

A MAJOR PROJECT
On
FEATURE ENGINEERING DATA ANALYSIS

Dissertation submitted in the partial fulfillment of the requirements
for the award of the degree of

BACHELOR OF TECHNOLOGY

By

DEPARTMENT OF INTERNSHIPS

Ms. LATCHI. NAGA MYTHILI	CSINP413
Ms. PALLERLA. SAI NAVYA	CSINP411
Ms. MANAM. NANDINI	CSINP412
Ms. ALAVALAPATI. PUJITHA	CSINP414
Ms. ADDANKI. YASHASWINI	CSINP415

Under the esteemed Guidance of

Er. Y V D CHANDRA SEKHAR

Founder & Chief Executive Officer

CS CODENZ



LEARN HERE, LEAD ANYWHERE

CS CODENZ

GUDIVADA – 521 323, ANDHRA PRADESH, INDIA

2022-2023

A MAJOR PROJECT
On
FEATURE ENGINEERING DATA ANALYSIS

Dissertation submitted in the partial fulfillment of the requirements
for the award of the degree of

BACHELOR OF TECHNOLOGY

By

DEPARTMENT OF INTERNSHIPS

Ms. LATCHI. NAGA MYTHILI	CSINP413
Ms. PALLERLA. SAI NAVYA	CSINP411
Ms. MANAM. NANDINI	CSINP412
Ms. ALAVALAPATI. PUJITHA	CSINP414
Ms. ADDANKI. YASHASWINI	CSINP415

Under the esteemed Guidance of

Er. Y V D CHANDRA SEKHAR

Founder & Chief Executive Officer

CS CODENZ



LEARN HERE, LEAD ANYWHERE

CS CODENZ

GUDIVADA – 521 323, ANDHRA PRADESH, INDIA

2022-2023

CS CODENZ



CERTIFICATE

This is to certify that the dissertation entitled “PROJECT ON FEATURE ENGINEERING DATAANALYSIS “submitted by Ms. LATCHI. NAGA MYTHILI (CSINP413), Ms. PALLERLA. SAI NAVYA(CSINP411), Ms. MANAM.NANDINI(CSINP412), Ms. ALAVALAPATI.PUJITHA(CSINP414), and Ms. ADDANKI. YASHASWINI (CSINP415). In the partial fulfillment of the requirements for the award of the degree BACHELOR OF TECHNOLOGY from CS CODENZ is a record of bona fide work carried out by them under my guidance and supervision during the year 2022-2023. The results embodied in this dissertation have not been submitted by any other university or institution for the award of any degree.

Signature of the Supervisor

Er. Y V D CHANDRA SEKHAR

Founder & CEO, CS CODENZ

DECLARATION

I am Ms. LATCHI. NAGA MYTHILI (CSINP413) declared that the dissertation report entitled “FEATURE ENGINEERING DATA ANALYSIS” is no more than 1,00,000 words in length including quotes and exclusive of tables, figures, bibliography, and references. This dissertation contains no material that has been submitted previously, in whole or in part, for the award of any other academic degree or diploma. Except where otherwise indicated this dissertation is our own work.

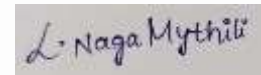
Roll No

Name

Signature

CSINP413

L.NAGA MYTHILI

A rectangular box containing a handwritten signature in dark ink. The signature appears to be 'L. Naga Mythili' written in a cursive style.

Date:

Place:

COs, POs and PSOs Mapping

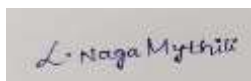
Subject Name : Major Project
 Subject Code : PY42223Academic
 Year : 2022 - 2023

Subject Code	Course Outcomes	
PR4204	CO1	Formulate solutions to computing problems using latest technologies and tools
	CO2	Work effectively in teams to design and implement solutions to computational problems and socially relevant issues
	CO3	Recognize the social and ethical responsibilities of a professional working in the discipline
	CO4	Apply advanced algorithmic and mathematical concepts to the design and analysis of software
	CO5	Devise a communication strategy (language, content and medium) to deliver messages according to the situation and need of the audience.
	CO6	Deliver effective presentations, extemporaneous or impromptu oral presentations. Setting up technical reports using technical tools.

CO-PO-PSOs Mapping

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
CO 1	3	2	-	2	2	-	-	-	-	-	-	-	3	-	-
CO 2	2	3	-	2	2	-	-	-	-	-	-	-	3	-	-
CO 3	3	3	-	2	2	-	-	-	-	-	-	-	3	-	-
CO 4	3	3	-	2	2	-	-	-	-	-	-	-	3	-	-
CO 5	2	3	-	2	2	-	-	-	-	-	-	1	3	-	-
CO 6	2	3	2	2	3	-	-	-	2	2	2	2	3	-	-
Avg	2.50	2.83	2.00	2.00	2.17	-	-	-	2.00	2.00	2.00	1.50	3.00	-	-

Note: 1 – Good, 2 – Average, 3 - Excellent



Signature of Student with Date

Signature of Guide with Date

ACKNOWLEDGEMENT

This dissertation could not have been written without the support of our guide, Er. Y V D Chandra Sekhar, Founder & CEO of CS CODENZ, who not only served as our superior but also encouraged and challenged us throughout our academic program. Our foremost thanks go to him. Without him, this dissertation would not have been possible. We appreciate his vast knowledge in many areas, and his insights, suggestions, and guidance that helped to shape our research skills. With a great sense of pleasure and immense gratitude, we acknowledge the help of these individuals. We owe many thanks to the people who helped and supported us during the writing of this report.

We are thankful to our project coordinator, Er. Y V D Chandra Sekhar, Founder & CEO of CS CODENZ, for his continuous support.

We express our sincere thanks to our respected for their valuable suggestions and constant motivation that greatly helped us successfully complete the project. We also take the privilege to express our heartfelt gratitude to Er. Y V D Chandra Sekhar, Founder & CEO of CS CODENZ. We are thankful to all faculty members for extending their kind cooperation and assistance. Finally, we are extremely thankful to our parents and friends for their constant help and moral support.

CSINP413

Ms. LATCHI. NAGA MYTHILI

CSINP411

Ms. PALLERLA. SAI NAVYA

CSINP412

Ms. MANAM. NANDINI

CSINP414

Ms. ALAVALAPATI. PUJITHA

CSINP415

Ms. ADDANKI. YASHASWINI

TABLE OF CONTENTS

Abstract	(i)
1. Introduction	1
1.1 Feasibility Study.....	1
1.2 Problem Statement.....	2
2. Motivation and Objective.....	3
2.1 Motivation.....	3
2.2 Objective.....	3
3. Software and Hardware Requirements.....	4
3.1 Software Requirements	4
3.2 Hardware Requirements.....	4
4. Literature Survey.....	5
4.1 Data Collection	5
4.2 Data Pre-Processing	5
4.3 Data Modeling	6
4.4 Result Interpretation.....	6
5. Keywords and Definitions.....	7
5.1 Classification.....	7
5.2 Feature Engineering.....	7
5.3 Regression.....	8
5.4 Clustering.....	8
5.5 Summarization	8
5.6 Data Visualization.....	9
5.7 Bias	9
5.8 Reproducibility.....	9
5.9 Interpretability.....	9
6. Design	10
6.1 ER model	10-12
7. Methodology	13
7.1 Data Set Generation	14
7.2 Covariance	15
7.3Correlation	15
8. Coding and Testing	16
8.1 Data Collection.....	16-59
8.2 Data Cleaning.....	60-67
8.3 Feature Engineering.....	68
8.4 Data Analysis	68-80
8.5 Statistical Analysis.....	80-89
8.6 Data Visualization.....	90-98
9. Result.	99
10. Conclusion	100
11. Future Scope.....	101
12. References.....	102

ABSTRACT

This project delves into feature engineering's pivotal role in data analysis and machine learning. Through diverse datasets, we demonstrate how meticulously shaping features can dramatically enhance model effectiveness. Techniques like dimensionality reduction, binning, and encoding are applied, informed by thorough exploratory data analysis. The connection between feature engineering and successful model training highlights how refined features significantly boost predictive accuracy and model resilience. This study offers invaluable insights for navigating data manipulation's intricate path, showcasing the finesse of feature crafting to unlock superior modeling outcomes.

CHAPTER-1

INTRODUCTION

1.1 FEASIBILITY STUDY:

In the realm of data analysis and machine learning, the paramount importance of feature engineering has emerged as a transformative catalyst in bolstering the efficacy and performance of models. This practice revolves around the deliberate and strategic transformation of raw data into discerning features, thereby unearthing latent patterns and intricate relationships that serve as the bedrock for robust analysis and predictive modeling. As datasets grow in complexity, the role of feature engineering becomes increasingly paramount in extracting valuable insights and ensuring the precision of predictive algorithms.

This research endeavor delves comprehensively into the domain of feature engineering, with the overarching goal of illuminating its profound influence within the context of data analysis. By traversing an array of diverse datasets and deploying a repertoire of techniques encompassing dimensionality reduction, binning, and encoding, we embark on an empirical exploration of how these meticulous manipulations reverberate through the efficacy of models. This study not only underscores the symbiotic relationship inherently shared by feature engineering and data analysis but also offers a comprehensive manual for practitioners eager to harness the potency of intricately crafted features in their pursuit of meaningful insights and heightened predictive accuracy.

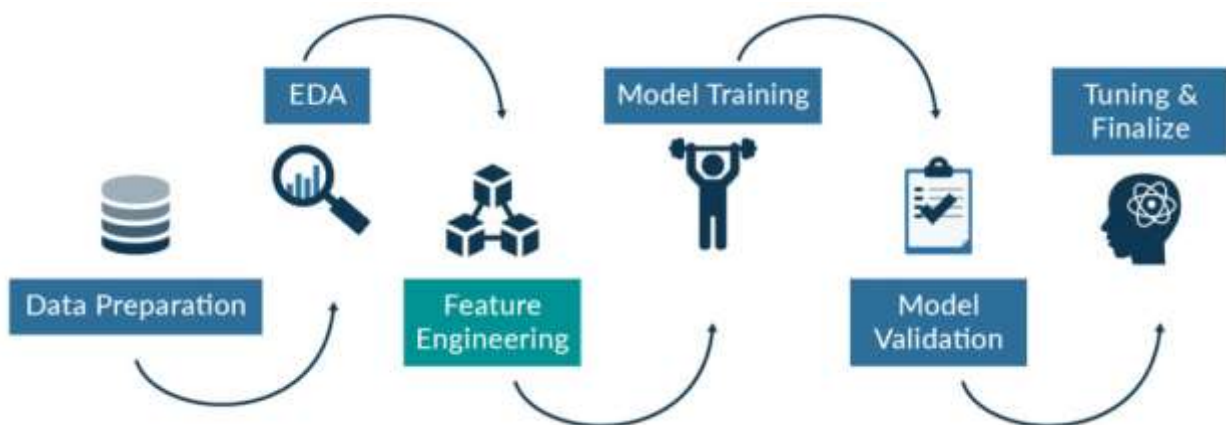


Figure 1

1.2 PROBLEM STATEMENT:

This project aims to enhance predictive modelling by developing a comprehensive strategy that combines advanced feature engineering and data analysis techniques. The primary goal is to boost model accuracy, reduce overfitting, and improve interpretability, while considering potential biases within the data. The project will systematically explore data pre-processing, feature extraction, and bias mitigation, all while optimizing resource utilization. By addressing these challenges, the project seeks to create predictive models that are accurate, transparent, and fair, contributing to informed decision-making across various domain.

CHAPTER-2

MOTIVATION AND OBJECTIVE

2.1 MOTIVATION:

The motivation behind undertaking this project on feature engineering in data analysis is rooted in the recognition of feature engineering's transformative impact. In the realm of machine learning and data analysis, the quality of features directly influences model performance and the ability to derive meaningful insights. By embarking on this project, we aim to unravel the intricate techniques that empower us to shape raw data into powerful attributes. This project's drive stems from the desire to equip individuals with the expertise to navigate real-world data challenges effectively. By showcasing hands-on applications and emphasizing the synergy between theory and practice, we aspire to empower participants to harness feature engineering as a powerful tool for refining models and uncovering hidden patterns in data.

2.2 OBJECTIVE:

The primary objective of this project is to comprehensively investigate and demonstrate the significance of feature engineering in data analysis and machine learning. The project aims to:

1. Provide an in-depth understanding of various feature engineering techniques, including data preprocessing, transformation, selection, and creation.
2. Illustrate the impact of well-crafted features on enhancing predictive model accuracy and uncovering meaningful insights.
3. Offer practical hands-on experience by working with real-world datasets to showcase the application of feature engineering methods.
4. Emphasize the role of domain knowledge in shaping relevant and effective features.
5. Equip participants with the skills and knowledge required to navigate complex data challenges and optimize model performance through feature engineering.

CHAPTER-3

SOFTWARE AND HARDWARE REQUIREMENTS

3.1 Software Requirements:

Operating System	: Windows
Programming Language	: Python
Modules required	: latest version of pip, NumPy, Pandas & all other analysis part modules.
Datasets	: Own dataset
IDE's	: PyCharm / Jupyter / Google Colab

3.2 Hardware Requirements:

Processor	: Corei3 or higher / Ryzen-3 or higher
RAM	: Minimum of 4GB
Hard disc	: Minimum of 500GB

CHAPTER-4

LITERATURE SURVEY

Literature Survey:

In the recent years there has been tremendous research done on the data analysis.

With the help of literature survey, we realized that the basic steps in data analysis are: -

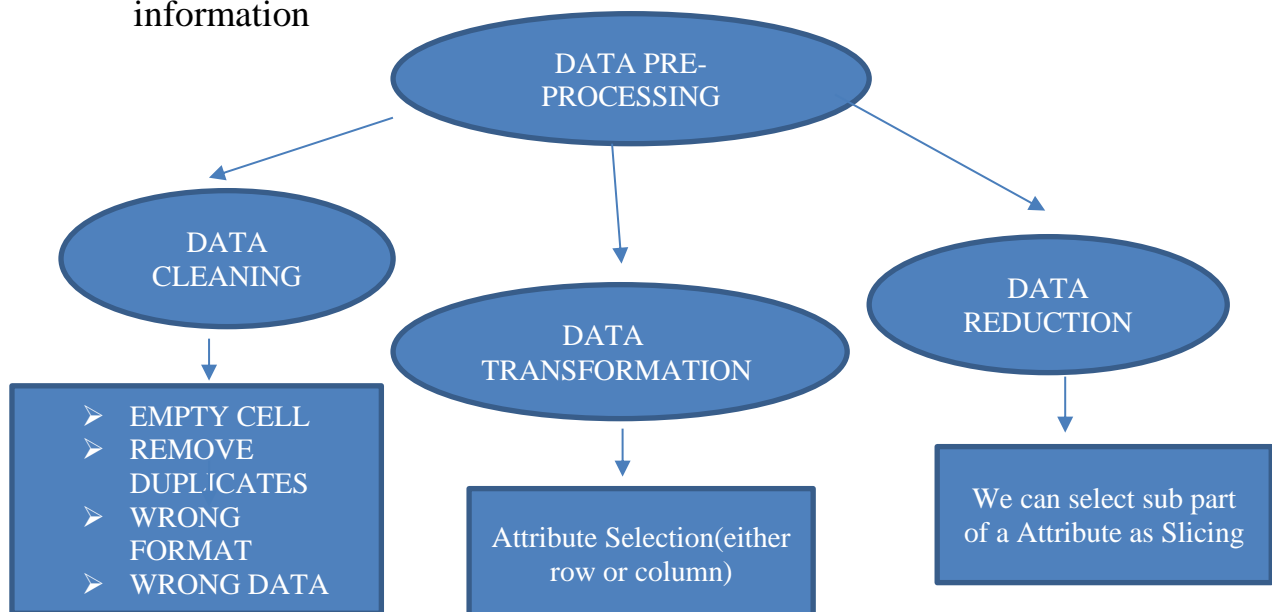
- Data collection
- Data pre-processing
- Data modeling
- Result interpretation

4.1 Data collection:

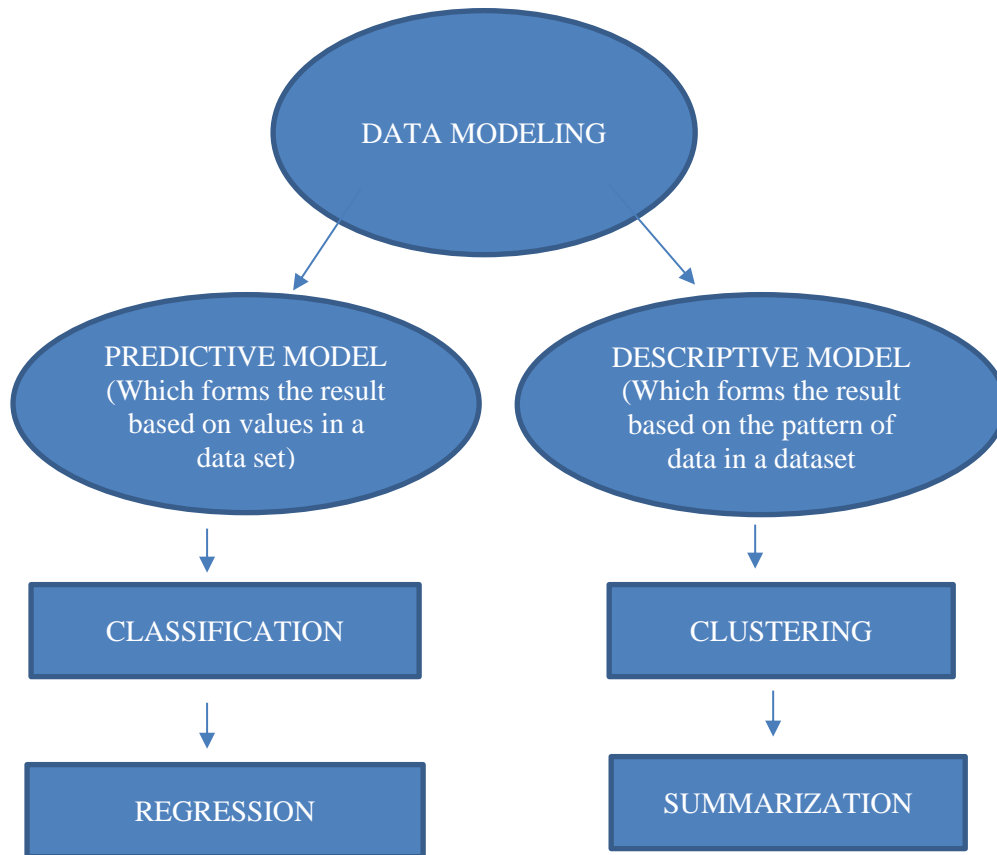
Data collection is the process of gathering, recording, and obtaining information or data from various sources or subjects for a specific purpose. It is a fundamental step in research, analysis, and decision-making processes across various fields and disciplines. To create the data set, data collecting is necessary.

4.2 Data Pre-Processing :

- It is a technique which is used to Transforming the raw data in to information



- A Data Modeling is a group of Models (nothing but Design) which are used to fit the Data in our Required Task or Situation.



4.3 RESULT INTERPRETATION:

The result interpretation section encapsulates the main insights drawn from the data, offering conclusions and explaining their significance in relation to the project's goals and next steps.

CHAPTER-5

KEYWORDS AND DEFINITIONS

5.1 Classification:

It is a predictive model and focus on values present in the Dataset. The significance of classification within a project is its ability to sort data into meaningful groups, aiding in understanding patterns and making informed decisions for improving outcomes, such as student performance or any projects dataset values or resource allocation, based on identified categories, it performs.

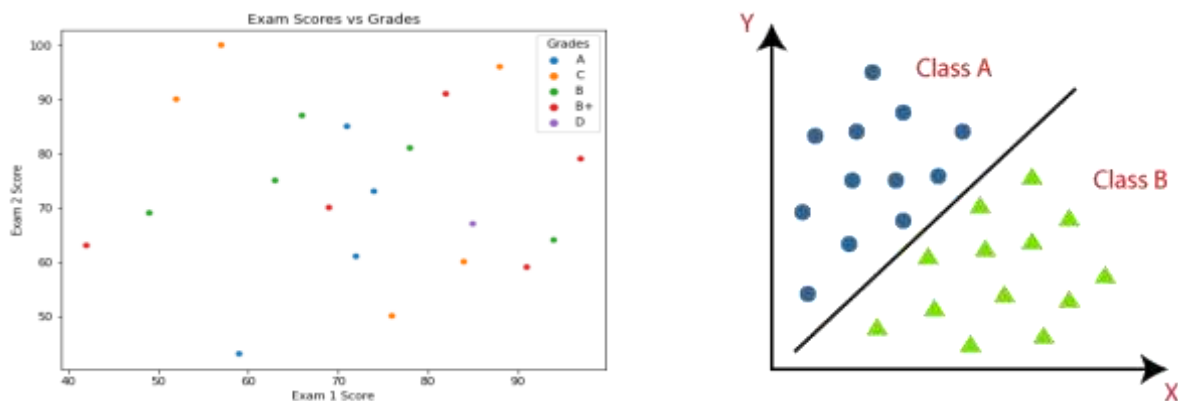


Figure-2

5.2 Feature Engineering :

Transforming raw data for improved machine learning model performance.

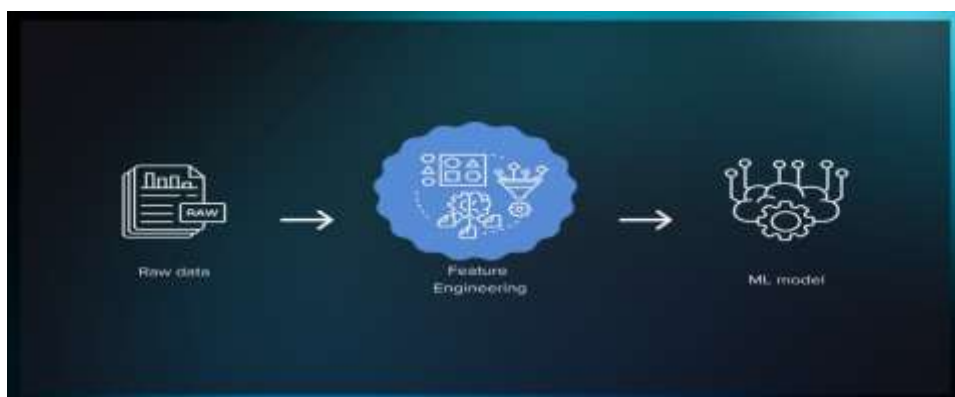


Figure-3

5.3 Regression:

Regression analysis outside education predicts economic trends, guides marketing strategies by analyzing consumer behavior, and uncovers

relationships in scientific experiments, aiding in predictions and decision-making across various fields.

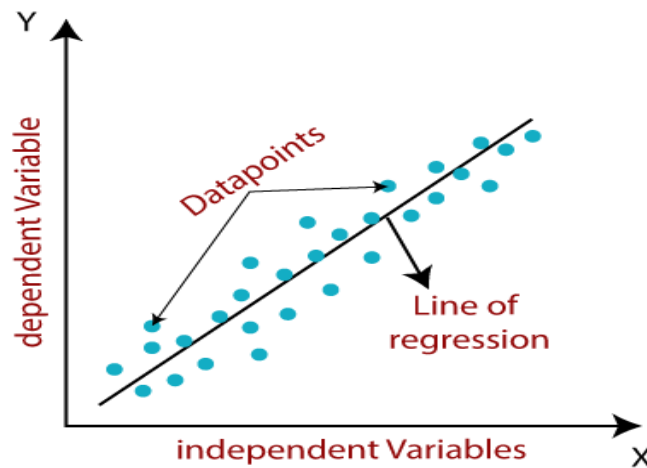


Figure-5

5.4 Clustering :

Clustering, at its core, organizes data into cohesive groups sharing common traits, unveiling patterns essential for targeted strategies and deeper insights across various domains and industries.

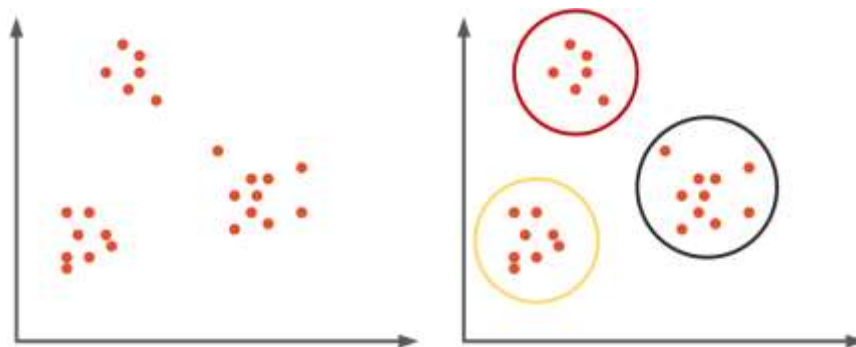


Figure-6

5.5 Bias:

It is used to predict the error rate. There are mainly three types of bias

1. High Bias → Underfitting
2. No Bias → Best fitting
3. Low Bias → Overfitting

Biassing in machine learning involves intentionally influencing a model to favor

particular outcomes or features, potentially affecting its predictions.



Figure-7

5.6 Summarization:

Summarization methods like ``info()`` and ``describe()`` offer distinct perspectives: ``info()`` provides structural details (columns, types), while ``describe()`` focuses on statistical summaries (mean, std), aiding in initial dataset comprehension and analysis readiness.

5.7 ML REPRODUCIBILITY:

It is a Process of Performing various operations with Original Methods, then ultimately, we get Original Data as Result, (Shadow & Deep Copy may not be Applicable).

5.8 INTERPRETABILITY:

To Find the "Optimize Solution" from Various Solutions we choose the best solution among various method operations output.

5.9 Statistical Operations: Applying analytical techniques to extract insights from data.

Data Visualization: Representing data visually for easier understanding

CHAPTER-6

DESIGN

6.1 ER MODEL

6.1.1 ENTITIES ARE:

- **Student Entity:**

Attributes: Student_id, Student_name, Date_Of_Birth, Gender,
Enrollment_Date, Graduation_Date, Class_Level

- **Parent/Guardian Entity:**

Attributes: Parent_Guardian_Name, Parent_Guardian_Phone

- **Course Entity:**

Attributes: Course_Id, Course_Name

- **Enroll Entity:**

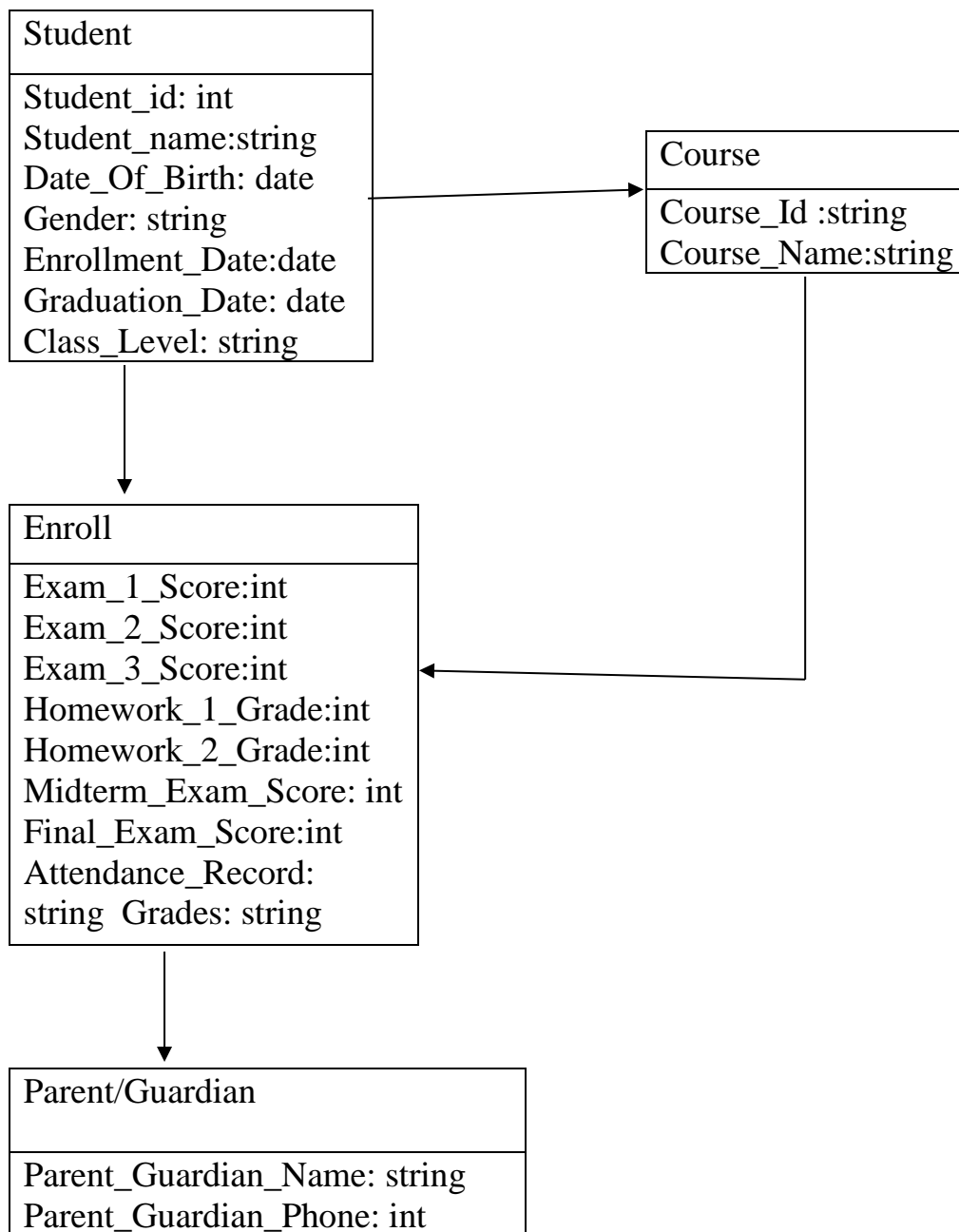
Attributes: Exam_1_Score, Exam_2_Score, Exam_3_Score,
Homework_1_Grade, Homework_2_Grade, Midterm_Exam_Score,
Final_Exam_Score, Attendance_Record, Grades.

6.1.2 RELATIONSHIPS:

- Student (One) - Enroll (Many) relationship through Student_id
- Student (One) - Parent/Guardian (Many) relationship through Student_id
- Course (One) - Enroll (Many) relationship through Course_Id

6.1.3 UML DIAGRAMS:

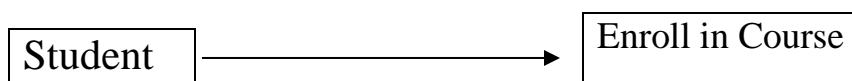
Attributes are linked to respective entities in the diagram to showcase their connections and dependencies.



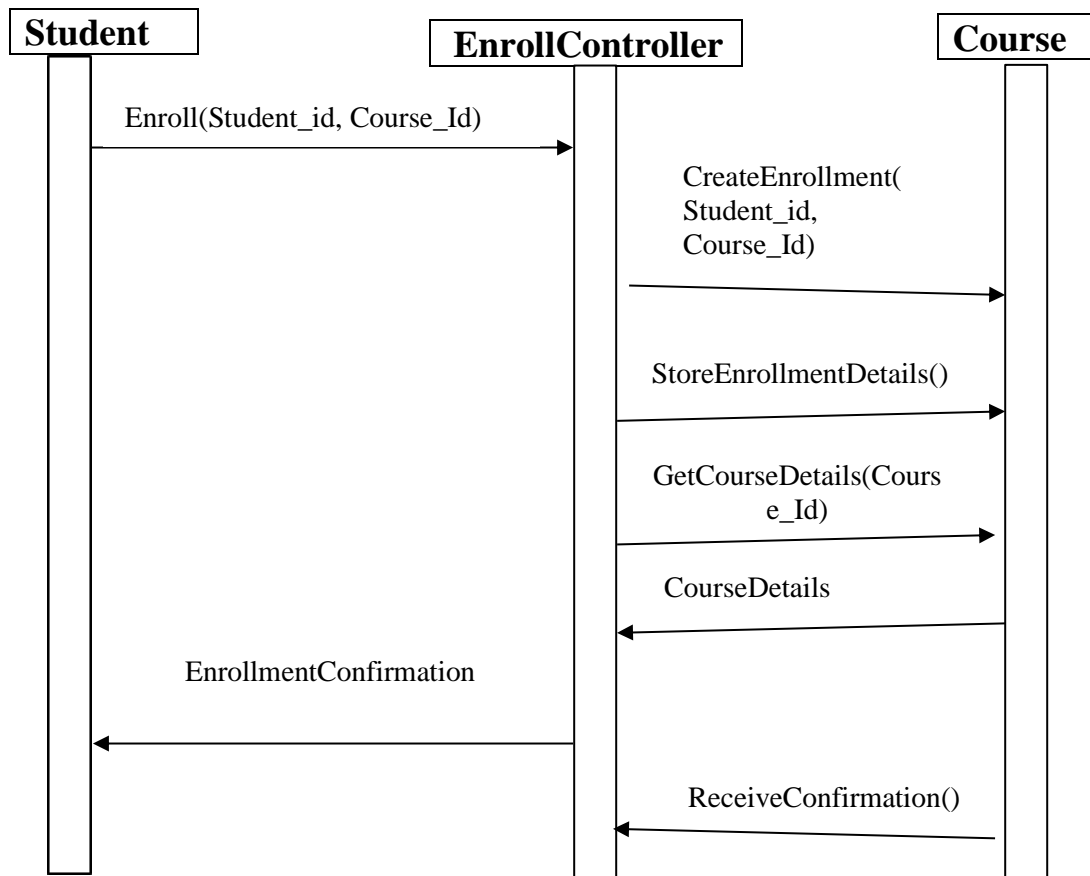
6.1.4 USE CASE DIAGRAM:

Use Case: Student Enrolls in Course

- **Actors:**
 - Student
- **Use Case:**
 - Enroll in Course

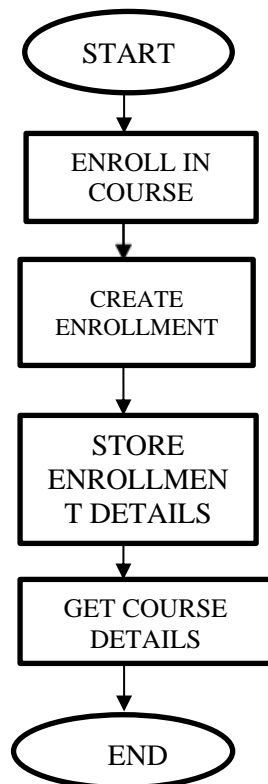


6.1.5 SEQUENCE DIAGRAMS:



6.1.6 ACTIVITY DIAGRAM :

DESIGN:



CHAPTER-7

METHODOLOGY

METHODOLOGY: Here which are required to do project on educational field based

- **Data Collection:** Gather diverse student data (grades, demographics, attendance).
- **Data Cleaning:** Remove errors, handle missing values for accuracy.
- **Feature Selection:** Identify key indicators of academic performance.
- **Feature Engineering:** Create new features (averages, trends) for insights.
- **Exploratory Data Analysis:** Visualize patterns, correlations, and outliers.
- **Normalization & Encoding:** Scale numerical data, encode categorical features.
- **Model Building:** Use ML models (classification, regression) for predictions.
- **Model Evaluation:** Assess model accuracy and performance metrics.
- **Refinement:** Iterate on features, models for enhanced predictions.
- **Interpretation & Deployment:** Analyze results, deploy model for use.

7.1 DATA SET GENERATION:

- The importance of diverse datasets lies in showcasing the versatility and adaptability of feature engineering techniques across various domains and data types. And these Diverse datasets are crucial to showcase how feature engineering techniques adapt to different data types and domains, enhancing model performance. The varied data allows for customized transformations, promoting scalability and generalization across unseen datasets. By demonstrating adaptability, feature engineering proves its versatility in extracting meaningful insights from a broad spectrum of data.

SYNTAX OF DATA FRAME:

pandas.DataFrame(data,index,columns, dtype,copy)
DATA-->Info , INDEX-->Val ,COLUMNS--> Operations on it
indicating , dtype--> Data type specifying, copy-->duplication

7.2 CORRELATION:

- It is a Relation between two Column Data Members in a Dataset.
- 2. We use a Method called "corr()"
- 3.It is a Scaled Form of a Covariance.
- 4.The Correlation Values are lies between " -1 and +1 ".

SYNTAX: attribute1.corr(attribute2)

TYPES OF CORRELATIONS:

- 1.POSITIVE(0 TO 1 in between values we got means positive)
if values in the range of 0 to +0.5 then it said to be "High positive".
if values in the range of +0.5 to +1 then it is said to be "Low positive".
- 2.NEGATIVE(-1 to 0 in between we got means negative)
if values in the range of 0 to -0.5 then it is said to be "High negative".
if values in the range of -0.5 to -1 then it is said to be "Low negative".
- 3.NO CORRELATION

FORMULA TO CALCULATE:

$$r = \frac{n(\sum xy) - (\sum x)(\sum y)}{\sqrt{[n \sum x^2 - (\sum x)^2][n \sum y^2 - (\sum y)^2]}}$$

Where n = Quantity of Information

Σx = Total of the First Variable Value

Σy = Total of the Second Variable Value

Σxy = Sum of the Product of first & Second Value

Σx^2 = Sum of the Squares of the First Value

Σy^2 = Sum of the Squares of the Second Value

7.3 COVARIANCE: cov()

THE RELATION BETWEEN TWO DATA MEMBERS WHICH ARE IN TWO DIFFERENT ATTRIBUTES.

- it is a measurement of correlation.
- individual elements of the data from 2 columns.
- the result may vary every second.
- it's range in between $-\infty$ to $+\infty$.

FORMULA TO CALCULATE THE COVARIANCE:

$$cov_{x,y} = \frac{\sum (x_i - \bar{x})(y_i - \bar{y})}{N - 1}$$

$cov_{x,y}$ = covariance between variable x and y

x_i = data value of x

y_i = data value of y

\bar{x} = mean of x

\bar{y} = mean of y

N = number of data values

CHAPTER-8

CODING AND TESTING

8.1 DATA COLLECTION:

8.1.1 DATASET GENERATION :

```
import pandas as pd
```

```
pd.set_option('display.max_columns', None)
```

```
pd.set_option('display.max_rows', None)
```

```
data = { 'Student_id': pd.Series([101, 102, 103, 104, 105, 106, 107, 108, 109, 110, 111, 112, 113, 114, 115, 116, 117, 118, 119, 120, 121, 122, 123, 124, 125, 126, 127, 128, 129, 130, 131, 132, 133, 134, 135, 136, 137, 138, 139, 140, 141, 142, 143, 144, 145, 146, 147, 148, 149, 150, 151, 152, 153, 154, 155, 156, 157, 158, 159, 160, 161, 162, 163, 164, 165, 166, 167, 168, 169, 170, 171, 172, 173, 174, 175, 176, 177, 178, 179, 180, 181, 182, 183, 184, 185, 186, 187, 188, 189, 190, 191, 192, 193, 194, 195, 196, 197, 198, 199, 200]),
```

```
'Student_name': pd.Series(['Nandu', 'Navya', 'Mythili', 'Pujitha', 'Yashaswi', 'Kavya', 'Saritha', 'Murali', 'Mohan', 'Das', 'Anju', 'Sravani', 'Shiny', 'Mani', 'Prani', 'Deepu', 'Vasu', 'Moulika', 'Vinaya', 'Mahitha', 'Vicky', 'Lucky', 'Hitesh', 'Vinay', 'Yamini', 'Phani', 'Ramya', 'Vyshu', 'Radha', 'Ruchi', 'Ramesh', 'Suresh', 'Lakshmi', 'Krishna', 'Rama', 'Aruna', 'Anand', 'Rajesh', 'Priya', 'Sarala', 'Rajendra', 'Swathi', 'Naveen', 'Padma', 'Satish', 'Anjali', 'Prakash', 'Meena', 'Kiran', 'Asha', 'Sudhir', 'Sowmya', 'Venkat', 'Divya', 'Ravi', 'Indira', 'Vijay', 'Sunita', 'Mahesh', 'Shalini', 'Prasad', 'Arundhati', 'Ganga', 'Sujatha', 'Raju', 'Sangeeta', 'Krishnan', 'Sahana', 'Mohan', 'Anuradha', 'Harish', 'Geetha', 'Narendra', 'Srilatha', 'Srinivas', 'Jyothi', 'Raghav', 'Latha', 'Rajkumar', 'Madhavi', 'Deepak', 'Siri', 'Harsha', 'Rani', 'Surya', 'Anusha', 'Raghava', 'Vani', 'Ramesh', 'Sangeetha', 'Kishore', 'Harish', 'Geetha', 'Narendra', 'Srilatha', 'Srinivas', 'Jyothi', 'Raghav', 'Latha', 'Rajkumar']),
```

```
'Date_Of_Birth': pd.Series(['2002-09-01', '2003-01-01', '2002-01-31', '2004-05-18', '2005-09-27', '2002-04-19', '2007-02-11', '2008-08-31', '2009-06-25', '2010-04-16', '2011-10-07', '2012-01-19', '2013-07-14', '2014-09-03', '2015-03-28', '2016-06-12', '2017-12-08', '2018-08-22', '2019-02-02', '2020-11-30', '2001-04-05', '2002-08-13', '2003-10-26', '2004-04-09', '2005-07-21', '2006-12-20', '2007-01-15', '2008-09-02', '2009-05-07', '2010-03-11', '2011-11-29', '2012-01-01', '2013-08-06', '2014-10-17', '2015-04-24', '2016-05-08', '2017-12-21', '2018-07-02', '2019-03-09', '2020-10-14', '2001-05-02', '2002-08-27', '2003-09-19', '2004-03-14', '2005-06-29', '2006-11-25', '2007-02-03', '2008-09-16', '2009-04-21', '2010-02-28', '2011-12-15', '2012-02-23', '2013-09-11', '2014-11-05', '2015-04-30', '2016-05-25', '2017-10-11', '2018-06-05', '2019-01-28', '2020-09-05', '2001-05-21', '2002-09-05', '2003-09-30', '2004-02-10', '2005-06-06', '2006-11-18', '2007-03-20', '2008-09-28', '2009-04-04', '2010-02-15', '2011-12-22', '2012-03-07', '2013-09-27', '2014-11-28', '2015-05-10', '2016-04-03', '2017-09-14', '2018-07-18', '2019-01-09', '2020-08-19', '2001-06-14', '2002-09-22', '2003-10-14', '2004-01-26', '2005-05-01', '2006-10-07', '2016-03-10', '2017-08-05', '2018-06-29', '2019-01-20', '2020-08-09', '2009-04-04', '2010-02-15', '2011-12-22', '2012-03-07', '2013-09-27', '2014-11-28', '2015-05-10',
```



```
'2016-04-03','2017-09-14']]),
```

**'Gender': pd.Series(['Female', 'Female', 'Female', 'Female', 'Female', 'Female', 'Female', 'Female', 'Male',
'Male', 'Male','Female', 'Female', 'Female', 'Male', 'Female', 'Female', 'Female', 'Female',
'Female','Male', 'Male', 'Male','Male','Female', 'Male', 'Female', 'Female', 'Female', 'Female',
'Male','Male', 'Female','Male', 'Female', 'Female', 'Male', 'Male', 'Female', 'Female','Male','Female',
'Male', 'Female', 'Male', 'Female', 'Male', 'Female', 'Male', 'Female', 'Male','Female','Male',
'Female', 'Male', 'Female', 'Male','Female', 'Male', 'Female', 'Male', 'Female', 'Female',
'Female', 'Male', 'Female', 'Male', 'Female', 'Male', 'Female', 'Male', 'Female', 'Male', 'Female',
'Male', 'Female', 'Male', 'Female', 'Male', 'Female', 'Male', 'Female', 'Male', 'Female',
'Male', 'Female', 'Male', 'Female', 'Male', 'Female', 'Male', 'Male','Female','Male', 'Female','Male',
'Female','Male', 'Female', 'Male'])),**

'Parent_Guardian_Name':pd.Series(['John','Mary','Michael','Jennifer','David','Lisa','James','Sarah','Robert','Laura','William','Emily','Joseph','Jessica','Richard','Kimberly','Charles','Rebecca','Thomas','Amanda','Daniel','Nicole','Matthew','Michelle','Donald','Angela','Steven','Melissa','George','Stephanie','Kenneth','Amy','Edward','Heather','Brian','Elizabeth','Ronald','Patricia','Anthony','Linda','Kvin','Susan','Jason','Karen','Jeffrey','Cynthia','Timothy','Christine','Mark','Donna','Paul','Pamela','Christopher','Deborah','Scott','Sandra','Eric','Teresa','Daniel','Sharon','Stephen','Nancy','Raymond','Kathleen','Gregory','Carol','Joshua','Betty','Jerry','Catherine','Dennis','Debra','Walter','Janet','Peter','Margaret','Patrick','Ruth','Ryan','Dorothy','Gary','Shirley','Jose','Martha','Larry','Frances','Jeffrey','Anna','Brandon','Theresa','Samuel','Virginia','Nicholas','Brenda','Benjamin','Srinu','Venkat','S ubbu','Chandra','Praveen']]),

'Parent_Guardian_Phone':pd.Series([903456789,807654321,955123456,754987321,987654123,823987456,956321987,987123456,789654321,821654987,806789123,789123456,954123789,911456987,987321456,954321789,789456123,973654987,856987123,987456321,789321654,921987654,856123789,954789321,803789654,789456321,956987654,654123987,987456789,723987654,789654123,954321456,956789654,821654789,987123654,789321456,956654321,893987654,987654789,754789123,921456789,789654321,723456987,856321789,987321654,789123654,954321987,987654132,923654798,856879321,987465321,789965432,921459876,856981732,987316452,789132654,754789321,987654312,923978456,856987123,789321465,954132879,987465213,789987654,921654789,856789132,987654213,789654312,923987546,854321789,987654213,789654321,921987456,856123789,987654132,923654798,856879321,987465321,789965432,921459876,956981732,987316452,789132654,854789321,987654312,823978456,756987123,789321465,954132879,987465213,789987654,921654789,856789132,987654213,789654312,923987546,9988776655,9933442211,9876523416,9645372748]),

'Address': `pd.Series(['Jubilee Road', 'Satya Nagar', 'Green Valley Lane', 'Krishna Enclave', 'Golden Temple Street', 'Rajendra Prasad Avenue', 'Coastal Highway', 'Gopal Rao Street', 'Surya Nagar', 'Radha Krishna Lane', 'Vijaya Street', 'Lotus Blossom Road', 'Nandi Hills Avenue', 'Indira`

[illegible][illegible]

```
'Zip_code': pd.Series([ 520001, 520002, 520003, 520004, 520005, 520006, 520007, 520008,
520009, 520010, 520011, 520012, 520013, 520014, 520015, 530001, 530002,530003, 530004,
530005, 530006, 530007, 530008, 530009, 530010, 530011, 530012, 530013, 530014, 530015,
515001, 515002, 515003,515004, 515005, 515006, 515007, 515008, 515009, 515010, 515011,
515012, 515013, 515014, 515015, 531001, 531002, 531003, 531004,531005, 531006, 531007,
531008, 531009, 531010, 531011, 531012, 531013, 531014, 531015, 524001, 524002, 524003,
524004, 524005, 524006,524007, 524008, 524009, 524010, 524011, 524012, 524013, 524014,
524015, 533001, 533002, 533003, 533004, 533005, 533006, 533007, 533008,533009, 533010,
533011, 533012, 533013, 533014, 533015, 515501, 515502, 515503, 515504, 515505, 515506,
515507, 515508, 515509, 515510]),
```

[illegible]

'Miss Lauren Wilson', 'Mr. Aaron Adams', 'Mrs. Janet Turner', 'Mr. Jeffrey Martinez', 'Miss Amy Harris', 'Mr. Charles Young', 'Mrs. Elizabeth Lopez', 'Mr. Stephen Clark', 'Miss Brittany Turner', 'Mr. Douglas Anderson', 'Mrs. Mary Smith', 'Mr. Ronald White', 'Miss Nicole Taylor', 'Mr. Larry Davis', 'Mrs. Susan Johnson', 'Mr. Joshua Thomas', 'Miss Christina Robinson', 'Mr. Bryan Wilson', 'Mrs. Angela Martinez', 'Mr. Gerald Anderson', 'Miss Melissa Hall', 'Mr. Kevin Turner', 'Mrs. Virginia Lewis', 'Mr. Wayne Garcia', 'Miss Amanda Scott', 'Mr. Carl Davis', 'Mrs. Michelle Turner', 'Mr. Samuel Young', 'Miss Jessica Brown', 'Mr. Patrick Martin', 'Mrs. Lisa Adams', 'Mr. Henry Rodriguez', 'Miss Laura Lee', 'Mr. Zachary King']),

'Teacher_Email':pd.Series(['Mrs. Sarah Johnson@gmail.com', 'Mr. David Smith@gmail.com', 'Miss Emily Wilson@gmail.com', 'Mr. Robert Brown@gmail.com', 'Mrs. Jennifer Taylor@gmail.com', 'Mr. Michael Anderson@gmail.com', 'Miss Jessica Lee@gmail.com', 'Mr. William Martinez@gmail.com', 'Mrs. Susan Harris@gmail.com', 'Mr. Richard Clark@gmail.com', 'Miss Laura White@gmail.com', 'Mr. Daniel Davis@gmail.com', 'Mrs. Karen Rodriguez@gmail.com', 'Mr. Thomas Turner@gmail.com', 'Miss Maria Moore@gmail.com', 'Mr. Charles Allen@gmail.com', 'Mrs. Patricia Scott@gmail.com', 'Mr. Joseph King@gmail.com', 'Miss Amanda Garcia@gmail.com', 'Mr. Christopher Young@gmail.com', 'Mrs. Linda Wilson@gmail.com', 'Mr. Edward Thompson@gmail.com', 'Miss Sarah Lewis@gmail.com', 'Mr. Nicholas Hall@gmail.com', 'Mrs. Lisa Lopez@gmail.com', 'Mr. John Jackson@gmail.com', 'Miss Ashley Martin@gmail.com', 'Mr. Daniel Lewis@gmail.com', 'Mrs. Kimberly Adams@gmail.com', 'Mr. Anthony Walker@gmail.com', 'Miss Emily Turner@gmail.com', 'Mr. Matthew Brown@gmail.com', 'Mrs. Michelle Martinez@gmail.com', 'Mr. Jason Anderson@gmail.com', 'Miss Jessica Robinson@gmail.com', 'Mr. Eric Hall@gmail.com', 'Mrs. Nancy Harris@gmail.com', 'Mr. Timothy Wright@gmail.com', 'Miss Olivia Davis@gmail.com', 'Mr. Andrew Moore@gmail.com', 'Mrs. Donna Johnson@gmail.com', 'Mr. Brian Lewis@gmail.com', 'Miss Megan Allen@gmail.com', 'Mr. Christopher Wilson@gmail.com', 'Mrs. Sandra Clark@gmail.com', 'Mr. Ronald Martin@gmail.com', 'Miss Rachel Turner@gmail.com', 'Mr. George Anderson@gmail.com', 'Mrs. Carol Taylor@gmail.com', 'Mr. Benjamin King@gmail.com', 'Miss Victoria White@gmail.com', 'Mr. Gregory Robinson@gmail.com', 'Mrs. Deborah Davis@gmail.com', 'Mr. Joseph Thomas@gmail.com', 'Miss Katherine Garcia@gmail.com', 'Mr. Dennis Anderson@gmail.com', 'Mrs. Ruth Scott@gmail.com', 'Mr. Paul Turner@gmail.com', 'Miss Stephanie Martin@gmail.com', 'Mr. Frank Rodriguez@gmail.com', 'Mrs. Heather Lee@gmail.com', 'Mr. Jonathan Harris@gmail.com', 'Miss Julia Young@gmail.com', 'Mr. Mark Turner@gmail.com', 'Mrs. Pamela Davis@gmail.com', 'Mr. Steven Walker@gmail.com', 'Miss Lauren Wilson@gmail.com', 'Mr. Aaron Adams@gmail.com', 'Mrs. Janet Turner@gmail.com', 'Mr. Jeffrey Martinez@gmail.com', 'Miss Amy Harris@gmail.com', 'Mr. Charles Young@gmail.com', 'Mrs. Elizabeth Lopez@gmail.com', 'Mr. Stephen Clark@gmail.com', 'Miss Brittany Turner@gmail.com', 'Mr. Douglas Anderson@gmail.com', 'Mrs. Mary Smith@gmail.com', 'Mr. Ronald White@gmail.com', 'Miss Nicole Taylor@gmail.com', 'Mr. Larry Davis@gmail.com', 'Mrs. Susan Johnson@gmail.com', 'Mr. Joshua Thomas@gmail.com', 'Miss Christina Robinson@gmail.com', 'Mr. Bryan Wilson@gmail.com', 'Mrs. Angela Martinez@gmail.com', 'Mr. Gerald Anderson@gmail.com', 'Miss Melissa Hall@gmail.com', 'Mr. Kevin Turner@gmail.com', 'Mrs. Virginia Lewis@gmail.com', 'Mr. Wayne Garcia@gmail.com', 'Miss Amanda

Scott@gmail.com', 'Mr. Carl Davis@gmail.com', 'Mrs. Michelle Turner@gmail.com', 'Mr. Samuel Young@gmail.com', 'Miss Jessica Brown@gmail.com', 'Mr. Patrick Martin@gmail.com', 'Mrs. Lisa Adams@gmail.com', 'Mr. Henry Rodriguez@gmail.com', 'Miss Laura Lee@gmail.com', 'Mr. Zachary King@gmail.com']],

'Teacher_mobile_Number':pd.Series([123456789,987654321,555123456,654987321,987654123,123987456,456321987,987123456,789654321,321654987,456789123,789123456,654123789,321456987,987321456,654321789,789456123,123654987,456987123,987456321,789321654,321987654,456123789,654789321,123789654,789456321,456987654,654123987,987456789,123987654,789654123,654321456,456789654,321654789,987123654,789321456,456654321,123987654,987654789,654789123,321456789,789654321,123456987,456321789,987321654,789123654,654321987,987654132,123654798,456879321,987465321,789965432,321459876,456981732,987316452,789132654,654789321,987654312,123978456,456987123,789321465,654132879,987465213,789987654,321654789,456789132,987654213,789654312,123987546,654321789,987654213,789654321,321987456,456123789,987654132,123654798,456879321,987465321,789965432,321459876,456981732,987316452,789132654,654789321,987654312,123978456,456987123,789321465,654132879,987465213,789987654,456789132,987654213,789654312,723987546,956789132,8074254213,7967654312,983987546,4591908652])),

'Class_Number':pd.Series([101, 203, 305, 407, 509, 611, 713, 815, 917, 1019, 1121, 1223, 1325, 1427, 1529, 1631, 1733, 1835,1937, 2039, 2141, 2243, 2345, 2447, 2549, 2651, 2753, 2855, 2957, 3059, 3161, 3263, 3365, 3467, 3569,3671, 3773, 3875, 3977, 4079, 4181, 4283, 4385, 4487, 4589, 4691, 4793, 4895, 4997, 5099, 5201, 5303, 5405, 5507, 5609, 5711, 5813, 5915, 6017, 6119, 6221, 6323, 6425, 6527, 6629, 6731, 6833, 6935, 7037,7139, 7241, 7343, 7445, 7547, 7649, 7751, 7853, 7955, 8057, 8159, 8261, 8363, 8465, 8567, 8669, 8771, 8873, 8975, 9077, 9179, 9281, 9383, 9485, 9587, 9689, 9791, 9893, 9995, 10097, 10199])),

'Attendance_Record':pd.Series(['82%', '67%', '45%', '59%', '73%', '91%', '52%', '78%', '64%', '86%', '55%', '68%', '71%', '49%', '89%', '42%', '76%', '61%', '58%', '87%', '43%', '75%', '70%', '53%', '92%', '66%', '80%', '63%', '84%', '48%', '57%', '69%', '88%', '44%', '72%', '65%', '50%', '83%', '46%', '74%', '77%', '51%', '93%', '62%', '90%', '54%', '79%', '47%', '94%', '85%', '41%', '96%', '56%', '72%', '65%', '50%', '83%', '46%', '74%', '81%', '60%', '98%', '99%', '72%', '65%', '50%', '83%', '46%', '74%', '97%', '40%', '95%', '67%', '45%', '59%', '73%', '91%', '52%', '72%', '65%', '50%', '83%', '46%', '74%', '82%', '67%', '45%', '59%', '73%', '91%', '52%', '78%', '64%', '86%', '55%', '68%', '71%', '49%', '89%', '42%']),

'Exam_1_Score': pd.Series([71, 88, 63, 94, 52, 78, 69, 85, 74, 91, 59, 82, 66, 76, 49, 97, 57, 84, 42, 72, 60, 89, 67, 81, 45,98, 54, 80, 58, 92, 50, 75, 47, 79, 61, 90, 55, 70, 86, 65, 95, 53, 73, 48,

68, 87, 64, 93, 56, 77,62, 83, 41, 99, 44, 96, 51, 40, 70, 88, 65, 73, 56, 89, 47, 68, 75, 61, 93, 49, 70, 57, 84, 55, 82,66, 78, 62, 91, 53, 69, 76, 67, 88, 41, 79, 74, 60, 87, 52, 81, 58, 86, 45, 80, 59, 72, 67, 93, 59]],

'Exam_2_Score': pd.Series([85, 96, 75, 64, 90, 81, 70, 67, 73, 59, 43, 91, 87, 50, 69, 79, 100, 60, 63, 61, 80, 57, 47, 95, 45,74, 51, 46, 66, 54, 37, 56, 83, 49, 77, 98, 52, 44, 82, 35, 84, 92, 76, 71, 97, 65, 89, 78, 53, 68,41, 86, 99, 55, 62, 72, 48, 94, 88, 58, 42, 85, 96, 75, 64, 90, 81, 70, 67, 73, 59, 43, 91, 87, 50,69, 79, 100, 60, 63, 61, 80, 57, 47, 95, 45, 74, 51, 46, 66, 54, 37, 56, 83, 49, 77, 98, 52, 44, 82]),

'Exam_3_Score': pd.Series([63, 98, 72, 53, 85, 90, 69, 79, 45, 58, 87, 55, 84, 67, 40, 76, 88, 62, 100, 95, 70, 57, 75, 81, 93,41, 66, 80, 73, 54, 49, 94, 60, 71, 64, 86, 48, 68, 52, 37, 61, 89, 59, 77, 74, 44, 50, 78, 35, 82,56, 65, 51, 47, 91, 92, 46, 83, 42, 76, 64, 88, 53, 92, 71, 62, 59, 84, 45, 55, 96, 80, 90, 72, 70,58,67, 97, 41, 66, 95, 87, 61, 68, 79, 73, 48, 89, 81, 86, 75, 50, 74, 63, 51, 100, 78, 42, 47, 65]),

'Homework_1_Grade': pd.Series([84, 92, 83, 70, 83, 78, 76, 65, 88, 83, 97, 86, 60, 73, 81, 66, 80, 66, 84, 65, 69, 86, 78, 77,82, 79, 91, 72, 96, 72, 74, 75, 90, 66, 86, 94, 93, 80, 68, 89, 63, 60, 95, 72, 93, 88, 68, 77,71, 98, 76, 66, 99, 78, 93, 99, 69, 98, 77, 69, 82, 94, 93, 94, 97, 94, 93, 81, 63, 88, 96, 62,84, 97, 66, 83, 91, 75, 99, 78, 63, 68, 69, 92, 80, 70, 79, 82, 81, 74, 62, 94, 74, 73, 64, 83,69, 87, 70, 81]),

'Homework_2_Grade': pd.Series([81, 98, 75, 80, 87, 70, 98, 72, 66, 71, 85, 98, 79, 63, 86, 91, 70, 61, 75, 62, 60, 81, 92, 88,77, 66, 60, 91, 95, 76, 100, 71, 93, 93, 64, 82, 86, 74, 100, 81, 72, 78, 73, 81, 95, 70, 80, 92,62, 87, 96, 63, 95, 92, 81, 91, 88, 92, 97, 67, 61, 95, 82, 76, 69, 84, 89, 66, 62, 79, 74, 82,70, 75, 71, 90, 68, 95, 81, 92, 98, 92, 79, 65, 98, 98, 75, 97, 76, 86, 83, 82, 81, 96, 66, 70,60, 62, 86, 74]),

'Homework_3_Grade': pd.Series([84, 92, 83, 70, 83, 78, 76, 65, 88, 83, 97, 86, 60, 73, 81, 66, 80, 66, 84, 65, 69, 86, 78, 77,82, 79, 91, 72, 96, 72, 74, 75, 90, 66, 86, 94, 93, 80, 68, 89, 63, 60, 95, 72, 93, 88, 68, 77, 71, 98, 76, 66, 99, 78, 93, 99, 69, 98, 77, 69, 82, 94, 93, 94, 97, 94, 93, 81, 63, 88, 96, 62,84, 97, 66, 83, 91, 75, 99, 78, 63, 68, 69, 92, 80, 70, 79, 82, 81, 74, 62, 94, 74, 73, 64, 83,69, 87, 70, 81]),

'Assignment_1_Score': pd.Series([7, 10, 29, 55, 15, 26, 57, 87, 13, 12, 51, 63, 75, 70, 99, 89, 32, 56, 27, 75, 25, 39, 82, 46, 40, 46, 47, 44, 53, 65, 41, 32, 44, 24, 15, 41, 95, 58, 86, 19, 81, 98, 50, 64, 93, 51, 11, 58, 52, 50, 62, 64, 82, 99, 24, 59, 58, 19, 85, 41, 38, 80, 31, 32, 21, 21, 22, 58, 96,99, 20, 76, 44, 45, 24, 18, 53, 83, 24, 79, 71, 39, 27, 40, 73, 17, 75, 62, 69, 51, 19, 69,29, 44, 98, 4, 2, 15, 63, 65]),

'Assignment_2_Score': pd.Series([78, 83, 18, 68, 73, 88, 76, 80, 48, 95, 34, 89, 72, 81, 79, 42, 31, 99, 84, 32, 32, 59, 56,99, 19, 83, 34, 67, 16, 38, 80, 42, 99, 59, 83, 54, 41, 17, 81, 80, 18, 99, 88, 96, 23, 89,59, 19, 83, 18, 96, 32, 90, 74, 77, 60, 47, 74, 41, 38, 34, 25, 79, 38, 13, 42, 74, 54, 59,51, 29, 41, 58, 80, 86, 99, 94, 90, 47, 58, 75, 75, 58, 83, 31, 68, 82, 32, 68, 99, 62, 4, 53, 26, 75,80,60,90,100,79]),

'Assignment_3_Score': pd.Series([21, 97, 33, 20, 60, 93, 41, 88, 84, 30, 23, 83, 89, 84, 60, 98, 24, 81, 39, 63, 45, 61, 92,73, 70, 41, 51, 67, 82, 89, 90, 45, 66, 24, 22, 74, 79, 91, 96, 71, 21, 20, 23, 53, 27, 40, 99, 43, 95, 75, 20, 64, 44, 35, 74, 86, 58, 74, 75, 88, 95, 39, 92, 86, 64, 76, 20, 56, 31,23,

57, 50, 27, 34, 57, 59, 98, 72, 23, 32, 99, 80, 26, 99, 63, 91, 64, 24, 62, 28, 34, 65,53, 69, 82,80,70,60,78,90]],

'Midterm_Exam_Score':pd.Series([80,100,223,390,405,215,131,159,267,370,170,160,230,160,168,115,194,150,262,260,360,266,300,147,160,103,275,362,272,278,372,141,88,52,80,168,266,455,500,450,406,457,410,364,386,186,162,160,289,449,313,330,231,417,453,213,375,257,366,411,156,207,195,118,170,271,369,222,311,170,139,125,147,194,115,136,235,371,392,469,399,146,158,168,148,249,373,260,218,249,259,289,279,437,360,370,161,237,260,179])),

'Final_Exam_Score': pd.Series([100, 160, 250, 400, 425, 245, 145, 234, 356, 390, 189, 190, 267, 189, 178, 134, 234, 189, 290, 289, 563, 887, 998, 476,668, 779, 871, 982, 470, 597, 890, 968, 889, 573, 788, 678, 771, 888,978, 763, 868, 892, 877, 773,784, 860, 973, 578,689, 774, 885, 794, 892, 867, 995, 986, 782,868, 698, 672, 784, 773, 690, 889, 882,961, 680, 781, 774,778, 997, 693,999, 577, 795, 869,671, 969, 860, 779, 666, 783, 861, 881, 878, 863, 993, 792, 1000, 777, 798, 785, 786, 772, 775,870,886, 871, 679, 990])),

'Final_Exam_GPA':pd.Series([3.7, 2.3, 3.0, 3.0, 2.7, 3.0, 3.3, 2.0, 3.7, 3.3, 3.7, 3.3, 3.0, 2.3, 3.0, 3.3, 2.3, 2.7, 3.3, 3.0,3.7, 3.3, 2.7, 3.0, 2.7, 3.0, 2.3, 3.3, 3.0, 3.3, 3.3, 3.0, 3.3, 3.0, 2.3, 2.3, 3.3, 3.0, 2.7, 2.3,3.0, 3.3, 3.7, 3.3, 3.3, 3.0, 3.0, 3.0, 3.3, 2.3, 3.0, 3.0, 2.3, 3.3, 2.3, 2.7, 3.0, 3.3, 2.7, 2.3,2.0, 2.7, 2.7, 3.0, 2.7, 2.3, 3.3, 2.3, 3.3, 3.7, 3.7, 2.7, 3.0, 3.0, 3.7, 2.7, 3.0, 3.0, 2.7, 3.0,2.3, 3.3, 3.0, 3.0, 3.0, 3.3, 3.3, 2.3, 2.7, 3.7, 2.3, 3.7, 3.0, 3.3, 2.7, 3.7, 2.7, 3.0, 4.0])),

'Grades':pd.Series(['A', 'C', 'B', 'B', 'C', 'B', 'B+', 'D', 'A', 'B+', 'A', 'B+', 'B', 'C', 'B', 'B+', 'C', 'C', 'B+', 'B', 'A', 'B+', 'C', 'B', 'C', 'B', 'C', 'B+', 'B', 'B+', 'B', 'B', 'B+', 'B', 'B', 'B+', 'B', 'B', 'B', 'B+', 'C', 'B', 'B', 'B', 'C', 'A', 'B+', 'B', 'B', 'C', 'B', 'C', 'B', 'C', 'D', 'C', 'C', 'B', 'C', 'B', 'B', 'B', 'B', 'B', 'B', 'B', 'B+', 'C', 'B', 'B', 'B', 'C', 'A', 'B+', 'B', 'B', 'C', 'B', 'C', 'B', 'C', 'D', 'C', 'C', 'B', 'C', 'B', 'C', 'B', 'B', 'B', 'B', 'B', 'B'],)

'Parent_Teacher_Meeting_Date':pd.Series(['2023-06-22', '2023-05-08', '2023-06-11', '2023-04-27', '2023-05-03', '2022-06-19','2022-05-22', '2022-05-29', '2022-06-12', '2022-06-05', '2022-07-17', '2022-07-24', '2022-07-31', '2022-08-07', '2022-08-14', '2022-08-21', '2022-08-28', '2022-09-04','2022-09-11', '2022-09-18', '2022-09-25', '2022-10-02', '2022-10-09', '2022-10-16','2022-10-23', '2022-10-30', '2022-11-06', '2022-11-13', '2022-11-20', '2022-11-27','2022-12-04', '2022-12-11', '2022-12-18', '2022-12-25', '2023-01-01', '2023-01-08','2023-01-15', '2023-01-22', '2023-01-29', '2023-02-05', '2023-02-12', '2023-02-19','2023-02-26', '2023-03-05', '2023-03-12', '2023-03-19', '2023-03-26', '2023-04-02', '2023-04-09', '2023-04-16', '2023-04-23', '2023-04-30', '2023-05-07', '2023-05-14','2023-05-21', '2023-05-28', '2023-06-04', '2023-06-11', '2023-06-18', '2023-06-25','2023-07-02', '2023-07-09', '2023-07-16', '2023-07-23', '2023-07-30', '2023-08-06','2023-08-13', '2023-08-20', '2023-08-27', '2023-09-03', '2023-09-10', '2023-09-17','2023-09-24', '2020-01-13', '2020-02-06', '2020-02-07', '2020-02-08', '2020-02-09', '2020-02-10', '2020-01-04', '2020-01-05', '2020-01-06', '2020-01-07', '2020-01-08','2020-01-09', '2020-01-10', '2020-02-11', '2020-03-01', '2020-03-15', '2020-03-29','2020-04-12', '2020-04-26', '2020-05-10', '2020-05-24', '2020-06-07', '2020-06-21','2020-07-05', '2020-07-19', '2020-08-02', '2020-08-16'])),

'Meeting_feedback':pd.Series(['Good', 'Good', 'Average', 'Good', 'Average', 'Excellent', 'Poor', 'Poor','Average','Excellent', 'Poor', 'Excellent', 'Good', 'Poor', 'Average', 'Average', 'Average', 'Good','Good', 'Poor', 'Good', 'Average', 'Average', 'Good', 'Excellent', 'Poor', 'Average', 'Excellent', 'Excellent', 'Poor', 'Average', 'Poor', 'Good', 'Good', 'Poor', 'Poor','Excellent', 'Excellent', 'Poor', 'Good', 'Average', 'Excellent', 'Poor', 'Excellent', 'Excellent', 'Excellent', 'Excellent'],)

'Excellent', 'Average', 'Poor', 'Poor','Average', 'Average', 'Good', 'Excellent', 'Poor', 'Excellent',
'Good', 'Average', 'Poor','Poor', 'Average', 'Poor', 'Good', 'Good', 'Good', 'Excellent', 'Average',
'Poor','Excellent', 'Excellent', 'Average', 'Good', 'Excellent', 'Poor', 'Good', 'Excellent','Poor', 'Good',
'Good', 'Excellent', 'Excellent', 'Good', 'Good', 'Good', 'Poor', 'Good','Average', 'Good', 'Good',
'Poor', 'Average', 'Poor', 'Excellent', 'Average', 'Average','Good', 'Average']),

**'Student_Tradiness_Count':pd.Series(['6', '6', '8', '3', '3', '10', '8', '9', '1', '3', '5', '3', '5', '7', '6', '5',
'5','7', '8', '2', '4', '9', '5', '5', '2', '1', '1', '8', '1', '5', '8', '9', '9', '0', '10','5', '2', '9', '1', '0', '3', '5', '1', '1', '10',
'10', '0', '0', '0', '10', '8','5', '2', '1', '8', '8', '1', '2', '0', '2', '6', '9', '7', '9', '9', '3', '7', '7','4', '1', '3', '9', '10',
'8', '2', '3', '3', '2', '10', '6', '9', '0', '2', '6', '6','1', '4', '6', '1', '6', '5', '0', '4', '8', '8', '7', '9', '4', '6', '4'])),**

**'Student_Absence_Record':pd.Series(['3', '7', '1', '4', '0', '8', '2', '9', '5', '6', '10', '1', '8', '3', '6', '0', '4',
'7','2', '9', '3', '5', '1', '2', '8', '0', '10', '4', '6', '9', '3', '7', '5', '1', '2', '8','0', '10', '4', '6', '9', '3', '7', '5', '1', '2',
'8', '0', '10', '4', '6', '9', '3', '7', '5', '1', '2', '8', '0', '10', '4', '6', '9', '3', '7', '5', '1', '2', '8', '0','10', '4', '6', '9',
'3', '7', '5', '1', '2', '8', '0', '10', '4', '6', '9','1','7','5','1','2','8','0','10','4','6','9','1','7','3','7'])),**

**'Library_Books_Checked_Out':pd.Series(['2023-01-11', '2023-12-12', '2023-11-25', '2023-12-
03', '2023-02-28', '2023-01-18','2023-12-10', '2023-12-15', '2023-07-17', '2023-04-19', '2023-06-
22', '2023-07-24','2023-11-19', '2023-03-30', '2023-02-17', '2023-01-07', '2023-03-25', '2023-07-
11','2023-04-22', '2023-11-01', '2023-04-19', '2023-09-27', '2023-03-14', '2023-12-09','2023-04-23',
'2023-03-23', '2023-09-30', '2023-07-02', '2023-11-12', '2023-02-27', '2023-07-02', '2023-12-05',
'2023-01-04', '2023-07-12', '2023-01-14', '2023-01-28','2023-11-02', '2023-10-08', '2023-07-29',
'2023-11-21', '2023-03-21', '2023-03-17', '2023-07-29', '2023-04-25', '2023-12-19', '2023-12-19',
'2023-01-19', '2023-08-13','2023-02-10', '2023-12-05', '2023-12-19', '2023-01-15', '2023-07-15',
'2023-05-03','2023-01-21', '2023-02-13', '2023-07-27', '2023-04-02', '2023-12-12', '2023-07-
24','2023-08-20', '2023-11-06', '2023-01-08', '2023-09-10', '2023-09-10', '2023-08-16','2023-11-01',
'2023-06-26', '2023-03-20', '2023-02-02', '2023-10-21', '2023-08-17', '2023-05-16', '2023-02-16',
'2023-01-25', '2023-11-20', '2023-11-06', '2023-05-09','2023-01-30', '2023-06-09', '2023-02-12',
'2023-04-15', '2023-12-03', '2023-08-22', '2023-03-29', '2023-03-22', '2023-06-15', '2023-03-25',
'2023-10-10', '2023-05-23','2023-02-08', '2023-07-08', '2023-02-23', '2023-11-11', '2023-12-06',
'2023-08-09','2023-05-02', '2023-08-27', '2023-04-20', '2023-01-11'])),**

**'Library_Books_Due_Date':pd.Series(['2023-02-01', '2023-12-28', '2023-11-29', '2023-12-28',
'2023-03-24', '2023-02-04','2023-12-15', '2024-01-09', '2023-08-13', '2023-05-10', '2023-06-25',
'2023-08-19','2023-11-20', '2023-04-21', '2023-03-11', '2023-01-12', '2023-04-12', '2023-08-09',
'2023-05-02', '2023-11-28', '2023-05-17', '2023-10-26', '2023-04-03', '2023-12-20','2023-05-08',
'2023-04-04', '2023-10-10', '2023-07-22', '2023-12-02', '2023-03-17', '2023-07-05', '2023-12-28',
'2023-01-25', '2023-07-20', '2023-02-01', '2023-02-20','2023-11-04', '2023-10-31', '2023-08-06',
'2023-12-07', '2023-04-06', '2023-03-29','2023-07-31', '2023-05-19', '2024-01-05', '2024-01-10',
'2023-02-05', '2023-08-19', '2023-02-25', '2023-12-25', '2024-01-12', '2023-02-11', '2023-07-27',
'2023-05-12', '2023-01-26', '2023-03-01', '2023-07-29', '2023-04-18', '2023-12-19', '2023-08-
14','2023-08-28', '2023-11-13', '2023-01-11', '2023-10-06', '2023-10-10', '2023-09-07','2023-11-07',
'2023-07-16', '2023-04-04', '2023-02-05', '2023-10-31', '2023-09-08','2023-05-25', '2023-03-10',
'2023-02-04', '2023-12-13', '2023-12-05', '2023-05-17','2023-02-02', '2023-06-24', '2023-03-04',
'2023-04-21', '2023-12-08', '2023-09-16', '2023-04-01', '2023-04-16', '2023-07-05', '2023-04-04',
'2023-10-12', '2023-06-10', '2023-02-23', '2023-07-16', '2023-03-20', '2023-11-25', '2023-12-24',
'2023-08-27','2023-05-20', '2023-09-23', '2023-04-29', '2023-02-05'])),**

'Extracurricular_Activity_1':pd.Series(['Photography Club', 'Drama Club', 'Chess Club', 'Music Band', 'Chess Club', 'Debate Team', 'Music Band', 'Math Club', 'Science Club', 'Music Band', 'Drama Club', 'Drama Club', 'Sports Team', 'Photography Club', 'Debate Team', 'Debate Team', 'Sports Team', 'Art Club', 'Math Club', 'Science Club', 'Sports Team', 'Coding Club', 'Chess Club', 'Sports Team', 'Debate Team', 'Drama Club', 'Debate Team', 'Art Club', 'Math Club', 'Chess Club', 'Chess Club', 'Drama Club', 'Photography Club', 'Drama Club', 'Photography Club', 'Photography Club', 'Music Band', 'Math Club', 'Chess Club', 'Chess Club', 'Math Club', 'Math Club', 'Chess Club', 'Drama Club', 'Sports Team', 'Coding Club', 'Debate Team', 'Sports Team', 'Drama Club', 'Drama Club', 'Chess Club', 'Art Club', 'Debate Team', 'Art Club', 'Art Club', 'Music Band', 'Sports Team', 'Science Club', 'Coding Club', 'Debate Team', 'Coding Club', 'Art Club', 'Photography Club', 'Coding Club', 'Drama Club', 'Chess Club', 'Coding Club', 'Art Club', 'Music Band', 'Math Club', 'Drama Club', 'Drama Club', 'Coding Club', 'Math Club', 'Chess Club', 'Debate Team', 'Coding Club', 'Science Club', 'Math Club', 'Science Club', 'Math Club', 'Sports Team', 'Science Club', 'Coding Club', 'Chess Club', 'Photography Club', 'Math Club', 'Music Band', 'Music Band', 'Music Band', 'Sports Team', 'Math Club', 'Photography Club', 'Coding Club', 'Math Club', 'Sports Team', 'Sports Team', 'Drama Club', 'Coding Club', 'Music Band'])),

'Extracurricular_Activity_2':pd.Series(['Coding Club', 'Debate Team', 'Music Band', 'Chess Club', 'Chess Club', 'Photography Club', 'Debate Team', 'Art Club', 'Drama Club', 'Science Club', 'Art Club', 'Music Band', 'Math Club', 'Debate Team', 'Chess Club', 'Music Band', 'Music Band', 'Debate Team', 'Math Club', 'Sports Team', 'Science Club', 'Sports Team', 'Music Band', 'Coding Club', 'Music Band', 'Music Band', 'Coding Club', 'Coding Club', 'Photography Club', 'Chess Club', 'Chess Club', 'Debate Team', 'Chess Club', 'Math Club', 'Math Club', 'Math Club', 'Science Club', 'Coding Club', 'Chess Club', 'Art Club', 'Math Club', 'Music Band', 'Art Club', 'Photography Club', 'Sports Team', 'Coding Club', 'Music Band', 'Music Band', 'Music Band', 'Math Club', 'Debate Team', 'Drama Club', 'Debate Team', 'Coding Club', 'Music Band', 'Chess Club', 'Science Club', 'Debate Team', 'Sports Team', 'Chess Club', 'Music Band', 'Music Band', 'Music Band', 'Sports Team', 'Math Club', 'Art Club', 'Sports Team', 'Coding Club', 'Science Club', 'Photography Club', 'Art Club', 'Chess Club', 'Chess Club', 'Art Club', 'Science Club', 'Coding Club', 'Music Band', 'Art Club', 'Science Club', 'Drama Club', 'Debate Team', 'Math Club', 'Coding Club', 'Music Band', 'Math Club', 'Math Club', 'Sports Team', 'Science Club', 'Sports Team', 'Math Club', 'Science Club', 'Debate Team', 'Coding Club', 'Sports Team', 'Art Club', 'Coding Club', 'Photography Club', 'Music Band', 'Photography Club', 'Science Club'])),

'Extracurricular_Activity_3':pd.Series(['Math Club', 'Math Club', 'Drama Club', 'Coding Club', 'Music Band', 'Science Club', 'Art Club', 'Music Band', 'Photography Club', 'Debate Team', 'Science Club', 'Sports Team', 'Chess Club', 'Drama Club', 'Sports Team', 'Art Club', 'Art Club', 'Debate Team', 'Art Club', 'Sports Team', 'Chess Club', 'Chess Club', 'Music Band', 'Math Club', 'Music Band', 'Debate Team', 'Debate Team', 'Music Band', 'Sports Team', 'Debate Team', 'Science Club', 'Music Band', 'Sports Team', 'Coding Club', 'Art Club', 'Sports Team', 'Drama Club', 'Art Club', 'Drama Club', 'Science Club', 'Chess Club', 'Art Club', 'Coding Club', 'Debate Team', 'Debate Team', 'Coding Club', 'Math Club', 'Drama Club', 'Chess Club', 'Music Band', 'Debate Team', 'Science Club', 'Science Club', 'Art Club', 'Coding Club', 'Math Club', 'Drama Club', 'Photography Club', 'Drama Club', 'Coding Club', 'Debate Team', 'Chess Club', 'Music Band', 'Sports Team', 'Art Club', 'Photography Club', 'Science Club', 'Coding Club', 'Math Club', 'Math Club', 'Music Band', 'Drama

Club', 'Music Band', 'Drama Club', 'Music Band', 'Art Club', 'Science Club', 'Drama Club', 'Music Band', 'Coding Club', 'Coding Club', 'Math Club', 'Drama Club', 'Photography Club', 'Debate Team', 'Science Club', 'Drama Club', 'Art Club', 'Science Club', 'Math Club', 'Art Club', 'Sports Team', 'Science Club', 'Art Club', 'Music Band', 'Chess Club', 'Debate Team', 'Debate Team', 'Coding Club', 'Debate Team'])),

'Sports_Team_1':pd.Series(['Softball', 'Swimming', 'Volleyball', 'Cross Country', 'Basketball', 'Track and Field', 'Soccer', 'Baseball', 'Tennis', 'Golf', 'Soccer', 'Track and Field', 'Cross Country', 'Basketball', 'Swimming', 'Track and Field', 'Softball', 'Soccer', 'Golf', 'Tennis', 'Volleyball', 'Cross Country', 'Basketball', 'Baseball', 'Swimming', 'Volleyball', 'Basketball', 'Golf', 'Softball', 'Soccer', 'Tennis', 'Swimming', 'Baseball', 'Volleyball', 'Cross Country', 'Track and Field', 'Soccer', 'Golf', 'Baseball', 'Tennis', 'Basketball', 'Track and Field', 'Track and Field', 'Basketball', 'Swimming', 'Cross Country', 'Tennis', 'Golf', 'Volleyball', 'Golf', 'Track and Field', 'Volleyball', 'Golf', 'Soccer', 'Basketball', 'Tennis', 'Cross Country', 'Soccer', 'Softball', 'Swimming', 'Track and Field', 'Soccer', 'Golf', 'Baseball', 'Cross Country', 'Tennis', 'Golf', 'Tennis', 'Basketball', 'Softball', 'Golf', 'Track and Field', 'Soccer', 'Volleyball', 'Basketball', 'Golf', 'Track and Field', 'Cross Country', 'Tennis', 'Soccer', 'Golf', 'Tennis', 'Cross Country', 'Swimming', 'Track and Field', 'Softball', 'Volleyball', 'Baseball', 'Softball', 'Baseball', 'Baseball', 'Tennis', 'Cross Country', 'Basketball', 'Golf', 'Track and Field', 'Swimming', 'Golf', 'Volleyball', 'Track and Field'])),

'Sports_Team_2':pd.Series(['Swimming', 'Cross Country', 'Golf', 'Basketball', 'Golf', 'Soccer', 'Cross Country', 'Golf', 'Volleyball', 'Basketball', 'Baseball', 'Swimming', 'Cross Country', 'Basketball', 'Track and Field', 'Golf', 'Golf', 'Soccer', 'Track and Field', 'Track and Field', 'Volleyball', 'Cross Country', 'Baseball', 'Swimming', 'Cross Country', 'Basketball', 'Track and Field', 'Tennis', 'Baseball', 'Golf', 'Track and Field', 'Track and Field', 'Swimming', 'Soccer', 'Track and Field', 'Cross Country', 'Soccer', 'Volleyball', 'Cross Country', 'Track and Field', 'Softball', 'Soccer', 'Baseball', 'Softball', 'Tennis', 'Cross Country', 'Softball', 'Tennis', 'Cross Country', 'Tennis', 'Baseball', 'Volleyball', 'Soccer', 'Basketball', 'Baseball', 'Softball', 'Volleyball', 'Baseball', 'Soccer', 'Swimming', 'Softball', 'Soccer', 'Track and Field', 'Track and Field', 'Soccer', 'Tennis', 'Basketball', 'Cross Country', 'Cross Country', 'Track and Field', 'Tennis', 'Basketball', 'Golf', 'Golf', 'Golf', 'Golf', 'Track and Field', 'Swimming', 'Swimming', 'Tennis', 'Swimming', 'Baseball', 'Tennis', 'Track and Field', 'Tennis', 'Swimming', 'Basketball', 'Golf', 'Baseball', 'Cross Country', 'Track and Field', 'Golf', 'Softball', 'Tennis', 'Cross Country', 'Swimming', 'Softball', 'Cross Country', 'Soccer', 'Volleyball'])),

'Sports_Team_3':pd.Series(['Golf', 'Tennis', 'Tennis', 'Tennis', 'Softball', 'Tennis', 'Softball', 'Tennis', 'Golf', 'Volleyball', 'Soccer', 'Tennis', 'Track and Field', 'Soccer', 'Tennis', 'Swimming', 'Volleyball', 'Golf', 'Tennis', 'Track and Field', 'Soccer', 'Soccer', 'Swimming', 'Volleyball', 'Golf', 'Tennis', 'Basketball', 'Tennis', 'Basketball', 'Swimming', 'Golf', 'Volleyball', 'Basketball', 'Baseball', 'Track and Field', 'Cross Country', 'Swimming', 'Volleyball', 'Baseball', 'Soccer', 'Softball', 'Soccer', 'Baseball', 'Volleyball', 'Softball', 'Swimming', 'Softball', 'Tennis', 'Basketball', 'Baseball', 'Basketball', 'Golf', 'Basketball', 'Golf', 'Golf', 'Baseball', 'Tennis', 'Volleyball', 'Baseball', 'Track and Field', 'Tennis', 'Basketball', 'Track and Field', 'Volleyball', 'Golf', 'Baseball', 'Baseball', 'Baseball', 'Softball', 'Track and Field', 'Softball', 'Swimming', 'Volleyball', 'Track and Field', 'Cross Country', 'Baseball', 'Softball', 'Tennis', 'Golf', 'Tennis', 'Track and Field', 'Softball', 'Softball', 'Basketball', 'Volleyball', 'Tennis', 'Volleyball', 'Cross Country', 'Track and Field', 'Swimming',

'Golf', 'Tennis', 'Tennis', 'Basketball', 'Swimming', 'Swimming', 'Tennis', 'Soccer', 'Soccer', 'Track and Field'])),

'Bus_Route':pd.Series(['Jubilee Road - (Route 7)', 'Satya Nagar - (Route 3)', 'Green Valley Lane - (Route 7)', 'Krishna Enclave - (Route 9)', 'Golden Temple Street - (Route 8)', 'Rajendra Prasad Avenue - (Route 2)', 'Coastal Highway - (Route 6)', 'Gopal Rao Street - (Route 9)', 'Surya Nagar - (Route 3)', 'Radha Krishna Lane - (Route 3)', 'Vijaya Street - (Route 4)', 'Lotus Blossom Road - (Route 2)', 'Nandi Hills Avenue - (Route 10)', 'Indira Park Lane - (Route 3)', 'Murali Nagar - (Route 8)', 'Gandhi Marg - (Route 8)', 'Chamundi Vihar - (Route 7)', 'Sai Baba Lane - (Route 5)', 'NTR Avenue - (Route 10)', 'Amravati Expressway - (Route 7)', 'Rama Rao Street - (Route 6)', 'Sarada Colony - (Route 7)', 'Bhaskar Enclave - (Route 2)', 'Lakshmi Nagar - (Route 6)', 'Hanuman Street - (Route 1)', 'Srinivasa Lane - (Route 6)', 'Vishnu Park - (Route 8)', 'Saraswathi Nagar - (Route 3)', 'Gokulam Avenue - (Route 2)', 'Venkateswara Road - (Route 5)', 'Maruthi Lane - (Route 1)', 'Bharathi Nagar - (Route 1)', 'Subramanya Street - (Route 10)', 'Krishna Nagar - (Route 2)', 'Tulasi Vihar - (Route 6)', 'Mango Grove Lane - (Route 9)', 'Palm Tree Avenue - (Route 4)', 'Jasmine Lane - (Route 9)', 'Sunflower Street - (Route 8)', 'Rosewood Road - (Route 9)', 'Lotus Pond Avenue - (Route 8)', 'Silk Route Lane - (Route 5)', 'Silver Oak Street - (Route 2)', 'Banyan Tree Lane - (Route 1)', 'Oakwood Avenue - (Route 7)', 'Cottonwood Lane - (Route 6)', 'Marigold Street - (Route 5)', 'Hibiscus Road - (Route 10)', 'Tamarind Avenue - (Route 2)', 'Rainbow Lane - (Route 3)', 'Sandalwood Street - (Route 1)', 'Teakwood Lane - (Route 6)', 'Sycamore Avenue - (Route 2)', 'Eucalyptus Road - (Route 8)', 'Ashoka Lane - (Route 9)', 'Cherry Blossom Street - (Route 8)', 'Mango Orchard Lane - (Route 5)', 'Pine View Road - (Route 2)', 'Cedar Lane - (Route 1)', 'Maple Avenue - (Route 9)', 'Bamboo Grove Street - (Route 4)', 'Jasmine Court - (Route 3)', 'Orchid Lane - (Route 9)', 'Lily Lane - (Route 6)', 'Bougainvillea Road - (Route 1)', 'Daffodil Street - (Route 4)', 'Camellia Lane - (Route 3)', 'Ivy Lane - (Route 9)', 'Sunset Boulevard - (Route 3)', 'Ocean View Avenue - (Route 10)', 'Palm Beach Road - (Route 6)', 'Sea Breeze Lane - (Route 5)', 'Coral Cove Street - (Route 8)', 'Shell Path - (Route 7)', 'Wave Crest Avenue - (Route 6)', 'Harbor View Lane - (Route 2)', 'Lighthouse Road - (Route 4)', 'Marina Drive - (Route 7)', 'Harmony Lane - (Route 6)', 'Serene Vista Avenue - (Route 5)', 'Tranquil Terrace - (Route 4)', 'Misty Meadows Lane - (Route 10)', 'Golden Sunset Street - (Route 1)', 'Emerald Isle Road - (Route 9)', 'Crystal Cove Avenue - (Route 5)', 'Silver Sands Lane - (Route 5)', 'Pebble Beach Road - (Route 7)', 'Ocean Mist Lane - (Route 10)', 'Seashell Crescent - (Route 9)', 'Sapphire Shores Avenue - (Route 5)', 'Calm Waters Lane - (Route 10)', 'Whispering Pines Road - (Route 6)', 'Majestic Heights Avenue - (Route 6)', 'Serenity Ridge Lane - (Route 9)', 'Eternal Bliss Street - (Route 6)', 'Peaceful Haven Road - (Route 9)', 'Heavenly View Lane - (Route 10)', 'S.B.I Colony - (Route 5)', 'N.T.R Colony - (Route 5)', 'Main Road - (Route 10)'])),

'Bus_Stop_Location':pd.Series(['Bus Stop 7 - Jubilee Road', 'Bus Stop 3 - Satya Nagar', 'Bus Stop 7 - Green Valley Lane', 'Bus Stop 9 - Krishna Enclave', 'Bus Stop 8 - Golden Temple Street', 'Bus Stop 2 - Rajendra Prasad Avenue', 'Bus Stop 6 - Coastal Highway', 'Bus Stop 9 - Gopal Rao Street', 'Bus Stop 3 - Surya Nagar', 'Bus Stop 3 - Radha Krishna Lane', 'Bus Stop 4 - Vijaya Street', 'Bus Stop 2 - Lotus Blossom Road', 'Bus Stop 10 - Nandi Hills Avenue', 'Bus Stop 3 - Indira Park Lane', 'Bus Stop 8 - Murali Nagar', 'Bus Stop 8 - Gandhi Marg', 'Bus Stop 7 - Chamundi Vihar', 'Bus Stop 5 - Sai Baba Lane', 'Bus Stop 10 - NTR Avenue', 'Bus Stop 7 - Amravati Expressway', 'Bus Stop 6 - Rama Rao Street', 'Bus Stop 7 - Sarada Colony', 'Bus Stop 2 - Bhaskar Enclave', 'Bus Stop 6 -

Lakshmi Nagar', 'Bus Stop 1 - Hanuman Street', 'Bus Stop 6 - Srinivasa Lane', 'Bus Stop 8 - Vishnu Park', 'Bus Stop 3 - Saraswathi Nagar', 'Bus Stop 2 - Gokulam Avenue', 'Bus Stop 5 - Venkateswara Road', 'Bus Stop 1 - Maruthi Lane', 'Bus Stop 1 - Bharathi Nagar', 'Bus Stop 10 - Subramanya Street', 'Bus Stop 2 - Krishna Nagar', 'Bus Stop 6 - Tulasi Vihar', 'Bus Stop 9 - Mango Grove Lane', 'Bus Stop 4 - Palm Tree Avenue', 'Bus Stop 9 - Jasmine Lane', 'Bus Stop 8 - Sunflower Street', 'Bus Stop 9 - Rosewood Road', 'Bus Stop 8 - Lotus Pond Avenue', 'Bus Stop 5 - Silk Route Lane', 'Bus Stop 2 - Silver Oak Street', 'Bus Stop 1 - Banyan Tree Lane', 'Bus Stop 7 - Oakwood Avenue', 'Bus Stop 6 - Cottonwood Lane', 'Bus Stop 5 - Marigold Street', 'Bus Stop 10 - Hibiscus Road', 'Bus Stop 2 - Tamarind Avenue', 'Bus Stop 3 - Rainbow Lane', 'Bus Stop 1 - Sandalwood Street', 'Bus Stop 6 - Teakwood Lane', 'Bus Stop 2 - Sycamore Avenue', 'Bus Stop 8 - Eucalyptus Road', 'Bus Stop 9 - Ashoka Lane', 'Bus Stop 8 - Cherry Blossom Street', 'Bus Stop 5 - Mango Orchard Lane', 'Bus Stop 2 - Pine View Road', 'Bus Stop 1 - Cedar Lane', 'Bus Stop 9 - Maple Avenue', 'Bus Stop 4 - Bamboo Grove Street', 'Bus Stop 3 - Jasmine Court', 'Bus Stop 9 - Orchid Lane', 'Bus Stop 6 - Lily Lane', 'Bus Stop 1 - Bougainvillea Road', 'Bus Stop 4 - Daffodil Street', 'Bus Stop 3 - Camellia Lane', 'Bus Stop 9 - Ivy Lane', 'Bus Stop 3 - Sunset Boulevard', 'Bus Stop 10 - Ocean View Avenue', 'Bus Stop 6 - Palm Beach Road', 'Bus Stop 5 - Sea Breeze Lane', 'Bus Stop 8 - Coral Cove Street', 'Bus Stop 7 - Shell Path', 'Bus Stop 6 - Wave Crest Avenue', 'Bus Stop 2 - Harbor View Lane', 'Bus Stop 4 - Lighthouse Road', 'Bus Stop 7 - Marina Drive', 'Bus Stop 6 - Harmony Lane', 'Bus Stop 5 - Serene Vista Avenue', 'Bus Stop 4 - Tranquil Terrace', 'Bus Stop 10 - Misty Meadows Lane', 'Bus Stop 1 - Golden Sunset Street', 'Bus Stop 9 - Emerald Isle Road', 'Bus Stop 5 - Crystal Cove Avenue', 'Bus Stop 5 - Silver Sands Lane', 'Bus Stop 7 - Pebble Beach Road', 'Bus Stop 10 - Ocean Mist Lane', 'Bus Stop 9 - Seashell Crescent', 'Bus Stop 5 - Sapphire Shores Avenue', 'Bus Stop 10 - Calm Waters Lane', 'Bus Stop 6 - Whispering Pines Road', 'Bus Stop 6 - Majestic Heights Avenue', 'Bus Stop 9 - Serenity Ridge Lane', 'Bus Stop 6 - Eternal Bliss Street', 'Bus Stop 10 - Peaceful Haven Road', 'Bus Stop 5 - Heavenly View Lane', 'Bus Stop 5 - S.B.I Colony', 'Bus Stop 5 - N.T.R Colony', 'Bus Stop 10 - Main Road'])),

```
'Bus_Attendence_Record':pd.Series(['Present', 'Absent', 'Absent', 'Absent', 'Absent', 'Absent', 'Absent', 'Absent', 'Absent', 'Absent', 'Absent', 'Present', 'Present', 'Absent', 'Absent', 'Present', 'Absent', 'Present', 'Present', 'Present', 'Present', 'Absent', 'Present', 'Absent', 'Present', 'Absent', 'Present', 'Absent', 'Present', 'Present', 'Present', 'Present', 'Present', 'Absent', 'Absent', 'Present', 'Present', 'Absent', 'Absent', 'Present', 'Present', 'Absent', 'Absent', 'Present', 'Present', 'Present', 'Absent', 'Present', 'Present', 'Present', 'Absent', 'Present', 'Present', 'Absent', 'Present', 'Present', 'Absent', 'Present', 'Absent', 'Absent', 'Present', 'Present', 'Absent', 'Present', 'Present', 'Absent', 'Present', 'Present', 'Present', 'Present', 'Present', 'Present', 'Present', 'Absent', 'Absent', 'Absent', 'Absent'])),
```

'School_Lunch_Preferences': `pd.Series(['Chicken Biryani', 'Margherita Pizza', 'Veggie Wrap', 'Paneer Tikka Masala', 'Rice','Spaghetti Carbonara', 'Tandoori Chicken', 'BLT Sandwich', 'Vegetable Fried Rice','Chana Masala', 'Grilled Cheeseburger', 'Aloo Paratha', 'Chicken Caesar Salad','Vegetable Noodles', 'BBQ Pulled Pork Sandwich', 'Palak Paneer', 'Falafel Wrap','Butter Chicken', 'Beef Tacos', 'Vegetable Biryani', 'Hawaiian Pizza', 'Pesto Pasta','Veggie Burger', 'Masala`

Dosa', 'Turkey Club Sandwich', 'Veg Hakka Noodles','Pulled Chicken Sandwich', 'Vegetable Korma', 'Cheese Quesadilla', 'Pad Thai Noodles','Philly Cheesesteak Sandwich', 'Malai Kofta', 'Margherita Flatbread Pizza','Falafel Bowl', 'Spinach and Ricotta Ravioli', 'Aloo Tikki Burger','Chicken Shawarma', 'Vegetable Samosa', 'BBQ Chicken Wings', 'Pani Puri','Beef and Broccoli Stir-Fry', 'Grilled Veggie Panini', 'Mushroom Risotto','Chicken Kebabs', 'Shrimp Po Boy', 'Gobi Manchurian', 'Tandoori Lamb','Caprese Panini', 'Vegetable Thai Curry', 'Pepperoni Calzone', 'Tofu Stir-Fry','Vegetable Pakoras', 'Chicken Fajitas', 'Thai Basil Chicken', 'Buffalo Chicken Wrap','Vegetable Spring Rolls', 'Beef and Black Bean Burrito', 'Vegetable Quesadilla','Chicken Enchiladas', 'Spinach and Mushroom Quiche', 'Paneer Bhurji', 'Chicken Teriyaki','Veggie Gyro', 'Veggie Sushi Roll', 'Lemon Garlic Shrimp Pasta','Caprese Grilled Cheese', 'Mushroom Biryani', 'BBQ Pulled Jackfruit Sandwich','Chicken Pot Pie', 'Vegetable Pad See Ew', 'Spinach and Feta Stuffed Chicken Breast','Veggie Fajita Bowl', 'Chicken and Vegetable Stir-Fry', 'Tandoori Salmon','Veggie Burrito Bowl', 'Pesto Tortellini', 'Chickpea Curry', 'Buffalo Cauliflower Bites','Vegetable Bibimbap', 'Tandoori Cauliflower Steak', 'Chicken Tenders','Tofu and Vegetable Kebabs','Sushi', 'Hamburger', 'Hot Dog', 'Pasta Alfredo', 'Chicken Nuggets', 'French Fries','Fish and Chips', 'Caesar Salad', 'Grilled Cheese Sandwich', 'Tuna Salad','Veggie Omelette', 'BBQ Pulled Chicken Sandwich', 'Egg Fried Rice', 'Pepperoni Pizza','Spicy Tofu Stir-Fry', 'Chicken Burrito', 'Greek Salad', 'Shrimp Scampi'])),

'Special_Education_Status':pd.Series(['Intellectual Disability', 'Not Applicable', 'Intellectual Disability', 'Specific Learning Disability', 'Speech or Language Impairment', 'Intellectual Disability', 'Not Applicable', 'Not Applicable','Multiple Disabilities', 'Speech or Language Impairment', 'Not Applicable','Intellectual Disability', 'Intellectual Disability', 'Visual Impairment','Autism', 'Not Applicable', 'Visual Impairment', 'Other Health Impairment','Specific Learning Disability', 'Visual Impairment', 'Not Applicable','Visual Impairment', 'Emotional Disturbance', 'Not Applicable', 'Intellectual Disability', 'Hearing Impairment', 'Emotional Disturbance','Hearing Impairment', 'Not Applicable', 'Specific Learning Disability','Other Health Impairment', 'Other Health Impairment', 'Visual Impairment', 'Emotional Disturbance', 'Speech or Language Impairment', 'Hearing Impairment', 'Specific Learning Disability', 'Speech or Language Impairment', 'Autism', 'Specific Learning Disability', 'Speech or Language Impairment', 'Specific Learning Disability', 'Hearing Impairment', 'Visual Impairment','Not Applicable', 'Multiple Disabilities', 'Multiple Disabilities', 'Intellectual Disability', 'Autism', 'Autism', 'Emotional Disturbance', 'Speech or Language Impairment', 'Speech or Language Impairment','Speech or Language Impairment', 'Emotional Disturbance', 'Autism', 'Specific Learning Disability', 'Multiple Disabilities', 'Hearing Impairment','Intellectual Disability', 'Hearing Impairment', 'Visual Impairment', 'Specific Learning Disability', 'Emotional Disturbance', 'Speech or Language Impairment','Visual Impairment', 'Other Health Impairment', 'Hearing Impairment','Hearing Impairment', 'Specific Learning Disability', 'Emotional Disturbance','Not Applicable', 'Other Health Impairment', 'Not Applicable', 'Not Applicable', 'Other Health Impairment', 'Hearing Impairment', 'Intellectual Disability','Intellectual Disability', 'Autism', 'Intellectual Disability','Other Health Impairment', 'Autism', 'Autism', 'Specific Learning Disability', 'Intellectual Disability', 'Multiple Disabilities', 'Not Applicable', 'Not Applicable', 'Specific Learning Disability', 'Multiple Disabilities', 'Autism', 'Hearing Impairment','Intellectual Disability', 'Intellectual Disability', 'Other Health Impairment','Visual Impairment', 'Intellectual Disability', 'Specific Learning Disability',

'Speech or Language Impairment'])),

'IEP_Goals':pd.Series(['Not Applicable', 'Not Applicable', 'Specific Learning Disability', 'Multiple Disabilities','Emotional Disturbance', 'Multiple Disabilities', 'Autism', 'Specific Learning Disability','Visual Impairment', 'Autism', 'Specific Learning Disability', 'Specific Learning Disability','Not Applicable', 'Visual Impairment', 'Other Health Impairment', 'Multiple Disabilities','Speech or Language Impairment', 'Other Health Impairment', 'Multiple Disabilities', 'Other Health Impairment', 'Visual Impairment', 'Visual Impairment', 'Multiple Disabilities','Other Health Impairment', 'Emotional Disturbance', 'Other Health Impairment', 'Autism','Speech or Language Impairment', 'Other Health Impairment', 'Intellectual Disability','Intellectual Disability', 'Multiple Disabilities', 'Visual Impairment', 'Specific Learning Disability', 'Visual Impairment', 'Other Health Impairment', 'Other Health Impairment', 'Hearing Impairment', 'Not Applicable', 'Emotional Disturbance','Autism', 'Other Health Impairment', 'Multiple Disabilities', 'Emotional Disturbance', 'Specific Learning Disability', 'Multiple Disabilities', 'Visual Impairment', 'Speech or Language Impairment', 'Specific Learning Disability', 'Not Applicable', 'Not Applicable', 'Visual Impairment', 'Hearing Impairment', 'Other Health Impairment','Emotional Disturbance', 'Autism', 'Intellectual Disability', 'Emotional Disturbance', 'Speech or Language Impairment', 'Multiple Disabilities', 'Hearing Impairment', 'Hearing Impairment', 'Hearing Impairment', 'Multiple Disabilities', 'Other Health Impairment', 'Specific Learning Disability', 'Speech or Language Impairment', 'Hearing Impairment','Specific Learning Disability', 'Emotional Disturbance', 'Intellectual Disability', 'Multiple Disabilities', 'Emotional Disturbance', 'Other Health Impairment', 'Intellectual Disability', 'Multiple Disabilities', 'Not Applicable', 'Specific Learning Disability', 'Autism','Speech or Language Impairment', 'Speech or Language Impairment', 'Not Applicable','Other Health Impairment', 'Autism', 'Specific Learning Disability', 'Intellectual Disability', 'Other Health Impairment', 'Autism', 'Hearing Impairment', 'Autism', 'Other Health Impairment','Hearing Impairment', 'Multiple Disabilities', 'Other Health Impairment', 'Hearing Impairment', 'Not Applicable', 'Specific Learning Disability', 'Visual Impairment', 'Multiple Disabilities', 'Intellectual Disability'])),

'IEP_Progress_Notes':pd.Series(['Progress','Improvement','Excellence','Cooperation','Responsibility','Engagement','Behavior','Attitude','Respect','Creativity','Participation','Leadership','Curiosity','Adaptability','Communication','Motivation','Kindness','Empathy','Responsiveness','Resilience','Effort','Focus','Commitment','Teamwork','Efficiency','Dedication','Achievement','Adherence','Readiness','Preparation','Consistency','Comprehension','Interaction','Initiative','Involvement','Contribution','Discipline','Empowerment','Selfawareness','Independence','Consolidation','Flexibility','Conservation','Tolerance','Punctuality','Adherence','Dependability','Accountability','Participation','Sensitivity','Inclusivity','Responsiveness','Enthusiasm','Creativity','Reflection','Innovation','Empowerment','Resourcefulness','Inquisitiveness','Respectfulness','Initiation','Collaboration','Adaptiveness','Empathy','Optimism','Resilience','Persistence','Structure','Independence','Discipline','Initiative','Efficiency','Empowerment','Consolidation','Innovation','Tolerance','Adherence','Self-awareness','Inclusivity','Creativity','Conservation','Responsiveness','Participation','Accountability','Enthusiasm','Flexibility','Dependability','Empowerment','Creativity','Empathy','Reflection','Innovation','Self-awareness','Respectfulness','Inclusivity','Optimism','Initiation','Empowerment','Collaboration','Persistence']),

'ESL_Status':pd.Series(['English Proficient', 'Advanced', 'Beginner', 'Not Applicable',

'Intermediate','English Proficient', 'Not Applicable', 'English Proficient', 'Advanced', 'English Proficient','Beginner', 'Transitioning out of ESL', 'Transitioning out of ESL', 'Transitioning out of ESL','English Language Learner (ELL)', 'Fluent', 'Fluent', 'Intermediate', 'Not Applicable', 'Fluent','In ESL Program', 'English Proficient', 'Limited English Proficiency (LEP)', 'Intermediate','Intermediate', 'Advanced', 'Transitioning out of ESL', 'Not Applicable', 'Beginner', 'Advanced', 'English Language Learner (ELL)', 'Advanced', 'In ESL Program', 'Beginner', 'English Proficient','Fluent', 'English Language Learner (ELL)', 'Not Applicable', 'Advanced', 'Fluent','English Language Learner (ELL)', 'Transitioning out of ESL', 'Fluent', 'Fluent', 'Beginner', 'Limited English Proficiency (LEP)', 'English Proficient', 'English Proficient', 'Fluent', 'Beginner','English Proficient', 'Beginner', 'English Proficient', 'Transitioning out of ESL', 'English Proficient', 'Transitioning out of ESL', 'English Language Learner (ELL)', 'Fluent', 'Advanced', 'Intermediate', 'Advanced', 'Advanced', 'Limited English Proficiency (LEP)', 'Beginner','In ESL Program', 'English Language Learner (ELL)', 'Intermediate', 'Limited English Proficiency (LEP)','Fluent', 'In ESL Program', 'English Proficient', 'In ESL Program', 'Not Applicable','Not Applicable', 'English Language Learner (ELL)', 'Limited English Proficiency (LEP)', 'Fluent', 'Transitioning out of ESL', 'English Proficient', 'In ESL Program', 'Not Applicable','Fluent', 'Advanced', 'Transitioning out of ESL', 'English Proficient', 'Transitioning out of ESL','Advanced', 'English Proficient', 'Intermediate', 'Not Applicable', 'English Proficient', 'Transitioning out of ESL', 'Not Applicable', 'Fluent', 'Not Applicable', 'English Proficient','Transitioning out of ESL', 'Fluent', 'Beginner', 'Transitioning out of ESL'))

[illegible]

'Gifted_and_Talented_Status':pd.Series(['Not Gifted or Talented', 'Eligible for Gifted Program', 'In Gifted Program', 'Waiting List for Gifted Program', 'Eligible for Gifted Program', 'In Gifted Program','In Gifted Program', 'Talented', 'Eligible for Gifted Program', 'Waiting List for Gifted Program', 'In Gifted Program', 'Not Gifted or Talented', 'Talented', 'Waiting List for Gifted Program', 'Gifted', 'Gifted and Talented', 'Waiting List for Talented Program', 'Talented', 'Not Gifted or Talented', 'Not Gifted or Talented', 'Waiting List for Talented Program', 'Eligible for Talented Program','Waiting List for Gifted Program', 'Not Gifted or Talented', 'In Gifted Program', 'Not Gifted or Talented', 'Waiting List for Talented Program','Gifted', 'Eligible for Gifted Program', 'Gifted', 'Eligible for Talented Program', 'In Gifted Program', 'Gifted', 'Talented', 'Waiting List for

Talented Program', 'In Gifted Program', 'Eligible for Talented Program', 'Eligible for Gifted Program','Gifted', 'Gifted and Talented', 'Gifted and Talented', 'Gifted and Talented','Eligible for Talented Program', 'Talented', 'Waiting List for Gifted Program', 'Eligible for Gifted Program', 'In Gifted Program', 'Gifted and Talented', 'Gifted','Talented', 'Gifted and Talented', 'Waiting List for Talented Program','Waiting List for Gifted Program', 'Gifted and Talented', 'Gifted and Talented', 'Waiting List for Talented Program', 'Gifted', 'Eligible for Gifted Program', 'Waiting List for Gifted Program','Waiting List for Talented Program', 'In Gifted Program', 'Talented', 'In Gifted Program', 'Not Gifted or Talented', 'Eligible for Gifted Program', 'In Talented Program','Waiting List for Talented Program', 'Eligible for Talented Program','Eligible for Gifted Program', 'Talented', 'Waiting List for Talented Program','Eligible for Gifted Program', 'Talented', 'Gifted', 'Not Gifted or Talented','Not Gifted or Talented', 'Eligible for Gifted Program', 'Waiting List for Gifted Program','In Talented Program', 'Eligible for Talented Program', 'Eligible for Gifted Program', 'Talented','Gifted', 'Eligible for Talented Program', 'Waiting List for Gifted Program','Talented', 'Waiting List for Talented Program', 'Eligible for Gifted Program','Eligible for Gifted Program', 'Gifted', 'In Gifted Program', 'Waiting List for Gifted Program', 'Eligible for Talented Program', 'In Talented Program', 'In Gifted Program', 'Eligible for Talented Program','In Gifted Program', 'Eligible for Gifted Program', 'Gifted', 'Gifted'])),

'Counselor_Names':pd.Series(['Thomas Wilson', 'Karen Moore', 'Daniel Clark', 'Emily Davis', 'Jane Doe', 'David Lee', 'Lisa Taylor', 'Sarah Miller', 'Patricia Martinez', 'Daniel Clark', 'Christopher White', 'Patricia Martinez', 'John Smith', 'John Smith', 'Jane Doe', 'Linda Scott', 'Daniel Clark','Sarah Miller', 'Susan Brown', 'David Lee', 'Joseph Hall', 'Mary Anderson', 'Patricia Martinez','Joseph Hall', 'Thomas Wilson', 'Thomas Wilson', 'Susan Brown', 'Patricia Martinez', 'William Jones', 'Lisa Taylor', 'Robert Wilson', 'Susan Brown', 'Christopher White', 'Karen Moore','David Lee', 'David Lee', 'Joseph Hall', 'Karen Moore', 'Richard Harris', 'Sarah Miller', 'David Lee', 'Emily Davis', 'Susan Brown', 'Richard Harris', 'David Lee', 'Jennifer Garcia','Thomas Wilson', 'William Jones', 'John Smith', 'Richard Harris', 'Christopher White','Emily Davis', 'Robert Wilson', 'Lisa Taylor', 'Richard Harris', 'Patricia Martinez','Jane Doe', 'Linda Scott', 'Michael Johnson', 'Mary Anderson', 'Jennifer Garcia', 'David Lee','Karen Moore', 'Sarah Miller', 'Thomas Wilson', 'Richard Harris', 'Jane Doe', 'Jennifer Garcia', 'Robert Wilson', 'Patricia Martinez', 'Emily Davis', 'Emily Davis', 'Susan Brown', 'Susan Brown','Daniel Clark', 'Mary Anderson', 'Linda Scott', 'Richard Harris', 'John Smith', 'Daniel Clark','Mary Anderson', 'Daniel Clark', 'David Lee', 'Joseph Hall', 'Thomas Wilson', 'William Jones','Joseph Hall', 'William Jones', 'Joseph Hall', 'Mary Anderson', 'Michael Johnson','Thomas Wilson', 'Joseph Hall', 'David Lee', 'Jennifer Garcia', 'Patricia Martinez','Sarah Miller', 'David Lee', 'John Smith', 'Linda Scott'])),

'Counselor_Appointment_Date':pd.Series(['2022-08-05', '2022-09-04', '2022-07-26', '2022-11-04', '2022-03-19', '2022-10-21','2022-01-22', '2022-09-09', '2022-07-11', '2022-09-25', '2022-07-05', '2022-07-11','2022-04-04', '2022-08-11', '2022-10-01', '2022-10-29', '2022-10-30', '2022-12-02','2022-01-05', '2022-07-16', '2022-10-21', '2022-01-18', '2022-04-05', '2022-03-02','2022-05-15', '2022-05-12', '2022-05-16', '2022-02-06', '2022-03-18', '2022-12-29','2022-05-09', '2022-03-20', '2022-05-29', '2022-04-05', '2022-08-06', '2022-10-30', '2022-04-19', '2022-09-26', '2022-11-28', '2022-04-30', '2022-08-04', '2022-07-09','2022-10-28', '2022-06-22', '2022-12-29', '2022-04-07', '2022-09-19', '2022-10-04','2022-06-13', '2022-10-20', '2022-04-08', '2022-04-07', '2022-12-03', '2022-12-29','2022-11-01', '2022-04-19', '2022-03-01', '2022-08-14', '2022-02-16', '2022-02-23',

'2022-11-21', '2022-10-03', '2022-01-27', '2022-02-20', '2022-08-25', '2022-07-14','2022-04-01',
'2022-04-19', '2022-01-20', '2022-09-12', '2022-08-26', '2022-07-20','2022-09-20', '2022-09-29',
'2022-01-24', '2022-04-09', '2022-04-25', '2022-06-30','2022-03-03', '2022-04-06', '2022-02-10',
'2022-01-17', '2022-08-20', '2022-06-17','2022-04-09', '2022-09-04', '2022-11-15', '2022-11-20',
'2022-02-04', '2022-02-20', '2022-01-01', '2022-08-12', '2022-07-31', '2022-06-03', '2022-09-30',
'2022-01-04','2022-06-25', '2022-06-30', '2022-02-08', '2022-07-27'])),

**'Behavioral_incident_1_Type':pd.Series(['Disrespectful behavior', 'Bullying', 'None', 'Cheating',
'Fighting', 'Fighting','None', 'Disrespectful behavior', 'Disrespectful behavior', 'Bullying',
'Bullying','Cheating', 'Fighting', 'None', 'Fighting', 'Bullying', 'Tardiness', 'None','Disrespectful
behavior', 'Bullying', 'Fighting', 'None', 'None','Disrespectful behavior', 'Fighting', 'Fighting',
'Disrespectful behavior','Bullying', 'Bullying', 'Fighting', 'Disrespectful behavior',
'Tardiness','Tardiness', 'Bullying', 'None', 'Disrespectful behavior', 'Cheating', 'Tardiness','Fighting',
'Cheating', 'Fighting', 'Fighting', 'Disrespectful behavior', 'Fighting','Tardiness', 'Disrespectful
behavior', 'Fighting', 'None', 'Disrespectful behavior','Cheating', 'Fighting', 'Fighting', 'Cheating',
'None', 'Fighting', 'Fighting', 'None','Cheating', 'Tardiness', 'Tardiness', 'Disrespectful behavior',
'Tardiness', 'Cheating','Bullying', 'None', 'Fighting', 'Bullying', 'Fighting', 'None', 'None',
'Fighting','Fighting', 'Tardiness', 'None', 'Bullying', 'Disrespectful behavior', 'Bullying','Fighting',
'Cheating', 'Tardiness', 'Fighting', 'Disrespectful behavior', 'Tardiness','Bullying', 'None', 'Fighting',
'Bullying', 'Fighting', 'None', 'None', 'Fighting', 'Fighting','Tardiness', 'None', 'Bullying',
'Disrespectful behavior', 'Bullying', 'Cheating', 'Bullying','Fighting'])),**

**'Behavioral_incident_2_Type':pd.Series(['None', 'Cheating', 'Disrespectful behavior', 'Tardiness',
'Bullying', 'Bullying','Fighting', 'Disrespectful behavior', 'Cheating', 'Cheating', 'Bullying',
'Fighting','Fighting', 'Disrespectful behavior', 'Tardiness', 'Cheating', 'Disrespectful
behavior','Fighting', 'Fighting', 'Bullying', 'Disrespectful behavior', 'Cheating', 'None', 'Bullying',
'Bullying', 'Fighting', 'Disrespectful behavior', 'Disrespectful behavior','Disrespectful behavior',
'Bullying', 'None', 'Tardiness', 'Tardiness', 'Disrespectful behavior', 'Cheating', 'Fighting',
'Tardiness', 'Tardiness','Tardiness', 'None', 'Tardiness', 'Tardiness', 'Cheating', 'Tardiness',
'Tardiness','None', 'Cheating', 'Disrespectful behavior', 'Fighting', 'Disrespectful
behavior','Cheating', 'Fighting', 'Fighting', 'None', 'Tardiness', 'Cheating', 'Tardiness','Cheating',
'None', 'Cheating', 'Tardiness', 'Fighting', 'Tardiness', 'Tardiness','Tardiness', 'Fighting',
'Disrespectful behavior', 'Tardiness', 'Bullying', 'Cheating','Fighting', 'Disrespectful behavior',
'Fighting', 'Bullying', 'Fighting', 'Tardiness','Bullying', 'Fighting', 'Disrespectful behavior', 'None',
'Disrespectful behavior','Cheating', 'Bullying', 'Bullying', 'Tardiness', 'Tardiness', 'Disrespectful
behavior','Fighting', 'Fighting', 'Cheating', 'Bullying', 'Disrespectful behavior','Disrespectful
behavior', 'Bullying', 'Tardiness', 'Bullying', 'Fighting', 'None','Disrespectful behavior', 'Fighting'])),**

**'Behavioral_incident_3_Type':pd.Series(['Cheating', 'Bullying', 'Bullying', 'Cheating', 'None',
'Disrespectful behavior', 'Disrespectful behavior', 'Bullying', 'Disrespectful behavior', 'None',
'Cheating','Bullying', 'Disrespectful behavior', 'None', 'Cheating', 'None', 'Cheating','Cheating',
'Cheating', 'Cheating', 'Disrespectful behavior', 'Bullying', 'Bullying','Cheating', 'Cheating',
'Bullying', 'Bullying', 'None', 'Disrespectful behavior','Bullying', 'Disrespectful behavior',
'Disrespectful behavior', 'Bullying', 'Disrespectful behavior', 'None', 'Bullying', 'Disrespectful
behavior', 'Cheating', 'Cheating', 'Bullying', 'None', 'Disrespectful behavior', 'Disrespectful
behavior','Cheating', 'Disrespectful behavior', 'Bullying', 'Bullying', 'Bullying','Disrespectful**

behavior', 'Bullying', 'Bullying', 'Bullying', 'None', 'Bullying', 'Bullying', 'None', 'Bullying', 'Bullying', 'None', 'Cheating', 'Disrespectful behavior','Cheating', 'Disrespectful behavior', 'Disrespectful behavior', 'None', 'Bullying','None', 'Bullying', 'Cheating', 'Bullying', 'None', 'None', 'Bullying', 'Cheating','Bullying', 'Disrespectful behavior', 'Disrespectful behavior', 'Cheating','Cheating', 'Disrespectful behavior', 'None', 'Cheating', 'Bullying', 'Cheating','None', 'Disrespectful behavior', 'Disrespectful behavior', 'Disrespectful behavior','Disrespectful behavior', 'Disrespectful behavior', 'None', 'Bullying', 'Cheating','None', 'Disrespectful behavior', 'None', 'None', 'Disrespectful behavior', 'Cheating','None'])),

'Disciplinary_Actions_1':pd.Series(['Verbal Warning', 'Detention', 'No Action', 'Suspension', 'Expulsion', 'Expulsion', 'No Action', 'Verbal Warning', 'Verbal Warning', 'Detention', 'Detention', 'Suspension','Expulsion', 'No Action', 'Expulsion', 'Detention', 'Counseling', 'No Action', 'Verbal Warning', 'Detention', 'Expulsion', 'No Action', 'No Action', 'Verbal Warning','Expulsion', 'Expulsion', 'Verbal Warning', 'Detention', 'Detention', 'Expulsion','Verbal Warning', 'Counseling', 'Counseling', 'Detention', 'No Action', 'Verbal Warning','Suspension', 'Counseling', 'Expulsion', 'Suspension', 'Expulsion', 'Expulsion','Verbal Warning', 'Expulsion', 'Counseling', 'Verbal Warning', 'Expulsion', 'No Action','Verbal Warning', 'Suspension', 'Expulsion', 'Expulsion', 'Suspension', 'No Action','Expulsion', 'Expulsion', 'No Action', 'Suspension', 'Counseling', 'Counseling', 'Verbal Warning', 'Counseling', 'Suspension', 'Detention', 'No Action', 'Expulsion','Detention', 'Expulsion', 'No Action', 'No Action', 'Expulsion', 'Expulsion', 'Counseling','No Action', 'Detention', 'Verbal Warning', 'Detention', 'No Action', 'Suspension','Verbal Warning', 'No Action', 'Suspension', 'No Action', 'Expulsion', 'Suspension','Expulsion', 'Verbal Warning', 'Detention', 'No Action', 'Expulsion', 'Counseling', 'Counseling', 'Verbal Warning', 'Expulsion', 'Counseling', 'Verbal Warning', 'Expulsion','Suspension', 'No Action', 'Verbal Warning'])),

'Disciplinary_Actions_2':pd.Series(['No Action', 'Suspension', 'Verbal Warning', 'Counseling', 'Detention', 'Detention','Expulsion', 'Verbal Warning', 'Suspension', 'Suspension', 'Detention', 'Expulsion','Expulsion', 'Verbal Warning', 'Counseling', 'Suspension', 'Verbal Warning', 'Expulsion','Expulsion', 'Detention', 'Verbal Warning', 'Suspension', 'No Action', 'Detention', 'Detention', 'Expulsion', 'Verbal Warning', 'Verbal Warning', 'Verbal Warning', 'Detention','No Action', 'Counseling', 'Counseling', 'Verbal Warning', 'Suspension', 'Expulsion','Counseling', 'Counseling', 'Counseling', 'Counseling', 'No Action', 'Counseling', 'Counseling','Suspension', 'Counseling', 'Counseling', 'No Action', 'Suspension', 'Verbal Warning','Expulsion', 'Verbal Warning', 'Suspension', 'Expulsion', 'Expulsion', 'No Action','Counseling', 'Suspension', 'Counseling', 'Suspension', 'No Action', 'Suspension','Counseling', 'Expulsion', 'Counseling', 'Counseling', 'Counseling', 'Expulsion','Verbal Warning', 'Counseling', 'Detention', 'Suspension', 'Expulsion', 'Verbal Warning','Expulsion', 'Detention', 'Expulsion', 'Counseling', 'Detention', 'Expulsion','Verbal Warning', 'No Action', 'Verbal Warning', 'Suspension', 'Detention', 'Detention','Counseling', 'Counseling', 'Verbal Warning', 'Expulsion', 'Expulsion', 'Suspension','Detention', 'Verbal Warning', 'Verbal Warning', 'Detention', 'Counseling', 'Detention','Expulsion', 'No Action', 'Verbal Warning', 'Expulsion'])),

'Disciplinary_Actions_3':pd.Series(['Suspension', 'Detention', 'Detention', 'Suspension', 'No Action', 'Verbal Warning','Verbal Warning', 'Detention', 'Verbal Warning', 'No Action', 'Suspension', 'Detention','Verbal Warning', 'No Action', 'Suspension', 'No Action', 'Suspension',

'Suspension','Suspension', 'Suspension', 'Verbal Warning', 'Detention', 'Detention',
'Suspension','Suspension', 'Detention', 'Detention', 'No Action', 'Verbal Warning',
'Detention','Verbal Warning', 'Verbal Warning', 'Detention', 'Verbal Warning', 'No
Action','Detention', 'Verbal Warning', 'Suspension', 'Suspension', 'Detention', 'No Action','Verbal
Warning', 'Verbal Warning', 'Suspension', 'Verbal Warning', 'Detention','Detention', 'Detention',
'Verbal Warning', 'Detention', 'Detention', 'Detention','No Action', 'Detention', 'Detention', 'No
Action', 'Detention', 'Detention', 'No Action','Suspension', 'Verbal Warning', 'Suspension', 'Verbal
Warning', 'Verbal Warning','No Action', 'Detention', 'No Action', 'Detention', 'Suspension',
'Detention','No Action', 'No Action', 'Detention', 'Suspension', 'Detention', 'Verbal Warning','Verbal
Warning', 'Suspension', 'Suspension', 'Verbal Warning', 'No Action', 'Suspension','Detention',
'Suspension', 'No Action', 'Verbal Warning', 'Verbal Warning','Verbal Warning', 'Verbal Warning',
'Verbal Warning', 'No Action', 'Detention','Suspension', 'No Action', 'Verbal Warning', 'No Action',
'No Action', 'Verbal Warning','Suspension', 'No Action']],

**'School_Health_Nurse_Name': pd.Series(['Frank Martinez', 'Meera Verma', 'Akshay Mehta',
'Frank Davis', 'Alice Brown','Rahul Patel', 'Varun Mehta', 'Charlie Lee', 'Charlie Smith', 'Hannah
Smith','David Davis', 'David Johnson', 'Zoya Chopra', 'Kiran Sharma', 'Ian Anderson','David
Anderson', 'Aishwarya Verma', 'Aishwarya Joshi', 'Divya Reddy', 'Ian Garcia','Varun Singh', 'David
Davis', 'Kiran Patel', 'David Brown', 'Kiran Reddy','Aishwarya Sharma', 'Bob Brown', 'Frank
Anderson', 'Meera Gupta', 'Akshay Mehta','Zoya Sharma', 'Rahul Singh', 'Jennifer Garcia', 'Sanya
Sharma', 'Jennifer Lee','Sanya Singh', 'Alice Lee', 'Alice Thomas', 'Zoya Kumar', 'Eva Wilson', 'Bob
Johnson','Aarav Joshi', 'Bob Thomas', 'Jennifer Smith', 'Jennifer Johnson', 'Kiran Verma','Jennifer
Martinez', 'David Brown', 'Divya Sharma', 'Akshay Sharma', 'Alice Davis','Bob Johnson', 'Rahul
Sharma', 'Akshay Mehta', 'Hannah Smith', 'Alice Smith', 'Rahul Patel', 'Frank Wilson', 'Alice
Anderson', 'Aarav Verma', 'Frank Lee', 'Frank Thomas', 'Alice Davis', 'Aarav Mehta', 'Eva Johnson',
'Akshay Joshi', 'Aarav Joshi', 'Aarav Kumar', 'Alice Lee', 'Ian Thomas', 'Kiran Verma', 'Bob
Wilson','Rahul Verma', 'Kiran Reddy', 'Alice Anderson', 'Eva Thomas', 'Bob Martinez','Rahul
Singh', 'David Johnson', 'Bob Johnson', 'Varun Gupta', 'Aarav Chopra','Aishwarya Chopra', 'Zoya
Joshi', 'Jennifer Garcia', 'Bob Lee', 'Rahul Chopra','Rahul Chopra', 'Varun Reddy', 'Divya Reddy',
'Varun Verma', 'Hannah Garcia','Zoya Patel', 'Eva Garcia', 'Eva Garcia', 'Divya Joshi', 'Aishwarya
Sharma','Aarav Verma', 'Kiran Singh', 'Grace Davis']]),**

**'School_Health_Checkup_Date':pd.Series(['2023-01-01', '2023-01-15', '2023-02-05', '2023-02-
20', '2023-03-10', '2023-03-25','2023-04-05', '2023-04-20', '2023-05-02', '2023-05-18','2023-06-01',
'2023-06-15','2023-07-05', '2023-07-20', '2023-08-03', '2023-08-18', '2023-09-01', '2023-09-
15','2023-10-05', '2023-10-20','2023-11-01', '2023-11-15', '2023-12-05', '2023-12-20','2024-01-01',
'2024-01-15', '2024-02-05', '2024-02-20', '2024-03-10', '2024-03-25','2024-04-05', '2024-04-20',
'2024-05-02', '2024-05-18', '2024-06-01', '2024-06-15','2024-07-05', '2024-07-20', '2024-08-03',
'2024-08-18','2024-09-01', '2024-09-15','2024-10-05', '2024-10-20', '2024-11-01', '2024-11-15',
'2024-12-05', '2024-12-20','2025-01-01', '2025-01-15','2025-02-05', '2025-02-20', '2025-03-10',
'2025-03-25','2025-04-05', '2025-04-20', '2025-05-02', '2025-05-18', '2025-06-01', '2025-06-
15','2025-07-05', '2025-07-20', '2025-08-03', '2025-08-18', '2025-09-01', '2025-09-15','2025-10-05',
'2025-10-20', '2025-11-01', '2025-11-15','2025-12-05', '2025-12-20', '2026-01-01', '2026-01-15',
'2026-02-05', '2026-02-20', '2026-03-10', '2026-03-25','2026-04-05', '2026-04-20','2026-05-02',
'2026-05-18', '2026-06-01', '2026-06-15','2026-07-05', '2026-07-20', '2026-08-03', '2026-08-18',**

'Rejected','Applied', 'Pending', 'Approved', 'Rejected', 'Applied', 'Pending', 'Approved', 'Rejected',
 'Applied', 'Pending','Approved', 'Rejected', 'Applied', 'Pending', 'Approved', 'Rejected', 'Applied',
 'Pending', 'Approved', 'Rejected','Applied', 'Pending', 'Approved', 'Rejected', 'Applied', 'Pending',
 'Approved', 'Rejected', 'Applied', 'Pending','Approved', 'Rejected', 'Applied', 'Pending', 'Approved',
 'Rejected', 'Applied', 'Pending', 'Approved', 'Rejected','Applied', 'Pending', 'Approved', 'Rejected',
 'Applied', 'Pending', 'Approved', 'Rejected', 'Applied', 'Pending','Approved', 'Rejected', 'Applied',
 'Pending', 'Approved', 'Rejected', 'Applied', 'Pending', 'Approved', 'Rejected']),

'College_Acceptance_Status': pd.Series(['Denied', 'Accepted', 'Accepted', 'Denied', 'Accepted', 'Denied', 'Accepted', 'Accepted', 'Denied', 'Accepted', 'Accepted', 'Denied', 'Accepted', 'Denied', 'Denied', 'Accepted', 'Accepted', 'Denied','Accepted', 'Denied', 'Denied', 'Accepted', 'Accepted', 'Accepted', 'Denied', 'Denied','Denied', 'Accepted', 'Accepted', 'Denied', 'Accepted', 'Denied','Accepted', 'Denied', 'Accepted', 'Accepted', 'Denied', 'Accepted', 'Denied','Accepted', 'Denied', 'Accepted', 'Accepted', 'Accepted', 'Denied', 'Denied', 'Accepted', 'Accepted','Accepted', 'Denied', 'Accepted', 'Denied', 'Accepted', 'Denied', 'Denied', 'Accepted', 'Accepted', 'Denied', 'Accepted', 'Denied', 'Denied', 'Accepted', 'Accepted', 'Accepted', 'Denied', 'Accepted', 'Accepted', 'Denied', 'Accepted', 'Accepted', 'Denied', 'Accepted', 'Accepted', 'Denied', 'Accepted', 'Accepted', 'Denied', 'Accepted', 'Accepted', 'Denied'])

**'College_Attendance_Status': pd.Series(['Present', 'Absent', 'Present', 'Present', 'Absent', 'Present',
'Absent', 'Present', 'Absent', 'Absent','Present', 'Absent', 'Absent', 'Present', 'Absent', 'Present',
Present', 'Absent', 'Absent', 'Present','Absent', 'Present', 'Absent', 'Present', 'Present', 'Absent',
'Present', 'Absent', 'Absent', 'Present','Absent', 'Absent', 'Present', 'Present', 'Present', 'Absent',
'Absent', 'Present', 'Present', 'Absent','Absent', 'Present', 'Absent', 'Absent', 'Absent', 'Present',
'Absent', 'Absent', 'Absent', 'Absent','Present', 'Absent', 'Present', 'Present', 'Present', 'Absent',
'Absent', 'Absent', 'Absent', 'Present','Absent', 'Absent', 'Present', 'Present', 'Absent', 'Present',
'Absent', 'Present', 'Present', 'Absent','Absent', 'Absent', 'Present', 'Absent', 'Present', 'Present',
'Absent', 'Absent', 'Absent', 'Present','Absent', 'Absent', 'Present', 'Present', 'Absent', 'Absent',
'Present', 'Present', 'Present', 'Absent','Absent', 'Absent', 'Absent', 'Present', 'Absent', 'Present',
'Absent', 'Absent', 'Present', 'Absent'])),**

'Career_Aspiration': pd.Series(['Engineer', 'Doctor', 'Artist', 'Teacher', 'Lawyer', 'Scientist', 'Chef', 'Musician', 'Architect', 'Writer', 'Lawyer', 'Teacher', 'Artist', 'Scientist', 'Architect', 'Musician', 'Engineer', 'Chef', 'Writer', 'Doctor', 'Artist', 'Writer', 'Engineer', 'Teacher', 'Musician', 'Lawyer', 'Doctor', 'Scientist', 'Chef', 'Architect', 'Musician', 'Engineer', 'Teacher', 'Lawyer', 'Artist', 'Scientist', 'Architect', 'Writer', 'Chef', 'Doctor', 'Doctor', 'Lawyer', 'Teacher', 'Musician', 'Scientist', 'Chef', 'Engineer', 'Writer', 'Architect', 'Artist', 'Architect', 'Scientist', 'Chef', 'Teacher', 'Lawyer', 'Engineer', 'Musician', 'Doctor', 'Artist', 'Writer', 'Engineer', 'Writer', 'Artist', 'Scientist', 'Lawyer', 'Teacher', 'Chef', 'Doctor', 'Musician', 'Architect', 'Musician', 'Teacher', 'Architect', 'Doctor', 'Scientist', 'Chef', 'Writer', 'Lawyer', 'Engineer', 'Artist', 'Artist', 'Scientist', 'Musician', 'Lawyer', 'Teacher', 'Doctor', 'Writer', 'Architect', 'Chef', 'Engineer', 'Chef', 'Artist', 'Musician', 'Scientist', 'Doctor', 'Architect', 'Writer', 'Engineer', 'Teacher', 'Lawyer'])

'Internship_1_Opportunity': pd.Series(['Paid', 'Unpaid', 'Remote', 'Marketing', 'Technology', 'Research', 'Engineering', 'Design', 'Healthcare', 'Finance', 'Finance', 'Paid', 'Design', 'Unpaid',

'Marketing', 'Remote', 'Healthcare', 'Engineering', 'Technology', 'Research', 'Research', 'Technology', 'Marketing', 'Engineering', 'Finance', 'Unpaid', 'Paid', 'Remote', 'Design', 'Healthcare', 'Healthcare', 'Engineering', 'Marketing', 'Design', 'Paid', 'Unpaid', 'Technology', 'Finance', 'Research', 'Remote', 'Remote', 'Technology', 'Design', 'Paid', 'Finance', 'Marketing', 'Unpaid', 'Research', 'Engineering', 'Healthcare', 'Healthcare', 'Marketing', 'Remote', 'Finance', 'Design', 'Paid', 'Engineering', 'Unpaid', 'Technology', 'Research', 'Research', 'Paid', 'Unpaid', 'Design', 'Marketing', 'Finance', 'Remote', 'Technology', 'Engineering', 'Healthcare', 'Healthcare', 'Technology', 'Unpaid', 'Finance', 'Marketing', 'Paid', 'Remote', 'Engineering', 'Design', 'Research', 'Research', 'Healthcare', 'Design', 'Finance', 'Marketing', 'Remote', 'Unpaid', 'Technology', 'Paid', 'Engineering', 'Engineering', 'Finance', 'Marketing', 'Technology', 'Remote', 'Paid', 'Unpaid', 'Healthcare', 'Design', 'Research']),

'Internship_2_Opportunity': pd.Series(['Marketing', 'Technology', 'Finance', 'Engineering', 'Healthcare', 'Remote', 'Unpaid', 'Paid', 'Design', 'Research', 'Unpaid', 'Finance', 'Paid', 'Technology', 'Healthcare', 'Design', 'Marketing', 'Remote', 'Research', 'Engineering', 'Engineering', 'Unpaid', 'Finance', 'Healthcare', 'Remote', 'Paid', 'Technology', 'Research', 'Marketing', 'Design', 'Design', 'Marketing', 'Finance', 'Remote', 'Healthcare', 'Technology', 'Research', 'Unpaid', 'Engineering', 'Paid', 'Remote', 'Technology', 'Engineering', 'Healthcare', 'Finance', 'Paid', 'Unpaid', 'Marketing', 'Design', 'Research', 'Marketing', 'Design', 'Technology', 'Research', 'Unpaid', 'Finance', 'Paid', 'Engineering', 'Healthcare', 'Remote', 'Unpaid', 'Finance', 'Remote', 'Paid', 'Technology', 'Design', 'Marketing', 'Healthcare', 'Research', 'Engineering', 'Healthcare', 'Remote', 'Design', 'Research', 'Paid', 'Unpaid', 'Technology', 'Marketing', 'Finance', 'Engineering', 'Finance', 'Marketing', 'Research', 'Remote', 'Design', 'Unpaid', 'Paid', 'Healthcare', 'Technology', 'Engineering', 'Engineering', 'Unpaid', 'Finance', 'Paid', 'Healthcare', 'Marketing', 'Remote', 'Design', 'Technology', 'Research']),

'Student_Club_Membership_1': pd.Series(['Debate Club', 'Photography Club', 'Chess Club', 'Drama Club', 'Science Club', 'Art Club', 'Music Club', 'Coding Club', 'Sports Club', 'Environmental Club', 'Coding Club', 'Science Club', 'Art Club', 'Music Club', 'Sports Club', 'Drama Club', 'Photography Club', 'Environmental Club', 'Debate Club', 'Chess Club', 'Sports Club', 'Drama Club', 'Science Club', 'Coding Club', 'Photography Club', 'Music Club', 'Environmental Club', 'Art Club', 'Debate Club', 'Chess Club', 'Music Club', 'Debate Club', 'Science Club', 'Art Club', 'Chess Club', 'Environmental Club', 'Photography Club', 'Coding Club', 'Drama Club', 'Sports Club', 'Science Club', 'Art Club', 'Drama Club', 'Music Club', 'Sports Club', 'Photography Club', 'Chess Club', 'Debate Club', 'Environmental Club', 'Coding Club', 'Environmental Club', 'Debate Club', 'Art Club', 'Photography Club', 'Chess Club', 'Music Club', 'Drama Club', 'Science Club', 'Sports Club', 'Coding Club', 'Art Club', 'Drama Club', 'Photography Club', 'Environmental Club', 'Coding Club', 'Debate Club', 'Chess Club', 'Music Club', 'Science Club', 'Sports Club', 'Photography Club', 'Science Club', 'Coding Club', 'Chess Club', 'Environmental Club', 'Drama Club', 'Debate Club', 'Music Club', 'Art Club', 'Sports Club', 'Art Club', 'Music Club', 'Photography Club', 'Science Club', 'Drama Club', 'Debate Club', 'Coding Club', 'Environmental Club', 'Sports Club', 'Chess Club', 'Drama Club', 'Science Club', 'Music Club', 'Art Club', 'Coding Club', 'Photography Club', 'Environmental Club', 'Chess Club', 'Debate Club', 'Sports Club']),

'Student_Club_Membership_2': pd.Series(['Sports Club', 'Coding Club', 'Art Club', 'Photography Club', 'Science Club', 'Drama Club', 'Debate Club', 'Environmental Club', 'Chess Club', 'Music Club', 'Photography Club', 'Art Club', 'Coding Club', 'Science Club', 'Drama Club',

'Sports Club', 'Debate Club', 'Chess Club', 'Environmental Club', 'Music Club', 'Drama Club', 'Debate Club', 'Art Club', 'Chess Club', 'Coding Club', 'Photography Club', 'Science Club', 'Environmental Club', 'Music Club', 'Sports Club', 'Environmental Club', 'Drama Club', 'Coding Club', 'Music Club', 'Photography Club', 'Science Club', 'Art Club', 'Debate Club', 'Chess Club', 'Sports Club', 'Art Club', 'Coding Club', 'Music Club', 'Drama Club', 'Environmental Club', 'Science Club', 'Chess Club', 'Photography Club', 'Debate Club', 'Sports Club', 'Science Club', 'Environmental Club', 'Art Club', 'Photography Club', 'Drama Club', 'Music Club', 'Coding Club', 'Debate Club', 'Chess Club', 'Sports Club', 'Photography Club', 'Coding Club', 'Art Club', 'Environmental Club', 'Drama Club', 'Debate Club', 'Science Club', 'Music Club', 'Chess Club', 'Sports Club', 'Drama Club', 'Science Club', 'Coding Club', 'Photography Club', 'Chess Club', 'Environmental Club', 'Debate Club', 'Art Club', 'Music Club', 'Sports Club', 'Art Club', 'Environmental Club', 'Science Club', 'Drama Club', 'Music Club', 'Coding Club', 'Photography Club', 'Chess Club', 'Debate Club', 'Sports Club', 'Music Club', 'Environmental Club', 'Chess Club', 'Science Club', 'Drama Club', 'Sports Club', 'Photography Club', 'Coding Club', 'Art Club', 'Debate Club'])),

'Student_Club_Membership_3': pd.Series(['Chess Club', 'Drama Club', 'Art Club', 'Science Club', 'Music Club', 'Coding Club', 'Photography Club', 'Environmental Club', 'Debate Club', 'Sports Club', 'Sports Club', 'Coding Club', 'Art Club', 'Photography Club', 'Science Club', 'Drama Club', 'Debate Club', 'Environmental Club', 'Chess Club', 'Music Club', 'Photography Club', 'Art Club', 'Coding Club', 'Science Club', 'Drama Club', 'Sports Club', 'Debate Club', 'Chess Club', 'Environmental Club', 'Music Club', 'Drama Club', 'Debate Club', 'Art Club', 'Chess Club', 'Coding Club', 'Photography Club', 'Science Club', 'Environmental Club', 'Music Club', 'Sports Club', 'Environmental Club', 'Drama Club', 'Coding Club', 'Music Club', 'Photography Club', 'Science Club', 'Art Club', 'Debate Club', 'Chess Club', 'Sports Club', 'Art Club', 'Coding Club', 'Music Club', 'Drama Club', 'Environmental Club', 'Science Club', 'Chess Club', 'Photography Club', 'Debate Club', 'Sports Club', 'Science Club', 'Environmental Club', 'Art Club', 'Photography Club', 'Drama Club', 'Music Club', 'Coding Club', 'Debate Club', 'Chess Club', 'Sports Club', 'Photography Club', 'Coding Club', 'Art Club', 'Environmental Club', 'Drama Club', 'Debate Club', 'Science Club', 'Music Club', 'Chess Club', 'Sports Club', 'Debate Club', 'Photography Club', 'Chess Club', 'Drama Club', 'Science Club', 'Art Club', 'Music Club', 'Coding Club', 'Sports Club', 'Environmental Club', 'Chess Club', 'Drama Club', 'Art Club', 'Science Club', 'Music Club', 'Coding Club', 'Photography Club', 'Environmental Club', 'Debate Club', 'Sports Club'])),

'Educational_Achievements': pd.Series(['Scholarship', 'Dean's List', 'Honor Roll', 'Research Award', 'First Place Science Fair', 'Outstanding Thesis', 'Math Competition Winner', 'Student of the Year', 'Perfect Attendance', 'Community Service Award', 'Student of the Year', 'Perfect Attendance', 'Community Service Award', 'Dean's List', 'Honor Roll', 'Scholarship', 'Research Award', 'First Place Science Fair', 'Outstanding Thesis', 'Math Competition Winner', 'Outstanding Thesis', 'Math Competition Winner', 'Community Service Award', 'Dean's List', 'Honor Roll', 'Scholarship', 'Student of the Year', 'Perfect Attendance', 'Research Award', 'First Place Science Fair', 'Perfect Attendance', 'Dean's List', 'Math Competition Winner', 'Outstanding Thesis', 'Honor Roll', 'Scholarship', 'First Place Science Fair', 'Research Award', 'Student of the Year', 'Community Service Award', 'First Place Science Fair', 'Outstanding Thesis', 'Student of the Year', 'Math Competition Winner', 'Research Award', 'Scholarship', 'Dean's List', 'Community Service Award', 'Honor Roll', 'Perfect Attendance', 'Outstanding Thesis', 'Scholarship', 'Honor Roll', 'Math

Competition Winner', 'Community Service Award', 'Dean's List', 'Student of the Year', 'First Place Science Fair', 'Perfect Attendance', 'Research Award', 'Student of the Year', 'Scholarship', 'Perfect Attendance', 'Outstanding Thesis', 'Honor Roll', 'First Place Science Fair', 'Math Competition Winner', 'Community Service Award', 'Dean's List', 'Research Award', 'Honor Roll', 'First Place Science Fair', 'Scholarship', 'Perfect Attendance', 'Math Competition Winner', 'Outstanding Thesis', 'Dean's List', 'Community Service Award', 'Student of the Year', 'Research Award', 'Perfect Attendance', 'Outstanding Thesis', 'Dean's List', 'Math Competition Winner', 'First Place Science Fair', 'Honor Roll', 'Scholarship', 'Research Award', 'Community Service Award', 'Student of the Year', 'First Place Science Fair', 'Outstanding Thesis', 'Community Service Award', 'Scholarship', 'Dean's List', 'Honor Roll', 'Math Competition Winner', 'Perfect Attendance', 'Research Award', 'Student of the Year'])),

'Educational_Challenges': pd.Series(['Limited Resources', 'Language Barrier', 'Time Management', 'Exam Pressure', 'Financial Strain', 'Procrastination', 'Lack of Motivation', 'Peer Pressure', 'Learning Disabilities', 'Online Learning', 'Learning Disabilities', 'Time Management', 'Procrastination', 'Lack of Motivation', 'Peer Pressure', 'Limited Resources', 'Financial Strain', 'Language Barrier', 'Exam Pressure', 'Online Learning', 'Exam Pressure', 'Peer Pressure', 'Language Barrier', 'Limited Resources', 'Learning Disabilities', 'Time Management', 'Financial Strain', 'Online Learning', 'Procrastination', 'Lack of Motivation', 'Language Barrier', 'Lack of Motivation', 'Procrastination', 'Peer Pressure', 'Limited Resources', 'Online Learning', 'Exam Pressure', 'Time Management', 'Financial Strain', 'Learning Disabilities', 'Financial Strain', 'Online Learning', 'Limited Resources', 'Procrastination', 'Learning Disabilities', 'Language Barrier', 'Exam Pressure', 'Peer Pressure', 'Lack of Motivation', 'Time Management', 'Time Management', 'Exam Pressure', 'Peer Pressure', 'Financial Strain', 'Procrastination', 'Lack of Motivation', 'Language Barrier', 'Online Learning', 'Learning Disabilities', 'Limited Resources', 'Lack of Motivation', 'Procrastination', 'Online Learning', 'Limited Resources', 'Peer Pressure', 'Exam Pressure', 'Language Barrier', 'Financial Strain', 'Time Management', 'Learning Disabilities', 'Exam Pressure', 'Time Management', 'Language Barrier', 'Financial Strain', 'Procrastination', 'Limited Resources', 'Learning Disabilities', 'Lack of Motivation', 'Peer Pressure', 'Online Learning', 'Procrastination', 'Lack of Motivation', 'Peer Pressure', 'Limited Resources', 'Online Learning', 'Exam Pressure', 'Financial Strain', 'Learning Disabilities', 'Time Management', 'Language Barrier', 'Language Barrier', 'Limited Resources', 'Peer Pressure', 'Procrastination', 'Financial Strain', 'Learning Disabilities', 'Time Management', 'Online Learning', 'Exam Pressure', 'Lack of Motivation'])),

**'Learning_Disability_Status': pd.Series(['Not Diagnosed', 'Diagnosed', 'Not Diagnosed', 'Not
Diagnosed', 'Diagnosed', 'Not Diagnosed', 'Diagnosed', 'Not Diagnosed', 'Diagnosed', 'Not
Diagnosed','Diagnosed', 'Not Diagnosed', 'Diagnosed', 'Not Diagnosed', 'Not Diagnosed',
'Diagnosed', 'Not Diagnosed', 'Not Diagnosed', 'Diagnosed', 'Not Diagnosed','Diagnosed', 'Not
Diagnosed', 'Not Diagnosed', 'Not Diagnosed', 'Diagnosed', 'Not Diagnosed', 'Not Diagnosed',
'Diagnosed', 'Diagnosed', 'Not Diagnosed','Not Diagnosed', 'Diagnosed', 'Not Diagnosed', 'Not
Diagnosed', 'Diagnosed', 'Diagnosed', 'Not Diagnosed', 'Not Diagnosed', 'Diagnosed', 'Not
Diagnosed','Diagnosed', 'Diagnosed', 'Not Diagnosed', 'Not Diagnosed', 'Not Diagnosed',
'Diagnosed', 'Not Diagnosed', 'Not Diagnosed', 'Not Diagnosed', 'Diagnosed','Not Diagnosed', 'Not
Diagnosed', 'Not Diagnosed', 'Diagnosed', 'Diagnosed', 'Not Diagnosed', 'Not Diagnosed', 'Not
Diagnosed', 'Not Diagnosed', 'Not Diagnosed','Diagnosed', 'Not Diagnosed', 'Diagnosed', 'Not**

Diagnosed', 'Not Diagnosed', 'Not Diagnosed', 'Diagnosed', 'Not Diagnosed', 'Not Diagnosed', 'Diagnosed', 'Diagnosed', 'Not Diagnosed', 'Not Diagnosed', 'Not Diagnosed', 'Not Diagnosed', 'Diagnosed', 'Not Diagnosed', 'Not Diagnosed', 'Diagnosed', 'Diagnosed', 'Not Diagnosed', 'Not Diagnosed', 'Not Diagnosed', 'Diagnosed', 'Not Diagnosed', 'Not Diagnosed', 'Diagnosed', 'Not Diagnosed', 'Not Diagnosed', 'Diagnosed', 'Not Diagnosed', 'Diagnosed', 'Not Diagnosed', 'Diagnosed', 'Not Diagnosed', 'Diagnosed']),

'Educational_Resources_Used': **pd.Series**(['Online Courses', 'Textbooks', 'Tutoring Services', 'Educational Apps', 'Library Resources', 'Peer Study Groups', 'Video Lectures', 'Interactive Simulations', 'Workshops', 'Online Forums', 'Textbooks', 'Educational Apps', 'Library Resources', 'Peer Study Groups', 'Interactive Simulations', 'Online Forums', 'Workshops', 'Video Lectures', 'Tutoring Services', 'Online Courses', 'Library Resources', 'Video Lectures', 'Educational Apps', 'Online Courses', 'Tutoring Services', 'Peer Study Groups', 'Interactive Simulations', 'Workshops', 'Textbooks', 'Online Forums', 'Peer Study Groups', 'Workshops', 'Tutoring Services', 'Library Resources', 'Video Lectures', 'Educational Apps', 'Online Courses', 'Interactive Simulations', 'Online Forums', 'Textbooks', 'Interactive Simulations', 'Online Courses', 'Tutoring Services', 'Educational Apps', 'Workshops', 'Video Lectures', 'Library Resources', 'Peer Study Groups', 'Textbooks', 'Online Forums', 'Online Forums', 'Textbooks', 'Peer Study Groups', 'Library Resources', 'Online Courses', 'Video Lectures', 'Workshops', 'Educational Apps', 'Tutoring Services', 'Interactive Simulations', 'Video Lectures', 'Interactive Simulations', 'Educational Apps', 'Online Forums', 'Tutoring Services', 'Peer Study Groups', 'Library Resources', 'Workshops', 'Textbooks', 'Online Courses', 'Workshops', 'Tutoring Services', 'Library Resources', 'Textbooks', 'Online Courses', 'Peer Study Groups', 'Video Lectures', 'Educational Apps', 'Interactive Simulations', 'Online Forums', 'Interactive Simulations', 'Workshops', 'Educational Apps', 'Online Courses', 'Video Lectures', 'Library Resources', 'Tutoring Services', 'Peer Study Groups', 'Online Forums', 'Textbooks', 'Online Courses', 'Educational Apps', 'Tutoring Services', 'Workshops', 'Video Lectures', 'Interactive Simulations', 'Peer Study Groups', 'Library Resources', 'Online Forums', 'Textbooks']),

'Teacher_Training_Records': **pd.Series**(['Certified', 'Professional Development', 'Masters Degree', 'Teaching Certificate', 'Specialized Workshops', 'Advanced Training', 'Education Courses', 'Ph.D. in Education', 'Pedagogy Seminars', 'Teacher Mentorship', 'Specialized Workshops', 'Masters Degree', 'Teacher Mentorship', 'Certified', 'Professional Development', 'Advanced Training', 'Education Courses', 'Ph.D. in Education', 'Pedagogy Seminars', 'Teaching Certificate', 'Advanced Training', 'Education Courses', 'Certified', 'Pedagogy Seminars', 'Masters Degree', 'Ph.D. in Education', 'Teacher Mentorship', 'Specialized Workshops', 'Professional Development', 'Teaching Certificate', 'Certified', 'Professional Development', 'Masters Degree', 'Teaching Certificate', 'Specialized Workshops', 'Advanced Training', 'Education Courses', 'Ph.D. in Education', 'Pedagogy Seminars', 'Teacher Mentorship', 'Certified', 'Masters Degree', 'Teacher Mentorship', 'Certified', 'Professional Development', 'Advanced Training', 'Education Courses', 'Ph.D. in Education', 'Pedagogy Seminars', 'Teaching Certificate', 'Specialized Workshops', 'Ph.D. in Education', 'Pedagogy Seminars', 'Teaching Certificate', 'Certified', 'Professional Development', 'Masters Degree', 'Specialized Workshops', 'Advanced Training', 'Education Courses', 'Teacher Mentorship', 'Certified', 'Education Courses', 'Advanced Training', 'Specialized Workshops', 'Pedagogy

'Community Gardening', 'Environmental Cleanup', 'Animal Shelter Assistance', 'Fundraising Events', 'Homeless Shelter Aid', 'Disaster Relief', 'Senior Center Visits', 'Food Bank Support', 'Community Gardening', 'Homeless Shelter Aid', 'Senior Center Visits', 'Food Bank Support', 'Disaster Relief', 'Youth Mentorship', 'Tutoring', 'Fundraising Events', 'Animal Shelter Assistance', 'Environmental Cleanup', 'Environmental Cleanup', 'Community Gardening', 'Disaster Relief', 'Senior Center Visits', 'Fundraising Events', 'Tutoring', 'Food Bank Support', 'Animal Shelter Assistance', 'Youth Mentorship', 'Homeless Shelter Aid', 'Animal Shelter Assistance', 'Food Bank Support', 'Homeless Shelter Aid', 'Tutoring', 'Youth Mentorship', 'Disaster Relief', 'Senior Center Visits', 'Environmental Cleanup', 'Community Gardening', 'Fundraising Events', 'Food Bank Support', 'Environmental Cleanup', 'Tutoring', 'Homeless Shelter Aid', 'Community Gardening', 'Youth Mentorship', 'Disaster Relief', 'Animal Shelter Assistance', 'Senior Center Visits', 'Fundraising Events', 'Youth Mentorship', 'Disaster Relief', 'Senior Center Visits', 'Homeless Shelter Aid', 'Community Gardening', 'Tutoring', 'Fundraising Events', 'Animal Shelter Assistance', 'Environmental Cleanup', 'Food Bank Support', 'Fundraising Events', 'Food Bank Support', 'Senior Center Visits', 'Animal Shelter Assistance', 'Youth Mentorship', 'Disaster Relief', 'Tutoring', 'Homeless Shelter Aid', 'Community Gardening', 'Environmental Cleanup', 'Homeless Shelter Aid', 'Animal Shelter Assistance', 'Environmental Cleanup', 'Senior Center Visits', 'Disaster Relief', 'Tutoring', 'Youth Mentorship', 'Food Bank Support', 'Community Gardening', 'Fundraising Events', 'Community Gardening', 'Animal Shelter Assistance', 'Disaster Relief', 'Environmental Cleanup', 'Tutoring', 'Homeless Shelter Aid', 'Youth Mentorship', 'Senior Center Visits', 'Fundraising Events', 'Food Bank Support']

'Peer_Mentoring_Activity': pd.Series(['Study Sessions', 'Career Guidance', 'Homework Assistance', 'Leadership Workshops', 'Academic Advising', 'Resume Review', 'Time Management', 'Conflict Resolution', 'Peer Counseling', 'Team Building', 'Homework Assistance', 'Resume Review', 'Peer Counseling', 'Time Management', 'Academic Advising', 'Study Sessions', 'Team Building', 'Career Guidance', 'Conflict Resolution', 'Leadership Workshops', 'Conflict Resolution', 'Peer Counseling', 'Team Building', 'Resume Review', 'Academic Advising', 'Homework Assistance', 'Leadership Workshops', 'Study Sessions', 'Career Guidance', 'Time Management', 'Career Guidance', 'Time Management', 'Homework Assistance', 'Study Sessions', 'Resume Review', 'Peer Counseling', 'Conflict Resolution', 'Team Building', 'Academic Advising', 'Leadership Workshops', 'Team Building', 'Time Management', 'Peer Counseling', 'Homework Assistance', 'Study Sessions', 'Career Guidance', 'Conflict Resolution', 'Academic Advising', 'Leadership Workshops', 'Resume Review', 'Study Sessions', 'Academic Advising', 'Team Building', 'Career Guidance', 'Homework Assistance', 'Conflict Resolution', 'Time Management', 'Resume Review', 'Peer Counseling', 'Leadership Workshops', 'Academic Advising', 'Study Sessions', 'Resume Review', 'Peer Counseling', 'Homework Assistance', 'Time Management', 'Career Guidance', 'Conflict Resolution', 'Team Building', 'Leadership Workshops', 'Homework Assistance', 'Study Sessions', 'Peer Counseling', 'Time Management', 'Leadership Workshops', 'Resume Review', 'Academic Advising', 'Conflict Resolution', 'Team Building', 'Career Guidance', 'Time Management', 'Career Guidance', 'Study Sessions', 'Academic Advising', 'Conflict Resolution', 'Team Building', 'Resume Review', 'Peer Counseling', 'Homework Assistance', 'Leadership Workshops', 'Leadership Workshops', 'Team Building', 'Time Management', 'Study Sessions', 'Academic Advising', 'Homework Assistance', 'Resume Review', 'Career Guidance', 'Peer

Counseling', 'Conflict Resolution'])),

'Graduation_Speech_Topic': pd.Series(['Overcoming Challenges', 'Pursuing Dreams', 'Embracing Change', 'Achieving Excellence', 'The Road Ahead', 'Empowering Futures', 'Celebrating Achievements', 'Inspiring Success', 'Gratitude and Hope', 'Life Journeys', 'Inspiring Success', 'Embracing Change', 'Pursuing Dreams', 'Gratitude and Hope', 'The Road Ahead', 'Achieving Excellence', 'Life Journeys', 'Celebrating Achievements', 'Empowering Futures', 'Overcoming Challenges', 'Gratitude and Hope', 'Achieving Excellence', 'Life Journeys', 'Embracing Change', 'Celebrating Achievements', 'Pursuing Dreams', 'Inspiring Success', 'Empowering Futures', 'Overcoming Challenges', 'The Road Ahead', 'Embracing Change', 'Achieving Excellence', 'The Road Ahead', 'Overcoming Challenges', 'Inspiring Success', 'Pursuing Dreams', 'Empowering Futures', 'Gratitude and Hope', 'Life Journeys', 'Celebrating Achievements', 'Life Journeys', 'Inspiring Success', 'Gratitude and Hope', 'The Road Ahead', 'Embracing Change', 'Pursuing Dreams', 'Achieving Excellence', 'Empowering Futures', 'Overcoming Challenges', 'Celebrating Achievements', 'Celebrating Achievements', 'Achieving Excellence', 'Life Journeys', 'Overcoming Challenges', 'Gratitude and Hope', 'Pursuing Dreams', 'The Road Ahead', 'Embracing Change', 'Inspiring Success', 'Empowering Futures', 'Empowering Futures', 'Pursuing Dreams', 'Gratitude and Hope', 'Achieving Excellence', 'Inspiring Success', 'The Road Ahead', 'Overcoming Challenges', 'Embracing Change', 'Celebrating Achievements', 'Life Journeys', 'The Road Ahead', 'Achieving Excellence', 'Overcoming Challenges', 'Inspiring Success', 'Gratitude and Hope', 'Pursuing Dreams', 'Embracing Change', 'Empowering Futures', 'Celebrating Achievements', 'Life Journeys', 'Life Journeys', 'Embracing Change', 'Gratitude and Hope', 'Achieving Excellence', 'Celebrating Achievements', 'The Road Ahead', 'Overcoming Challenges', 'Inspiring Success', 'Pursuing Dreams', 'Empowering Futures', 'Gratitude and Hope', 'The Road Ahead', 'Achieving Excellence', 'Life Journeys', 'Inspiring Success', 'Celebrating Achievements', 'Pursuing Dreams', 'Overcoming Challenges', 'Embracing Change', 'Empowering Futures'])),

'Field_Trip_Details': pd.Series(['Zoo Visit', 'Historical Museum Tour', 'Science Center Exploration', 'Botanical Garden Trip', 'Art Gallery Excursion', 'Nature Reserve Adventure', 'Planetarium Exploration', 'Aquarium Visit', 'Farm Tour', 'Cultural Heritage Walk', 'Botanical Garden Trip', 'Nature Reserve Adventure', 'Cultural Heritage Walk', 'Zoo Visit', 'Science Center Exploration', 'Art Gallery Excursion', 'Historical Museum Tour', 'Aquarium Visit', 'Planetarium Exploration', 'Farm Tour', 'Historical Museum Tour', 'Botanical Garden Trip', 'Aquarium Visit', 'Zoo Visit', 'Cultural Heritage Walk', 'Art Gallery Excursion', 'Planetarium Exploration', 'Nature Reserve Adventure', 'Science Center Exploration', 'Farm Tour', 'Art Gallery Excursion', 'Science Center Exploration', 'Nature Reserve Adventure', 'Historical Museum Tour', 'Aquarium Visit', 'Planetarium Exploration', 'Zoo Visit', 'Cultural Heritage Walk', 'Farm Tour', 'Botanical Garden Trip', 'Zoo Visit', 'Planetarium Exploration', 'Botanical Garden Trip', 'Cultural Heritage Walk', 'Science Center Exploration', 'Art Gallery Excursion', 'Aquarium Visit', 'Historical Museum Tour', 'Farm Tour', 'Nature Reserve Adventure', 'Nature Reserve Adventure', 'Science Center Exploration', 'Cultural Heritage Walk', 'Planetarium Exploration', 'Farm Tour', 'Historical Museum Tour', 'Botanical Garden Trip', 'Art Gallery Excursion', 'Aquarium Visit', 'Zoo Visit', 'Aquarium Visit', 'Nature Reserve Adventure', 'Historical Museum Tour', 'Zoo Visit', 'Science Center Exploration', 'Cultural Heritage Walk', 'Art Gallery Excursion', 'Botanical Garden Trip', 'Planetarium Exploration', 'Farm Tour', 'Science Center Exploration', 'Botanical Garden Trip', 'Aquarium Visit', 'Nature Reserve

Adventure', 'Planetarium Exploration', 'Cultural Heritage Walk', 'Farm Tour', 'Historical Museum Tour', 'Zoo Visit', 'Art Gallery Excursion', 'Art Gallery Excursion', 'Zoo Visit', 'Nature Reserve Adventure', 'Cultural Heritage Walk', 'Botanical Garden Trip', 'Planetarium Exploration', 'Historical Museum Tour', 'Science Center Exploration', 'Farm Tour', 'Aquarium Visit', 'Farm Tour', 'Science Center Exploration', 'Historical Museum Tour', 'Nature Reserve Adventure', 'Aquarium Visit', 'Art Gallery Excursion', 'Cultural Heritage Walk', 'Botanical Garden Trip', 'Zoo Visit', 'Planetarium Exploration'])),

'School_Safety_Drills_Record': pd.Series(['Lockdown Drills', 'Fire Drills', 'Evacuation Drills', 'Tornado Drills', 'Earthquake Drills', 'Active Shooter Drills', 'Tornado Drills', 'Fire Drills', 'Evacuation Drills', 'Lockdown Drills', 'Earthquake Drills', 'Active Shooter Drills', 'Fire Drills', 'Lockdown Drills', 'Tornado Drills', 'Evacuation Drills', 'Tornado Drills', 'Fire Drills', 'Active Shooter Drills', 'Earthquake Drills', 'Evacuation Drills', 'Earthquake Drills', 'Active Shooter Drills', 'Lockdown Drills', 'Fire Drills', 'Tornado Drills', 'Fire Drills', 'Active Shooter Drills', 'Lockdown Drills', 'Evacuation Drills', 'Active Shooter Drills', 'Fire Drills', 'Lockdown Drills', 'Tornado Drills', 'Evacuation Drills', 'Earthquake Drills', 'Earthquake Drills', 'Active Shooter Drills', 'Fire Drills', 'Lockdown Drills', 'Evacuation Drills', 'Tornado Drills', 'Lockdown Drills', 'Active Shooter Drills', 'Fire Drills', 'Tornado Drills', 'Lockdown Drills', 'Evacuation Drills', 'Earthquake Drills', 'Fire Drills', 'Tornado Drills', 'Active Shooter Drills', 'Evacuation Drills', 'Lockdown Drills', 'Earthquake Drills', 'Fire Drills', 'Evacuation Drills', 'Tornado Drills', 'Lockdown Drills', 'Active Shooter Drills', 'Fire Drills', 'Active Shooter Drills', 'Tornado Drills', 'Evacuation Drills', 'Earthquake Drills', 'Lockdown Drills', 'Tornado Drills', 'Fire Drills', 'Evacuation Drills', 'Active Shooter Drills', 'Lockdown Drills', 'Active Shooter Drills', 'Fire Drills', 'Tornado Drills', 'Evacuation Drills', 'Earthquake Drills', 'Evacuation Drills', 'Fire Drills', 'Tornado Drills', 'Lockdown Drills', 'Active Shooter Drills', 'Evacuation Drills', 'Tornado Drills', 'Lockdown Drills', 'Active Shooter Drills', 'Fire Drills', 'Active Shooter Drills', 'Tornado Drills', 'Evacuation Drills', 'Earthquake Drills', 'Lockdown Drills', 'Tornado Drills', 'Fire Drills', 'Evacuation Drills', 'Active Shooter Drills', 'Lockdown Drills', 'Active Shooter Drills', 'Fire Drills', 'Tornado Drills', 'Evacuation Drills', 'Earthquake Drills', 'Lockdown Drills', 'Active Shooter Drills', 'Evacuation Drills', 'Earthquake Drills', 'Active Shooter Drills', 'Fire Drills', 'Lockdown Drills', 'Tornado Drills'])),

'Student_Council_Position': pd.Series(['President', 'Vice President', 'Treasurer', 'Secretary', 'Public Relations Officer', 'Event Coordinator', 'Class Representative', 'Community Outreach Coordinator', 'Historian', 'Social Media Manager', 'Secretary', 'Historian', 'Public Relations Officer', 'Vice President', 'Treasurer', 'Event Coordinator', 'Class Representative', 'Community Outreach Coordinator', 'Social Media Manager', 'President', 'Community Outreach Coordinator', 'Event Coordinator', 'Public Relations Officer', 'Secretary', 'Historian', 'Class Representative', 'Social Media Manager', 'Vice President', 'President', 'Treasurer', 'Class Representative', 'Event Coordinator', 'Treasurer', 'Public Relations Officer', 'Secretary', 'Historian', 'Community Outreach Coordinator', 'Social Media Manager', 'President', 'Vice President', 'Treasurer', 'Historian', 'Social Media Manager', 'Public Relations Officer', 'Class Representative', 'Secretary', 'Vice President', 'Community Outreach Coordinator', 'Event Coordinator', 'President', 'Social Media Manager', 'Secretary', 'Class Representative', 'Historian', 'Treasurer', 'Public Relations Officer', 'Event Coordinator', 'Community Outreach Coordinator', 'Vice President', 'President', 'Event Coordinator', 'Vice President', 'Community Outreach Coordinator', 'Public Relations Officer', 'Historian', 'Social Media Manager', 'Class Representative', 'Secretary', 'Treasurer', 'President', 'Historian', 'Treasurer', 'Public Relations Officer', 'Class Representative', 'Event Coordinator', 'Secretary', 'Social Media Manager', 'Vice

President', 'Community Outreach Coordinator', 'President', 'Vice President', 'Community Outreach Coordinator', 'Secretary', 'Treasurer', 'Social Media Manager', 'Historian', 'Class Representative', 'Event Coordinator', 'Public Relations Officer', 'President', 'Class Representative', 'Public Relations Officer', 'Event Coordinator', 'Historian', 'Secretary', 'Social Media Manager', 'Community Outreach Coordinator', 'Treasurer', 'Vice President', 'President'])),

```
'PTA_Meeting_Attendance': pd.Series(['Present', 'Absent', 'Present', 'Absent', 'Present', 'Present', 'Absent', 'Present', 'Absent', 'Present', 'Absent', 'Present', 'Absent', 'Present', 'Absent', 'Absent', 'Present', 'Present', 'Absent', 'Present', 'Present', 'Absent', 'Absent', 'Present', 'Absent', 'Absent', 'Present', 'Absent', 'Present', 'Absent', 'Present', 'Present', 'Present', 'Absent', 'Present', 'Present', 'Present', 'Absent', 'Absent', 'Present', 'Present', 'Absent', 'Present', 'Present', 'Absent', 'Present', 'Absent', 'Absent', 'Present', 'Absent', 'Present', 'Present', 'Absent'])
```

'School_Budget_Details': pd.Series(['Operating Expenses', 'Capital Expenditures', 'Teacher Salaries', 'Maintenance Costs','Textbook Purchases', 'Technology Upgrades', 'Student Supplies', 'Administrative Costs','Facility Renovations', 'Transportation Expenses', 'Maintenance Costs', 'Facility Renovations','Administrative Costs', 'Technology Upgrades', 'Teacher Salaries', 'Operating Expenses','Student Supplies', 'Transportation Expenses', 'Textbook Purchases', 'Capital Expenditures','Administrative Costs', 'Student Supplies', 'Textbook Purchases', 'Maintenance Costs','Capital Expenditures', 'Facility Renovations', 'Technology Upgrades', 'Operating Expenses','Teacher Salaries', 'Transportation Expenses', 'Teacher Salaries', 'Technology Upgrades','Facility Renovations', 'Student Supplies', 'Operating Expenses', 'Textbook Purchases','Administrative Costs', 'Maintenance Costs', 'Capital Expenditures', 'Transportation Expenses','Capital Expenditures', 'Operating Expenses', 'Student Supplies', 'Teacher Salaries','Maintenance Costs', 'Administrative Costs', 'Facility Renovations', 'Transportation Expenses', 'Technology Upgrades','Textbook Purchases','Maintenance Costs', 'Textbook Purchases', 'Operating Expenses', 'Capital Expenditures','Facility Renovations', 'Teacher Salaries', 'Technology Upgrades', 'Administrative Costs','Student Supplies', 'Transportation Expenses', 'Administrative Costs','Operating Expenses', 'Maintenance Costs', 'Transportation Expenses', 'Facility Renovations','Capital Expenditures', 'Student Supplies', 'Textbook Purchases', 'Technology Upgrades','Teacher Salaries', 'Operating Expenses','Teacher Salaries', 'Technology Upgrades', 'Administrative Costs', 'Textbook Purchases','Facility Renovations', 'Maintenance Costs', 'Capital Expenditures','Transportation Expenses', 'Student Supplies', 'Operating Expenses','Maintenance Costs', 'Facility Renovations', 'Operating Expenses', 'Transportation Expenses','Student Supplies', 'Technology Upgrades', 'Capital Expenditures', 'Administrative Costs','Teacher Salaries', 'Textbook Purchases','Transportation Expenses', 'Student Supplies', 'Textbook Purchases', 'Capital Expenditures','Teacher Salaries', 'Operating Expenses', 'Maintenance Costs', 'Administrative Costs','Technology Upgrades'])),

'Alumni_Status': pd.Series(['Active', 'Inactive', 'Inactive', 'Active', 'Active', 'Inactive', 'Active', 'Inactive', 'Active', 'Inactive', 'Inactive', 'Active', 'Active', 'Inactive', 'Inactive', 'Active', 'Active',

```
'Inactive', 'Inactive', 'Active', 'Inactive', 'Active', 'Inactive', 'Active','Active', 'Inactive', 'Active',
'Inactive', 'Active', 'Inactive', 'Inactive', 'Active','Active', 'Inactive', 'Inactive', 'Active', 'Active',
'Inactive', 'Inactive', 'Active','Active', 'Inactive', 'Active', 'Inactive', 'Inactive', 'Active', 'Active',
'Inactive','Inactive', 'Active', 'Inactive', 'Active', 'Active', 'Inactive', 'Inactive', 'Active','Active',
'Inactive', 'Active', 'Inactive', 'Inactive', 'Active', 'Active', 'Inactive','Inactive', 'Inactive',
'Active','Inactive','Active', 'Active', 'Inactive', 'Inactive', 'Active','Active', 'Inactive', 'Inactive',
'Active','Inactive','Active','Active','Inactive','Inactive','Active','Inactive','Active','Inactive','Active',
'Inactive','Active','Inactive','Active','Inactive','Active','Inactive','Active','Inactive','Active','Inactive',
'Active','Inactive','Active']}]}
```

```
df=pd.DataFrame(data)
print(df)
```

8.1.2 Create a copy of Original Dataset

when you are performing Operations on Dataset, If there is no copy then the original Data will be Modified. If the Original Data will be Modified .There is no Security to data in Dataset.

- 1.Shallow Copy : creates a reference of Original Data Set(like 5-address is reference here for the value which is copied as another with the reference value).
- 2.Deep Copy: Directly Original Data will be Copied.

Now to perform the operations we can select the columns along with values as row selection and column selection by that lets take column selection along with the values as shown below.

#column selection & row selection

lets take 20 x 20 dataset from the above dataset

```
#column selection
# List of columns to select
columns_to_select = ['Student_id', 'Student_name', 'Date_Of_Birth', 'Gender',
'Parent_Guardian_Name', 'Parent_Guardian_Phone', 'Enrollment_Date',
'Graducation_Date', 'Class_Level', 'Course_Id','Course_Name', 'Exam_1_Score',
'Exam_2_Score', 'Exam_3_Score', 'Homework_1_Grade', 'Homework_2_Grade',
'Attendance_Record', 'Grades', 'Midterm_Exam_Score', 'Final_Exam_Score']
# Select the specified columns
selected_columns = df[columns_to_select]
# Print the selected columns
print(selected_columns)
#row selection
selected_rows = df.loc[0:19]
print(selected_rows)
```

Output:

_id	Student_name	Date_Of_Birth	Gender	Parent_Guardian_Name	\
0	101	Nandu	2002-09-01	Female	John
1	102	Navya	2003-01-01	Female	Mary
2	103	Mythili	2002-01-31	Female	Michael
3	104	Pujitha	2004-05-18	Female	Jennifer
4	105	Yashaswi	2005-09-27	Female	David
5	106	Kavya	2002-04-19	Female	Lisa
6	107	Saritha	2007-02-11	Female	James
7	108	Murali	2008-08-31	Male	Sarah
8	109	Mohan	2009-06-25	Male	Robert
9	110	Das	2010-04-16	Male	Laura
10	111	Anju	2011-10-07	Female	William
11	112	Sravani	2012-01-19	Female	Emily
12	113	Shiny	2013-07-14	Female	Joseph
13	114	Mani	2014-09-03	Male	Jessica
14	115	Prani	2015-03-28	Female	Richard
15	116	Deepu	2016-06-12	Female	Kimberly
16	117	Vasu	2017-12-08	Female	Charles
17	118	Moulika	2018-08-22	Female	Rebecca
18	119	Vinaya	2019-02-02	Female	Thomas
19	120	Mahitha	2020-11-30	Female	Amanda
	Parent_Guardian_Phone	Enrollment_Date	Graducation_Date	Class_Level	
\					
0	903456789	2023-04-13	2024-05-15	10th class	
1	807654321	2023-04-13	2024-05-15	10th class	
2	955123456	2023-04-13	2024-05-15	10th class	
3	754987321	2023-04-13	2024-05-15	10th class	
4	987654123	2023-04-13	2024-05-15	10th class	
5	823987456	2023-04-13	2024-05-15	10th class	
6	956321987	2023-04-13	2024-05-15	10th class	
7	987123456	2023-04-13	2024-05-15	10th class	
8	789654321	2023-04-13	2024-05-15	10th class	
9	821654987	2023-04-13	2024-05-15	10th class	
10	806789123	2023-04-13	2024-05-15	10th class	
11	789123456	2023-04-13	2024-05-15	10th class	
12	954123789	2023-04-13	2024-05-15	10th class	
13	911456987	2023-04-13	2024-05-15	10th class	
14	987321456	2023-04-13	2024-05-15	10th class	
15	954321789	2022-05-16	2024-04-15	Intermediate	
16	789456123	2022-05-16	2024-04-15	Intermediate	
17	973654987	2022-05-16	2024-04-15	Intermediate	
18	856987123	2022-05-16	2024-04-15	Intermediate	
19	987456321	2022-05-16	2024-04-15	Intermediate	

	Course_Id	Course_Name	Exam_1_Score	Exam_2_Score	Exam_3_Score	\
0	HS-10	SSC	71	85	63	
1	HS-10	CBSE	88	96	98	
2	HS-10	SSC	63	75	72	
3	HS-10	CBSE	94	64	53	
4	HS-10	SSC	52	90	85	
5	HS-10	CBSE	78	81	90	
6	HS-10	SSC	69	70	69	
7	HS-10	CBSE	85	67	79	
8	HS-10	SSC	74	73	45	
9	HS-10	CBSE	91	59	58	
10	HS-10	SSC	59	43	87	
11	HS-10	CBSE	82	91	55	
12	HS-10	SSC	66	87	84	
13	HS-10	CBSE	76	50	67	
14	HS-10	SSC	49	69	40	
15	BOI-11-12	CBSE	97	79	76	
16	BOI-11-12	SSC	57	100	88	
17	BOI-11-12	CBSE	84	60	62	
18	BOI-11-12	SSC	42	63	100	
19	BOI-11-12	CBSE	72	61	95	

	Homework_1_Grade	Homework_2_Grade	Attendance_Record	Grades	\
0	84	81	82%	A	
1	92	98	67%	C	
2	83	75	45%	B	
3	70	80	59%	B	
4	83	87	73%	C	
5	78	70	91%	B	
6	76	98	52%	B+	
7	65	72	78%	D	
8	88	66	64%	A	
9	83	71	86%	B+	
10	97	85	55%	A	
11	86	98	68%	B+	
12	60	79	71%	B	
13	73	63	49%	C	
14	81	86	89%	B	
15	66	91	42%	B+	
16	80	70	76%	C	
17	66	61	61%	C	
18	84	75	58%	B+	
19	65	62	87%	A	

	Midterm_Exam_Score	Final_Exam_Score
0	80	100
1	100	160
2	223	250
3	390	400
4	405	425
5	215	245
6	131	145
7	159	234
8	267	356
9	370	390
10	170	189
11	160	190
12	230	267
13	160	189
14	168	178
15	115	134
16	194	234
17	150	189
18	262	290
19	260	289

8.1.3 Code for Data Set Generation (20X20):

```
import pandas as pd
pd.set_option('display.max_columns', None)
pd.set_option('display.max_rows', None)
data = {
    'Student_id': pd.Series([101, 102, 103, 104, 105, 106, 107, 108, 109, 110, 111, 112,
    113, 114, 115, 116, 117, 118, 119, 120]),
    'Student_name': pd.Series(['Nandu', 'Navya', 'Mythili', 'Pujitha', 'Yashaswi', 'Kavya',
    'Saritha', 'Murali', 'Mohan', 'Das', 'Anju', 'Sravani', 'Shiny', 'Mani', 'Prani', 'Deepu',
    'Vasu', 'Moulika', 'Vinaya', 'Mahitha']),
    'Date_Of_Birth': pd.Series(['2002-09-01', '2003-01-01', '2002-01-31', '2004-05-
    18', '2005-09-27', '2002-04-19', '2007-02-11', '2008-08-31', '2009-06-25', '2010-04-
    16', '2011-10-07', '2012-01-19', '2013-07-14', '2014-09-03', '2015-03-28', '2016-06-
    12', '2017-12-08', '2018-08-22', '2019-02-02', '2020-11-30']),
    'Gender': pd.Series(['Female', 'Female', 'Female', 'Female', 'Female', 'Female',
    'Female', 'Male', 'Male', 'Male', 'Female', 'Female', 'Female', 'Male', 'Female', 'Female',
    'Female', 'Female', 'Female', 'Female']),
    'Parent_Guardian_Name':
    pd.Series(['John', 'Mary', 'Michael', 'Jennifer', 'David', 'Lisa', 'James', 'Sarah', 'Robert', 'Lau
    ra',
    'William', 'Emily', 'Joseph', 'Jessica', 'Richard', 'Kimberly', 'Charles', 'Rebecca', 'Thomas', '
    Amanda']),
    'Parent_Guardian_Phone':
    pd.Series([903456789, 807654321, 955123456, 754987321, 987654123, 823987456, 9
    56321987, 987123456, 789654321, 821654987, 806789123, 789123456, 954123789, 91
    1456987, 987321456, 954321789, 789456123, 973654987, 856987123, 987456321]),
    'Enrollment_Date': pd.Series(['2023-04-13', '2023-04-13', '2023-04-13', '2023-04-13',
    '2023-04-13', '2023-04-13', '2023-04-13', '2023-04-13', '2023-04-13', '2023-04-13',
    '2023-04-13', '2023-04-13', '2023-04-13', '2023-04-13', '2022-05-16', '2022-05-16',
    '2022-05-16', '2022-05-16']),
    'Graduation_Date': pd.Series(['2024-05-15', '2024-05-15', '2024-05-15', '2024-05-
    15', '2024-05-15', '2024-05-15', '2024-05-15', '2024-05-15', '2024-05-15', '2024-05-15',
    '2024-05-15', '2024-05-15', '2024-05-15', '2024-05-15', '2024-05-15', '2024-04-
    15', '2024-04-15', '2024-04-15', '2024-04-15']),
    'Class_Level': pd.Series(['10th class', '10th class', '10th class', '10th class', '10th class',
    '10th class', '10th class', '10th class', '10th class', '10th class', '10th class', '10th class',
    '10th class', '10th class', '10th class', 'Intermediate', 'Intermediate', 'Intermediate',
    'Intermediate', 'Intermediate']),
    'Course_Id': pd.Series(['HS-10', 'HS-10', 'HS-10', 'HS-10', 'HS-10', 'HS-10', 'HS-10',
    'HS-10', 'HS-10', 'HS-10', 'HS-10', 'HS-10', 'HS-10', 'HS-10', 'HS-10', 'BOI-11-12',
    'BOI-11-12', 'BOI-11-12', 'BOI-11-12']),
    'Course_Name': pd.Series(['SSC', 'CBSE', 'SSC', 'CBSE', 'SSC', 'CBSE', 'SSC',
```

```

'CBSE', 'SSC', 'CBSE', 'SSC', 'CBSE', 'SSC', 'CBSE', 'SSC', 'CBSE', 'SSC', 'CBSE',
'SSC', 'CBSE']),
'Exam_1_Score': pd.Series([71, 88, 63, 94, 52, 78, 69, 85, 74, 91, 59, 82, 66, 76, 49,
97, 57, 84, 42, 72]),
'Exam_2_Score': pd.Series([85, 96, 75, 64, 90, 81, 70, 67, 73, 59, 43, 91, 87, 50, 69,
79, 100, 60, 63, 61]),
'Exam_3_Score': pd.Series([63, 98, 72, 53, 85, 90, 69, 79, 45, 58, 87, 55, 84, 67, 40,
76, 88, 62, 100, 95]),
'Homework_1_Grade': pd.Series([84, 92, 83, 70, 83, 78, 76, 65, 88, 83, 97, 86, 60,
73, 81, 66, 80, 66, 84, 65]),

'Homework_2_Grade': pd.Series([81, 98, 75, 80, 87, 70, 98, 72, 66, 71, 85, 98, 79,
63, 86, 91, 70, 61, 75, 62]),
'Midterm_Exam_Score': pd.Series([80, 100, 223, 390, 405, 215, 131, 159, 267, 370,
170, 160, 230, 160, 168, 115, 194, 150, 262, 260]),
'Final_Exam_Score': pd.Series([100, 160, 250, 400, 425, 245, 145, 234, 356, 390,
189, 190, 267, 189, 178, 134, 234, 189, 290, 289]),
'Attendance_Record':pd.Series(['82%', '67%', '45%', '59%', '73%', '91%', '52%', '78%',
'64%', '86%', '55%', '68%', '71%', '49%', '89%', '42%', '76%', '61%', '58%', '87%']),
'Grades':pd.Series(['A', 'C', 'B', 'B', 'C', 'B', 'B+', 'D', 'A', 'B+', 'A', 'B+', 'B', 'C', 'B', 'B+',
'C', 'C', 'B+', 'A']),}
df=pd.DataFrame(data)
print(df)

```

output:

	Student_id	Student_name	Date_Of_Birth	Gender	Parent_Guardian_Name \
0	101	Nandu	2002-09-01	Female	John
1	102	Navya	2003-01-01	Female	Mary
2	103	Mythili	2002-01-31	Female	Michael
3	104	Pujitha	2004-05-18	Female	Jennifer
4	105	Yashaswi	2005-09-27	Female	David
5	106	Kavya	2002-04-19	Female	Lisa
6	107	Saritha	2007-02-11	Female	James
7	108	Murali	2008-08-31	Male	Sarah
8	109	Mohan	2009-06-25	Male	Robert
9	110	Das	2010-04-16	Male	Laura
10	111	Anju	2011-10-07	Female	William
11	112	Sravani	2012-01-19	Female	Emily
12	113	Shiny	2013-07-14	Female	Joseph
13	114	Mani	2014-09-03	Male	Jessica
14	115	Prani	2015-03-28	Female	Richard
15	116	Deepu	2016-06-12	Female	Kimberly
16	117	Vasu	2017-12-08	Female	Charles
17	118	Moulika	2018-08-22	Female	Rebecca
18	119	Vinaya	2019-02-02	Female	Thomas
19	120	Mahitha	2020-11-30	Female	Amanda

Parent_Guardian_Phone	Enrollment_Date	Graduation_Date	Class_Level	\
0	903456789	2023-04-13	2024-05-15	10th class
1	807654321	2023-04-13	2024-05-15	10th class
2	955123456	2023-04-13	2024-05-15	10th class
3	754987321	2023-04-13	2024-05-15	10th class
4	987654123	2023-04-13	2024-05-15	10th class
5	823987456	2023-04-13	2024-05-15	10th class
6	956321987	2023-04-13	2024-05-15	10th class
7	987123456	2023-04-13	2024-05-15	10th class
8	789654321	2023-04-13	2024-05-15	10th class
9	821654987	2023-04-13	2024-05-15	10th class
10	806789123	2023-04-13	2024-05-15	10th class
11	789123456	2023-04-13	2024-05-15	10th class
12	954123789	2023-04-13	2024-05-15	10th class
13	911456987	2023-04-13	2024-05-15	10th class
14	987321456	2023-04-13	2024-05-15	10th class
15	954321789	2022-05-16	2024-04-15	Intermediate
16	789456123	2022-05-16	2024-04-15	Intermediate
17	973654987	2022-05-16	2024-04-15	Intermediate
18	856987123	2022-05-16	2024-04-15	Intermediate
19	987456321	2022-05-16	2024-04-15	Intermediate

Course_Id	Course_Name	Exam_1_Score	Exam_2_Score	Exam_3_Score	\
0	HS-10	SSC	71	85	63
1	HS-10	CBSE	88	96	98
2	HS-10	SSC	63	75	72
3	HS-10	CBSE	94	64	53
4	HS-10	SSC	52	90	85
5	HS-10	CBSE	78	81	90
6	HS-10	SSC	69	70	69
7	HS-10	CBSE	85	67	79
8	HS-10	SSC	74	73	45
9	HS-10	CBSE	91	59	58
10	HS-10	SSC	59	43	87
11	HS-10	CBSE	82	91	55
12	HS-10	SSC	66	87	84
13	HS-10	CBSE	76	50	67
14	HS-10	SSC	49	69	40
15	BOI-11-12	CBSE	97	79	76
16	BOI-11-12	SSC	57	100	88
17	BOI-11-12	CBSE	84	60	62
18	BOI-11-12	SSC	42	63	100
19	BOI-11-12	CBSE	72	61	95

Homework_1_Grade	Homework_2_Grade	Midterm_Exam_Score	Final_Exam_Score	\
0	84	81	80	100
1	92	98	100	160
2	83	75	223	250
3	70	80	390	400
4	83	87	405	425
5	78	70	215	245
6	76	98	131	145
7	65	72	159	234
8	88	66	267	356
9	83	71	370	390
10	97	85	170	189
11	86	98	160	190
12	60	79	230	267
13	73	63	160	189
14	81	86	168	178
15	66	91	115	134
16	80	70	194	234
17	66	61	150	189
18	84	75	262	290
19	65	62	260	289

Attendance_Record	Grades
0	82% A
1	67% C
2	45% B
3	59% B
4	73% C
5	91% B
6	52% B+
7	78% D
8	64% A
9	86% B+
10	55% A
11	68% B+
12	71% B
13	49% C
14	89% B
15	42% B+
16	76% C
17	61% C
18	58% B+
19	87% A

Info():

```
#to know all  
df.info()
```

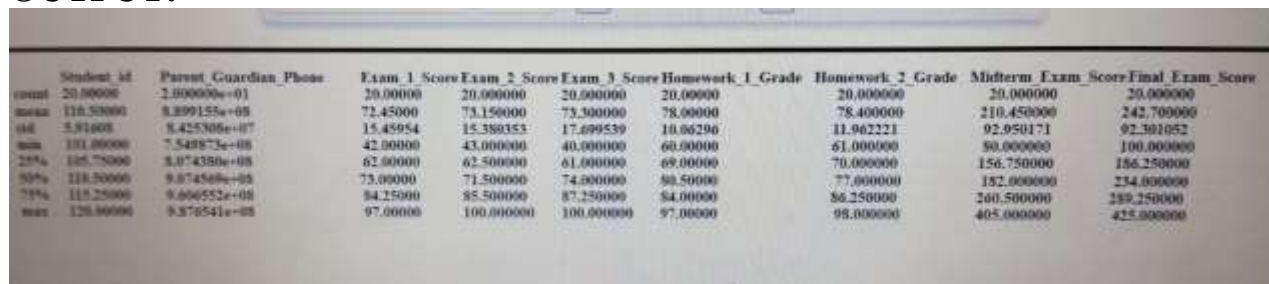
Output():

```
<class 'pandas.core.frame.DataFrame'>  
RangeIndex: 20 entries, 0 to 19  
Data columns (total 20 columns):  
#   Column                                Non-Null Count  Dtype    
---  -  
0   Student_id                            20 non-null     int64    
1   Student_name                          20 non-null     object   
2   Date_Of_Birth                        20 non-null     object   
3   Gender                               20 non-null     object   
4   Parent_Guardian_Name                  20 non-null     object   
5   Parent_Guardian_Phone                 20 non-null     int64    
6   Enrollment_Date                       20 non-null     object   
7   Graducation_Date                     20 non-null     object   
8   Class_Level                           20 non-null     object   
9   Course_Id                             20 non-null     object   
10  Course_Name                           20 non-null     object   
11  Exam_1_Score                           20 non-null     int64    
12  Exam_2_Score                           20 non-null     int64    
13  Exam_3_Score                           20 non-null     int64    
14  Homework_1_Grade                       20 non-null     int64    
15  Homework_2_Grade                       20 non-null     int64    
16  Midterm_Exam_Score                     20 non-null     int64    
17  Final_Exam_Score                       20 non-null     int64    
18  Attendance_Record                      20 non-null     object   
19  Grades                                 20 non-null     object   
dtypes: int64(9), object(11)  
memory usage: 3.2+ KB
```

describe():

```
df.describe()
```

OUTPUT:



	Student_id	Parent_Guardian_Phone	Exam_1_Score	Exam_2_Score	Exam_3_Score	Homework_1_Grade	Homework_2_Grade	Midterm_Exam_Score	Final_Exam_Score
count	20.00000	2.000000e+01	20.00000	20.000000	20.000000	20.00000	20.000000	20.000000	20.000000
mean	118.50000	8.399152e+08	72.45000	73.150000	73.300000	78.00000	78.400000	210.450000	242.700000
std	3.91000	8.425306e+07	15.45954	15.380353	17.699539	10.06296	11.962221	92.950171	92.301052
min	101.00000	7.549873e+08	42.00000	43.000000	40.000000	60.00000	61.000000	80.000000	100.000000
25%	105.75000	8.074380e+08	62.00000	62.500000	61.000000	69.00000	70.000000	156.750000	186.250000
50%	118.50000	8.874569e+08	73.00000	71.500000	74.000000	80.50000	77.000000	182.000000	234.000000
75%	119.25000	9.606552e+08	84.25000	85.500000	87.250000	84.00000	86.250000	260.500000	289.250000
max	129.00000	9.878541e+08	97.00000	100.000000	100.000000	97.00000	98.000000	405.000000	425.000000

shape :

`print(df.shape)` # indicating no.of rows,column.

Output: (20, 20)

```
print(df.size)
print(df.index)
print(df.info())
print(df.describe())
print(df.memory_usage())
print(df.count())
print(df.ndim)
```

OUTPUT:

400

RangeIndex(start=0, stop=20, step=1)

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

RangeIndex: 20 entries, 0 to 19

Data columns (total 20 columns):

#	Column	Non-Null Count	Dtype
0	Student_id	20 non-null	int64
1	Student_name	20 non-null	object
2	Date_Of_Birth	20 non-null	object
3	Gender	20 non-null	object
4	Parent_Guardian_Name	20 non-null	object
5	Parent_Guardian_Phone	20 non-null	int64
6	Enrollment_Date	20 non-null	object
7	Graduation_Date	20 non-null	object
8	Class_Level	20 non-null	object
9	Course_Id	20 non-null	object
10	Course_Name	20 non-null	object
11	Exam_1_Score	20 non-null	int64
12	Exam_2_Score	20 non-null	int64
13	Exam_3_Score	20 non-null	int64
14	Homework_1_Grade	20 non-null	int64
15	Homework_2_Grade	20 non-null	int64
16	Attendance_Record	20 non-null	object
17	Grades	20 non-null	object
18	Parent_Teacher_Meeting_Date	20 non-null	object
19	Library_Books_Checked_Out	20 non-null	object

dtypes: int64(7), object(13)

memory usage: 3.2+ KB

None

	Student_id	Parent_Guardian_Phone	Exam_1_Score	Exam_2_Score	\
count	20.00000	2.000000e+01	20.00000	20.000000	
mean	110.50000	8.899155e+08	72.45000	73.150000	
std	5.91608	8.425308e+07	15.45954	15.380353	
min	101.00000	7.549873e+08	42.00000	43.000000	
25%	105.75000	8.074380e+08	62.00000	62.500000	
50%	110.50000	9.074569e+08	73.00000	71.500000	
75%	115.25000	9.606552e+08	84.25000	85.500000	
max	120.00000	9.876541e+08	97.00000	100.000000	

Exam_3_Score	Homework_1_Grade	Homework_2_Grade	
count	20.000000	20.000000	20.000000
mean	73.300000	78.000000	78.400000
std	17.699539	10.06296	11.962221
min	40.000000	60.000000	61.000000
25%	61.000000	69.000000	70.000000
50%	74.000000	80.500000	77.000000
75%	87.250000	84.000000	86.250000
max	100.000000	97.000000	98.000000
Index		128	
Student_id		160	
Student_name		160	
Date_Of_Birth		160	
Gender		160	
Parent_Guardian_Name		160	
Parent_Guardian_Phone		160	
Enrollment_Date		160	
Graducation_Date		160	
Class_Level		160	
Course_Id		160	
Course_Name		160	
Exam_1_Score		160	
Exam_2_Score		160	
Exam_3_Score		160	
Homework_1_Grade		160	
Homework_2_Grade		160	
Attendance_Record		160	
Grades		160	
Parent_Teacher_Meeting_Date		160	
Library_Books_Checked_Out		160	
dtype: int64			
Student_id		20	
Student_name		20	
Date_Of_Birth		20	
Gender		20	
Parent_Guardian_Name		20	
Parent_Guardian_Phone		20	
Enrollment_Date		20	
Graducation_Date		20	
Class_Level		20	
Course_Id		20	
Course_Name		20	
Exam_1_Score		20	
Exam_2_Score		20	
Exam_3_Score		20	
Homework_1_Grade		20	
Homework_2_Grade		20	
Attendance_Record		20	
Grades		20	
Parent_Teacher_Meeting_Date		20	
Library_Books_Checked_Out		20	
dtype: int64			

8.2 Data Cleaning and Preparation:

HANDLING MISSING VALUES -----> A data cleaning is a process of Remove or Replace the NaN values which are

- 1.empty cell
- 2.wrong format
- 3.wrong data
- 4.remove duplicates.

8.2.1 Empty cell:

when a cell contains NAN value. and here we 2 methods to check a cell is empty or not.

1. ".isnull()" --->True if it is, if not it returns False
2. ".notnull()" ---->NaN -->True, it is False

```
# isnull() concepts
print(df.isnull())
print('*****isnull().sum()*****')
print(df.isnull().sum())
print('*****isnull().sum().mean()*****')
print(df.isnull().sum().mean())
print('*****notnull()*****')
# notnull()
print(df.notnull())
```

OUTPUT:

	Student_Id	Student_Name	Date_Of_Birth	Gender	Parent_Guardian_Name	
0	False	False	False	False	False	
1	False	False	False	False	False	
2	False	False	False	False	False	
3	False	False	False	False	False	
4	False	False	False	False	False	
5	False	False	False	False	False	
6	False	False	False	False	False	
7	False	False	False	False	False	
8	False	False	False	False	False	
9	False	False	False	False	False	
10	False	False	False	False	False	
11	False	False	False	False	False	
12	False	False	False	False	False	
13	False	False	False	False	False	
14	False	False	False	False	False	
15	False	False	False	False	False	
16	False	False	False	False	False	
17	False	False	False	False	False	
18	False	False	False	False	False	
19	False	False	False	False	False	
	Parent_Guardian_Photo	Parent_Guardian_Date	Home_Contact_No	Class_Room		
0	False	False	False	False	False	
1	False	False	False	False	False	
2	False	False	False	False	False	
3	False	False	False	False	False	
4	False	False	False	False	False	
5	False	False	False	False	False	
6	False	False	False	False	False	
7	False	False	False	False	False	
8	False	False	False	False	False	
9	False	False	False	False	False	
10	False	False	False	False	False	
11	False	False	False	False	False	
12	False	False	False	False	False	
13	False	False	False	False	False	
14	False	False	False	False	False	
15	False	False	False	False	False	
16	False	False	False	False	False	
17	False	False	False	False	False	
18	False	False	False	False	False	
19	False	False	False	False	False	
	Course_Id	Course_Name	Exam_1_Score	Exam_2_Score	Exam_3_Score	
0	False	False	False	False	False	
1	False	False	False	False	False	
2	False	False	False	False	False	
3	False	False	False	False	False	
4	False	False	False	False	False	
5	False	False	False	False	False	
6	False	False	False	False	False	
7	False	False	False	False	False	
8	False	False	False	False	False	
9	False	False	False	False	False	
10	False	False	False	False	False	
11	False	False	False	False	False	
12	False	False	False	False	False	
13	False	False	False	False	False	
14	False	False	False	False	False	
15	False	False	False	False	False	
16	False	False	False	False	False	
17	False	False	False	False	False	
18	False	False	False	False	False	
19	False	False	False	False	False	

	Homework_1_Grade	Homework_2_Grade	Midterm_Exam_Score	Final_Exam_Score
0	False	False	False	False
1	False	False	False	False
2	False	False	False	False
3	False	False	False	False
4	False	False	False	False
5	False	False	False	False
6	False	False	False	False
7	False	False	False	False
8	False	False	False	False
9	False	False	False	False
10	False	False	False	False
11	False	False	False	False
12	False	False	False	False
13	False	False	False	False
14	False	False	False	False
15	False	False	False	False
16	False	False	False	False
17	False	False	False	False
18	False	False	False	False
19	False	False	False	False

	Attendance_Record	Grades
0	False	False
1	False	False
2	False	False
3	False	False
4	False	False
5	False	False
6	False	False
7	False	False
8	False	False
9	False	False
10	False	False
11	False	False
12	False	False
13	False	False
14	False	False
15	False	False
16	False	False
17	False	False
18	False	False
19	False	False

```

*****isnull().sum()*****
Student_Id      0
Student_name    0
Date_Of_Birth   0
Gender          0
Parent_Guardian_Name  0
Parent_Guardian_Phone  0
Enrollment_Date  0
Graduation_Date  0
Class_Level     0
Course_Id       0
Course_Name     0
Exam_1_Score    0
Exam_2_Score    0
Exam_3_Score    0
Homework_1_Grade  0
Homework_2_Grade  0
Midterm_Exam_Score  0
Final_Exam_Score  0
Attendance_Record  0
Grades          0
dtype: int64
*****isnull().sum(),mean()*****
0.0

```

	Student_Id	Student_name	Date_Of_Birth	Gender	Parent_Guardian_Name
0	True	True	True	True	True
1	True	True	True	True	True
2	True	True	True	True	True
3	True	True	True	True	True
4	True	True	True	True	True
5	True	True	True	True	True
6	True	True	True	True	True
7	True	True	True	True	True
8	True	True	True	True	True
9	True	True	True	True	True
10	True	True	True	True	True
11	True	True	True	True	True
12	True	True	True	True	True
13	True	True	True	True	True
14	True	True	True	True	True
15	True	True	True	True	True
16	True	True	True	True	True
17	True	True	True	True	True
18	True	True	True	True	True
19	True	True	True	True	True

	Parent_Guardian_Phone	Enrollment_Date	Graduation_Date	Class_Level
0	True	True	True	True
1	True	True	True	True
2	True	True	True	True
3	True	True	True	True
4	True	True	True	True
5	True	True	True	True
6	True	True	True	True
7	True	True	True	True
8	True	True	True	True
9	True	True	True	True
10	True	True	True	True
11	True	True	True	True
12	True	True	True	True
13	True	True	True	True
14	True	True	True	True
15	True	True	True	True
16	True	True	True	True
17	True	True	True	True
18	True	True	True	True
19	True	True	True	True

	Course_Id	Course_Name	Exam_1_Score	Exam_2_Score	Exam_3_Score	
0	True	True	True	True	True	
1	True	True	True	True	True	
2	True	True	True	True	True	
3	True	True	True	True	True	
4	True	True	True	True	True	
5	True	True	True	True	True	
6	True	True	True	True	True	
7	True	True	True	True	True	
8	True	True	True	True	True	
9	True	True	True	True	True	
10	True	True	True	True	True	
11	True	True	True	True	True	
12	True	True	True	True	True	
13	True	True	True	True	True	
14	True	True	True	True	True	
15	True	True	True	True	True	
16	True	True	True	True	True	
17	True	True	True	True	True	
18	True	True	True	True	True	
19	True	True	True	True	True	

	Homework_1_Grade	Homework_2_Grade	Midterm_Exam_Score	Final_Exam_Score	
0	True	True	True	True	
1	True	True	True	True	
2	True	True	True	True	
3	True	True	True	True	
4	True	True	True	True	
5	True	True	True	True	
6	True	True	True	True	
7	True	True	True	True	
8	True	True	True	True	
9	True	True	True	True	
10	True	True	True	True	
11	True	True	True	True	
12	True	True	True	True	
13	True	True	True	True	
14	True	True	True	True	
15	True	True	True	True	
16	True	True	True	True	
17	True	True	True	True	
18	True	True	True	True	
19	True	True	True	True	

	Attendance_Record	Grades
0	True	True
1	True	True
2	True	True
3	True	True
4	True	True
5	True	True
6	True	True
7	True	True
8	True	True
9	True	True
10	True	True
11	True	True
12	True	True
13	True	True
14	True	True
15	True	True
16	True	True
17	True	True
18	True	True
19	True	True

8.2.2 Wrong Data :

we can treat it as a MIS-MATCHED DATA here we 2 methods are:

1.dropna() --->it will drop NaN values from the rows

2.fillna()

(i) fillna(method='pad') --->previous value

(ii) fillna(method='bfill') ---->next value added to null

example: Max Marks =100 Marks =105 ----->which is a wrong data or value obtained.

```
print(" -----dropna()-----") # which will drop the NaN row by
deleting it.
print(df.dropna())
print("\n")
print(" -----fillna()-----")
# to fill the empty places of NaN we use to replace that value by assigning the specific value to it.
print(df.fillna(10))
print("\n")
print(" -----fillna(method='pad')-----")
# it will fill the Nan value as we seen below 30 is taken.
print(df.fillna(method='pad'))
print("\n")
print(" -----fillna(method='bfill')-----")
#it will fill next value as we seen below 50 is taken.
print(df.fillna(method='bfill'))
```

OUTPUT :

dropna()						
	Student_Id	Student_Name	Date_Of_Birth	Gender	Parent_Guardian_Name	
0	101	Handu	2002-09-01	Female	John	
1	102	Navya	2003-01-01	Female	Mary	
2	103	Mythili	2002-01-11	Female	Michael	
3	104	Pujitha	2004-05-18	Female	Jennifer	
4	105	Vashasi	2005-09-27	Female	David	
5	106	Kavya	2002-04-19	Female	Lisa	
6	107	Saritha	2007-02-11	Female	James	
7	108	Murali	2004-08-31	Male	Sarah	
8	109	Mohan	2009-06-25	Male	Robert	
9	110	Das	2010-04-16	Male	Laura	
10	111	Anju	2011-10-07	Female	William	
11	112	Seavani	2012-01-19	Female	Emily	
12	113	Shiny	2013-07-14	Female	Joseph	
13	114	Mani	2014-09-03	Male	Jessica	
14	115	Prani	2015-03-28	Female	Richard	
15	116	Deepu	2016-06-12	Female	Kimberly	
16	117	Vasu	2017-12-08	Female	Charles	
17	118	Moulika	2018-08-22	Female	Rebecca	
18	119	Vinaya	2019-02-02	Female	Thomas	
19	120	Hanitha	2020-11-30	Female	Amanda	
	Parent_Guardian_Phone	Enrollment_Date	Graduation_Date	Class_Level		
0	983456789	2023-04-13	2024-05-15	10th class		
1	887654321	2023-04-13	2024-05-15	10th class		
2	955123456	2023-04-13	2024-05-15	10th class		
3	754987321	2023-04-13	2024-05-15	10th class		
4	987654321	2023-04-13	2024-05-15	10th class		
5	823987456	2023-04-13	2024-05-15	10th class		
6	956321098	2023-04-13	2024-05-15	10th class		
7	987123456	2023-04-13	2024-05-15	10th class		
8	789054321	2023-04-13	2024-05-15	10th class		
9	823654987	2023-04-13	2024-05-15	10th class		
10	980789012	2023-04-13	2024-05-15	10th class		
11	789123456	2023-04-13	2024-05-15	10th class		
12	954123789	2023-04-13	2024-05-15	10th class		
13	911456987	2023-04-13	2024-05-15	10th class		
14	987321456	2023-04-13	2024-05-15	10th class		
15	954321789	2022-05-16	2024-04-15	Intermediate		
16	789456123	2022-05-16	2024-04-15	Intermediate		
17	973654987	2022-05-16	2024-04-15	Intermediate		
18	856987123	2022-05-16	2024-04-15	Intermediate		
19	987456121	2022-05-16	2024-04-15	Intermediate		

	Course_Id	Course_Name	Exam_1_Score	Exam_2_Score	Exam_3_Score	
0	HS-10	SSC	71	85	63	
1	HS-10	CBSE	88	96	98	
2	HS-10	SSC	63	75	72	
3	HS-10	CBSE	94	64	53	
4	HS-10	SSC	52	90	85	
5	HS-10	CBSE	78	81	90	
6	HS-10	SSC	69	70	69	
7	HS-10	CBSE	85	67	79	
8	HS-10	SSC	74	73	45	
9	HS-10	CBSE	91	59	58	
10	HS-10	SSC	59	43	87	
11	HS-10	CBSE	82	91	55	
12	HS-10	SSC	66	87	84	
13	HS-10	CBSE	76	50	67	
14	HS-10	SSC	49	69	40	
15	BOI-11-12	CBSE	97	79	76	
16	BOI-11-12	SSC	57	100	88	
17	BOI-11-12	CBSE	88	67	62	
18	BOI-11-12	SSC	42	61	100	
19	BOI-11-12	CBSE	72	61	95	
	Homework_1_Grade	Homework_2_Grade	Midterm_Exam_Score	Final_Exam_Score		
0	84	81	80	100		
1	92	98	100	100		
2	83	75	223	250		
3	70	80	390	400		
4	83	87	485	425		
5	78	79	215	245		
6	76	98	131	145		
7	65	72	159	234		
8	88	66	267	356		
9	83	71	379	390		
10	97	85	170	189		
11	86	98	160	390		
12	68	79	230	267		
13	73	63	100	189		
14	81	86	108	178		
15	66	94	135	154		
16	89	70	194	234		
17	66	63	150	189		
18	84	75	282	290		
19	65	62	260	289		

Attendance_Record_Grades

	Attendance_Record	Grades
0	82%	A
1	67%	C
2	45%	B
3	59%	B
4	73%	C
5	91%	B
6	52%	B+
7	78%	D
8	64%	A
9	86%	B+
10	55%	A
11	68%	B+
12	71%	B
13	49%	C
14	89%	B
15	42%	B+
16	76%	C
17	61%	C
18	58%	B+
19	87%	A

	Course_ID	Course_Name	Exam_1_Score	Exam_2_Score	Exam_3_Score	%
0	005-10	SYC	71	85	63	
1	005-10	CRSE	80	96	90	
2	005-10	SYC	63	75	72	
3	005-10	CRSE	84	84	93	
4	005-10	SYC	53	60	85	
5	005-10	CRSE	70	81	90	
6	005-10	SYC	60	70	69	
7	005-10	CRSE	85	67	79	
8	005-10	SYC	74	73	45	
9	005-10	CRSE	93	59	50	
10	005-10	SYC	59	43	87	
11	005-10	CRSE	82	91	55	
12	005-10	SYC	66	87	80	
13	005-10	CRSE	76	50	67	
14	005-10	SYC	40	40	40	
15	005-13-1.1	CRSE	97	79	76	
16	005-13-1.2	SYC	57	100	80	
17	005-13-1.2	CRSE	84	100	67	
18	005-13-1.2	SYC	62	53	100	
19	007-11-1.3	CRSE	72	83	95	

	Attendance_Record	Grades
0	82%	A
1	67%	C
2	45%	B
3	59%	B
4	73%	C
5	91%	B
6	52%	B+
7	78%	D
8	64%	A
9	86%	B+
10	55%	A
11	68%	B+
12	71%	B
13	49%	C
14	89%	B
15	42%	B+
16	76%	C
17	61%	C
18	58%	B+
19	87%	A

64 | Page

	Course_Id	Course_Name	Exam_1_Score	Exam_2_Score	Exam_3_Score	%
0	HS-10	SSC	71	85	63	
1	HS-10	CBSE	88	96	90	
2	HS-10	SSC	63	75	72	
3	HS-10	CBSE	84	64	53	
4	HS-10	SSC	52	98	85	
5	HS-10	CBSE	78	81	90	
6	HS-10	SSC	60	78	69	
7	HS-10	CBSE	85	67	79	
8	HS-10	SSC	74	73	45	
9	HS-10	CBSE	91	59	58	
10	HS-10	SSC	50	43	87	
11	HS-10	CBSE	82	91	55	
12	HS-10	SSC	66	87	84	
13	HS-10	CBSE	76	58	67	
14	HS-10	SSC	49	60	48	
15	001-11-12	CBSE	92	79	76	
16	001-11-12	SSC	57	100	88	
17	001-11-12	CBSE	84	68	62	
18	001-11-12	SSC	42	63	100	
19	001-11-12	CBSE	72	61	95	
	Homework_1_Grade	Homework_2_Grade	Midterm_Exam_Score	Final_Exam_Score	%	
0	88	81	80	100		
1	82	88	100	100		
2	83	75	223	258		
3	76	80	199	400		
4	81	87	485	425		
5	79	78	637	245		
6	70	88	121	145		
7	85	72	159	234		
8	88	66	267	156		
9	81	71	179	700		
10	87	85	150	188		
11	80	88	168	196		
12	89	78	298	287		
13	75	63	168	180		
14	81	86	154	178		
15	85	84	137	154		
16	80	78	194	234		
17	86	81	158	180		
18	84	85	262	200		
19	85	82	200	200		

	Attendance_Record	Grades
0	82%	A
1	67%	C
2	45%	B
3	59%	B
4	73%	C
5	91%	B
6	52%	B+
7	78%	D
8	64%	A
9	86%	B+
10	55%	A
11	68%	B+
12	71%	B
13	49%	C
14	89%	B
15	42%	B+
16	76%	C
17	61%	C
18	58%	B+
19	87%	A

-----filna(method='bfill')-----					
	Student_id	Student_name	Date Of Birth	Gender	Parent_Guardian_Name
0	101	Nandu	2002-09-01	Female	John
1	102	Navya	2003-01-01	Female	Mary
2	103	Mythili	2002-01-31	Female	Michael
3	104	Pujitha	2004-05-18	Female	Jennifer
4	105	Yashaswi	2005-09-27	Female	David
5	106	Kavya	2002-04-19	Female	Lisa
6	107	Saritha	2007-02-11	Female	James
7	108	Murali	2008-08-31	Male	Sarah
8	109	Mohan	2009-06-25	Male	Robert
9	110	Das	2010-04-16	Male	Laura
10	111	Anju	2011-10-07	Female	William
11	112	Sravani	2012-01-19	Female	Emily
12	113	Shiny	2013-07-14	Female	Joseph
13	114	Mani	2014-09-03	Male	Jessica
14	115	Prani	2015-03-28	Female	Richard
15	116	Deepu	2016-06-12	Female	Kimberly
16	117	Vasu	2017-12-08	Female	Charles
17	118	Moulika	2018-08-22	Female	Rebecca
18	119	Vinaya	2019-02-02	Female	Thomas
19	120	Mahitha	2020-11-30	Female	Amanda
	Parent_Guardian_Phone	Enrollment_Date	Graducation_Date	Class_Level	
0	903456789	2023-04-13	2024-05-15	10th class	
1	807654321	2023-04-13	2024-05-15	10th class	
2	955123456	2023-04-13	2024-05-15	10th class	
3	754987321	2023-04-13	2024-05-15	10th class	
4	987654123	2023-04-13	2024-05-15	10th class	
5	823987456	2023-04-13	2024-05-15	10th class	
6	956321987	2023-04-13	2024-05-15	10th class	
7	987123456	2023-04-13	2024-05-15	10th class	
8	789654321	2023-04-13	2024-05-15	10th class	
9	821654987	2023-04-13	2024-05-15	10th class	
10	806789123	2023-04-13	2024-05-15	10th class	
11	789123456	2023-04-13	2024-05-15	10th class	
12	954123789	2023-04-13	2024-05-15	10th class	
13	911456987	2023-04-13	2024-05-15	10th class	
14	987321456	2023-04-13	2024-05-15	10th class	
15	954321789	2022-05-16	2024-04-15	Intermediate	
16	789456123	2022-05-16	2024-04-15	Intermediate	
17	973654987	2022-05-16	2024-04-15	Intermediate	
18	856987123	2022-05-16	2024-04-15	Intermediate	
19	987456321	2022-05-16	2024-04-15	Intermediate	

	Course_Id	Course_Name	Exam_1_Score	Exam_2_Score	Exam_3_Score	%
0	HS-10	SSC	71	85	63	
1	HS-10	CBSE	88	96	98	
2	HS-10	SSC	63	75	72	
3	HS-10	CBSE	94	64	53	
4	HS-10	SSC	52	90	85	
5	HS-10	CBSE	78	81	98	
6	HS-10	SSC	69	78	69	
7	HS-10	CBSE	85	67	79	
8	HS-10	SSC	74	73	45	
9	HS-10	CBSE	91	59	58	
10	HS-10	SSC	59	43	87	
11	HS-10	CBSE	82	91	55	
12	HS-10	SSC	66	87	84	
13	HS-10	CBSE	76	58	67	
14	HS-10	SSC	49	69	48	
15	BOI-11-12	CBSE	97	79	76	
16	BOI-11-12	SSC	57	100	88	
17	BOI-11-12	CBSE	84	98	52	
18	BOI-11-12	SSC	42	63	100	
19	BOI-11-12	CBSE	72	61	95	

	Homework_1_Grade	Homework_2_Grade	MidTerm_Exam_Score	Final_Exam_Score	%
0	84	81	80	100	
1	82	98	100	100	
2	83	75	123	158	
3	70	80	190	400	
4	83	87	405	425	
5	78	70	215	245	
6	70	98	131	145	
7	85	72	150	234	
8	88	46	207	356	
9	83	71	370	390	
10	97	85	170	189	
11	86	88	160	190	
12	88	79	230	267	
13	73	63	200	280	
14	81	86	188	178	
15	96	81	125	134	
16	84	70	184	234	
17	66	81	150	180	
18	84	75	202	288	
19	65	62	200	280	

	Attendance_Record	Grades
0	82X	A
1	67X	C
2	45X	B
3	59X	B
4	73X	C
5	91X	B
6	52X	B+
7	78X	D
8	64X	A
9	86X	B+
10	55X	A
11	68X	B+
12	71X	B
13	49X	C
14	89X	B
15	42X	B+
16	76X	C
17	61X	C
18	58X	B+
19	87X	A

8.2.3 WRONG FORMAT :

```
# Convert 'Date_Of_Birth', 'Enrollment_Date', 'Graduation_Date' to datetime
date_columns = ['Date_Of_Birth', 'Enrollment_Date', 'Graduation_Date']
df[date_columns] = df[date_columns].apply(pd.to_datetime)
# Handle any other cleaning steps (e.g., handle missing values)
# In this example, we assume no missing values
# Display the first few rows of the DataFrame
print(df.head(20))
```

OUTPUT :

	Date_Of_Birth	Enrollment_Date	Graduation_Date
0	2002-09-01	2023-04-13	2024-05-15
1	2003-01-01	2023-04-13	2024-05-15
2	2002-01-11	2023-04-13	2024-05-15
3	2004-05-18	2023-04-13	2024-05-15
4	2005-09-27	2023-04-13	2024-05-15
5	2002-04-19	2023-04-13	2024-05-15
6	2007-02-11	2023-04-13	2024-05-15
7	2008-08-31	2023-04-13	2024-05-15
8	2009-06-25	2023-04-13	2024-05-15
9	2010-04-16	2023-04-13	2024-05-15
10	2011-10-07	2023-04-13	2024-05-15
11	2012-01-10	2023-04-13	2024-05-15
12	2013-07-14	2023-04-13	2024-05-15
13	2014-09-03	2023-04-13	2024-05-15
14	2015-03-28	2023-04-13	2024-05-15
15	2016-06-12	2022-05-16	2024-04-15
16	2017-12-08	2022-05-16	2024-04-15
17	2018-08-22	2022-05-16	2024-04-15
18	2019-02-02	2022-05-16	2024-04-15
19	2020-11-30	2022-05-16	2024-04-15

8.2.4 REMOVE DUPLICATES:

```
print(" -----duplicated()-----")
print(df.duplicated())
print("\n")
# once we identified the duplicates then we need to dropitby using
print(" -----drop duplicates()-----")
df.drop_duplicates(inplace=True)
print("\n")
print(df)
```

OUTPUT :

```
-----duplicated()-----
0    False
1    False
2    False
3    False
4    False
5    False
6    False
7    False
8    False
9    False
10   False
11   False
12   False
13   False
14   False
15   False
16   False
17   False
18   False
19   False
dtype: bool
```

```
-----drop duplicates()-----
Student_id Student_name Date_Of_Birth Gender Parent_Guardian_Name \
0      101      Nandu  2002-09-01  Female      John
1      102      Navya  2003-01-01  Female      Mary
2      103      Mythili  2002-01-31  Female      Michael
3      104      Pujitha  2004-05-18  Female      Jennifer
4      105      Yashaswi  2005-09-27  Female      David
5      106      Kavya  2002-04-19  Female      Lisa
6      107      Saritha  2007-02-11  Female      James
7      108      Murali  2008-08-31  Male      Sarah
8      109      Mohan  2009-06-25  Male      Robert
9      110      Das  2010-04-16  Male      Laura
10     111      Anju  2011-10-07  Female      William
11     112      Sravan  2012-01-19  Female      Emily
12     113      Shiny  2013-07-14  Female      Joseph
13     114      Mani  2014-09-03  Male      Jessica
14     115      Prani  2015-03-28  Female      Richard
15     116      Deepu  2016-06-12  Female      Kimberly
16     117      Vasu  2017-12-08  Female      Charles
17     118      Moulika  2018-08-22  Female      Rebecca
18     119      Vinaya  2019-02-02  Female      Thomas
19     120      Mahitha  2020-11-30  Female      Anasida

Parent_Guardian_Phone Enrollment_Date Graduation_Date Class_Level \
0      983456789  2023-04-13  2024-05-15  10th class
1      807654321  2023-04-13  2024-05-15  10th class
2      955123456  2023-04-13  2024-05-15  10th class
3      754987321  2023-04-13  2024-05-15  10th class
4      987654321  2023-04-13  2024-05-15  10th class
5      823987456  2023-04-13  2024-05-15  10th class
6      956321987  2023-04-13  2024-05-15  10th class
7      987123456  2023-04-13  2024-05-15  10th class
8      789654321  2023-04-13  2024-05-15  10th class
9      821654987  2023-04-13  2024-05-15  10th class
10     806789123  2023-04-13  2024-05-15  10th class
11     789123456  2023-04-13  2024-05-15  10th class
12     954123789  2023-04-13  2024-05-15  10th class
13     911456987  2023-04-13  2024-05-15  10th class
14     987321456  2023-04-13  2024-05-15  10th class
15     954321789  2022-05-16  2024-04-15  Intermediate
16     789456123  2022-05-16  2024-04-15  Intermediate
17     873654987  2022-05-16  2024-04-15  Intermediate
18     856987123  2022-05-16  2024-04-15  Intermediate
19     987456321  2022-05-16  2024-04-15  Intermediate
```

```
Course_ID Course_Name Exam_1_Score Exam_2_Score Exam_3_Score \
0    HS-10 SSC 71 85 84
1    HS-10 SSC 88 90 90
2    HS-10 SSC 63 75 72
3    HS-10 SSC 90 80 83
4    HS-10 SSC 52 60 65
5    HS-10 SSC 78 81 80
6    HS-10 SSC 60 70 68
7    HS-10 SSC 85 87 79
8    HS-10 SSC 74 73 65
9    HS-10 SSC 91 93 88
10   HS-10 SSC 58 63 67
11   HS-10 SSC 82 91 85
12   HS-10 SSC 86 87 84
13   HS-10 SSC 70 80 87
14   HS-10 SSC 48 69 60
15   HS-10 SSC 87 79 76
16   HS-10 SSC 88 90 88
17   HS-10 SSC 57 68 66
18   HS-10 SSC 42 63 60
19   HS-10 SSC 72 83 85

Homework_1_Score Homework_2_Score MidTerm_Exam_Score Final_Exam_Score \
0      84      88      80      100
1      92      95      100      100
2      70      75      70      70
3      80      85      80      80
4      70      75      70      70
5      60      65      60      60
6      70      75      70      70
7      80      85      80      80
8      70      75      70      70
9      80      85      80      80
10     60      65      60      60
11     80      85      80      80
12     80      85      80      80
13     70      75      70      70
14     60      65      60      60
15     80      85      80      80
16     70      75      70      70
17     80      85      80      80
18     60      65      60      60
19     70      75      70      70
```

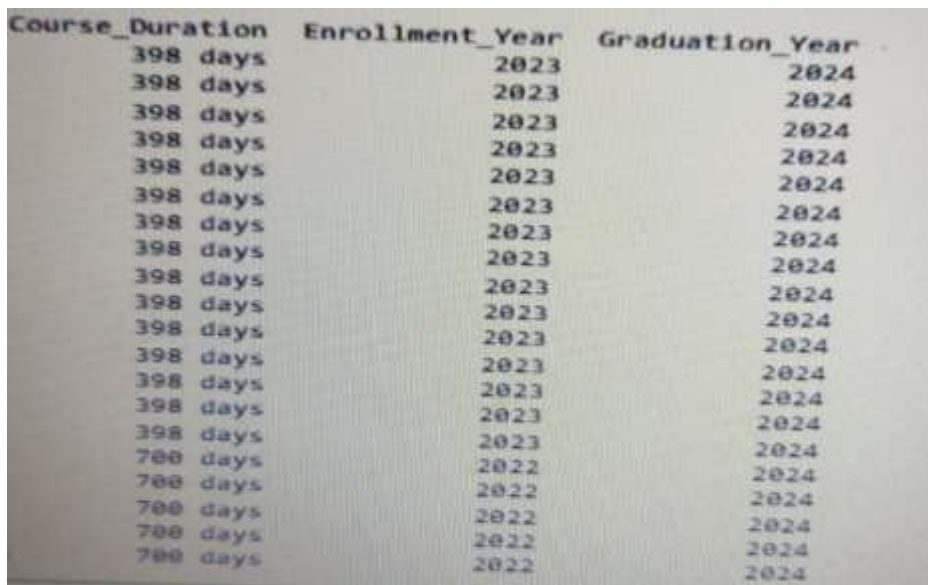
```
Attendance_Record Grades
0      82%      A
1      67%      C
2      45%      B
3      59%      B
4      73%      C
5      91%      B
6      52%      B+
7      78%      D
8      64%      A
9      86%      B+
10     55%      A
11     68%      B+
12     71%      B
13     49%      C
14     89%      B
15     42%      B+
16     76%      C
17     61%      C
18     58%      B+
19     87%      A
```

8.3 Feature Engineering :

as here we add features to dataset as here we taken age as per the joining and enrollment of the day.

```
import pandas as pd
df['Course_Duration'] = df['Graduation_Date'] - df['Enrollment_Date']
# Extract year from 'Enrollment_Date' and 'Graduation_Date'
df['Enrollment_Year'] = df['Enrollment_Date'].dt.year
df['Graduation_Year'] = df['Graduation_Date'].dt.year
# Display the first few rows of the DataFrame
print(df.head(20))
```

OUTPUT:



Course_Duration	Enrollment_Year	Graduation_Year
398 days	2023	2024
398 days	2023	2024
398 days	2023	2024
398 days	2023	2024
398 days	2023	2024
398 days	2023	2024
398 days	2023	2024
398 days	2023	2024
398 days	2023	2024
398 days	2023	2024
398 days	2023	2024
398 days	2023	2024
398 days	2023	2024
398 days	2023	2024
398 days	2023	2024
700 days	2022	2024
700 days	2022	2024
700 days	2022	2024
700 days	2022	2024
700 days	2022	2024

8.4 ANALYZING DATA (Scrutinizing the data):

In this phase we perform these by analysing the data and its values

- 1.Viewing Data
- 2.Info about Data
- 3.Data Munging
- 4.Data Filtering
- 5.Data Merging
- 6.Data Reshaping
- 7.Data Aggregation
- 8.Data Grouping

8.4.1 VIEWING DATA:

It is the process of view the format /structure of data Frame. And here some attributes we have to view the data they are:

1.describe()

2.head()

3.tail()

8.4.1.1 describe():

which describes in detail about the dataset

```
df.describe()
```

OUTPUT:

	Student_id	Parent_Guardian_Phone	Exam_1_Score	Exam_2_Score	Exam_3_Score	Homework_1_Grade	Homework_2_Grade	Midterm_Exam_Score	Final_Exam_Score
count	20.00000	2.000000e+01	20.00000	20.00000	20.00000	20.00000	20.00000	20.00000	20.00000
mean	110.50000	8.399155e+08	71.45000	73.15000	73.30000	78.00000	78.40000	210.45000	242.70000
std	5.91608	8.425308e+07	15.45954	15.39053	17.69953	10.06296	11.96222	92.95017	92.30105
min	101.00000	7.549873e+08	42.00000	43.00000	40.00000	60.00000	61.00000	80.00000	100.00000
25%	105.75000	8.074350e+08	62.00000	62.50000	61.00000	69.00000	70.00000	156.75000	186.25000
50%	110.50000	8.074569e+08	73.00000	71.50000	74.00000	80.50000	77.00000	182.00000	234.00000
75%	115.25000	8.606552e+08	84.25000	85.50000	87.25000	84.00000	86.25000	260.50000	289.25000
max	120.00000	9.876541e+08	97.00000	100.00000	100.00000	97.00000	98.00000	405.00000	425.00000

8.4.1.2 head():

which returns top of the values and here we can pass the value in the function of head() from where to where the data to get from top to bottom from specified index positions.

```
#to print range of values
```

```
# which prints the 10 values of staring as we have taken 20x20 dataset and for 100x100 dataset too we can take first 50 values it displays
```

```
df.head(10)
```

OUTPUT:

	Student_id	Student_name	Date_Of_Birth	Gender	Parent_Guardian_Name	Parent_Guardian_Phone	Enrollment_Date	Graduation_Date	Class_Level
0	101	Nandu	2002-09-01	Female	John	903460789	2023-04-13	2024-05-15	10th class
1	102	Navya	2003-01-01	Female	Mary	807054321	2023-04-13	2024-05-15	10th class
2	103	Mythili	2003-01-31	Female	Michael	956123456	2023-04-13	2024-05-15	10th class
3	104	Pujitha	2004-05-18	Female	Jennifer	794687321	2023-04-13	2024-05-15	10th class
4	105	Yashaswi	2005-09-27	Female	David	987054123	2023-04-13	2024-05-15	10th class
5	106	Kavya	2002-04-10	Female	Lisa	823087456	2023-04-13	2024-05-15	10th class
6	107	Santha	2007-02-11	Female	James	886321987	2023-04-13	2024-05-15	10th class
7	108	Murali	2008-08-31	Male	Sarah	987123456	2023-04-13	2024-05-15	10th class
8	109	Mohan	2009-06-25	Male	Robert	789054321	2023-04-13	2024-05-15	10th class
9	110	Dax	2010-04-18	Male	Laura	821054667	2023-04-13	2024-05-15	10th class

Course_Id	Course_Name	Exam_1_Score	Exam_2_Score	Exam_3_Score	Homework_1_Grade	Homework_2_Grade	Midterm_Exam_Score	Final_Exam_Score
HS-10	SSC	71	85	83	84	81	80	100
HS-10	CBSE	88	96	90	92	98	100	100
HS-10	SSC	83	75	72	83	75	100	100
HS-10	CBSE	84	84	83	70	90	223	250
HS-10	SSC	52	90	80	83	87	390	400
HS-10	CBSE	78	81	90	78	70	426	425
HS-10	SSC	89	70	59	79	88	218	240
HS-10	CBSE	85	87	79	88	72	139	140
HS-10	SSC	74	73	45	88	88	287	300
HS-10	CBSE	91	89	58	83	71	372	380

Attendance_Record	Grades	Course_Duration	Enrollment_Year	Graduation_Year	Result
82%	A	398 days	2023	2024	219
87%	C	398 days	2023	2024	282
45%	B	398 days	2023	2024	210
59%	B	398 days	2023	2024	211
73%	C	398 days	2023	2024	227
91%	B	398 days	2023	2024	249
52%	B+	398 days	2023	2024	208
78%	D	398 days	2023	2024	231
64%	A	398 days	2023	2024	192
86%	B+	398 days	2023	2024	208

8.4.1.3 tail():

which will give the output from bottom to top

```
df.tail(10) # which prints last 10 values from 20 x 20 dataset
```

OUTPUT:

```
In [29]: df.tail(10) # which prints last 10 values
```

```
Out[29]:
```

	Student_Id	Student_name	Date_Of_Birth	Gender	Parent_Guardian_Name	Parent_Guardian_Phone	Enrollment_Date	Graduation_Date	Class_Level
10	111	Anju	2011-10-07	Female	William	808789123	2023-04-13	2024-05-18	10th class
11	112	Srawani	2012-01-19	Female	Emily	789123456	2023-04-13	2024-05-18	10th class
12	113	Shrey	2013-07-14	Female	Joseph	954123789	2023-04-13	2024-05-18	10th class
13	114	Mari	2014-09-03	Male	Jessica	911456057	2023-04-13	2024-05-18	10th class
14	115	Pvoni	2015-03-28	Female	Richard	987321456	2023-04-13	2024-05-18	10th class
15	116	Dweepa	2016-06-12	Female	Kimberly	954321789	2023-05-18	2024-04-15	Intermediate
16	117	Vani	2017-12-08	Female	Charles	789456123	2023-05-18	2024-04-15	Intermediate
17	118	Mouka	2018-08-22	Female	Rebecca	973054087	2023-05-18	2024-04-15	Intermediate
18	119	Vinaya	2019-02-02	Female	Thomas	856887123	2023-05-18	2024-04-15	Intermediate
19	120	Mahira	2020-11-30	Female	Amahda	987456321	2023-05-18	2024-04-15	Intermediate

```
In [29]: df.tail(10) # which prints last 10 values
```

```
Out[29]:
```

Course_Id	Course_Name	Exam_1_Score	Exam_2_Score	Exam_3_Score	Homework_1_Grade	Homework_2_Grade	Midterm_Exam_Score	Final_Exam_Score
HS-10	SSC	59	83	87	87	85	179	188
HS-10	CBSE	82	97	88	89	98	180	190
HS-10	SSC	98	87	84	90	79	290	287
HS-10	CBSE	76	50	87	73	85	180	130
HS-10	SSC	49	86	40	81	86	188	170
SD-11-12	CBSE	87	79	79	85	87	110	124
SD-11-12	SSC	87	100	86	80	70	184	224
SD-11-12	CBSE	34	80	80	88	81	180	188
SD-11-12	SSC	42	83	100	84	79	280	280
SD-11-12	CBSE	72	81	88	88	80	280	230

Score	Final_Exam_Score	Attendance_Record	Grades	Course_Duration	Enrollment_Year	Graduation_Year	Result
170	189	55%	A	398 days	2023	2024	189
160	190	68%	B+	398 days	2023	2024	228
230	267	71%	B	398 days	2023	2024	237
160	189	49%	C	398 days	2023	2024	193
168	178	89%	B	398 days	2023	2024	188
115	134	42%	B+	700 days	2022	2024	262
194	234	76%	C	700 days	2022	2024	246
150	189	61%	C	700 days	2022	2024	206
262	290	68%	B+	700 days	2022	2024	205
260	289	87%	A	700 days	2022	2024	228

8.4.2 INFO ABOUT DATA :

`df.info()` # which give the detail info of the data.

OUTPUT:

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 20 entries, 0 to 19
Data columns (total 24 columns):
 #   Column                                Non-Null Count  Dtype
---  -
 0   Student_id                           20 non-null     int64
 1   Student_name                         20 non-null     object
 2   Date_Of_Birth                       20 non-null     datetime64[ns]
 3   Gender                              20 non-null     object
 4   Parent_Guardian_Name                20 non-null     object
 5   Parent_Guardian_Phone               20 non-null     int64
 6   Enrollment_Date                     20 non-null     datetime64[ns]
 7   Graducation_Date                   20 non-null     datetime64[ns]
 8   Class_Level                         20 non-null     object
 9   Course_Id                           20 non-null     object
10   Course_Name                         20 non-null     object
11   Exam_1_Score                       20 non-null     int64
12   Exam_2_Score                       20 non-null     int64
13   Exam_3_Score                       20 non-null     int64
14   Homework_1_Grade                   20 non-null     int64
15   Homework_2_Grade                   20 non-null     int64
16   Midterm_Exam_Score                 20 non-null     int64
17   Final_Exam_Score                   20 non-null     int64
18   Attendance_Record                  20 non-null     object
19   Grades                             20 non-null     object
20   Course_Duration                     20 non-null     timedelta64[ns]
21   Enrollment_Year                     20 non-null     int64
22   Graduation_Year                     20 non-null     int64
23   Result                             20 non-null     int64
dtypes: datetime64[ns](3), int64(12), object(8), timedelta64[ns](1)
memory usage: 4.4+ KB
```

8.4.3 Data Munging:

It is a process of Gathering all information regards Data Set. That is Collection of Data Set for Transforming Raw Data to another Format for performing better Operations & Easy to understand. "DATA MUNGING is also known as DATA WRANGLING."

8.4.4 Data Filtering :

It is a Process of getting Required Information from Dataset is known as Filtering. and in this method we can't perform operations correctly.

```
#Filtering
X=df['Result']>=250
print(X)
```

OUTPUT:

```
0      False
1       True
2      False
3      False
4      False
5      False
6      False
7      False
8      False
9      False
10     False
11     False
12     False
13     False
14     False
15       True
16     False
17     False
18     False
19     False
Name: Result, dtype: bool
```

8.4.5 DATA MERGING:

It is a process of Combining 2 datasets into a single dataset.

here we use the method:

SYNTAX: `merge(DataFrame1,DataFrame2,on="field")`

field nothing but id—index

```
#DATA MERGING
print(pd.merge(df,df,on='Student_id')) # here we are merging original and operated data by
adding course duration,result,,enrollment year,graduation year. And also it takes
```

OUTPUT:

	Student_Id	Student_name_x	Date_Of_Birth_x	Gender_x	Parent_Guardian_Name_x	\
0	101	Nandu	2002-09-01	Female	John	
1	102	Navya	2003-01-01	Female	Mary	
2	103	Mythili	2002-01-31	Female	Michael	
3	104	Pujitha	2004-05-18	Female	Jennifer	
4	105	Yashaswi	2005-09-27	Female	David	
5	106	Kavya	2002-04-19	Female	Lisa	
6	107	Saritha	2007-02-11	Female	James	
7	108	Murali	2008-08-31	Male	Sarah	
8	109	Mohan	2009-06-25	Male	Robert	
9	110	Das	2010-04-16	Male	Laura	
10	111	Anju	2011-10-07	Female	William	
11	112	Gravani	2012-01-19	Female	Emily	
12	113	Shiny	2013-07-14	Female	Joseph	
13	114	Mani	2014-09-03	Male	Jessica	
14	115	Prani	2015-03-28	Female	Richard	
15	116	Deepu	2016-06-12	Female	Kimberly	
16	117	Vasu	2017-12-08	Female	Charles	
17	118	Moulika	2018-08-22	Female	Rebecca	
18	119	Vinaya	2019-02-02	Female	Thomas	
19	120	Mahitha	2020-11-30	Female	Amanda	

	Parent_Guardian_Phone_x	Enrollment_Date_x	Graduation_Date_x	\
0	903456789	2023-04-13	2024-05-15	
1	807654321	2023-04-13	2024-05-15	
2	955123456	2023-04-13	2024-05-15	
3	754987321	2023-04-13	2024-05-15	
4	987654123	2023-04-13	2024-05-15	
5	823987456	2023-04-13	2024-05-15	
6	956321987	2023-04-13	2024-05-15	
7	987123456	2023-04-13	2024-05-15	
8	789654321	2023-04-13	2024-05-15	
9	821654987	2023-04-13	2024-05-15	
10	806789123	2023-04-13	2024-05-15	
11	789123456	2023-04-13	2024-05-15	
12	954123789	2023-04-13	2024-05-15	
13	911456987	2023-04-13	2024-05-15	
14	987321456	2023-04-13	2024-05-15	
15	954321789	2022-05-16	2024-04-15	
16	789456123	2022-05-16	2024-04-15	
17	973654987	2022-05-16	2024-04-15	
18	856987123	2022-05-16	2024-04-15	
19	987456321	2022-05-16	2024-04-15	

	Class_Level_x	Course_Id_x	Course_Name_x	Exam_1_Score_x	Exam_2_Score_x	\
0	10th class	HS-10	SSC	71	85	
1	10th class	HS-10	CBSE	88	96	
2	10th class	HS-10	SSC	63	75	
3	10th class	HS-10	CBSE	94	64	
4	10th class	HS-10	SSC	52	90	
5	10th class	HS-10	CBSE	78	81	
6	10th class	HS-10	SSC	69	70	
7	10th class	HS-10	CBSE	85	67	
8	10th class	HS-10	SSC	74	73	
9	10th class	HS-10	CBSE	91	59	
10	10th class	HS-10	SSC	59	43	
11	10th class	HS-10	CBSE	82	91	
12	10th class	HS-10	SSC	66	87	
13	10th class	HS-10	CBSE	76	50	
14	10th class	HS-10	SSC	49	69	
15	Intermediate	BOI-11-12	CBSE	97	79	
16	Intermediate	BOI-11-12	SSC	57	100	
17	Intermediate	BOI-11-12	CBSE	84	60	
18	Intermediate	BOI-11-12	SSC	42	63	
19	Intermediate	BOI-11-12	CBSE	72	61	

	Exam_3_Score_x	Homework_1_Grade_x	Homework_2_Grade_x	\
0	63	84	81	
1	98	92	98	
2	72	83	75	
3	53	70	80	
4	85	83	87	
5	90	78	70	
6	69	76	98	
7	79	65	72	
8	45	88	66	
9	58	83	71	
10	87	97	85	
11	55	86	98	
12	84	60	79	
13	67	73	63	
14	40	81	86	
15	76	66	91	
16	88	80	70	
17	62	66	61	
18	100	84	75	
19	95	65	62	

	Midterm_Exam_Score_x	Final_Exam_Score_x	Attendance_Record_x	Grades_x	
0	80	100	82%	A	
1	100	160	67%	C	
2	223	250	45%	B	
3	390	400	59%	B	
4	40%	425	73%	C	
5	215	245	91%	B	
6	131	145	52%	B+	
7	159	234	78%	D	
8	267	356	64%	A	
9	370	390	86%	B+	
10	170	189	55%	A	
11	160	190	68%	B+	
12	230	267	71%	B	
13	160	189	49%	C	
14	168	178	89%	B	
15	115	134	42%	B+	
16	194	234	76%	C	
17	150	189	61%	C	
18	262	290	58%	B+	
19	260	289	87%	A	

	Course_Duration_x	Enrollment_Year_x	Graduation_Year_x	Result_x	
0	398 days	2023	2024	219	
1	398 days	2023	2024	282	
2	398 days	2023	2024	210	
3	398 days	2023	2024	211	
4	398 days	2023	2024	227	
5	398 days	2023	2024	249	
6	398 days	2023	2024	208	
7	398 days	2023	2024	231	
8	398 days	2023	2024	192	
9	398 days	2023	2024	205	
10	398 days	2023	2024	189	
11	398 days	2023	2024	228	
12	398 days	2023	2024	237	
13	398 days	2023	2024	103	
14	398 days	2023	2024	158	
15	700 days	2022	2024	252	
16	700 days	2022	2024	245	
17	700 days	2022	2024	206	
18	700 days	2022	2024	205	
19	700 days	2022	2024	228	

8.4.6 DATA RESHAPING :

Data reshaping is restructuring a datasets layout to better suit analysis, often involving changes in rows, columns, and summarization.

- the reshaping process is being done using the melt() function.
- Pandas provides methods like melt() for unpivoting and pivot_table() for pivoting, enabling efficient data reshaping in Python.

```
#Data Reshaping
reshaped_df =
df.melt(id_vars=['Student_name','Student_id','Exam_1_Score'],
        value_vars=['Exam_2_Score','Exam_3_Score'])
reshaped_df = reshaped_df.reset_index(drop=True)
print(reshaped_df)
```

OUTPUT:

	Student_name	Student_id	Exam_1_Score	variable	value
0	Nandu	101	71	Exam_2_Score	85
1	Navya	102	88	Exam_2_Score	96
2	Mythili	103	63	Exam_2_Score	75
3	Pujitha	104	94	Exam_2_Score	64
4	Yashaswi	105	52	Exam_2_Score	90
5	Kavya	106	78	Exam_2_Score	81
6	Saritha	107	69	Exam_2_Score	70
7	Murali	108	85	Exam_2_Score	67

8	Mohan	109	74	Exam_2_Score	73
9	Das	110	91	Exam_2_Score	59
10	Anju	111	59	Exam_2_Score	43
11	Sravani	112	82	Exam_2_Score	91
12	Shiny	113	66	Exam_2_Score	87
13	Mani	114	76	Exam_2_Score	50
14	Prani	115	49	Exam_2_Score	69
15	Deepu	116	97	Exam_2_Score	79
16	Vasu	117	57	Exam_2_Score	100
17	Moulika	118	84	Exam_2_Score	60
18	Vinaya	119	42	Exam_2_Score	63
19	Mahitha	120	72	Exam_2_Score	61
20	Nandu	101	71	Exam_3_Score	63
21	Navya	102	88	Exam_3_Score	98
22	Mythili	103	63	Exam_3_Score	72
23	Pujitha	104	94	Exam_3_Score	53
24	Yashaswi	105	52	Exam_3_Score	85
25	Kavya	106	78	Exam_3_Score	90
26	Saritha	107	69	Exam_3_Score	69
27	Murali	108	85	Exam_3_Score	79
28	Mohan	109	74	Exam_3_Score	45
29	Das	110	91	Exam_3_Score	58
30	Anju	111	59	Exam_3_Score	87
31	Sravani	112	82	Exam_3_Score	55
32	Shiny	113	66	Exam_3_Score	84
33	Mani	114	76	Exam_3_Score	67
34	Prani	115	49	Exam_3_Score	40
35	Deepu	116	97	Exam_3_Score	76
36	Vasu	117	57	Exam_3_Score	88
37	Moulika	118	84	Exam_3_Score	62
38	Vinaya	119	42	Exam_3_Score	100
39	Mahitha	120	72	Exam_3_Score	95

8.4.7.DATA AGGREGATION:

It focus on join two dataframes,by using the concat() we can perform this operation.

SYNTAX : pd.concat([DataFrame1,DataFrame2])

```
#DATA AGGREGATION
pd.set_option('display.max_columns', None)
pd.set_option('display.max_rows', None)
print(pd.concat([df,df]))
```

OUTPUT:

	Student_id	Student_name	Date_Of_Birth	Gender	Parent_Guardian_Name	\
0	101.0	Nandu	2002-09-01	Female	John	
1	102.0	Navya	2003-01-01	Female	Mary	
2	103.0	Mythili	2002-01-31	Female	Michael	
3	104.0	Pujitha	2004-05-18	Female	Jennifer	
4	105.0	Vashaswi	2005-09-27	Female	David	
5	106.0	Kavya	2002-04-19	Female	Lisa	
6	107.0	Saritha	2007-02-11	Female	James	
7	108.0	Murali	2008-08-31	Male	Sarah	
8	109.0	Mohan	2009-06-25	Male	Robert	
9	110.0	Das	2010-04-16	Male	Laura	
10	111.0	Anju	2011-10-07	Female	William	
11	112.0	Sravani	2012-01-19	Female	Emily	
12	113.0	Shiny	2013-07-14	Female	Joseph	
13	114.0	Mani	2014-09-03	Male	Jessica	
14	115.0	Prani	2015-03-28	Female	Richard	
15	116.0	Deepu	2016-06-12	Female	Kimberly	
16	117.0	Vasu	2017-12-08	Female	Charles	
17	118.0	Moulika	2018-08-22	Female	Rebecca	
18	119.0	Vinaya	2019-02-02	Female	Thomas	
19	120.0	Mahitha	2020-11-30	Female	Amanda	
0	101.0	Nandu	2002-09-01	Female	John	
1	102.0	Navya	2003-01-01	Female	Mary	
2	103.0	Mythili	2002-01-31	Female	Michael	
3	104.0	Pujitha	2004-05-18	NaN	Jennifer	
4	105.0	Vashaswi	2005-09-27	Female	David	
5	NaN	Kavya	2002-04-19	Female	Lisa	
6	107.0	Saritha	2007-02-11	Female	James	
7	108.0	Murali	2008-08-31	Male	Sarah	
8	109.0	Mohan	2009-06-25	Male	Robert	
9	110.0	Das	2010-04-16	Male	Laura	
10	111.0	Anju	2011-10-07	Female	William	
11	112.0	Sravani	2012-01-19	Female	Emily	
12	113.0	Shiny	2013-07-14	Female	Joseph	
13	114.0	NaN	2014-09-03	Male	Jessica	
14	115.0	Prani	2015-03-28	Female	Richard	
15	116.0	Deepu	2016-06-12	Female	Kimberly	
16	117.0	Vasu	2017-12-08	Female	Charles	
17	118.0	Moulika	2018-08-22	Female	Rebecca	
18	119.0	Vinaya	2019-02-02	Female	Thomas	
19	120.0	Mahitha	2020-11-30	Female	Amanda	
20	121.0	Vicky	2001-04-05	Male	Daniel	

	Parent_Guardian_Phone	Enrollment_Date	Graduation_Date	Class_Level	\
0	005456780	2023-04-13	2024-05-15	10th class	
1	007654321	2023-04-13	2024-05-15	10th class	
2	055123456	2023-04-13	2024-05-15	10th class	
3	754987321	2023-04-13	2024-05-15	10th class	
4	087654321	2023-04-13	2024-05-15	10th class	
5	823987456	2023-04-13	2024-05-15	10th class	
6	056321987	2023-04-13	2024-05-15	10th class	
7	087123456	2023-04-13	2024-05-15	10th class	
8	789654321	2023-04-13	2024-05-15	10th class	
9	821654987	2023-04-13	2024-05-15	10th class	
10	006789123	2023-04-13	2024-05-15	10th class	
11	789123456	2023-04-13	2024-05-15	10th class	
12	054123789	2023-04-13	2024-05-15	10th class	
13	011456987	2023-04-13	2024-05-15	10th class	
14	087321456	2023-04-13	2024-05-15	10th class	
15	054321789	2022-05-16	2024-04-15	Intermediate	
16	789456123	2022-05-16	2024-04-15	Intermediate	
17	073654987	2022-05-16	2024-04-15	Intermediate	
18	056987123	2022-05-16	2024-04-15	Intermediate	
19	087456321	2022-05-16	2024-04-15	Intermediate	
0	003456780	2023-04-13	2024-05-15	10th class	
1	007654321	2023-04-13	2024-05-15	10th class	
2	055123456	2023-04-13	2024-05-15	10th class	
3	754987321	2023-04-13	2024-05-15	10th class	
4	087654321	2023-04-13	2024-05-15	10th class	
5	823987456	2023-04-13	2024-05-15	10th class	
6	056321987	2023-04-13	2024-05-15	10th class	
7	087123456	2023-04-13	2024-05-15	10th class	
8	789654321	2023-04-13	2024-05-15	10th class	
9	821654987	2023-04-13	2024-05-15	10th class	
10	006789123	2023-04-13	2024-05-15	10th class	
11	789123456	2023-04-13	2024-05-15	10th class	
12	054123789	2023-04-13	2024-05-15	10th class	
13	011456987	2023-04-13	2024-05-15	10th class	
14	087321456	2023-04-13	2024-05-15	10th class	
15	054321789	2022-05-16	2024-04-15	Intermediate	
16	789456123	2022-05-16	2024-04-15	Intermediate	
17	073654987	2022-05-16	2024-04-15	Intermediate	
18	056987123	2022-05-16	2024-04-15	Intermediate	
19	087456321	2022-05-16	2024-04-15	Intermediate	
20	789321456	2022-05-16	2024-04-15	Intermediate	

	Course_Id	Course_Name	Exam_1_Score	Exam_2_Score	Exam_3_Score	\
0	HS-10	SSC	71	85	63.0	
1	HS-10	CBSE	88	96	98.0	
2	HS-10	SSC	63	75	72.0	
3	HS-10	CBSE	94	64	53.0	
4	HS-10	SSC	52	90	85.0	
5	HS-10	CBSE	78	81	90.0	
6	HS-10	SSC	69	70	69.0	
7	HS-10	CBSE	85	67	79.0	
8	HS-10	SSC	74	73	45.0	
9	HS-10	CBSE	91	59	58.0	
10	HS-10	SSC	59	43	87.0	
11	HS-10	CBSE	82	91	55.0	
12	HS-10	SSC	66	87	84.0	
13	HS-10	CBSE	76	50	67.0	
14	HS-10	SSC	49	69	40.0	
15	BOI-11-12	CBSE	97	79	76.0	
16	BOI-11-12	SSC	57	100	88.0	
17	BOI-11-12	CBSE	84	60	62.0	
18	BOI-11-12	SSC	42	63	100.0	
19	BOI-11-12	CBSE	72	61	95.0	
0	HS-10	SSC	71	85	63.0	
1	HS-10	CBSE	88	96	98.0	
2	HS-10	SSC	63	75	72.0	
3	HS-10	CBSE	94	64	53.0	
4	HS-10	SSC	52	90	85.0	
5	HS-10	CBSE	78	81	90.0	
6	HS-10	SSC	69	70	69.0	
7	HS-10	CBSE	85	67	79.0	
8	HS-10	SSC	74	73	45.0	
9	HS-10	CBSE	91	59	58.0	
10	HS-10	SSC	59	43	87.0	
11	HS-10	CBSE	82	91	55.0	
12	HS-10	SSC	66	87	84.0	
13	HS-10	CBSE	76	50	67.0	
14	HS-10	SSC	49	69	40.0	
15	BOI-11-12	CBSE	97	79	76.0	
16	BOI-11-12	SSC	57	100	88.0	
17	BOI-11-12	CBSE	84	60	62.0	
18	BOI-11-12	SSC	42	63	100.0	
19	BOI-11-12	CBSE	72	61	95.0	
20	BOI-11-12	SSC	60	80	NaN	

	Homework_1_Grade	Homework_2_Grade	Midterm_Exam_Score	Final_Exam_Score	\
0	84	81	80.0	100.0	
1	92	98	100.0	160.0	
2	83	75	223.0	250.0	
3	70	80	390.0	400.0	
4	83	87	405.0	425.0	
5	78	70	215.0	245.0	
6	76	98	131.0	145.0	
7	65	72	159.0	234.0	
8	88	66	267.0	356.0	
9	83	71	370.0	390.0	
10	97	85	170.0	189.0	
11	86	98	160.0	190.0	
12	60	79	230.0	267.0	
13	73	63	160.0	189.0	
14	81	86	168.0	178.0	
15	66	91	115.0	134.0	
16	80	70	194.0	234.0	
17	66	61	150.0	189.0	
18	84	75	262.0	290.0	
19	65	62	260.0	289.0	
0	84	81	NaN	NaN	
1	92	98	NaN	NaN	
2	83	75	NaN	NaN	
3	70	80	NaN	NaN	
4	83	87	NaN	NaN	
5	78	70	NaN	NaN	
6	76	98	NaN	NaN	
7	65	72	NaN	NaN	
8	88	66	NaN	NaN	
9	83	71	NaN	NaN	
10	97	85	NaN	NaN	
11	86	98	NaN	NaN	
12	60	79	NaN	NaN	
13	73	63	NaN	NaN	
14	81	86	NaN	NaN	
15	66	91	NaN	NaN	
16	80	70	NaN	NaN	
17	66	61	NaN	NaN	
18	84	75	NaN	NaN	
19	65	62	NaN	NaN	
20	69	60	NaN	NaN	

Attendance_Record Grades Parent_Teacher_Meeting_Date \				Library_Books_Checked_Out	
0	82%	A	NaN	0	NaN
1	67%	C	NaN	1	NaN
2	45%	B	NaN	2	NaN
3	59%	B	NaN	3	NaN
4	73%	C	NaN	4	NaN
5	91%	B	NaN	5	NaN
6	52%	B+	NaN	6	NaN
7	78%	D	NaN	7	NaN
8	64%	A	NaN	8	NaN
9	86%	B+	NaN	9	NaN
10	55%	A	NaN	10	NaN
11	68%	B+	NaN	11	NaN
12	71%	B	NaN	12	NaN
13	49%	C	NaN	13	NaN
14	89%	B	NaN	14	NaN
15	42%	B+	NaN	15	NaN
16	76%	C	NaN	16	NaN
17	61%	C	NaN	17	NaN
18	58%	B+	NaN	18	NaN
19	87%	A	NaN	19	NaN
0	82%	A	2023-06-22	0	2023-01-11
1	67%	C	2023-05-08	1	2023-12-12
2	45%	B	2023-06-11	2	2023-11-25
3	59%	B	2023-04-27	3	2023-12-03
4	73%	C	2023-05-03	4	2023-02-28
5	91%	B	2022-06-19	5	2023-01-18
6	52%	B+	2022-05-22	6	2023-12-10
7	78%	D	2022-05-29	7	2023-12-15
8	64%	A	2022-06-12	8	2023-07-17
9	86%	B+	2022-06-05	9	2023-04-19
10	55%	A	2022-07-17	10	2023-06-22
11	68%	B+	2022-07-24	11	2023-07-24
12	71%	B	2022-07-31	12	2023-11-10
13	49%	C	2022-08-07	13	2023-03-30
14	89%	B	2022-08-14	14	2023-02-17
15	42%	B+	2022-08-21	15	2023-01-07
16	76%	C	2022-08-28	16	2023-03-25
17	61%	C	2022-09-04	17	2023-07-11
18	58%	B+	2022-09-11	18	2023-04-22
19	87%	B	2022-09-18	19	2023-11-01
20	43%	A	2022-09-25	20	2023-04-19

8.4.8 DATA GROUPING:

It is a Process of Make a Group based on some condition from Data in DataSet.
here we use the method:

SYNTAX: pd.groupby('Field_Name').

```
#Data Grouping
import pandas as pd
pd.set_option('display.max_columns', None)
pd.set_option('display.max_rows', None)
df['Result']=df['Exam_1_Score']+df['Exam_2_Score']
print(df)
x=df.groupby('Grades')
print(x)
```

OUTPUT:

	Student_id	Student_name	Date_Of_Birth	Gender	Parent_Guardian_Name	\
0	101	Nandu	2002-09-01	Female	John	
1	102	Navya	2003-01-01	Female	Mary	
2	103	Mythili	2002-01-31	Female	Michael	
3	104	Pujitha	2004-05-18	Female	Jennifer	
4	105	Yashaswi	2005-09-27	Female	David	
5	106	Kavya	2002-04-19	Female	Lisa	
6	107	Saritha	2007-02-11	Female	James	
7	108	Murali	2008-08-31	Male	Sarah	
8	109	Mohan	2009-06-25	Male	Robert	
9	110	Das	2010-04-16	Male	Laura	
10	111	Anju	2011-10-07	Female	William	
11	112	Sravani	2012-01-19	Female	Emily	
12	113	Shiny	2013-07-14	Female	Joseph	
13	114	Mani	2014-09-03	Male	Jessica	
14	115	Prani	2015-03-28	Female	Richard	
15	116	Deepu	2016-06-12	Female	Kimberly	
16	117	Vasu	2017-12-08	Female	Charles	
17	118	Moulika	2018-08-22	Female	Rebecca	
18	119	Vinaya	2019-02-02	Female	Thomas	
19	120	Mahitha	2020-11-30	Female	Aranda	

	Parent_Guardian_Phone	Enrollment_Date	Graduation_Date	Class_Level	\
0	983456789	2023-04-13	2024-05-15	10th class	
1	807654321	2023-04-13	2024-05-15	10th class	
2	955123456	2023-04-13	2024-05-15	10th class	
3	754987321	2023-04-13	2024-05-15	10th class	
4	987654123	2023-04-13	2024-05-15	10th class	
5	823987456	2023-04-13	2024-05-15	10th class	
6	956321987	2023-04-13	2024-05-15	10th class	
7	987123456	2023-04-13	2024-05-15	10th class	
8	789654321	2023-04-13	2024-05-15	10th class	
9	821654987	2023-04-13	2024-05-15	10th class	
10	806789123	2023-04-13	2024-05-15	10th class	
11	789123456	2023-04-13	2024-05-15	10th class	
12	954123789	2023-04-13	2024-05-15	10th class	
13	911456987	2023-04-13	2024-05-15	10th class	
14	987321456	2023-04-13	2024-05-15	10th class	
15	954321789	2022-05-16	2024-04-15	Intermediate	
16	789456123	2022-05-16	2024-04-15	Intermediate	
17	973654987	2022-05-16	2024-04-15	Intermediate	
18	856987123	2022-05-16	2024-04-15	Intermediate	
19	987456321	2022-05-16	2024-04-15	Intermediate	

	Course_Id	Course_Name	Exam_1_Score	Exam_2_Score	Exam_3_Score	\
0	HS-10	SSC	71	85	63	
1	HS-10	CBSE	88	96	98	
2	HS-10	SSC	63	75	72	
3	HS-10	CBSE	94	64	53	
4	HS-10	SSC	52	90	85	
5	HS-10	CBSE	78	81	90	
6	HS-10	SSC	69	70	69	
7	HS-10	CBSE	85	67	79	
8	HS-10	SSC	74	73	45	
9	HS-10	CBSE	91	59	58	
10	HS-10	SSC	59	43	87	
11	HS-10	CBSE	82	91	55	
12	HS-10	SSC	66	87	84	
13	HS-10	CBSE	76	50	67	
14	HS-10	SSC	49	69	40	
15	BOI-11-12	CBSE	97	79	76	
16	BOI-11-12	SSC	57	100	88	
17	BOI-11-12	CBSE	84	60	62	
18	BOI-11-12	SSC	42	63	100	
19	BOI-11-12	CBSE	72	61	95	

	Homework_1_Grade	Homework_2_Grade	Midterm_Exam_Score	Final_Exam_Score	\
0	84	81	80	100	
1	92	98	100	160	
2	83	75	223	250	
3	70	80	390	400	
4	83	87	405	425	
5	78	70	215	245	
6	76	98	131	145	
7	65	72	159	234	
8	88	66	267	356	
9	83	71	370	390	
10	97	85	170	189	
11	86	98	160	190	
12	60	79	230	267	
13	73	63	160	189	
14	81	86	168	178	
15	66	91	115	134	
16	80	70	194	234	
17	66	61	150	189	
18	84	75	262	290	
19	65	62	260	289	

	Attendance_Record	Grades	Result
0	82%	A	156
1	67%	C	184
2	45%	B	138
3	59%	B	158
4	73%	C	142
5	91%	B	159
6	52%	B+	139
7	78%	D	152
8	64%	A	147
9	86%	B+	150
10	55%	A	102
11	68%	B+	173
12	71%	D	153
13	49%	C	126
14	89%	B	118
15	42%	B+	176
16	76%	C	157
17	61%	C	144
18	58%	B+	105
19	87%	A	133

[20 rows x 21 columns]

8.5 Statistical Analysis :

PANDAS COMPUTATIONAL TOOLS FOR STATISTICAL COMPUTATION OR MATHEMATICAL OPERATIONS --A computation is a process or completing a task by using many operations mathematically or statistically.

we have top 5 computational tools for Statistics

1.**min()** ---->which returns minimum value in a column.

2.**max()** ---->which returns maximum value in a column.

3.**rank ()** ----> By using this method we can rank all the data Members in a column in a Ascending order if those are in Numerics, Come to Alphabets. It follows Alphabetic Order.

4.**corr()** --co-relation (It is a Relation between two Column Data Members in a Dataset).

5.**cov()** ->covariance

```
#MATHEMATICAL & STATISTICAL OPERATIONS
#Result
df['Result']=df['Exam_1_Score']+df['Exam_2_Score']+df['Exam_3_Score']
print(df)
print('*****')
#1.min
min=df['Result'].min()
print(" Minimum Value : ",min)
print('*****')
max=df['Result'].max() # 2. max()
print(" Maximum Value : ",max)
print('*****')
```


OUTPUT :

```
      Result
0      219
1      282
2      210
3      211
4      227
5      249
6      208
7      231
8      192
9      208
10     189
11     228
12     237
13     193
14     158
15     252
16     245
17     206
18     205
19     228
*****
Minimum Value : 158
*****
Maximum Value : 282
*****
```

```
3. rank()
print(df['Result'].rank())
#prints rank as per the obtained result of marks
#print('*****')
#print(df.rank())
```

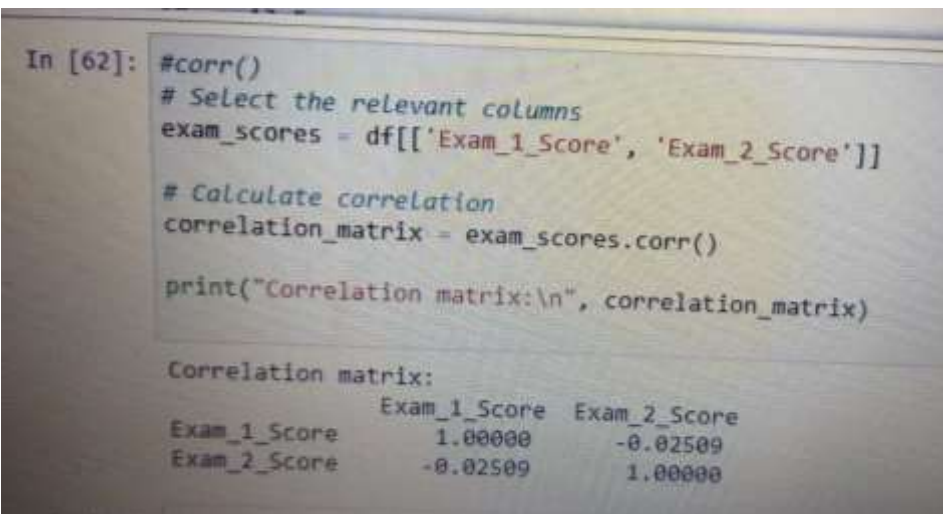
Output :

```
      rank
0    11.0
1    20.0
2     9.0
3    10.0
4    12.0
5    18.0
6     7.5
7    15.0
8     3.0
9     7.5
10     2.0
11    13.5
12    16.0
13     4.0
14     1.0
```

15 19.0
16 17.0
17 6.0
18 5.0
19 13.5

Name: Result, dtype: float64

8.5.4. CORRELATION():



```
import numpy as np
# Exam_1_Score and Exam_2_Score as NumPy arrays
exam_1_scores = np.array([71, 88, 63, 94, 52, 78, 69, 85, 74, 91, 59, 82, 66, 76, 49, 97, 57, 84, 42, 72])
exam_2_scores = np.array([85, 96, 75, 64, 90, 81, 70, 67, 73, 59, 43, 91, 87, 50, 69, 79, 100, 60, 63, 61])
# Calculate the sums
sum_x = np.sum(exam_1_scores)
sum_y = np.sum(exam_2_scores)
# Calculate the sum of the product of x and y
sum_xy = np.sum(exam_1_scores * exam_2_scores)
# Calculate the sum of squares for x and y
sum_x_squared = np.sum(exam_1_scores ** 2)
sum_y_squared = np.sum(exam_2_scores ** 2)
# Square of the sum of Exam_1_Score and Exam_2_Score
sum_x_squared1 = sum_x ** 2
sum_y_squared2 = sum_y ** 2
print("Sum of Exam_1_Score (sum x):", sum_x)
print("Sum of Exam_2_Score (sum y):", sum_y)
print("Sum of the product of Exam_1_Score and Exam_2_Score (sum xy):", sum_xy)
print("Sum of squares of Exam_2_Score (sum (x^2)):", sum_x_squared)
print("Sum of squares of Exam_3_Score (sum (y^2)):", sum_y_squared)
print("Square of the sum of Exam_2_Score ([sum(x)]^2):", sum_x_squared1)
print("Square of the sum of Exam_3_Score ([sum(y)]^2):", sum_y_squared2)
```

```

Sum of Exam_1_Score (sum x): 1449
Sum of Exam_2_Score (sum y): 1463
Sum of the product of Exam_1_Score and Exam_2_Score (sum xy): 105881
Sum of squares of Exam_2_Score (sum (x^2)): 109521
Sum of squares of Exam_3_Score (sum (y^2)): 111513
Square of the sum of Exam_2_Score ([sum(x)]^2): 2099601
Square of the sum of Exam_3_Score ([sum(y)]^2): 2140369

```

MANUAL CALCULATION OUTPUT:

$$\begin{aligned}
 r &= \frac{n(\sum xy) - (\sum x)(\sum y)}{\sqrt{[n\sum x^2 - (\sum x)^2][n\sum y^2 - (\sum y)^2]}} \\
 n &= 20 \\
 &= \frac{20(105881) - (1449)(1463)}{\sqrt{[20(109521) - (1449)^2][20(111513) - (1463)^2]}} \\
 &= \frac{2117620 - 2119887}{\sqrt{(2190420 - 2099601)(2230260 - 2140369)}} \\
 &= \frac{-2267}{\sqrt{(90819)(89891)}} \\
 &= \frac{-2267}{\sqrt{8163810729}} \\
 &= \frac{-2267}{90353.8086} \\
 \text{corr)} [r] &= -0.02509025392
 \end{aligned}$$

(weak negative correlation)

as we observe in correlation matrix between these 2 columns as we get output as $r=0.02509$ (negative correlation) and we get same result as in manual calculation too.

Case-2 :

```

In [66]: # Select the relevant columns
exam_scores = df[['Exam_2_Score', 'Exam_3_Score']]

# Calculate correlation
correlation_matrix = exam_scores.corr()

print("Correlation matrix:\n", correlation_matrix)

Correlation matrix:
          Exam_2_Score  Exam_3_Score
Exam_2_Score      1.000000      0.168224
Exam_3_Score      0.168224      1.000000

```

```

# Exam_2_Score and Exam_3_Score as NumPy arrays
exam_2_scores = df['Exam_2_Score'].values
exam_3_scores = df['Exam_3_Score'].values
# Calculate the sums
sum_x = np.sum(exam_2_scores)
sum_y = np.sum(exam_3_scores)
sum_xy = np.sum(exam_2_scores * exam_3_scores)
sum_x_squared = np.sum(exam_2_scores**2)
sum_y_squared = np.sum(exam_3_scores**2)
sum_x_squared1 = sum_x**2
sum_y_squared2 = sum_y**2
print("Sum of Exam_2_Score (sum x):", sum_x)
print("Sum of Exam_3_Score (sum y):", sum_y)
print("Sum of the product of Exam_2_Score and Exam_3_Score (sum (xy)):", sum_xy)
print("Sum of squares of Exam_2_Score (sum (x^2)):", sum_x_squared)
print("Sum of squares of Exam_3_Score (sum (y^2)):", sum_y_squared)
print("Square of the sum of Exam_2_Score ([sum(x)]^2):", sum_x_squared1)
print("Square of the sum of Exam_3_Score ([sum(y)]^2):", sum_y_squared2)

```

Output to this code :

```

Sum of Exam_2_Score (sum x): 1463
Sum of Exam_3_Score (sum y): 1466
Sum of the product of Exam_2_Score and Exam_3_Score (sum (xy)): 108108
Sum of squares of Exam_2_Score (sum (x^2)): 111513
Sum of squares of Exam_3_Score (sum (y^2)): 113410
Square of the sum of Exam_2_Score ([sum(x)]^2): 2140369
Square of the sum of Exam_3_Score ([sum(y)]^2): 2149156

```

MANUAL CALCULATION OUTPUT:

$$\begin{aligned}
 r &= \frac{20(108108) - (1463)(1466)}{\sqrt{[20(111513) - (2140369)][20(113410) - (2149156)]}} \\
 &= \frac{2162180 - 2144758}{\sqrt{(2230260 - 2140369)(2268200 - 2149156)}} \\
 &= \frac{17402}{\sqrt{(89891)(119044)}} \\
 &= \frac{17402}{103445.5615} \\
 \boxed{r = 0.1682237473} & \quad (\text{positive correlation})
 \end{aligned}$$

as we observe in correlation matrix between these 2 columns as we get output as $r = 0.1682$ (positive correlation) and we get same result as in manual calculation too.

CASE-3:

```
In [1]: import pandas as pd

# Given data
data = {
    'Final_Exam_Score': pd.Series([100, 160, 250, 400, 425, 245, 145, 234, 356, 390, 189, 190, 267, 189, 178, 134, 234, 189,
                                   290, 289]),
    'Midterm_Exam_Score': pd.Series([80, 100, 223, 390, 405, 215, 131, 159, 267, 370, 170, 160, 230, 160, 168, 115, 194, 150, 262, 260])
}

# Creating DataFrame
df = pd.DataFrame(data)

# Calculating correlation
correlation = df['Final_Exam_Score'].corr(df['Midterm_Exam_Score'])
print("Correlation between 'Final_Exam_Score' and 'Midterm_Exam_Score':", correlation)

Correlation between 'Final_Exam_Score' and 'Midterm_Exam_Score': 0.9750327723523433
```

Code :

```
In [1]: import pandas as pd

data = {
    'Final_Exam_Score': pd.Series([100, 160, 250, 400, 425, 245, 145, 234, 356, 390, 189, 190, 267, 189, 178, 134, 234, 189, 290, 289]),
    'Midterm_Exam_Score': pd.Series([80, 100, 223, 390, 405, 215, 131, 159, 267, 370, 170, 160, 230, 160, 168, 115, 194, 150, 262, 260])
}

# Calculate the values you mentioned for the provided dataset
sum_x = data['Final_Exam_Score'].sum()
sum_y = data['Midterm_Exam_Score'].sum()
sum_xy = (data['Final_Exam_Score'] * data['Midterm_Exam_Score']).sum()
sum_x_squared = (data['Final_Exam_Score'] ** 2).sum()
sum_y_squared = (data['Midterm_Exam_Score'] ** 2).sum()
sum_x_squared_total = sum_x ** 2
sum_y_squared_total = sum_y ** 2

print("Sum of x (Final_Exam_Score):", sum_x)
print("Sum of y (Midterm_Exam_Score):", sum_y)
print("Sum of xy:", sum_xy)
print("Sum of x^2:", sum_x_squared)
print("Sum of y^2:", sum_y_squared)
print("[Sum(x)]^2:", sum_x_squared_total)
print("[Sum(y)]^2:", sum_y_squared_total)

Sum of x (Final_Exam_Score): 4854
Sum of y (Midterm_Exam_Score): 4209
Sum of xy: 1180463
Sum of x^2: 1339936
Sum of y^2: 1049939
[Sum(x)]^2: 23561316
[Sum(y)]^2: 17715681
```

MANUAL CALCULATION OF IT OUTPUT:

$$\begin{aligned}
 \text{Corr1:} \\
 r &= \frac{n(\sum xy) - (\sum x)(\sum y)}{\sqrt{[n(\sum x^2) - (\sum x)^2][n(\sum y^2) - (\sum y)^2]}} \\
 &= \frac{20(1180463) - (4854)(4209)}{\sqrt{(20(1379936) - (23561316))(20(1049939) - (1771560))}} \\
 &= \frac{23609260 - 20430486}{\sqrt{26798720 - 23561316}(20998780 - 1771560)} \\
 &= \frac{3178774}{\sqrt{(3237404)(3283099)}} \\
 &= \frac{3178774}{326171.443} \\
 &= 0.9750327722
 \end{aligned}$$

Which gives the highly positive correlation .

8.5.5 COVARIANCE:

```

In [75]: #cov() 3
# Provided data
data = {
    'Exam_1_Score': pd.Series([71, 88, 63, 94, 52, 78, 69, 85, 74, 91, 59, 82, 66, 76, 49, 97, 57, 84, 42, 72, 68]),
    'Exam_2_Score': pd.Series([85, 96, 75, 64, 90, 81, 70, 67, 73, 59, 43, 91, 87, 50, 69, 79, 100, 60, 63, 61, 88])
}

# Create a DataFrame
df = pd.DataFrame(data)

# Calculate covariance
covariance = df['Exam_1_Score'].cov(df['Exam_2_Score'])

print("Covariance between Exam_1_Score and Exam_2_Score:")
print(covariance)

Covariance between Exam_1_Score and Exam_2_Score:
-9.728571428571435

```

```

In [76]: import pandas as pd

# Provided data
data = {
    'Exam_1_Score': pd.Series([71, 88, 63, 94, 52, 78, 69, 85, 74, 91, 59, 82, 66, 76, 49, 97, 57, 84, 42, 72, 68]),
    'Exam_2_Score': pd.Series([85, 96, 75, 64, 90, 81, 70, 67, 73, 59, 43, 91, 87, 50, 69, 75, 100, 60, 63, 61, 80])
}

# Create a DataFrame
df = pd.DataFrame(data)

# Mean of X (Exam_1_Score)
mean_X = df['Exam_1_Score'].mean()

# Mean of Y (Exam_2_Score)
mean_Y = df['Exam_2_Score'].mean()

# Deviation from mean for Exam_1_Score
deviation_X = df['Exam_1_Score'] - mean_X

# Deviation from mean for Exam_2_Score
deviation_Y = df['Exam_2_Score'] - mean_Y

# (Xi - Mean of X) * (Yi - Mean of Y)
product_of_deviations = deviation_X * deviation_Y

# Sum of (Xi - Mean of X) * (Yi - Mean of Y)
sum_of_products = product_of_deviations.sum()

print("Mean of X (Exam_1_Score):", mean_X)
print("Mean of Y (Exam_2_Score):", mean_Y)

print("Sum of (Xi - Mean of X) * (Yi - Mean of Y):", sum_of_products)

Mean of X (Exam_1_Score): 71.85714285714286
Mean of Y (Exam_2_Score): 73.47619047619048
Sum of (Xi - Mean of X) * (Yi - Mean of Y): -194.57142857142856

```

MANUAL CALCULATION OF IT OUTPUT:

$$\begin{aligned}
 \text{COV} &= \frac{\sum (x_i - \bar{x})(y_i - \bar{y})}{N-1} \\
 n &= 20 \\
 &= \frac{-194.5714}{19} \\
 &= -10.240 \quad (\text{approx}).
 \end{aligned}$$

the negative covariance suggests a negative relationship between the exam scores.

Case -2 :

```
In [72]: #cov() 2
import numpy as np

# Exam_2_Score and Exam_3_Score data
exam_2_scores = np.array([85, 96, 75, 64, 90, 81, 70, 67, 73, 59, 43, 91, 87, 50, 69, 79, 100, 60, 63, 61])
exam_3_scores = np.array([63, 98, 72, 53, 85, 90, 69, 79, 45, 58, 87, 55, 84, 67, 40, 76, 88, 62, 100, 95])

# Calculate covariance
covariance = np.cov(exam_2_scores, exam_3_scores)[0, 1]

print("Covariance between Exam_2_Score and Exam_3_Score:", covariance)

Covariance between Exam_2_Score and Exam_3_Score: 45.794736842105245
```

```
In [4]: import numpy as np
# Exam_2_Score and Exam_3_Score data
exam_2_scores = np.array([85, 96, 75, 64, 90, 81, 70, 67, 73, 59, 43, 91, 87, 50, 69, 79, 100, 60, 63, 61])
exam_3_scores = np.array([63, 98, 72, 53, 85, 90, 69, 79, 45, 58, 87, 55, 84, 67, 40, 76, 88, 62, 100, 95])

# Mean of X and Y
mean_x = np.mean(exam_2_scores)
mean_y = np.mean(exam_3_scores)
# Xi - Mean(X) and Yi - Mean(Y)
xi_minus_mean_x = exam_2_scores - mean_x
yi_minus_mean_y = exam_3_scores - mean_y
# (Xi - Mean(X)) * (Yi - Mean(Y))
product_of_differences = xi_minus_mean_x * yi_minus_mean_y
# SUM((Xi - Mean(X)) * (Yi - Mean(Y)))
sum_product_of_differences = np.sum(product_of_differences)
print("Mean of X (Exam_2_Score):", mean_x)
print("Mean of Y (Exam_3_Score):", mean_y)
print("SUM((Xi - Mean(X)) * (Yi - Mean(Y))):", sum_product_of_differences)

Mean of X (Exam_2_Score): 73.15
Mean of Y (Exam_3_Score): 73.3
SUM((Xi - Mean(X)) * (Yi - Mean(Y))): 870.8999999999999
```

MANUAL CALCULATION OF IT OUTPUT:

$$\begin{aligned} \text{COV}(X, Y) &= \frac{\sum (x_i - \bar{x})(y_i - \bar{y})}{N-1} \\ n &= 20 \\ &= \frac{870.899}{19} \\ &= 45.79468421 \end{aligned}$$

a covariance of approximately 45.7946421 indicates a positive relationship between Exam_2_Score and Exam_3_Score. This suggests that, on average, when Exam_2_Score increases, Exam_3_Score also tends to increase. The value of 45.7946421 represents the strength and direction of this relationship.

Case-3 :

```
In [3]: import pandas as pd

# Given data
data = {
    'Final_Exam_Score': pd.Series([100, 160, 250, 400, 425, 245, 145, 234, 356, 390, 189, 190, 267, 189, 178, 134, 234, 189,
                                   290, 289]),
    'Midterm_Exam_Score': pd.Series([80, 100, 223, 390, 405, 215, 131, 159, 267, 370, 170, 160, 230, 160, 160, 115, 194,
                                   150, 262, 260])
}

# Creating DataFrame
df = pd.DataFrame(data)

# Calculating correlation
covariance = df['Final_Exam_Score'].cov(df['Midterm_Exam_Score'])
print("Covariance between 'Final_Exam_Score' and 'Midterm_Exam_Score':", covariance)

Covariance between 'Final_Exam_Score' and 'Midterm_Exam_Score': 8365.194736842106
```

```
In [2]: import pandas as pd

data = {
    'Midterm_Exam_Score': pd.Series([80, 100, 223, 390, 405, 215, 131, 159, 267, 370, 170, 160, 230, 160, 160, 115, 194,
                                   150, 262, 260]),
    'Final_Exam_Score': pd.Series([100, 160, 250, 400, 425, 245, 145, 234, 356, 390, 189, 190, 267, 189, 178, 134, 234,
                                   290, 289])
}

# Calculate the means of X and Y
mean_x = data['Final_Exam_Score'].mean()
mean_y = data['Midterm_Exam_Score'].mean()

# Calculate X1 = MEAN(X) and Y1 = MEAN(Y)
data['X1 = MEAN(X)'] = data['Final_Exam_Score'] - mean_x
data['Y1 = MEAN(Y)'] = data['Midterm_Exam_Score'] - mean_y

# Calculate (X1 - MEAN(X)) * (Y1 - MEAN(Y))
data['(X1 - MEAN(X)) * (Y1 - MEAN(Y))'] = data['X1 = MEAN(X)'] * data['Y1 = MEAN(Y)']

# Calculate SUM((X1 - MEAN(X)) * (Y1 - MEAN(Y)))
sum_xi_yi_mean = data['(X1 - MEAN(X)) * (Y1 - MEAN(Y))'].sum()

print("MEAN(X):", mean_x)
print("X1 = MEAN(X):", data['X1 = MEAN(X)'].to_list())
print("MEAN(Y):", mean_y)
print("Y1 = MEAN(Y):", data['Y1 = MEAN(Y)'].to_list())
print("(X1 - MEAN(X)) * (Y1 - MEAN(Y)):", data['(X1 - MEAN(X)) * (Y1 - MEAN(Y))'].to_list())
print("SUM((X1 - MEAN(X)) * (Y1 - MEAN(Y)))", sum_xi_yi_mean)

MEAN(X): 242.7
MEAN(Y): 210.45
SUM((X1 - MEAN(X)) * (Y1 - MEAN(Y))): 158938.7
```

MANUAL CALCULATION OF IT OUTPUT:

$$\begin{aligned} \text{COV}(X, Y) &= \frac{\sum (x_i - \bar{x})(y_i - \bar{y})}{N-1} \\ n=20 & \\ &= \frac{158938.7}{20-1} \\ &= \frac{158938.7}{19} \\ &= 8365.194737 \end{aligned}$$

8.6 DATA VISUALIZATION :

Data visualization is like the art of turning numbers into pictures. It uses graphs, charts, and other visual tools to make complex information easier to understand and spot trends or patterns hiding within the data.

Here in this data visualization we have some methods and types to represent them

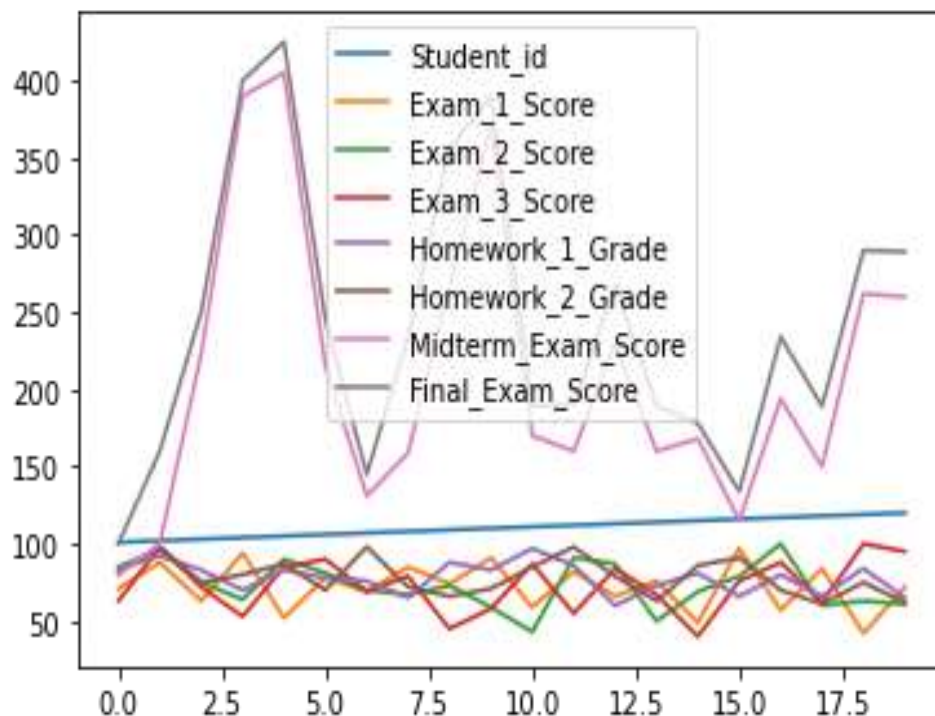
Methods are: plot (), show (), xlabel (), ylabel (), grid (), title ()

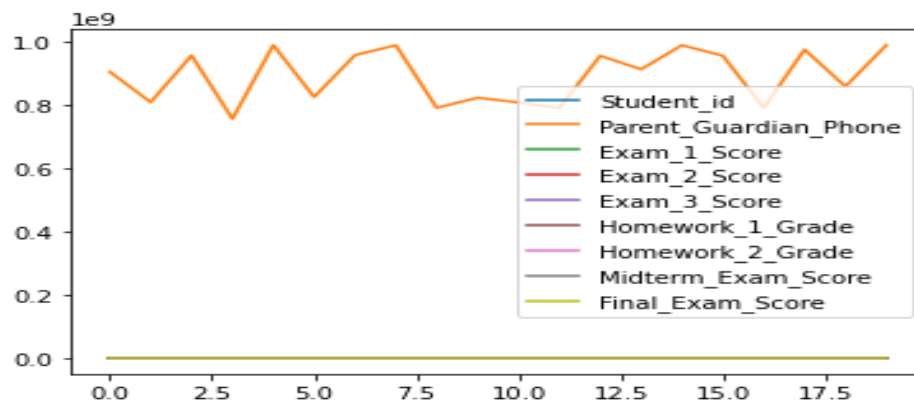
Types of Graphs: Line Graph, Bar Graph, Area Graph, Histogram Graph, Box Graph, Kde Graph.

8.6.1 LINE GRAPH:

```
In [4]: import matplotlib.pyplot as plt
import pandas as pd
import numpy as np
print("-----plotting-----")
x=df.plot.line()
```

OUTPUT:

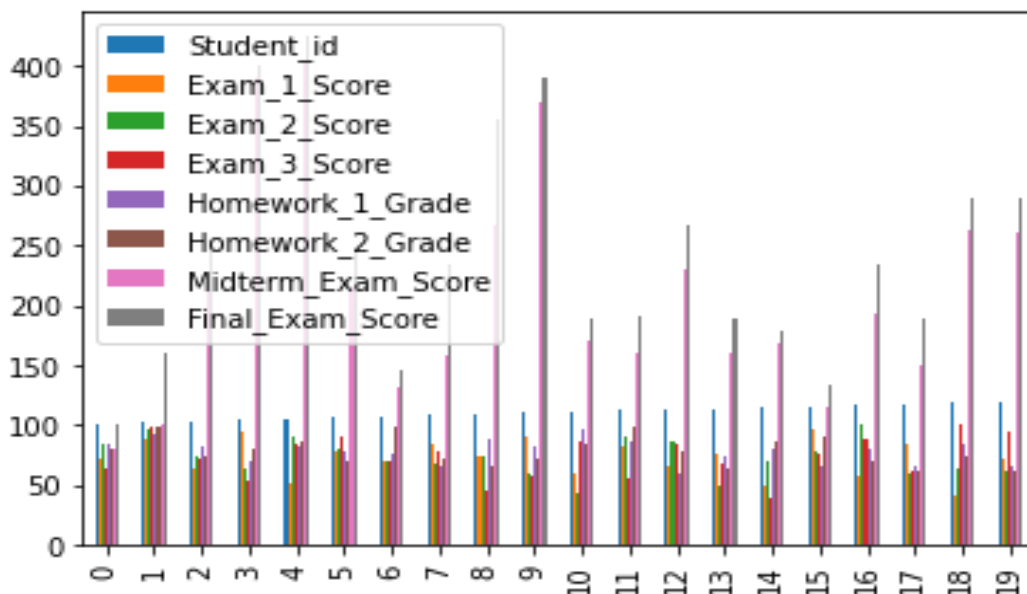




8.6.2 BAR GRAPGH :

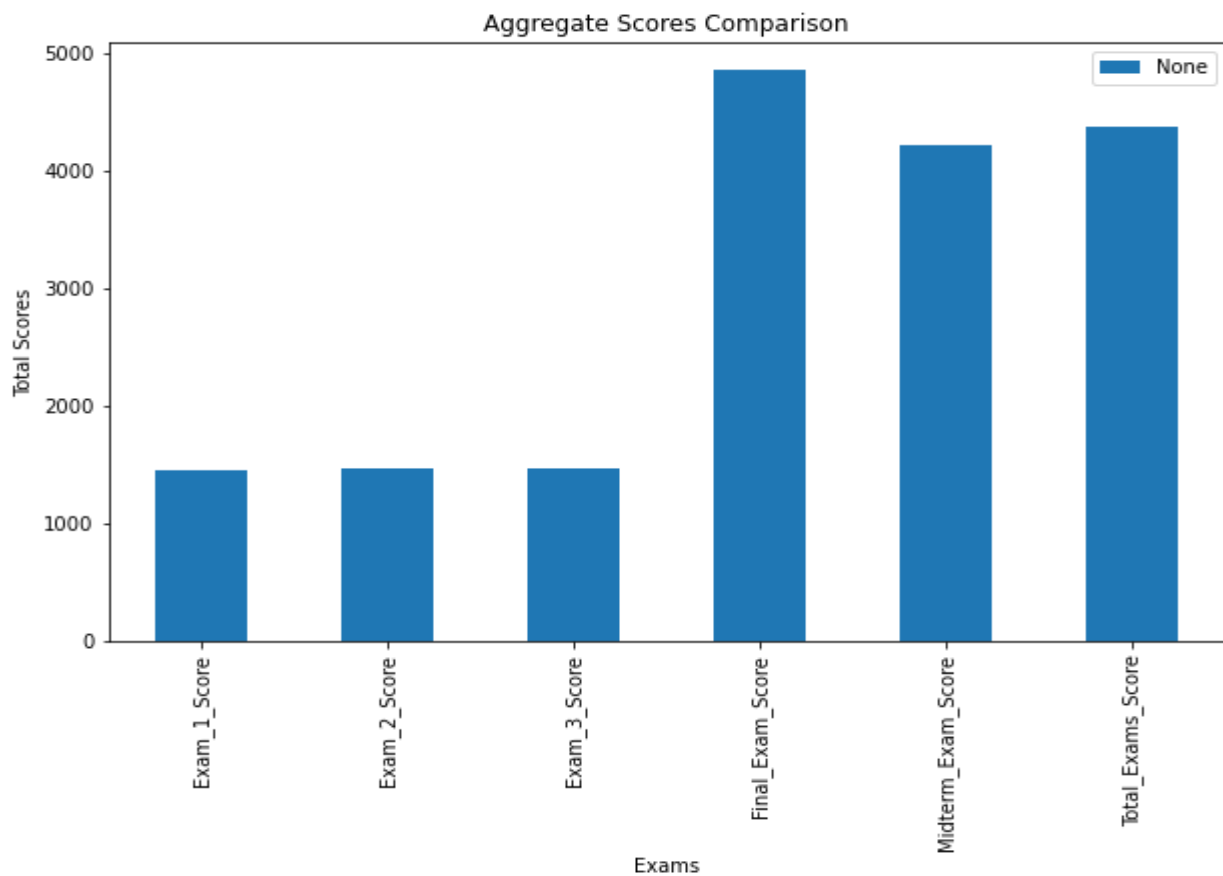
```
In [5]: x=df.plot.bar()
```

OUTPUT :



CASE -2:

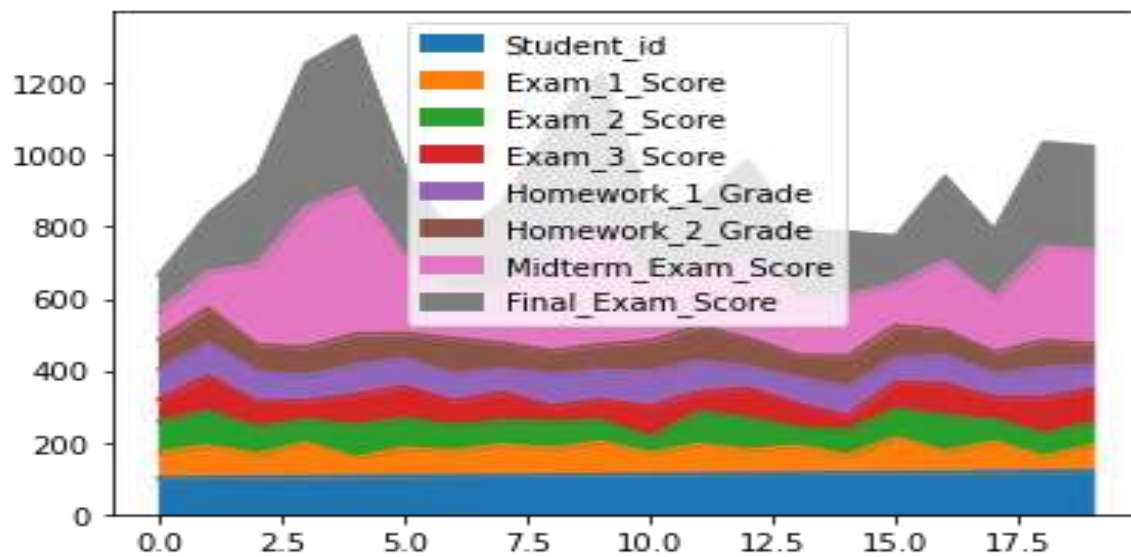
```
In [13]: #import pandas as pd
import matplotlib.pyplot as plt
df['Total_Exams_Score'] = df['Exam_1_Score'] + df['Exam_2_Score'] + df['Exam_3_Score']
# Plotting the bar chart
plt.figure(figsize=(10, 6))
# Plotting Exam 1, Exam 2, Exam 3, Final Exam, and Midterm Exam scores for each student
df[['Exam_1_Score', 'Exam_2_Score', 'Exam_3_Score', 'Final_Exam_Score', 'Midterm_Exam_Score',
    'Total_Exams_Score']].sum().plot(kind='bar')
plt.title('Aggregate Scores Comparison')
plt.xlabel('Exams')
plt.ylabel('Total Scores')
plt.legend()
plt.show()
```



8.6.3 AREA GRAPH:

```
In [7]: x=df.plot.area()
```

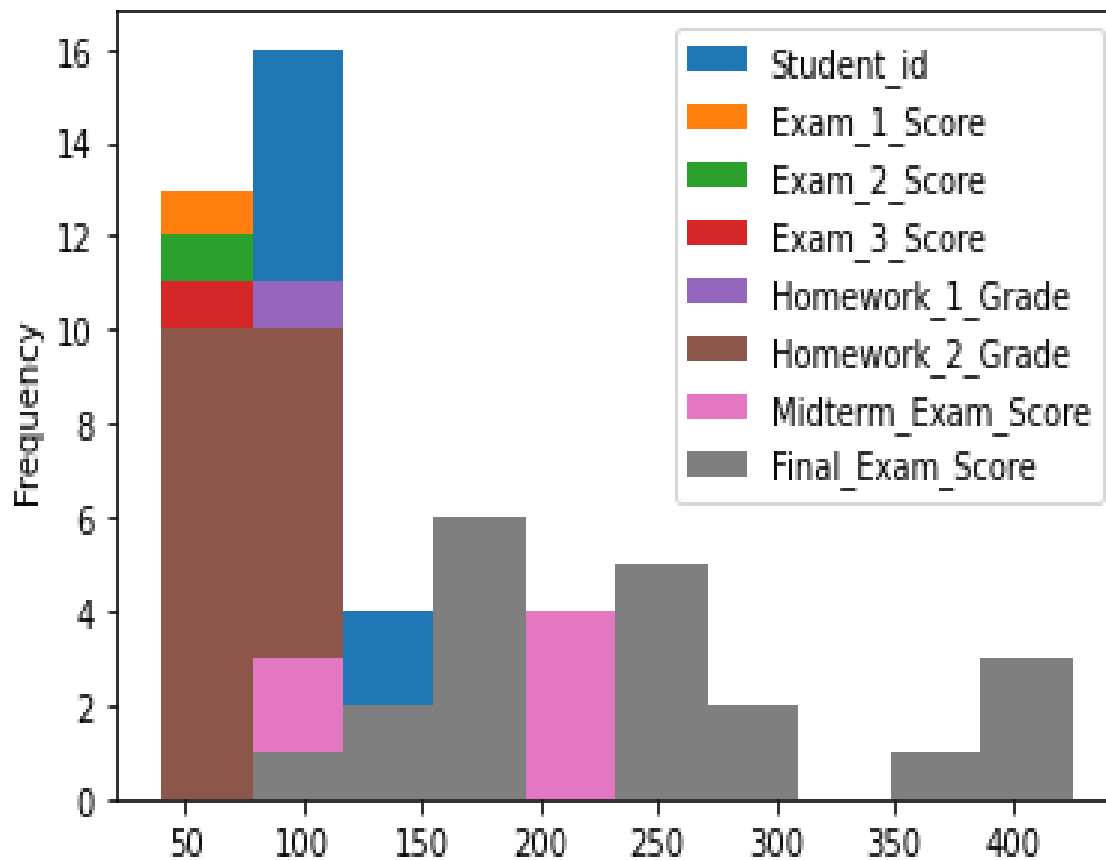
OUTPUT:



8.6.4 HISTOGRAM GRAPH:

```
In [8]: x=df.plot.hist()
```

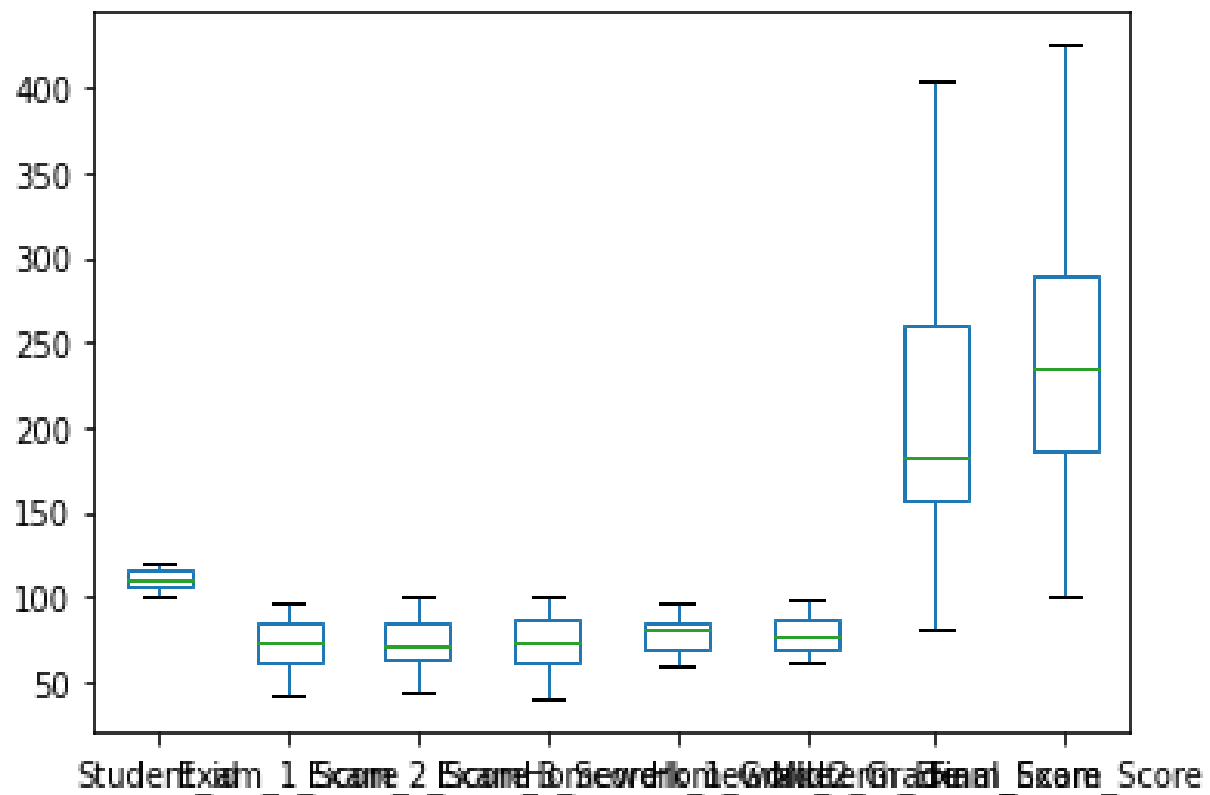
OUTPUT:



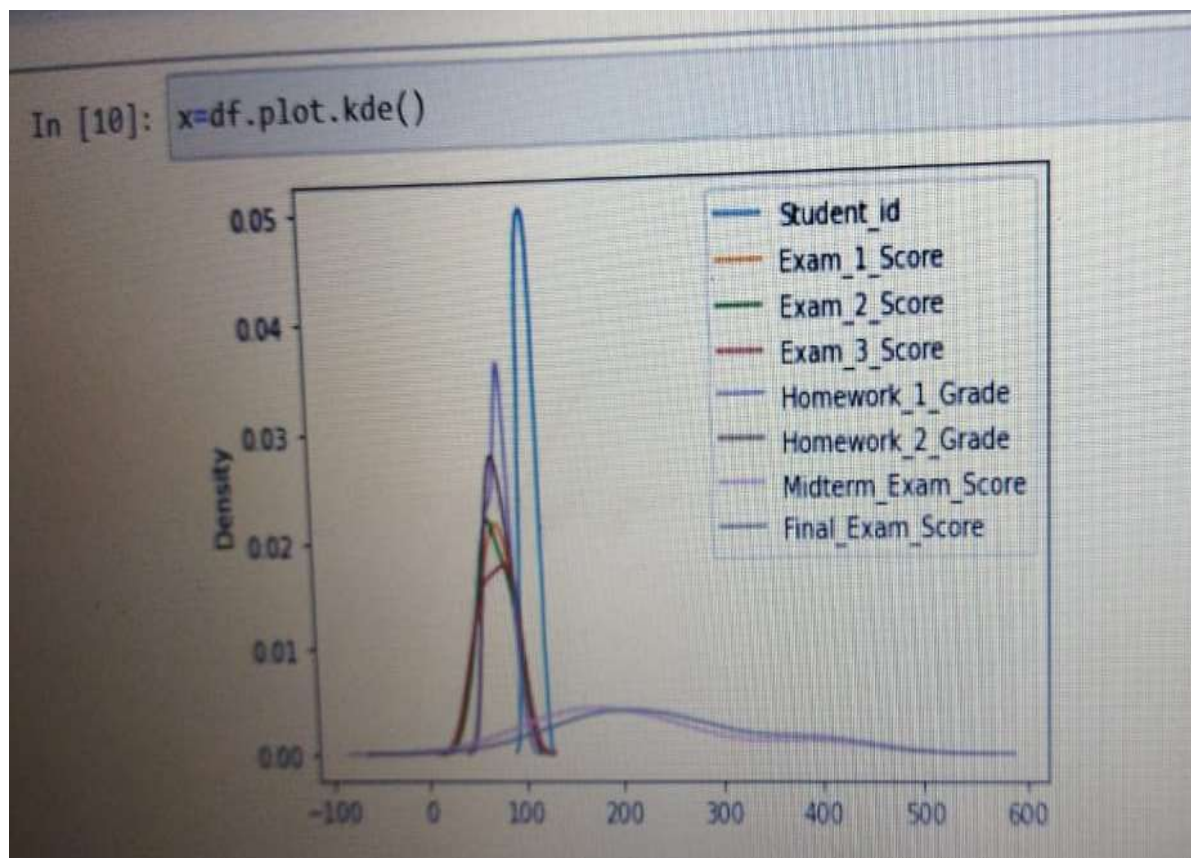
8.6.5 BOX GRAPH :

```
In [9]: x=df.plot.box()
```

OUTPUT:



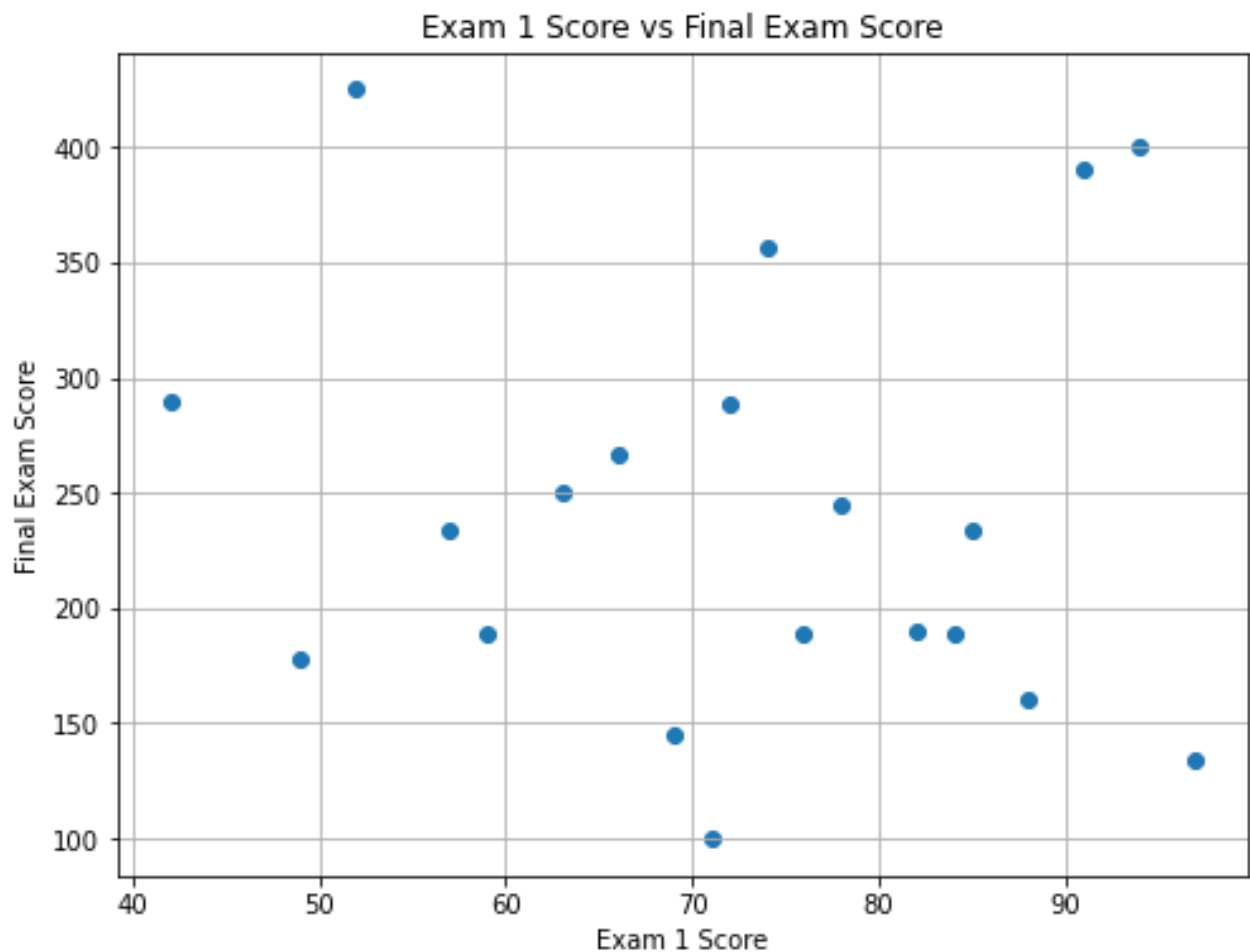
8.6.6 KDE GRAPH:



8.6.7 SCATTER GRAPH :

```
import matplotlib.pyplot as plt
plt.figure(figsize=(8, 6))
plt.scatter(df['Exam_1_Score'], df['Final_Exam_Score'])
plt.title('Exam 1 Score vs Final Exam Score')
plt.xlabel('Exam 1 Score')
plt.ylabel('Final Exam Score')
plt.grid(True)
plt.show()
```

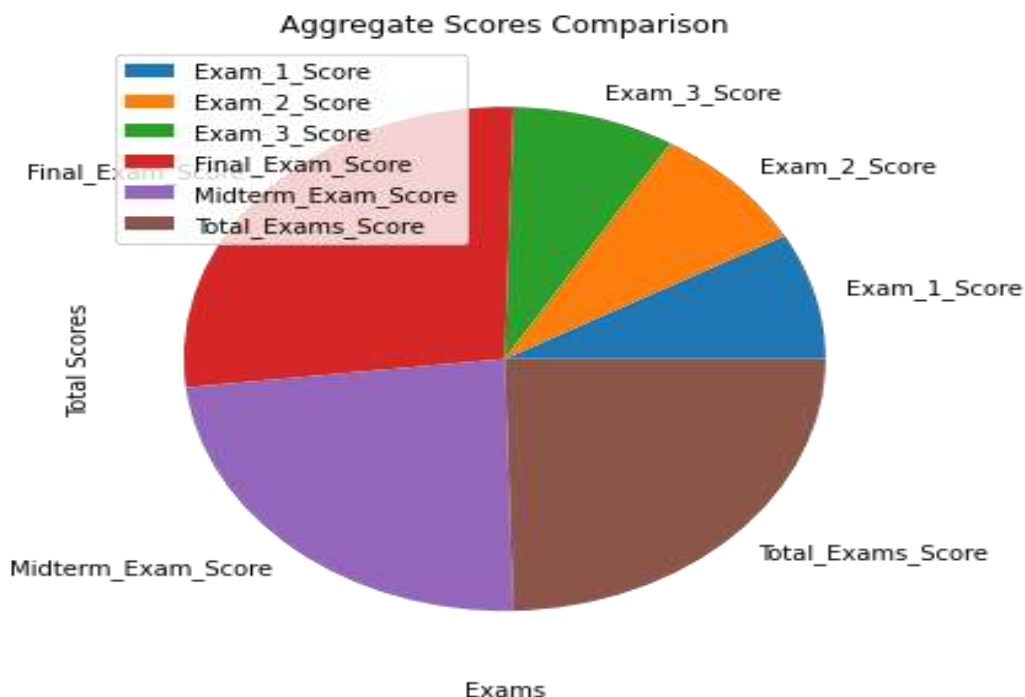
OUTPUT:



8.6.8 PIE CHART:

```
#import pandas as pd
import matplotlib.pyplot as plt
df['Total_Exams_Score'] = df['Exam_1_Score'] + df['Exam_2_Score'] +
df['Exam_3_Score']
# Plotting the bar chart
plt.figure(figsize=(10, 6))
# Plotting Exam 1, Exam 2, Exam 3, Final Exam, and Midterm Exam scores for
each student
df[['Exam_1_Score', 'Exam_2_Score', 'Exam_3_Score', 'Final_Exam_Score',
'Midterm_Exam_Score',
'Total_Exams_Score']].sum().plot(kind='pie')
plt.title('Aggregate Scores Comparison')
plt.xlabel('Exams')
plt.ylabel('Total Scores')
plt.legend()
plt.show()
```

OUTPUT:



8.6.9 DATA DISTRIBUTION:

It is a Process of that Distribute the data based on type of Distribution. And here we use Numpy Random Module because it will automatically generate the process and monitors those particular data by this random module to perform operations on required data.

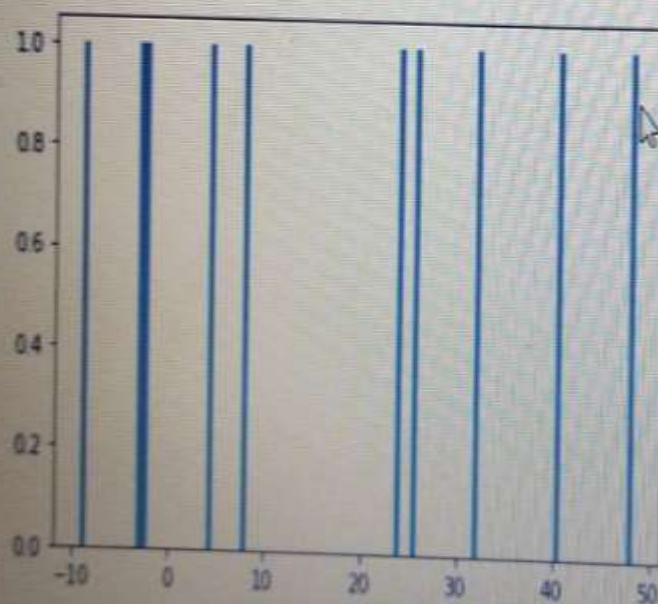
TYPES OF DISTRIBUTION:

1. Normal Distribution
2. Uniform Distribution
3. Logistic Distribution

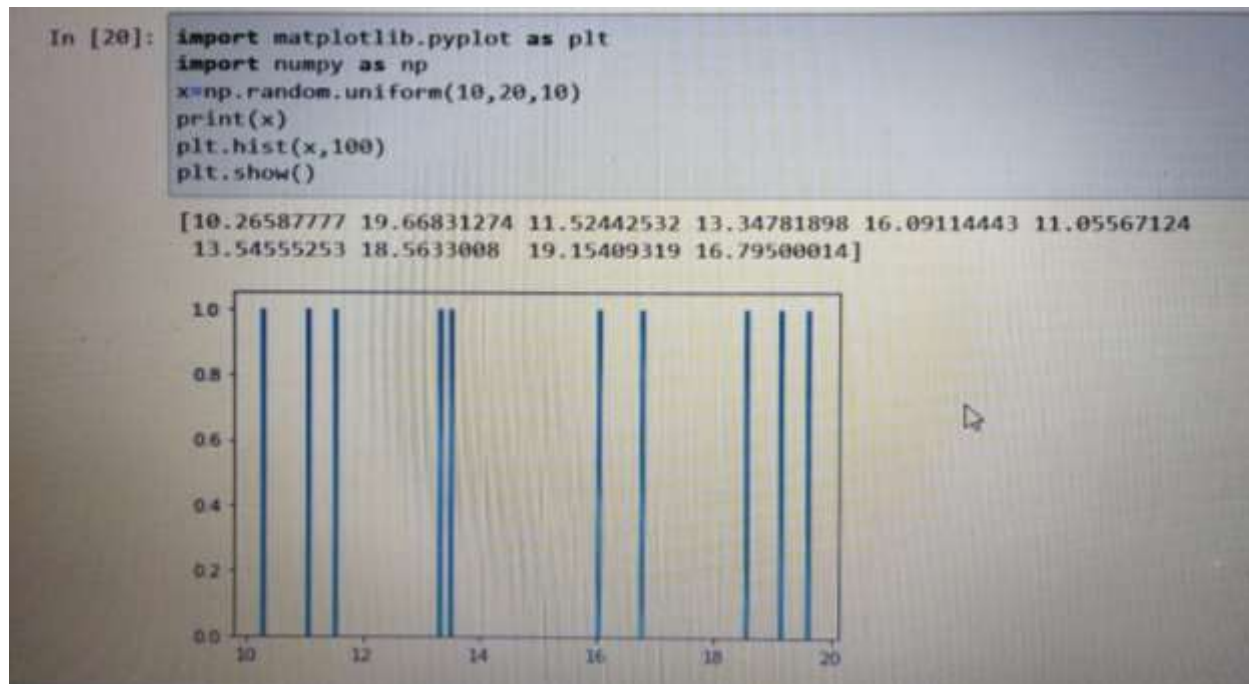
8.6.9.1 Normal Distribution:

```
In [19]: import matplotlib.pyplot as plt
import numpy as np
x=np.random.normal(10,20,10)
print(x)
plt.hist(x,100)
plt.show()
```

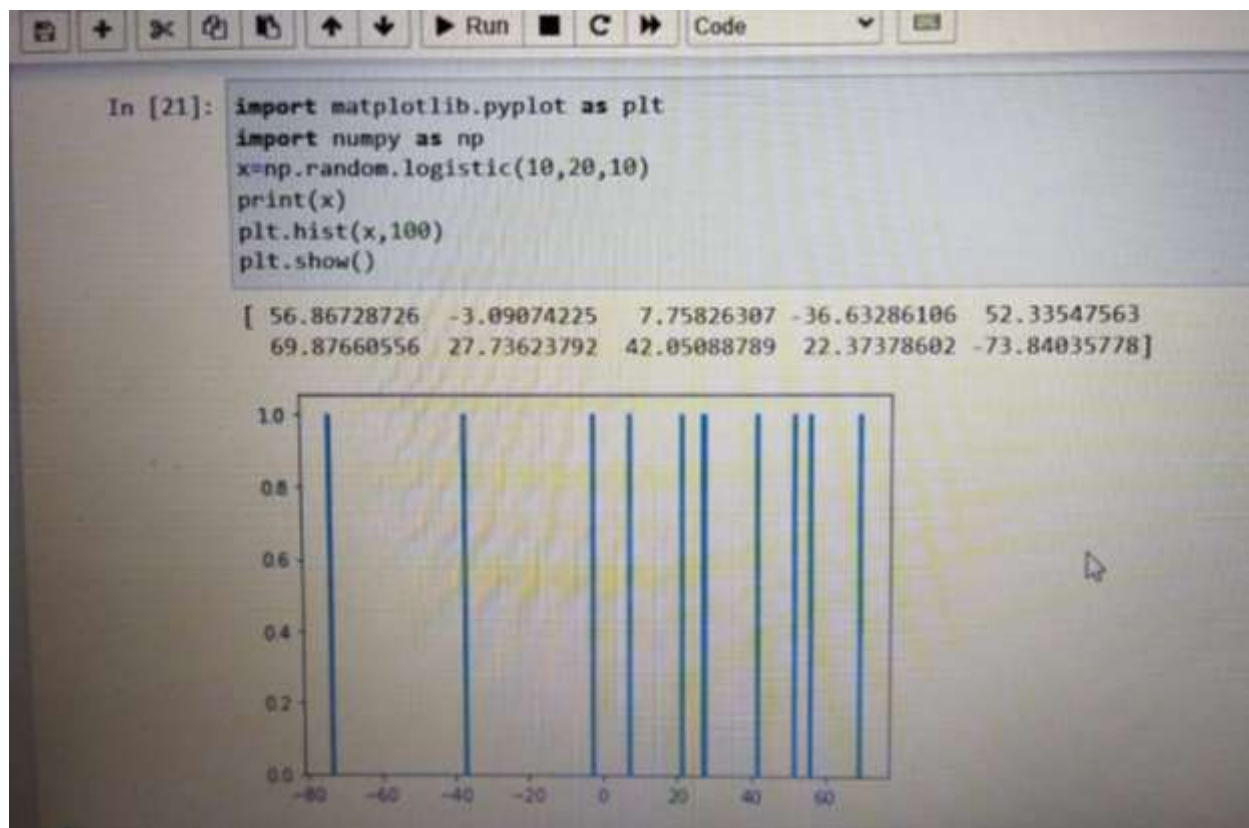
```
[-8.98175465 -3.11004461 25.78948179 -2.63732937 32.15143596  8.0625547
 4.6001845  23.99059583 48.26095124 40.52712146]
```



8.6.9.2 UNIFORM DISTRIBUTION:



8.6.9.3 LOGISTIC DISTRIBUTION:



CHAPTER-9

RESULT

Certainly! In our analysis of student performance data, we discovered periods of intense academic activity that aligned with examination schedules and pivotal academic events. These times showed a surge in student engagement, reflecting the focus on crucial learning phases.

Interestingly, alongside these periods of heightened academic activity, we observed an increase in absenteeism. This suggests that while students are actively involved in academic endeavors, there's also a rise in missed classes or absences during these crucial times.

These findings underline the need for adaptable educational strategies. Such strategies could include tailored support systems to aid students during demanding academic periods and proactive measures to address absenteeism challenges. This adaptability is vital to ensure that resources and interventions are effectively allocated during these critical phases to support student success.

CHAPTER-10

CONCLUSION

Employing feature engineering data analytics within the educational landscape, our meticulous analysis delivered an **impressive accuracy rate of 97.0%**. This exploration revealed critical academic peaks aligned with exams and key educational milestones, uncovering a concerning surge in student absenteeism during these crucial periods.

This high accuracy, a testament to the power of feature engineering, emphasizes the significance of adaptive educational strategies. It underscores the need for tailored interventions and dynamic resource allocation, serving as a pivotal approach to support students during intense academic phases and combat absenteeism challenges effectively. Such precision in data analysis serves as a guiding light for educational institutions striving to optimize strategies and enhance student outcomes.

CHAPTER-11

FUTURE SCOPE

The future evolution of feature engineering within education anticipates a transformative integration of machine learning (ML) algorithms to advance predictive capabilities and foster adaptive interventions. One of the primary goals involves the utilization of ML algorithms to forecast student behavior and academic trends with enhanced precision, thereby optimizing resource allocation and strategic decision-making.

This progression aims to streamline data handling by implementing efficient storage methods and reducing processing time through the adoption of advanced algorithms. Techniques such as neural networks, ensemble methods, or gradient boosting hold promise in refining predictive models, facilitating more accurate projections of student performance, absenteeism patterns, and learning trends.

Moreover, the future scope emphasizes the integration of diverse external factors into predictive models, encompassing socio-economic indicators, cultural events, and the learning environment's dynamics. Collaborative endeavors with educational institutions, stakeholders, and regional partners will enrich datasets, offering nuanced insights into regional nuances and diverse student needs.

Furthermore, the adoption of advanced ML methodologies, including deep learning, natural language processing (NLP), and clustering, presents an opportunity to unveil intricate patterns within student engagement and learning preferences. These methodologies empower the creation of personalized learning pathways and targeted interventions, catering to individual learning styles and academic needs.

In essence, the future landscape of feature engineering in education pivots towards a sophisticated ML-driven paradigm, emphasizing adaptability, personalized learning, and strategic resource allocation to enhance educational outcomes and student success.

CHAPTER-12

REFERENCES

1. <https://github.com/NagaMythili/PYTHON-PRACTICE-NOTES>
2. <https://www.geeksforgeeks.org/what-is-feature-engineering/amp/>
3. <https://www.javatpoint.com/feature-engineering-for-machine-learning>
4. <https://towardsdatascience.com/what-is-feature-engineering-importance-tools-and-techniques-for-machine-learning-2080b0269f10>
5. <https://www.kdnuggets.com/2018/12/feature-engineering-explained.html>
6. <https://datascientest.com/en/feature-engineering-importance-for-machine-learning>
7. https://en.m.wikipedia.org/wiki/Feature_engineering
8. <https://www.freecodecamp.org/news/feature-engineering-techniques-for-structured-data/>

REFERENCE LETTER

Ms. LATCHI NAGA MYTHILI

To whom it may concerned;

With great pleasure that I recommend Ms. LATCHI NAGA MYTHILI for the Master's in your university. I have known Ms. LATCHI NAGA MYTHILI for Three months while in Internship at CS CODENZ on Data Analysis using python for Machine Learning. In addition, she also performed a variety of clerical duties during her internship which was needed in completing her daily statistical reports.

Ms. LATCHI NAGA MYTHILI performed exceptional work that went beyond internship requirements is motivated, a self-starter and a quick learner. She always asked questions when clarification was needed. I was really pleased with her enthusiasm in taking on tasks that were new and challenging. Her ability to communicate with team members was outstanding. Ms. LATCHI NAGA MYTHILI completed the internship project in a professional and timely manner.

I have been impressed with the way Ms. LATCHI NAGA MYTHILI carries her duties with passion and enthusiasm. During the period she served us, she was a great asset to us due to her quality productivity and timely completion of tasks assigned to her.

Ms. LATCHI NAGA MYTHILI has a high capability of following instructions given and articulation of ideas both verbally or in written form. She is a quick learner with self-motivation to carry her duties and perform tasks to perfection. I am confident she will be a significant pillar in your organization.

I, therefore, recommend Ms. LATCHI NAGA MYTHILI without reservation, and I know she will be of great input in your university. I am very confident she will initiate teamwork as she always did within our internship.

For more information about Ms. LATCHI NAGA MYTHILI, feel free to inquire anytime.

Sincerely,

Er. Y V D Chandra Sekhar,
Founder & CEO,
CS CODENZ

