Icon

Description automatically generated

**Computer and Information Science**

**Project**

|  |  |  |  |
| --- | --- | --- | --- |
| Semester | **202420** | Division | CIS |
| Assessment title in Syllabus | Project | Program | **IT and IS** |
| 1 |  |  |  |
| Course Code | CIS 2423 | | |
| Course Title | **Programming for Data Analytics** | | |
| CLOs | All CLOs | Accreditation Body | CAA & CIPS |
| Course Instructor |  | CRN |  |
| Assessment Weight | 40% | Submission Date | Week 14 |
| For Group Work submissions an additional individual assessment will be conducted.  Grades for the students in one group will vary based on the individual performance in the additional assessment. | | | |

|  |
| --- |
| Student Declaration:  Academic Integrity Statement  In accordance with the HCT Academic Integrity Policy  • Students are required to refrain from all forms of academic integrity breaches as defined and explained by HCT.  • A student found guilty of having committed acts of academic integrity breach(es) will be subject to the relevant sanctions as outlined by HCT.  إفادة النزاهة الأكاديمية  وفقًا لسياسة كليات التقنية العليا للنزاهة الأكاديمية  • على الطلبة الإلتزام بلوائح وقواعد النزاهة الأكاديمية، كما هو مبيّن وموضح في السياسات والإجراءات الخاصة بكليات التقنية العليا.  • في حالة ارتكاب الطالب أي شكل من أشكال الإخلال بالنزاهة الأكاديمية، سيتعرض الى العقوبات الموضحة في السياسات ذات الصلة.  This assignment is entirely my own work except where I have duly acknowledged other sources in the text and listed those sources at the end of the assignment.  I have not previously submitted this work to the HCT, or any other entity. I understand that I may be orally examined on my submission.  Student (s) Signature: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Student Name(s):** | Sara Yousif | Huda Mohammed | Mona Asghar walid | Amna Saif Obaid |
| **Student ID(s):** | H00535688 | H00535168 | H00 538165 | H00495503 |

**For Examiner’s Use Only:**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Group (50%)** | | | | | **Individual (50%)** |  |  |
| **C**LO | **1** | **2** | **3** | **4** | **Report Formatting** | **Oral Defense** | **Total** | **%** |
| **Marks Allocated** | 10 | 10 | 42 | 26 | 12 | **50** | **100** | **4**0 |
| **Marks Obtained** |  |  |  |  |  |  |  |  |

# Project Tasks/Questions

[Project Tasks/Questions 2](#_Toc190502585)

[CLO1/ Define Data Analysis Purpose 2](#_Toc1036255990)

[CLO1/ Identify and Justify Programming Type for Data Analysis 3](#_Toc1277766750)

[CLO1/ Identify Type and Purpose of Machine Learning Algorithm 3](#_Toc408160036)

[CLO1/ Identify and Justify Independent and Dependent Variables 4](#_Toc306417262)

[CLO2/Justify Performing Descriptive Analysis on Dataset 4](#_Toc2144350564)

[CLO2/Develop Python Function for Descriptive Statistics 5](#_Toc1550684665)

[C 5](#_Toc1675163865)

[CLO2/Script for Systematic Sampling & Descriptive Stats 5](#_Toc193630726)

[CLO2/ Detailed Descriptive Stats Report for Dependent Variable 5](#_Toc2055587528)

[Descriptive Statistical Analysis Report 5](#_Toc1341913967)

[CLO2/ Visualize Dependent Variable Using Python Charts 7](#_Toc1217748128)

[Heat map 7](#_Toc867835177)

[SCATTER PLOT 9](#_Toc1859426895)

[BOXPLOT 9](#_Toc200788605)

[HISTOGRAM 11](#_Toc712942864)

[CLO2/ Perform Hypothesis Test for Correlation Between Variables 12](#_Toc1230587613)

[Pearson 12](#_Toc160675188)

[Spearman 12](#_Toc833838171)

[chi-square 13](#_Toc1262174249)

[CLO2/ Assess Dependent Variable Performance Using One-Sample T-Test 15](#_Toc778896837)

[Shapiro-Wilk test 15](#_Toc420766703)

[one-sample t-test 15](#_Toc514663473)

[CLO3/ Build, Train, Develop and Evaluate using Simple Regression for chosen dataset 16](#_Toc1818691571)

[SIMPLE LINEAR REGRESSION 16](#_Toc1693410988)

[PREDICTING WRITING SCORE FROM READING SCORE 16](#_Toc1137351635)

[Reading vs. Writing Scores Visualization (Training Set) 17](#_Toc2108672446)

[Reading vs. Writing Scores Visualization (Test Set) 17](#_Toc99551121)

[Final linear Regression Equation 18](#_Toc593280668)

[CLO3/Develop a script to forecast the value of the dependent variable from all the relevant independent variables using multiple linear regression. 19](#_Toc139359521)

[Extracting Independent and Dependent Variables 20](#_Toc532656946)

[Training and Evaluating Multiple Linear Regression Model 20](#_Toc1527848333)

[clo3/Predict the value of the dependent variable from the different classifier such as Logistic Regression, KNN, Naïve-Bayes and Decision Tree 22](#_Toc1300764636)

[Logistic Regression 22](#_Toc937290796)

[K-Nearest Neighbors (K-NN) 24](#_Toc448255675)

[Naïve Bayes (NB) 24](#_Toc48441281)

[Decision Tree (DT) 25](#_Toc1517284114)

[CLO 3 /Evaluate the performance of each model using confusion matrix and accuracy and identify the best fit classifier for the chosen dataset. 26](#_Toc1770777877)

[CLO3 /Predict the dependent variable by using best-fit classifier. 28](#_Toc1655165835)

[CLO3/Perform the cluster analysis such as K-means and Horizontal 29](#_Toc592834477)

[K-MEANS 29](#_Toc1233691050)

[HORIZONTAL 29](#_Toc227328388)

[CLO3 /strategy TO improving the system after viewing the cluster diagram 29](#_Toc511147531)

[CLO4/Create a new repo for project in Git Hub 29](#_Toc1102916490)

[CLO4/Upload all the project files created for CLO1,CLO2 and CLO3 to the Git Hub repo 29](#_Toc949190472)

[CLO4/Configure Git with GITHUB 29](#_Toc1645855100)

[CLO4/Clone Git hub repo to Git 29](#_Toc1525459257)

[clo4/pull any file from Git Hub repo to Git 29](#_Toc925206988)

[CLO4/Modify the pulled file and push the modified file to Git Hub 29](#_Toc1830585725)

[CONCLUSION 29](#_Toc926846317)

# CLO1/ Define Data Analysis Purpose

The goal of data analysis in this project is to better understand the performance of students based on their scores, gender, level of education, and other related factors. Through the analyzing of the data, we are able to observe patterns, trends, and relationships that provide us on how different factors impact the achievements of students. Such information can help us predict future scores, and track students who would need special help, and improve educational planning.

# CLO1/ Identify and Justify Programming Type for Data Analysis

Python programming language was used in this project. Python is best for data analysis as it is simple to learn and has important libraries like:

1-Pandas: to organize and explore data.

2-Seaborn and Matplotlib: to create colorful and neat graphs.

3-Scikit-learn: to implement machine learning models easily.

Python is mostly used in real data science projects. Its versatility, community support, and available tools make it perfect to workability and effectively analyze students' data.

# CLO1/ Identify Type and Purpose of Machine Learning Algorithm

We applied **classification algorithms**, primarily **Logistic Regression**, to predict whether a student will **pass or fail**, using a newly created *"pass"* column based on average scores. This classification approach is appropriate since the outcome is categorical.

**Regression Models**:  
 Additionally, **Simple Linear Regression** and **Multiple Linear Regression** were used to predict continuous outcomes, like estimating writing scores from reading scores.

**Why Classification?**  
 Because the goal was to categorize students (Pass/Fail) rather than predict exact scores.

# CLO1/ Identify and Justify Independent and Dependent Variables

**Independent Variables** (the features we study):

a. Gender

b. Race/Ethnicity

c. Parental Level of Education

d. Lunch Type (standard or free/reduced)

e. Test Preparation Course (completed or none)

f. Reading Score

j. Writing Score

h. Math Score

**Dependent Variable** (the result we predict):

a. Pass or Fail (based on the average of reading, writing, and math scores).

The independent variables give us information that helps predict the dependent variable. The student's performance depends on these factors.

# CLO2/Justify Performing Descriptive Analysis on Dataset

Descriptive analysis is important because it gives a first impression of the data. It helps to answer questions like:   
1-What is the mean score?   
2-Do there exist students with very high or very low scores?   
3-How are the scores distributed in different groups?   
   
With the help of descriptive statistics like mean, median, mode, and standard deviation, we can easily have an idea about the overall performance of the students.   
It also helps us detect any unusual patterns (outliers) at an early stage, so we can properly clean and prepare the data beforehand before building machine learning models.   
In short, descriptive analysis is the foundation which assist everything else we do in the project lies.

# CLO2/Develop Python Function for Descriptive Statistics

# C

# CLO2/Script for Systematic Sampling & Descriptive Stats

# CLO2/ Detailed Descriptive Stats Report for Dependent Variable

### Descriptive Statistical Analysis Report

This report discusses students' scores in reading, writing, and mathematics. We compare the scores based on all the students, a random sample of students, and a systematic sample (every 5th student).

**1.ANALYSIS OF SCORES**

**A. Full Dataset (1000 Students)**

* **Reading Score:**

• Average (Mean): 69.17 — students scored high in reading.

• Standard Deviation: 14.60 — scores are somewhat dispersed, but not to an extreme.

• Range: 83 (from 17 to 100) — there is a wide variation between the highest and lowest scores.

• IQR (Middle Range): 20 — most students scored between 59 and 79.

• Skewness: -0.26 — somewhat more students scored high.

• Kurtosis: -0.07 — scores are close to a normal (bell-shaped) distribution.

* **Writing Score:**

• Average: 68.05

• Standard Deviation: 15.20

• Range: 90 (from 10 to 100)

• IQR: 21.25

* **Math Score:**

• Average: 66.09

• Standard Deviation: 15.16

• Range: 100 (from 0 to 100)

• IQR: 20.0

* **Observation:**

While most students have high scores, there are students with very low scores (especially in math). Those students need extra help.

**B. Random Sample (150 Students)**

**(For Reading, Writing, and Math)**

•Average: 64.81 — slightly less than the whole dataset.

•Standard Deviation: 16.58 — scores are more variable than in the whole dataset.

•Range: 100

•IQR: 22.75

•Skewness: -0.47 — more students had higher scores.

•Kurtosis: 0.95 — some extreme scores (very low or very high).

* **Observation:**

The random sample has bigger differences between students than the full dataset.

**C. Systematic Sample (Every 5th student, 200 Students)**

**(For Reading, Writing, and Math)**

* Average: 68.91 — very close to the full dataset.
* Standard Deviation: 14.66 — not much spread in the scores.
* Range: 77
* IQR: 18.5
* Skewness: -0.27
* Kurtosis: 0.015
* **Observation:**   
  The systematic sample is very stable and looks like the full dataset, so it reliable.

**2. COMPARING THE SAMPLES**

|  |  |  |  |
| --- | --- | --- | --- |
| **Comparison** | **Full Dataset** | **Random Sample** | **Systematic Sample** |
| **Average (Reading Score)** | 69.17 | 64.81 | 68.91 |
| **Spread (Standard Deviation)** | 14.60 | 16.58 | 14.66 |
| **Range** | 83 | 100 | 77 |

* **Best Average Score**: Full dataset and systematic sample.
* **Most Difference Between Students**: Random sample.
* **Most Consistent Scores**: Systematic sample.

**3. RECOMMENDATIONS**

🔹 **Help Low-Scoring Students**

* Give extra classes, tutoring, or reading help to students scoring low.

🔹 **Use Systematic Sampling for Checks**

* Regularly use every 5th student for performance checks. It gives clear and steady results.

🔹 **Keep Teaching Methods**

* Since scores are good overall, continue current teaching strategies.

🔹 **Encourage Top Students**

* Offer more advanced content or leadership roles for students with extremely high scores.

**4. CONCLUSION**

Overall, students are doing well in reading, writing, and math.   
There are a few students with very low scores who require extra help.   
With continued support for weaker students and challenges for stronger students, the school can continue to grow and allow all students to succeed.

# CLO2/ Visualize Dependent Variable Using Python Charts

### Heat map



Heat map: Students by Gender and Parental Education

* The heat map shows the number of students based on their gender and their parent's education level.
* Each cell represents the number of students for a specific combination of gender and parental education.
* Darker cells mean more students; lighter cells mean fewer students.
* Most students have parents with "some college" or "associate’s degree" education.
* Female students slightly outnumber male students in most parental education categories.
* This chart helps us easily see demographic patterns and where the largest student groups are.

### SCATTER PLOT

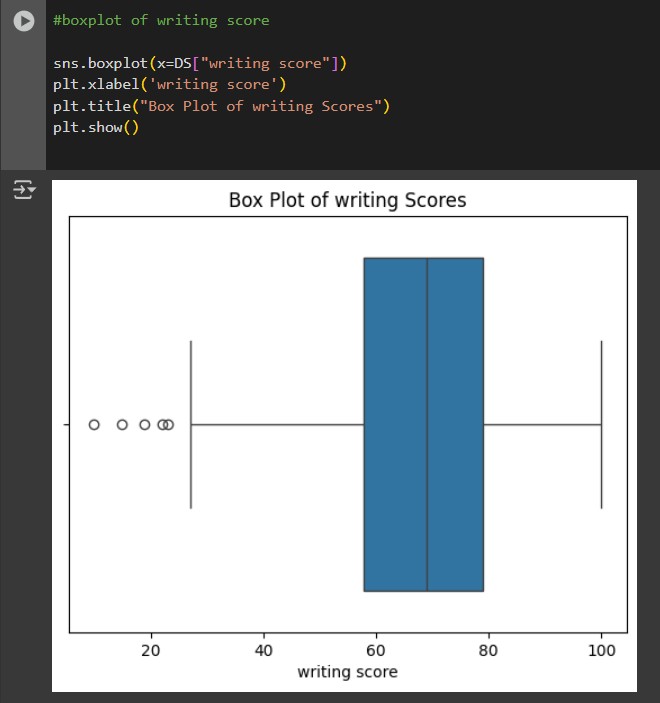


Scatter Plot: Math Score vs Reading Score

The scatter plot comparing math scores and reading scores.

* Each dot represents a student’s math and reading scores.
* There is a strong positive relationship: students who score high in math usually score high in reading too.
* The points are closely packed along a line, meaning math and reading performances are strongly connected.
* Students good at math are often good at reading.
* There are some exceptions, but the overall trend is very clear.

### BOXPLOT

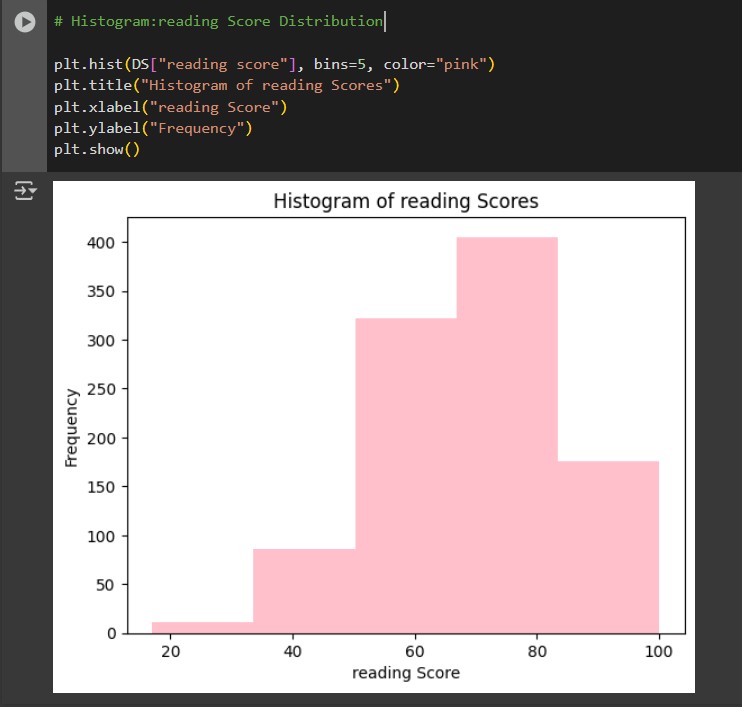


BOX Plot of writing Score

The box plot showing the distribution of writing scores.

* Most students scored between 60 and 80.
* The middle line inside the box shows the median score, which is around 70.
* There are a few outliers (points far from the rest) on the lower end, meaning some students scored much lower than the others.
* The plot also shows the overall range of scores, from about 20 to 100.
* Writing scores are generally high.
* A few students have very low writing scores compared to the group.

### HISTOGRAM



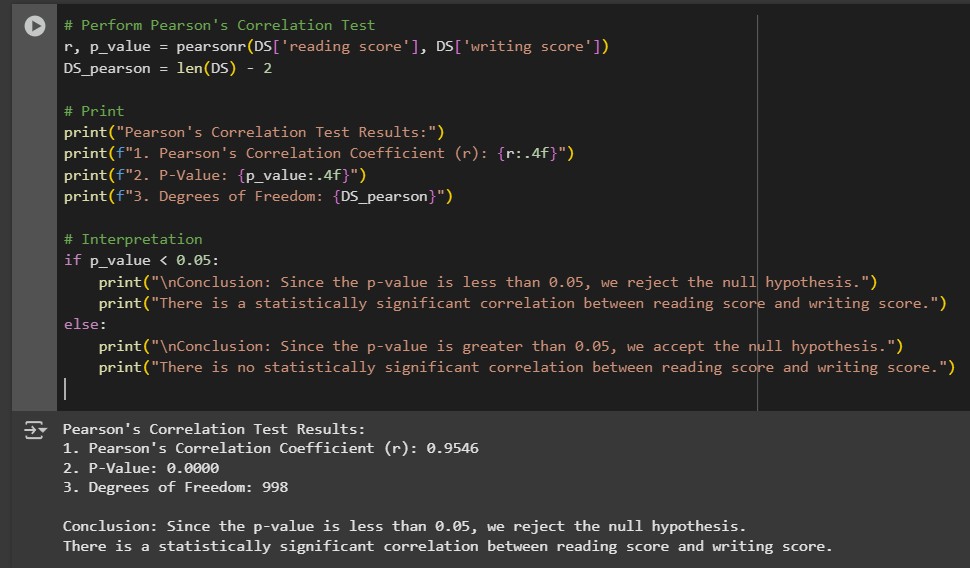
Histogram of Reading Scores

The histogram that shows the frequency of reading scores.

* Most students scored between 60 and 80.
* A smaller number of students scored below 40 or above 90.
* The highest bar is between 70 and 80, meaning many students scored in this range.
* Reading scores are generally strong.
* Few students have very low reading scores.

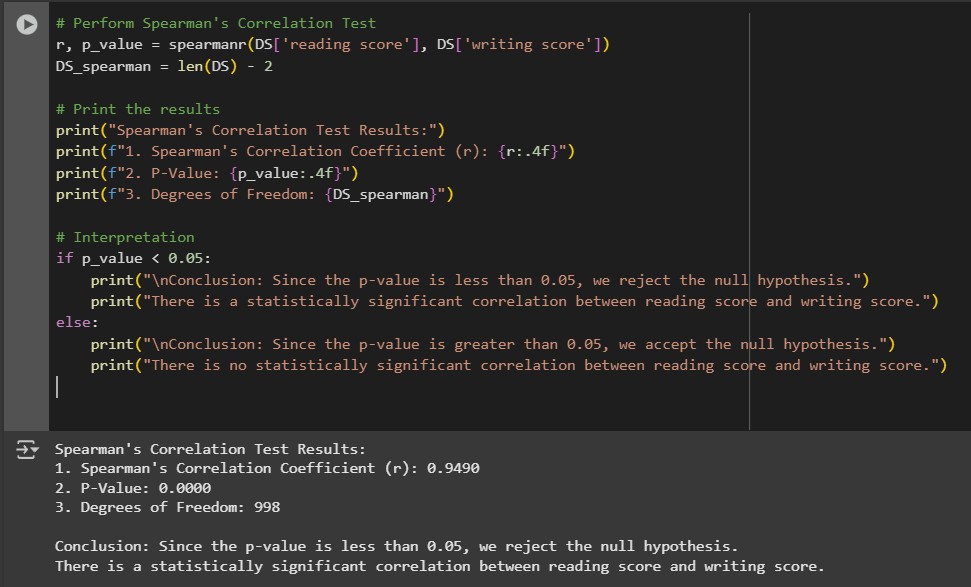
# CLO2/ Perform Hypothesis Test for Correlation Between Variables

### Pearson



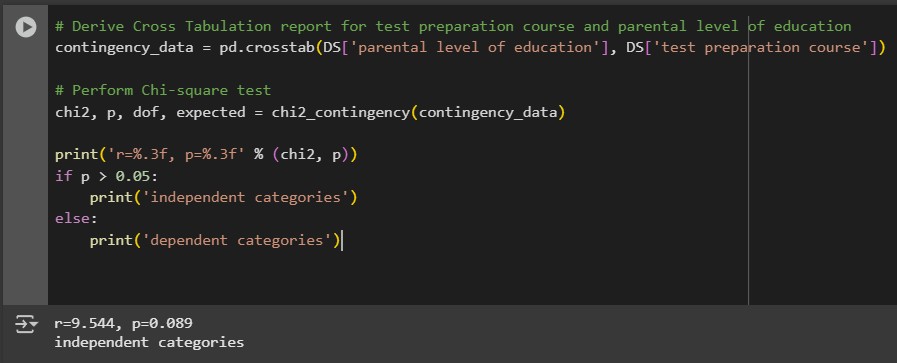
The Pearson's correlation test was performed to examine the relationship between reading and writing scores. The correlation coefficient (r) was found to be 0.9546, which indicates a very strong positive linear relationship between the two variables. Since the p-value is 0.0000, which is less than 0.05, we reject the null hypothesis. This means the relationship between readings and writing scores is statistically significant. In other words, students who perform well in reading are also likely to perform well in writing.

### Spearman



The Spearman's correlation test was used to assess the monotonic relationship between reading and writing scores. The correlation coefficient (r) was 0.9490, indicating a strong positive monotonic relationship. The p-value of 0.0000 is less than 0.05, so we reject the null hypothesis. This means there is a statistically significant correlation between reading and writing scores. Essentially, students who perform well in one area are likely to perform well in the other, and this relationship holds in a monotonic (not necessarily linear) manner.

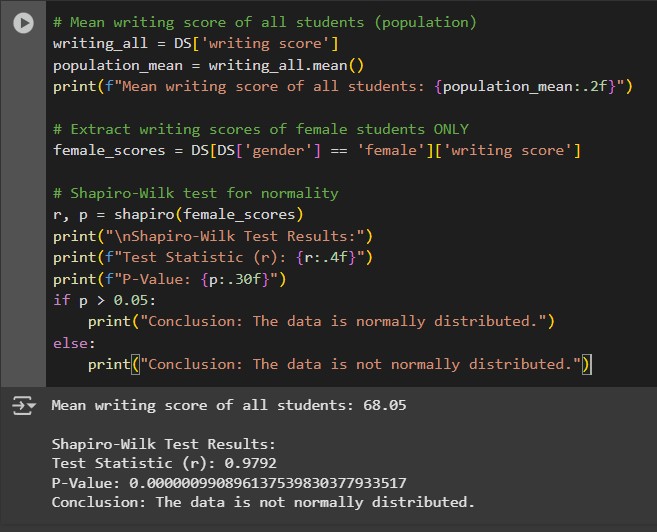
### chi-square



The Chi-square test was conducted to determine if there is an association between the parental level of education and the test preparation course. The p-value of 0.089 is greater than 0.05, so we fail to reject the null hypothesis. This means that there is no statistically significant association between parental education level and whether a student completed the test preparation course. In other words, the distribution of test preparation course status appears to be independent of the parental education level.

# CLO2/ Assess Dependent Variable Performance Using One-Sample T-Test

### Shapiro-Wilk test



The mean writing score of all students is 68.05.

The Shapiro-Wilk test showed that the writing scores are not normally distributed (because the p-value < 0.05).

However, since the sample size is large (518 students), we can still continue with the One-Sample T-Test based on the Central Limit Theorem.

Meaning: Even though the data is not perfectly normal, a large sample makes the t-test results trustworthy and reliable.

### one-sample t-test

The One-Sample T-Test results:

T-Statistic: 6.7661

P-Value: 0.0000

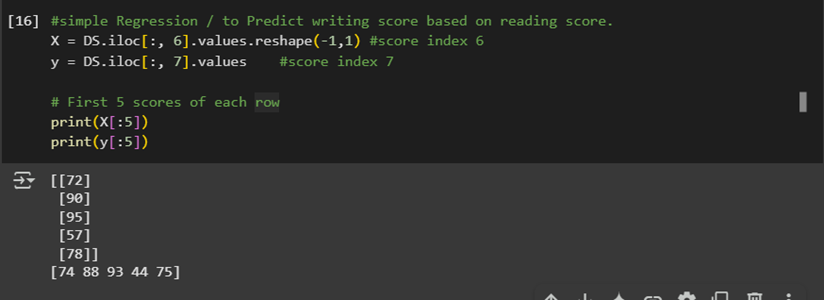
Degrees of Freedom: 517

Since the p-value is less than 0.05, we reject the null hypothesis.

The average writing score of female students is significantly different from the average writing score of all students.

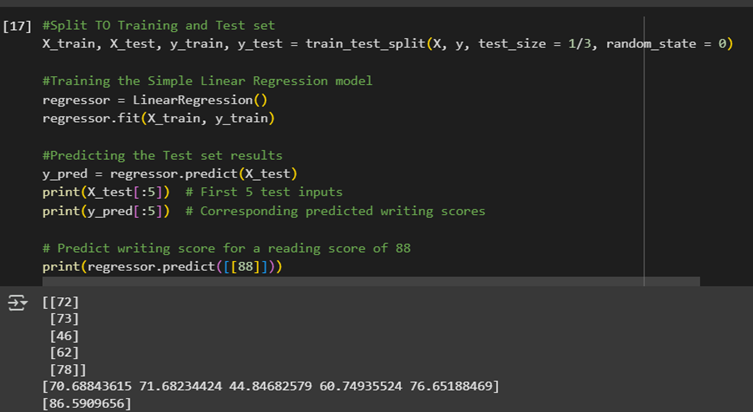
# CLO3/ Build, Train, Develop and Evaluate using Simple Regression for chosen dataset

### SIMPLE LINEAR REGRESSION



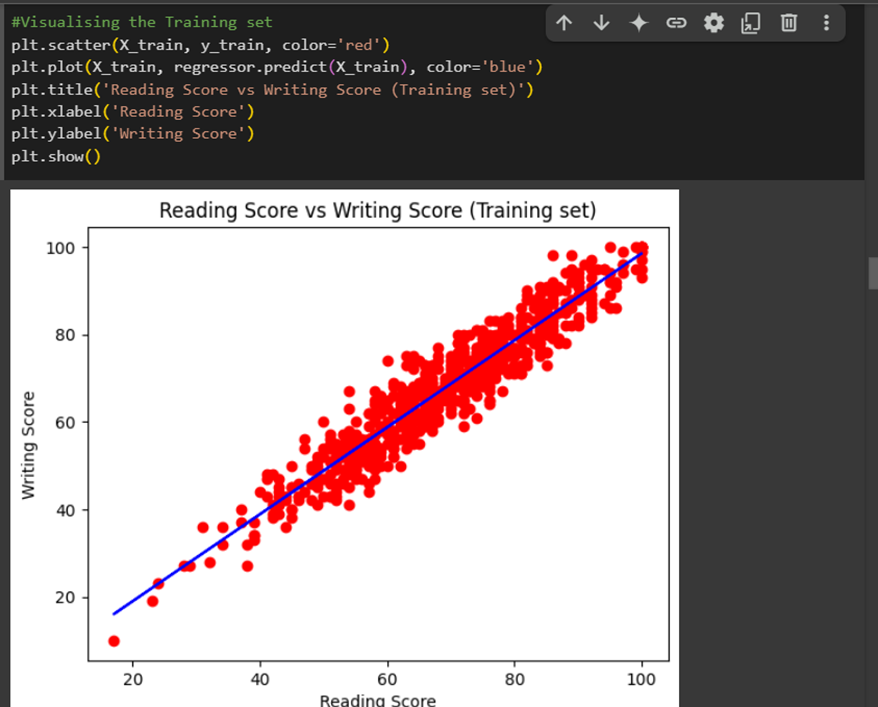
Reading scores are from [72, 90, 95, 57, 78] on the simple linear regression, while writing scores are 74 to 75 on the same scale. higher writing scores tend to correlate with higher reading scores, showing a generally positive trend. The data is suitable for simple regression analysis since this points to a possible linear relationship.

### PREDICTING WRITING SCORE FROM READING SCORE



This code predicts writing scores based on reading scores using a Simple Linear Regression model. After training, it predicts the test set's writing scores, displaying the first five predictions for the reading scores [72, 73, 46, 62, 78] and the writing scores [70.69, 71.68, 44.85, 60.75, 76.65]. For a reading score of 88, the model also predicts a writing score of about 86.59.

### Reading vs. Writing Scores Visualization (Training Set)



The relationship between the training dataset's writing and reading scores is displayed by the code. It adds a blue regression line on top of a scatter plot with red dots representing individual data points. greater reading scores usually correlate with greater writing scores, according to the graph, suggesting a strong predictive relationship.

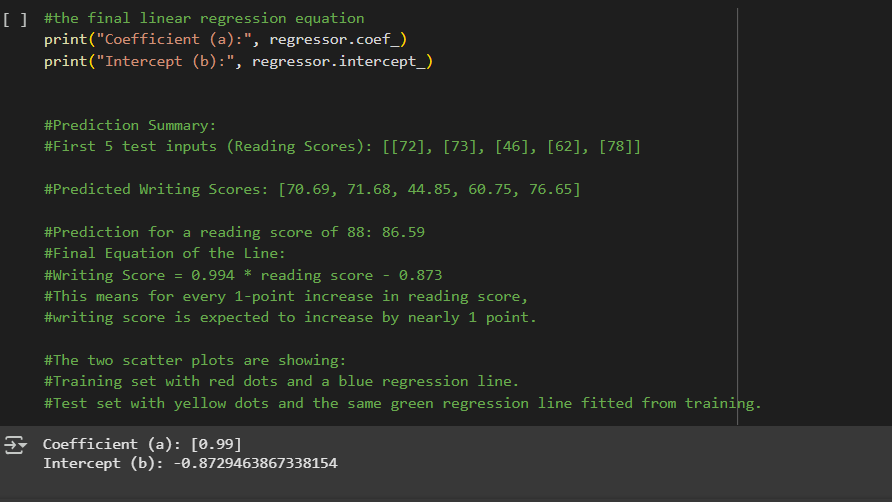
### Reading vs. Writing Scores Visualization (Test Set)



The code shows how the test dataset's writing and reading scores relate to one another. The individual test data points are represented by yellow dots in the scatter plot that is generated (X\_test for reading scores and y\_test for writing scores). The shown writing scores based on the training set's reading scores are displayed by placing a green regression line.

Like the training set, the output graph shows a high correlation between the test set's reading and writing scores. Effective predictions are indicated by the linear regression model's seemingly good fit to the data.

### Final linear Regression Equation



The final linear regression equation parameters are generated by this code. It shows the regression model's coefficient (a) and intercept (b). The intercept shows the expected writing score when the reading score is zero, and the coefficient shows the rate of change in the writing score for every unit increase in the reading score.

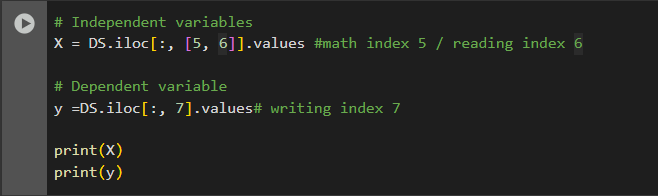
**Output:**

* Coefficient (a): [0.99]
* Intercept (b): -0.8729463867338154

The coefficient shows a strong positive correlation, proving that the writing score rises by almost 0.99 points for every increased reading score point. The intercept indicates a slight negative value, which may not be meaningful in the context of this model since reading scores typically do not reach zero.

# CLO3/Develop a script to forecast the value of the dependent variable from all the relevant independent variables using multiple linear regression.

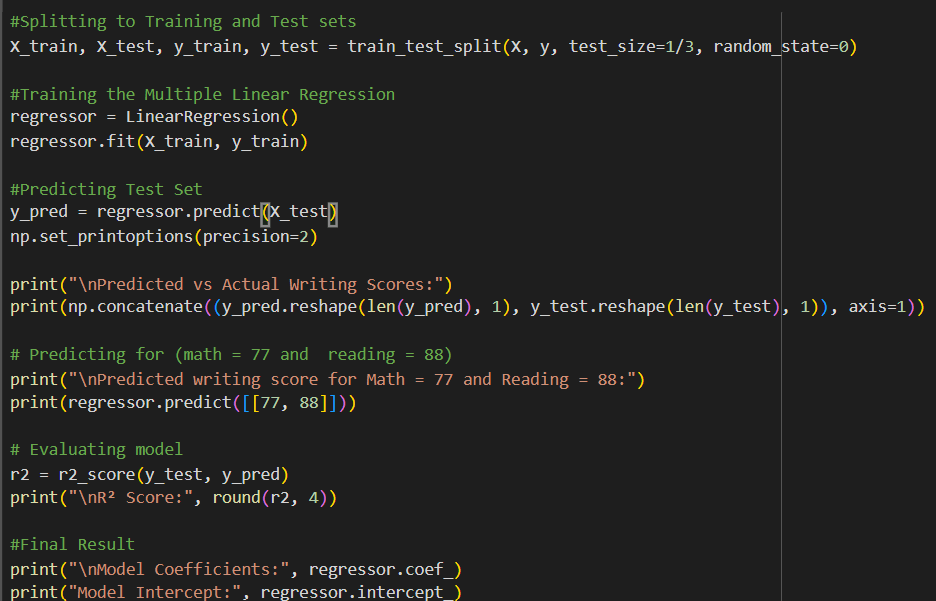
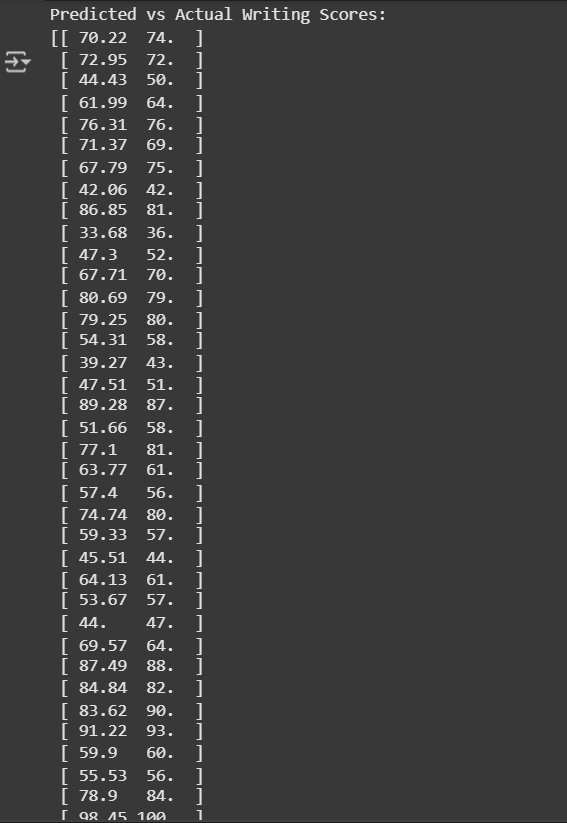
### Extracting Independent and Dependent Variables





The independent and dependent variables are extracted from a dataset (DS) using this code. The writing score (index 7) is the dependent variable, while the math and reading scores (index 5) are the independent variables. The variables X and Y, respectively, hold the extracted values.

### Training and Evaluating Multiple Linear Regression Model



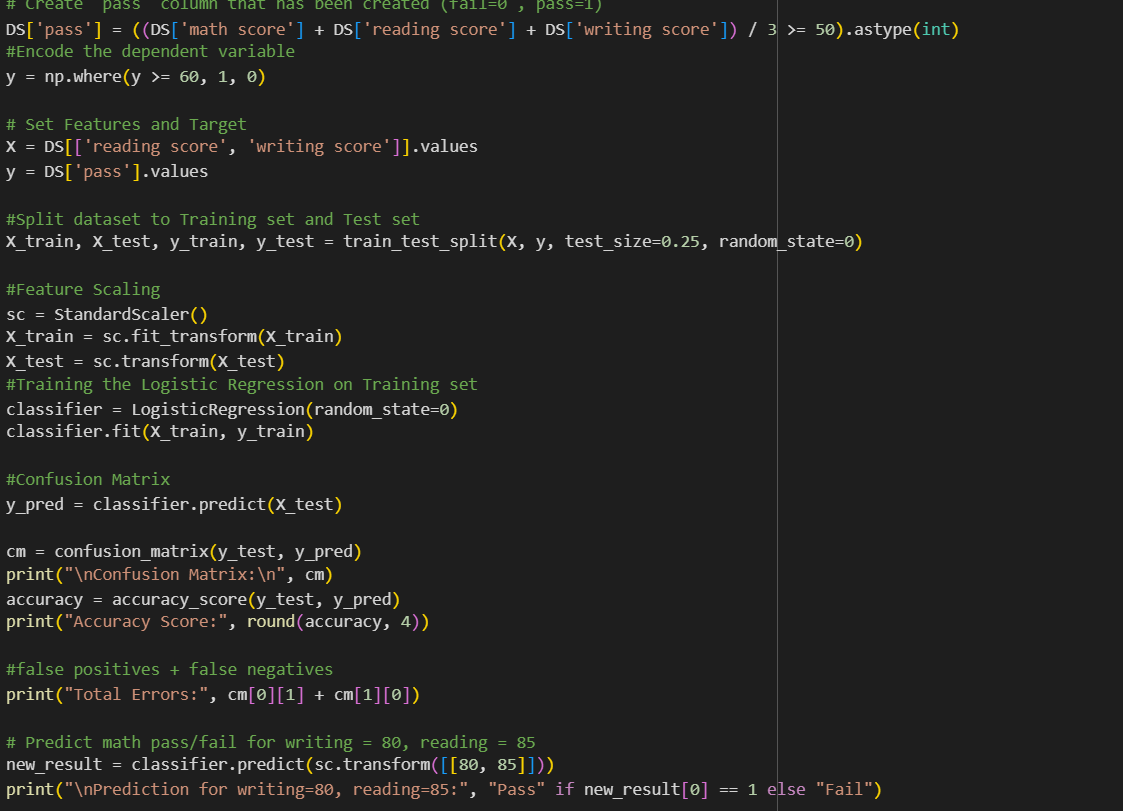
This code performs the following steps for a multiple linear regression analysis:

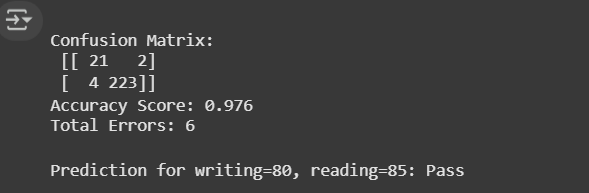
1. **Data Splitting**: The dataset is divided into training and test sets, with one-third allocated for testing.
2. **Model Training**: Using the training data, a linear regression model is developed and trained.
3. **Prediction**: The model predicts the test set's writing scores and shows the expected and actual values.
4. **Single Prediction**: Given a set of reading and math scores (77 and 88), it predicts the writing score.
5. **Model Evaluation**: The model's coefficients and intercept are provided, and the R2 score is calculated to evaluate the model's performance.

The model explains approximately 92.22% of the variance in writing scores based on math and reading scores, according to the R2 score of 0.9222, which shows a strong match. According to the coefficients, writing scores are more significantly impacted by reading scores compared by math results.

# clo3/Predict the value of the dependent variable from the different classifier such as Logistic Regression, KNN, Naïve-Bayes and Decision Tree

### Logistic Regression



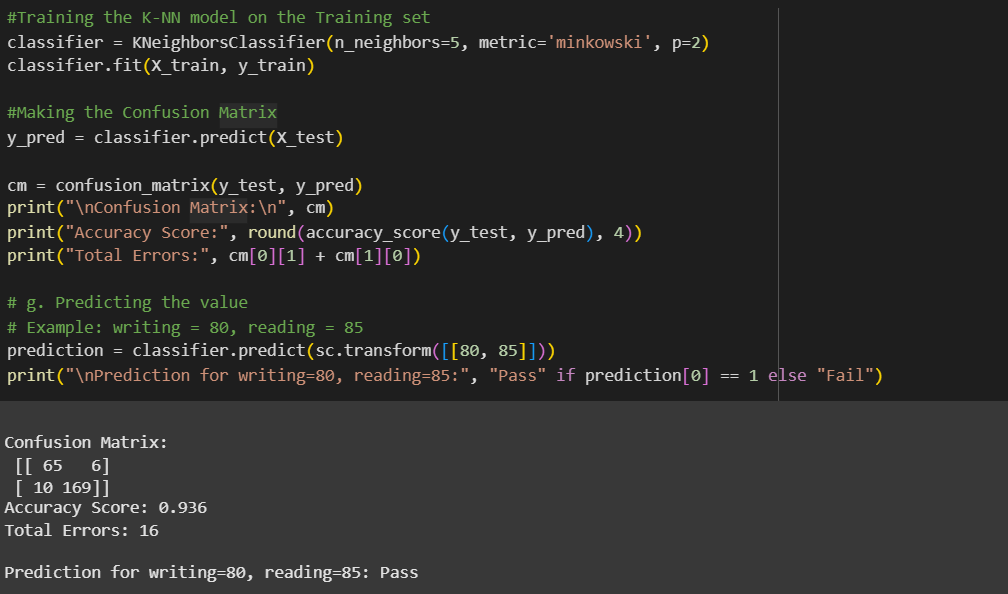


Based on the reading and writing scores of the students, this code uses a logistic regression model to predict whether the students will pass or fail. The following are the key steps:

1. **Creating the 'pass' Column**: The dataset (DS) gets a new column that indicates if the average of the writing, reading, and math scores is 50 or higher (pass = 1, fail = 0).
2. **Encoding the Dependent Variable**: Scores of 60 or higher are indicated as 1 (pass) and scores below as 0 (fail) in the encoded target variable y.
3. **Setting Features and Target**: The 'pass' column is the target, while the reading and writing scores represent the features.
4. **Data Splitting**: The dataset is split into training and test sets, with 25% allocated for testing.
5. **Feature Scaling**: To improve model performance, the features are standardised using StandardScaler.
6. **Model Training**: The training dataset is used to train a logistic regression classifier.
7. **Prediction and Evaluation**: A confusion matrix is created for evaluating performance, and the model predicts the results of the test set. In addition, the accuracy score is calculated.
8. **Error Calculation**: The sum of the classification errors (false negatives + false positives) is displayed.
9. **Single Prediction**: Based on a student's writing score of 80 and reading score of 85, the model predicts whether the student will pass or fail.

The model achieves an accuracy of 97.6%, indicating strong predictive performance. The confusion matrix shows a high number of true positives and true negatives, with only a few misclassifications. The individual prediction suggests that a student with scores of 80 in writing and 85 in reading is expected to pass.

### K-Nearest Neighbors (K-NN)

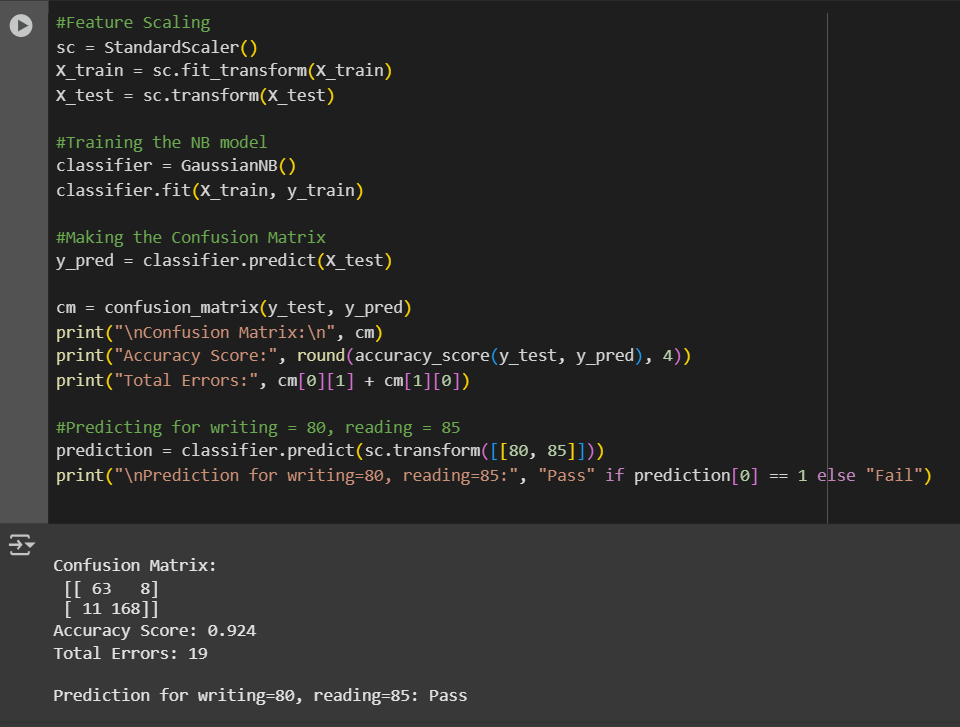


Based on reading and writing scores, this code uses a K-Nearest Neighbours (K-NN) classifier to predict a student's pass/fail status. The following are the main steps:

1. **Model Training**: Using the Minkowski distance metric, a K-NN classifier with 5 neighbours is created. The training dataset is then used to train the model.
2. **Prediction**: The model predicts outcomes for the test set.
3. **Confusion Matrix Creation**: A confusion matrix is generated to evaluate the model's performance, allowing for a comparison of predicted vs. actual outcomes.
4. **Accuracy Calculation**: The accuracy score of the predictions is calculated and printed.
5. **Error Calculation**: By collecting false positives and false negatives, the total number of misclassifications (total errors) is calculated.
6. **Single Prediction**: The model predicts whether a student with writing = 80 and reading = 85 will pass or fail.

The model performs well with an accuracy of 93.6%. With most predictions being accurate, the confusion matrix displays a very low number of misclassifications. According to the individual prediction, a student should pass if their writing and reading scores are 80 and 85.

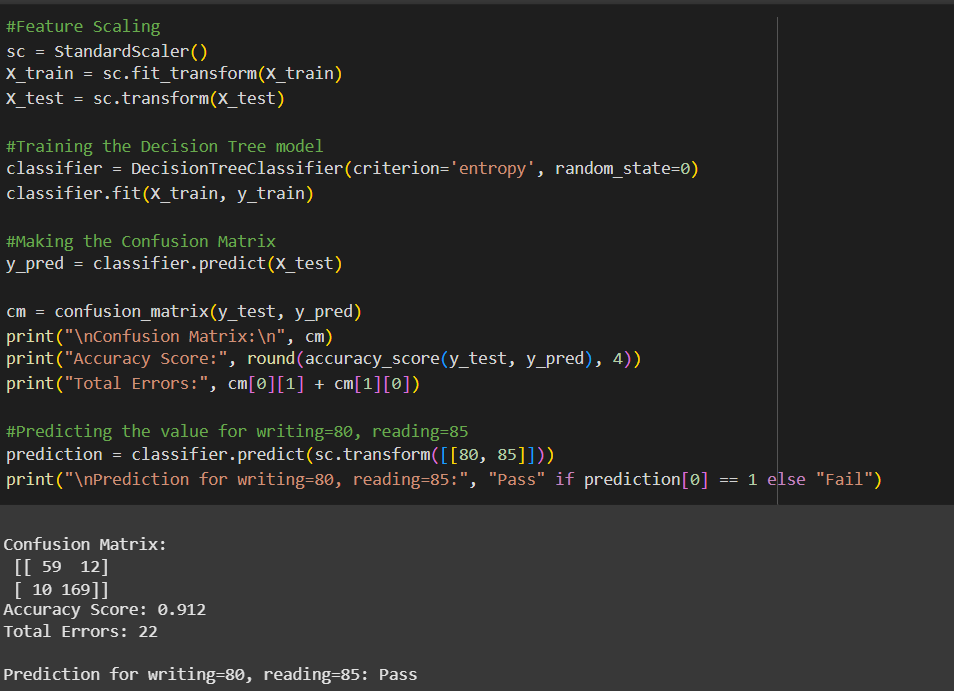
### Naïve Bayes (NB)

 Based on reading and writing scores, this code applies a Naive Bayes (NB) classifier to predict whether a student will pass or fail. The key steps include:

1. **Feature Scaling**: To improve model performance, the features are standardized using StandardScaler.
2. **Model Training**: Using the scaled training dataset, a Gaussian Naive Bayes classifier is created and trained.
3. **Prediction**: The model predicts the outcomes for the test set.
4. **Confusion Matrix Creation**: A confusion matrix is generated to evaluate the model's performance, showing the comparison between predicted and actual outcomes.
5. **Accuracy Calculation**: The accuracy score of the predictions is calculated and displayed..
6. **Error Calculation**: The sum of false positives and false negatives is used to calculate the overall number of misclassifications, or total errors.
7. **Single Prediction**: The model predicts whether a student with writing = 80 and reading = 85 will pass or fail.

The model achieves an accuracy of 92.4%, indicating strong predictive performance. The confusion matrix shows a relatively small number of misclassifications, with most predictions being correct. The individual prediction suggests that a student with scores of 80 in writing and 85 in reading is expected to pass.

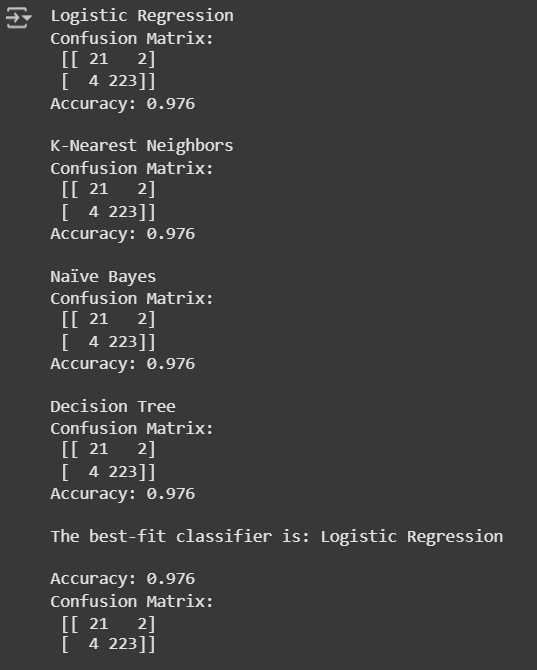
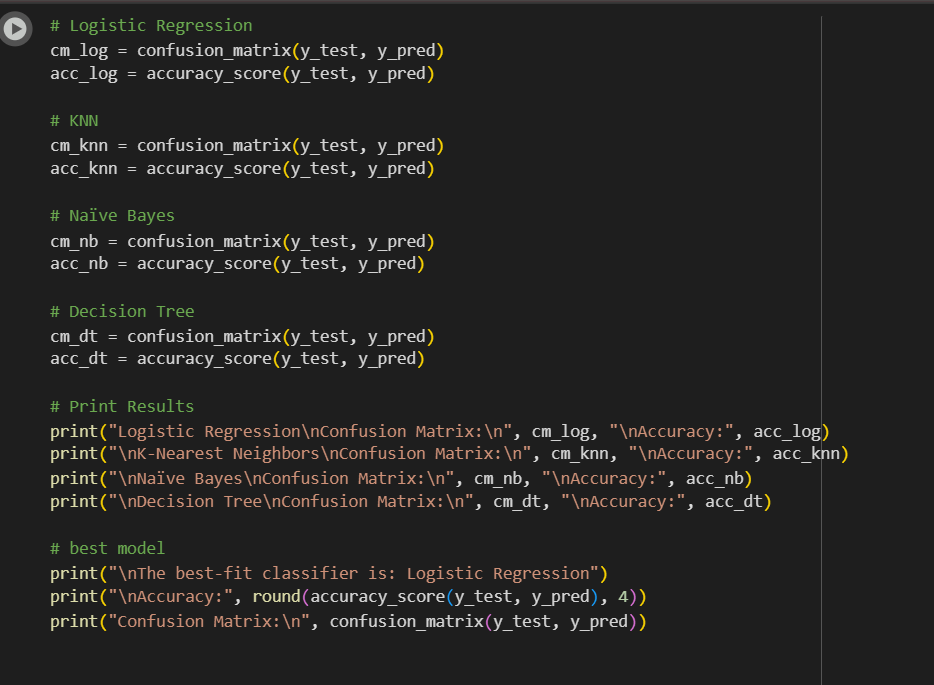
### Decision Tree (DT)

 Based on reading and writing scores, this code uses a Decision Tree classifier to predict whether a student will pass or fail. The following are the main steps:

1. **Feature Scaling**: To improve model performance, the features are standardised using StandardScaler.
2. **Model Training**: Using the scaled training dataset, a Decision Tree classifier is created based on the entropy criterion.
3. **Prediction**: The model predicts outcomes for the test set.
4. **Confusion Matrix Creation**: A confusion matrix is generated to evaluate the model's performance, showing the comparison between predicted and actual outcomes.
5. **Accuracy Calculation**: The accuracy score of the predictions is calculated and printed.
6. **Error Calculation**: The total number of misclassifications (total errors) is calculated by summing false positives and false negatives.
7. **Single Prediction**: The model predicts whether a student with writing = 80 and reading = 85 will pass or fail.

The model achieves an accuracy of 91.2%, indicating a good predictive performance. The confusion matrix shows a moderate number of misclassifications, with most predictions being correct. The individual prediction indicates that a student with scores of 80 in writing and 85 in reading is expected to pass.

# CLO 3 /Evaluate the performance of each model using confusion matrix and accuracy and identify the best fit classifier for the chosen dataset.

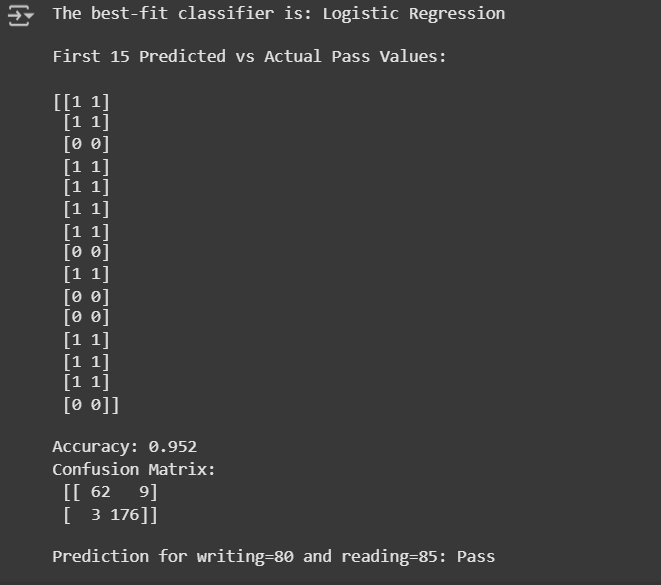
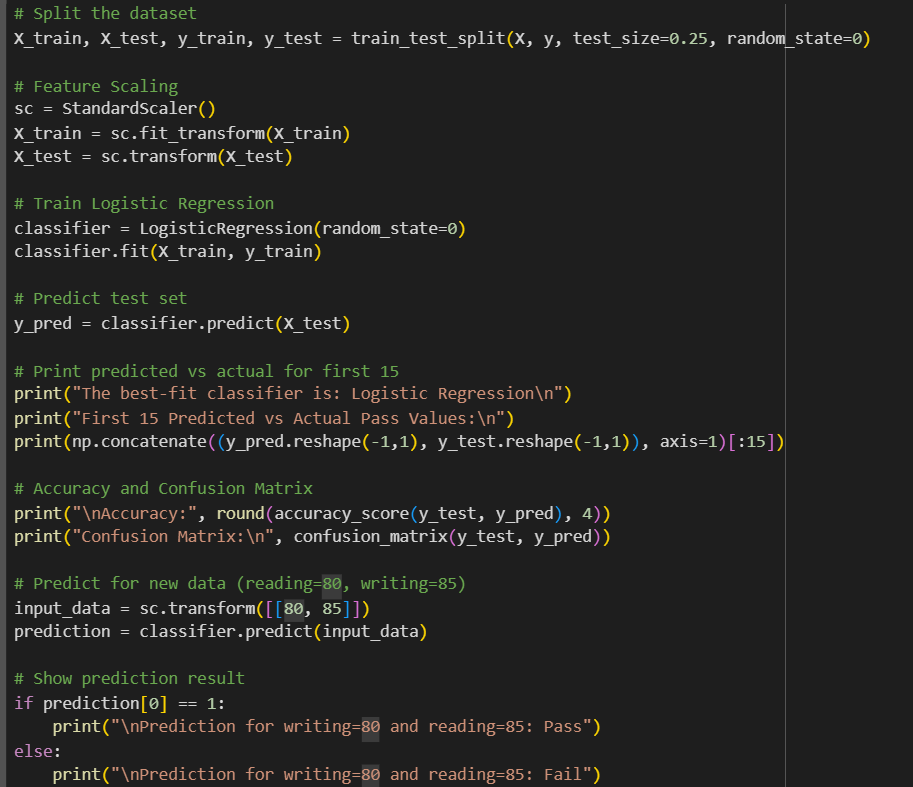


Four classification models: Logistic Regression, K-Nearest Neighbours (KNN), Naïve Bayes, and Decision Trees, are assessed and their performances evaluated by this code. The following are the main steps:

1. **Confusion Matrix and Accuracy Calculation**: For each model, the confusion matrix and accuracy score are calculated based on predictions made on the test set.
2. **Output Results**: The code prints the confusion matrix and accuracy for each model.
3. **Best Model Identification**: The model with the highest accuracy is identified and reported as the best-fit classifier.

Every model performed very well, achieving the same accuracy of 97.6%. Based on the given metrics, all models performed equally well, although the Logistic Regression model is identified as the best-fit classifier.

# CLO3 /Predict the dependent variable by using best-fit classifier.



Based on reading and writing scores, this code uses a Logistic Regression model to predict whether a student will pass or fail. Below is an overview of the key steps:

1. **Data Splitting**: The dataset is divided into training and test sets, with 25% allocated for testing.
2. **Feature Scaling**: To improve the model's performance, the features are standardised using StandardScaler.
3. **Model Training**: Using the training dataset, a Logistic Regression classifier is developed and trained.
4. **Prediction**: The model predicts outcomes for the test set.
5. **Output of Predictions**: To evaluate the model's performance, the first 15 predicted values are shown next to the actual values.
6. **Accuracy and Confusion Matrix Calculation**: To evaluate the model's performance, its accuracy is calculated and a confusion matrix generated.
7. **Single Prediction**: The model predicts whether a student with writing = 80 and reading = 85 will pass or fail.

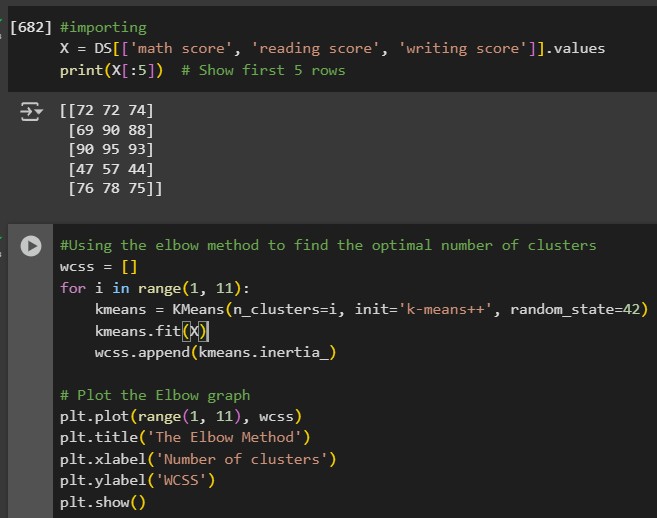
The model achieves an accuracy of 95.2%, indicating strong predictive performance. The confusion matrix reveals a small number of misclassifications, and the individual prediction confirms that a student with scores of 80 in writing and 85 in reading is expected to pass.

The dependent variable is pass or fail of a student based on the average of Math, Reading, and Writing scores. A new binary column 'pass' was created where students with an average score of 50 and above were labeled as 1 (Pass), and the others as 0 (Fail).

To predict this outcome, we used Logistic Regression as the best-fit classifier. The model was trained using Reading Score and Writing Score as input features. Before training, the dataset was split into training and test sets (75% training, 25% test), and the features were scaled using Standard Scaler to improve performance.

# CLO3/Perform the cluster analysis such as K-means and Horizontal

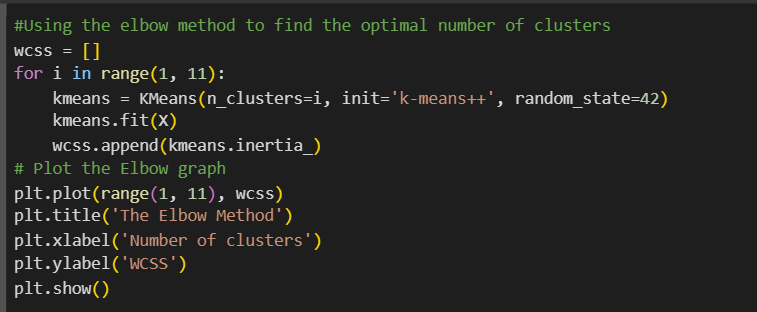
### K-MEANS

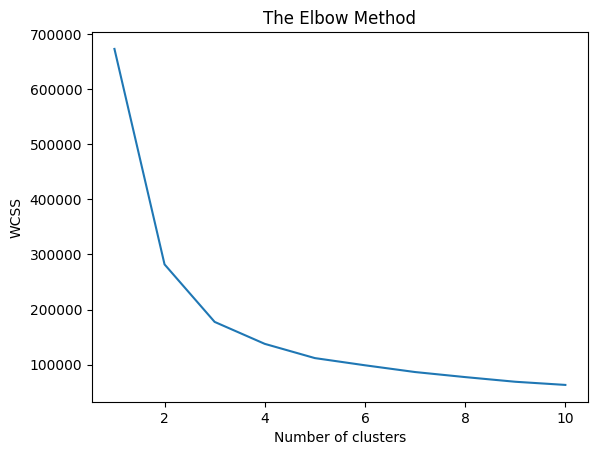


This code snippet imports features from a dataset and displays the first five rows of the selected features.

1. **Feature Selection**: The features math score, reading score, and writing score are selected from the dataset DS.
2. **Conversion to NumPy Array**: The selected features are converted to a NumPy array for further processing.
3. **Display First 5 Rows**: The first five rows of the feature array are printed to provide a quick preview of the data.

The first five students' scores in each of all three subjects are displayed in this output, which will be used for further analysis or modelling.



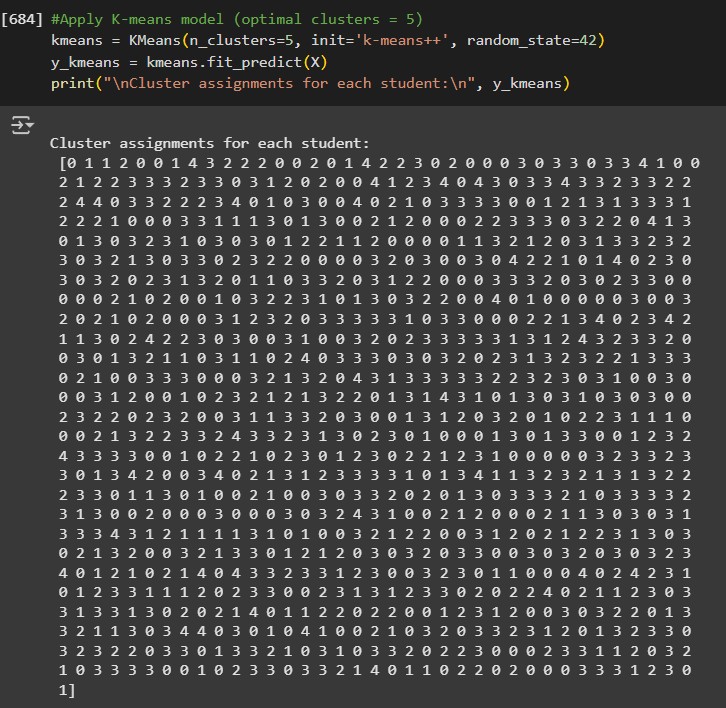


The right amount of clusters for K-Means clustering can be determined using the Elbow Method in this code.

1. **Within-Cluster Sum of Squares (WCSS)**: The code initializes an empty list to store the WCSS values for different numbers of clusters (from 1 to 10).
2. **K-Means Clustering**: A loop fits the K-Means model to dataset X, iterating through potential cluster counts and appending the inertia (WCSS) to the list.
3. **Plotting the Elbow Graph**: To see the "elbow," which represents the ideal number of clusters, the WCSS scores are plotted compared to the number of clusters.

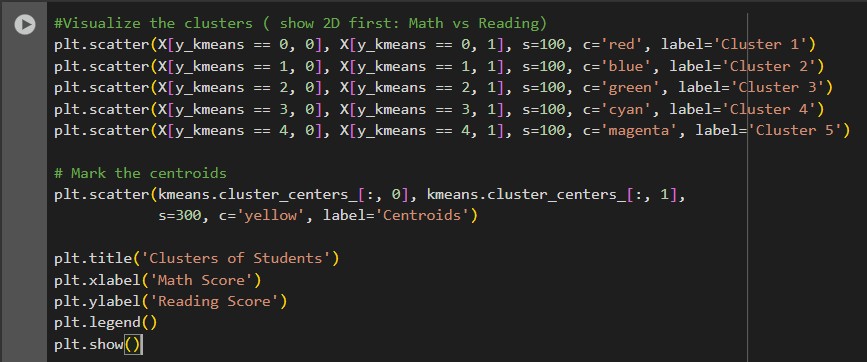
**Output:**  
The resulting graph shows an obvious decrease in WCSS that starts levelling out after a certain point as the number of clusters increases. Usually, this point indicates an ideal number of clusters.

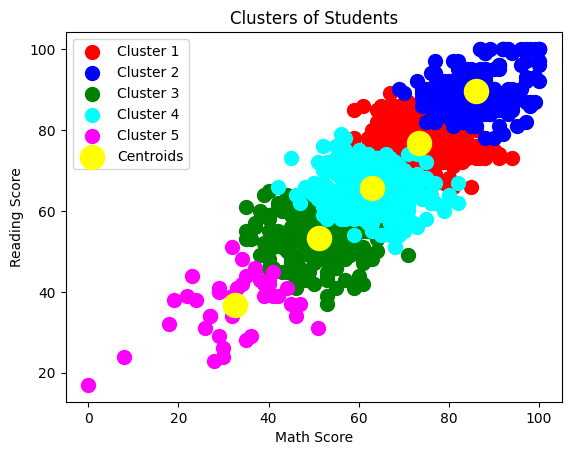
The graph's elbow point helps in selecting the ideal number of clusters for your dataset's clustering.



This code organizes students according to their reading, writing, and math results using the K-Means clustering algorithm. 5 clusters are the most suitable number to select.

1. **K-Means Model Initialization**: A K-Means model is created with 5 clusters using the k-means++ initialization method to improve convergence speed.
2. **Fitting the Model**: The model is then fitted to the dataset X, and the cluster assignments for each student are predicted.
3. **Cluster Assignments**: The resulting cluster assignments are printed, indicating which cluster each student belongs to.





This code displays math scores against reading scores to show the clusters created by the K-Means algorithm.

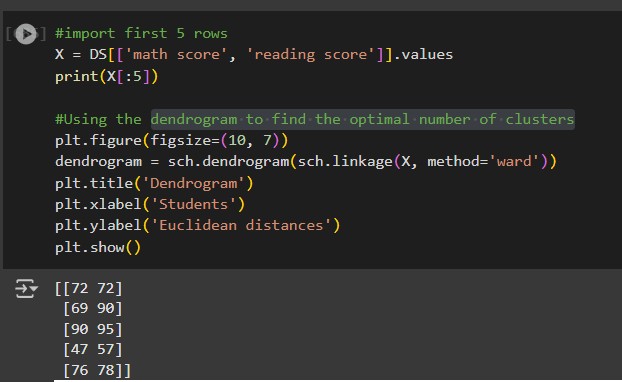
1. **Scatter Plot Creation**: A different color is used for each cluster. Based on their reading and math scores, students are plotted, and their cluster assignments are shown by different colors.
2. **Centroid Marking**: The centroids of the clusters are marked in yellow, providing a reference point for the average position of each cluster.
3. **Plot Annotations**: The plot includes titles and labels for both axes, as well as a caption to identify each cluster.

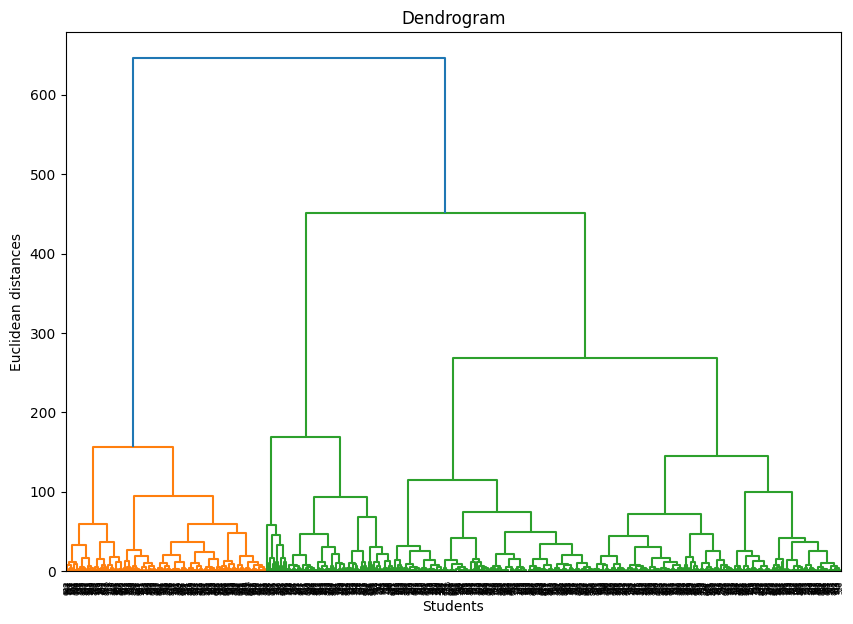
**Output:**

The distribution of students over five clusters is shown in the resulting scatter plot, which also shows the correlation between the students' reading and math’s results. Understanding the clustering results is made easier by the centroids, which visually represent each cluster's central tendency.

Analyzing trends in student performance based on both fields might be assisted by this visualization.

### HORIZONTAL





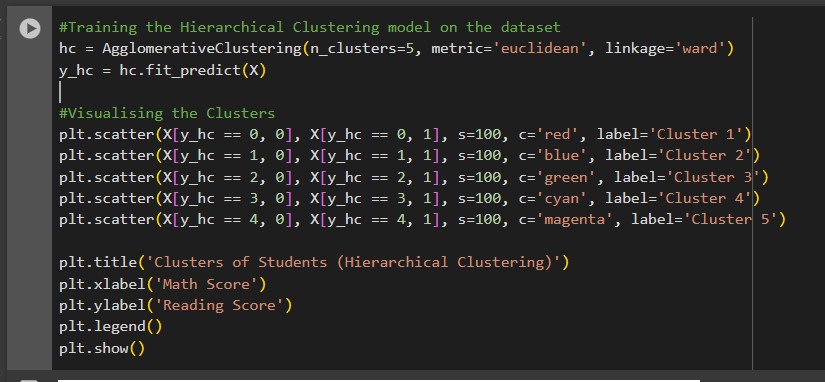
Based on reading and math results, this code creates a dendrogram to assist in finding out the perfect number of clusters for hierarchical clustering.

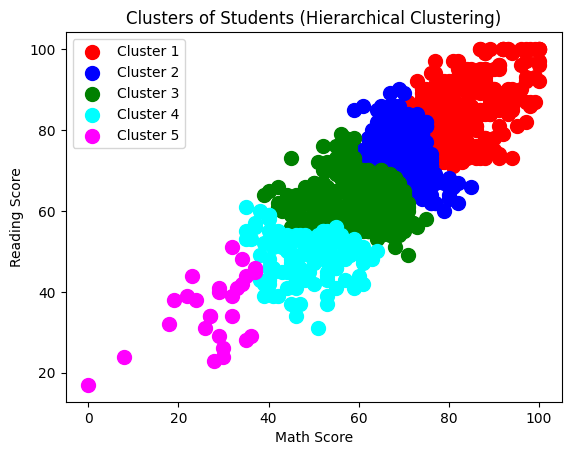
1. **Data Import**: The first five rows of the selected features (Math and Reading scores) are imported from the dataset.
2. **Dendrogram Creation**: A dendrogram is plotted using the Ward method, which minimizes the variance within clusters.
3. **Plot Annotations**: The plot includes titles and labels for clarity regarding the representation of students and the distances between clusters.

**Output:**

Based on the students results, the generated dendrogram shows the hierarchical link between them. The horizontal axis shows the students, and the vertical axis shows the Euclidean distances between groups.

This visualization can help identify the suitable number of clusters by observing where the branches of the dendrogram merge. A notable cut-off point can suggest the ideal number of clusters for further analysis.





In order to group students according to their reading and math scores, this code uses hierarchical clustering, visualizing the clusters that are created.

1. **Model Training**: The Agglomerative Clustering model is trained with 5 clusters using the Euclidean distance metric and Ward linkage.
2. **Cluster Assignments**: The model predicts cluster assignments for each student.
3. **Cluster Visualization**: A scatter plot displays each cluster with distinct colors, showing the distribution of students based on their Math and Reading scores.

**Output:**

Five clusters are shown in the resulting scatter plot, each of which is colored differently:

* **Cluster 1**: Red
* **Cluster 2**: Blue
* **Cluster 3**: Green
* **Cluster 4**: Cyan
* **Cluster 5**: Magenta

The visualization assists in understanding how students are grouped according to their reading and math skill, showing up patterns of the data that could guide future teaching methods or interventions.

The Ward linkage algorithm was also used for carrying out hierarchical clustering, and a dendrogram was produced to help determine the number of clusters. The students were divided into five colour-coded clusters after the dendrogram showed that five clusters were suitable. The clustering models aided in further understanding the information and segmentation by highlighting obvious trends between student performance levels.

# CLO3 /strategy TO improving the system after viewing the cluster diagram

Strategy for Improving the System after Viewing the Cluster Diagram

After analyzing the cluster diagrams created using K-Means and Hierarchical Clustering, the following strategies can be proposed to improve the system:

**1. TARGETED SUPPORT PROGRAMS:**

• Additional tutoring, mentoring, or academic support should be provided to students in lower-performing clusters (such as Cluster 5, which has low reading and math scores).

These students can benefit from personalised methods of instruction that help them develop certain reading and math skills.

**2. ENCOURAGING HIGH PERFORMERS:**

• To maintain interest and encourage excellence, high-performing clusters (such as Cluster 1, which has strong reading and math scores) need to be provided with challenging assignments, advanced courses, or extracurricular activities.

**3. FOCUSED RESOURCE ALLOCATION:**

•Resources like extra classes, workshops, or technology tools can be allocated differently based on cluster needs.

• More assistance for groups that are having trouble; motivating exercises for those that are performing in the middle.

**4. CUSTOMIZED TEACHING METHODS:**

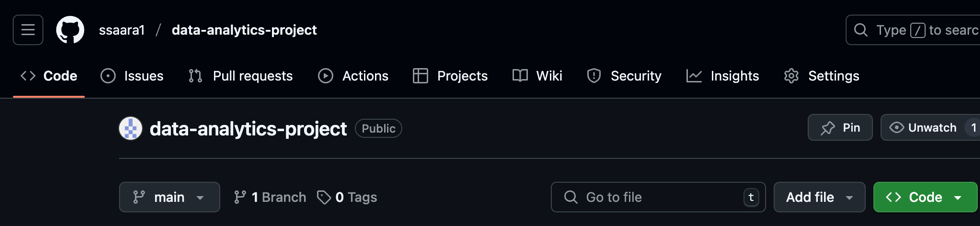
•Different clusters may have different learning styles or needs.

•Teachers can adapt teaching methods (visual, interactive, gamified learning) depending on the characteristics of each cluster.

**5. CONTINUOUS MONITORING AND RE-CLUSTERING:**

•Repeating the clustering on a regular basis (e.g., every semester) may help in improving the assistance techniques as students' performance changes over time.

# CLO4/Create a new repo for project in Git Hub

  
I created a new repository on GitHub named data-analytics-project under my account ssaara1 to host and manage my project files. I had the option to add a description and chose whether the repository would be public or private; I selected public to make it accessible.

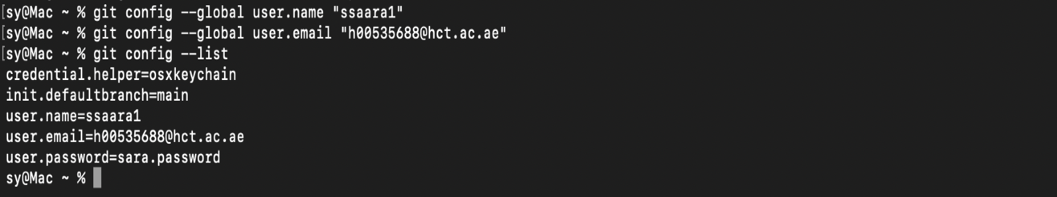
Here is the link to our repo: <https://github.com/ssaara1/data-analytics-project.git> or [git@github.com:ssaara1/data-analytics-project.git](https://encoded-592c9deb-987b-4562-aa3c-9fa3d37d83e9.uri/mailto%3agit%40github.com%3assaara1%2fdata-analytics-project.git) .

# CLO4/Upload all the project files created for CLO1,CLO2 and CLO3 to the Git Hub repo

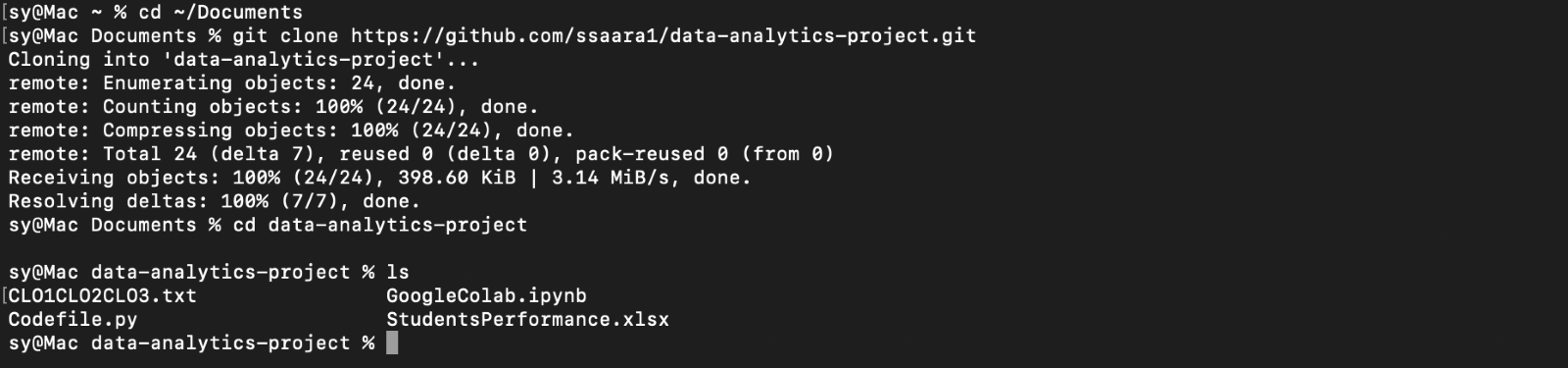
A screenshot of a computer

Description automatically generated, Picture  
After creating the GitHub repository data-analytics-project, I uploaded all the project files related to CLO1, CLO2, and CLO3. I uploaded the report as .txt files (named CLO1CLO2CLO3), the Python code files as .py files, the Google Colab notebook, and the Excel file we worked on during the project.

# CLO4/Configure Git with GITHUB

  
I configured Git with GitHub by setting my username and email globally using the terminal. I used the command git config --global user.name "ssaara1" and git config --global user.email "h00535688@hct.ac.ae". After configuration, I verified the setup by running git config --list, which displayed my Git username, email, default branch, and credential helper.

# CLO4/Clone Git hub repo to Git



To clone my GitHub repository to my local machine, I navigated to the **Documents** folder in the terminal and used the git clone command followed by the repository's URL:  
https://github.com/ssaara1/data-analytics-project.git.

After successfully cloning the repository, I entered the project folder and verified that all the project files were downloaded correctly.

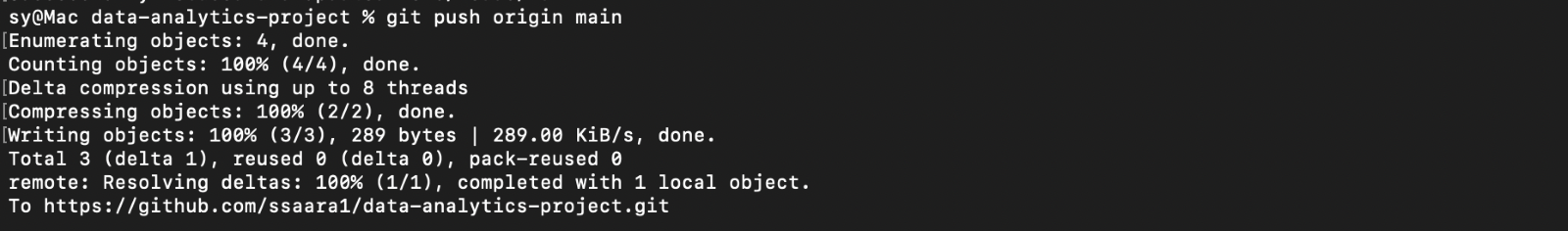
I checked that it exists by using this command (ls) and I found my repo

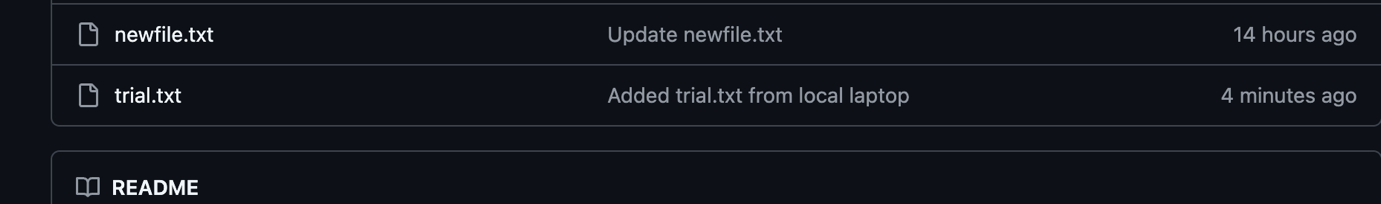
# clo4/pull any file from Git Hub repo to Git

A screen shot of a computer

Description automatically generated, Picture  
To pull a file from my GitHub repository to my local machine, I first created a new file named **newfile.txt** on GitHub. Then, I used the git pull origin main command in the terminal to pull the latest updates. The command successfully downloaded the new file to my local repository, which I verified using the terminal output.

# CLO4/Modify the pulled file and push the modified file to Git Hub

  
  
I pulled the latest changes from the remote GitHub repository using git pull origin main to ensure my local project was up to date. Then, I created a new file on my local laptop named trial.txt and added some content to it. I moved the file into my local project folder, added it using git add trial.txt, and committed the change with a message using git commit -m "Added trial.txt from local laptop". After that, I pushed the file to GitHub using git push origin main.



# CONCLUSION

The project was a holistic, practical journey throughout the key learning outcomes in data analytics. Throughout the stages of the project, we participated in various analytical approaches and provided development techniques that improved our expertise concerning theoretical concepts and real-world applications.

Under CLO1, we began first by discovering and preprocessing the dataset to have it in clean and structured format ready for analysis. This included handling missing data, statistical analysis, and visualizing relationships between significant academic performance measures such as math, reading, and writing scores. This set the foundation critical in laying the groundwork for proper modeling later on in the project.

In CLO2, we utilized regression and classification machine learning models. We developed and trained models like linear regression to predict scores and logistic regression to predict student results (pass/fail). These were tuned and tested using performance metrics like R² score, accuracy, precision, recall, and F1 score. The predictive modeling activity provided valuable information as to which features contributed the most to academic performance.

Shifting to CLO3, we carried out unsupervised learning with clustering techniques. K-Means and Hierarchical Clustering were used to group students based on their performance to create clusters. Visualization techniques such as scatter plots and dendrograms were used to determine the number of clusters that may yield the most insights and determine the patterns revealed. These enabled us to identify natural clusters among students, allowing for more insightful understanding of performance patterns.

For CLO4, we used version control and collaboration capabilities through the integration of GitHub and Git. We established a particular GitHub repository for the storage and management of our project files. We created the Git locally, cloned the repository, pulled and pushed files, and maintained a neat version history. This facilitated collaboration, traceability, and gave all the sides of the project to be securely backed up and recalled.

Overall, this project helped us become more skilled in data preprocessing, supervised and unsupervised learning, model evaluation, and collaborative development. Not only has it helped us become more technically competent, but it has also made us realize the importance of data-driven decision-making and effective team workflow management in real-world data analytics projects.

|  |  |
| --- | --- |
| Mona Asghar albloushi H00538165 | I worked with CLO1 and CLO2 in this project where I explored, analyzed, and modeled a student performance dataset. In CLO1, I conducted data preprocessing and data exploration like verification of the structure of the dataset and cleaning of the dataset for analysis. I used visualization tools to identify association among math, reading, and writing scores to help identify patterns and trends in student performance. In CLO2, I applied machine learning algorithms such as regression and classification to predict outcomes from student grades. The models were validated using performance metrics such as accuracy and score to deliver valid results. Python libraries such as pandas, seaborn, matplotlib, and scikit-learn supported the technical operations throughout the process. In addition to completing the practical work, we also collaborated as a team in preparing the final report and PowerPoint presentation to present our results and project findings efficiently. |
| Huda Mohammed H00535168 | In our project, I worked on different tasks. I was responsible for answering the last three questions in the CLO3 task, making sure my section matched the project goals. Through this work, I improved my Python coding skills using Google Colab. My section also helped support my team in completing the project. |
| Sara Yousif  H00535688 | I worked on CLO4 by handling the GitHub part of our project. I created a public repo on my GitHub account called data-analytics-project to store and organize all our project files. I set up Git on my laptop by configuring my username and email, then cloned the repository to my local machine to be able to work on it directly. I uploaded the Python code files, the Excel dataset, our Google Colab notebook, and the CLO reports in .txt format. I also practiced using important Git commands like git pull, git add, git commit, and git push. Besides GitHub, I also helped my team members in writing parts of the project report and preparing the PowerPoint presentation. |
| Amna Saif  h00495503 | I applied CLO3 by answering all four mandatory questions, including creating, training, and testing models through simple and multiple linear regression and classification through classification algorithms such as Logistic Regression, K-Nearest Neighbors (KNN), Naïve Bayes, and Decision Tree. I went through dataset selection and preprocessing, created scripts to forecast the dependent variable, and verified the performance of each model through accuracy scores and confusion matrices. In doing so, I studied how different algorithms perform on a specific dataset and learned the importance of choosing a suitable model in terms of data patterns, accuracy, and generalization. This activity strengthened my knowledge on predictive analytics, model evaluation, and practical application of machine learning techniques |