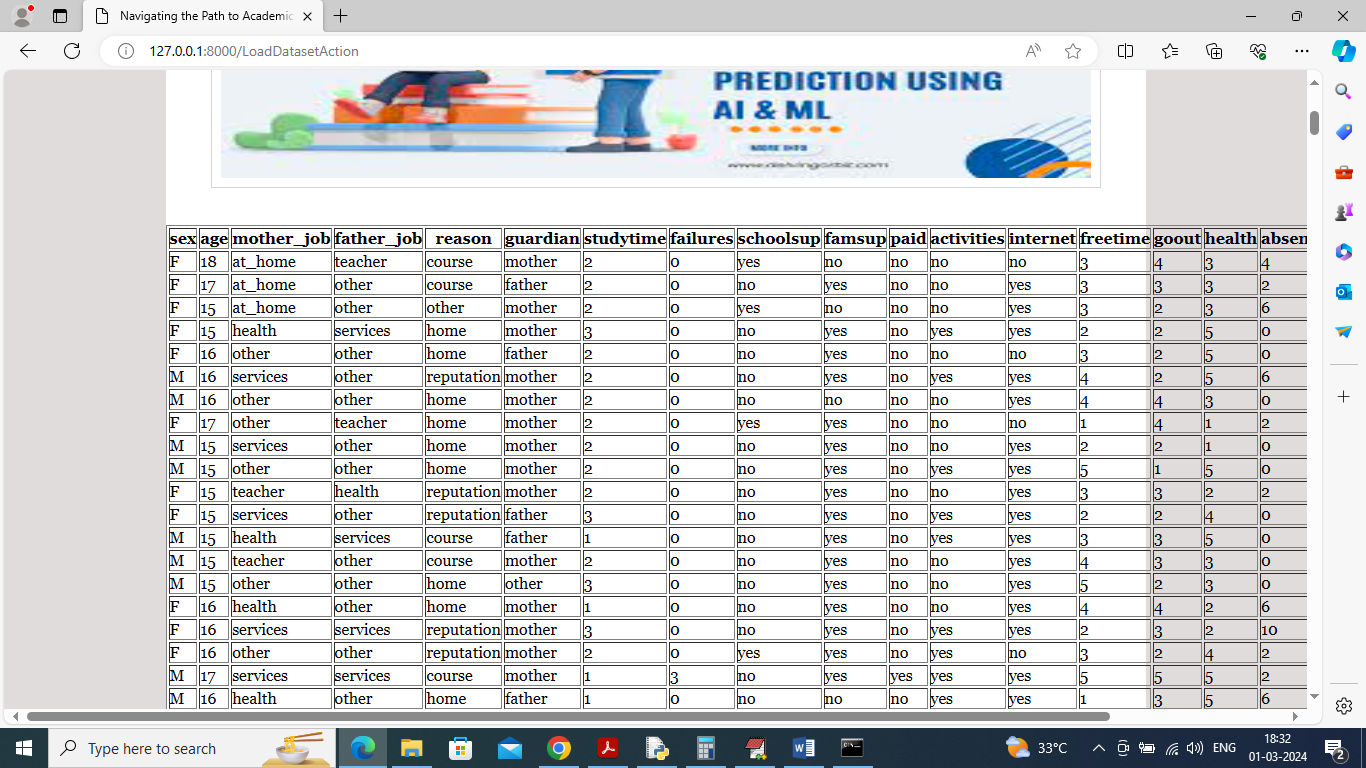


Fig7.5: In above screen dataset loaded and can see all columns and its values and now click on ‘Train ML Algorithm’ link to train all algorithms and get below page



Fig7.6: In above screen can see each algorithm performance in tabular format and in graph format. In graph x-axis represents algorithm names and y-axis represents accuracy and other metrics in different colour bars

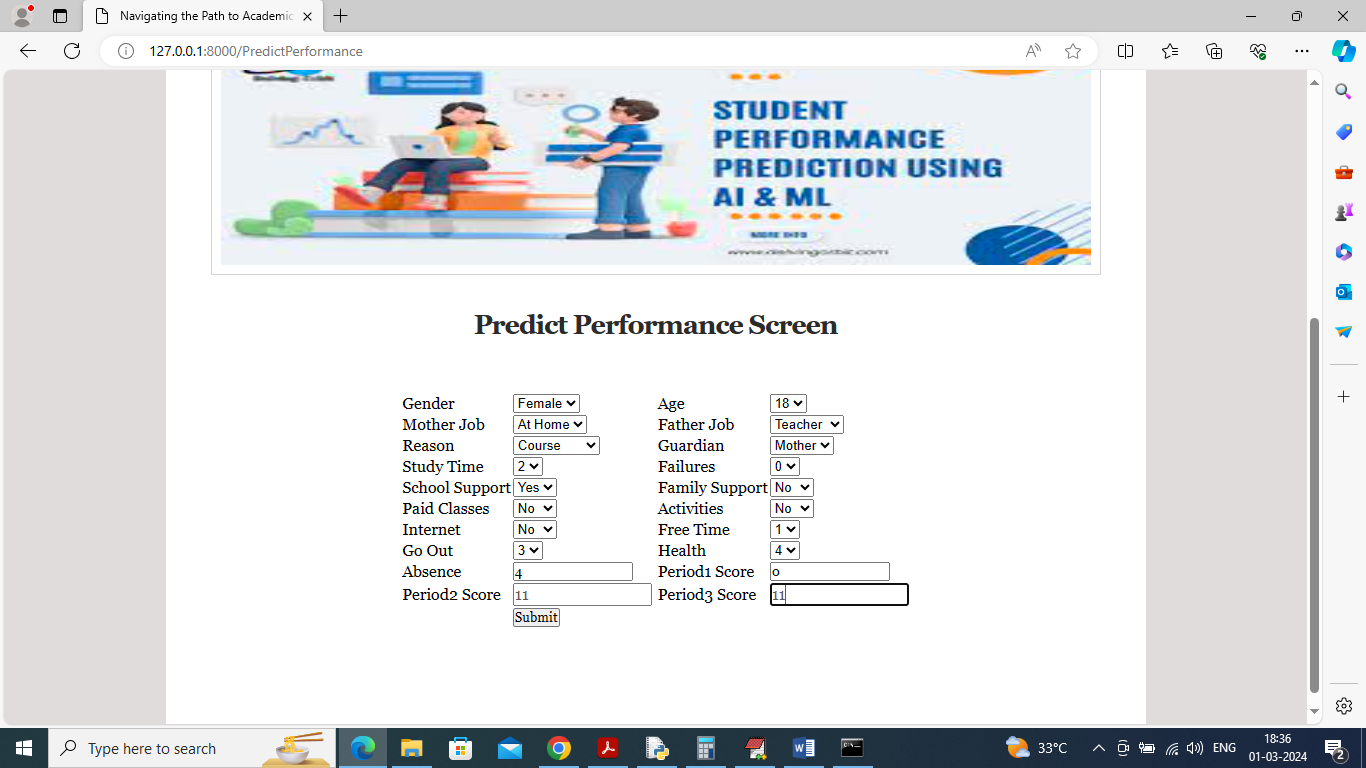


Fig7.7: In above screen user will enter and select academic details and then click on ‘Submit’ button to get below output

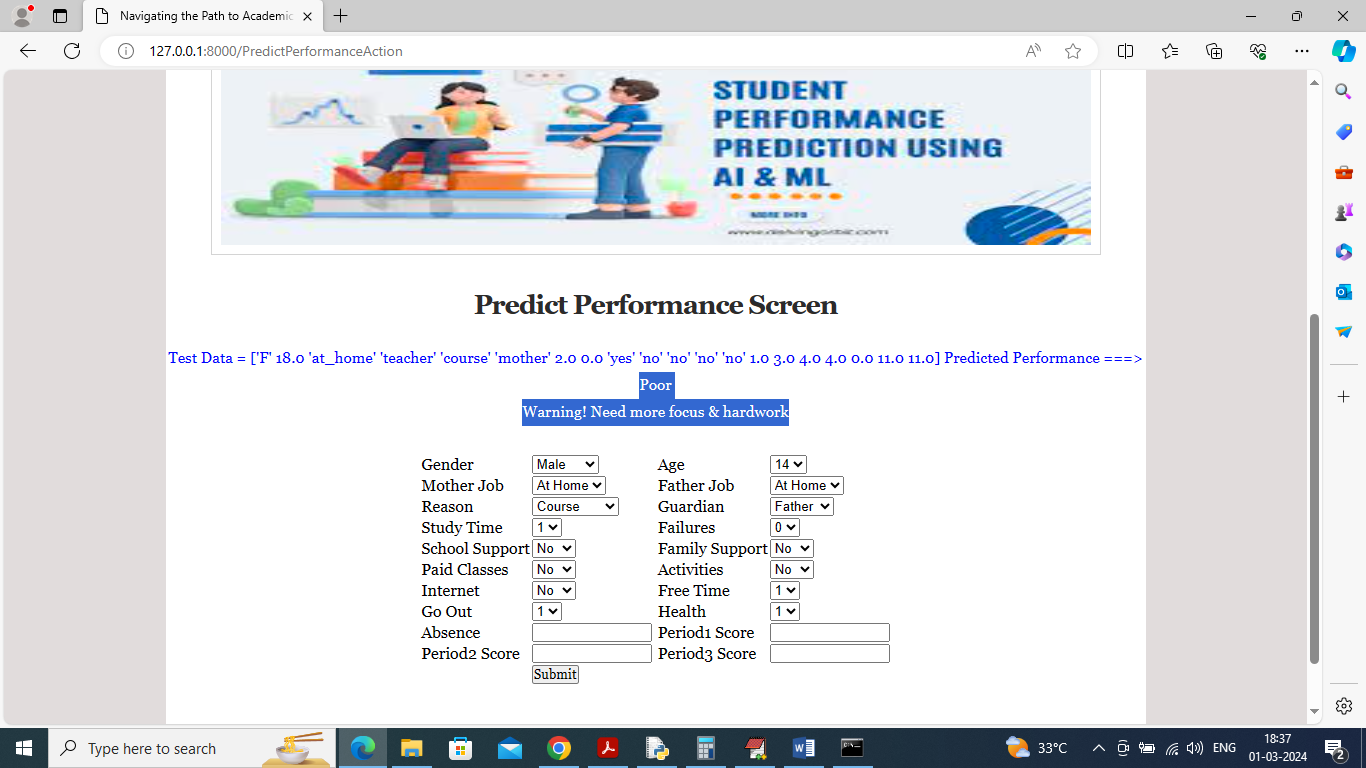


Fig7.8: In above screen in blue colour can see user academic data and then can predicted performance as ‘Poor’ with alert message to improve. Similarly you can input any details and get performance predicted. Below is another output

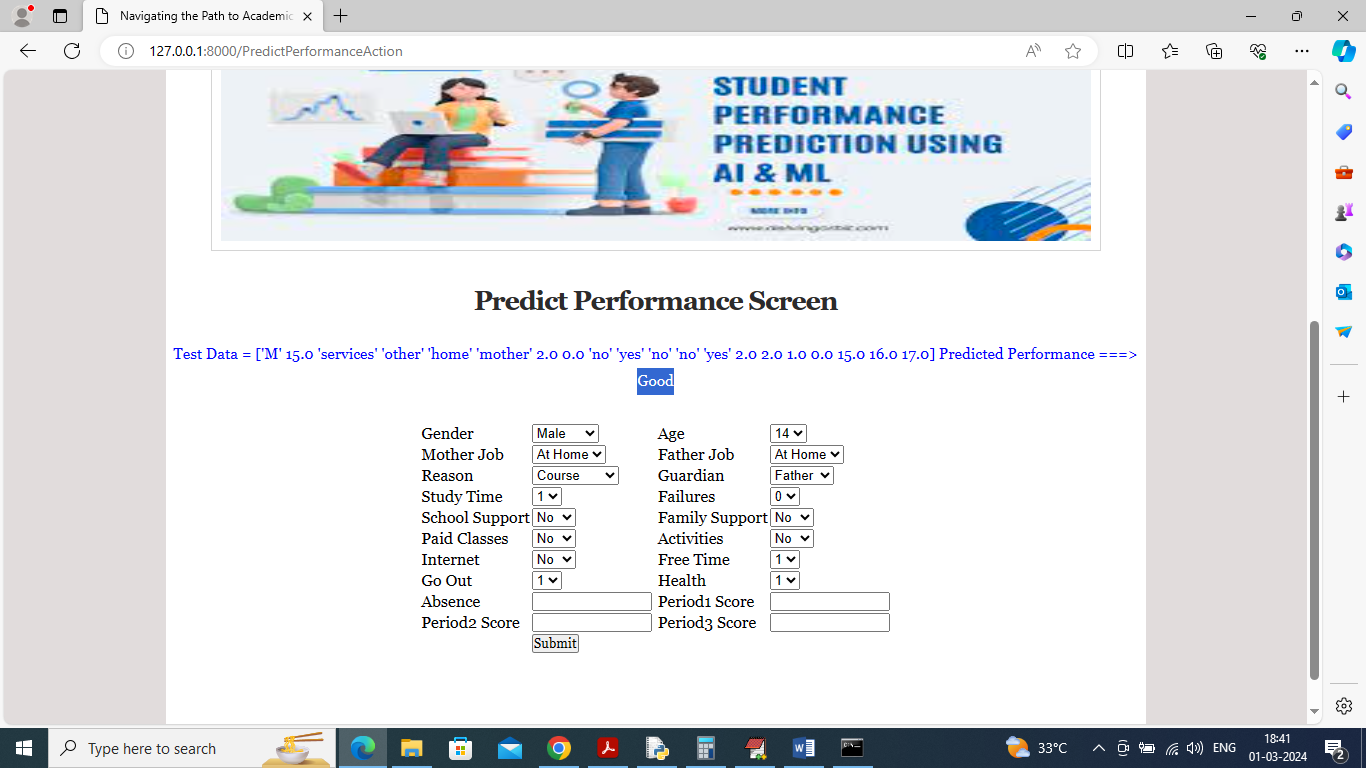


Fig7.9: In above screen predicted performance is ‘Good’.

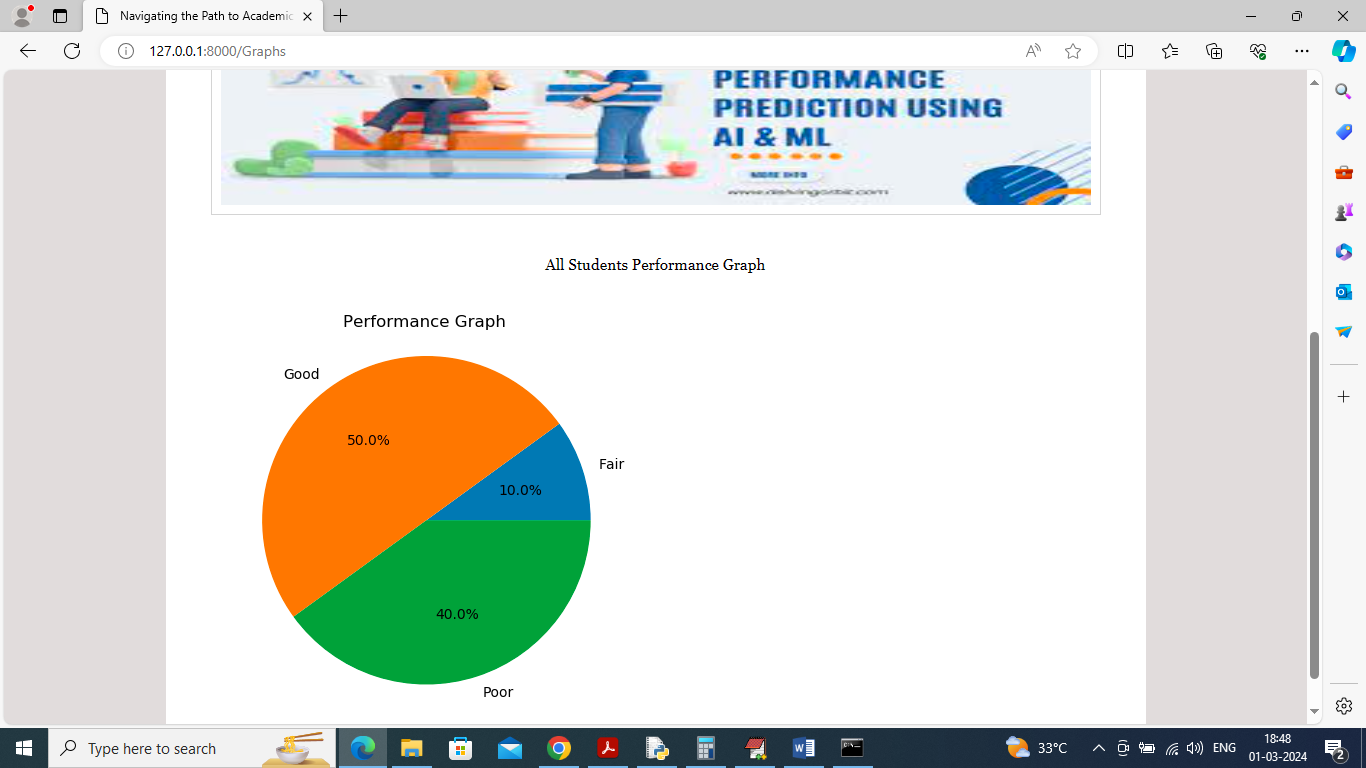


Fig7.10: In above pie chart graph can see overall performance of all students

**CHAPTER – 8**

#### CONCLUSION

A machine learning approach for tracking and predicting student performance in an undergraduate program leverages historical academic data, demographic factors, and behavioural patterns to identify trends and forecast outcomes. Techniques such as regression models, decision trees, support vector machines, and neural networks analyse features like attendance, grades, coursework**,** and engagement levels to predict future performance. Feature selection and dimensionality reduction optimize model efficiency, while ensemble learning enhances prediction accuracy. Regular updates with real-time data ensure adaptability, enabling early intervention strategies for at-risk students. This data-driven approach facilitates personalized learning, improves academicsupport systems, and enhances overall student success rates.

**CHAPTER - 9**

**FUTURE ENHANCEMENT**

Future enhancements in machine learning for tracking and predicting student performance in undergraduate programs will focus on advanced predictive models, personalized learning, and ethical AI. Deep learning techniques, such as transformers and hybrid models, will improve prediction accuracy, while reinforcement learning and recommender systems will enable adaptive learning experiences. Explainable AI (XAI) methods like SHAP and LIME will make predictions more transparent, allowing educators to understand and act on insights effectively. Integrating multimodal data sources, including sensor-based engagement tracking, NLP-based sentiment analysis, and real-time streaming analytics, will provide a comprehensive view of student progress. Additionally, federated learning and bias mitigation strategies will ensure privacy and fairness, reducing discrimination in academic predictions. Cloud and edge computing will enhance scalability, enabling real-time monitoring through AI-powered assistants, chatbots, and automated grading systems.

**CHAPTER - 10**

**BIBLIOGRAPHY**

**Books Referred :**

* Object Oriented Software Engineering: using UML, patterns and java, brooch
* The UML user guide – Grady Brooch
* Think Python-Allen Downey, Green Tea Press.
* Python Programming- W.Chun, Pearson

**Reference:**

**Journal Papers Referred:**

1. "Making college affordable", 2016, [online] Available: https://www.whitehouse.gov/issues/education/higher-education/making-college-affordable.

2. "Four-year myth: Making college more affordable", 2016, [online] Available: http://completecollege.org/wp-content/uploads/2014/11/4-Year-Myth.pdf.

3. H. Cen, K. Koedinger and B. Junker, "Learning factors analysis–A general method for cognitive model evaluation and improvement" in International Conference on Intelligent Tutoring Systems, New York, NY, USA:Springer, pp. 164-175, 2006.

4. M. Feng, N. Heffernan and K. Koedinger, "Addressing the assessment challenge with an online system that tutors as it assesses", User Model. User-Adapt. Interact., vol. 19, no. 3, pp. 243-266, 2009.

5. H.-F. Yu et al., "Feature engineering and classifier ensemble for KDD Cup 2010", Proc. KDD Cup 2010 Workshop, pp. 1-16, 2010.

6. Z. A. Pardos and N. T. Heffernan, "Using HMMs and bagged decision trees to leverage rich features of user and skill from an intelligent tutoring system dataset", J. Mach. Learn. Res. W CP, pp. 1-16, 2010.

7. Y. Meier, J. Xu, O. Atan and M. van der Schaar, "Personalized grade prediction: A data mining approach", Proc. 2015 IEEE Int. Conf. Data Mining, pp. 907-912, 2015.

8. C. G. Brinton and M. Chiang, "MOOC performance prediction via clickstream data and social learning networks", Proc. 2015 IEEE Conf. Comput. Commun. (INFOCOM), pp. 2299-2307, 2015.

9. "Educational data minding challenge", [online] Available: https://pslcdatashop.web.cmu.edu/KDDCup/.

10. Y. Jiang, R. S. Baker, L. Paquette, M. San Pedro and N. T. Heffernan, "Learning moment-by-moment and over the long term" in Proceedings of the International Conference on Artificial Intelligence in Education, New York, NY, USA:Springer, pp. 654-657, 2015.