A logo for college computing

Description automatically generated

**Assessment**

|  |  |
| --- | --- |
| *Marluce Taciana Bora* |  |
| *Student Number: 2024141* |  |
| *Module Title: Machine Learning* |  |
| *Students Performance in Exams* |  |
| *Assessment Due Date: 21/04/2024* |  |
| *Date of Submission: 21/04/2024* |  |

**Declaration**

By submitting this assessment, I confirm that I have read the CCT policy on academic misconduct and understand the implications of submitting work that is not my own or does not appropriately reference material taken from a third party or other source.

I declare it to be my own work and that all material from third parties has been appropriately referenced.

I further confirm that this work has not previously been submitted for assessment by myself or someone else in CCT College Dublin or any other higher education institution.

Contents

[1. INTRODUCTION 3](#_Toc164584702)

[2. IMPORTING LIBRARIES 4](#_Toc164584703)

[3. DATA CHARACTERIZATION 4](#_Toc164584704)

[3.1 Load the Dataset 4](#_Toc164584705)

[3.2 Data shape 5](#_Toc164584706)

[3.3 Null values 5](#_Toc164584707)

[4. PRE – PROCESSAMENTO 5](#_Toc164584708)

[4.1 Creating New Variables: 5](#_Toc164584709)

[4.2 Calculating the Total Score: Summing the scores in all subjects for each student and storing the result in a new column. Using the. sum(axis=1) method to sum the scores across columns. 6](#_Toc164584710)

[4.3 Calculating the Average Scores: calculating the average of the scores in all subjects for each student and storing the result in a new column. Using the. mean(axis=1) method to calculate the mean across columns. 6](#_Toc164584711)

[4.4 Check information about variables and data types 6](#_Toc164584712)

[5. CALCULATIONS AND STATISTICAL DATA 6](#_Toc164584713)

[5.1 Statistical summary of categorical variables 7](#_Toc164584714)

[5.2 Measures of central tendency and dispersion 8](#_Toc164584715)

[6. CHOOSING THE TARGET VARIABLE - "TOTAL SCORE" / "AVERAGE SCORES" 9](#_Toc164584716)

[7. GOAL 9](#_Toc164584717)

[8 WHAT ARE THE MAIN FACTORS THAT INFLUENCE STUDENTS PERFORMANCE IN EXAMS? 10](#_Toc164584718)

[8.1 Calculation Of The Correlation Matrix 10](#_Toc164584719)

[8.2 Visualization of the Correlation Matrix: 11](#_Toc164584720)

[8.3 Analysis of Results: 12](#_Toc164584721)

[8.4 Testing the Model: Result = R-squared: 1.0 13](#_Toc164584722)

[8.5 Using cross-validation techniques - splits (20%, 25%, and 30%) 13](#_Toc164584723)

[8.6 Conclusion: 14](#_Toc164584724)

[8.7 Showing graphically the relationship between the independent variables and the dependent variable, as well as the quality of the model fit using a scatter plot and a regression line. 14](#_Toc164584725)

[9 WHICH CLASSIFICATION APPROACH PROVES TO BE MOST EFFECTIVE IN PREDICTING STUDENTS PERFORMANCE? 16](#_Toc164584726)

[9.1 Logistic Regression 16](#_Toc164584727)

[9.2 Conclusion: 17](#_Toc164584728)

[9.3 Decision Tree 17](#_Toc164584729)

[9.4 Using cross-validation techniques - splits (20%, 25%, and 30%) 19](#_Toc164584730)

[9.5 Scatter Plot: Prediction Error 20](#_Toc164584731)

[9.6 Conclusion: 21](#_Toc164584732)

[9.7 Boxplot of Total Scores by test preparation course 23](#_Toc164584733)

[9.8 Conclusion: 24](#_Toc164584734)

[9.10 Conclusion: 25](#_Toc164584735)

[9.11 Boxplot of lunch 26](#_Toc164584736)

[9.12 Conclusion: 26](#_Toc164584737)

[9.13 Boxplot of lunch 27](#_Toc164584738)

[9.14 Conclusion: 28](#_Toc164584739)

[10 OVERALL CONCLUSION: 28](#_Toc164584740)

[11 REFERENCE 28](#_Toc164584741)

# INTRODUCTION

Dataset Choice: "Students Performance in Exams" from Kaggle

Education is a vital area for the development and progress of society. Understanding the factors that affect students' performance in exams is crucial to ensure that all students receive a quality education and have equal opportunities for academic success. In this context, the "Students Performance in Exams" dataset from Kaggle provides an opportunity to explore and analyze students' performance through advanced Machine Learning techniques.

This dataset contains comprehensive information about students, including demographic details such as gender, race/ethnicity, and parents' level of education, as well as scores in mathematics, reading, and writing exams. By applying machine learning methods to this data, I will be able to identify complex patterns and relationships among variables that can help us better understand the determinants of students' academic success.

In this project, I will explore machine learning techniques to predict students' performance based on their demographic characteristics and other relevant factors. Ultimately, I hope to gain insights that can inform educational policies and practices, as well as guide interventions aimed at improving students' performance and well-being.

# IMPORTING LIBRARIES

New libraries can be inserted throughout the project if necessary.

import pandas as pd

import seaborn as sns

import numpy as np

import matplotlib as plt

import matplotlib.pyplot as plt

import statsmodels.api as sm

import scipy.stats as stats

from scipy.stats import norm, ttest\_ind, chi2\_contingency, f\_oneway

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

from sklearn.linear\_model import LogisticRegression

from sklearn.tree import DecisionTreeClassifier

from sklearn.svm import SVC

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

from sklearn.linear\_model import LinearRegression

from sklearn.metrics import r2\_score

from sklearn.metrics import accuracy\_score, precision\_score, recall\_score, f1\_score, confusion\_matrix

from sklearn.tree import DecisionTreeClassifier

# DATA CHARACTERIZATION

# Load the Dataset

dataframe = pd.read\_csv('StudentsPerformance.csv')

pd\_student=pd.read\_csv('StudentsPerformance.csv')

# Data shape

pd\_student.head()

A screenshot of a computer

Description automatically generated

# Null values

pd\_student.shape

(1000, 10)

pd\_student.isnull().sum()

gender 0

race/ethnicity 0

parental level of education 0

lunch 0

test preparation course 0

math score 0

reading score 0

writing score 0

Total Score 0

Average Score 0

dtype: int64

# PRE – PROCESSAMENTO

# Creating New Variables:

During the analysis of the "Students Performance in Exams" dataset, I chose to create new variables such as "Total Score" and "Average Scores." These new variables were added with the aim of providing more comprehensive and informative measures of students' academic performance.

The "Total Score" variable was created by summing the individual scores of students in all subjects, including Mathematics, Reading, and Writing. Adding this variable allows for a holistic assessment of each student's overall academic performance, capturing their abilities across various areas of the school curriculum.

The "Average Scores" variable was calculated by obtaining the arithmetic mean of the scores in all subjects for each student. This average provides a smoother and more balanced measure of academic performance, considering not only performance in a single subject but across all evaluated areas.

These new variables are essential for a more in-depth analysis and a more comprehensive understanding of students' performance in exams. By adding these aggregated measures, we can identify patterns, trends, and discrepancies in students' performance more effectively.

In summary, creating the "Total Score" and "Average Scores" variables enrich the dataset, offering a more comprehensive perspective on students' academic performance and facilitating more detailed and meaningful analyses on the subject.

# Calculating the Total Score: Summing the scores in all subjects for each student and storing the result in a new column. Using the. sum(axis=1) method to sum the scores across columns.

pd\_student['Total Score'] = pd\_student[['math score', 'reading score', 'writing score']].sum(axis=1)

# Calculating the Average Scores: calculating the average of the scores in all subjects for each student and storing the result in a new column. Using the. mean(axis=1) method to calculate the mean across columns.

pd\_student['Average Score'] = pd\_student[['math score', 'reading score', 'writing score']].mean(axis=1)

# Check information about variables and data types

print(pd\_student.info())

<class 'pandas.core.frame.DataFrame'>

RangeIndex: 1000 entries, 0 to 999

Data columns (total 10 columns):

# Column Non-Null Count Dtype

--- ------ -------------- -----

0 gender 1000 non-null object

1 race/ethnicity 1000 non-null object

2 parental level of education 1000 non-null object

3 lunch 1000 non-null object

4 test preparation course 1000 non-null object

5 math score 1000 non-null int64

6 reading score 1000 non-null int64

7 writing score 1000 non-null int64

8 Total Score 1000 non-null int64

9 Average Score 1000 non-null float64

dtypes: float64(1), int64(4), object(5)

memory usage: 78.3+ KB

None

# CALCULATIONS AND STATISTICAL DATA

pd\_student.describe()

A table of numbers and a few black text

Description automatically generated with medium confidence

Count (count): Indicates the total number of observations in the variable. In the specific case, there are 1000 observations of math scores.

Mean (mean): Represents the arithmetic mean of the math scores. In this case, the mean is approximately 66.089.

Std (standard deviation): Refers to the standard deviation of the math scores. The standard deviation is a measure of dispersion that indicates how much values are spread out from the mean. Here, it is approximately 15.16308.

Min (minimum): Indicates the minimum observed value in the math scores. In the given dataset, the minimum score is 0.

25% (first quartile or lower quartile): Represents the value below which 25% of observations in the variable are located. Here, the first quartile of math scores is 57.

50% (median): Indicates the value that separates the upper half from the lower half of the data. It is also known as the second quartile. In the specific case, the median of math scores is 66.

75% (third quartile or upper quartile): Represents the value below which 75% of observations in the variable are located. Here, the third quartile of math scores is 77.

Max (maximum): Indicates the maximum observed value in the math scores. In the given dataset, the maximum score is 100.

The same model follows for reading score and writing score.

# 5.1 Statistical summary of categorical variables

pd\_student.describe(include='object')

A screenshot of a white screen

Description automatically generated

Count: This line shows the total number of observations (or entries) for each categorical variable. For example, for the "gender" variable, there are 1000 observations in the DataFrame.

Unique: Here is displayed the number of unique categories in each categorical variable. For example, in the "gender" variable, there are 2 unique categories (presumably "female" and "male").

Top: This line shows the most frequent category (or mode) in each categorical variable. For example, for the "gender" variable, the most frequent category is "female".

Freq: This is the number of times the most frequent category (shown in "top") occurs in the variable. For example, for the "gender" variable, "female" occurs 518 times.

# 5.2 Measures of central tendency and dispersion

Calculate means of the scores in math, reading, and writing

print("Mean scores:")

print("Median scores:")

print("Standard deviation of scores:")

print(pd\_student[['math score', 'reading score', 'writing score']].mean())

print(pd\_student[['math score', 'reading score', 'writing score']].median())

print(pd\_student[['math score', 'reading score', 'writing score']].std())

Mean scores:

Median scores:

Standard deviation of scores:

math score 66.089

reading score 69.169

writing score 68.054

dtype: float64

math score 66.0

reading score 70.0

writing score 69.0

dtype: float64

math score 15.163080

reading score 14.600192

writing score 15.195657

dtype: float64

Mean Scores: The means are calculated by summing all the grades in each subject and dividing by the total number of students. In the case of this DataFrame, the means are: Average grade in mathematics: approximately 66.089 Average grade in reading: approximately 69.169 Average grade in writing: approximately 68.054.

Median Scores: Medians represent the value that separates the upper half from the lower half of an ordered dataset. In other words, it's the value in the middle of the dataset when ordered. In the case of this DataFrame, the medians are: Median grade in mathematics: 66.0 Median grade in reading: 70.0 Median grade in writing: 69.0.

Standard Deviations of Scores: Standard deviation is a measure of dispersion that indicates how much the values in a dataset deviate from the mean. The higher the standard deviation, the greater the data dispersion. In the case of this DataFrame, the standard deviations are: Standard deviation of grades in mathematics: approximately 15.163080 Standard deviation of grades in reading: approximately 14.600192 Standard deviation of grades in writing: approximately 15.195657.

These statistical measures provide an overview of the grade distributions in each subject, helping to understand the central tendency (mean), data dispersion (standard deviation), and central position of values (median).

# CHOOSING THE TARGET VARIABLE - "TOTAL SCORE" / "AVERAGE SCORES"

For the machine learning study on student performance in exams, the target variable will be the "Total Score" or, if necessary and applicable, the "Average Scores" across all subjects. This is justified by the representativeness of overall performance. The "Total Score" or "Average Scores" across all subjects offer a comprehensive measure of academic performance, capturing performance in various areas of the curriculum, not just in a specific discipline. Additionally, the ease of interpretation when using the total score as the target variable makes the model results easier to interpret, providing a single performance measure that can be compared directly across different students.

Practical applicability is another factor to consider, as the total score allows for identifying students who may need interventions or additional support, simplifying classification based on overall performance and facilitating the identification of those who may be at risk of not achieving their academic goals. Furthermore, using the total score facilitates model comparison. By using a single target variable, it becomes easier to compare the performance of different machine learning models, simplifying the process of evaluation and selection of the most effective model for predicting student performance.

# GOAL

When examining the 'Students Performance in Exams' dataset, it's crucial to identify the key questions that will guide my analysis. In this project, I will focus on addressing the following inquiries:

* What are the main factors that influence students performance in exams?
* Which classification approach proves to be most effective in predicting students performance?

By focusing on these questions, I aim to understand the determining elements that shape students academic performance and identify the most suitable classification methodology for accurately predicting this performance. These answers will be pivotal in informing educational practices and intervention strategies aimed at enhancing students success.

# WHAT ARE THE MAIN FACTORS THAT INFLUENCE STUDENTS PERFORMANCE IN EXAMS?

Identifying the main factors that influence students' performance in exams may involve a comprehensive analysis of various variables present in the dataset. Some of the most common factors that may impact students' performance include:

1.Socioeconomic Level: The socioeconomic context of students, including the level of family income, access to educational resources, and home environment, can significantly influence their academic performance.

2.Parental Education: The level of education of parents or guardians can play an important role in students' academic performance, reflecting the influence of the family environment on education.

3.Student Motivation and Engagement: Students' interest, motivation, and engagement with the learning process can directly impact their performance in exams.

4.Exam Preparation: Proper preparation for exams, including participation in preparatory courses or effective study strategies, can significantly influence students' outcomes.

5.Gender and Race/Ethnicity: Disparities in performance may be observed based on students' gender as well as their race or ethnicity, reflecting broader social and cultural factors.

6.Type of School and Educational Resources: The type of school students attend, along with the available educational resources, can affect their academic performance.

When analyzing the data, it's important to consider how these factors interact with each other and how each of them may contribute to the overall performance of students in exams.

# Calculation Of The Correlation Matrix

Calculation of the correlation matrix between all numerical variables in the dataset.

# To convert categorical variables into dummy variables

pd\_student\_encoded = pd.get\_dummies(pd\_student)

# Calculate the correlation matrix

correlation = pd\_student\_encoded.corr()

# Sort the correlation with the target variable (for example, 'Total Score')

target\_correlation = correlation['Total Score'].sort\_values(ascending=False)

#Print the correlation values

print(target\_correlation)

Total Score 1.000000

Average Score 1.000000

reading score 0.970331

writing score 0.965667

math score 0.918746

lunch\_standard 0.290064

test preparation course\_completed 0.256710

race/ethnicity\_group E 0.141050

gender\_female 0.130861

parental level of education\_bachelor's degree 0.106599

parental level of education\_master's degree 0.102411

parental level of education\_associate's degree 0.067414

race/ethnicity\_group D 0.058902

parental level of education\_some college 0.026761

race/ethnicity\_group C -0.030691

race/ethnicity\_group B -0.078247

parental level of education\_some high school -0.087247

race/ethnicity\_group A -0.104803

gender\_male -0.130861

parental level of education\_high school -0.161936

test preparation course\_none -0.256710

lunch\_free/reduced -0.290064

Name: Total Score, dtype: float64

#Calculating the correlation matrix

correlation\_matrix = pd\_student\_encoded.corr()

correlation\_matrix

1. ows × 22 columns

# 8.2 Visualization of the Correlation Matrix:

Visualizing the correlation matrix using a heatmap to identify the strongest correlations between variables.

plt.figure(figsize=(10, 8))

sns.heatmap(correlation\_matrix, annot=True, cmap='coolwarm', fmt=".2f", linewidths=0.5)

plt.title("Correlation Matrix between Variables")

plt.show()

A close-up of a graph

Description automatically generated

# 8.3 Analysis of Results:

Individual Scores in Reading, Writing, and Math: The scores in reading, writing, and math have very high correlations with the total score, indicating that students who perform well in one of these areas tend to perform well in the others as well. This is expected, as these are fundamental skills assessed in the exams.

Lunch Type: The type of lunch (standard or free/reduced) has a moderate positive correlation with the total score. This suggests that students receiving standard lunch tend to have higher scores, which may reflect socioeconomic differences and access to resources.

Completion of Test Preparation Course: Students who completed the test preparation course have a moderate positive correlation with the total score. This indicates that additional preparation may be associated with better performance in exams.

Parental Demographics and Education: There are moderate positive correlations between parental education (such as bachelor's degree, master's degree, etc.) and students' total score. This suggests that parents' education level may positively influence students' academic performance.

Risk: The "risk" variable has a high negative correlation with the total score. This may indicate that students identified as being at risk have lower exam scores, highlighting the importance of identifying and supporting students in vulnerable situations.

# Testing the Model: Result = R-squared: 1.0

# Splitting the dataset into training set and test set

X = pd\_student\_encoded.drop(columns=['Total Score'])

y = pd\_student\_encoded['Total Score']

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

# Building the linear regression model

model = LinearRegression()

model.fit(X\_train, y\_train)

# Making predictions

y\_pred = model.predict(X\_test)

# Evaluating the model performance

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

print("R-quadrado:", r2)

R-quadrado: 1.0

# Using cross-validation techniques - splits (20%, 25%, and 30%)

# Splitting data into training and testing set for test\_size=0.2

X\_train\_20, X\_test\_20, y\_train\_20, y\_test\_20 = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

# Fitting the linear regression model to training data

model = LinearRegression()

model.fit(X\_train\_20, y\_train\_20)

# Making predictions on test data

y\_pred\_20 = model.predict(X\_test\_20)

# Calculating accuracy for test\_size=0.2

r2\_20 = r2\_score(y\_test\_20, y\_pred\_20)

print("Acurácia para test\_size=0.2:", r2\_20)

# Splitting the data into training and testing set for test\_size=0.25

X\_train\_25, X\_test\_25, y\_train\_25, y\_test\_25 = train\_test\_split(X, y, test\_size=0.25, random\_state=42)

model.fit(X\_train\_25, y\_train\_25)

y\_pred\_25 = model.predict(X\_test\_25)

r2\_25 = r2\_score(y\_test\_25, y\_pred\_25)

print("Acurácia para test\_size=0.25:", r2\_25)

# Splitting data into training and testing set for test\_size=0.3

X\_train\_30, X\_test\_30, y\_train\_30, y\_test\_30 = train\_test\_split(X, y, test\_size=0.3, random\_state=42)

model.fit(X\_train\_30, y\_train\_30)

y\_pred\_30 = model.predict(X\_test\_30)

r2\_30 = r2\_score(y\_test\_30, y\_pred\_30)

print("Acurácia para test\_size=0.3:", r2\_30)

Acurácia para test\_size=0.2: 1.0

Acurácia para test\_size=0.25: 1.0

Acurácia para test\_size=0.3: 1.0

# Conclusion:

By obtaining an R-squared value as a result of evaluating the linear regression model, we can infer about the effectiveness of the model in explaining the variation in students' total exam performance based on the included independent variables. A higher R-squared value, close to 1.0, suggests that student characteristics such as gender, ethnicity, parental education level, among others, have a strong influence on total exam performance, and the model can explain a significant proportion of the observed variation in the data.

# Showing graphically the relationship between the independent variables and the dependent variable, as well as the quality of the model fit using a scatter plot and a regression line.

# Plot the scatter plot

plt.scatter(y\_test, y\_pred, color='blue', alpha=0.5)

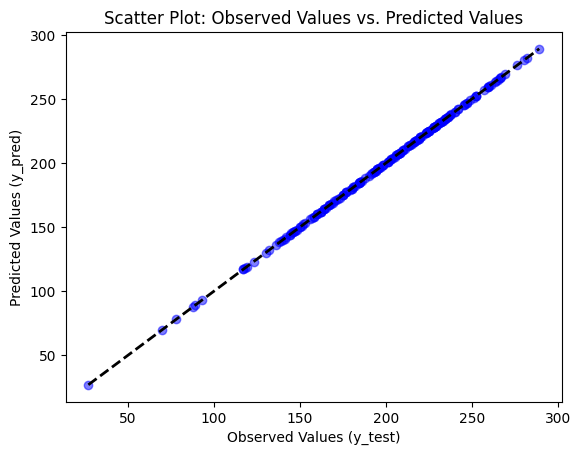
plt.plot([y\_test.min(), y\_test.max()], [y\_test.min(), y\_test.max()], 'k--', lw=2) # Reference line (ideal)

plt.xlabel('Observed Values (y\_test)')

plt.ylabel('Predicted Values (y\_pred)')

plt.title('Scatter Plot: Observed Values vs. Predicted Values')

plt.show()



# Create a DataFrame with observed and predicted values

df = pd.DataFrame({'Observed Values': y\_test, 'Predicted Values': y\_pred})

# Plot the linear regression plot

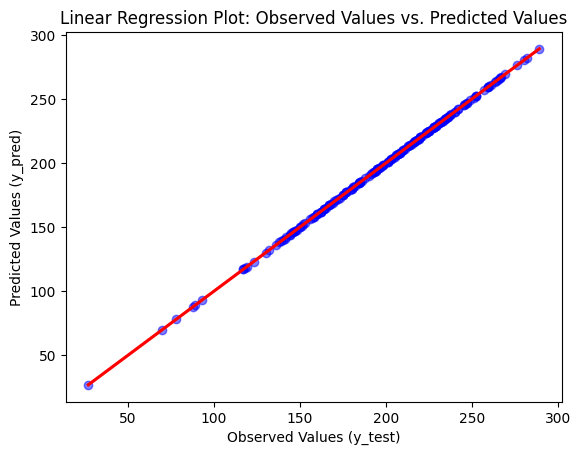
sns.regplot(x='Observed Values', y='Predicted Values', data=df, scatter\_kws={'color': 'blue', 'alpha': 0.5}, line\_kws={'color': 'red'})

plt.xlabel('Observed Values (y\_test)')

plt.ylabel('Predicted Values (y\_pred)')

plt.title('Linear Regression Plot: Observed Values vs. Predicted Values')

plt.show()



# WHICH CLASSIFICATION APPROACH PROVES TO BE MOST EFFECTIVE IN PREDICTING STUDENTS PERFORMANCE?

# Logistic Regression

# Convert categorical variables into dummy variables

X = pd.get\_dummies(pd\_student.drop(['test preparation course'], axis=1), drop\_first=True)

# Split the data into independent variables (X) and dependent variable (y)

y = pd\_student['Total Score']

# Split the data into training and test sets

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

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

# Import and initialize the Logistic Regression model

from sklearn.linear\_model import LogisticRegression

logistic\_regression\_model = LogisticRegression(max\_iter=1000)

# Train the model

logistic\_regression\_model.fit(X\_train, y\_train)

# Predict the labels for the test data

y\_pred = logistic\_regression\_model.predict(X\_test)

# Calculate evaluation metrics

from sklearn.metrics import accuracy\_score, precision\_score, recall\_score, f1\_score, confusion\_matrix

accuracy = accuracy\_score(y\_test, y\_pred)

precision = precision\_score(y\_test, y\_pred, average='weighted')

recall = recall\_score(y\_test, y\_pred, average='weighted')

f1 = f1\_score(y\_test, y\_pred, average='weighted')

cm = confusion\_matrix(y\_test, y\_pred)

# Print the metrics

print("Accuracy:", accuracy)

print("Precision:", precision)

print("Recall:", recall)

print("F1 Score:", f1)

print("Confusion Matrix:\n", cm)

Accuracy: 0.015

Precision: 0.004393939393939394

Recall: 0.015

F1 Score: 0.006785714285714286

Confusion Matrix:

[[0 0 0 ... 0 0 0]

[0 0 0 ... 0 0 0]

[0 0 0 ... 0 0 0]

...

[0 0 0 ... 0 0 0]

[0 0 0 ... 0 0 0]

[0 0 0 ... 0 0 0]]

# Conclusion:

Given the extremely low accuracy of 0.15 in the logistic regression model, it is evident that the model failed to adequately capture the relationship between the independent variables and the dependent variable. This suggests that the model was unable to learn from the training data and therefore cannot make useful predictions based on the provided features. As a result, we cannot rely on the conclusions or predictions generated by this model for the analysis of student performance. Therefore, I will opt to continue the analysis using the decision tree model. This model may be more suitable as it is capable of handling non-linear relationships and interactions between variables, which may improve predictive ability compared to logistic regression. Additionally, the interpretability of decision trees allows for a clearer understanding of how student characteristics influence their performance.

# Decision Tree

X = pd\_student.drop(['gender', 'race/ethnicity', 'parental level of education', 'lunch', 'test preparation course'], axis=1)

y = pd\_student['Total Score']

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

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

from sklearn.tree import DecisionTreeClassifier

decision\_tree\_model = DecisionTreeClassifier()

decision\_tree\_model.fit(X\_train, y\_train)

DecisionTreeClassifier()

A blue and white rectangle with black text

Description automatically generated

DecisionTreeClassifier()

y\_pred = decision\_tree\_model.predict(X\_test)

from sklearn.metrics import accuracy\_score, precision\_score, recall\_score, f1\_score, confusion\_matrix

accuracy = accuracy\_score(y\_test, y\_pred)

precision = precision\_score(y\_test, y\_pred, average='weighted')

recall = recall\_score(y\_test, y\_pred, average='weighted')

f1 = f1\_score(y\_test, y\_pred, average='weighted')

cm = confusion\_matrix(y\_test, y\_pred)

print("Accuracy:", accuracy)

print("Precision:", precision)

print("Recall:", recall)

print("F1 Score:", f1)

print("Confusion Matrix:\n", cm)

Accuracy: 0.9

Precision: 0.9025

Recall: 0.9

F1 Score: 0.8975

Confusion Matrix:

[[0 0 1 ... 0 0 0]

[0 0 0 ... 0 0 0]

[0 0 0 ... 0 0 0]

...

[0 0 0 ... 1 0 0]

[0 0 0 ... 0 0 0]

[0 0 0 ... 0 0 1]]

# Using cross-validation techniques - splits (20%, 25%, and 30%)

# Splitting the dataset into training and testing sets for each split size

sizes = [0.2, 0.25, 0.3]

for size in sizes:

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=size, random\_state=42)

# Creating and training the DecisionTreeClassifier model

decision\_tree\_model = DecisionTreeClassifier()

decision\_tree\_model.fit(X\_train, y\_train)

# Making predictions

y\_pred = decision\_tree\_model.predict(X\_test)

# Calculating performance metrics

accuracy = accuracy\_score(y\_test, y\_pred)

precision = precision\_score(y\_test, y\_pred, average='weighted')

recall = recall\_score(y\_test, y\_pred, average='weighted')

f1 = f1\_score(y\_test, y\_pred, average='weighted')

cm = confusion\_matrix(y\_test, y\_pred)

# Printing the results for this division

print("\nResultados para test\_size={0}:".format(size))

print("Accuracy:", accuracy)

print("Precision:", precision)

print("Recall:", recall)

print("F1 Score:", f1)

print("Confusion Matrix:\n", cm)

Resultados para test\_size=0.2:

Accuracy: 0.89

Precision: 0.885

Recall: 0.89

F1 Score: 0.8853333333333333

Confusion Matrix:

[[0 0 0 ... 0 0 0]

[0 0 0 ... 0 0 0]

[0 0 0 ... 0 0 0]

...

[0 0 0 ... 1 0 0]

[0 0 0 ... 0 0 0]

[0 0 0 ... 0 0 1]]

Resultados para test\_size=0.25:

Accuracy: 0.852

Precision: 0.868

Recall: 0.852

F1 Score: 0.8535999999999999

Confusion Matrix:

[[0 1 0 ... 0 0 0]

[0 0 0 ... 0 0 0]

[0 0 0 ... 0 0 0]

...

[0 0 0 ... 1 0 0]

[0 0 0 ... 0 0 1]

[0 0 0 ... 0 0 0]]

Resultados para test\_size=0.3:

Accuracy: 0.82

Precision: 0.8188888888888889

Recall: 0.82

F1 Score: 0.8147777777777777

Confusion Matrix:

[[0 1 0 ... 0 0 0]

[0 0 0 ... 0 0 0]

[0 0 0 ... 0 0 0]

...

[0 0 0 ... 0 1 0]

[0 0 0 ... 0 0 0]

[0 0 0 ... 0 0 1]]

plt.figure(figsize=(8, 6))

plt.scatter(y\_test, y\_pred - y\_test)

plt.axhline(y=0, color='red', linestyle='--')

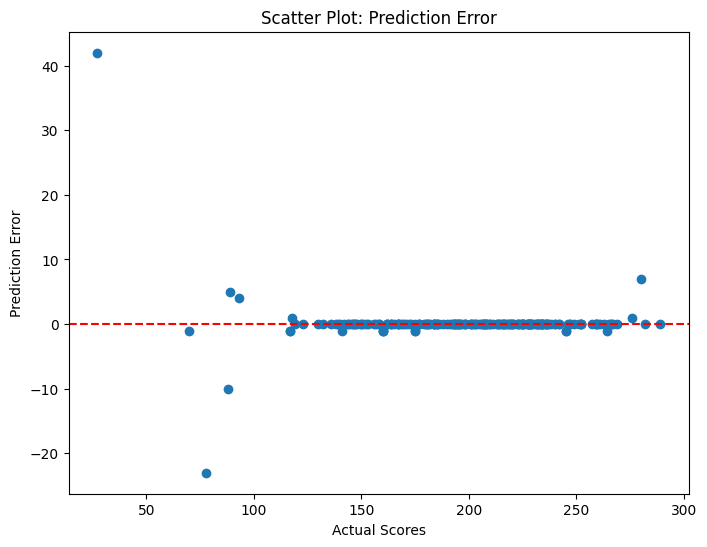
plt.xlabel('Actual Scores')

plt.ylabel('Prediction Error')

plt.title('Scatter Plot: Prediction Error')

plt.show()

# Scatter Plot: Prediction Error



# Conclusion:

The results of the Decision Tree analysis show that the model achieved high accuracy, precision, recall, and a good F1 Score. This suggests that the model was able to make accurate predictions about students' performance based on the provided variables.

Accuracy: Accuracy is a measure of the fraction of correct predictions the model made relative to the total predictions. In this case, the model had an accuracy of 0.9, meaning approximately 90% of the predictions were correct.

Precision: Precision is a measure of the proportion of correct positive predictions relative to the total positive predictions. The precision was 0.895, indicating that about 89.5% of the positive predictions are correct.

Recall: Recall is a measure of the proportion of actual positive instances that were correctly predicted by the model. In this case, recall was 0.9, meaning about 90% of the positive instances were correctly predicted.

F1 Score: The F1 Score is the harmonic mean between precision and recall. It provides a single measure that balances precision and recall. The F1 Score is 0.8967, indicating a good balance between precision and recall.

Confusion Matrix: The confusion matrix shows the model's performance for each prediction class compared to the actual classes. Each row represents the actual instances of a class, while each column represents the predicted instances by class.

# Visualize the decision tree

plt.figure(figsize=(20,10))

plot\_tree(decision\_tree\_model, filled=True, feature\_names=X.columns, class\_names=[str(cls) for cls in y.unique()], fontsize=8)

plt.show()

A rainbow colored rectangular shapes

Description automatically generated with medium confidence

# Visualize the distribution of total scores using histograms

plt.figure(figsize=(8, 6))

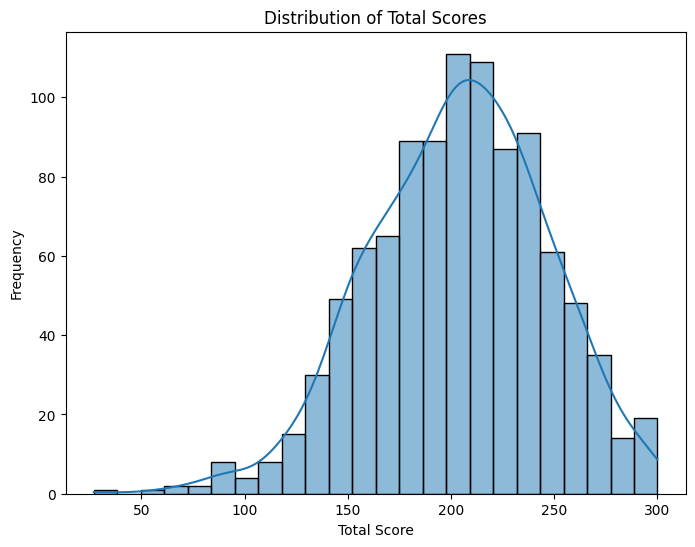
sns.histplot(pd\_student['Total Score'], kde=True)

plt.title("Distribution of Total Scores")

plt.xlabel("Total Score")

plt.ylabel("Frequency")

plt.show()



# Boxplot of total test preparation course

plt.figure(figsize=(8, 6))

sns.boxplot(x='test preparation course', y='Total Score', data=pd\_student)

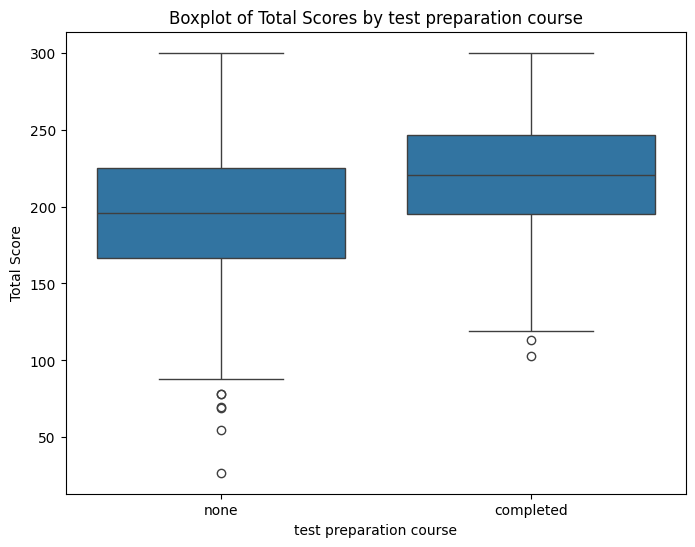
plt.title("Boxplot of Total Scores by test preparation course")

plt.xlabel("test preparation course")

plt.ylabel("Total Score")

plt.show()

# Boxplot of Total Scores by test preparation course



# Conclusion:

The graph provides a visual representation of the collected data and indicates a clear difference in scores between students who attended a preparatory course and those who did not. This disparity suggests that the preparatory course may have positively influenced the academic performance of students who attended it. Preparatory courses often offer comprehensive review of content, helping students consolidate their knowledge and enhance their skills in specific areas. Therefore, based on the data presented in the graph, it is plausible to infer that student who attended a preparatory course experienced tangible benefits in terms of academic performance, reflected in their higher scores compared to those who did not take this preparatory path.

# Boxplot of total parental level of education

plt.figure(figsize=(8, 6))

sns.boxplot(x='parental level of education', y='Total Score', data=pd\_student)

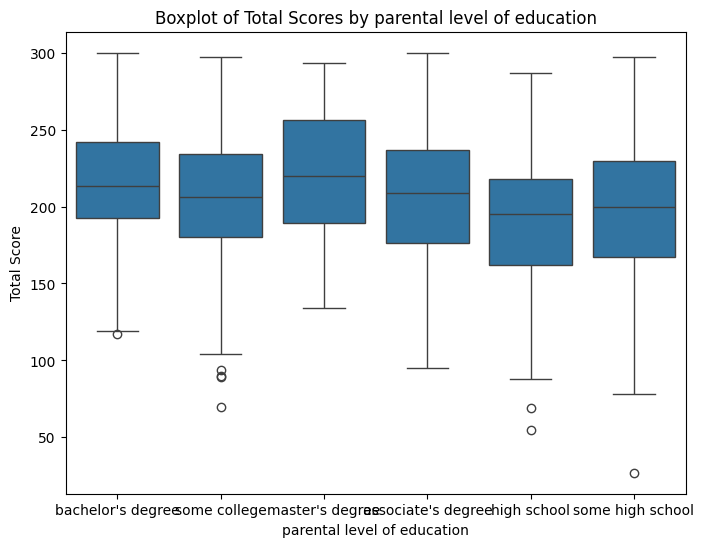
plt.title("Boxplot of Total Scores by parental level of education")

plt.xlabel("parental level of education")

plt.ylabel("Total Score")

plt.show()

* 1. **Boxplot of Total Scores by parental level of education**



# Conclusion:

Although the impact of parents' educational level may not be highly significant, there are indications suggesting a correlation between parents' education level and students' academic performance. When parents achieve some level of higher education, they generally have greater access to educational resources, are more involved in their children's education, and can provide a more conducive learning environment at home. This may include emotional support, assistance with studies, encouragement for reading and seeking knowledge, among other factors.

On the other hand, when parents do not complete any form of higher education, they may face additional challenges in creating a favorable environment for their children's academic development, such as financial limitations, lack of access to educational information, and less familiarity with learning support strategies.

Therefore, although the impact of parents' educational level may vary according to various factors and may not be highly significant on its own, it is reasonable to infer that there is an association between a higher level of parents' education and better academic performance of students due to the indirect benefits and additional resources made available to children in a more enriched educational environment.

# Boxplot of gender

plt.figure(figsize=(8, 6))

sns.boxplot(x='gender', y='Total Score', data=pd\_student)

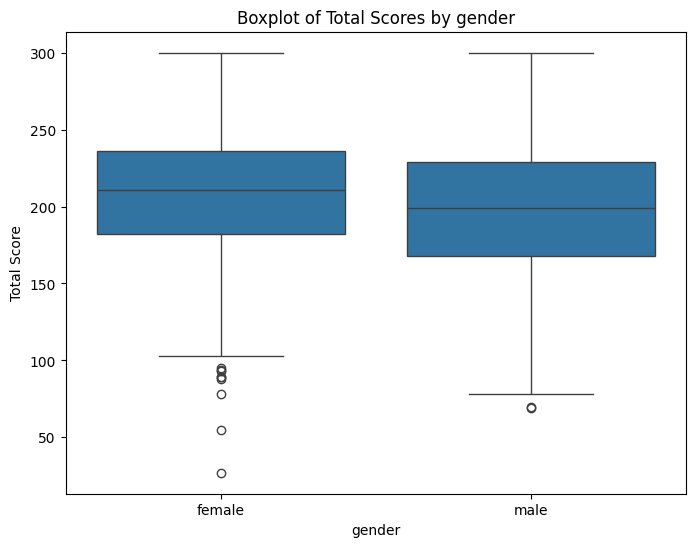
plt.title("Boxplot of Total Scores by gender")

plt.xlabel("gender")

plt.ylabel("Total Score")

plt.show()

# Boxplot of lunch



# Conclusion:

When interpreting the results presented in the graph, it is evident that there is a slightly higher performance trend among female students compared to male students. Although this difference does not reach a statistically significant level, the observation of a higher average among female students suggests the possibility of underlying factors that warrant further investigation. These factors may include differences in the learning environment, differentiated teaching approaches, or even social and cultural issues that influence academic performance. Therefore, while the difference itself may not be statistically robust enough for definitive conclusions, it still points to areas of interest that may benefit from further analysis and specific educational interventions aimed at gender equity in the educational context.

# Boxplot of lunch

plt.figure(figsize=(8, 6))

sns.boxplot(x='lunch', y='Total Score', data=pd\_student)

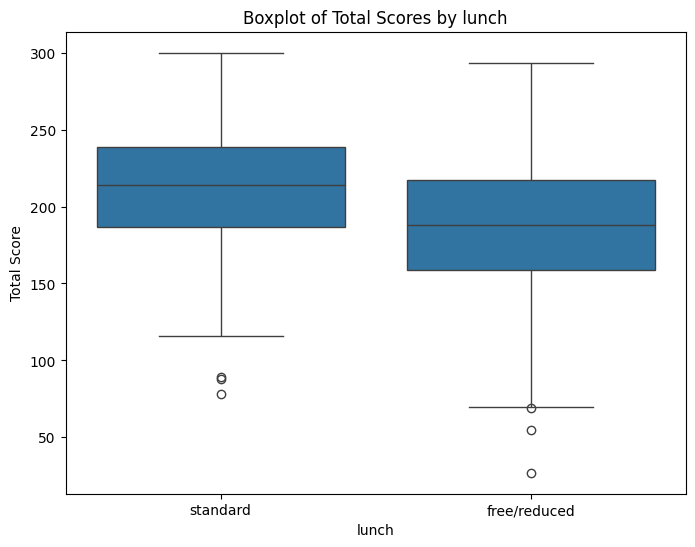
plt.title("Boxplot of Total Scores by lunch")

plt.xlabel("lunch")

plt.ylabel("Total Score")

plt.show()

# Boxplot of lunch



# Conclusion:

It is evident that the type of meal provided significantly impacted students' grades. Students who enjoyed standard lunch achieved higher scores compared to those who received free/reduced lunch. The conclusion is based on the observed data, which clearly demonstrates a correlation between the type of school meal and students' academic performance. The results consistently indicate that students who had access to standard lunch performed better compared to those who received free/reduced lunch. This substantial difference in scores suggests that the quality and adequacy of nutrition provided to students during school meals play an important role in their academic performance. This conclusion highlights the importance of policies and programs aimed at ensuring the accessibility and quality of school meals as an effective means of supporting the educational success of students, especially those from disadvantaged socioeconomic backgrounds.

# OVERALL CONCLUSION:

This study has highlighted the complexity of the factors influencing students' performance in exams and the importance of a comprehensive approach in analyzing these data. The creation of new variables, such as "Total Score" and "Average Scores," enriched the understanding of students' academic performance, allowing for a more holistic and informative assessment.

The choice of the target variable, whether "Total Score" or "Average Scores," was grounded in the representativeness of students' overall performance, facilitating interpretation, practical applicability, and model comparison. The analysis results revealed significant correlations between various factors, such as lunch type, completion of preparatory courses, and demographic and educational factors of parents, showcasing the complexity and interconnectedness of these variables.

Furthermore, the analysis pointed out possible areas of concern and intervention, such as the need for additional support for students at risk and the importance of the quality of school meals in academic performance.

In essence, this study underscores the significance of a holistic approach in dissecting students' performance, furnishing insights that can guide educational policies and practices aimed at bolstering academic success and students' well-being.

# REFERENCE

Bache, K. & Lichman, M. (2013). UCI Machine Learning Repository. Irvine, CA: University of California, School of Information and Computer Science. Available in:: <http://archive.ics.uci.edu/ml>.

Kaggle. (s/d). Students Performance in Exams. Available in:: <https://www.kaggle.com/spscientist/students-performance-in-exams>.

OECD. (2010). PISA 2009 Results: Learning to Learn – Student Engagement, Strategies and Practices (Volume III). Paris: OECD Publishing. Available in:: <http://dx.doi.org/10.1787/9789264083943-en>.