

A Project report on

**Academic Performance Prediction using Multisource
Multi Feature Behavioral Data**

A Dissertation submitted to JNTU Hyderabad in partial fulfillment of the
academic requirements for the award of the degree.

Bachelor of Technology
in
Computer Science and Engineering

Submitted by

IFFAT MARIA
19H51A05A2

YELLARAM ARCHANA
19H51A05C1

BHUVANA VANGARI
19H51A05J2

Under the esteemed guidance of

Mr. G SAIDULU
Assistant Professor



Department of Computer Science and Engineering

CMR COLLEGE OF ENGINEERING & TECHNOLOGY

(UGC Autonomous)

*Approved by AICTE *Affiliated to JNTUH *NAAC Accredited with A⁺ Grade

KANDLAKOYA, MEDCHAL ROAD, HYDERABAD - 501401.

2019- 2023

CMR COLLEGE OF ENGINEERING & TECHNOLOGY

KANDLAKOYA, MEDCHAL ROAD, HYDERABAD – 501401

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING



CERTIFICATE

This is to certify that the Major Project Phase-1 report entitled "**Academic Performance Prediction using Multisource Multi Feature Behavioral Data**" being submitted by Iffat Maria (19H51A05A2), Yellaram Archana (19H51A05C1), Bhuvana Vangari (19H51A05J2) in partial fulfillment for the award of **Bachelor of Technology in Computer Science and Engineering** is a record of bonafide work carried out his/her under my guidance and supervision.

The results embodied in this project report have not been submitted to any other University or Institute for the award of any Degree.

Mr. G Saidulu
Assistant Professor
Dept. of CSE

Dr. Siva Skandha Sanagala
Associate Professor and HOD
Dept. of CSE

ACKNOWLEDGEMENT

With great pleasure we want to take this opportunity to express my heartfelt gratitude to all the people who helped in making this project work a grand success.

We are grateful to **Mr. G Saidulu, Assistant Professor**, Department of Computer Science and Engineering for his valuable technical suggestions and guidance during the execution of this project work.

We would like to thank **Dr. Siva Skandha Sanagala**, Head of the Department of Computer Science and Engineering, CMR College of Engineering and Technology, who is the major driving forces to complete my project work successfully.

We are very grateful to **Dr. Vijaya Kumar Koppula**, Dean-Academic, CMR College of Engineering and Technology, for his constant support and motivation in carrying out the project work successfully.

We are highly indebted to **Major Dr. V A Narayana**, Principal, CMR College of Engineering and Technology, for giving permission to carry out this project in a successful and fruitful way.

We would like to thank the **Teaching & Non- teaching** staff of Department of Computer Science and Engineering for their co-operation

We express our sincere thanks to **Shri. Ch. Gopal Reddy**, Secretary, CMR Group of Institutions, for his continuous care.

Finally, we extend thanks to our parents who stood behind us at different stages of this Project. We sincerely acknowledge and thank all those who gave support directly and indirectly in completion of this project work.

Iffat Maria	19H51A05A2
Yellaram Archana	19H51A05C1
Bhuvana Vangari	19H51A05J2

TABLE OF CONTENTS

CHAPTER NO.	TITLE	PAGE NO.
	LIST OF FIGURES	ii
	LIST OF TABLES	iii
	ABSTRACT	iv
1	INTRODUCTION	1
	1.1 Problem Statement	3
	1.2 Research Objective	4
	1.3 Project Scope and Limitations	4
	1.3.1. Scope	4
	1.3.2. Limitations	5
2	BACKGROUND WORK	6
	2.1. Logistic Regression Model	7
	2.1.1. Introduction	7
	2.1.2. Merits, Demerits and Challenges	8
	2.1.3. Implementation of Logistic Regression Model	9
	2.2. Naïve Bayes Model	11
	2.2.1. Introduction	11
	2.2.2. Merits, Demerits and Challenges	13
	2.2.3. Implementation of Naïve Bayes Model	14
	2.3. Decision Tree Model	15
	2.3.1. Introduction	15
	2.3.2. Merits, Demerits and Challenges	17
	2.3.3. Implementation of Decision Tree Model	18
3	PROPOSED SYSTEM	19
	3.1. Objective of Proposed Model	21
	3.2. Algorithms Used for Proposed Model	22
	3.2.1. Long Short-Term Memory	22

3.3.	Designing	24
3.3.1.	Python	24
3.3.2.	Python Libraries Used	25
3.4	Design	27
3.4.1.	UML Diagram	27
3.4.1.1.	Architecture Diagram	27
3.4.1.2.	Class Diagram	28
3.4.1.3.	Data Flow Diagram	29
3.4.1.4.	Sequence Diagram	30
3.4.1.5.	Use Case Diagram	31
3.4.2.	Flow Charts	32
3.4.2.1.	Remote User Flow Chart	32
3.4.2.2.	Service Provider Flow Chart	33
3.5	Stepwise Implementation and Code	34
3.5.1.	Implementation	34
3.5.2.	Code	38
3.5.2.1.	Remote User Code	38
3.5.2.2	Service Provider	39
4	RESULTS AND DISCUSSION	41
4.1.	Comparison of Existing Solutions	42
4.2.	Data Collection and Performance metrics	43
4.2.1.	Data set	43
4.2.2.	Attribute Information	44
4.2.3.	Performance Matrix	46
4.3.	Results	47
5	CONCLUSION	53
5.1	Conclusion and Future Enhancement	54
	REFERENCES	55
	GitHub Link	56

List of Figures**FIGURE**

NO.	TITLE	PAGE NO.
2.1	Plot Diagram of Logistic Regression	7
2.2	Flowchart of Naive Bayes Model	12
2.3	Decision tree Algorithm	15
2.4	Flowchart of Decision Tree Model	16
3.1	Block Diagram of Proposed Model	20
3.2	Flow chart of the System	21
3.3	Sigmoid Function, the “gate” in LSTM	22
3.4	Python Logo	24
3.5	Architecture Diagram of the System	27
3.6	Class Diagram for the System	28
3.7	Data Flow Diagram for the System	29
3.8	Sequence Diagram for the System	30
3.9	Use Case diagram	31
3.10	Flowchart for Remote Users	32
3.11	Service Provider Flowchart	33
3.12	Wamp Server	34
3.13	Login Page	34
3.14	Database Upload	35
3.15	Command Prompt	35
3.16	User Login	36
3.17	Service Provider	36
3.18	Remote User	37
3.19	Login to Remote User	38
3.20	Register & View Profile	38
3.21	View student Records & Search Student Data	39
3.22	College Server Login & Performance Prediction	39

3.23	View Behavioral Change by LSTM	39
3.24	Extraction of Poor Performing Students	40
4.1	Glimpse of Dataset	45
4.2	Remote User Login Page	47
4.3	Service Provider Login Page	48
4.4	Service Provider: View All Student Data	49
4.5	Service Provider: Search Student Data	50
4.6	Service Provider: View Behavioural Problem	51
4.7	Service Provider: View Registered Users	51
4.8	Service Provider: View Poor Performing Students	52
4.9	Remote User: Upload Dataset	52

List of Tables

FIGURE

NO.	TITLE	PAGE NO.
2.1.	Performance Metrics of LR Model	9
2.2.	Performance Metrics of Naïve Bayes Model	13
2.3.	Performance Metrics of Decision Tree Model	17
4.1	Performance Measures of Dataset	42
4.2	ROC Index	43
4.3	Dataset Information	43
4.4	Attributes and Domain Values	44
4.5	Performance Metrics of Proposed Models	46
4.6	Performance Metrics of Existing Models	46

ABSTRACT

Online education is becoming increasingly popular, and it is frequently combined with traditional on-campus study to improve learning efficiency for university students. Since students have access to a wealth of online learning data, it offers a useful way to forecast students' academic performance and facilitates pre-intervention for at-risk students. Modern academic institutions operate in a highly competitive and complex environment. Analyzing performance, providing high-quality education, strategies for evaluating the students' performance, and future actions are among the prevailing challenges universities face. Student intervention plans must be implemented in these universities to overcome problems experienced by the students during their studies. Current data sources for predicting student performance are limited to data from the corresponding learning platform, where only learning behaviors from that course can be observed. However, students' academic performance will be related to other behavioral factors. This model's core idea is that it combines two-dimensional convolutional networks to extract correlation characteristics between distinct types of behavior while using long short-term memory networks to capture inherent time-series features for each type of activity. Four different types of everyday behavior data from university students in Beijing were used in our tests. The results of the experiments show that the deep model method is superior than other machine learning algorithms. The presented study is significant for educational authorities to predict students' performance before they drop out.

CHAPTER 1

INTRODUCTION

CHAPTER 1

INTRODUCTION

University is a very important place for students where they shape their futures. Sometimes, many students cannot adapt themselves to the university's environment. Due to various known and unknown reasons, students' performances in the university in many cases tend to be low which in turn affects their results. Students' performance [1] is a key indicator in measuring the quality of academic education and is also closely related to students' mental health. Related studies have shown that students with poor academic performance are prone to anxiety and depression, and their risk of suicide is much higher than that of students with excellent performance. Achievement prediction aims to identify students with high academic risk in advance, which reminds administrators, teachers, and students themselves of taking timely targeted intervention actions to avoid poor performance, such as failing courses, dropping out, staying out, and so on. Therefore, student achievement prediction has been receiving extensive attention and research. Modern learning institutions operate in a highly competitive and complex environment. Thus, analyzing performance, providing high-quality education, formulating strategies for evaluating the students' performance, and identifying future needs are some challenges faced by most universities today. Student intervention plans are implemented in universities to overcome students' problems during their studies.

Student performance prediction at entry-level and during the subsequent periods helps the universities effectively develop and evolve the intervention plans, where both the management and educators are the beneficiaries of the students' performance prediction plans. If the major factors are identified and monitored, it will give the course teachers and the administration an opportunity to improve the study environment. Also, students can work on improving their performance. Machine learning algorithms are useful tools for predicting early students at risk and their dropout chances by utilizing the derived log data. This technique is more advanced than the traditional on-campus where students' records, such as quizzes, attendance, exams, and marks, are used to evaluate and predict their academic performance.

In this project, we mostly refer to behavioral information that describes a student's social and time management skills, as well as their ability to manage stress. These aspects are frequently overlooked and not given much weight, yet research indicates that behavioral modifications can have both positive and negative effects on a student's academic success [2]. Identifying high-risk students for unsatisfactory academic achievement as soon as feasible is crucial to raising the standard of instruction. This model's core idea is that it combines two-dimensional convolutional networks to extract correlation characteristics between distinct types of behavior while using long short-term memory networks to capture inherent time-series features for each type of activity. Different types of everyday behavior data from university students of our own college were used in our tests. The results of the experiments show the students are at high risk and need improvement in their social as well as academic performance.

1.1. PROBLEM STATEMENT

The performance of students is predicted by performance predictors using ML algorithms on static data. As a result, teachers failed to track the actual learning curve of individual students. So, in order to foresee student outcomes, a multidimensional behavioral student dataset that is clean, devoid of missing values, and free of outliers, must be used. Hence a model to predict long-term student performance using supervised machine learning is needed. It provides fresh perspective on identifying the most important learning activity and helps educators monitor the academic progress of their students. Additionally, it has been shown that students who tend to have a healthy balance between their personal and professional lives perform better than those who exhibit the opposite. A projected list of at-risk students with poor academic performance is shown, which makes it much simpler for the teacher to focus on the students who need extra help.

1.2. RESEARCH OBJECTIVE

The following duties will be carried out by the performance prediction platform we want to design, develop, and implement:

- Obtaining a comprehensive picture by gathering a student's adequately detailed profile and merging the data.
- Developing a prediction model with great accuracy by looking into the aspects influencing students' academic success.
- Gaining knowledge of the proportion of pupils who perform poorly in both academics and behaviour.

1.3. PROJECT SCOPE AND LIMITATIONS

1.3.1. SCOPE

The stakeholder of this project is the university/ college as well as the students. The scope of this project is:

- In order to forecast student performance.
- To develop a reliable machine learning system to forecast student grades, so that students who are at danger can be warned in a timely and reliable manner.
- Look at each person's specific behavioral issues that have a bad impact on their academic worries and behavioral changes that have a negative impact on student performance.

1.3.2. LIMITATIONS

Predicting and assessing student performance is critical in assisting educators identify flaws and improving academic results. However, achieving accurate predictions is challenging due to the following limitations:

- We reduced the scale of the dataset by only using student-generated data from a single course in order to obtain a multisource dataset. The universality of the forecast regarding academic performance may be adversely affected by this restriction.
- Also, the primary focus of this study is behavioral modifications. This study did not analyze additional traits/features that need consideration (such as peer effect or sleep).

CHAPTER 2

BACKGROUND WORK

CHAPTER 2

BACKGROUND WORK

This section discusses findings and observations done by some research works on Academic Forecasting using various Machine Learning techniques.

2.1. LOGISTIC REGRESSION MODEL [3]

2.1.1. INTRODUCTION

Logistic Regression (LR) represents a mathematical modelling technique which describes the relationship between several independent variables, $X_1 \dots X_K$, and a dependent variable, D . The logistic model uses the logistic function as a mathematical form which has the range between 0 and 1 for any given input. The logistic model can describe the probability of an event which is always a value between 0 and 1. The following formula represents the logistic model.

$$P(D = 1 | X_1, X_2, \dots, X_k) = 1 / (1 + e^{-(\alpha + \sum_{i=1}^k \beta_i x_i)}) \quad (1)$$

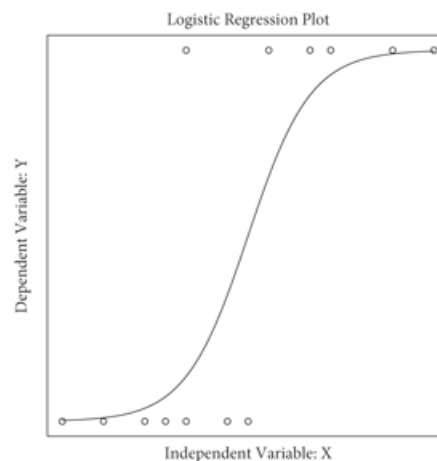


Fig 2.1. Plot diagrams of Logistic regression

Where α and β are the model's parameters that can be learned from a set of labelled instances in the training dataset.

The research instrument used is the questionnaire method coupled with extraction of information from the respective student's record in the department of Mathematics and Statistics of the University. The questionnaire consisted of 41 questions making up the predictor variable investigated. It is conveniently grouped into nine subsections. Details of the questionnaire are provided in the appendix. The questionnaire was tested for content validity and test-re-test reliability before administering it to students. The target population consists of 200 – 400 level students of Mathematics and Statistics department of the University of Maiduguri, Maiduguri, Nigeria. A random sample of 60 and 103 students from the target population of 140 were administered the questionnaire for the pilot and main survey respectively.

2.1.2. MERITS, DEMERITS AND CHALLENGES

2.1.2.1. Merits

- It is easy to implement and doesn't give discrete output, gives Probability associated with each output.
- It is robust to small noise in the data and is not particularly affected by small cases of multi-collinearity.
- No parameters to tune.

2.1.2.2. Demerits

- Cannot handle non-linearities in the data.
- Features need to be scaled and normalized.
- Cannot learn non-linear boundaries and has high bias.
- Cannot handle missing values.

2.1.2.3. Challenges

- Appropriate features must be selected before fitting the model or the model should be regularized with lasso to select the features.
- When classes are completely separable, the estimation of parameters becomes unstable in LR due to the use of logistic function.

Table 2.1. Performance Metrics of LR Model

Model	Accuracy	Precision	Recall	F1
LR	74.53	79.23	71.91	74.87

2.1.3. IMPLEMENTATION OF LR MODEL

- The data used were derived from the departmental examination record office and the questionnaire administered to students using appropriate numerical codes. The grades of the students in MTH101 course, which is the binary dependent variable, are coded as mentioned earlier. The questionnaire covered largely forty-one independent variables that were investigated.
- The choice of response to a question could be categorical, multiple choice, or alternative answers provided on a likert scale.
- The coding for the likert scale is done as follows:
 - Strongly agree is 4, agree 3, disagree 2, strongly disagree 1 and undecided 0.
- All other types of choice of response were coded in a serial sequence. The coded form of the response to the questionnaire by the 103 students in the study constitutes the data for analysis.

- The data were analyzed using statistical package for social sciences (SPSS). A stepwise Logistic Regression method was performed. The parameters were obtained by maximum likelihood method. The Neglerke R2 , Chi – Square Test, Hosmer- Lemeshow test were used to assess the model fit. The wald statistic, likelihood ratio test and odds ratio with 95% Confidence Interval (C.I) were used to assess the significance of the individual coefficient.
- Discriminant analysis was also performed on the data. Here the students falling respectively under the codes 0 and 1 of the independent variables formed the two groups for the Discriminant analysis.
- A Discriminant analysis was performed primarily as a confirmatory analysis. This is because of the advantages (Joseph et al. 2010) of the logistic regression over the Discriminant analysis. These are:
 - i. Logistic regression does not require any specific assumption on the distributional form of the independent variable.
 - ii. Heteroscedasticity does not come to play as it does in Discriminant analysis.
 - iii. Discriminant analysis relies strictly on meeting the assumptions of multivariate normality and equal variance-covariance matrices across groups – assumptions that are not met in many situations.
 - iv. Logistic regression is much more robust when these assumptions are not met

2.2. NAÏVE BAYES MODEL [4]

2.2.1. INTRODUCTION

Naïve Bayes classification model is considered as the simplest variation of the Bayesian network. This model assumes that every feature attribute is independent from the other attributes given the target attribute state. Each instance x in the dataset contains attribute values a_1, a_2, \dots, a_i . The target function $f(x)$ equals any form predefined finite set $V = (v_1, v_2, \dots, v_j)$. Naïve Bayes uses the following equation.

$$V_{\max} = \max_{v_j \in V} P(v_j) \prod_i P(a_i | v_j) \quad (2)$$

Where v represents the target of the model, $P(a_i | v_j)$ and $P(v_j)$ could be found by calculating their frequencies in training dataset.

This work predicts the student performance in terms of students' performance in the subjective and objective type of exams. The dataset of the students gets prepared through the questioner. The dataset is given as input to density-based clustering which calculate the dense region and from the dense region EPS value is calculated which is the centroid point of dense region. To calculate similarity between the points technique of Euclidian distance is applied which is given similar and dissimilar type of data. The output of density-based clustering is input to classification algorithm which will classify the data points. Here, on the basis of nearest training, the samples are classified. The feature vectors are stored along with the labels of training pictures within the training process. Towards the label of its k-nearest neighbors, the unlabeled question point is doled out during the classification process. On the basis of labels of its k nearest neighbors, the object is characterized. The object is classified essentially as the class of the object that is nearest to it in the event when $k=1$. k is known to be an odd integer in case when there are only two classes present. To improve performance of prediction analysis neural networks will be applied with density-based.

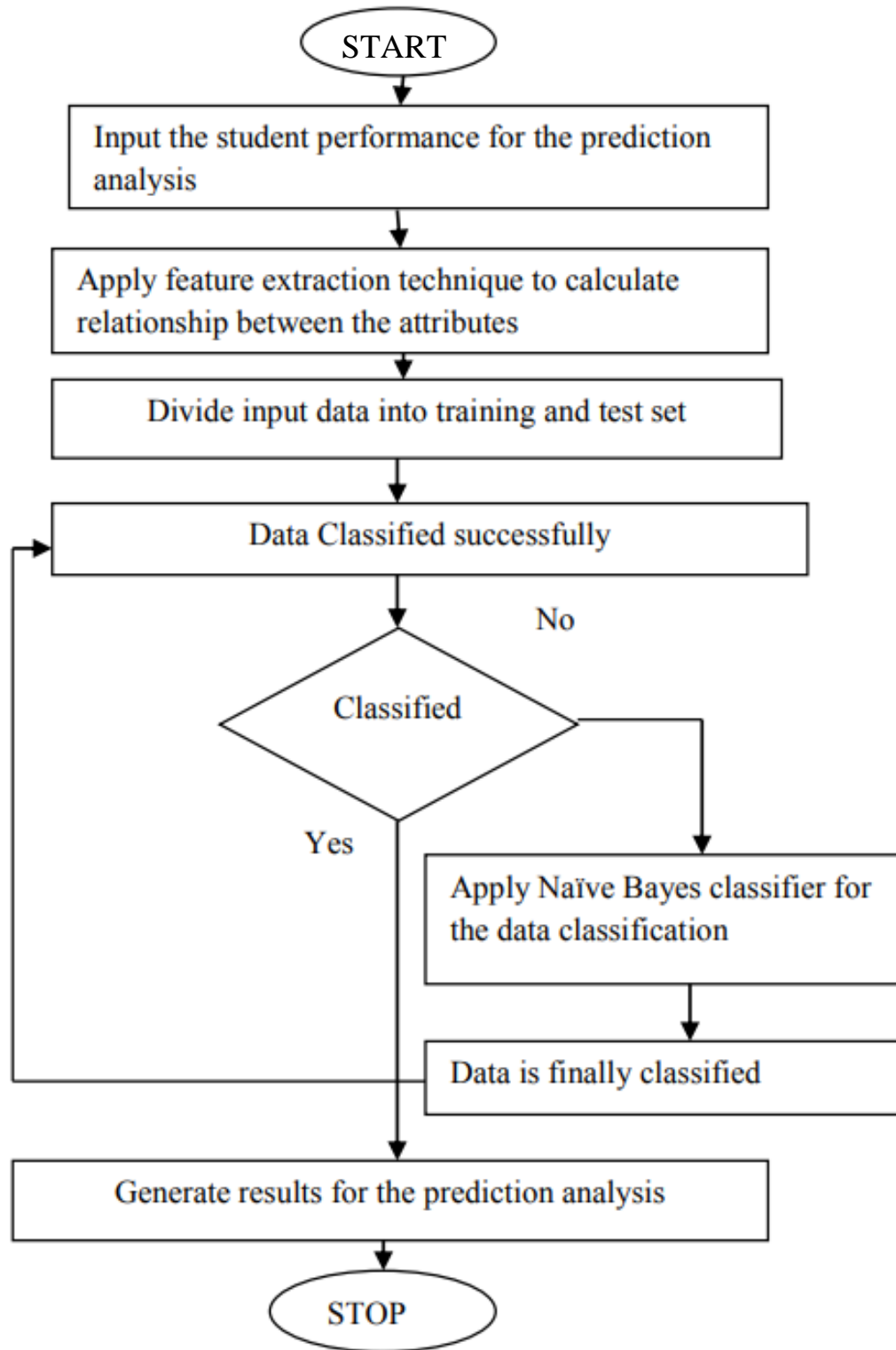


Fig 2.2. Flowchart of Naïve Bayes Model

2.2.2. MERITS, DEMERITS AND CHALLENGES

2.2.2.1. Merits

- It is simple and easy to implement
- It doesn't require as much training data
- It handles both continuous and discrete data
- It is highly scalable with the number of predictors and data points

2.2.2.2. Demerits

- Conditional Independence Assumption does not always hold. In most situations, the feature shows some form of dependency.
- **Zero probability problem:** When we encounter words in the test data for a particular class that are not present in the training data, we might end up with zero class probabilities.

2.2.2.3. Challenges

- If the model comes across a categorical feature that isn't present in the training set, the probability of 0 is assigned to that new category. This is very dangerous, as multiplying 0 with other features' probabilities will result in 0.

Table 2.2. Performance Metrics of Naïve Bayes Model

Model	Accuracy	Precision	Recall	F1
Naïve Bayes	66.52	70.51	64.27	67.21

2.2.3. IMPLEMENTATION OF NAÏVE BAYES MODEL

For building the Naïve Bayes model, the optimization operator has been set to find the best values for the Laplace correction, the estimation mode, using the application grid, the bandwidth selection, the number of kernels, and the size of application grid.

1. Separate By Class.
2. Summarize Dataset.
3. Summarize Data by Class.
4. Gaussian Probability Density Function.
5. Class Probabilities.

2.3. DECISION TREE MODEL [5]

2.3.1. INTRODUCTION

In a decision tree model, each internal node represents a test on an attribute of the dataset, each tree branch reflects the test result, and each leaf node represents a target feature label. They don't require extensive classification parameter configuration or prior knowledge of the problem domain. The approach utilises an attribute or feature selection measure to choose the attribute or feature that best separates the dataset instances into discrete target classes. This study will make use of the Knowledge Discovery in Database (KDD) process. KDD revolves on the investigation and creation of knowledge, processes, algorithms and the mechanisms for retrieving potential knowledge from data collections. This study utilized the Data mining technique, specifically; the J48 algorithm was used to create the Decision Tree Model in predicting the Student Performance in Data Structures and Algorithms. For model accuracy, K-fold cross-validation and Receiving Operating Characteristics Curve (ROC) was used.

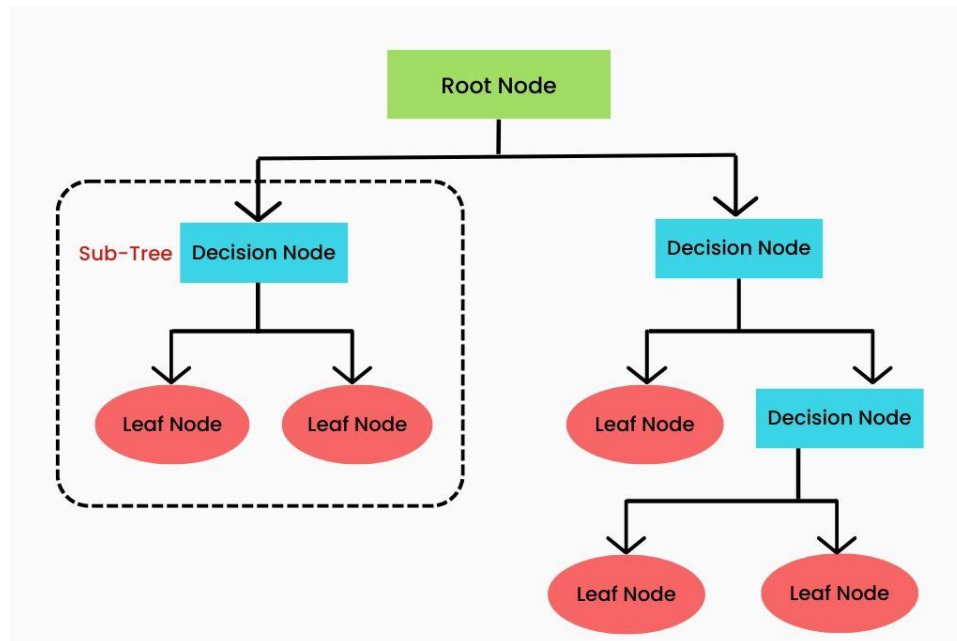


Fig 2.3. Decision tree algorithm

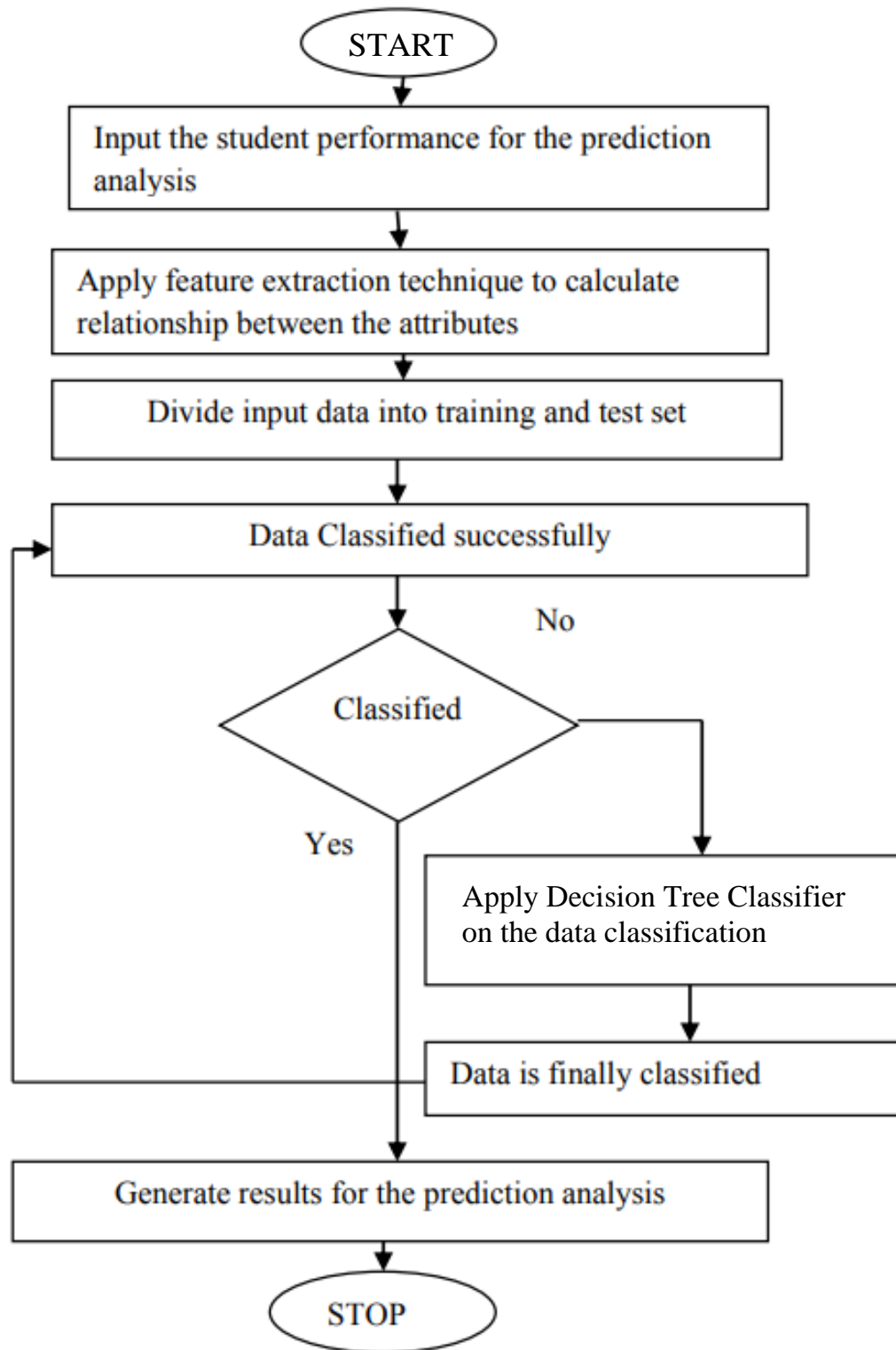


Fig 2.4. Flowchart of Decision Tree Model

2.3.2. MERITS, DEMERITS AND CHALLENGES

2.3.2.1. *Merits*

- Decision trees are simple to understand, interpret, and visualize.
- They can effectively handle both numerical and categorical data.
- Decision trees require little data preparation and data normalization.
- They perform well, even if the actual model violates the assumptions

2.3.2.2. *Demerits*

- Overfitting of data.
- Optimization at every level.
- Instability.

2.3.2.3. *Challenges*

- A small change in the data can cause a large change in the structure of the decision tree causing instability.
- Involves higher time to train the model.
- Training is relatively expensive as the complexity and time taken are more.

Table 2.3. Performance Metrics of Decision Tree Model

Model	Accuracy	Precision	Recall	F1
Decision Tree	76.93	77.96	77.83	77.88

2.3.3. IMPLEMENTATION OF DECISION TREE MODEL

- The primary data modelling stage consists of five phases: Training, pattern, testing, result evaluation and knowledge representation.
 - Next, cross-validation was used. Cross-validation is a model evaluation method where the entire data will not be utilized when training a learner.
 - Its most straightforward technique is called the holdout method. Here, data is divided into two, namely, the training set and test set. The training set is used to train the model, while the test set is used to evaluate it.
 - The J48 algorithm [6] is used in the training stage and was used to build a model. The J48 classifier is a simple C4.5 decision tree for classification for the creation of a binary tree¹⁹. The testing stage, on the other hand, is where the K-fold cross validation is performed.
 - For building the DT model, the Optimize Parameters operator has been set to find the best value of the splitting criterion, and the minimal size for split properties. Also, apply pruning property has be set by the optimization operator. All the other parameters has been set to the default values.
1. Pre-process the dataset.
 2. Split the dataset from train and test using Python sklearn package.
 3. Train the classifier.
 4. Make predictions.
 5. Calculate the accuracy.

CHAPTER 3

PROPOSED SYSTEM

CHAPTER 3

PROPOSED SYSTEM

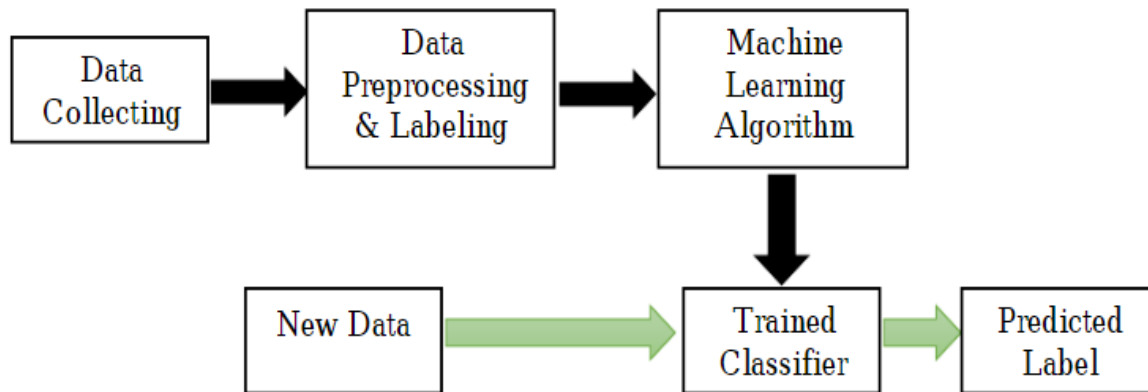


Fig 3.1. Block Diagram of Proposed Model

In the proposed system, this model mainly consists of the following three modules:

- **Data Module** in which multisource data on campus covering a large variety of data trails are aggregated and fused, and the characteristics/features that can represent students' behavioral change from three different perspectives are evaluated.
- **Prediction Module** in which academic performance prediction is considered a classification problem that is solved by machine learning (ML)-based algorithms; and
- **Feedback Module** in which visualized feedback is delivered individually based on the predictions made and feature analysis.

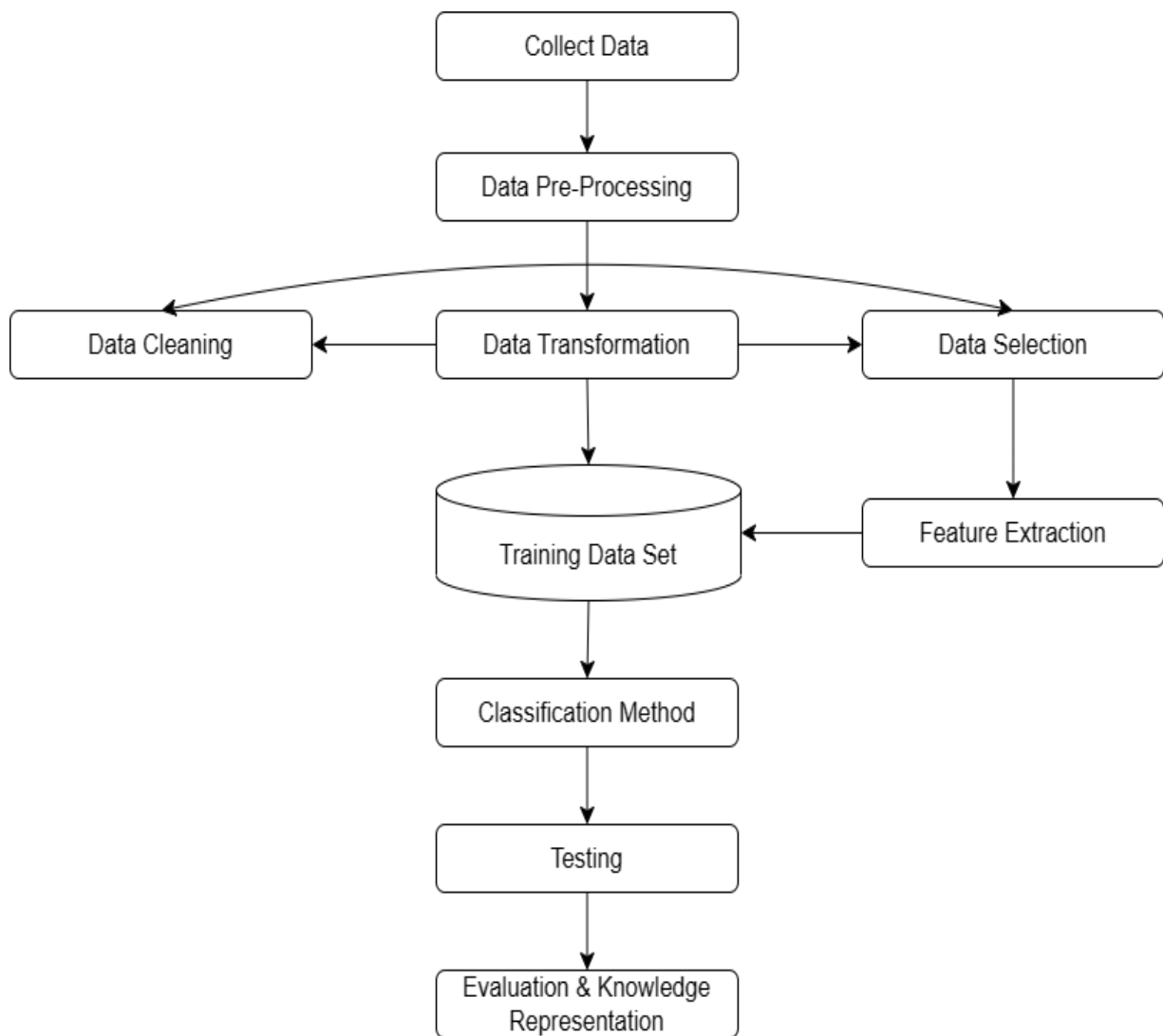


Fig 3.2. Flow chart of the system

3.1. OBJECTIVE OF PROPOSED MODEL

The ambition of this project is to:

- Employ efficient classification methodology and to forecast the student performance.
- To identify students that are likely to perform poorly in examinations to upgrade their performance by the end semester.

3.2. ALGORITHM FOR PROPOSED MODEL

3.2.1. LONG SHORT-TERM MEMORY

Long Short-Term Memory (LSTM) [7] networks are a type of Recurrent Neural Network that can learn order dependence. The output of the previous step is used as input in the current step in RNN. It addressed the issue of RNN long-term dependency, in which the RNN is unable to predict words stored in long-term memory but can make more accurate predictions based on current data. RNN does not provide an efficient performance as the gap length rises. The LSTM may keep information for a long time by default. It is used for time-series data processing, prediction, and classification. It has feedback connections, unlike conventional feed-forward neural networks. It can handle not only single data points (like photos) but also complete data streams (such as speech or video) and can be used for tasks like unsegmented, linked handwriting recognition, or speech recognition.

It saves information and deals with long sequences effectively by using memory cells and three gates. It uses structured gates to add or forget information to control memory cells. Forget gate is used to decide which information is to be removed. The sigmoid function is used for this purpose:

- If output is 1, information is remembered, and forgets the information if the output is 0.

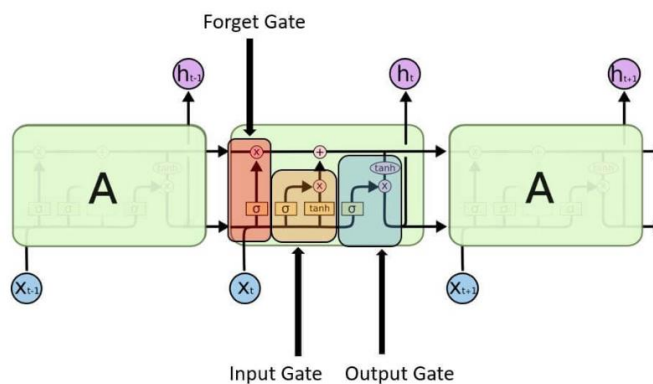


Fig 3.3. Sigmoid Function, the “gate” in LSTM

The cells store information, whereas the gates manipulate memory. There are three entrances:

- **Input Gate:** It determines which of the input values should be used to change the memory. The **sigmoid** function determines whether to allow 0 or 1 values through. And the **tanh** function assigns weight to the data provided, determining their importance on a scale of -1 to 1.

$$i_t = \sigma(W_i \cdot [h_t - 1, x_t] + b_i)$$
$$C_t = \tanh(W_c \cdot [h_t - 1, x_t] + b_c)$$

- **Forget Gate:** It finds the details that should be removed from the block. It is decided by a **sigmoid** function. For each number in the cell state C_{t-1} , it looks at the preceding state (h_{t-1}) and the content input (x_t) and produces a number between 0 (omit this) and 1 (keep this).

$$f_t = \sigma(W_f \cdot [h_t - 1, x_t] + b_f)$$

- **Output Gate:** The block's input and memory are used to determine the output. The **sigmoid** function determines whether to allow 0 or 1 values through. And the **tanh** function determines which values are allowed to pass through 0, 1. And the **tanh** function assigns weight to the values provided, determining their relevance on a scale of -1 to 1 and multiplying it with the sigmoid output.

$$O_t = \sigma(W_o[h_t - 1, x_t] + b_o)$$
$$h_t = o_t * \tanh(C_t)$$

The recurrent neural network uses long short-term memory blocks to provide context for how the software accepts inputs and creates outputs. Because the program uses a structure based on short-term memory processes to build longer-term memory, the unit is dubbed a long short-term memory block. In natural language processing, these systems are extensively used.

3.3. SOFTWARE ENVIRONMENT AND LIBRARIES

3.3.1. PYTHON

Python is a high-level, general-purpose programming language. [8] Its design philosophy emphasizes code readability with the use of significant indentation via the off-side rule. Python is dynamically typed and garbage-collected. It supports multiple programming paradigms, including structured (particularly procedural), object-oriented and functional programming. It is often described as a "batteries included" language due to its comprehensive standard library.

Python was conceived in the late 1980s by Guido van Rossum at Centrum Wiskunde & Informatica (CWI) in the Netherlands as a successor to the ABC programming language, which was inspired by SETL, capable of exception handling and interfacing with the Amoeba operating system. Its implementation began in December 1989. Van Rossum shouldered sole responsibility for the project, as the lead developer, until 12 July 2018, when he announced his "permanent vacation" from his responsibilities as Python's "benevolent dictator for life", a title the Python community bestowed upon him to reflect his long-term commitment as the project's chief decision-maker.



Fig 3.4. Python Logo

3.3.2. PYTHON LIBRARIES USED IN PROJECT

The Python Standard Library contains the exact syntax, semantics and tokens of python. It contains built-in modules that provide access to basic system functionality like I/O and some other core modules. Most of the python libraries are written in C programming language. Python Standard library plays a very important role. Without it, the programmers cannot have access to functionalities of python. The following are the libraries we have used:

- **Appconfig:** appconfig is a python module that simplifies the usage of ini based config files. It uses the Python ConfigParser module. The advantage of the appconfig module is that it allows defining default values for all config parameters and to provide a description of each parameter.
- **Django:** Django is a Python framework that makes it easier to create web sites using Python. Django takes care of the difficult stuff so that you can concentrate on building your web applications. Django emphasizes reusability of components, also referred to as DRY (Don't Repeat Yourself) and comes with ready-to-use features like login system, database connection and CRUD operations (Create Read Update Delete).
- **Django.shortcuts:** Django shortcuts module is a collection of helper functions that are generally used in view function/classes. There are many shortcuts available in module django.shortcuts.
- **Django.db.Q:** is an object used to encapsulate a collection of keyword arguments. These keyword arguments are specified as in Field lookups above. Q objects can be combined using the & and | operators. When an operator is used on two Q objects, it yields a new Q object.

- **Contrib:** The messages framework allows you to temporarily store messages in one request and retrieve them for display in a subsequent request (usually the next one). Every message is tagged with a specific level that determines its priority (e.g., info, warning, or error).
- **CASCADE:** will remove the child object when the foreign object is deleted.
- **SET_NULL:** will set the child object foreign key to null. **SET_DEFAULT:** will set the child object to the default data given while creating the model.
- **Datetime:** It is the combination between dates and times. The attributes of this class are similar to both date and separate classes. These attributes include day, month, year, minute, second, microsecond, hour, and tzinfo.
- **Openpyxl:** It is a Python library to read/write Excel 2010 xlsx/xlsm/xltx/xltm files. It was born from lack of existing library to read/write natively from Python the Office Open XML format. By default, openpyxl does not guard against quadratic blowup or billion laughs xml attacks. To guard against these attacks, install defusedxml.
- **Sys:** This module provides access to some variables used or maintained by the interpreter and to functions that interact strongly with the interpreter. The python sys module contains methods and variables for modifying many elements of the Python Runtime Environment. It allows us to access parameters and functionalities specific to the system.

3.4. DESIGN

Tools which assist in the design process are the UML Diagrams and Flowcharts.

3.4.1 UML DIAGRAMS

3.4.1.1. Architecture Diagram

The architectural diagram in Fig 3.5. is a visual representation that maps out the physical implementation for components of a software system. The main components of the system are the Admin, Service Provider and Remote User.

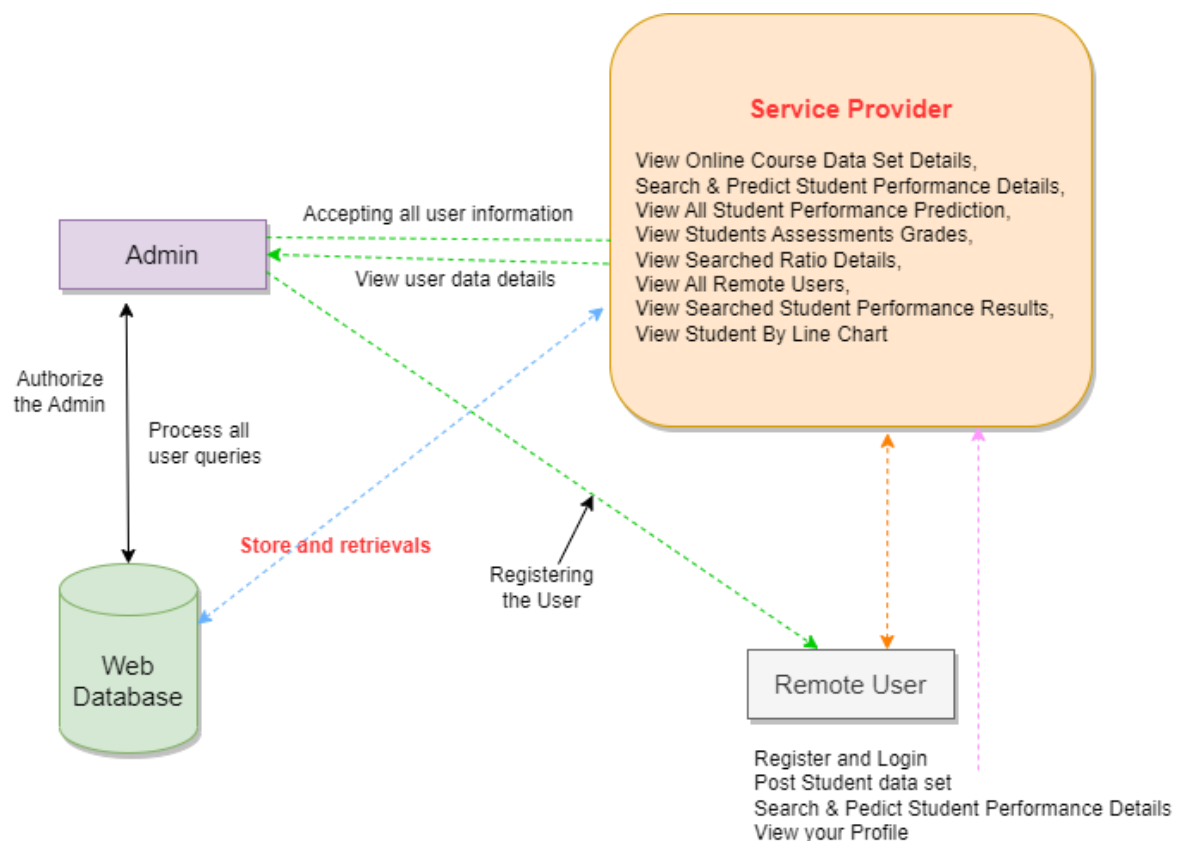


Fig 3.5. Architecture Diagram of the System

3.4.1.2. Class Diagram

The Class Diagram in Fig 3.6. describes the attributes and operations of a class and also the constraints imposed on the system.

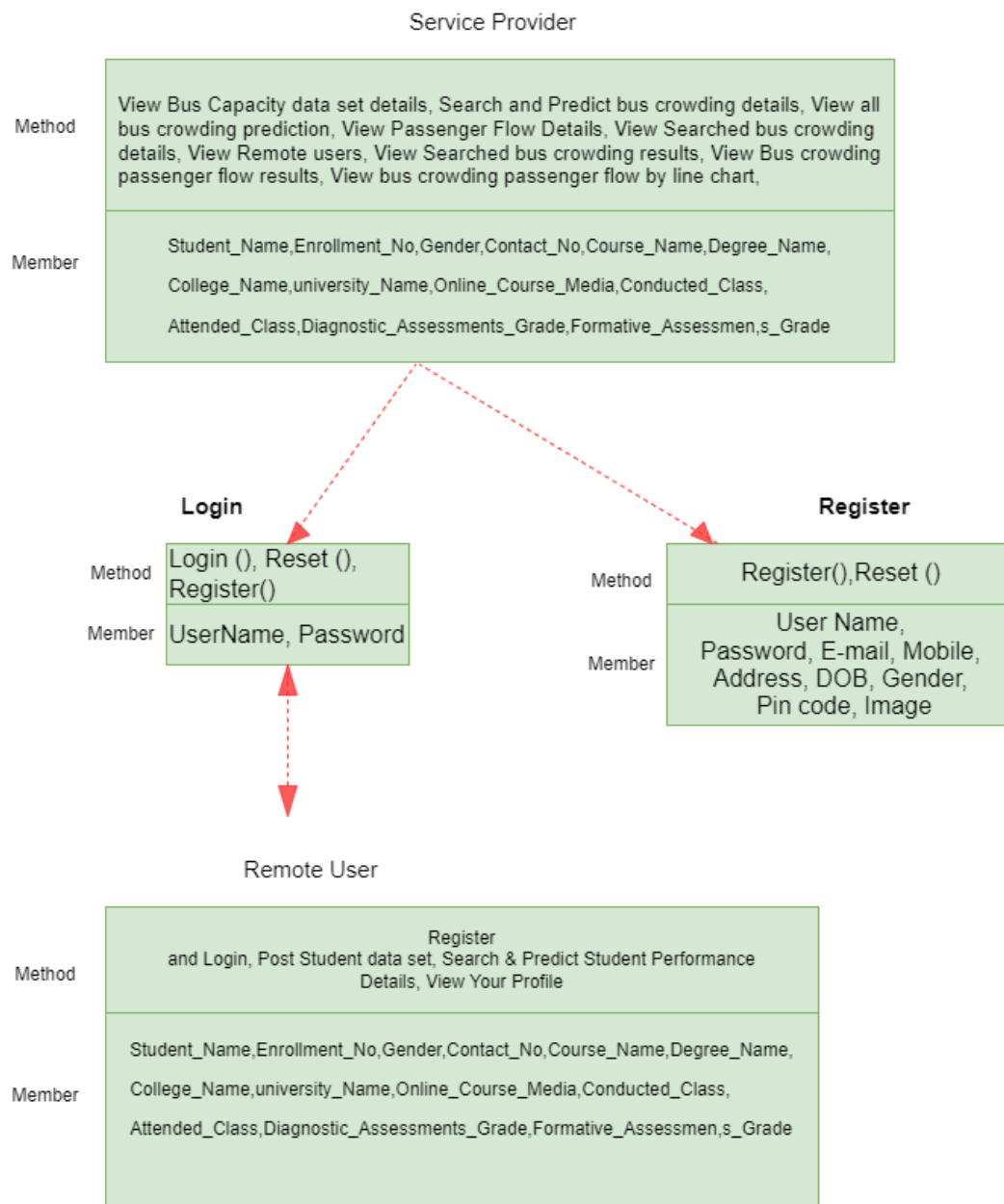


Fig 3.6. Class Diagram for the System

3.4.1.3. Data Flow Diagram

The Data Flow Diagram in Fig 3.7. maps out the flow of information for the system.

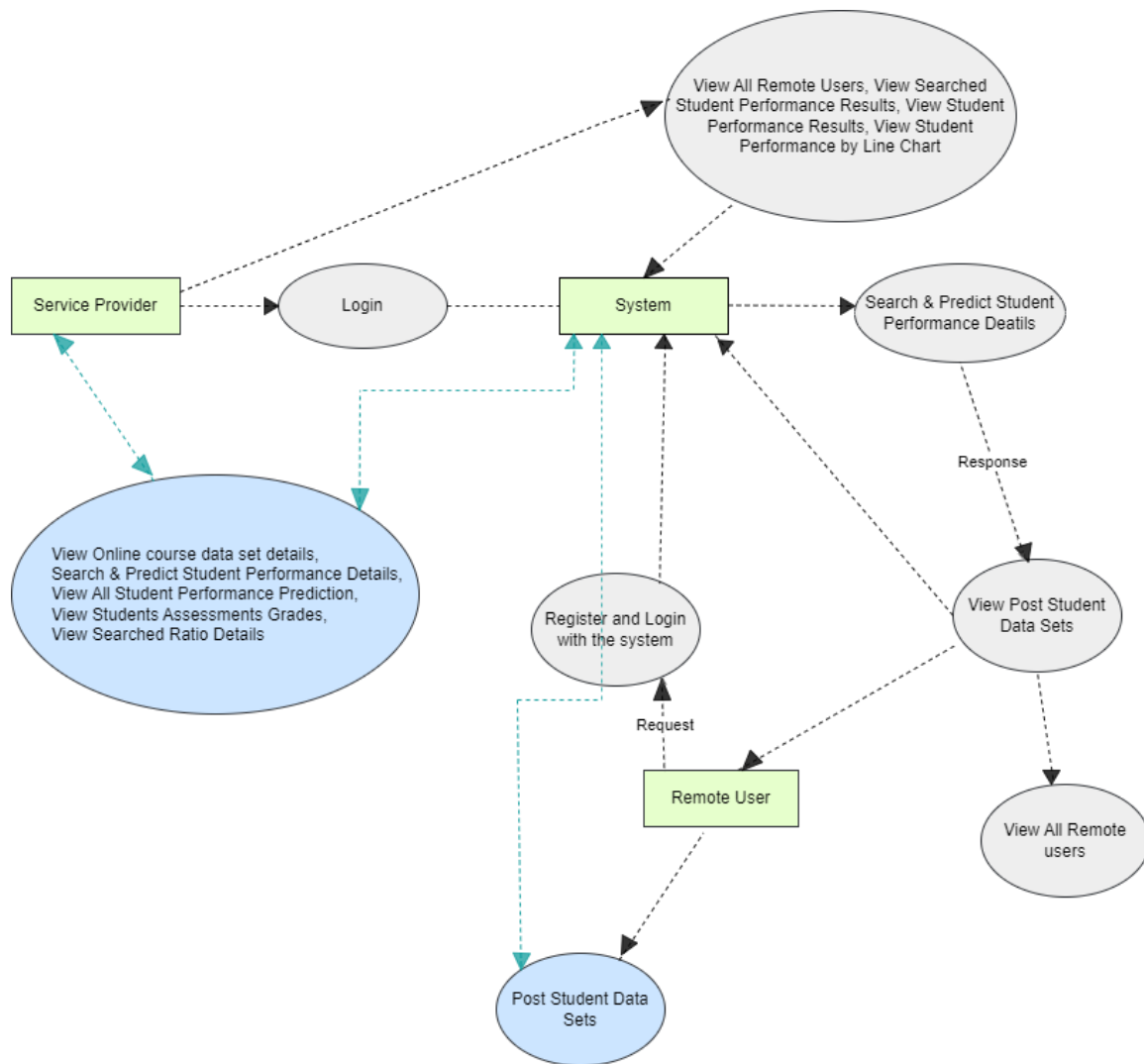


Fig 3.7. Data Flow Diagram for the System

3.4.1.4. Sequence Diagram

The Sequence Diagram in Fig 3.8. captures the interaction between objects in the context of a collaboration.

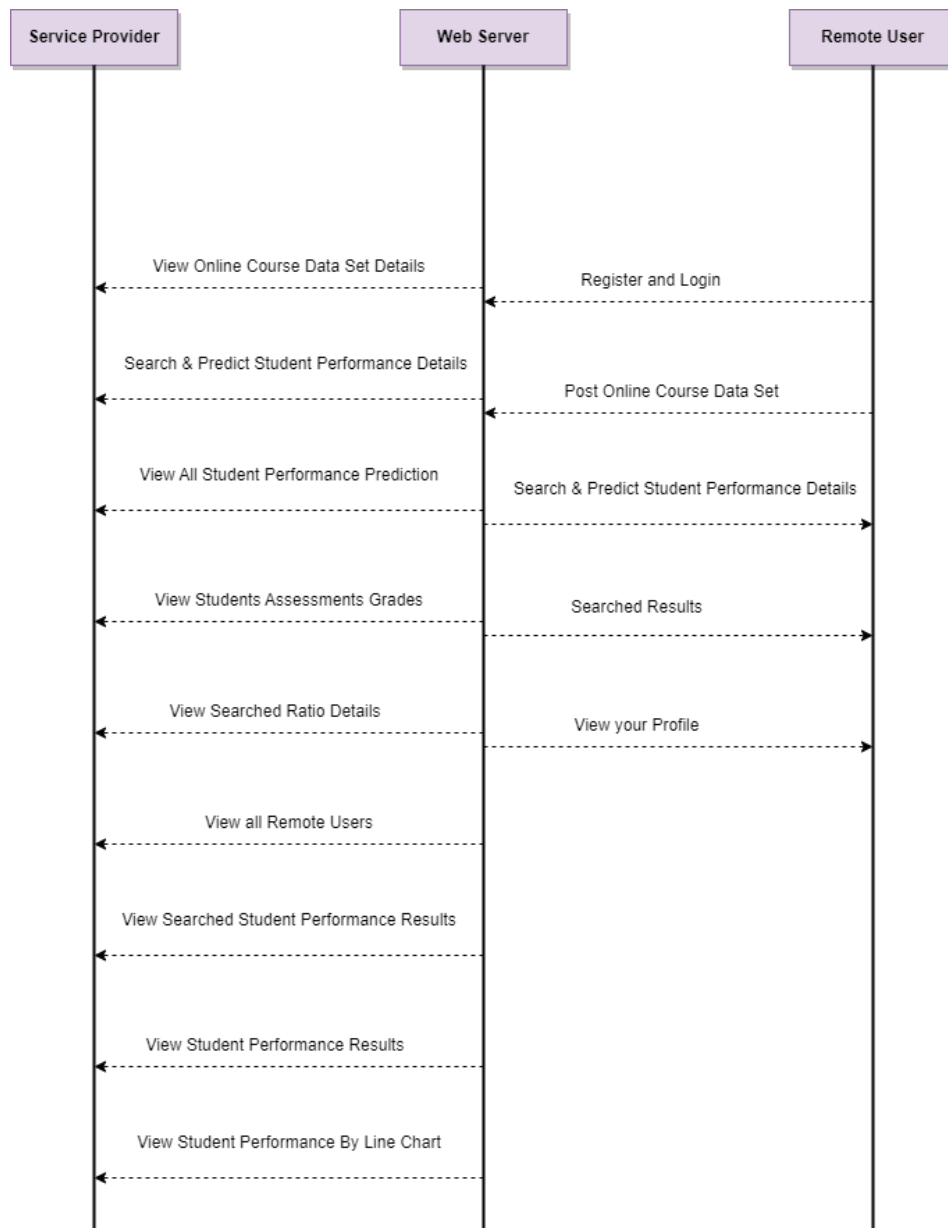


Fig 3.8. Sequence Diagram for the System

3.4.1.5. Use Case Diagram

The Use Case Diagram in Fig 3.9. graphically depicts the interactions among the elements of the system. The main actors of this Use Case Diagram are: Service Provider and Remote Users.

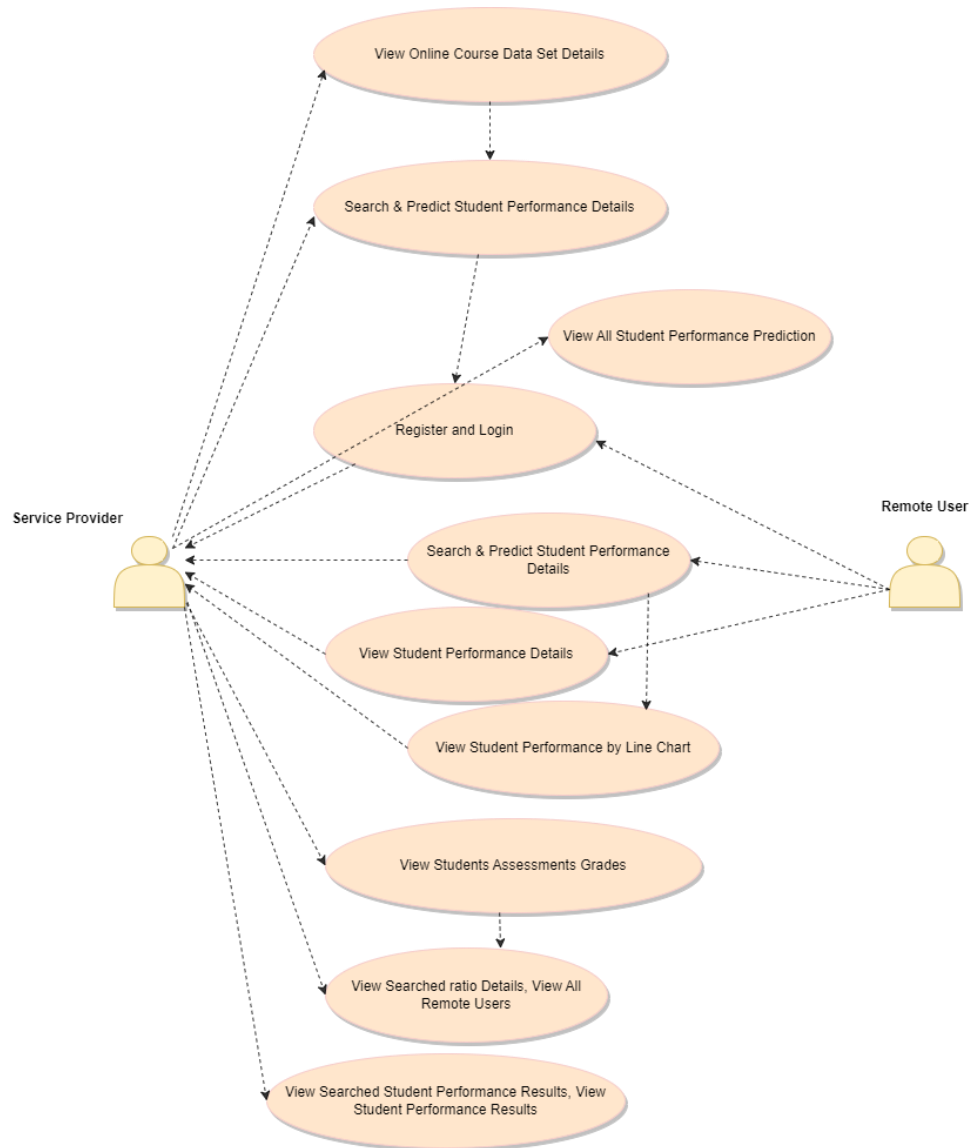


Fig 3.9. Use Case Diagram

3.4.2. FLOWCHARTS

3.4.2.1. Remote User Flowchart

The Fig 3.10. depicts the process of working of the Remote User Module of the system.

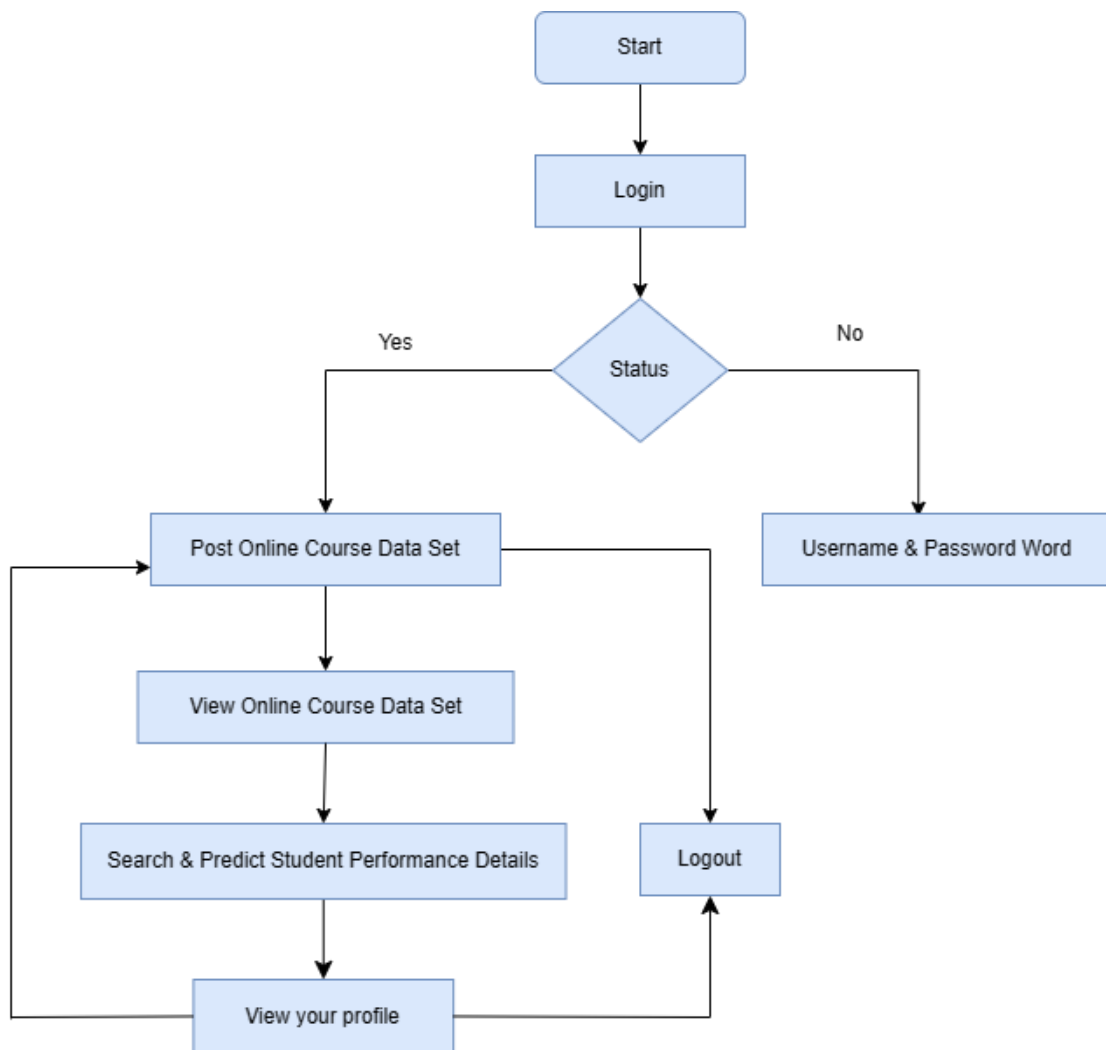


Fig 3.10. Flowchart for Remote User

3.4.2.2. Service Provider Flowchart

The Fig 3.11. depicts the process of working of the Service Provider Module of the system.

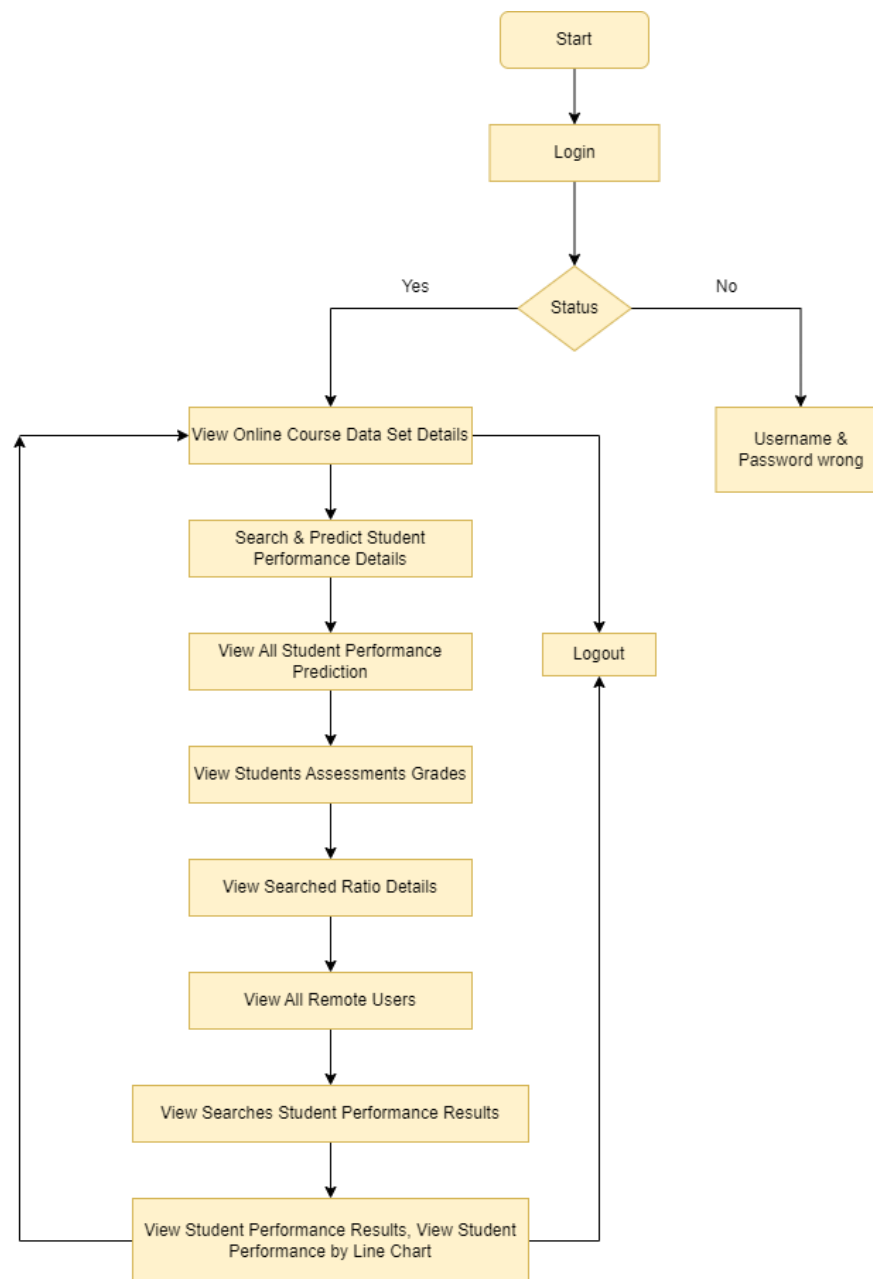


Fig 3.11. Service Provider Flowchart

3.5. STEPWISE IMPLEMENTATION AND CODE

3.5.1. IMPLEMENTATION

1. Open the WAMP Server.



Fig 3.12 Wamp Server

2. Login using the id “root”.



Fig 3.13. Login Page

3. Upload the database.

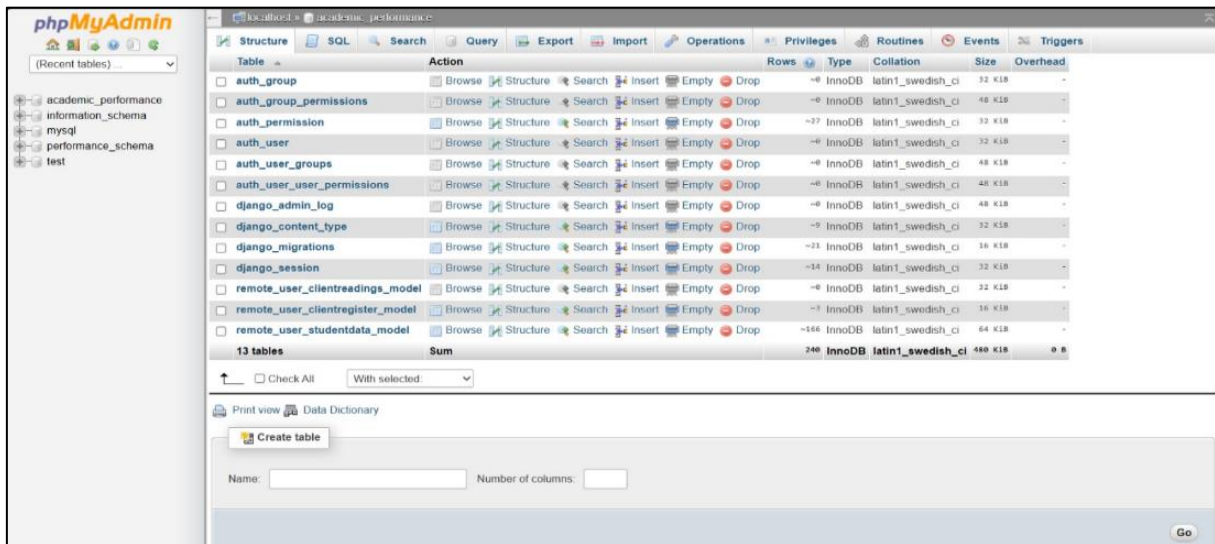


Fig 3.14. Database upload

4. Run the following command in command prompt: python manage.py runserver.

```
C:\Windows\System32\cmd.exe - python manage.py runserver
Microsoft Windows [Version 10.0.19045.2728]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Archana\Desktop\Original MP\Academic_Performance\academic_performance>python manage.py runserver
Performing system checks...

System check identified no issues (0 silenced).

You have 1 unapplied migration(s). Your project may not work properly until you apply the migrations for app(s): admin.
Run 'python manage.py migrate' to apply them.
April 12, 2023 - 12:29:03
Django version 2.1.7, using settings 'academic_performance.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
■
```

Fig 3.15. Command Prompt

5. Open the portal in a webpage by copy pasting.

Fig 3.16. User Login

6. There are 2 modules in the proposed system: Service Provider and Remote User.
 - a. Service Provider
 - In this module, the admin has to login by using valid user name and password: “Server”.
 - Viewing and Authorizing Users.

ID	Name	Gender	Age	Height	Weight	GPA	Academic Performance	Behavioral Pattern	Digital Campus	Machine Learning	Long Short-Term Memory	Academic Performance Prediction	Behavioral Pattern Prediction	Digital Campus Prediction
21H51A0501	Arjun	Female	18.0	5.2	56.0	Good	Normal	Normal	manageable	Normal	Yes	Good	2 times	No
21H51A0502	Roja	Female	18.0	5.8	50.0	Good	Normal	Normal	manageable	Normal	Yes	Good	1 times	No
21H51A0504	Kumar	Female	18.0	5.4	47.0	Good	Bad	Normal	Not Manageable	Poor	No	Good	2 times	No
21H51A0531	Amar	Female	18.0	5.2	48.0	Good	Normal	Normal	manageable	Normal	Yes	Good	2 times	No
21h51a0537	Firoz	Male	18.0	5.3	50.0	Good	Normal	Normal	Not Manageable	Normal	Average	Good	2 times	No
21H51A0578	Amalya	Female	18.0	5.1	58.0	Good	Bad	Normal	Not Manageable	Poor	No	Good	2 times	No
21H51A0590	Akila	Female	18.0	5.2	48.0	Good	Normal	Normal	manageable	Normal	Yes	Good	2 times	No
21H51A0592	Arokiyaraj	Female	18.0	5.2	56.0	Bad	Normal	Normal	Not	Poor	Yes	Good	2	No

Fig 3.17. Service Provider

b. User

- In this module, there are n numbers of users are present.
- User should register before performing any operations. Once user registers, their details will be stored to the database.
- After registration successful, he has to login by using authorized user name and password.
- Once Login is successful user can perform some operations like: browse student data sets, view all student record details, search student data set.
- Viewing Profile Details: In this module, the user can see their own profile details, such as their address, email, mobile number, profile Image.

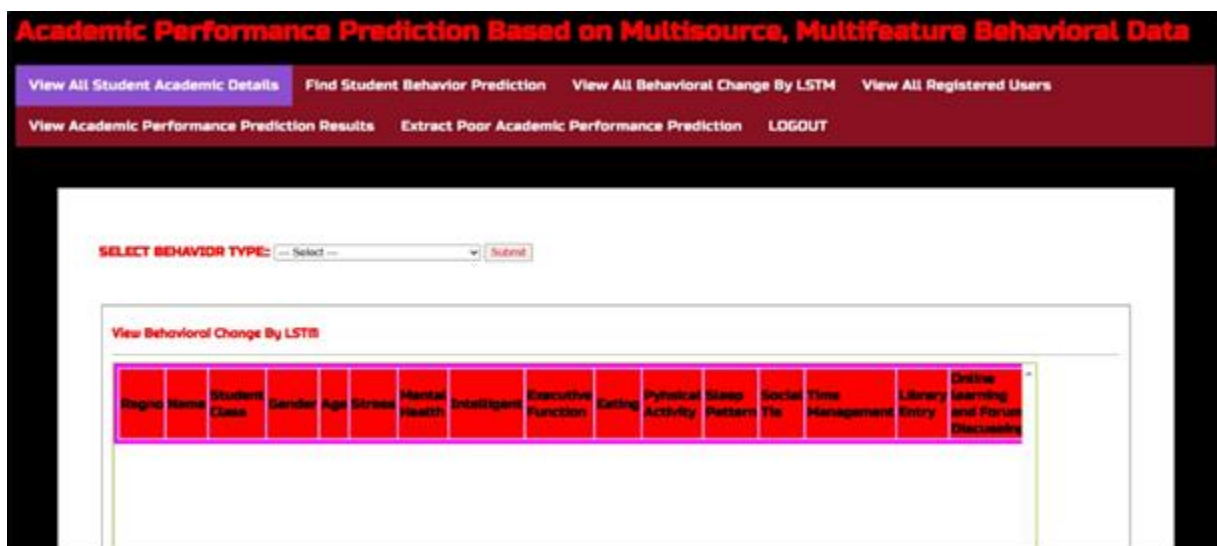


Fig 3.18. Remote User

3.5.2. CODE

3.5.2.1 Remote User

```
def login(request):  
  
    if request.method == "POST" and 'submit1' in request.POST:  
  
        username = request.POST.get('username')  
        password = request.POST.get('password')  
        try:  
            enter = ClientRegister_Model.objects.get(username=username, password=password)  
            request.session["userid"] = enter.id  
            return redirect('Add_Data_Sets')  
        except:  
            pass  
  
    return render(request, 'RUser/login.html')
```

Fig 3.19. Login to Remote User

```
def Register1(request):  
  
    if request.method == "POST":  
        username = request.POST.get('username')  
        email = request.POST.get('email')  
        password = request.POST.get('password')  
        phoneno = request.POST.get('phoneno')  
        country = request.POST.get('country')  
        state = request.POST.get('state')  
        city = request.POST.get('city')  
        ClientRegister_Model.objects.create(username=username, email=email, password=password, phoneno=phoneno,  
| | | | | | | | | | country=country, state=state, city=city)  
  
        return render(request, 'RUser/Register1.html')  
    else:  
        return render(request, 'RUser/Register1.html')  
  
def ViewYourProfile(request):  
    userid = request.session['userid']  
    obj = ClientRegister_Model.objects.get(id= userid)  
    return render(request, 'RUser/ViewYourProfile.html', {'object':obj})
```

Fig 3.20. Register & View Profile

```
def View_Student_Records(request):
    userid = request.session['userid']
    obj = studentdata_Model.objects.all()

    return render(request, 'RUser/View_Student_Records.html', {'list_objects': obj})

def Search_Student_Data(request):
    value = request.POST.get('regno')
    obj1 = studentdata_Model.objects.all().filter(Q(Regno=value)| Q(names=value))

    return render(request, 'RUser/Search_Student_Data.html', {'list_objects': obj1})
```

Fig 3.21. View Student Records & Search Student Data

3.5.2.2. Service Provider

```
def collegeserverlogin(request):
    if request.method == "POST":
        admin = request.POST.get('admin')
        password = request.POST.get('password')
        if admin == "Server" and password == "Server":
            return redirect('viewallclients')

    return render(request, 'TServer/collegeserverlogin.html')

def Find_Student_Behavior_Prediction(request):
    regno = request.POST.get('regno')
    obj1 = studentdata_Model.objects.all().filter(Regno=regno)
    return render(request, 'TServer/Find_Student_Behavior_Prediction.html', {'objects': obj1})
```

Fig 3.22. College Server login & Performance Prediction

```
def View_All_Behavioral_Change_ByLSTM(request):
    obj1=''
    btype= request.POST.get('btype')
    if btype=="Stress" :
        obj1 = studentdata_Model.objects.values('Regno', 'names', 'Stu_Class', 'Gender', 'Age', 'Stress')
    elif btype=="Mental Health" :
        obj1 = studentdata_Model.objects.values('Regno', 'names', 'Stu_Class', 'Gender', 'Age', 'Mental_Health')
    elif btype == "Intelligent":
        obj1 = studentdata_Model.objects.values('Regno', 'names', 'Stu_Class', 'Gender', 'Age', 'Intelligent')
    elif btype == "Executive Function":
        obj1 = studentdata_Model.objects.values('Regno', 'names', 'Stu_Class', 'Gender', 'Age', 'Executive_fun')
    elif btype == "Eating":
        obj1 = studentdata_Model.objects.values('Regno', 'names', 'Stu_Class', 'Gender', 'Age', 'Eating')
    elif btype == "Pyhsical Activity":
        obj1 = studentdata_Model.objects.values('Regno', 'names', 'Stu_Class', 'Gender', 'Age', 'Pyhsical_Activity')
    elif btype == "Sleep Pattern":
        obj1 = studentdata_Model.objects.values('Regno', 'names', 'Stu_Class', 'Gender', 'Age', 'Sleep_pattern')
    elif btype == "Social Tie":
        obj1 = studentdata_Model.objects.values('Regno', 'names', 'Stu_Class', 'Gender', 'Age', 'Social_Tie')
    elif btype == "Time Management":
        obj1 = studentdata_Model.objects.values('Regno', 'names', 'Stu_Class', 'Gender', 'Age', 'Time_Management')
    elif btype == "Time Management":
        obj1 = studentdata_Model.objects.values('Regno', 'names', 'Stu_Class', 'Gender', 'Age', 'Time_Management')
    elif btype == "Library Entry":
        obj1 = studentdata_Model.objects.values('Regno', 'names', 'Stu_Class', 'Gender', 'Age', 'Library_Entry')
    elif btype == "Online learning and Forum Discueeing":
        obj1 = studentdata_Model.objects.values('Regno', 'names', 'Stu_Class', 'Gender', 'Age', 'Online_learning')
```

Fig 3.23. View Behavior Changes by LSTM


```
def dislikeschart(request,dislike_chart):
    charts = studentdata_Model.objects.values('names').annotate(dcount=Avg('dislikes'))
    return render(request,"TServer/dislikeschart.html", {'form':charts, 'dislike_chart':dislike_chart})

def View_College_Dataset_Details(request):
    chart = studentdata_Model.objects.all()
    return render(request,'TServer/View_College_Dataset_Details.html',{'objects':chart})

def Extract_PoorAcademic_PerformancePrediction(request):
    objs = studentdata_Model.objects.all().filter(Q(Stress='Not Manageable'), Q(Mental_Health='Poor') | Q(Pyhsical_Activity='No'))
    return render(request,'TServer/Extract_PoorAcademic_PerformancePrediction.html',{'objects':objs})
```

Fig 3.24. Extraction of Poor Performing Students

CHAPTER 4

RESULTS AND DISCUSSION

CHAPTER 4

RESULTS AND DISCUSSION

4.1. COMPARISON OF EXISTING SOLUTIONS

Table 4.1. Performance Measures of Dataset

Model	TP	FP	TN	FN
LR	402	110	376	156
Naïve Bayes	356	149	337	201
Decision Tree	434	123	363	123

- From Table 3.1, TP, True Positives, is the number of data rows in the test set which had a positive target and that were predicted to have a positive target. TN, True Negatives, is the number of data rows in the test set that had a negative target and that were predicted to have a negative target. FP, False Positives, is the number of data rows in the test set which had a negative target but that were predicted to have positive target. FN, False Negative, is the number of data rows in the test set that had a positive target but that were predicted to have a negative target.

- The ROC index, the area under the curve, performance measure has been used to evaluate the performance by using the prediction scores. [9]

$$ROC\ index = \sum_{i=2}^{|T|} (FPR(T[i]) - FPR(T[i - 1])) \times (TPR(T[i]) + TPR(T[i - 1])) / 2 \quad (3)$$

- Where $|T|$ represents the number of thresholds that are used, $FPR(T[i])$ represents the false positive rate at the threshold i , and $TPR(T[i])$ represents the true positive rate at the threshold i . A larger ROC index indicates a better classification model. A model with ROC index above 0.7 considered a strong model while a model with ROC index below 0.6 considered a weak model.

Table 4.2. ROC Index

Model	ROC index
LR	0.767
Naïve Bayes	0.697
Decision Tree	0.762

4.2. DATA COLLECTION AND PERFORMANCE METRICS

4.2.1. DATASET

- The dataset for our model was inspired from the UCI Machine Learning Repository [5] | Student Performance Data Set.
- This data contains 166 instances (students).

Table 4.3. Dataset Information

Data Set Characteristics:	Multivariate	Number of Instances:	166	Area:	Social
Attribute Characteristics:	Integer	Number of Attributes:	22	Missing Values:	N/A

- The data attributes include student health, fitness, demographic, social and school related features and it was collected by using school reports and questionnaires.
- The dataset contains real-life instances collected from students pursuing II, III and IV years respectively at our college.

4.2.2. ATTRIBUTE INFORMATION**Table 4.4 Attributes and Domain values**

ATTRIBUTE	DOMAIN
Name	Alphabets
Id	Alpha numeric
Gender	Binary: 'F' – female or 'M' – male
Age	Numeric: from 15 to 22
Height	Numeric: in feet
Weight	Numeric: in kgs
Physical Fitness	Nominal: 'Good', 'Bad' or 'Normal'
Cardio Fitness	Nominal: 'Good', 'Bad' or 'Normal'
Aerobic Fitness	Nominal: 'Good', 'Bad' or 'Normal'
Stress <i>Student's perception of stress</i>	Nominal: 'Manageable' and 'Not Manageable'
Mental Health	Nominal: 'Normal', 'Poor'
Intelligence <i>Student's IQ levels and General Knowledge</i>	Nominal: 'Yes', 'No' or 'Average'
Activities <i>Engagement in co-scholastic activities</i>	Nominal: 'Good', 'Bad' or 'Normal'
Eating <i>No. of full meals a day</i>	Numeric: from 2 to 4
Physical Activity	Binary: 'Yes' or 'No'
Sleep Pattern	Nominal: 'Correct Time' or 'Late Night'
Social Tie Connections in-and-out of college	Nominal: 'Good', 'Bad' or 'Normal'
Time Management	Nominal: 'Perfect' or 'Poor'

4.2.3. PERFORMANCE METRICS

Table 3.5 Shows the Performance Metrics of the Proposed Model on the Student Dataset discussed earlier.

Table 4.5 Performance Metrics of Proposed Model

Model	Accuracy	Precision	Recall	F measure	ROC index
Random Forest	81.25	80.25	86.30	83.25	0.812

Table 4.6 Performance Metrics of Existing Models

Model	Accuracy	Precision	Recall	F measure	ROC index
LR	74.53	79.23	71.91	74.87	0.760
Naïve Bayes	66.52	70.51	64.27	67.21	0.697
Decision Tree	76.93	77.96	77.83	77.88	0.762

The performance metrics have been calculated from Table 3.1. using the following formulae[6]:

$$Accuracy = \frac{TN + TP}{TN + FP + TP + FN} \quad (4)$$

$$Precision = \frac{TP}{TP + FP} \quad (5)$$

$$Recall = \frac{TP}{TP + FN} \quad (6)$$

$$F1 \text{ Score} = 2 * \frac{Precision * Recall}{Precision + Recall} \quad (7)$$

From the Performance Metrics, it is visible that Naïve Bayes Classifier gives the least accuracy and ROC index. Whereas Decision Tree Classifier gives comparatively better accuracy and ROC index. However, final results show that random forest algorithm is the better one.

4.3. RESULTS

The screenshot displays a web interface for a system titled "Academic Performance Prediction Based on Multisource, Multifeature Behavioral Data". Below the title is a subtitle: "academic performance prediction, behavioral pattern, digital campus, machine learning (ML), long short-term memory (LSTM)". The interface features a login section with the heading "LOGIN USING YOUR ACCOUNT:". It includes two input fields for "User Name" and "Password", followed by a "sign_in" button. Below this, there is another "LOGIN USING YOUR ACCOUNT:" heading, and two buttons labeled "COLLEGE SERVER" and "REGISTER".

Fig 4.2. Remote User Login Page

The Remote User Login page in Fig. 4.2, is for every individual user to login to their profile or register themselves. Also, they can look up student performance information, check their individual profiles, and anticipate it.

Academic Performance Prediction Based on Multisource, Multifeature Behavioral Data

academic performance prediction, behavioral pattern, digital campus, machine learning (ML), long short-term memory (LSTM). .

College Server Login:

User Name

Password

CS Login

User Login

Fig 4.3. Service Provider Login Page

The login page for the service provider in Fig 4.3., the administrator must login from this page using a valid username and password. Here, the service provider uses all user information and authorizes each user for login rights. The service provider has access to user information such name, address, email address, etc.

Academic Performance Prediction Based on Multisource, Multifeature Behavioral Data

View All Student Academic Details Find Student Behavior Prediction View All Behavioral Change By LSTM View All Registered Users

View Academic Performance Prediction Results Extract Poor Academic Performance Prediction LOGOUT

VIEW ALL STUDENTS ACADMIC DETAILS III

Z1H5IA0501	Arjun	Female	18.0	5.2	56.0	Good	Normal	Normal	manageable	Normal	Yes	Good	2 times	No
Z1H5IA0502	Roja	Female	18.0	5.8	50.0	Good	Normal	Normal	manageable	Normal	Yes	Good	1 times	No
Z1H5IA0504	Kumar	Female	18.0	5.4	47.0	Good	Bad	Normal	Not Manageable	Poor	No	Good	2 times	No
Z1H5IA0531	Amar	Female	18.0	5.2	48.0	Good	Normal	Normal	manageable	Normal	Yes	Good	2 times	No
Z1H5IA0537	Firoz	Male	18.0	5.3	50.0	Good	Normal	Normal	Not Manageable	Normal	Average	Good	2 times	No
Z1H5IA0578	Amalya	Female	18.0	5.1	58.0	Good	Bad	Normal	Not Manageable	Poor	No	Good	2 times	No
Z1H5IA0590	Akila	Female	18.0	5.2	48.0	Good	Normal	Normal	manageable	Normal	Yes	Good	2 times	No
Z1H5IA0592	Arokiyaraj	Female	18.0	5.2	56.0	Bad	Normal	Normal	Not	Poor	Yes	Good	2	No

Fig 4.4. Service Provider: View All Student Data

Fig 4.4. illustrates one of the several privileges granted to the service provider, namely the ability to examine all student data. The student roll number, name, email address, mental health status, time management, etc. are all contained in this data.

Academic Performance Prediction Based on Multisource, Multifeature Behavioral Data

View All Student Academic Details Find Student Behavior Prediction View All Behavioral Change By LSTM View All Registered Users

View Academic Performance Prediction Results Extract Poor Academic Performance Prediction LOGOUT

VIEW ALL STUDENTS BEHAVIOR PREDICTIONS !!!

ENTER STUDENT REGISTER NUMBER : [View All Behaviours](#)

Regno	Name	Student Class	Gender	Age	Stress	Mental Health	Intelligent	Executive Fun	Eating	Pyhsical Activity	Sleep Pattern	Social Tie	Time Management	Library Entry	Online learning and Forum Discueeing

Fig 4.5. Service Provider: Search Student Data

The above Fig 4.5. shows how a service provider is allowed to search and view individual student data. Here, an overview of how a particular student performs in his/her academics and behavioral changes if any. In the field “Enter Student Registered number”, if we enter any of the student’s roll number provided should be available in the dataset fed as input, an overview of that student’s individual performance is displayed.

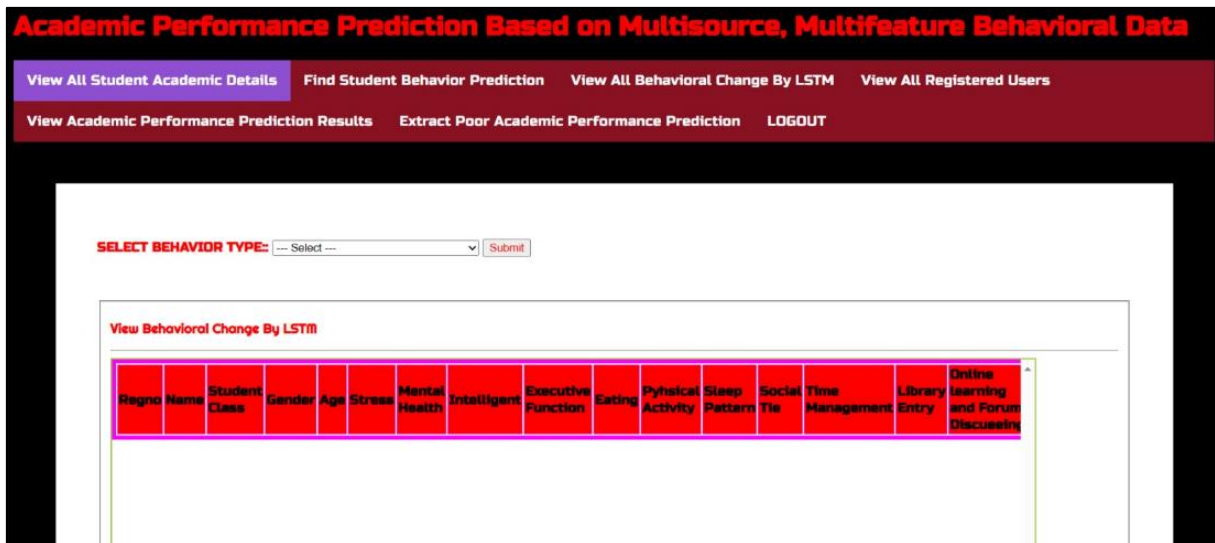


Fig 4.6. Service Provider: View Behavior Patterns

Fig.4.5 above illustrates how a service provider may see pupils in relation to a specific behavioral feature. For instance, information about the pupils who fit into that behavior category is displayed if we choose a certain behavior from the drop-down menu.

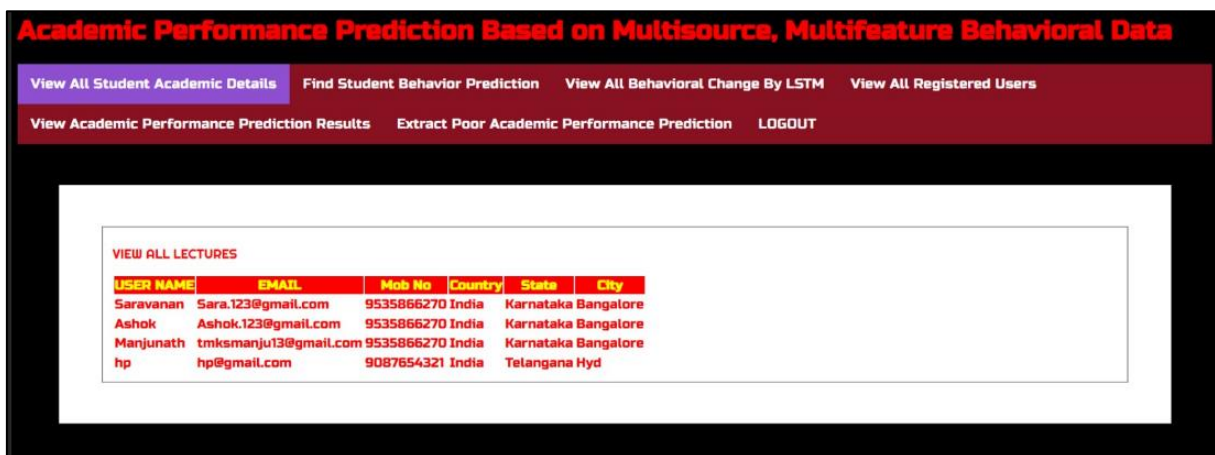


Fig 4.7. Service Provider: View Registered Users

The registered users are displayed in Fig. 4.7 above. It includes Username, Email, Mobile, Nation, State, and City information. For instance, after selecting "View Registered Users" from the drop-down menu that displays, the registered user's details are shown.

Academic Performance Prediction Based on Multisource, Multifeature Behavioral Data

View All Student Academic Details Find Student Behavior Prediction View All Behavioral Change By LSTM View All Registered Users

View Academic Performance Prediction Results Extract Poor Academic Performance Prediction LOGOUT

VIEW POOR STUDENTS BEHAVIOR PREDICTIONS(Based On Any two Parameters) !!!

Regno	Name	Student Class	Gender	Age	Stress	Mental Health	Intelligent	Executive Fun	Eating	Physical Activity	Sleep Pattern	Social Tie	Time Management	Library Entry	Online Learning and Forum Discussing
BE203881	Arokiyaraj	BE	Male	19.0	Not Manageable	Poor	Yes	Good	3 times	Yes	Late Night	Good	Poor	Yes	Yes
BE203891	Suman	BE	Male	19.0	Not Manageable	Poor	Yes	Good	3 times	Yes	Late Night	Good	Poor	Yes	Yes
21H5IA0592	Arokiyaraj	BTech	Female	18.0	Not Manageable	Poor	Yes	Good	2 times	No	Late Night	Bad	Poor	No	Yes
21H5IA0546	Suman	bTech	Female	19.0	Not Manageable	Poor	Yes	Good	3 times	Yes	Late Night	Bad	Poor	No	Yes
21H5IA05CD	Arokiyaraj	BTech	Female	19.0	Not Manageable	Poor	Yes	Good	1 times	Yes	Late Night	Bad	Poor	Yes	Yes
21H5IA05G5	Suman	bTech	Male	19.0	Not Manageable	Poor	Yes	Good	1 times	No	Late Night	Bad	Poor	No	Yes

Fig 4.8. Service Provider: View Poor Performing Students

The information about the pupils who performed poorly is shown in fig. 4.8 above. By forecasting the behavioral data, we can use this to view the details of the poor students.

Academic Performance Prediction Based on Multisource, Multifeature Behavioral Data

BROWSE STUDENT DATA SETS VIEW ALL STUDENT RECORD DETAILS SEARCH STUDENT DATA SET VIEW YOUR PROFILE LOGOUT

Browse Student Data Set No file chosen

Regno	Name	Gender	Age	Height	Weight	Physical_Fitness	Cardio_Fitness	Aerobic_Fitness	Stress	Mental_Health	Intelligent	Executive_fun	Eating	Physical_Activity	Si
-------	------	--------	-----	--------	--------	------------------	----------------	-----------------	--------	---------------	-------------	---------------	--------	-------------------	----

Fig 4.9. Remote User: Upload Dataset

The page where the data set was uploaded and used to forecast the performance of the students is depicted in Fig. 4.8 above.

CHAPTER 5

CONCLUSION

CHAPTER 5

CONCLUSION

5.1. CONCLUSION AND FUTURE ENHANCEMENT

Academic performance prediction has drawn a lot of attention from scholars since it is a crucial topic in the field of education data mining. Yet, there are still many difficulties with prediction accuracy and interpretability because to the lack of richness and diversity in both data sources and features. Our study aims to get a thorough understanding of student behavioural patterns in order to enable students better understand how to interact with the university. This will help to initially relieve the problem. A comprehensive profile of a student is created using multifeatured data, and behavioural change is assessed using linear, nonlinear, and deep learning (LSTM) techniques, providing a systematic view of students' behavioural patterns.

In conclusion, our study is based on a complete passive daily data capture system that exists in most modern universities. This system can potentially lead to continual investigations on a larger scale. The knowledge obtained in this study can also potentially contribute to related research among students.

In future works, more studies and experiments are needed to increase the accuracy of prediction in the first days of course duration with additional preprocessing methods like trying to make the dataset balanced using under-sampling or oversampling techniques. Furthermore, study the effect of the functional model instead of the sequential model design, using assessment data on predicting student performance, assigning weights for each class based on the class importance, and aggregating clickstream data periodically on the accuracy of predicting student's performance.

REFERENCES

REFERENCES

- [1].A. Furnham, and J. Monsen, "Personality traits and intelligence predict academic school grades," Learning and Individual Differences, vol. 19, no. 1, pp. 0-33, 2009.
- [2].M. A. Conard, "Aptitude is not enough: How personality and behavior predict academic performance," Journal of Research in Personality, vol. 40, no. 3, pp. 339-346, 2006.
- [3].D. G. Kleinbaum and M. Klein, Logistic Regression a Self-Learning Text, 3rd ed. New York: Springer-Verlag New York, 2010.
- [4].J. Han, M. Kamber, and J. Pei, Data Mining: Concepts and Techniques, 3rd ed. Morgan Kaufmann publications, 2012.
- [5].J. D. Kelleher, B. Mac Name, and A. D'Arcy, Fundamentals of Machine Learning for Predictive Data Analytics. Algorithms, Worked Examples, and Case Studies. 2015.
- [6].<https://medium.com/@nilimakhanna1/j48-classification-c4-5-algorithm-in-a-nutshell-24c50d20658e>
- [7].<https://intellipaat.com/blog/what-is-lstm/#:~:text=LSTM%20Explained,-Now%2C%20let's%20understand&text=First%2C%20you%20must%20be%20wondering,especially%20in%20sequence%20prediction%20problems>.
- [8].<https://www.python.org/>
- [9].P. Cortez and A. Silva. Using Data Mining to Predict Secondary School Student Performance. In A. Brito and J. Teixeira Eds., Proceedings of 5th Future Business Technology Conference (FUBUTEC 2008) pp. 5-12, Porto, Portugal, April, 2008, EUROSIS, ISBN 978-9077381-39-7
- [10].<https://medium.com/analytics-vidhya/confusion-matrix-accuracy-precision-recall-f1-score-ade299cf63cd>

GitHub Link

Click [here](#).