AI & ML INTERNSHIP

Task 1: Understanding Dataset & Data Types

Dataset Analysis Report (Students Performance Dataset)

1. Introduction

Understanding the dataset is a crucial step in the machine learning pipeline. Before building any predictive model, it is important to analyze the dataset structure, identify data types, detect data quality issues, and evaluate whether the dataset is suitable for machine learning tasks.
This report presents a detailed analysis of the **Students Performance Dataset** with respect to data types, feature distribution, and ML readiness.

2. Dataset Overview

The **Students Performance Dataset** contains academic performance records of students along with demographic and socio-economic attributes.
Each row represents a student, and each column represents a specific feature related to academic scores or background information.

3. Tools and Technologies Used

Python

Pandas

NumPy

Jupyter Notebook

These tools were used for data loading, inspection, and exploratory analysis.

4. Data Loading and Initial Inspection

The dataset was loaded using Pandas. The first and last few records were displayed to understand:

Column names

Type of values stored

Dataset structure

This step provides a high-level overview of the data.

5. Identification of Data Types

Based on column values and names, the dataset features were classified as follows:

Numerical Data:

math score

reading score

writing score

Categorical Data:

gender

race/ethnicity

parental level of education

lunch

test preparation course

There are no ordinal or binary variables explicitly present in this dataset.

6. Dataset Information and Statistical Summary

The dataset was analyzed using df.info() and df.describe():

df.info() helped identify data types and check for missing values

df.describe() provided statistical insights such as mean, minimum, maximum, and standard deviation

The dataset was found to be clean with no missing values.

7. Categorical Data Distribution

Unique values and value counts were analyzed for categorical columns to understand class distribution.
This step helps detect dominance of certain categories that may influence model predictions.

8. Target Variable and Input Features

For machine learning suitability:

Target Variable:

math score (can also be reading score or writing score depending on the problem)

Input Features:

gender

race/ethnicity

parental level of education

lunch

test preparation course

reading score

writing score

This dataset is suitable for **regression problems**.

9. Dataset Size and ML Suitability

The dataset contains an adequate number of records and features, making it suitable for:

Regression modeling

Feature importance analysis

With proper preprocessing such as encoding categorical variables, it can be effectively used for machine learning.

10. Data Quality Observations

No missing values detected

Categorical features require encoding

Numerical features may require normalization

Overall, the dataset is clean and ML-ready.

11. Final Outcome

By completing this task, the dataset was successfully analyzed for:

Data structure

Feature classification

Data quality

ML readiness

This fulfills the objectives of **Task 1: Understanding Dataset & Data Types**.

✅ REQUIRED PYTHON CODE (Jupyter Notebook)

Use the following code **as-is** in your notebook.

1. Import Libraries

```
import pandas as pd
import numpy as np
```

2. Load the Dataset

```
df = pd.read_csv("StudentsPerformance.csv")
```

3. Display First and Last Records

```
df.head()
```

```
[9]: import pandas as pd
     import numpy as np
     df = pd.read_csv("StudentsPerformance.csv")
     df.head()
```

[9]:

	school	sex	age	address	famsize	Pstatus	Medu	Fedu	Mjob	Fjob	...	famrel	freetime	goout	Dalc	Walc	health	absences	G1	G2	G3
0	GP	F	18	U	GT3	A	4	4	at_home	teacher	...	4	3	4	1	1	3	6	5	6	6
1	GP	F	17	U	GT3	T	1	1	at_home	other	...	5	3	3	1	1	3	4	5	5	6
2	GP	F	15	U	LE3	T	1	1	at_home	other	...	4	3	2	2	3	3	10	7	8	10
3	GP	F	15	U	GT3	T	4	2	health	services	...	3	2	2	1	1	5	2	15	14	15
4	GP	F	16	U	GT3	T	3	3	other	other	...	4	3	2	1	2	5	4	6	10	10

5 rows × 33 columns

```
df.tail()
```

```
[10]: df.tail()
```

[10]:

	school	sex	age	address	famsize	Pstatus	Medu	Fedu	Mjob	Fjob	...	famrel	freetime	goout	Dalc	Walc	health	absences	G1	G2	G3
390	MS	M	20	U	LE3	A	2	2	services	services	...	5	5	4	4	5	4	11	9	9	9
391	MS	M	17	U	LE3	T	3	1	services	services	...	2	4	5	3	4	2	3	14	16	16
392	MS	M	21	R	GT3	T	1	1	other	other	...	5	5	3	3	3	3	3	10	8	7
393	MS	M	18	R	LE3	T	3	2	services	other	...	4	4	1	3	4	5	0	11	12	10
394	MS	M	19	U	LE3	T	1	1	other	at_home	...	3	2	3	3	3	5	5	8	9	9

5 rows × 33 columns

4. Dataset Information

```
df.info()
```

```
[11]: df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 395 entries, 0 to 394
Data columns (total 33 columns):
 #   Column      Non-Null Count  Dtype
---  ------      --------------  -----
 0   school      395 non-null    object
 1   sex         395 non-null    object
 2   age         395 non-null    int64
 3   address     395 non-null    object
 4   famsize     395 non-null    object
 5   Pstatus     395 non-null    object
 6   Medu        395 non-null    int64
 7   Fedu        395 non-null    int64
 8   Mjob        395 non-null    object
 9   Fjob        395 non-null    object
 10  reason      395 non-null    object
 11  guardian    395 non-null    object
 12  traveltime  395 non-null    int64
 13  studytime   395 non-null    int64
 14  failures    395 non-null    int64
 15  schoolsup   395 non-null    object
 16  famsup      395 non-null    object
 17  paid        395 non-null    object
 18  activities  395 non-null    object
 19  nursery     395 non-null    object
 20  higher      395 non-null    object
 21  internet    395 non-null    object
 22  romantic    395 non-null    object
 23  famrel      395 non-null    int64
 24  freetime    395 non-null    int64
 25  goout       395 non-null    int64
 26  Dalc        395 non-null    int64
 27  Walc        395 non-null    int64
 28  health      395 non-null    int64
```

5. Statistical Summary

df.describe()

```
[12]: df.describe()
```

[12]:

	age	Medu	Fedu	traveltime	studytime	failures	famrel	freetime	goout	Dalc	Walc	health	absences	
count	395.000000	395.000000	395.000000	395.000000	395.000000	395.000000	395.000000	395.000000	395.000000	395.000000	395.000000	395.000000	395.000000	39
mean	16.696203	2.749367	2.521519	1.448101	2.035443	0.334177	3.944304	3.235443	3.108861	1.481013	2.291139	3.554430	5.708861	1
std	1.276043	1.094735	1.088201	0.697505	0.839240	0.743651	0.896659	0.998862	1.113278	0.890741	1.287897	1.390303	8.003096	
min	15.000000	0.000000	0.000000	1.000000	1.000000	0.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	0.000000	
25%	16.000000	2.000000	2.000000	1.000000	1.000000	0.000000	4.000000	3.000000	2.000000	1.000000	1.000000	3.000000	0.000000	
50%	17.000000	3.000000	2.000000	1.000000	2.000000	0.000000	4.000000	3.000000	3.000000	1.000000	2.000000	4.000000	4.000000	1
75%	18.000000	4.000000	3.000000	2.000000	2.000000	0.000000	5.000000	4.000000	4.000000	2.000000	3.000000	5.000000	8.000000	1
max	22.000000	4.000000	4.000000	4.000000	4.000000	3.000000	5.000000	5.000000	5.000000	5.000000	5.000000	5.000000	75.000000	1

6. Identify Data Types Manually

df.dtypes

```
[13]: df.dtypes

[13]: school        object
      sex           object
      age            int64
      address       object
      famsize       object
      Pstatus       object
      Medu           int64
      Fedu           int64
      Mjob          object
      Fjob          object
      reason        object
      guardian      object
      traveltime     int64
      studytime      int64
      failures       int64
      schoolsup     object
      famsup        object
      paid          object
      activities    object
      nursery       object
      higher        object
      internet      object
      romantic      object
      famrel         int64
      freetime       int64
      goout          int64
      Dalc           int64
      Walc           int64
      health         int64
      absences       int64
      G1             int64
      G2             int64
      G3             int64
      dtype: object
```

7. Check Unique Values in Categorical Columns

```
categorical_columns = df.select_dtypes(include='object').columns
for col in categorical_columns:
    print(f"\nUnique values in {col}:")
    print(df[col].value_counts())
```

```
[14]: categorical_columns = df.select_dtypes(include='object').columns
      for col in categorical_columns:
          print(f"\nUnique values in {col}:")
          print(df[col].value_counts())
```

```
Unique values in school:
school
GP    349
MS     46
Name: count, dtype: int64

Unique values in sex:
sex
F    208
M    187
Name: count, dtype: int64

Unique values in address:
address
U    307
R     88
Name: count, dtype: int64
```

8. Check for Missing Values

df.isnull().sum()

```
[15]: df.isnull().sum()
```

```
[15]: school        0
      sex           0
      age           0
      address       0
      famsize       0
      Pstatus       0
      Medu          0
      Fedu          0
      Mjob          0
      Fjob          0
      reason        0
      guardian      0
      traveltime    0
      studytime     0
      failures      0
      schoolsup     0
      famsup        0
      paid          0
      activities    0
      nursery       0
      higher        0
      internet      0
      romantic      0
      famrel        0
      freetime      0
      goout         0
      Dalc          0
      Walc          0
      health        0
      absences      0
      G1            0
      G2            0
      G3            0
      dtype: int64
```

9. Dataset Shape

df.shape

```
[21]: df.shape

[21]: (395, 33)
```

12. Conclusion

This analysis provided a clear understanding of the Students Performance Dataset in terms of structure, feature types, and machine learning suitability. Performing this analysis before modeling ensures better data preparation and improved model performance.