MINI PROJECT REPORT

On

Student Performance Prediction

Submitted by

Akshay Dubey (171500029)
Himanshu Pathak (171500136)
Ganesh Pandey(171500106)
Saurabh Singhal(171500304)

Department of Computer Engineering & Applications

Institute of Engineering & Technology

GLA University
Mathura,U.P.-281406, INDIA

2019

Department of computer Engineering and Applications GLA University, Mathura
17 km. Stone NH#2, Mathura-Delhi Road, P.O. – Chaumuha, Mathura – 281406

Declaration

I hereby declare that the work which is being presented in the Mini Project "**Student Performance Prediction",** in fulfillment of the requirements for Mini Project viva voce, is an authentic record of our own work carried under the supervision of Assistant Professor "**Mr. Vinay Agrawal**".

Team Members: Akshay Dubey (171500029)

Himanshu Pathak (171500136)

Ganesh Pandey (171500106)

Saurabh Singhal (171500304)

Course: B.tech. (Computer Science and Engineering)

Year: 3rd

Semester: 6th

Acknowledgement

We have taken efforts in this project "**Student Performance Prediction**". However, it would not have been possible without the kind support and help of many individuals and organizations. We would like to extend our sincere thanks to all of them.

We are highly indebted to **Mr. Vinay Agrawal** for their guidance and constant supervision as well as for providing necessary information regarding the project & also for their support in completing the project.

We would like to express our special gratitude and thanks to industry persons for giving me such attention and time.

Our thanks and appreciations also go to my colleague in developing the project and people who have willingly helped us out with their abilities.

Abstract

Prediction of student's performance became an urgent desire in most of educational entities and institutes. That is essential in order to help at-risk students and assure their retention, providing the excellent learning resources and experience, and improving the university's ranking and reputation. However, that might be difficult to be achieved for startup to mid-sized universities, especially those which are specialized in graduate and post graduate programs, and have small students' records for analysis.

So, the main aim of this project is to prove the possibility of training and modeling a small dataset size and the feasibility of creating a prediction model with credible accuracy rate. This research explores as well the possibility of identifying the key indicators in the small dataset, which will be utilized in creating the prediction model, using machine learning algorithms. Best indicators were fed into multiple machine learning algorithms to evaluate them for the most accurate model. Among the selected algorithms, the results proved the ability of clustering algorithm in identifying key indicators in small datasets.

The main outcomes of this study have proved the efficiency of algorithms in training small dataset size and in producing an acceptable classification's accuracy and reliability test rates.

Contents

Chapter 1

Introduction

Introduction to Student Performance Prediction

Students performance is an essential part in higher learning institutions. This is because one of the criteria for a high quality university is based on its excellent record of academic achievements. Generally, most of higher learning institutions used the final grades to evaluate students performance. Final grades are based on course structure, assessment mark, final exam score and also extracurricular activities. The evaluation is important to maintain students performances and the effectiveness of learning process.

By analyzing students performance, a strategic program can be well planned during their period of studies in an institution.

As a result, it would assist the educators in providing an effective teaching approach. Besides, educators could also monitor their students achievements. Students could improve their learning activities, allowing the administration to improve the systems performance.

The results show that it is possible to predict the graduation performance in 4th year at university using only pre-university marks and marks of 1st and 2nd year courses, no socio-economic or demographic features, with a reasonable accuracy. Furthermore courses that are indicators of particularly good or poor performance have been identified.

It will act as an aid for the universities or institutions as universities gather large volumes of data with reference to their students in electronic form. The advances in the data mining field make it possible to mine these educational data and find information that allow for innovative ways of supporting both teachers and students. It will help students to improve their overall grade and also the university or institutions to improve their overall result.

Reason for selecting the topic

To help Teachers and Students both in which area they are lagging. From the point of teachers it will help them by showing them the students to whom they should pay more attention. For students it will show them the subjects in which they are lagging or they should apply more efforts in that field.

Pre-requisites

Hands-on knowledge of machine learning as well as machine learning algorithms and python before working on Student Performance Prediction. Make sure that you have the complete dataset before implementing it.

Chapter 2

Software Requirement

Introduction to Python

Python is a popular programming language. It was created by Guido van Rossum, and released in 1991.Python works on different platforms (Windows, Mac, Linux, Raspberry Pi, etc).It is used for web development, software development,system scripting.

Advantages of Python :

1. Presence of third-party modules
2. Extensive support libraries
3. Open source

Applications of Python :

1. GUI based desktop applications(Games, Scientific Applications)
2. Web frameworks and applications
3. Enterprise and Business applications
4. Language Development

Introduction to NumPy

NumPy is a python library used for working with arrays.It also has functions for working in domain of linear algebra, fourier transform, and matrices.

NumPy was created in 2005 by Travis Oliphant. It is an open source project and you can use it freely. NumPy stands for Numerical Python.

Introduction to Pandas

Pandas is an open-source Python Library providing high-performance data manipulation and analysis tool using its powerful data structures. The name Pandas is derived from the word Panel Data – an Econometrics from Multidimensional data.

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.

Introduction to PIL

Python Imaging Library (abbreviated as **PIL**) (in newer versions known as Pillow) is a free library for the Python programming language that adds support for opening, manipulating, and saving many different image file formats. It is available for Windows, Mac OS X and Linux. The latest version of PIL is 1.1.7, was released in September 2009 and supports Python 1.5.2–2.7, with Python 3 support to be released "later".

Introduction to ImageTk

The ImageTk module contains support to create and modify Tkinter BitmapImage and PhotoImage objects from PIL images.

Introduction to Tkinter

Python offers multiple options for developing GUI (Graphical User Interface). Out of all the GUI methods, tkinter is most commonly used method. It is a standard Python interface to the Tk GUI toolkit shipped with Python. Python with tkinter outputs the fastest and easiest way to create the GUI applications. Creating a GUI using tkinter is an easy task.

To create a tkinter:

1. Importing the module – tkinter
2. Create the main window (container)
3. Add any number of widgets to the main window
4. Apply the event Trigger on the widgets.

There are two main methods used you the user need to remember while creating the Python application with GUI:-

1. Tk(): To create a main window, tkinter offers a method 'Tk()'. To change the name of the window, you can change the className to the desired one. The basic code used to create the main window of the application is:

   m=tkinter.Tk() where m is the name of the main window object

2. mainloop(): There is a method known by the name mainloop() is used when you are ready for the application to run. mainloop() is an infinite loop used to run the application, wait for an event to occur and process the event till the window is not closed.

   m.mainloop()

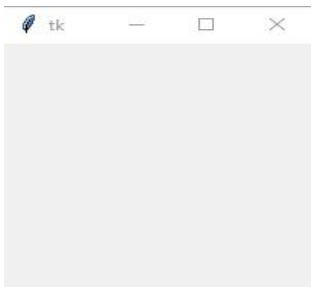


Fig 2.1 Window appears on executing mainloop()

There are mainly three geometry manager classes class:-

1. **pack() method**:It organizes the widgets in blocks before placing in the parent widget.
2. **grid() method**:It organizes the widgets in grid (table-like structure) before placing in the parent widget.
3. **place() method**:It organizes the widgets by placing them on specific positions directed by the programmer.

There are a number of widgets which you can put in your tkinter application. Some of the major widgets are explained below:

1. **Button**:To add a button in your application, this widget is used.
   The general syntax is:
   w=Button(master, option=value)
   master is the parameter used to represent the parent window.

Parameters:

1. activebackground: to set the background color when button is under the cursor.
2. activeforeground: to set the foreground color when button is under the cursor.
3. bg: to set he normal background color.
4. command: to call a function.
5. font: to set the font on the button label.
6. image: to set the image on the button.
7. width: to set the width of the button.
8. height: to set the height of the button.

2. **Entry**:It is used to input the single line text entry from the user.. For multi-line text input, Text widget is used.The general syntax is:
   w=Entry(master, option=value)

Parameters:

1. bd: to set the border width in pixels.
2. bg: to set the normal background color.
3. cursor: to set the cursor used.
4. width: to set the width of the button.
5. height: to set the height of the button.

3. **Label**: It refers to the display box where you can put any text or image which can be updated any time as per the code.The general syntax is: w=Label(master, option=value)

Parameters:

1. bg: to set he normal background color.
2. bg to set he normal background color.
3. font: to set the font on the button label.
4. image: to set the image on the button.
5. width: to set the width of the button.
6. height: to set the height of the button.

4. **Text**: To edit a multi-line text and format the way it has to be displayed.The general syntax is: w =Text(master, option=value)

Parameters:

1. highlightcolor: To set the color of the focus highlight when widget has to be focused.
2. insertbackground: To set the background of the widget.
3. bg: to set he normal background color.
4. font: to set the font on the button label.
5. image: to set the image on the widget.
6. width: to set the width of the widget.
7. height: to set the height of the widget.

5. **Listbox**: It offers a list to the user from which the user can accept any number of options.The general syntax is: w = Listbox(master, option=value)

Parameters:

1. highlightcolor: To set the color of the focus highlight when widget has to be focused.
2. bg: to set he normal background color.
3. bd: to set the border width in pixels.
4. font: to set the font on the button label.
5. image: to set the image on the widget.
6. width: to set the width of the widget.
7. height: to set the height of the widget.

Fig 2.2:-Listbox showing the list items

6. **Frame:** The Frame widget is very important for the process of grouping and organizing other widgets in a somehow friendly way. It works like a container, which is responsible for arranging the position of other widgets. It uses rectangular areas in the screen to organize the layout and to provide padding of these widgets. A frame can also be used as a foundation class to implement complex widgets.The general syntax is:Frame(master,options…..)

Parameters:

1. master − This represents the parent window.
2. options − Here is the list of most commonly used options for this widget. These options can be used as key-value pairs separated by commas.Options are bg, bd, cursor, height etc.

Messagebox module:- To show a minimalistic Tkinter message box, use the function showinfo() and showerror() where the parameters are the window title and text.

Ex:-

```
import tkinter
from tkinter import messagebox
root = tkinter.Tk()
root.withdraw()
messagebox.showerror("Error", "Error message")
messagebox.showwarning("Warning","Warning message")
messagebox.showinfo("Information","Informative message")
```

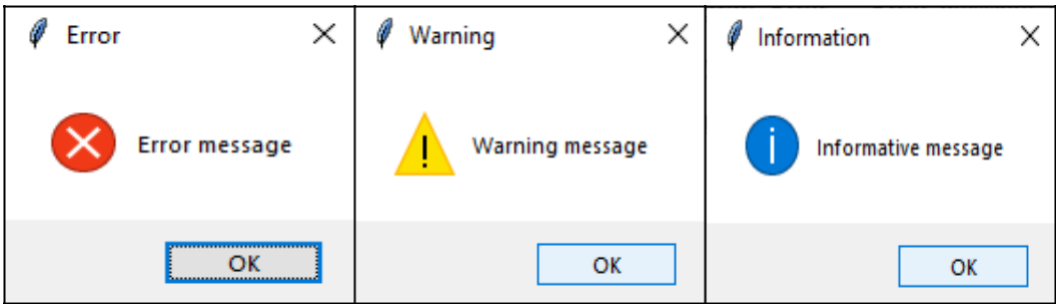


Fig 2.3 All 3 types of message boxes

Introduction to Machine Learning

The term Machine Learning was coined by Arthur Samuel in 1959, an American pioneer in the field of computer gaming and artificial intelligence and stated that "it gives computers the ability to learn without being explicitly programmed".

Machine Learning is a latest buzzword floating around. It deserves to, as it is one of the most interesting subfield of Computer Science.

Within the field of data analytics, machine learning is used to devise complex models and algorithms that lend themselves to prediction; in commercial use, this is known as predictive analytics. These analytical models allow researchers, data scientists, engineers, and analysts to "produce reliable, repeatable decisions and results" and uncover "hidden insights" through learning from historical relationships and trends in the data set(input).

Classification of Machine Learning :

1. **Supervised learning:** When an algorithm learns from example data and associated target responses that can consist of numeric values or string labels, such as classes or tags, in order to later predict the correct response when posed with new examples comes under the category of Supervised learning.

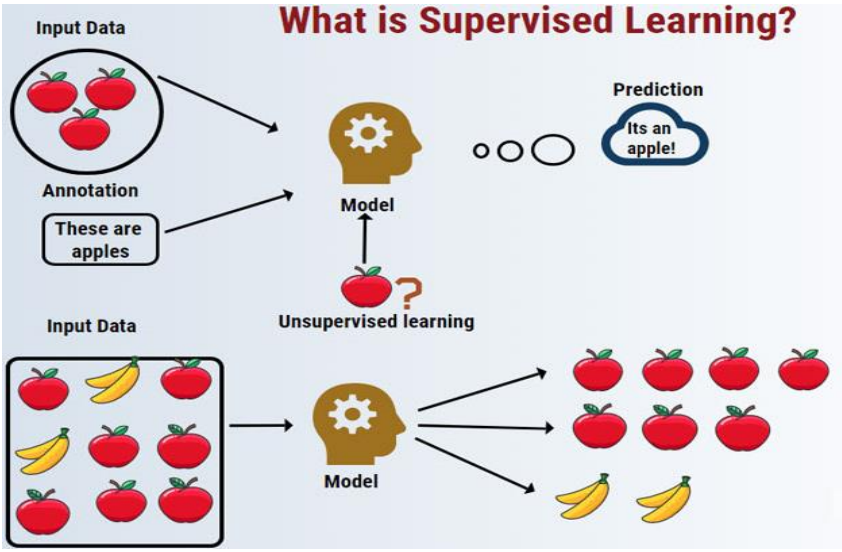


Fig. 2.4 :- Supervised Learning

2. **Unsupervised learning:** Whereas when an algorithm learns from plain examples without any associated response, leaving to the algorithm to determine the data patterns on its own. This type of algorithm tends to restructure the data into something else, such as new features that may represent a class or a new series of un-correlated values.

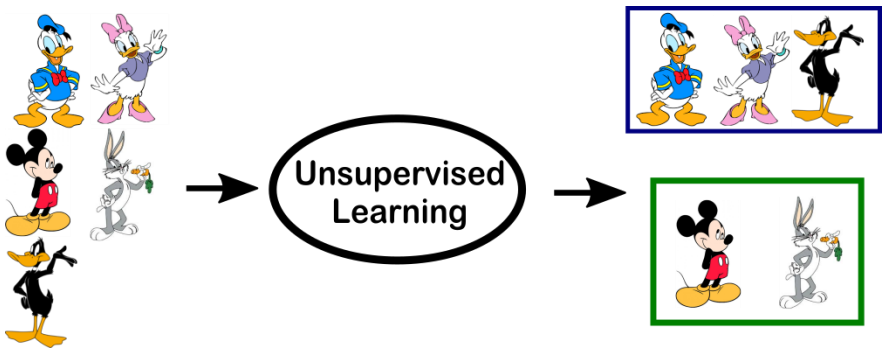


Fig.2.5 :- Unsupervised Learning

3. **Reinforcement learning:** When you present the algorithm with examples that lack labels, as in unsupervised learning. However, you can accompany an example with positive or negative feedback according to the solution the algorithm proposes comes under the category of Reinforcement learning, which

   is connected to applications for which the algorithm must make decisions (so the product is prescriptive, not just descriptive, as in unsupervised learning), and the decisions bear consequences. In the human world, it is just like learning by trial and error.

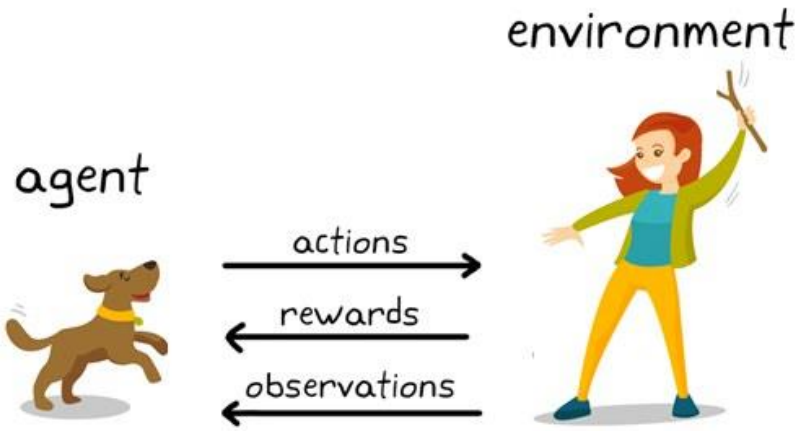


Fig.2.6 :- Reinforcement Learning

4. **Semi-supervised learning :** where an incomplete training signal is given: a training set with some (often many) of the target outputs missing. There is a special case of this principle known as Transduction where the entire set of problem instances is known at learning time, except that part of the targets are missing.

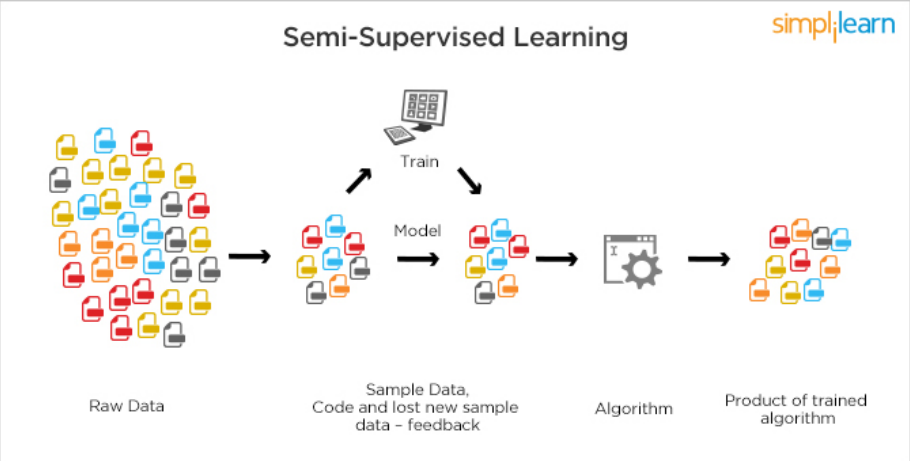


Fig.2.7 :- Semi-supervised Learning

Categorizing on the basis of required Output:

1. **Classification :** When inputs are divided into two or more classes, and the learner must produce a model that assigns unseen inputs to one or more (multi-label classification) of these classes. This is typically tackled in a supervised way. Spam filtering is an example of classification, where the inputs are email (or other) messages and the classes are “spam” and “not spam”.

2. **Regression :** Which is also a supervised problem, A case when the outputs are continuous rather than discrete.

3. **Clustering :** When a set of inputs is to be divided into groups. Unlike in classification, the groups are not known beforehand, making this typically an unsupervised task.

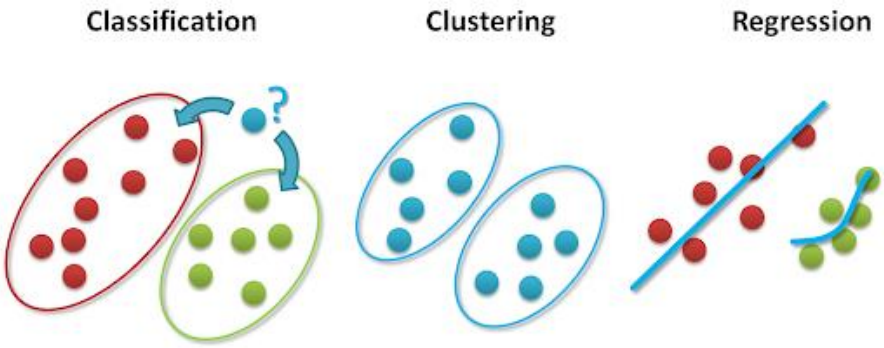


Fig.2.8 :- Classification Clustering Regression

Chapter 3
Implementation and User Interface

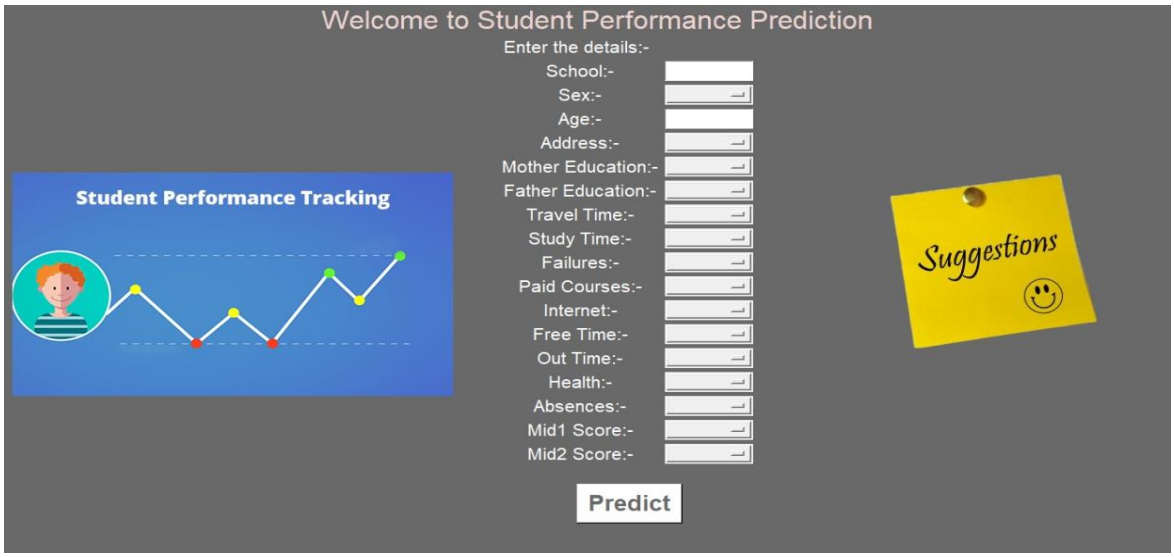


Fig 3.1 Welcome Window

- This is the Welcome window of the Student Performance Prediction.
- Here the multiple attributes are available to fill out and predict the result.

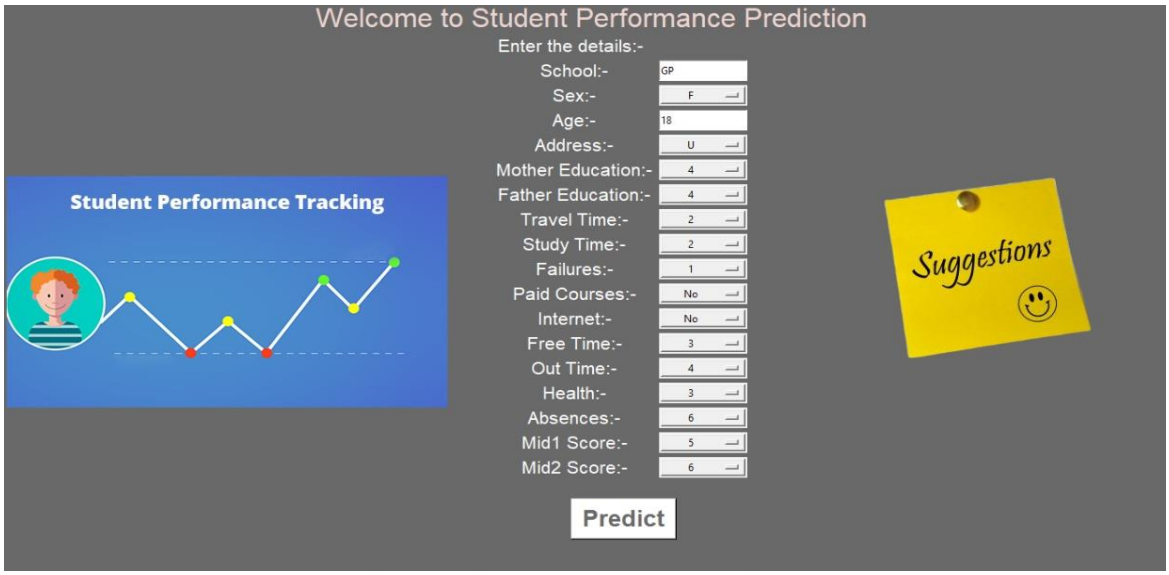


Fig 3.2 Details Entering Window

- In this window, the user is entering the details.
- After entering the details, click on the Predict button.

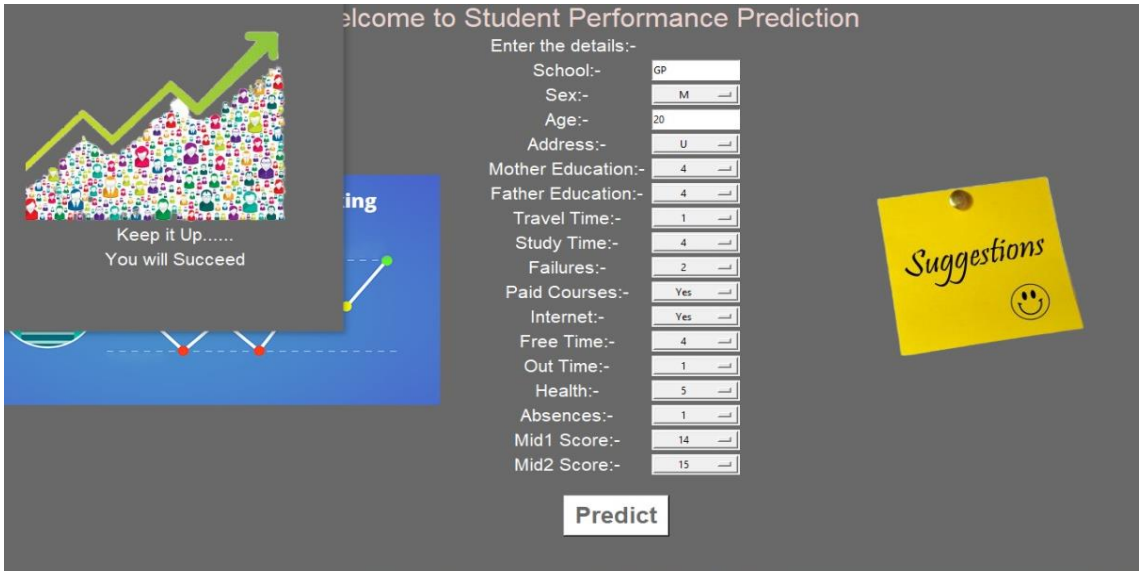


Fig 3.3 Success Outcome Window

- In this window, the details entered by student is evaluated.
- On the basis of the data, the new wizard is pop out which says that the student will succeed.

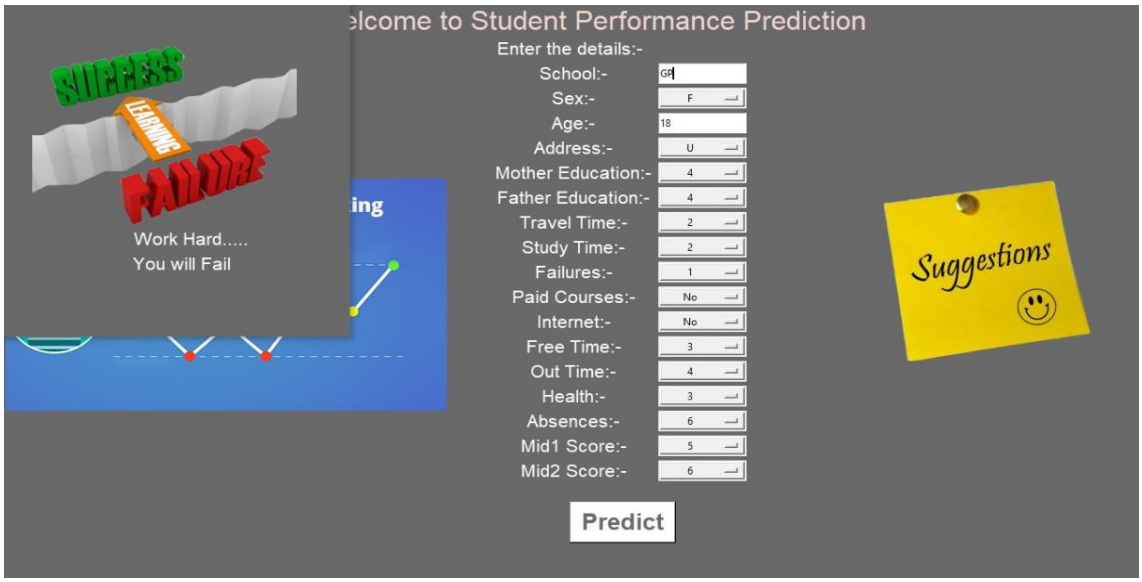
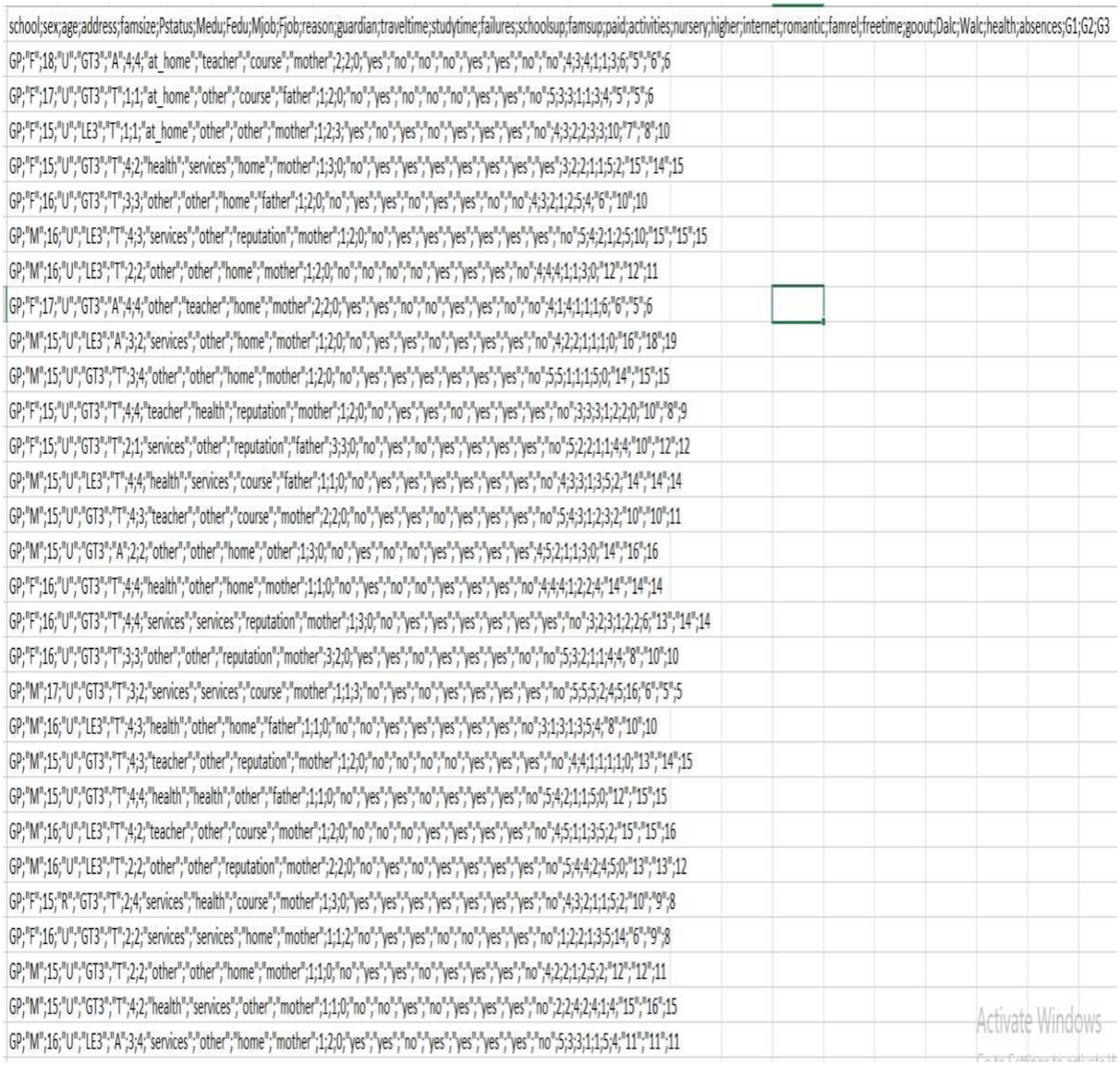


Fig 3.4 Failure Outcome Window

- In this window, the details entered by student is evaluated.
- On the basis of the data, the new wizard is pop out which says that the student will not succeed.



school;sex;age;address;famsize;Pstatus;Medu;Fedu;Mjob;Fjob;reason;guardian;traveltime;studytime;failures;schoolsup;famsup;paid;activities;nursery;higher;internet;romantic;famrel;freetime;goout;Dalc;Walc;health;absences;G1;G2;G3
GP;"F";18;"U";"GT3";"A";4;4;"at_home";"teacher";"course";"mother";2;2;0;"yes";"no";"no";"no";"yes";"yes";"no";"no";4;3;4;1;1;3;6;"5";"6";6
GP;"F";17;"U";"GT3";"T";1;1;"at_home";"other";"course";"father";1;2;0;"no";"yes";"no";"no";"no";"yes";"yes";"no";5;3;3;1;1;3;4;"5";"5";6
GP;"F";15;"U";"LE3";"T";1;1;"at_home";"other";"other";"mother";1;2;3;"yes";"no";"yes";"no";"yes";"yes";"yes";"no";4;3;2;2;3;3;10;"7";"8";10
GP;"F";15;"U";"GT3";"T";4;2;"health";"services";"home";"mother";1;3;0;"no";"yes";"yes";"yes";"yes";"yes";"yes";"yes";3;2;2;1;1;5;2;"15";"14";15
GP;"F";16;"U";"GT3";"T";3;3;"other";"other";"home";"father";1;2;0;"no";"yes";"yes";"no";"yes";"yes";"no";"no";4;3;2;1;2;5;4;"6";"10";10
GP;"M";16;"U";"LE3";"T";4;3;"services";"other";"reputation";"mother";1;2;0;"no";"yes";"yes";"yes";"yes";"yes";"yes";"no";5;4;2;1;2;5;10;"15";"15";15
GP;"M";16;"U";"LE3";"T";2;2;"other";"other";"home";"mother";1;2;0;"no";"no";"no";"no";"yes";"yes";"yes";"no";4;4;4;1;1;3;0;"12";"12";11
GP;"F";17;"U";"GT3";"A";4;4;"other";"teacher";"home";"mother";2;2;0;"yes";"yes";"no";"no";"yes";"yes";"no";"no";4;1;4;1;1;1;6;"6";"5";6
GP;"M";15;"U";"LE3";"A";3;2;"services";"other";"home";"mother";1;2;0;"no";"yes";"yes";"no";"yes";"yes";"yes";"no";4;2;2;1;1;1;0;"16";"18";19
GP;"M";15;"U";"GT3";"T";3;4;"other";"other";"home";"mother";1;2;0;"no";"yes";"yes";"yes";"yes";"yes";"yes";"no";5;5;1;1;1;5;0;"14";"15";15
GP;"F";15;"U";"GT3";"T";4;4;"teacher";"health";"reputation";"mother";1;2;0;"no";"yes";"yes";"no";"yes";"yes";"yes";"no";3;3;3;1;2;2;0;"10";"8";9
GP;"F";15;"U";"GT3";"T";2;1;"services";"other";"reputation";"father";3;3;0;"no";"yes";"no";"yes";"yes";"yes";"yes";"no";5;2;2;1;1;4;4;"10";"12";12
GP;"M";15;"U";"LE3";"T";4;4;"health";"services";"course";"father";1;1;0;"no";"yes";"yes";"yes";"yes";"yes";"yes";"no";4;3;3;1;3;5;2;"14";"14";14
GP;"M";15;"U";"GT3";"T";4;3;"teacher";"other";"course";"mother";2;2;0;"no";"yes";"yes";"no";"yes";"yes";"yes";"no";5;4;3;1;2;3;2;"10";"10";11
GP;"M";15;"U";"GT3";"A";2;2;"other";"other";"home";"other";1;3;0;"no";"yes";"no";"no";"yes";"yes";"yes";"yes";4;5;2;1;1;3;0;"14";"16";16
GP;"F";16;"U";"GT3";"T";4;4;"health";"other";"home";"mother";1;1;0;"no";"yes";"no";"no";"yes";"yes";"yes";"no";4;4;4;1;2;2;4;"14";"14";14
GP;"F";16;"U";"GT3";"T";4;4;"services";"services";"reputation";"mother";1;3;0;"no";"yes";"yes";"yes";"yes";"yes";"yes";"no";3;2;3;1;2;2;6;"13";"14";14
GP;"F";16;"U";"GT3";"T";3;3;"other";"other";"reputation";"mother";3;2;0;"yes";"yes";"no";"yes";"yes";"yes";"no";"no";5;3;2;1;1;4;4;"8";"10";10
GP;"M";17;"U";"GT3";"T";3;2;"services";"services";"course";"mother";1;1;3;"no";"yes";"no";"yes";"yes";"yes";"yes";"no";5;5;5;2;4;5;16;"6";"5";5
GP;"M";16;"U";"LE3";"T";4;3;"health";"other";"home";"father";1;1;0;"no";"no";"yes";"yes";"yes";"yes";"yes";"no";3;1;3;1;3;5;4;"8";"10";10
GP;"M";15;"U";"GT3";"T";4;3;"teacher";"other";"reputation";"mother";1;2;0;"no";"no";"no";"no";"yes";"yes";"yes";"no";4;4;1;1;1;1;0;"13";"14";15
GP;"M";15;"U";"GT3";"T";4;4;"health";"health";"other";"father";1;1;0;"no";"yes";"yes";"no";"yes";"yes";"yes";"no";5;4;2;1;1;5;0;"12";"15";15
GP;"M";16;"U";"LE3";"T";4;2;"teacher";"other";"course";"mother";1;2;0;"no";"no";"no";"yes";"yes";"yes";"yes";"no";4;5;1;1;3;5;2;"15";"15";16
GP;"M";16;"U";"LE3";"T";2;2;"other";"other";"reputation";"mother";2;2;0;"no";"yes";"no";"yes";"yes";"yes";"yes";"no";5;4;4;2;4;5;0;"13";"13";12
GP;"F";15;"R";"GT3";"T";2;4;"services";"health";"course";"mother";1;3;0;"yes";"yes";"yes";"yes";"yes";"yes";"yes";"no";4;3;2;1;1;5;2;"10";"9";8
GP;"F";16;"U";"GT3";"T";2;2;"services";"services";"home";"mother";1;1;2;"no";"yes";"yes";"no";"no";"yes";"yes";"no";1;2;2;1;3;5;14;"6";"9";8
GP;"M";15;"U";"GT3";"T";2;2;"other";"other";"home";"mother";1;1;0;"no";"yes";"yes";"no";"yes";"yes";"yes";"no";4;2;2;1;2;5;2;"12";"12";11
GP;"M";15;"U";"GT3";"T";4;2;"health";"services";"other";"mother";1;1;0;"no";"no";"yes";"no";"yes";"yes";"yes";"no";2;2;4;2;4;1;4;"15";"16";15
GP;"M";16;"U";"LE3";"A";3;4;"services";"other";"home";"mother";1;2;0;"yes";"yes";"no";"yes";"yes";"yes";"yes";"no";5;3;3;3;1;1;5;4;"11";"11";11

Fig 3.5 Dataset

- This is the snapshot of the dataset, which we have provided to get an accuracy.

```
Model Accuracy Knowing G1 & G2 Scores
=====================================
Mean Model Accuracy: 0.9178075396825397
C:\Users\User\AppData\Roaming\Python\Python37\site-packages\sklearn\svm\_base.py
  "the number of iterations.", ConvergenceWarning)
     school  sex  age  address  Medu  ...  goout  health  absences  G1  G2
226       0    0   17        1     3  ...      4       3        10   1   1
35        0    0   15        1     2  ...      1       5         0   0   0
219       0    0   17        1     2  ...      3       4         4   0   1
80        0    1   15        1     2  ...      2       3         2   1   1
64        0    0   15        1     4  ...      4       2         0   1   1
..      ...  ...  ...      ...   ...  ...    ...     ...       ...  ..  ..
138       0    1   16        1     1  ...      4       5         0   1   1
53        0    0   15        1     4  ...      4       5         0   0   1
112       0    0   16        1     2  ...      2       5         6   1   1
107       0    1   16        1     3  ...      3       5         2   1   1
101       0    1   16        1     4  ...      3       4         0   1   1

[79 rows x 17 columns]
False Pass Rate:  0.08
False Fail Rate:  0.07407407407407407


Model Accuracy Knowing Only G1 Score
====================================
Mean Model Accuracy: 0.83234126984127
C:\Users\User\AppData\Roaming\Python\Python37\site-packages\sklearn\svm\_base.py:947:
  "the number of iterations.", ConvergenceWarning)
     school  sex  age  address  Medu  ...  freetime  goout  health  absences  G1
226       0    0   17        1     3  ...         3      4       3        10   1
35        0    0   15        1     2  ...         5      1       5         0   0
219       0    0   17        1     2  ...         3      3       4         4   0
80        0    1   15        1     2  ...         2      2       3         2   1
64        0    0   15        1     4  ...         4      4       2         0   1
..      ...  ...  ...      ...   ...  ...       ...    ...     ...       ...  ..
138       0    1   16        1     1  ...         4      4       5         0   1
53        0    0   15        1     4  ...         3      4       5         0   0
112       0    0   16        1     2  ...         1      2       5         6   1
107       0    1   16        1     3  ...         3      3       5         2   1
101       0    1   16        1     4  ...         4      3       4         0   1

[79 rows x 16 columns]
False Pass Rate:  0.12
False Fail Rate:  0.14814814814814814


Model Accuracy Without Knowing Scores
=====================================
Mean Model Accuracy: 0.6962797619047618
C:\Users\User\AppData\Roaming\Python\Python37\site-packages\sklearn\svm\_ba
  "the number of iterations.", ConvergenceWarning)
     school  sex  age  address  ...  freetime  goout  health  absences
226       0    0   17        1  ...         3      4       3        10
35        0    0   15        1  ...         5      1       5         0
219       0    0   17        1  ...         3      3       4         4
80        0    1   15        1  ...         2      2       3         2
64        0    0   15        1  ...         4      4       2         0
..      ...  ...  ...      ...  ...       ...    ...     ...       ...
138       0    1   16        1  ...         4      4       5         0
53        0    0   15        1  ...         3      4       5         0
112       0    0   16        1  ...         1      2       5         6
107       0    1   16        1  ...         3      3       5         2
101       0    1   16        1  ...         4      3       4         0

[79 rows x 15 columns]
False Pass Rate:  0.68
False Fail Rate:  0.05555555555555555
```

Result :

Our Accuracy :

- Model Accuracy with knowing G1 and G2 scores : **91.78 %**
- Model Accuracy with knowing only G1 score : **83.23 %**
- Model Accuracy without knowing scores : **69.62 %**

Chapter 4

Conclusion

As title suggests the aim of project is to predict Student Performance. Universities gather large volumes of data with reference to their students in electronic form. The advances in the data mining field make it possible to mine these educational data and find information that allow for innovative ways of supporting both teachers and students.

It will act as an aid for the universities or institutions as universities gather large volumes of data with reference to their students in electronic form. The advances in the data mining field make it possible to mine these educational data and find information that allow for innovative ways of supporting both teachers and students.

This project can be extended as proper analysis platform for students and suggesting them the video lectures for the subjects they have to focus. So, this project can contribute in making our education pattern better.

This project will make contribution in universities and institutions with great accuracy to improve their students' performance. It will also help students in making their future bright. The massive growth in the educational sector needs to create awareness about handling the huge volume of student data. The educational data mining is a technique to extract information from these volumes of data. Nowadays educational data mining technique plays a vital role in predicting academic performance.

Chapter 5

Bibliography

www.geeksforgeeks.org/

https://towardsdatascience.com/machine-learning-an-introduction-23b84d51e6d0

https://www.researchgate.net/publication/254558610_Predicting_Student_Performance_A_Statistical_and_Data_Mining_Approach

https://educationaltechnologyjournal.springeropen.com/articles/10.1186/s41239-019-0160-3

https://www.pantechsolutions.net/student-performance-prediction-using-machine-learning

Chapter 6

Appendices

```
import numpy as np
import pandas as pd
from tkinter import *
from PIL import ImageTk
ds = pd.read_csv(r"D:\Project\Student Performance Prediction\student-mat.csv", sep=";")

from sklearn.preprocessing import LabelEncoder
from sklearn.model_selection import train_test_split
from sklearn.metrics import confusion_matrix
from sklearn.model_selection import GridSearchCV, cross_val_score

from sklearn.pipeline import Pipeline
from sklearn.feature_selection import SelectKBest, chi2
from sklearn.svm import LinearSVC # Support Vector Machine Classifier model

def split_data(X, Y):
    return train_test_split(X, Y, test_size=0.2, random_state=17)

def confuse(y_true, y_pred):
    cm = confusion_matrix(y_true=y_true, y_pred=y_pred)
    # print("\nConfusion Matrix: \n", cm)
    fpr(cm)
    ffr(cm)

""" False Pass Rate """
def fpr(confusion_matrix):
    fp = confusion_matrix[0][1]
    tf = confusion_matrix[0][0]
    rate = float(fp) / (fp + tf)
    print("False Pass Rate: ", rate)

""" False Fail Rate """
def ffr(confusion_matrix):
    ff = confusion_matrix[1][0]
    tp = confusion_matrix[1][1]
    rate = float(ff) / (ff + tp)
    print("False Fail Rate: ", rate)

    return rate

model=""

""" Train Model and Print Score """
def train_and_score(X, y):
    X_train, X_test, y_train, y_test = split_data(X, y)

    clf = Pipeline([
        ('reduce_dim', SelectKBest(chi2, k=2)),
        ('train', LinearSVC(C=100))
    ])

    scores = cross_val_score(clf, X_train, y_train, cv=5, n_jobs=2)
    print("Mean Model Accuracy:", np.array(scores).mean())

    clf.fit(X_train, y_train)
    print(X_test)
    confuse(y_test, clf.predict(X_test))
    print()
    return clf

""" Main Program """
def main():
    global model
    print("\nStudent Performance Prediction")

    df=ds.drop(columns=['famsize','Pstatus','Mjob','Fjob','reason','guardian','schoolsup','famsup',\
                        'activities','nursery','higher','romantic','famrel','Dalc','Walc'])

    class_le = LabelEncoder()
    for column in df[["school", "sex", "address","paid","internet"]].columns:
        df[column] = class_le.fit_transform(df[column].values)

    # Encode G1, G2, G3 as pass or fail binary values
    for i, row in df.iterrows():
        if row["G1"] >= 10:
            df["G1"][i] = 1
```

```
        else:
            df["G1"][i] = 0

        if row["G2"] >= 10:
            df["G2"][i] = 1
        else:
            df["G2"][i] = 0

        if row["G3"] >= 10:
            df["G3"][i] = 1
        else:
            df["G3"][i] = 0

    # Target values are G3
    y = df.pop("G3")

    # Feature set is remaining features
    X = df

    print("\n\nModel Accuracy Knowing G1 & G2 Scores")
    print("======================================")
    model=train_and_score(X, y)


# Remove grade report 2
    X.drop(["G2"], axis = 1, inplace=True)
    print("\n\nModel Accuracy Knowing Only G1 Score")
    print("======================================")
    train_and_score(X, y)

    # Remove grade report 1
    X.drop(["G1"], axis=1, inplace=True)
    print("\n\nModel Accuracy Without Knowing Scores")
    print("======================================")
    train_and_score(X, y)

    root=Tk()
    root.title("Student Performance Prediction")
    root.geometry("1350x700+0+0")
    bg_i=ImageTk.PhotoImage(file=r"D:\Project\Student Performance Prediction\Tracking_Headline.png")
    bg_l = Label(root, image=bg_i,bg="dimgray")
    bg_l.place(x=10, y=200,width=500,height=300)
    bg_i2=ImageTk.PhotoImage(file=r"D:\Project\Student Performance Prediction\suggestions.png")
    bg_l1 = Label(root, image=bg_i2,bg="dimgray")
    bg_l1.place(x=1000, y=200,width=240,height=230)
    T=Text(root,height=1,width=35,font=("bold",24),bg="dimgray",bd=0,fg="mistyrose2")
    T.pack()
    T.insert(END,'Welcome to Student Performance Prediction')
    root.configure(background="dimgray")
    l=Label(root,text='Enter the details:-',font=('bold',16),fg="white",bg="dimgray")
    l.place(x=550,y=40,width=200,height=25)
    l=Label(root,text='School:-',font=('bold',16),fg="white",bg="dimgray")
    l.place(x=554,y=70,width=200,height=25)
    e1=Entry(root)
    e1.place(x=750,y=70,width=100,height=25)
    l=Label(root,text='Sex:-',font=('bold',16),fg="white",bg="dimgray")
    l.place(x=554,y=100,width=200,height=25)
    svar=StringVar(root)
    s=["M","F"]
    w=OptionMenu(root,svar,*s)
    w.place(x=750,y=100,width=100,height=25)
    l=Label(root,text='Age:-',font=('bold',16),fg="white",bg="dimgray")
    l.place(x=554,y=130,width=200,height=25)
    e2=Entry(root)
    e2.place(x=750,y=130,width=100,height=25)
    add=StringVar(root)
    s2=["U","R"]
    l=Label(root,text='Address:-',font=('bold',16),fg="white",bg="dimgray")
    l.place(x=554,y=160,width=200,height=25)
    w=OptionMenu(root,add,*s2)
    w.place(x=750,y=160,width=100,height=25)
    medu=StringVar(root)
    fedu=StringVar(root)
    s1=['1','2','3','4']
    s3=['1','2','3','4','5','6','7','8','9','10']
    s4=['Yes','No']
```

```
s5=['1','2','3','4','5']
s6=list(range(0,21))
l=Label(root,text='Mother Education:-',font=('bold',16),fg="white",bg="dimgray")
l.place(x=554,y=190,width=200,height=25)
w=OptionMenu(root,medu,*s1)
w.place(x=750,y=190,width=100,height=25)

l=Label(root,text='Father Education:-',font=('bold',16),fg="white",bg="dimgray")
l.place(x=554,y=220,width=200,height=25)
w=OptionMenu(root,fedu,*s1)
w.place(x=750,y=220,width=100,height=25)

tt=StringVar(root)
l=Label(root,text='Travel Time:-',font=('bold',16),fg="white",bg="dimgray")
l.place(x=554,y=250,width=200,height=25)
w=OptionMenu(root,tt,*s3)
w.place(x=750,y=250,width=100,height=25)

st=StringVar(root)
l=Label(root,text='Study Time:-',font=('bold',16),fg="white",bg="dimgray")
l.place(x=554,y=280,width=200,height=25)
w=OptionMenu(root,st,*s3)
w.place(x=750,y=280,width=100,height=25)

f=StringVar(root)
l=Label(root,text='Failures:-',font=('bold',16),fg="white",bg="dimgray")
l.place(x=554,y=310,width=200,height=25)
w=OptionMenu(root,f,*s3)
w.place(x=750,y=310,width=100,height=25)

pc=StringVar(root)
l=Label(root,text='Paid Courses:-',font=('bold',16),fg="white",bg="dimgray")
l.place(x=554,y=340,width=200,height=25)
w=OptionMenu(root,pc,*s4)
w.place(x=750,y=340,width=100,height=25)

inte=StringVar(root)
l=Label(root,text='Internet:-',font=('bold',16),fg="white",bg="dimgray")
l.place(x=554,y=370,width=200,height=25)
w=OptionMenu(root,inte,*s4)
w.place(x=750,y=370,width=100,height=25)

ft=StringVar(root)
l=Label(root,text='Free Time:-',font=('bold',16),fg="white",bg="dimgray")
l.place(x=554,y=400,width=200,height=25)
w=OptionMenu(root,ft,*s3)
w.place(x=750,y=400,width=100,height=25)

ot=StringVar(root)
l=Label(root,text='Out Time:-',font=('bold',16),fg="white",bg="dimgray")
l.place(x=554,y=430,width=200,height=25)
w=OptionMenu(root,ot,*s3)
w.place(x=750,y=430,width=100,height=25)

h=StringVar(root)
l=Label(root,text='Health:-',font=('bold',16),fg="white",bg="dimgray")
l.place(x=554,y=460,width=200,height=25)
w=OptionMenu(root,h,*s5)
w.place(x=750,y=460,width=100,height=25)

ab=StringVar(root)
l=Label(root,text='Absences:-',font=('bold',16),fg="white",bg="dimgray")
l.place(x=554,y=490,width=200,height=25)
w=OptionMenu(root,ab,*s3)
w.place(x=750,y=490,width=100,height=25)

sc1=StringVar(root)
l=Label(root,text='Mid1 Score:-',font=('bold',16),fg="white",bg="dimgray")
l.place(x=554,y=520,width=200,height=25)
w=OptionMenu(root,sc1,*s6)
w.place(x=750,y=520,width=100,height=25)

sc2=StringVar(root)
l=Label(root,text='Mid2 Score:-',font=('bold',16),fg="white",bg="dimgray")
l.place(x=554,y=550,width=200,height=25)
w=OptionMenu(root,sc2,*s6)
```

```
    w.place(x=750,y=550,width=100,height=25)
    def success():
        tp=Toplevel()
        tp.title("Success")
        tp.geometry("400x400+0+0")
        tp.configure(background="dimgray")
        bg_i=ImageTk.PhotoImage(file=r"D:\Project\Student Performance Prediction\success.png")
        bg_l = Label(tp, image=bg_i,bg="dimgray")
        bg_l.place(x=40, y=10,width=300,height=300)
        l=Label(tp,text='Keep it Up......',font=('bold',16),fg="white",bg="dimgray")
        l.place(x=10,y=270,width=400,height=25)
        l=Label(tp,text='You will Succeed',font=('bold',16),fg="white",bg="dimgray")
        l.place(x=10,y=300,width=400,height=25)
        tp.mainloop()
    def failure():
        tp=Toplevel()
        tp.title("Failure")
        tp.geometry("400x400+0+0")
        tp.configure(background="dimgray")
        bg_i=ImageTk.PhotoImage(file=r"D:\Project\Student Performance Prediction\failure.png")
        bg_l = Label(tp, image=bg_i,bg="dimgray")
        bg_l.place(x=40, y=10,width=300,height=300)
        l=Label(tp,text='Work Hard.....',font=('bold',16),fg="white",bg="dimgray")
        l.place(x=10,y=270,width=420,height=25)
        l=Label(tp,text='You will Fail',font=('bold',16),fg="white",bg="dimgray")
        l.place(x=10,y=300,width=400,height=25)
        tp.mainloop()

    def suggest():
        di={}
        di['school']=[e1.get()]
        di['sex']=[svar.get()]
        di['age']=[int(e2.get())]
        di['address']=[add.get()]
        di['Medu']=[int(medu.get())]
        di['Fedu']=[int(fedu.get())]
        di['traveltime']=[int(tt.get())]
        di['studytime']=[int(st.get())]
        di['failures']=[int(f.get())]
        di['paid']=[pc.get()]
        di['internet']=[inte.get()]
        di['freetime']=[int(ft.get())]
        di['goout']=[int(ot.get())]
        di['health']=[int(h.get())]
        di['absences']=[int(ab.get())]
        di['G1']=[int(sc1.get())]
        di['G2']=[int(sc2.get())]
        df1=pd.DataFrame(di)
        class_le = LabelEncoder()
        for column in df1[["school", "sex", "address","paid","internet"]].columns:
            df1[column] = class_le.fit_transform(df1[column].values)
        for i, row in df1.iterrows():
            if row["G1"] >= 10:
                df1["G1"][i] = 1
            else:
                df1["G1"][i] = 0

            if row["G2"] >= 10:
                df1["G2"][i] = 1
            else:
                df1["G2"][i] = 0
        pred=model.predict(df1)
        print(df1)
        print(pred[0])
        if pred[0]==0:
            failure()
        else:
            success()

    btn4=Button(root,command=suggest,text="Predict",compound=LEFT,\
                font=("Industry Inc Detail Fill",20, "bold"), bg="white", fg="dimgray")
    btn4.place(x=650,y=600,height=50,width=120)
    root.mainloop()
main()
```