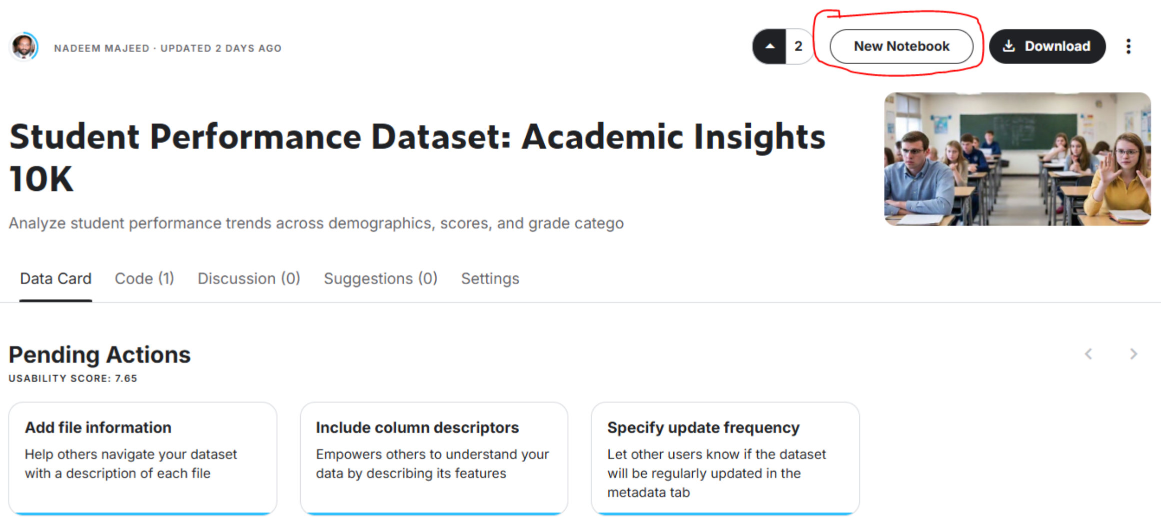
For Even Roll number please use this dataset.

https://www.kaggle.com/datasets/nadeemajeedch/students-performance-10000-clean-data-eda

login with your Kaggle account. And click the new Notebook.



Student Performance Dataset: Academic Insights 10K

Analyze student performance trends across demographics, scores, and grade catego

Pending Actions

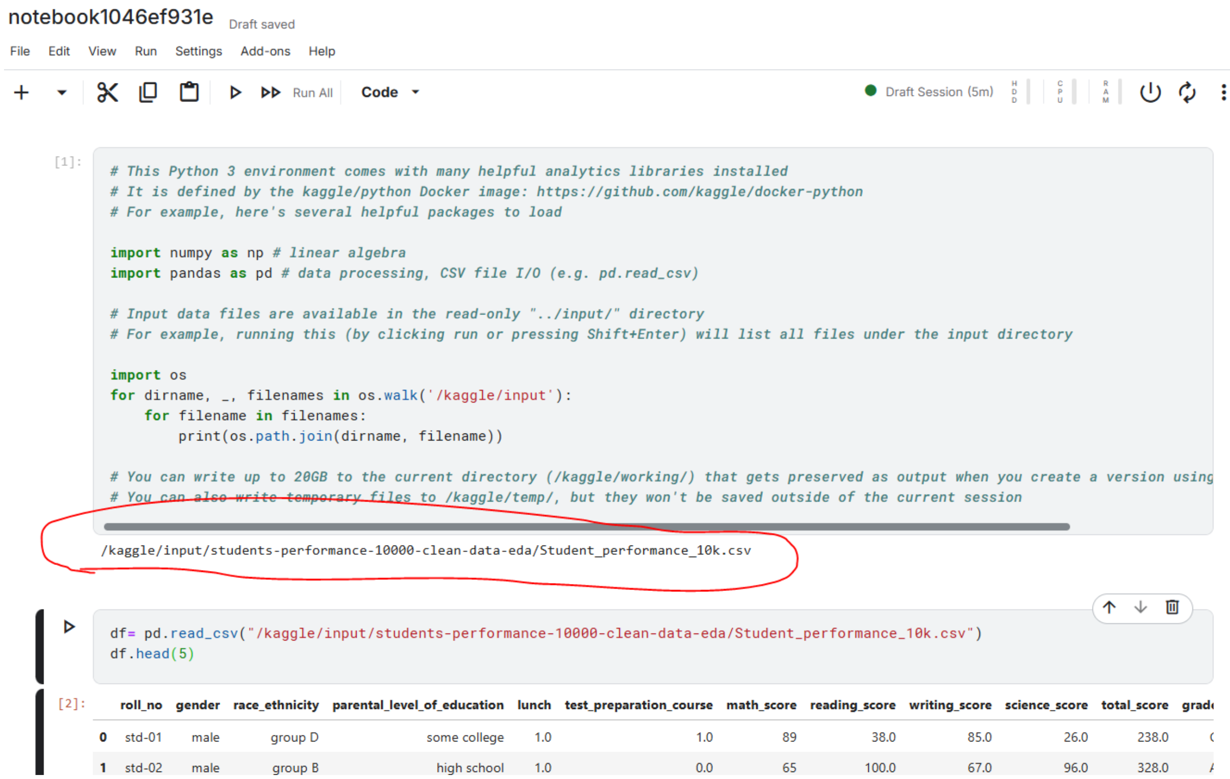
USABILITY SCORE: 7.65

Add file information
Help others navigate your dataset with a description of each file

Include column descriptors
Empowers others to understand your data by describing its features

Specify update frequency
Let other users know if the dataset will be regularly updated in the metadata tab

You will get the following Notebook page.



```
[1]:
# This Python 3 environment comes with many helpful analytics libraries installed
# It is defined by the kaggle/python Docker image: https://github.com/kaggle/docker-python
# For example, here's several helpful packages to load

import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)

# Input data files are available in the read-only "../input/" directory
# For example, running this (by clicking run or pressing Shift+Enter) will list all files under the input directory

import os
for dirname, _, filenames in os.walk('/kaggle/input'):
    for filename in filenames:
        print(os.path.join(dirname, filename))

# You can write up to 20GB to the current directory (/kaggle/working/) that gets preserved as output when you create a version using
# You can also write temporary files to /kaggle/temp/, but they won't be saved outside of the current session
```

```
/kaggle/input/students-performance-10000-clean-data-eda/Student_performance_10k.csv
```

```
df= pd.read_csv("/kaggle/input/students-performance-10000-clean-data-eda/Student_performance_10k.csv")
df.head(5)
```

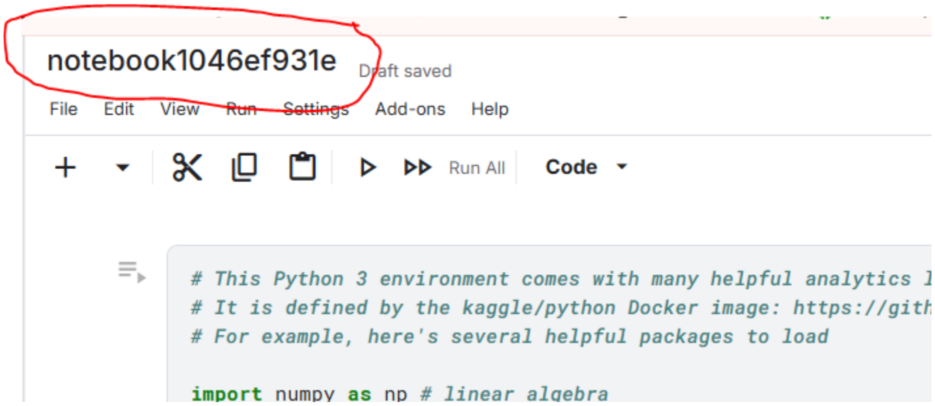
[2]:

	roll_no	gender	race_ethnicity	parental_level_of_education	lunch	test_preparation_course	math_score	reading_score	writing_score	science_score	total_score	grad
0	std-01	male	group D	some college	1.0	1.0	89	38.0	85.0	26.0	238.0	C
1	std-02	male	group B	high school	1.0	0.0	65	100.0	67.0	96.0	328.0	A

On execution of the cell, you will get the path of the file. Use this path to load the data file.

df= pd.read_csv("/kaggle/input/students-performance-10000-clean-data-eda/Student_performance_10k.csv")

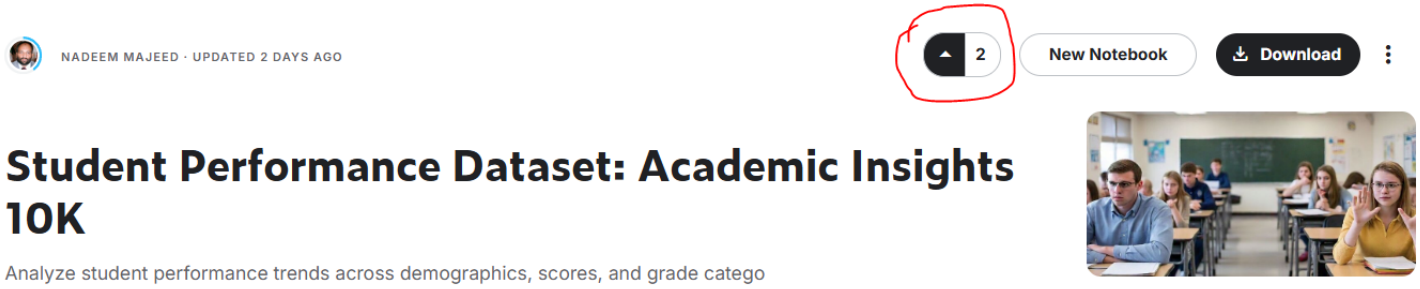
Click here and change the file name.



```
# This Python 3 environment comes with many helpful analytics l
# It is defined by the kaggle/python Docker image: https://gith
# For example, here's several helpful packages to load

import numpy as np # linear algebra
```

Don't forget to vote for the dataset. 😊



NADEEM MAJEED · UPDATED 2 DAYS AGO

Student Performance Dataset: Academic Insights 10K

Analyze student performance trends across demographics, scores, and grade catego

Tasks: Preprocessing and EDA Steps

Step 1: Load the Data

- Import the necessary libraries (`pandas`, `numpy`, `matplotlib`, `seaborn`, etc.).
- Load the dataset into a pandas DataFrame using `pd.read_csv()`.
- Display the first few rows of the dataset using `.head()`.

Step 2: Understand the Data

1. Check the shape of the dataset using `.shape` to see the number of rows and columns.
2. Display the column names using `.columns`.
3. Use `.info()` to examine the data types and the number of non-null values in each column.
4. Use `.describe()` to get a summary of numeric columns (mean, min, max, standard deviation).

Step 3: Identify Missing Values

1. Check for missing values using `.isnull().sum()`.
2. Visualize missing data using a heatmap (`sns.heatmap`) to see patterns of missingness.
3. Decide how to handle missing values:
   - For numeric columns, use mean or median imputation.
   - For categorical columns, use mode imputation or a placeholder (e.g., "Unknown").

Step 4: Handle Duplicates

- Check for duplicate rows using `.duplicated().sum()`.
- Drop duplicates using `.drop_duplicates()` if any are found.

Step 5: Check for Inconsistent or Faulty Data

1. Examine categorical columns (`gender`, `race_ethnicity`, etc.) for typos or inconsistent values using `.unique()`.
2. Ensure numeric columns (`math_score`, `total_score`, etc.) contain valid numbers (e.g., no special characters like `?` or negative values).
   - Convert `math_score` to numeric using `pd.to_numeric()` with `errors='coerce'`.
   - Handle invalid entries by replacing them with `NaN` and imputing or dropping them.

Step 6: Drop Irrelevant Columns

- Decide if any columns (like `roll_no`) should be dropped because they do not contribute to analysis.
- Drop columns using `.drop()`.

Step 7: Convert Data Types

- Ensure all columns have appropriate data types:
  - Convert categorical columns (e.g., `gender`, `grade`) to `category` using `.astype('category')`.
  - Convert scores and other numeric data to `float` or `int` as needed.

Step 8: Explore Distributions

1. Use `.value_counts()` to explore the distribution of categorical variables (e.g., `gender`, `grade`).

2. Plot the distributions of numeric variables (`math_score`, `reading_score`, etc.) using histograms (`sns.histplot`).
3. Use box plots (`sns.boxplot`) to detect outliers in numeric columns.

Step 9: Handle Outliers

1. Use box plots or the Interquartile Range (IQR) method to identify outliers in numeric columns.
2. Decide whether to remove, transform, or cap outliers.

Step 10: Encode Categorical Variables

1. Use one-hot encoding or label encoding to convert categorical columns into numeric formats for analysis.
2. Use `pd.get_dummies()` for one-hot encoding or `LabelEncoder` for label encoding.

Step 11: Correlation Analysis

1. Use `.corr()` to find correlations between numeric variables.
2. Visualize the correlation matrix using a heatmap (`sns.heatmap`).

Step 12: Investigate Relationships

1. Explore relationships between variables using scatter plots (`sns.scatterplot`).
   - Example: Compare `math_score` vs. `total_score`.
2. Use bar plots (`sns.barplot`) to analyze the impact of categorical variables (e.g., `gender` or `race_ethnicity`) on numeric scores.

Step 13: Feature Engineering

1. Create new features, if applicable:
   - Example: Add a `performance_ratio = total_score / max_score` column.
2. Bin numeric columns into categories (e.g., "low", "medium", "high") using `pd.cut()`.

Step 14: Summarize Findings

1. Summarize key insights from the data exploration.
2. Highlight any patterns, anomalies, or trends observed during preprocessing or EDA.

If you think any other related task, you can add in notebook.

Happy Learning 😊