**1.INTRODUCTION**

This report provide a detailed account of the processes which our group has used to design and implement Student grade prediction system that can be used to predict the final grades of the students. A student grade prediction project is a machine learning project that aims to predict the grades of students in a course or program based on historical data. The project typically involves collecting data on students' past grades, demographic information, and other relevant factors, and then using that data to train a predictive model. Once the model has been trained, it can be used to make predictions about the grades of new students based on their demographic information and other relevant factors. This can be used to identify students who may be at risk of performing poorly, and intervene to provide support and resources to help them succeed. Additionally, it can also be used to predict the performance of students who haven't taken the course yet, and adjust the teaching methodologies or curriculum to better accommodate the student population.

* 1. **STUDENT GRADE PREDICTION SYSTEM**

Since universities are prestigious places of higher education, students’ retention in these universities is a matter of high concern. It has been found that most of the students’ drop-out from the universities during their first year is due to lack of proper support in undergraduate courses. Due to this reason, the first year of the undergraduate student is referred as a “make or break” year. Without getting any support on the course domain and its complexity, it may demotivate a student and can be the cause to withdraw the course.

There is a great need to develop an appropriate solution to assist students retention at higher education institutions. Early grade prediction is one of the solutions that have a tendency to monitor students’ progress in the degree courses at the University and will lead to improving the students’ learning process based on predicted grades.

Using machine learning with Educational Data Mining can improve the learning process of students. Different models can be developed to predict students’ grades in the enrolled courses, which provide valuable information to facilitate students’ retention in those courses. This information can be used to early identify students at-risk based on which a system can 1 suggest the instructors to provide special attention to those students. This information can also help in predicting the students’ grades in different courses to monitor their performance in a better way that can enhance the students’ retention rate of the universities.

Using various packages such as cufflinks, seaborn & matplotlib to represent the data along with different attributes graphically or pictorially to analyse the dataset for predicting the Final Grade(G3).

* 1. **PROJECT OBJECTIVE**

Prediction of the final grade of Portugese high school students

* 1. **PROJECT SPECIFICATION**

This project aims at prediction of Student’s Grade that predicts the grade of the student and it also helps to monitor the student activities and performance which helps to increase their performance.

**2.SYSTEM SPECIFICATION**

**2.1Hardware specification**

* Processor : Intel dual core
* Processor speed: 1.04GHZ
* Ram : 1GB
* Monitor
* Keyboard
* Mouse

**2.2** **Software** **specification**

* OS: Microsoft Windows
* Language: Python
* Compiler : googlecolab

**3.PACKAGES**

**3.1 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.

**INSTALLING NUMPY PACKAGE**

pip install numpy

**WHY USE NUMPY?**

In Python we have lists that serve the purpose of arrays, but they are slow to process.

NumPy aims to provide an array object that is up to 50x faster than traditional Python lists.

The array object in NumPy is called ndarray, it provides a lot of supporting functions that make working with ndarray very easy.

Arrays are very frequently used in data science, where speed and resources are very important.

**IMPORT NUMPY**

Once NumPy is installed, import it in your applications by adding the import keyword:

import numpy

**NUMPY AS np:**

NumPy is usually imported under the np.

Create an np with the as keyword while importing:

import numpy as np

Now the NumPy package can be referred to as np instead of numpy.

**Example:**

import numpy as np

arr = np.array([1, 2, 3, 4, 5])

print(arr)

**0-D Arrays**

0-D arrays, or Scalars, are the elements in an array. Each value in an array is a 0-D array.

**1-D Arrays**

An array that has 0-D arrays as its elements is called uni-dimensional or 1-D array.

These are the most common and basic arrays.

**2-D Arrays**

An array that has 1-D arrays as its elements is called a 2-D array.

These are often used to represent matrix or 2nd order tensors.

**3.2 PANDAS**

* Pandas is a Python library used for working with data sets.
* It has functions for analyzing, cleaning, exploring, and manipulating data.
* The name "Pandas" has a reference to both "Panel Data", and "Python Data Analysis" and was created by Wes McKinney in 2008.

**Why Use Pandas**

Pandas allows us to analyze big data and make conclusions based on statistical theories.

Pandas can clean messy data sets, and make them readable and relevant.

Relevant data is very important in data science.

Pandas gives you answers about the data. Like:

* Is there a correlation between two or more columns?
* What is average value?
* Max value?
* Min value?
* Pandas are also able to delete rows that are not relevant, or contains wrong values, like empty or NULL values. This is called *cleaning* the data.

**INSTALLING PANDAS PACKAGE**

pip install pandas

**Import Pandas**

Once Pandas is installed, import it in your applications by adding the import keyword:

import pandas

Now Pandas is imported and ready to use

**Example:**

Importpandas

mydataset={'cars':["BMW","Volvo","Ford"],'passings':[3,7,2]}  
myvar=pandas.DataFrame(mydataset)  
print(myvar)

**Pandas as pd**

Pandas is usually imported under the pd

Create an pd with the as keyword while importing:

import pandas as pd

Now the Pandas package can be referred to as pd instead of pandas.

**3.3 MATPLOTLIB**

* Matplotlib is a cross-platform, data visualization and graphical plotting library for Python and its numerical extension NumPy.
* As such, it offers a viable open source alternative to **MATLAB.** Developers can also use matplotlib’s APIs(Application Programming Interfaces) to embed plots inGUI applications.

A Python matplotlib script is structured so that a fewlines of code are all that is required in most instancesto generate a visual data plot.

The matplotlib scripting layer overlays two APIs:

* The pyplot API is a hierarchy of Python codeobjects topped by matplotlib.pyplot
* An OO (Object-Oriented) API collection of objectsthat can be assembled with greater flexibility thanpyplot. This API provides direct access to Matplotlib’sbackend layers.

**Matplotlib and Pyplot in Python :**

The pyplot API has a convenient MATLAB-style statefulinterface. In fact, matplotlib was originally written as an open source alternative for MATLAB. The OO API and its interface is more customizable and powerful than pyplot, but considered more difficult to use. As a result, the pyplot interface is more commonly used, and is referred to by default in this article.

Understanding matplotlib’s pyplot API is key to understanding how to work with plots:

* **matplotlib.pyplot.figure**: Figure is the top-level container. It includes everything visualized in a plot including one or more Axes.
* **matplotlib.pyplot.axes**: Axes contain most of the elements in a plot: Axis, Tick, Line2D, Text, etc., and sets the coordinates. It is the area in which data is plotted. Axes include the X-Axis, Y-Axis, and possibly a Z-Axis, as well.

**Installing Matplotlib :**

pip install matplotlib

**3.3.1 MATPLOTLIB BAR PLOT:**

A bar plot or bar chart is a graph that represents the category of data with rectangular bars with lengths and heights that is proportional to the values which they represent. The bar plots can be plotted horizontally or vertically. A bar chart describes the comparisons between the discrete categories. One of the axis of the plot represents the specific categories being compared, while the other axis represents the measured values corresponding to those categories.

**Creating a bar plot:**

The matplotlib API in Python provides the bar() function which can be used in MATLAB style use or as an object-oriented API. The syntax of the bar() function to be used with the axes is as follows:- plt.bar(x, height, width, bottom, align).The function creates a bar plot bounded with a rectangle depending on the given parameters. Following is a simple example of the bar plot, which represents the number of students enrolled in different courses of an institute.

**EXAMPLE:**

import numpy as np

import matplotlib.pyplot as plt

data = {'C':20, 'C++':15, 'Java':30,'Python':35}

courses = list(data.keys())

values = list(data.values())

fig = plt.figure(figsize = (10, 5))

plt.bar(courses, values, color ='maroon',width = 0.4)

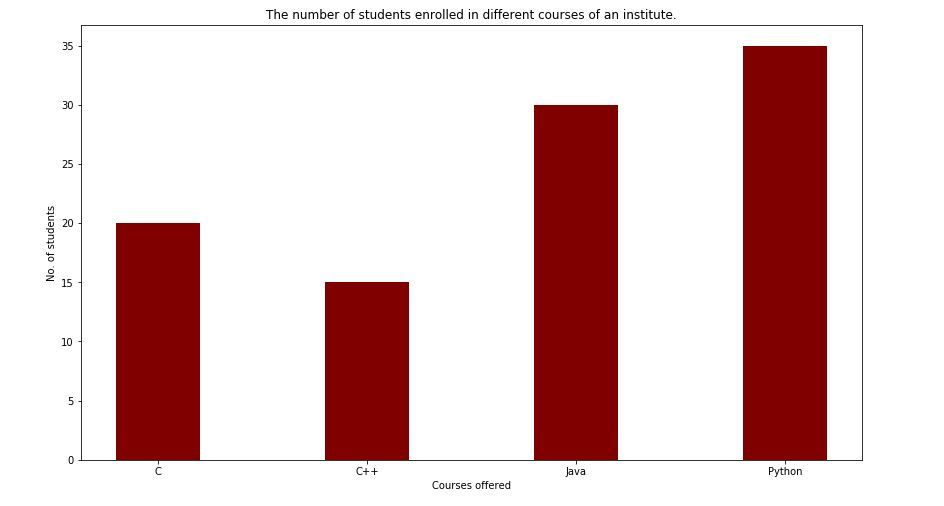
plt.xlabel("Courses offered")

plt.ylabel("No. of students enrolled")

plt.title("Students enrolled in different courses")

plt.show()

**Output:**

**FIGURE 1 :BAR CHART-Courses offered vs No.of Students**

**4.APPENDIX**

**4.1 SOURCE CODE AND SCREENSHOTS**

 I will start this task by importing the necessary Python libraries and the dataset:

|  |  |
| --- | --- |
|  | import numpy as np |
|  | import pandas as pd |
|  | from sklearn.linear\_model import LinearRegression |
|  | data = pd.read\_csv("student-mat.csv") |
|  | data.head() |

The dataset that I am using for the task of students grade prediction is based on the achievements of the students of the Portuguese schools. In this dataset the G1 represents the grades of the first period, G2 represents the grades of the second period, and G3 represents the final grades. Now let’s prepare the data and let’s see how we can predict the final grades of the students.

data = data[["G1", "G2", "G3", "studytime", "failures", "absences"]]

predict = "G3"

x = np.array(data.drop([predict], 1))

y = np.array(data[predict])

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

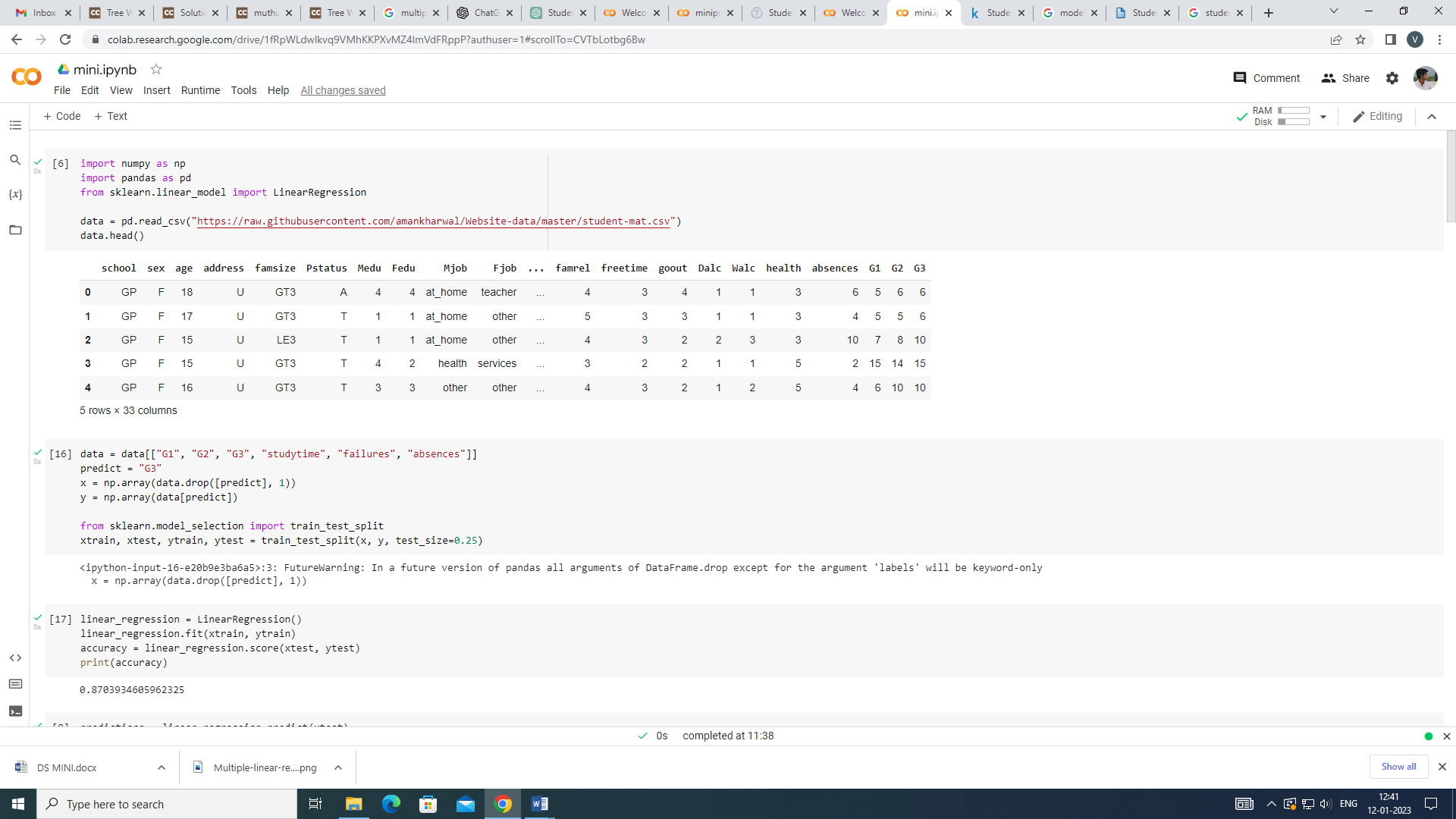
xtrain, xtest, ytrain, ytest = train\_test\_split(x, y, test\_size=0.2)

The linear regression model gave an accuracy of about 84% which is not bad in this task. Now let’s have a look at the predictions made by the students’ grade prediction model:

predictions = linear\_regression.predict(xtest)

for i in range(len(predictions)):

print(predictions[x], xtest[x], [ytest[x]])

FIGURE 2:

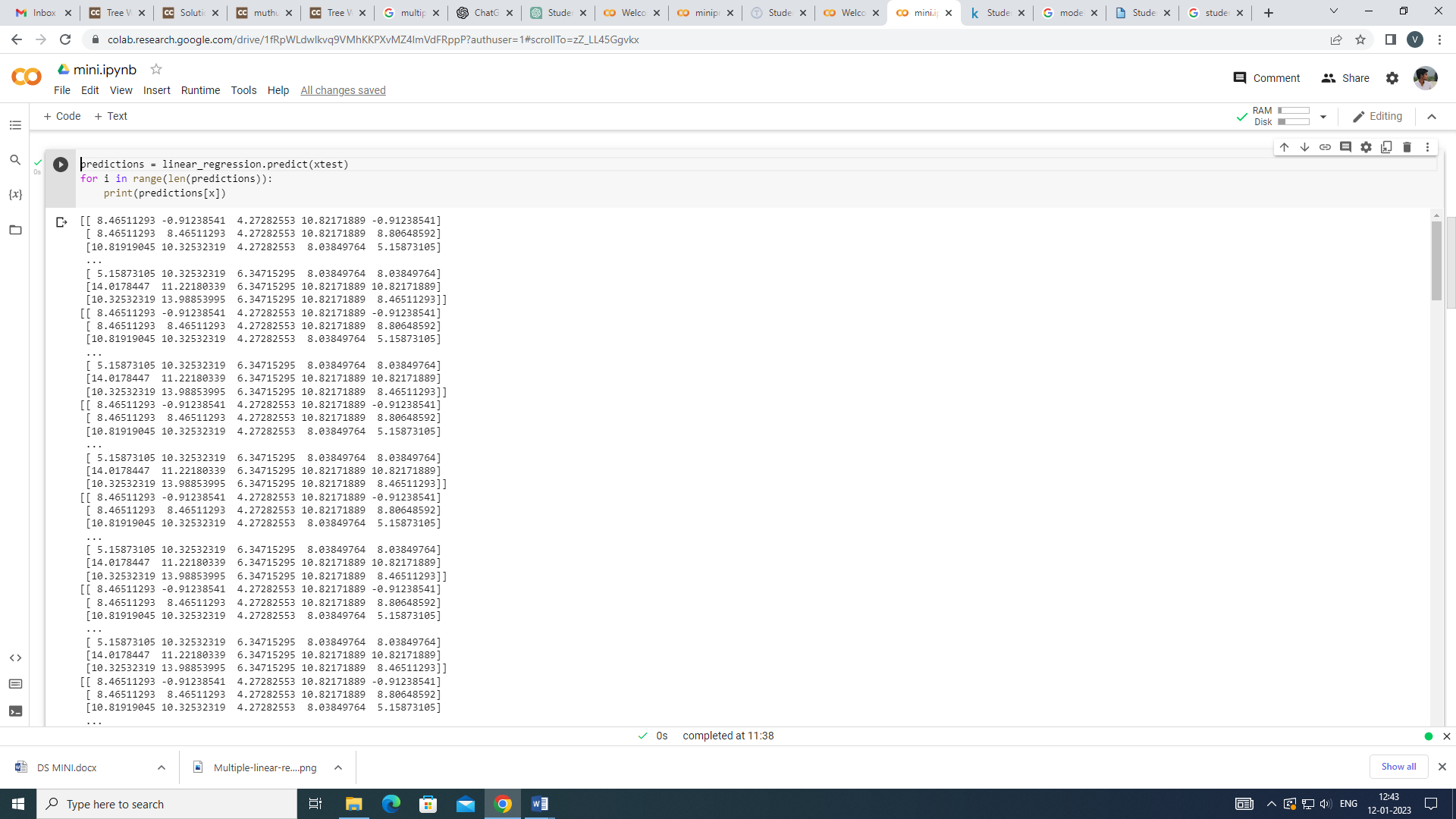


FIGURE 3:

**DATA VISUALIZATION:**

Data Visualization involves the use of graphs, charts, and tables to present data visually. These visual tools show how the different data attributes are related to each other. Data Modeling is used to ensure that data is stored in a database and represented accurately.

Some examples of the data visualization in our project:

# Visualizing the regression line

import matplotlib.pyplot as plt

linear\_regression.coef\_

linear\_regression.intercept\_

plt.scatter(data['G1'],data['G2'],y)

a=((-1.5596) + (0.1578\*x)+(0.9822\*x)+(-0.2128\*x)+(-0.1002\*x)+(0.0369\*x))

plt.plot(x,a,'r')

plt.show()

**SCATTERPLOT:**

A scatter plot (also called a scatterplot, scatter graph, scatter chart, scattergram, or scatter diagram) is a type of plot or mathematical diagram using Cartesian coordinates to display values for typically two variables for a set of data. If the points are coded (color/shape/size), one additional variable can be displayed. The data are displayed as a collection of points, each having the value of one variable determining the position on the horizontal axis and the value of the other variable determining the position on the vertical axis.

# Import matplotlib libraries to visualize the data

import matplotlib.pyplot as plt

plt.scatter(data.studytime,data.G3)

plt.title("studytime vs G3")

plt.xlabel("Studytime")

plt.ylabel("G3")

plt.show()

**Correlation Matrix:**

A correlation matrix is simply a table which displays the correlation coefficients for different variables. The matrix depicts the correlation between all the possible pairs of values in a table. It is a powerful tool to summarize a large dataset and to identify and visualize patterns in the given data

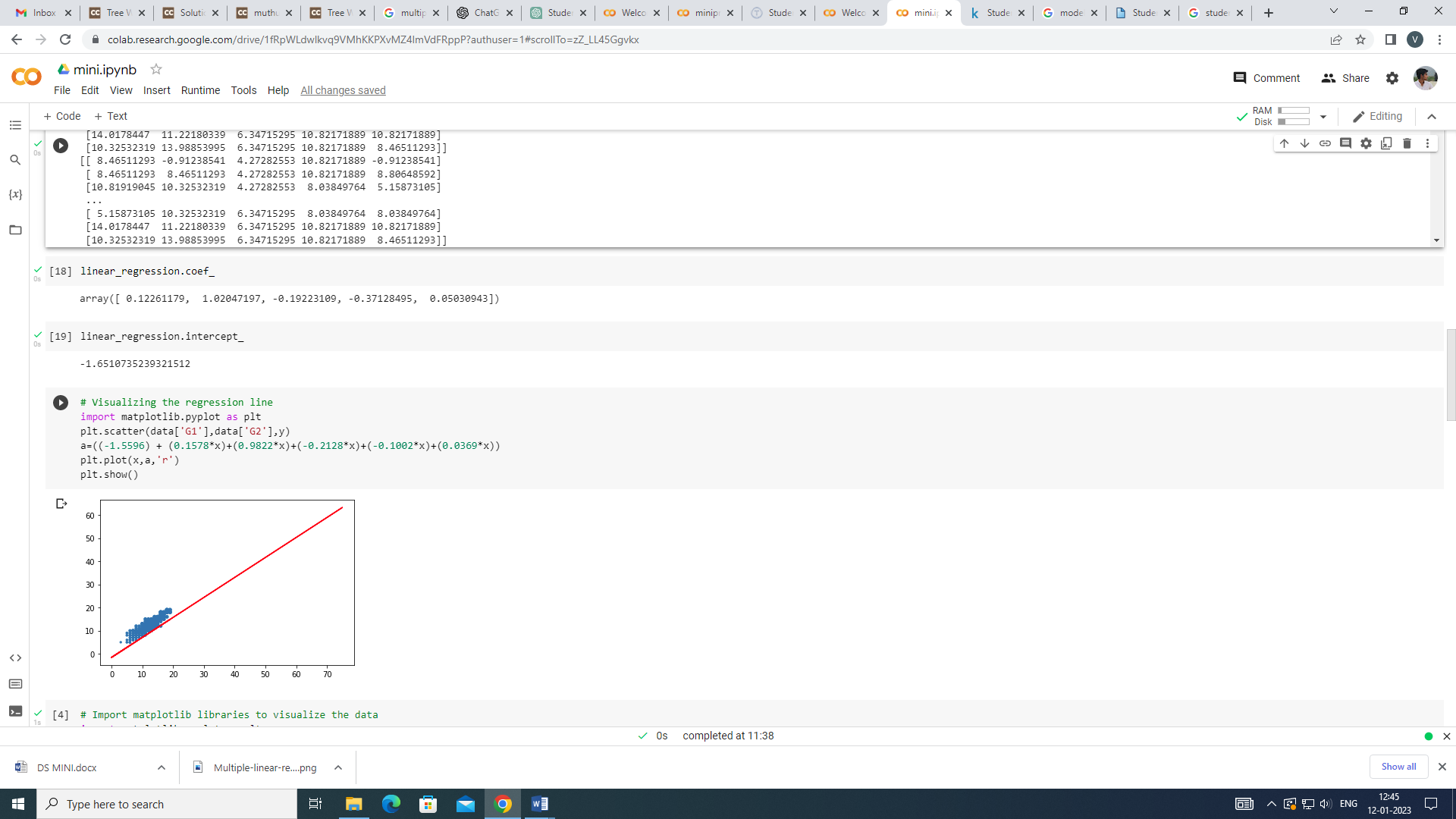
corr = data.corr()

corr.style.background\_gradient(cmap='coolwarm')

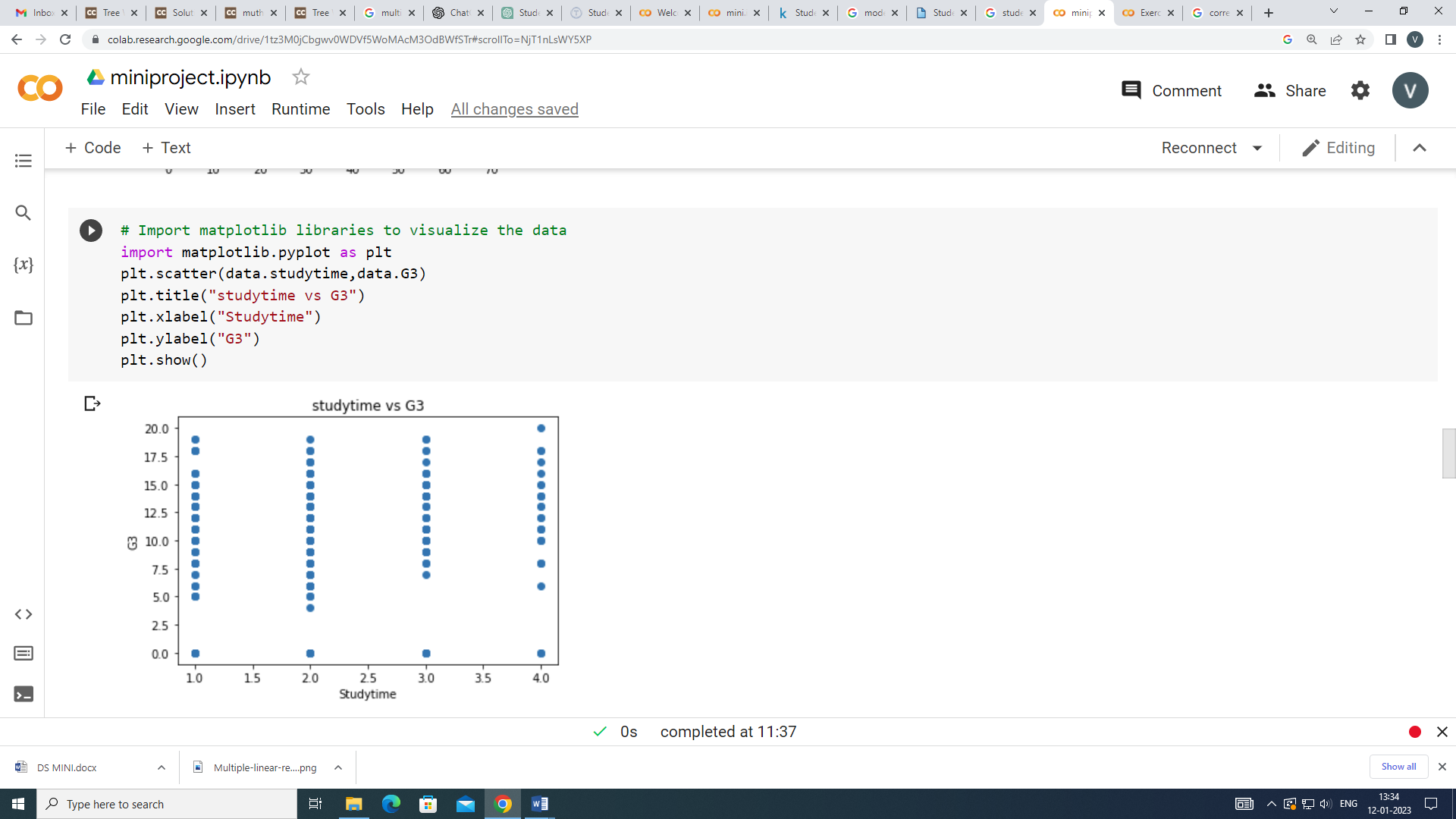
import seaborn as sns

plt.figure(figsize = (5,5))

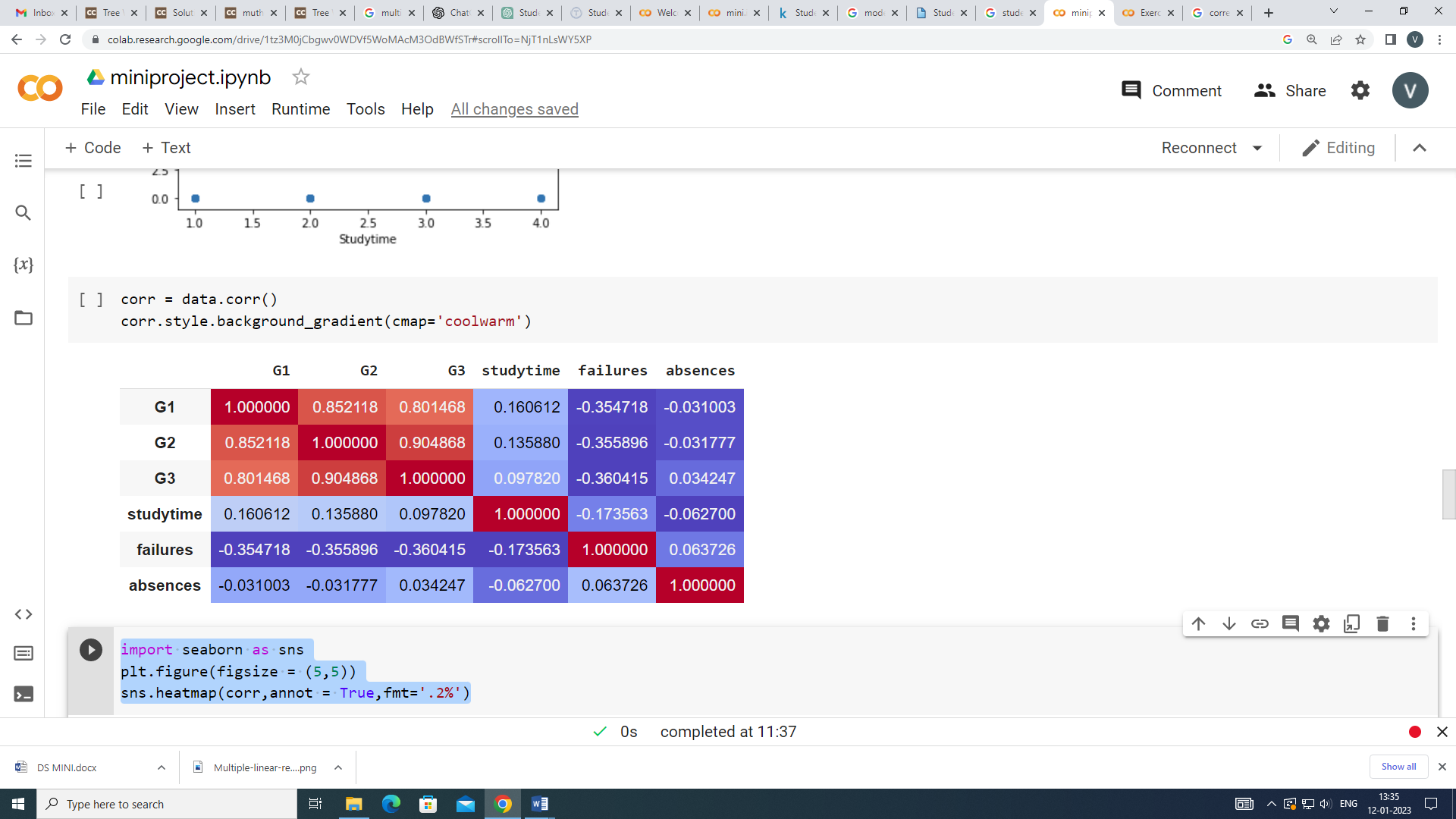
sns.heatmap(corr,annot = True,fmt='.2%')



**FIGURE 4: Scatter Plot –G1 vs G2**



**FIGURE 5: Scatter Plot-Study time vs G3**



**FIGURE 6: Correlation Heat map**



**FIGURE 7: Heat map**

**Predictions on the test set and Residual analysis:**

**RESIDUAL**

A residual is **a measure of how well a line fits an individual data point**. This vertical distance is known as a residual. For data points above the line, the residual is positive, and for data points below the line, the residual is negative. The closer a data point's residual is to 0, the better the fit.

y\_pred = linear\_regression.predict(xtest)

# Creating residuals from the y\_train and y\_pred

res = (ytest - y\_pred)

import warnings

warnings.filterwarnings('ignore')

import seaborn as sns

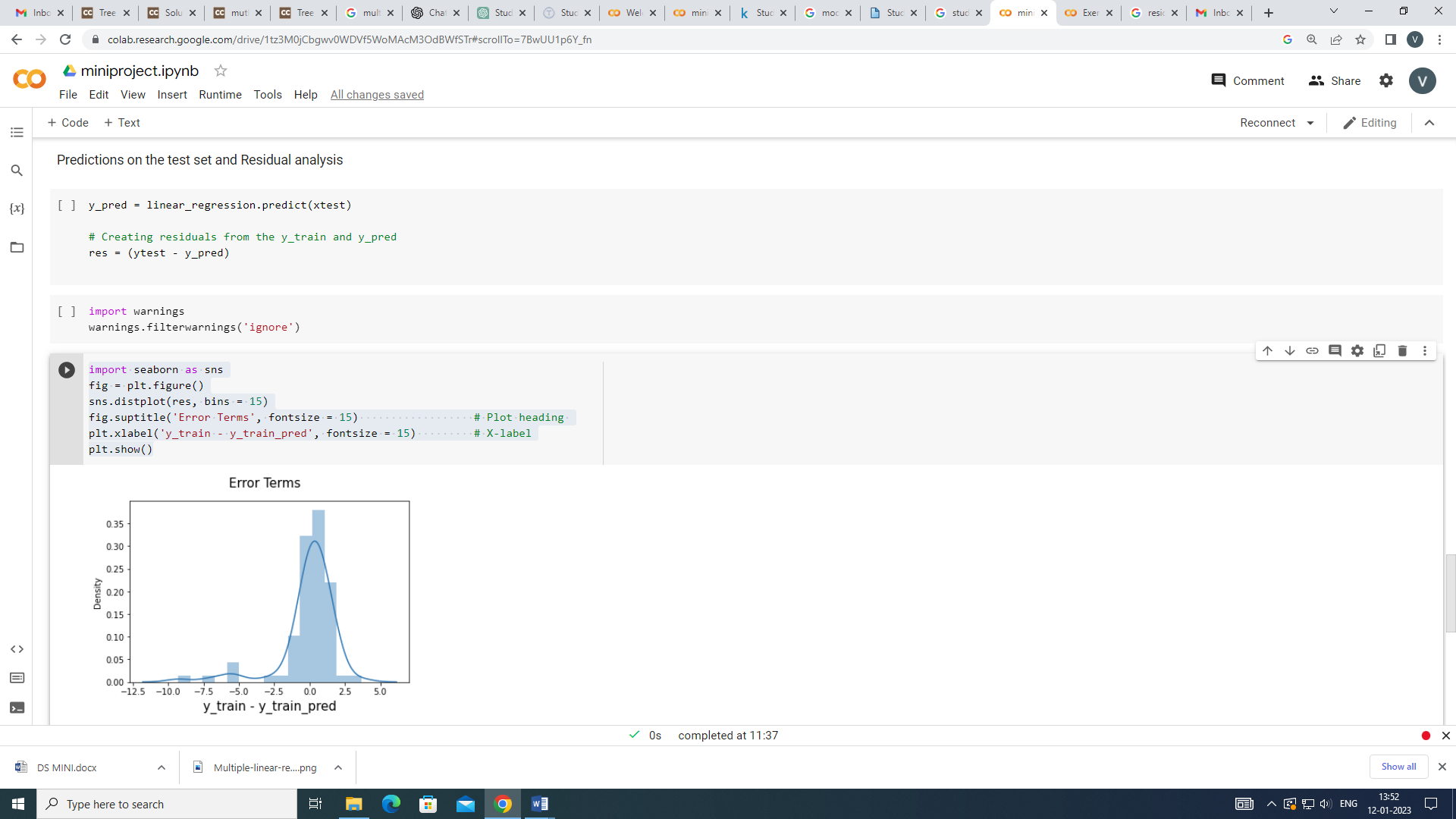
fig = plt.figure()

sns.distplot(res, bins = 15)

fig.suptitle('Error Terms', fontsize = 15)

plt.xlabel('y\_train - y\_train\_pred', fontsize = 15)

plt.show()



**FIGURE 8: Distribution Plot**

**EVALUATING MODEL**

Model evaluation is the process of using different evaluation metrics to understand a machine learning model's performance, as well as its strengths and weaknesses. Model evaluation is important to assess the efficacy of a model during initial research phases, and it also plays a role in model monitoring.

# Checking the R-squared value

r\_squared = r2\_score(y\_test, y\_pred)

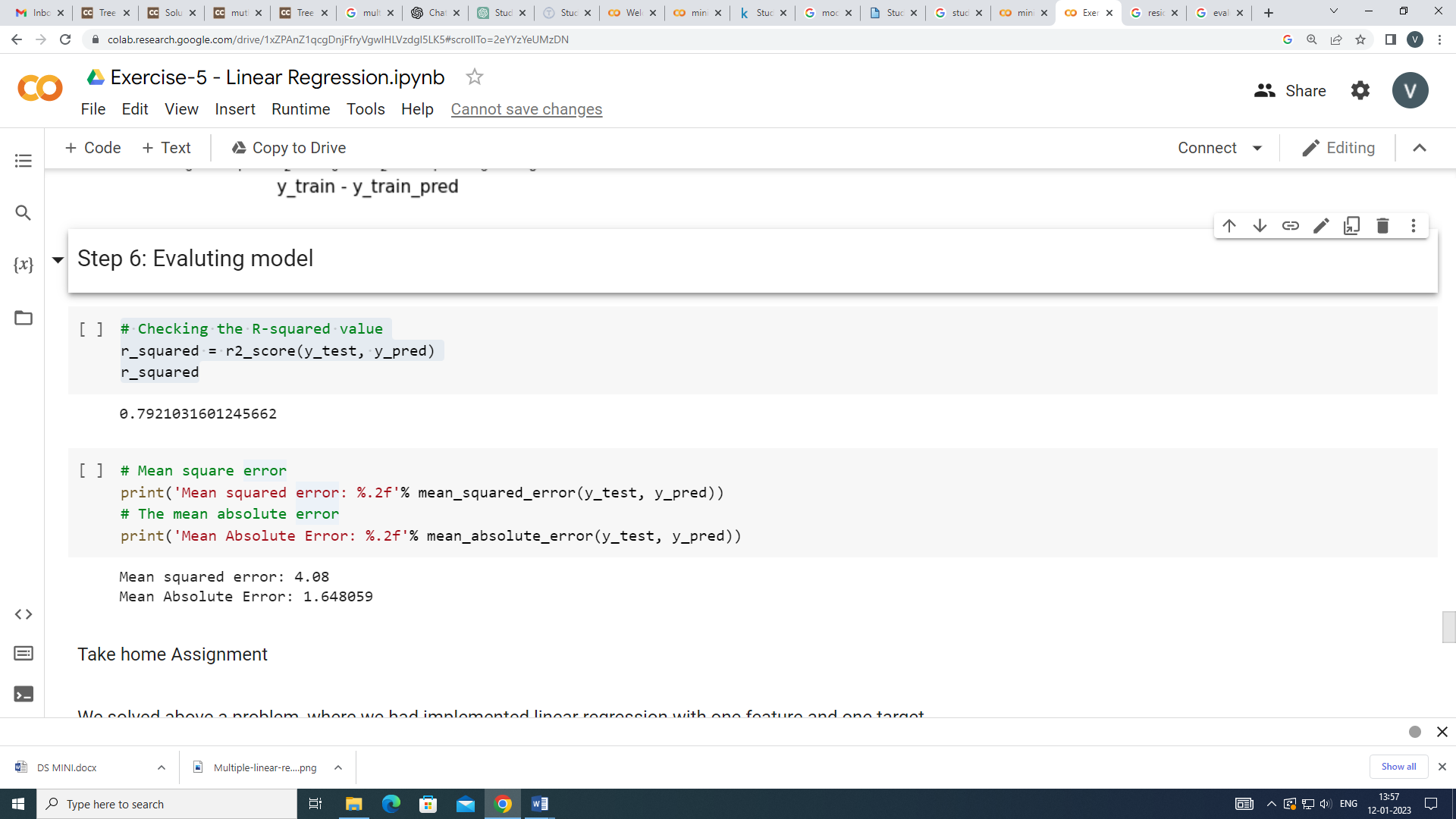
r\_squared

# Mean square error

print('Mean squared error: %.2f'% mean\_squared\_error(y\_test, y\_pred))

# The mean absolute error

print('Mean Absolute Error: %.2f'% mean\_absolute\_error(y\_test, y\_pred))



**FIGURE 9: Error prediction**

**5. CONCLUSION**

In this project we did a deep analysis of what could be possible factor on whether a student is likely to get a high score or a low score. The data does not contain that much information but still we were able to predict a pretty precise Linear Regression algorithm that predicts what score a student will get in the foreseen feature by analysing the features. It is to my understanding that the linear regression model is used to predict values with a given number of features. Here we learned how to tuned Hyper parameters in a more automatic way in order for our model to have better predictions when new features of students will come in and will let us know as data comes by if a student is most likely to pass the class or not.

**6.FUTURE SCOPE**

Data cleaning and analysis can be better done and other machine learning algorithms can be applied on the model to improve the accuracy. Increased dataset will give out more accurate predictions. To improve the results, a dataset with sufficient features and increase in quantity must be obtained. Further research must be conducted in enhancing the existing machine learning techniques to work in real time and develop an efficient model. Also, the models developed must be tested on data with different volumes to test its scalability and performance. In future work, the result of regression on balanced dataset can be studied by changing the data distribution. This can be done by selecting a sample of dataset or removing certain records to balance the type of data.