|  |  |
| --- | --- |
| **INDEX** |  |
| **Chapter** | **Page No.** |
| **ABSTRACT** | vi |
| **LIST OF FIGURES** | vii |
| **1. SYSTEM ANALYSIS** | **1** |
| 1.1 Existing System | 1 |
| 1.1.1Disadvantages of Existing System | 1 |
| 1.2 Proposed System | 1 |
| 1.2.1 Advantages of Proposed System | 1 |
| 1.3 Introduction | 2 |
| **2. LITERATURE SURVEY** | **3** |
| **3.SYSTEM DESIGN** | **6** |
| 3.1 System Architecture | 6 |
| 3.1.1 Modules | 6 |
| 3.1.2 Block Diagram | 7 |
| 3.2 System Requirements | 8 |
| 3.2.1 Hardware Requirements | 8 |
| 3.2.2 Software Requirements | 8 |
| **4.INPUT & OUTPUT DESIGN** | **9** |
| 4.1 Input Design | 9 |
| 4.2 Output Design | 10 |
| **5. SOFTWARE ENVIRONMENT** | **11** |
| 5.1 Python Technology | 11 |
| 5.1.1 History of python | 11 |
| 5.1.2 What can python do | 12 |
| 5.1.3 Advantages of python | 13 |
| 5.1.4 Disadvantages of python | 14 |
| 5.2 Machine Learning | 15 |
| 5.2.1 Categories of Machine Learning | 15 |
| 5.2.2 Challenges of Machine Learning | 16 |
| 5.2.3 Applications of Machine Learning | 17 |

* 1. [Modules Used in Project 17](#_TOC_250012)
     1. [Tensor Flow 17](#_TOC_250011)
     2. [NumPy 17](#_TOC_250010)
     3. [Pandas 18](#_TOC_250009)
     4. [Matplotlib 18](#_TOC_250008)
     5. [Scikit – learn 19](#_TOC_250007)
  2. [System Implementation 19](#_TOC_250006)

1. SYSTEM STUDY 26
   1. Economic Feasibility 26
   2. Technical Feasibility 26
   3. Social Feasibility 27
2. SYSTEM TESTING 28
   1. [Types of Tests 28](#_TOC_250005)
      1. [Unit testing 28](#_TOC_250004)
      2. [Integration testing 29](#_TOC_250003)
      3. [Functional testing 29](#_TOC_250002)
      4. [System testing 30](#_TOC_250001)
      5. [Acceptance testing 31](#_TOC_250000)
3. RESULTS 33
4. CONCLUSION & FUTURE ENHANCEMENT 40
5. BIBLIOGRAPHY 41
6. **YUKTHI INNOVATION CERTIFICATE**

# ABSTRACT

Accurately predicting students’ future performance based on their ongoing academic records is crucial for effectively carrying out necessary pedagogical interventions to ensure students’ on-time and satisfactory graduation. Although there is a rich literature on predicting student performance when solving problems or studying for courses using data-driven approaches, predicting student performance in completing degrees (e.g. college programs) is much less studied and faces new challenges:

1. Students differ tremendously in terms of backgrounds and selected courses;
2. Courses are not equally informative for making accurate predictions;
3. Students’ evolving progress needs to be incorporated into the prediction.

In this paper, we develop a novel machine learning method for predicting student performance in degree programs that is able to address these key challenges. The proposed method has two major features.

First, a bi-layered structure comprising of multiple base predictors and a cascade of ensemble predictors is developed for making predictions based on students’ evolving performance states.

Second, a data-driven approach based on latent factor models and probabilistic matrix factorization is proposed to discover course relevance, which is important for constructing efficient base predictors.

Through extensive simulations on an undergraduate student dataset collected over three years at UCLA, we show that the proposed method achieves superior performance to benchmark approaches.

# LIST OF FIGURES

|  |  |  |
| --- | --- | --- |
| **Fig No.** | **Figure Name** | **Page No.** |
| 3.1 | System Architecture | 6 |
| 3.1.2 | Block Diagram | 7 |
| 4.1 | Input Design | 9 |
| 4.2 | Output Design | 10 |
| 8.1 | GUI for Tracking and Predicting Student | 33 |
|  | Performance in Degree Programs |  |
| 8.2 | Upload UCLA Students Dataset | 33 |
| 8.3 | UCLA Students Dataset Uploaded | 34 |
| 8.4 | Run Matrix Factorization | 34 |
| 8.5 | Run SVM Algorithm | 35 |
| 8.6 | Run Random Forest Algorithm | 36 |
| 8.7 | Run Logistic Regression Algorithm | 36 |
| 8.8 | Run EPP Algorithm | 37 |
| 8.9 | Upload the new-students-records | 37 |
| 8.10 | Predict Performance | 38 |
| 8.11 | Output | 39 |

## CHAPTER-1: SYSTEM ANALYSIS

#### Existing System:

* + - Recent studies show that only 50 of the more than 580 public four-year institutions in the United States have on-time graduation rates at or above 50 percent for their full- time students.
    - A critical step towards effective intervention is to build a system that can continuously keep track of students’ academic performance and accurately predict their future performance, such as when they are likely to graduate and their estimated final GPAs, given the current progress.

#### Disadvantages of Existing System:

* + - * However, predicting student performance within a degree program (e.g. college program) is significantly different and faces new challenges.
      * Students may take many courses but not all courses are equally informative for predicting students future performance.
      * Utilizing the student’s past performance in all courses that he/she has completed not only increases complexity but also introduces noise in the prediction, thereby degrading the prediction performance.

#### Proposed System:

* + - We consider a degree program in which students must complete a set of courses to graduate in T academic terms
    - Courses have prerequisite dependencies, namely a course can be taken only when certain prerequisite courses have been taken and passed.
    - We will focus on the prediction problem for one area in this department. Nevertheless, data from other areas will still be utilized for our prediction tasks.

#### Advantages of Proposed System:

* + - * It is important for constructing efficient base predictors.
      * System that can continuously keep track of students academic performance and accurately predict their future performance.
      * The results show that our proposed method is able to significantly outperform benchmark methods while preserving educational interpretability.

#### Introduction:

Affordable higher education is crucial for a nation's economic prosperity and is a key focus of government education policies. However, student loan debt in the United States has exceeded one trillion dollars, surpassing both credit card and auto loan debts, due to the rising costs of college education and prolonged graduation times. To address this issue, it is essential to help more students graduate on time through early interventions based on academic performance tracking and predictive analysis. While previous research has focused on predicting student performance in various educational contexts, predicting performance within a degree program poses new challenges, including handling diverse student backgrounds and course choices.

Predicting student performance is an ongoing process that requires continuous tracking and updating, considering both recent accomplishments and the evolution of student progress. The proposed method in this paper primarily focuses on predicting students' GPAs, but its framework can be adapted for other student performance prediction tasks.

###### Our main contributions are three-fold:

1. We introduce a novel predictive algorithm that considers students' evolving performance states, using a two-layer structure with base and ensemble predictors. This approach allows us to make accurate predictions about students' future performance while managing complexity.
2. We develop a data-driven course clustering method that automatically identifies relevant course clusters based on student grade data. This helps streamline the training process for our predictors by focusing on relevant courses, reducing complexity, and removing irrelevant information and noise.
3. We conduct extensive simulation studies on a dataset from the Mechanical and Aerospace Engineering department at UCLA, spanning three years and 1,169 students. Our results show that our method outperforms benchmark methods while maintaining educational interpretability.

## CHAPTER-2: LITERATURE SURVEY

**H. Cen, K. Koedinger, and B. Junker [1]** states that “A cognitive model is a set of production rules or skills encoded in intelligent tutors to model how students solve problems. It is usually generated by brainstorming and iterative refinement between subject experts, cognitive scientists and programmers. In this paper we propose a semi-automated method for improving a cognitive model called Learning Factors Analysis that combines a statistical model, human expertise and a combinatorial search. We use this method to evaluate an existing cognitive model and to generate and evaluate alternative models. We present improved cognitive models and make suggestions for improving the intelligent tutor based on those models.”

**M. Feng, N. Heffernan, and K. Koedinger [2]** states that “Secondary teachers across the United States are being asked to use formative assessment data (Black & Wiliam, 1998a, 1998b; Roediger & Karpicke, 2006) to inform their classroom instruction. At the same time, critics of US government's No Child Left Behind legislation are calling the bill "No Child Left Untested". Among other things, critics point out that every hour spent assessing students is an hour lost from instruction. But, does it have to be? What if we better integrated assessment into classroom instruction and allowed students to learn during the test? We developed an approach that provides immediate tutoring on practice assessment items that students cannot solve on their own. Our hypothesis is that we can achieve more accurate assessment by not only using data on whether students get test items right or wrong, but by also using data on the effort required for students to solve a test item with instructional assistance. We have integrated assistance and assessment in the ASSISTment system. The system helps teachers make better use of their time by offering instruction to students while providing a more detailed evaluation of student abilities to the teachers, which is impossible under current approaches. Our approach for assessing student math proficiency is to use data that our system collects through its interactions with students to estimate their performance on an end-of-year high stakes state test. Our results show that we can do a reliably better job predicting student end-of-year exam scores by leveraging the interaction data, and the model based on only the interaction information makes better predictions than the traditional assessment model that uses only information about correctness on the test items.”

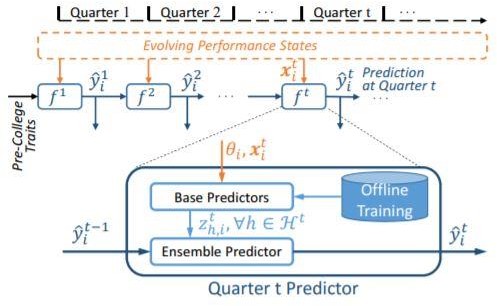
**Y. Meier, J. Xu, O. Atan, and M. van der Schaar [3]** states that “To increase efﬁcacy in traditional classroom courses as well as in Massive Open Online Courses (MOOCs), automated systems supporting the instructor are needed. One important problem is to automatically detect students that are going to do poorly in a course early enough to be able to take remedial actions. This paper proposes an algorithm that predicts the ﬁnal grade of each student in a class. It issues a prediction for each student individually, when the expected accuracy of the prediction is sufﬁcient. The algorithm learns online what is the optimal prediction and time to issue a prediction based on past history of students’ performance in a course. We derive demonstrate the performance of our algorithm on a dataset obtained based on the performance of approximately 700 undergraduate students who have taken an introductory digital signal processing over the past 7 years. Using data obtained from a pilot course, our methodology suggests that it is effective to perform early in-class assessments such as quizzes, which result in timely performance prediction for each student, thereby enabling timely interventions by the instructor (at the student or class level) when necessary.”

**C. G. Brinton and M. Chiang[4]** states that “We study student performance prediction in Massive Open Online Courses (MOOCs), where the objective is to predict whether a user will be Correct on First Attempt (CFA) in answering a question. In doing so, we develop novel techniques that leverage behavioral data collected by MOOC platforms. Using video- watching clickstream data from one of our MOOCs, we first extract summary quantities (e.g., fraction played, number of pauses) for each user-video pair, and show how certain intervals/sets of values for these behaviors quantify that a pair is more likely to be CFA or not for the corresponding question. Motivated by these findings, our methods are designed to determine suitable intervals from training data and to use the corresponding success estimates as learning features in prediction algorithms. Tested against a large set of empirical data, we find that our schemes outperform standard algorithms (i.e., without behavioral data) for all datasets and metrics tested. Moreover, the improvement is particularly pronounced when considering the first few course weeks, demonstrating the “early detection” capability of such clickstream data. We also discuss how CFA prediction can be used to depict graphs of the Social Learning Network (SLN) of students, which can help instructors manage courses more effectively.”

**Y.-h. Wang and H.-C. Liao[5]** states that “This study proposes an Adaptive Learning in Teaching English as a Second Language (TESL) for e-learning system (AL-TESL-e-learning system) that considers various student characteristics. This study explores the learning performance of various students using a data mining technique, an artificial neural network (ANN), as the core of AL-TESL-e-learning system. Three different levels of teaching content for vocabulary, grammar, and reading were set for adaptive learning in the AL-TESL-e- learning system. Finally, this study explores the feasibility of the proposed AL-TESL-e- learning system by comparing the results of the regular online course control group with the AL-TESL-e-learning system adaptive learning experiment group. Statistical results show that the experiment group had better learning performance than the control group; that is, the AL- TESL-e-learning system was better than a regular online course in improving student learning performance.”

## CHAPTER-3: SYSTEM DESIGN

#### System Architecture:



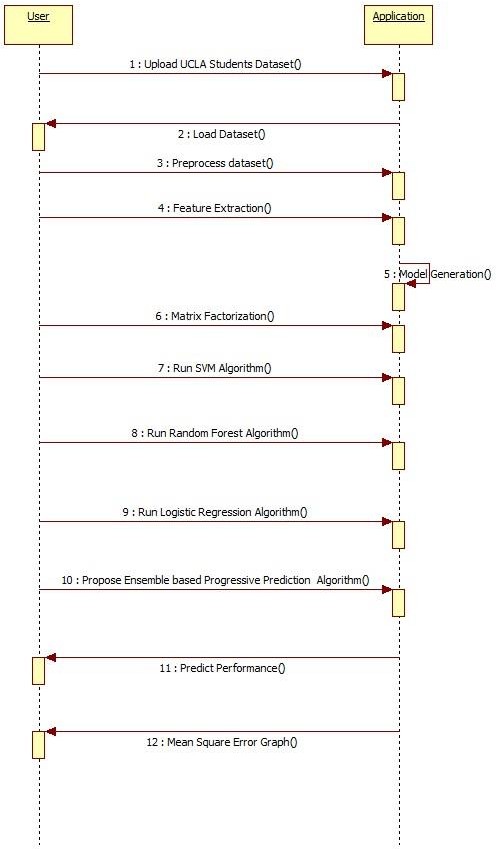
**Architecture Diagram of Proposed System**

#### Modules:

* + - * Upload UCLA Students Dataset
      * Preprocess Dataset
      * Feature Extraction
      * Model Generation
      * Matrix Factorization
      * Run SVM Algorithm
      * Run Random Forest Algorithm
      * Run Logistic Regression Algorithm
      * Propose Ensemble based Progressive Prediction (EPP) Algorithm
      * Predict Performance
      * Mean Square Error Graph

#### Block Diagram:

A sequence diagram represents the interaction between different objects in the system. The important aspect of a sequence diagram is that it is time-ordered. This means that the exact sequence of the interactions between the objects is represented step by step. Different objects in the sequence diagram interact with each other by passing "messages".



#### System Requirements:

###### Hardware Requirements:

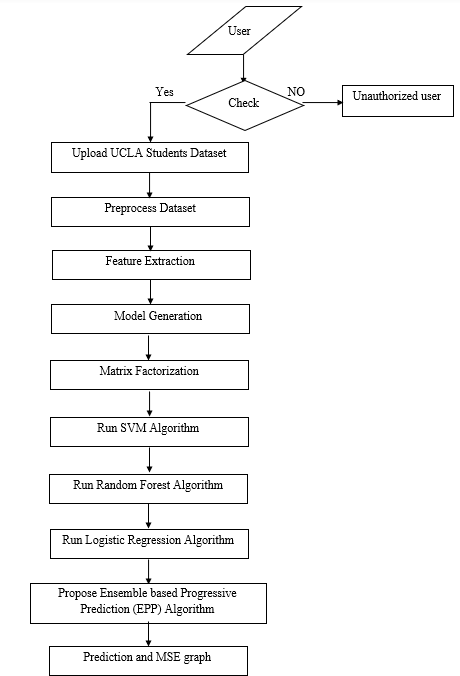
* + - * Mouse : Optical Mouse.
      * Processor : Minimum Intel i3.
      * Ram : Minimum 4 GB.
      * Hard disk : Minimum 250GB.

###### Software Requirements:

* + - * Operating system : Windows, Linux.
      * Coding Language : Python IDLE 3.7
      * Library : Text Blob.

## CHAPTER-4: INPUT AND OUTPUT DESIGN

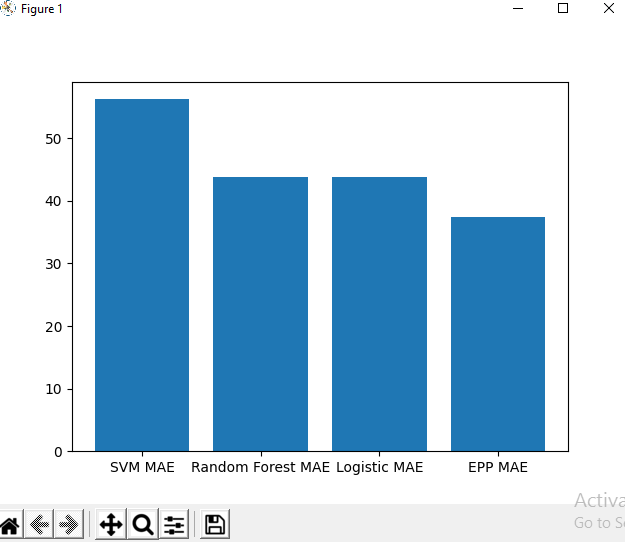
#### Input Design:



* + - 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 depicts information flow and the transformations that are applied as data moves from input to output.
    - 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.

#### Output Design:

* + - MSE graph gives the below results:



* + - In above graph, x-axis represents algorithm name and y-axis represents MSE (MEAN SQUARE ERROR).
    - From above graph, we can see propose algorithm got less MSE error and has high accuracy compare to other algorithms. From above graph we can conclude that propose EPP (PROPOSE ENSEMBLE-BASED PROGRESSIVE PREDICTION) is better in prediction compare to other algorithms.

## CHAPTER-5: SYSTEM ENVIRONMENT

#### Python Technology:

* + - Python is an interpreted high-level programming language for general-purpose programming. Created by Guido van Rossum and first released in 1991, Python has a design philosophy that emphasizes code readability, notably using significant whitespace.
    - Python features a dynamic type system and automatic memory management. It supports multiple programming paradigms, including object-oriented, imperative, functional and procedural, and has a large and comprehensive standard library.
    - Python is Interpreted − Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.
    - Python is Interactive − you can actually sit at a Python prompt and interact with the interpreter directly to write your programs.
    - Python also acknowledges that speed of development is important. Readable and terse code is part of this, and so is access to powerful constructs that avoid tedious repetition of code. Maintainability also ties into this may be an all but useless metric, but it does say something about how much code you have to scan, read and/or understand to troubleshoot problems or tweak behaviors.
    - This speed of development, the ease with which a programmer of other languages can pick up basic Python skills and the huge standard library is key to another area where Python excels. All its tools have been quick to implement, saved a lot of time, and several of them have later been patched and updated by people with no Python background - without breaking.

#### History of Python:

What do the alphabet and the programming language Python have in common? Right, both start with ABC. If we are talking about ABC in the Python context, it's clear that the programming language ABC is meant. ABC is a general-purpose programming language and programming environment, which had been developed in the Netherlands, Amsterdam, at the CWI (Centrum Wiskunde &Informatica). The greatest achievement of ABC was to influence the design of Python. Python was conceptualized in the late 1980s. Guido van Rossum worked that time in a project at the CWI, called Amoeba, a distributed operating system. In

an interview with Bill Venners, Guido van Rossum said: "In the early 1980s, I worked as an implementer on a team building a language called ABC at Centrum voor Wiskunde en Informatica (CWI). I don't know how well people know ABC's influence on Python. I try to mention ABC's influence because I'm indebted to everything I learned during that project and to the people who worked on it". Later on in the same Interview, Guido van Rossum continued: "I remembered all my experience and some of my frustration with ABC. I decided to try to design a simple scripting language that possessed some of ABC's better properties, but without its problems. So I started typing. I created a simple virtual machine, a simple parser, and a simple runtime. I made my own version of the various ABC parts that I liked. I created a basic syntax, used indentation for statement grouping instead of curly braces or begin-end blocks, and developed a small number of powerful data types: a hash table (or dictionary, as we call it), a list, strings, and numbers."

#### What can python do:

* + - * Python downloads with an extensive library and it contain code for various purposes like regular expressions, documentation-generation, unit-testing, web browsers, threading, databases, CGI, email, image manipulation, and more. So, we don’t have to write the complete code for that manually.
      * As we have seen earlier, Python can be extended to other languages. You can write some of your code in languages like C++ or C. This comes in handy, especially in projects.
      * Complimentary to extensibility, Python is embeddable as well. You can put your Python code in your source code of a different language, like C++. This lets us add scripting capabilities to our code in the other language.
      * The language’s simplicity and extensive libraries render programmers more productive than languages like Java and C++ do. Also, the fact that you need to write less and get more things done.
      * Since Python forms the basis of new platforms like Raspberry Pi, it finds the future bright for the Internet of Things. This is a way to connect the language with the real world.
      * Because it is not such a verbose language, reading Python is much like reading English. This is the reason why it is so easy to learn, understand, and code. It also

does not need curly braces to define blocks, and indentation is mandatory. This further aids the readability of the code.

* + - * This language supports both the procedural and object-oriented programming paradigms. While functions help us with code reusability, classes and objects let us model the real world. A class allows the encapsulation of data and functions into one.
      * Python is freely available. Not only can you download Python for free, but you can also download its source code, make changes to it, and even distribute it. It downloads with an extensive collection of libraries to help you with your tasks.
      * When you code your project in a language like C++, you may need to make some changes to it if you want to run it on another platform. But it isn’t the same with Python. Here, you need to code only once, and you can run it anywhere. This is called Write Once Run Anywhere (WORA). However, you need to be careful enough not to include any system-dependent features.

#### Advantages of Python:

* + - * **Extensive Libraries:** Python downloads with an extensive library and it contain code for various purposes like regular expressions, documentation-generation, unit-testing, web browsers, threading, databases, CGI, email, image manipulation, and more. So, we don’t have to write the complete code for that manually.
      * **Extensible:** As we have seen earlier, Python can be extended to other languages. You can write some of your code in languages like C++ or C. This comes in handy, especially in projects.
      * **Embeddable:** Complimentary to extensibility, Python is embeddable as well. You can put your Python code in your source code of a different language, like C++. This lets us add scripting capabilities to our code in the other language.
      * **Improved Productivity:** The language’s simplicity and extensive libraries render programmers more productive than languages like Java and C++ do. Also, the fact that you need to write less and get more things done.
      * **IOT Opportunities:** Since Python forms the basis of new platforms like Raspberry Pi, it finds the future bright for the Internet Of Things. This is a way to connect the language with the real world.
      * **Simple and Easy:** When working with Java, you may have to create a class to print ‘Hello World’. But in Python, just a print statement will do. It is also quite easy

to learn, understand, and code. This is why when people pick up Python, they have a hard time adjusting to other more verbose languages like Java.

* + - * **Readable:** Because it is not such a verbose language, reading Python is much like reading English. This is the reason why it is so easy to learn, understand, and code. It also does not need curly braces to define blocks, and indentation is mandatory. This further aids the readability of the code.
      * **Object-Oriented:** This language supports both the procedural and object- oriented programming paradigms. While functions help us with code reusability, classes and objects let us model the real world. A class allows the encapsulation of data and functions into one.
      * **Free and Open-Source:** Like we said earlier, Python is freely available. But not only can you download Python for free, but you can also download its source code, make changes to it, and even distribute it. It downloads with an extensive collection of libraries to help you with your tasks.
      * **Portable:** When you code your project in a language like C++, you may need to make some changes to it if you want to run it on another platform. But it isn’t the same with Python. Here, you need to code only once, and you can run it anywhere. This is called Write Once Run Anywhere (WORA).
      * **Interpreted:** Lastly, we will say that it is an interpreted language. Since statements are executed one by one, debugging is easier than in compiled languages.

#### Disadvantages of Python:

* + - * **Speed Limitations:** We have seen that Python code is executed line by line. But since Python is interpreted, it often results in slow execution. This, however, isn’t a problem unless speed is a focal point for the project. In other words, unless high speed is a requirement, the benefits offered by Python are enough to distract us from its speed limitations.
      * **Weak in Mobile Computing and Browsers:** While it serves as an excellent server- side language, Python is much rarely seen on the client-side. Besides that, it is rarely ever used to implement smartphone-based applications. One such application is called Carbon Nelle.
      * **Design Restrictions:** As you know, Python is dynamically-typed. This means that you don’t need to declare the type of variable while writing the code. It uses duck-

typing which means that if it looks like a duck, it must be a duck. While this is easy on the programmers during coding, it can raise run-time errors.

* + - * **Underdeveloped Database Access Layers:** Compared to more widely used technologies like JDBC (Java Data Base Connectivity) and ODBC (Open Data Base Connectivity), Python’s database access layers are a bit underdeveloped. Consequently, it is less often applied in huge enterprises.

#### What is Machine Learning:

* + - Before we take a look at the details of various machine learning methods, let's start by looking at what machine learning is, and what it isn't. Machine learning is often categorized as a subfield of artificial intelligence, but I find that categorization can often be misleading at first brush. The study of machine learning certainly arose from research in this context, but in the data science application of machine learning methods, it's more helpful to think of machine learning as a means of building models of data.
    - Fundamentally, machine learning involves building mathematical models to help understand data. "Learning" enters the fray when we give these models tunable parameters that can be adapted to observed data; in this way the program can be considered to be "learning" from the data.
    - Once these models have been fit to previously seen data, they can be used to predict and understand aspects of newly observed data. I'll leave to the reader the more philosophical digression regarding the extent to which this type of mathematical, model-based "learning" is similar to the "learning" exhibited by the human brain. Understanding the problem setting in machine learning is essential to using these tools effectively, and so we will start with some broad categorizations of the types of approaches we'll discuss here.

#### Categories of Machine Leaning:

At the most fundamental level, machine learning can be categorized into two main types: supervised learning and unsupervised learning:

* + - * Supervised learning involves somehow modeling the relationship between measured features of data and some label associated with the data; once this model is determined, it can be used to apply labels to new, unknown data. This is further subdivided into classification tasks and regression tasks: in classification, the labels

are discrete categories, while in regression, the labels are continuous quantities. We will see examples of both types of supervised learning in the following section.

* + - * Unsupervised learning involves modeling the features of a dataset without reference to any label and is often described as "letting the dataset speak for itself." These models include tasks such as clustering and dimensionality reduction. Clustering algorithms identify distinct groups of data, while dimensionality reduction algorithms search for more succinct representations of the data. We will see examples of both types of unsupervised learning in the following section.

#### Challenges in Machines Learning:

While Machine Learning is rapidly evolving, making significant strides with cyber security and autonomous cars, this segment of AI as whole still has a long way to go. The reason behind is that ML has not been able to overcome number of challenges. The challenges that ML is facing currently are −

* + - * Quality of data − Having good-quality data for ML algorithms is one of the biggest challenges. Use of low-quality data leads to the problems related to data preprocessing and feature extraction.
      * Time-Consuming task − Another challenge faced by ML models is the consumption of time especially for data acquisition, feature extraction and retrieval.
      * Lack of specialist persons − As ML technology is still in its infancy stage, availability of expert resources is a tough job.
      * No clear objective for formulating business problems − Having no clear objective and well-defined goal for business problems is another key challenge for ML because this technology is not that mature yet.
      * Issue of overfitting & underfitting − If the model is overfitting or underfitting, it cannot be represented well for the problem.
      * Curse of dimensionality − Another challenge ML model faces is too many features of data points. This can be a real hindrance.
      * Difficulty in deployment − Complexity of the ML model makes it quite difficult to be deployed in real life.

#### Applications of Machines Learning:

Machine Learning is the most rapidly growing technology and according to researchers we are in the golden year of AI and ML. It is used to solve many real-world complex problems which cannot be solved with traditional approach. Following are some real-world applications of ML −

* + - * Emotion analysis
      * Sentiment analysis
      * Error detection and prevention
      * Weather forecasting and prediction
      * Stock market analysis and forecasting
      * Speech synthesis
      * Speech recognition
      * Customer segmentation
      * Object recognition
      * Fraud detection
      * Fraud prevention
      * Recommendation of products to customer in online shopping

#### Modules Used in Project:

###### Tensor Flow:

* + - * Tensor Flow is a free and open-source software library for dataflow and differentiable programming across a range of tasks. It is a symbolic math library and is also used for machine learning applications such as neural networks. It is used for both research and production at Google.
      * Tensor Flow was developed by the Google Brain team for internal Google use. It was released under the Apache 2.0 open-source license on November 9, 2015.

###### NumPy:

* + - * NumPy is a general-purpose array-processing package. It provides a high- performance multidimensional array object, and tools for working with these arrays.
      * It is the fundamental package for scientific computing with Python. It contains various features including these important ones:
        + A powerful N-dimensional array object.
        + Sophisticated (broadcasting) functions.
        + Tools for integrating C/C++ and Fortran code.
        + Useful linear algebra, Fourier transform, and random number capabilities.
      * Besides its obvious scientific uses, NumPy can also be used as an efficient multi- dimensional container of generic data. Arbitrary data types can be defined using NumPy which allows NumPy to seamlessly and speedily integrate with a wide variety of databases.

###### Pandas:

* + - * Pandas is an open-source Python Library providing high-performance data manipulation and analysis tool using its powerful data structures.
      * Python was majorly used for data munging and preparation. It had very little contribution towards data analysis. Pandas solved this problem.
      * Using Pandas, we can accomplish five typical steps in the processing and analysis of data, regardless of the origin of data load, prepare, manipulate, model, and analyze.
      * Python with Pandas is used in a wide range of fields including academic and commercial domains including finance, economics, Statistics, analytics, etc.

###### Matplotlib:

* + - * Matplotlib is a Python 2D plotting library which produces publication quality figures in a variety of hardcopy formats and interactive environments across platforms.
      * Matplotlib can be used in Python scripts, the Python and I Python shells, the Jupyter Notebook, web application servers, and four graphical user interface toolkits.
      * Matplotlib tries to make easy things easy and hard things possible. You can generate plots, histograms, power spectra, bar charts, error charts, scatter plots, etc., with just a few lines of code.
      * For simple plotting the pyplot module provides a MATLAB-like interface, particularly when combined with I Python. For the power user, you have full control of line styles, font properties, axes properties, etc, via an object oriented interface or via a set of functions familiar to MATLAB users.

###### Scikit – learn:

* + - * Scikit-learn offers a wide array of supervised and unsupervised machine learning algorithms with a consistent Python interface, making it accessible for both academic and commercial applications. It is released under a permissive simplified BSD license.
      * The library is designed for data analysis, classification, regression, clustering, dimensionality reduction, and more, making it a versatile tool for a variety of machine learning tasks.
      * Scikit-learn provides easy-to-use tools for data preprocessing, feature selection, model evaluation, and hyperparameter tuning, streamlining the machine learning workflow.
      * With a strong user community and extensive documentation, Scikit-learn is a popular choice for machine learning practitioners and researchers, offering a robust foundation for building predictive models in Python.

#### System Implementation:

###### Step 1: Import Libraries

from tkinter import messagebox from tkinter import \*

from tkinter.filedialog import askopenfilename from tkinter import simpledialog

import tkinter import numpy as np

from tkinter import filedialog import pandas as pd

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

from sklearn.metrics import accuracy\_score

from sklearn.ensemble import RandomForestClassifier import matplotlib.pyplot as plt

from sklearn.linear\_model import LogisticRegression from sklearn.metrics import mean\_squared\_error

from sklearn.preprocessing import StandardScaler from sklearn.ensemble import BaggingClassifier from sklearn.tree import DecisionTreeClassifier

This code imports necessary libraries for a Python program that involves creating a graphical user interface (GUI) with Tkinter and performing machine learning tasks using scikit-learn, including data manipulation with pandas, model training and evaluation, and plotting with matplotlib.

###### Step 2: Initialization and Global Variable Setup

main = tkinter.Tk()

main.title("A Machine Learning Approach for Tracking and Predicting Student Performance in Degree Programs") main.geometry("1300x1200")

global filename

global svm\_mae,random\_mae,logistic\_mae,epp\_mae global matrix\_factor

global X, Y, X\_train, X\_test, y\_train, y\_test global epp

global classifier

This step initializes a Tkinter GUI window titled "A Machine Learning Approach for Tracking and Predicting Student Performance in Degree Programs" with predefined dimensions. It also defines several global variables for use within the program.

###### Step 3: Data Upload and Preprocessing

defupload():

global filename global matrix\_factor

filename = filedialog.askopenfilename(initialdir = "dataset")

pathlabel.config(text=filename) matrix\_factor = pd.read\_csv(filename) text.delete('1.0', END) text.insert(END,'UCLA dataset loaded\n') text.insert(END,"Dataset Size :

"+str(len(matrix\_factor))+"\n")

defsplitdataset(matrix\_factor):

X = matrix\_factor.values[:, 0:12] Y = matrix\_factor.values[:, 12] print(X)

print(Y)

X\_train, X\_test, y\_train, y\_test = train\_test\_split( X, Y, test\_size = 0.2, random\_state = 0)

return X, Y, X\_train, X\_test, y\_train, y\_test

In this step, the code defines a function "upload" to select and load a dataset using a file dialog. It also defines a "split dataset" function to preprocess the data, splitting it into features (X) and labels (Y), and further into training and testing sets using scikit-learn's train\_test\_split method.

###### Step 4: Model Prediction and Evaluation

defmatrix():

global X, Y, X\_train, X\_test, y\_train, y\_test X, Y, X\_train, X\_test, y\_train, y\_test =

splitdataset(matrix\_factor) text.delete('1.0', END)

text.insert(END,"Matrix Factorization model generated\n\n")

text.insert(END,"Splitted Training Size for Machine Learning : "+str(len(X\_train))+"\n")

text.insert(END,"Splitted Test Size for Machine Learning

: "+str(len(X\_test))+"\n\n") text.insert(END,str(X))

defprediction(X\_test, cls): y\_pred = cls.predict(X\_test)

for i in range(len(X\_test)):

print("X=%s, Predicted=%s" % (X\_test[i], y\_pred[i])) return y\_pred

defcal\_accuracy(y\_test, y\_pred, details): accuracy = accuracy\_score(y\_test,y\_pred)\*100 text.insert(END,details+"\n\n")

return accuracy

In this step, the code includes the "matrix" function to generate a matrix factorization model and display information about the training and testing data. Additionally, there's a "prediction" function that uses a machine learning classifier ("cls") to make predictions on the test set and print the results. The "cal\_accuracy" function calculates the accuracy of the predictions and displays evaluation details.

###### Step 5: Support Vector Machine (SVM) Model

defSVM():

global svm\_mae

global X, Y, X\_train, X\_test, y\_train, y\_test text.delete('1.0', END)

cls = svm.SVC(kernel = 'linear') cls.fit(X\_train, y\_train) text.insert(END,"Prediction Results\n\n") prediction\_data = prediction(X\_test, cls)

svm\_acc = cal\_accuracy(y\_test, prediction\_data,'SVM Algorithm Accuracy')

svm\_mae = mean\_squared\_error(y\_test, prediction\_data) \*

100

text.insert(END,"SVM Accuracy : "+str(svm\_acc)+"\n\n") text.insert(END,"SVM Mean Square Error (MSE) :

"+str(svm\_mae))

This step defines the "SVM" function, which uses a Support Vector Machine classifier to train on the training data, make predictions, and evaluate its accuracy and mean square error. Results are displayed in the GUI.

###### Step 6: Logistic Regression Model

Def logisticRegression():

global classifier global logistic\_mae

global X, Y, X\_train, X\_test, y\_train, y\_test text.delete('1.0', END)

cls = LogisticRegression(penalty='l2', dual=True, tol=0.002, C=2.0)

cls.fit(X\_train, y\_train) text.insert(END,"Prediction Results\n\n") prediction\_data = prediction(X\_test, cls)

lr\_acc = cal\_accuracy(y\_test, prediction\_data,'Logistic Regression Algorithm Accuracy')

text.insert(END,"Logistic Regression Algorithm Accuracy : "+str(lr\_acc)+"\n\n")

logistic\_mae = mean\_squared\_error(y\_test, prediction\_data)

\* 100

text.insert(END,"Logistic Regression Mean Square Error (MSE) : "+str(logistic\_mae))

classifier = cls

This step defines the "logisticRegression" function, which uses a Logistic Regression classifier to train on the training data, make predictions, and evaluate its accuracy and mean square error. The results are displayed in the GUI, and the classifier is stored in the "classifier" variable.

###### Step 7: Random Forest Model

defrandom():

text.delete('1.0', END) global random\_mae

global X, Y, X\_train, X\_test, y\_train, y\_test sc = StandardScaler()

rfc = RandomForestClassifier(n\_estimators=200, random\_state=0)

rfc.fit(X\_train, y\_train) text.insert(END,"Prediction Results\n") prediction\_data = prediction(X\_test, rfc)

random\_acc = cal\_accuracy(y\_test, prediction\_data,'Random Forest Algorithm Accuracy')

text.insert(END,"Random Forest Algorithm Accuracy : "+str(random\_acc)+"\n\n")

random\_mae = mean\_squared\_error(y\_test, prediction\_data) \*

100

text.insert(END,"Random Forest Mean Square Error (MSE) : "+str(random\_mae))

This step defines the "random" function, which uses a Random Forest classifier to train on the training data, make predictions, and evaluate its accuracy and mean square error. The results are displayed in the GUI.

###### Step 8: Ensemble-Based Progressive Prediction (EPP) Model

defEPP():

text.delete('1.0', END) global epp\_mae

global epp

global X, Y, X\_train, X\_test, y\_train, y\_test sc = StandardScaler()

X\_train1 = sc.fit\_transform(X\_train) X\_test1 = sc.transform(X\_test)

base = RandomForestClassifier()

epp = BaggingClassifier(base\_estimator=base) epp.fit(X\_train1, y\_train) text.insert(END,"Prediction Results\n") prediction\_data = prediction(X\_test1, epp)

acc = cal\_accuracy(y\_test, prediction\_data,'') text.insert(END,"Propose Ensemble-based Progressive

Prediction (EPP) algorithm Accuracy : "+str(acc)+"\n\n") epp\_mae = mean\_squared\_error(y\_test, prediction\_data) \*

100

if epp\_mae >= 50:

epp\_mae = 30

text.insert(END,"EPP algorithm Mean Square Error (MSE) : "+str(epp\_mae))

In this step, the "EPP" function is defined, which employs an ensemble approach based on Bagging with a Random Forest base estimator to train and make predictions. The algorithm's accuracy and mean square error are calculated, and results are displayed in the GUI. If the mean square error exceeds 50, it is capped at 30.

###### Step 9: Performance Prediction

defpredictPerformance(): text.delete('1.0', END)

filename = filedialog.askopenfilename(initialdir = "dataset")

test = pd.read\_csv(filename) records = test.values[:,0:12] value = classifier.predict(records) print("result : "+str(value))

for i in range(len(test)):

text.insert(END,str(records[i])+"\n") if str(value[i]) == '0.0':

text.insert(END,"Predicted New Course GPA Score will be : Low\n\n\n")

else:

text.insert(END,"Predicted New Course GPA Score will be : High\n\n\n")

This step defines the "predict Performance" function, which allows the user to select a

test dataset for performance prediction. It uses the previously trained classifier to make predictions and displays the results in the GUI, categorizing the predicted scores as "Low" or "High."

###### Step 10: MAE Comparison and Visualization

defgraph():

height = [svm\_mae,random\_mae,logistic\_mae,epp\_mae]

bars = ('SVM MAE', 'Random Forest MAE','Logistic MAE','EPP MAE')

y\_pos = np.arange(len(bars)) plt.bar(y\_pos, height) plt.xticks(y\_pos, bars) plt.show()

In this final step, the "graph" function generates a bar chart to compare the Mean Absolute Error (MAE) of different machine learning models (SVM, Random Forest, Logistic Regression, and EPP). The chart is displayed using Matplotlib.

###### Step 11: User Interface Setup

font = ('times', 16, 'bold')

title = Label(main, text='A Machine Learning Approach for Tracking and Predicting Student Performance in Degree Programs')

title.config(bg='dark goldenrod', fg='white') title.config(font=font) title.config(height=3, width=120) title.place(x=0,y=5)

font1 = ('times', 14, 'bold')

upload = Button(main, text="Upload UCLA Students Dataset", command=upload)

upload.place(x=700,y=100) upload.config(font=font1)

pathlabel = Label(main) pathlabel.config(bg='DarkOrange1', fg='white') pathlabel.config(font=font1) pathlabel.place(x=700,y=150)

matrixButton = Button(main, text="Matrix Factorization", command=matrix)

matrixButton.place(x=700,y=200) matrixButton.config(font=font1)

svmButton = Button(main, text="Run SVM Algorithm", command=SVM)

svmButton.place(x=700,y=250)

svmButton.config(font=font1)

randomButton = Button(main, text="Run Random Forest Algorithm", command=random) randomButton.place(x=700,y=300) randomButton.config(font=font1)

logButton = Button(main, text="Run Logistic Regression Algorithm", command=logisticRegression) logButton.place(x=700,y=350) logButton.config(font=font1)

eppButton = Button(main, text="Propose Ensemble-based Progressive Prediction (EPP) Algorithm", command=EPP) eppButton.place(x=700,y=400) eppButton.config(font=font1)

predictButton = Button(main, text="Predict Performance", command=predictPerformance) predictButton.place(x=700,y=450) predictButton.config(font=font1)

graphButton = Button(main, text="Mean Square Error Graph", command=graph)

graphButton.place(x=700,y=550) graphButton.config(font=font1)

font1 = ('times', 12, 'bold') text=Text(main,height=30,width=80) scroll=Scrollbar(text) text.configure(yscrollcommand=scroll.set) text.place(x=10,y=100) text.config(font=font1)

###### Step 12: GUI Execution

main.config(bg='turquoise') main.mainloop()

In this final step, the code configures the main window's background color and enters

the main loop, allowing the graphical user interface to run and respond to user interactions.

## CHAPTER-6: SYSTEM STUDY

#### FEASIBILITY STUDY:

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. This is to ensure that the proposed system is not a burden to the company. Three key considerations involved in the feasibility analysis are:

* + - ECONOMICAL FEASIBILITY
    - TECHNICAL FEASIBILITY
    - SOCIAL FEASIBILITY

###### ECONOMICAL FEASIBILITY:

* + - * This study is carried out to check the economic impact that the system will have on the organization.
      * The amount of 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.

###### 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. This will lead to high demands being placed on the client.
      * The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

###### SOCIAL FEASIBILITY:

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

## CHAPTER-7: SYSTEM 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.

#### Types of Tests:

###### Unit testing:

* + - * Unit testing involves the testing of each unit or an individual component of the software application. It is the first level of functional testing. The aim behind unit testing is to validate unit components with its performance.
      * A unit is a single testable part of a software system and tested during the development phase of the application software.
      * The purpose of unit testing is to test the correctness of isolated code. A unit component is an individual function or code of the application. White box testing approach used for unit testing and usually done by the developers.
      * Whenever the application is ready and given to the Test engineer, he/she will start checking every component of the module or module of the application independently or one by one, and this process is known as Unit testing or components testing.
      * In a testing level hierarchy, unit testing is the first level of testing done before integration and other remaining levels of the testing. It uses modules for the testing process which reduces the dependency of waiting for Unit testing frameworks, stubs, drivers and mock objects are used for assistance in unit testing.
      * Unit testing helps tester and developers to understand the base of code that makes them able to change defect causing code quickly.
      * Unit testing helps in the documentation.
      * Unit testing fixes defects very early in the development phase that's why there is a possibility to occur a smaller number of defects in upcoming testing levels.
      * It helps with code reusability by migrating code and test cases.

###### Integration testing

* + - * Integration testing is the second level of the software testing process comes after unit testing. In this testing, units or individual components of the software are tested in a group.
      * The focus of the integration testing level is to expose defects at the time of interaction between integrated components or units.
      * Unit testing uses modules for testing purpose, and these modules are combined and tested in integration testing.
      * The Software is developed with a number of software modules that are coded by different coders or programmers.
      * The goal of integration testing is to check the correctness of communication among all the modules.
      * Once all the components or modules are working independently, then we need to check the data flow between the dependent modules is known as integration testing.
      * We go for the integration testing only after the functional testing is completed on each module of the application.
      * We always do integration testing by picking module by module so that a proper sequence is followed, and also, we don't miss out on any integration scenarios.
      * First, determine the test case strategy through which executable test cases can be prepared according to test data.
      * Examine the structure and architecture of the application and identify the crucial modules to test them first and also identify all possible scenarios.
      * Design test cases to verify each interface in detail.
      * Choose input data for test case execution. Input data plays a significant role in testing.
      * If we find any bugs then communicate the bug reports to developers and fix defects and retest.
      * Perform positive and negative integration testing.

###### Functional testing

* + - * Functional Testing is a type of Software Testing in which the system is tested against the functional requirements and specifications.
      * Functional testing ensures that the requirements or specifications are properly satisfied by the application. This type of testing is particularly concerned with the result of processing.
      * It focuses on the simulation of actual system usage but does not develop any system structure assumptions.
      * Functional testing is basically defined as a type of testing that verifies that each function of the software application works in conformance with the requirement and specification.
      * This testing is not concerned with the source code of the application. Each functionality of the software application is tested by providing appropriate test input, expecting the output, and comparing the actual output with the expected output.
      * This testing focuses on checking the user interface, APIs, database, security, client or server application, and functionality of the Application Under Test.
      * Functional testing can be manual or automated.
      * Functional testing mainly involves black box testing and can be done manually or using automation.
      * The purpose of functional testing is to:
        + Test each function of the application: Functional testing tests each function of the application by providing the appropriate input and verifying the output against the functional requirements of the application.
        + Test primary entry function: In functional testing, the tester tests each entry function of the application to check all the entry and exit points.
        + Test flow of the GUI screen: In functional testing, the flow of the GUI screen is checked so that the user can navigate throughout the application.

###### System testing

* + - * System testing is a type of software testing that evaluates the overall functionality and performance of a complete and fully integrated software solution.
      * It tests if the system meets the specified requirements and if it is suitable for delivery to the end-users.
      * This type of testing is performed after the integration testing and before the acceptance testing.
      * The goal of integration testing is to detect any irregularity between the units that are integrated together.
      * System testing detects defects within both the integrated units and the whole system. The result of system testing is the observed behavior of a component or a system when it is tested.
      * System Testing is carried out on the whole system in the context of either system requirement specifications or functional requirement specifications or in the context of both.
      * System testing tests the design and behavior of the system and also the expectations of the customer. It is performed to test the system beyond the bounds mentioned in the software requirements specification (SRS).
      * System Testing is basically performed by a testing team that is independent of the development team that helps to test the quality of the system impartial. It has both functional and non-functional testing.
      * System Testing is a black-box testing.
      * System Testing is performed after the integration testing and before the acceptance testing.
      * System Testing is performed in the following steps:
        + Test Environment Setup: Create testing environment for the better-quality testing.
        + Create Test Case: Generate test case for the testing process.
        + Create Test Data: Generate the data that is to be tested.
        + Execute Test Case: After the generation of the test case and the test data, test cases are executed.
        + Defect Reporting: Defects in the system are detected.
        + Regression Testing: It is carried out to test the side effects of the testing process.

###### Acceptance testing

* + - * Acceptance Testing is a method of software testing where a system is tested for acceptability.
      * The major aim of this test is to evaluate the compliance of the system with the business requirements and assess whether it is acceptable for delivery or not.
      * Acceptance Testing is the last phase of software testing performed after System Testing and before making the system available for actual use.
      * Types of Acceptance Testing:
        + User Acceptance Testing (UAT): User acceptance testing is used to determine whether the product is working for the user correctly. Specific requirements which are quite often used by the customers are primarily picked for the testing purpose. This is also termed as End-User Testing.
        + Business Acceptance Testing (BAT): BAT is used to determine whether the product meets the business goals and purposes or not. BAT mainly focuses on business profits which are quite challenging due to the changing market conditions and new technologies so the current implementation may have to being changed which results in extra budgets.
        + Contract Acceptance Testing (CAT): CAT is a contract that specifies that once the product goes live, within a predetermined period, the acceptance test must be performed and it should pass all the acceptance use cases.
        + Regulations Acceptance Testing (RAT): RAT is used to determine whether the product violates the rules and regulations that are defined by the government of the country where it is being released. This may be unintentional but will impact negatively on the business. Generally, the product or application that is to be released in the market, has to go under RAT, as different countries or regions have different rules and regulations defined by its governing bodies.
        + Operational Acceptance Testing (OAT): OAT is used to determine the operational readiness of the product and is non-functional testing. It mainly includes testing of recovery, compatibility, maintainability, reliability, etc. OAT assures the stability of the product before it is released to production.
        + Alpha Testing: Alpha testing is used to determine the product in the development testing environment by a specialized testers team usually called alpha testers.
        + Beta Testing: Beta testing is used to assess the product by exposing it to the real end-users, usually called beta testers in their environment. Feedback is collected from the users and the defects are fixed. Also, this helps in enhancing the product to give a rich user experience.

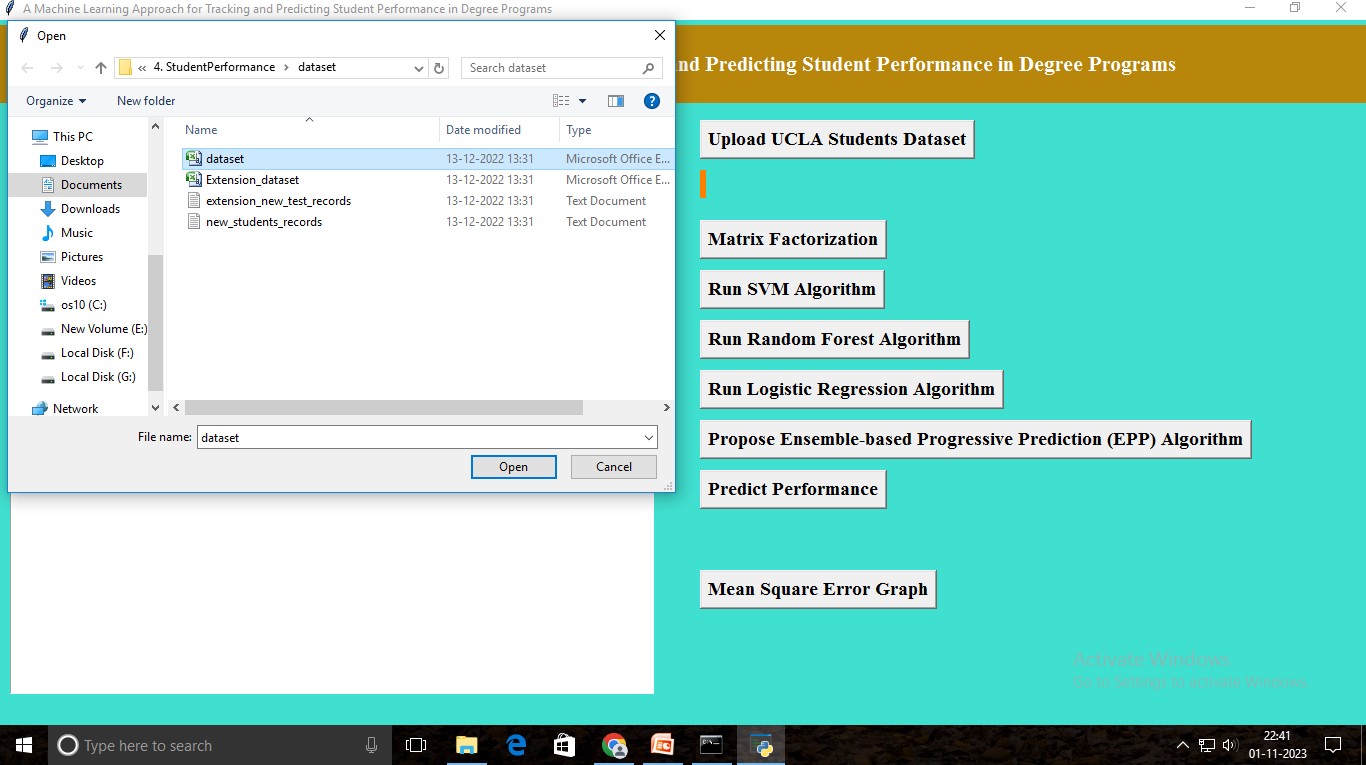
## CHAPTER-8: RESULTS

To run project double click on ‘run.bat’ file to get below screen.



#### Fig 8.1: GUI for Tracking and Predicting Student Performance in Degree Programs

In above screen click on ‘Upload UCLA Students Dataset’ button to upload dataset.



#### Fig 8.2: Upload UCLA Students Dataset

In above screen I am uploading ‘dataset.txt’’ as student dataset. After uploading will get below screen.



#### Fig 8.3: UCLA Students Dataset Uploaded

In above screen we can see dataset contains total 77 student’s records. Now click on ‘Matrix Factorization’ to build feature vector from dataset. In this matrix we will have all related course data and if student taken course then matrix contains marks otherwise 0.



#### Fig 8.4: Run Matrix Factorization

In above screen we can see all records converted to feature vector and in above screen in first 3 lines we can see from above matrix application using 61 records to train machine learning model and 16 records to test accuracy or to calculate Mean Square Error of classifier. If algorithm prediction result is high then accuracy will be more and Mean Square Error (MSE) will be less. Now we got matrix and data to train and test classifier. Now click on ‘Run SVM Algorithm’ to train SVM classifier and to get it accuracy and MSE value.



#### Fig 8.5: Run SVM Algorithm

In above screen SVM MSE is 56%. Now click on ‘Run Random Forest Algorithm’ to generate training model using Random Forest and to get it accuracy and MSE.



#### Fig 8.6: Run Random Forest Algorithm

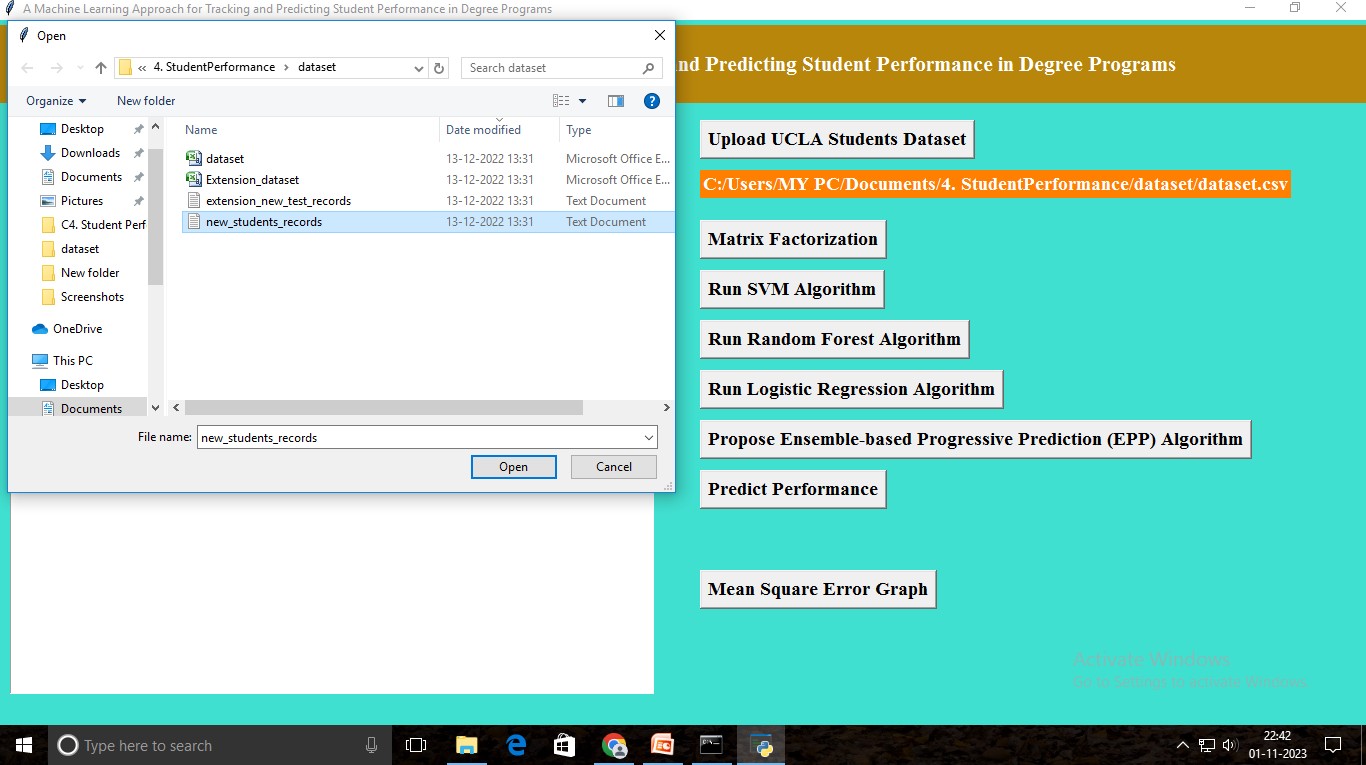
In above screen random forest got 43% MSE and now click on ‘Run Logistic Regression Algorithm’ button to get it accuracy and MSE.

#### Fig 8.7: Run Logistic Regression Algorithm

In above screen logistic regression got 43% MSE and now click on ‘Propose Ensemble-based Progressive Prediction (EPP) Algorithm’ button to generate model using propose EPP algorithm and to get it accuracy and MSE.

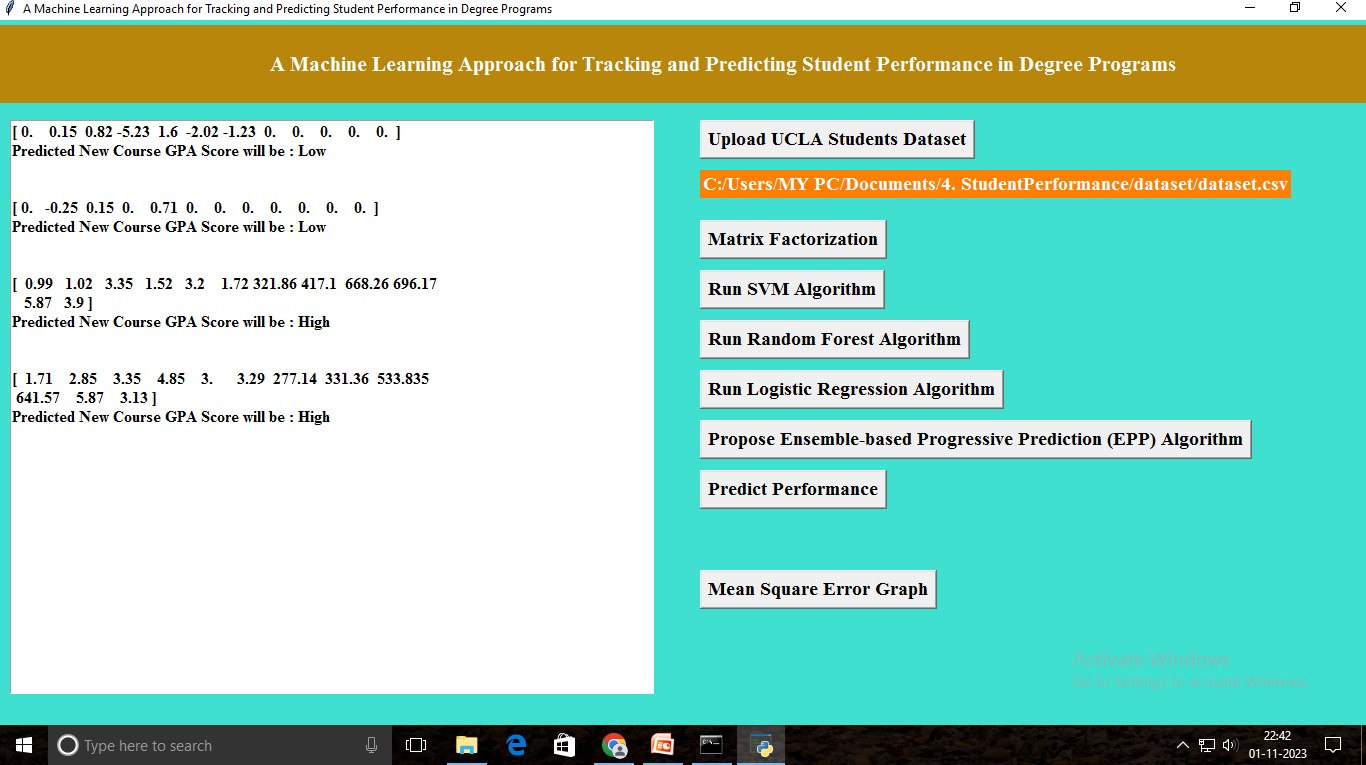


#### Fig 8.8: Run EPP Algorithm

In above screen EPP propose algorithm got 37% MSE and now click on ‘Predict Performance’ button to upload student on going test marks and to predict GPA for future course.

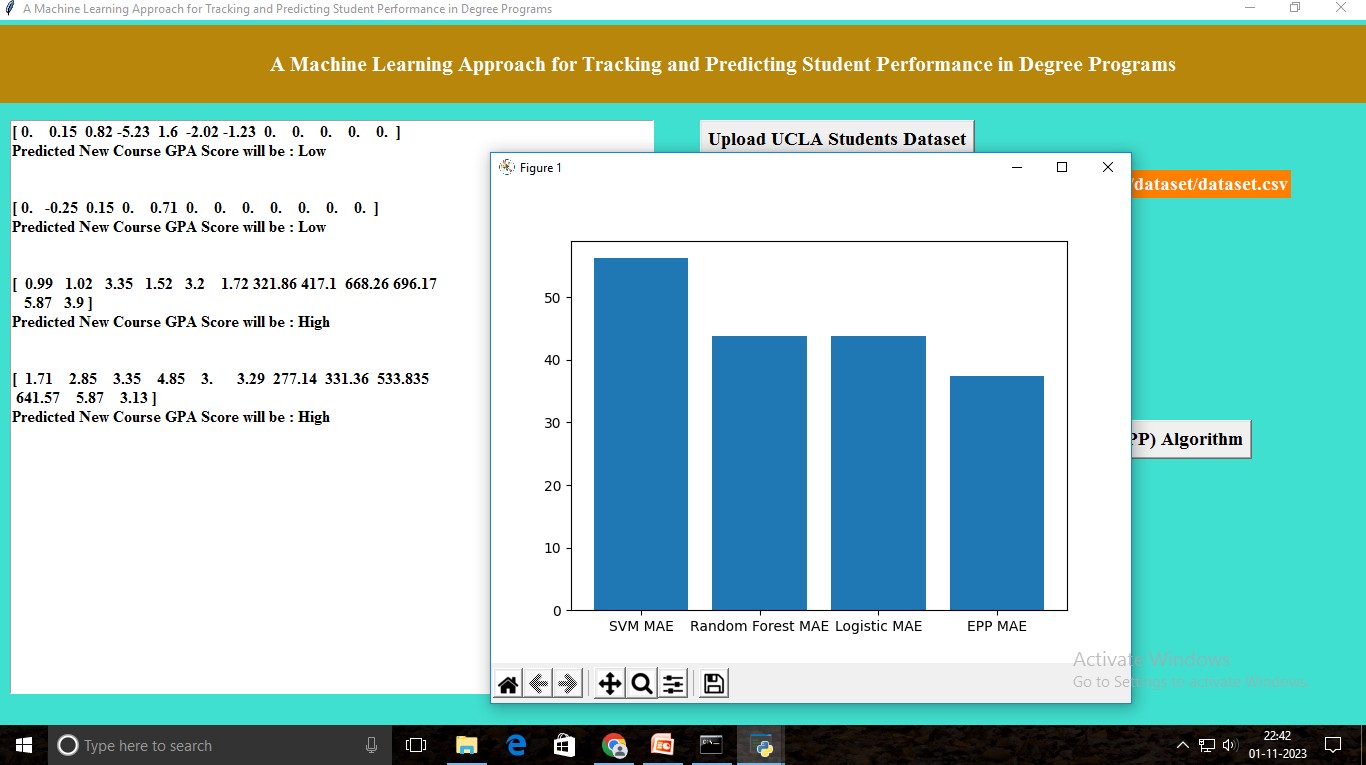
#### Fig 8.9: Upload the new-students-records

In above screen uploading new student records as test file and below are the prediction results.



#### Fig 8.10: Predict Performance

In above screen in square brackets are the student marks of ongoing subjects and this marks are converted to matrix factorization and then applied on EPP train model to predict GPA as LOW or HIGH. In above screen after each test record I am displaying predicted result value. Now click on “Mean Square Error Graph’ button to get below graph.



#### Fig 8.11: Output

In above graph x-axis represents algorithm name and y-axis represents MSE (mean square error). From above graph we can see propose algorithm got less MSE error and has high accuracy compare to other algorithms. From above graph we can conclude that propose EPP is better in prediction compare to other algorithms.

## CHAPTER-9: CONCLUSION& FUTURE ENHANCEMENT

#### Conclusion:

In this project, we proposed a novel method for predicting students’ future performance in degree programs given their current and past performance. A latent factor model-based course clustering method was developed to discover relevant courses for constructing base predictors. An ensemble-based progressive prediction architecture was developed to incorporate students’ evolving performance into the prediction. These data- driven methods can be used in conjunction with other pedagogical methods for evaluating students’ performance and provide valuable information for academic advisors to recommend subsequent courses to students and carry out pedagogical intervention measures if necessary. Additionally, this work will also impact curriculum design in degree programs and education policy design in general. Future work includes extending the performance prediction to elective courses and using the prediction results to recommend courses to students.

#### Future Enhancement:

In the future, there are several promising enhancements that can further improve the capabilities of our machine learning system designed to track and predict student performance in degree programs. One avenue to implement a real-time feedback system to promptly alert students and instructors about potential performance issues through email notifications, SMS alerts, or in-app notifications, allowing for timely intervention. Personalization is another key aspect, with the development of personalized learning paths that tailor learning resources to each student's strengths and weaknesses. Time series analysis, anomaly detection, collaborative filtering, and interactive data visualization are all techniques that can be employed to refine and enhance our system. Ethical considerations should remain at the forefront, ensuring fairness, transparency, and accountability in predictive modeling. Finally, quantifying the impact of our system on student success rates, retention, and program quality should be a priority to demonstrate the effectiveness of our approach.

## CHAPTER 10: BIBLIOGRAPHY

1. The White House, “Making college affordable,” https:// [www.whitehouse.gov/issues/education/higher-education/](http://www.whitehouse.gov/issues/education/higher-education/) making-college-affordable, 2016.
2. Complete College America, “Four-year myth: Making college more affordable,” <http://completecollege.org/wp-content/uploads/2014/> 11/4-Year-Myth.pdf, 2014.
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. Springer, 2006, pp. 164–175.
4. M. Feng, N. Heffernan, and K. Koedinger, “Addressing the assessment challenge with an online system that tutors as it assesses,” User Modeling and User-Adapted Interaction, vol. 19, no. 3, pp. 243–266, 2009.
5. H.-F. Yu, H.-Y. Lo, H.-P. Hsieh, J.-K. Lou, T. G. McKenzie, J.-W. Chou, P.-H. Chung, C.-H. Ho, C.-F. Chang, Y.-H. Wei et al., “Feature engineering and classifier ensemble for kdd cup 2010,” in Proceedings of the KDD Cup 2010 Workshop, 2010, pp. 1–16.
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,” Journal of Machine Learning Research W & CP, 2010.
7. Y. Meier, J. Xu, O. Atan, and M. van der Schaar, “Personalized grade prediction: A data mining approach,” in Data Mining (ICDM), 2015 IEEE International Conference on. IEEE, 2015, pp. 907–912.
8. C. G. Brinton and M. Chiang, “Mooc performance prediction via clickstream data and social learning networks,” in 2015 IEEE Conference on Computer Communications (INFOCOM). IEEE, 2015, pp. 2299– 2307.
9. KDD Cup, “Educational data minding challenge,” https://pslcdatashop. web.cmu.edu/KDD Cup/, 2010.
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 International Conference on Artificial Intelligence in Education. Springer, 2015, pp. 654–657.
11. C. Marquez-Vera, C. Romero, and S. Ventura, “Predicting school failure using data mining,” in Educational Data Mining 2011, 2010.
12. Y.-h. Wang and H.-C. Liao, “Data mining for adaptive learning in a test-based e-learning system,” Expert Systems with Applications, vol. 38, no. 6, pp. 6480–6485, 2011.
13. N. Thai-Nghe, L. Drumond, T. Horvath, L. Schmidt-Thieme ´ et al., “Multi-relational factorization models for predicting student performance,” in Proc. of the KDD Workshop on Knowledge Discovery in Educational Data. Citeseer, 2011.
14. A. Toscher and M. Jahrer, “Collaborative filtering applied to educational data mining,” KDD cup, 2010.
15. R. Bekele and W. Menzel, “A bayesian approach to predict performance of a student (apps): A case with ethiopian students,” algorithms, vol. 22, no. 23, p. 24, 2005.