A Data science project
using python

STUDENT PERFORMANCE ANALYSIS

Prepared by :

KOPPARTHI NISHYAVIKA

9848973998

kopparthinishyavika@gmail.com

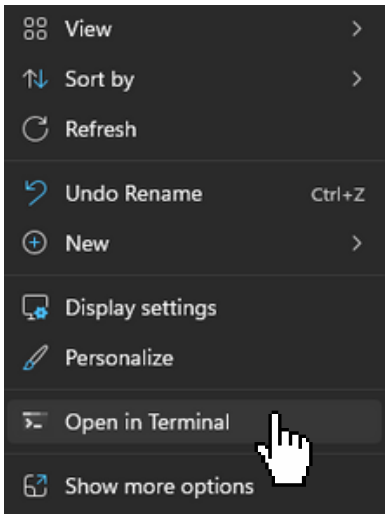
DECLARATION

I HEREBY DECLARE THAT THE PROJECT TITLED STUDENT PERFORMANCE ANALYSIS IS MY ORIGINAL WORK AND HAS NOT BEEN SUBMITTED ELSEWHERE.

SOFTWARE INSTALLATION/ ENVIRONMENT SET UP

1. DOWNLOAD PYTHON

- open a web browser
- go to www.python.org
- click on downloads
- select python 3.14(Latest version



2. RUN INSTALLER

- open the downloaded setup file
- tick the option add python to path
- click install now

3. VERIFY INSTALLATION

- open the command prompt

(click windows+R → cmd →/ desktop → right click → terminal)

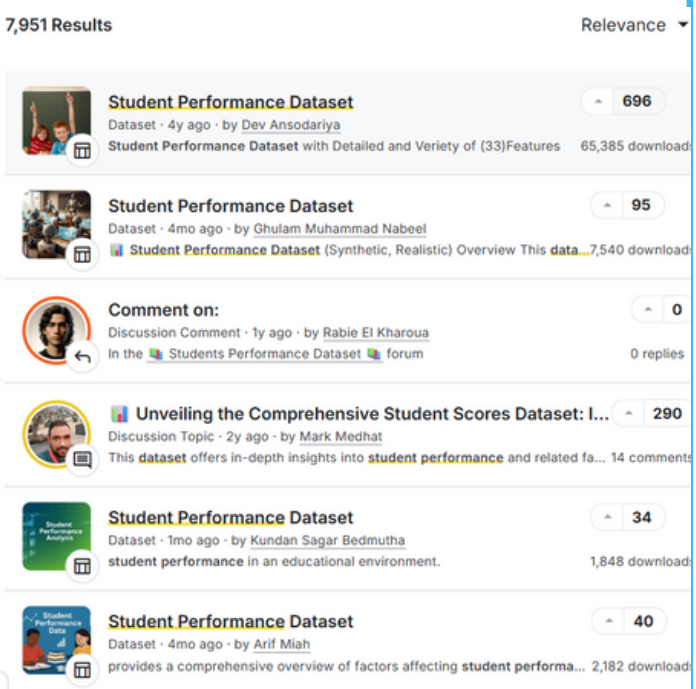
- type - python (python version should be displayed)

4. INSTALL REQUIRED LIBRARIES

- type - pip install pandas matplotlib numpy (incommand prompt)
- open jupyter notebook optional

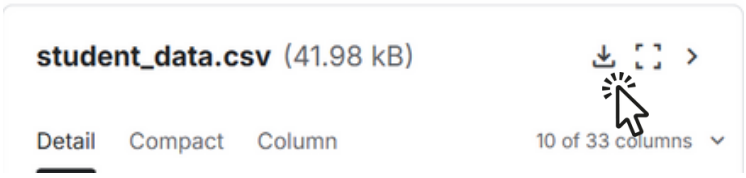
5. DATA SET INSTALLATION

- dataset source - kaggle.com
- search for target data set (student performance analysis)
- download the dataset



6. EXTRACT ZIP FILE

- this csv file is used for analysis in python
- place the file in project folder (optional)
- dataset is now ready for data analysis



1. **IMPORT REQUIRED LIBRARY -**

- Importing Pandas Library

Pandas Library is imported to handle and analyze tabular data efficiently

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

```
[3]: data = pd.read_csv("Student_Performance.csv")
```

2. LOAD THE DATASET

- Loading the data set

The student performance dataset is loaded into python using the read_csv() function

3. VIEW COLUMN NAMES

- Understanding Dataset Structure

The column names of the dataset are displayed to understand the available attributes for analysis.

4. IDENTIFY IMPORTANT COLUMNS

- Identifying Required Columns

From the available columns, only those related to academic performance and study behaviour are selected for further analysis

5. SELECT REQUIRED COLUMNS

- Selecting Relevent Features

Unnecessary columns are removed to simplify analysis and focus on important performance factors.

```
required_data = data[
    ['gender', 'study_hours', 'attendance_percentage',
     'internet_access', 'math_score', 'science_score',
     'english_score', 'overall_score', 'final_grade']]
```

6. DISPLAY SAMPLE DATA

- The head() function is used to display the first few records of the dataset. This helps in understanding the data values and structure.

```
data.head()
```

	student_id	age	gender	school_type	parent_education	study_hours	attendance_percentage	internet_access	travel_time
0	1	14	male	public	post graduate	3.1	84.3	yes	<15 min
1	2	18	female	public	graduate	3.7	87.8	yes	>60 min
2	3	17	female	private	post graduate	7.9	65.5	no	<15 min
3	4	16	other	public	high school	1.1	58.1	no	15-30 min
4	5	16	female	public	high school	1.3	61.0	yes	30-60 min

7. DATASET INFORMATION

- The info() function provides a summary of the dataset including number of rows, columns, data types, and missing values.

data.infor()

The dataset contains 25,000 records with 16 columns. All columns have non-null values, indicating no missing data.

```
[5]: data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 25000 entries, 0 to 24999
Data columns (total 16 columns):
 #   Column                Non-Null Count  Dtype
---  ------                --------------  -----
 0   student_id            25000 non-null  int64
 1   age                   25000 non-null  int64
 2   gender                25000 non-null  object
 3   school_type           25000 non-null  object
 4   parent_education      25000 non-null  object
 5   study_hours           25000 non-null  float64
 6   attendance_percentage 25000 non-null  float64
 7   internet_access       25000 non-null  object
 8   travel_time           25000 non-null  object
 9   extra_activities      25000 non-null  object
 10  study_method          25000 non-null  object
 11  math_score            25000 non-null  float64
 12  science_score         25000 non-null  float64
 13  english_score         25000 non-null  float64
 14  overall_score         25000 non-null  float64
 15  final_grade           25000 non-null  object
dtypes: float64(6), int64(2), object(8)
memory usage: 3.1+ MB
```

8. CHECK FOR MISSING VALUES

- This step checks whether the dataset contains any missing values.

```
data.isnull().sum()
```

The output shows zero missing values for all columns

9. CREATE ANALYSIS DATASET

- Creating a New Dataset for Analysis

A new dataset is created by selecting only subject scores, study hours, and attendance percentage for focused analysis.

```
data2 = data[
    ["math_score", "science_score", "english_score",
     "study_hours", "attendance_percentage"]
].copy()
```

10. CALCULATE AVERAGE MARKS

- The average marks are calculated by taking the mean of math, science, and English scores for each student

```
data2["average_marks"] = (
    data2["math_score"] +
    data2["science_score"] +
    data2["english_score"]
) / 3
```

11. DISPLAY UPDATED DATASET

- Viewing Updated Dataset

The updated dataset is displayed to verify the newly calculated average marks

```
data2.head()
```

```
[8]: data2.head()
```

[8]:

	math_score	science_score	english_score	study_hours	attendance_percentage	average_marks
0	42.7	55.4	57.0	3.1	84.3	51.700000
1	57.6	68.8	64.8	3.7	87.8	63.733333
2	84.8	95.0	79.2	7.9	65.5	86.333333
3	44.4	27.5	54.7	1.1	58.1	42.200000
4	8.9	32.7	30.0	1.3	61.0	23.866667

12. CLASSIFY STUDENT RESULT

- Pass and Fail Classification

Students are classified as Pass or Fail based on their average marks. Students scoring 40 or above are considered Pass.

```
data2["result"] = data2["average_marks"].apply(
    lambda x: "Pass" if x >= 40 else "Fail"
)
```

13. COUNT PASS AND FAIL STUDENTS

- Result Summary

The total number of students who passed and failed is calculated to understand overall performance.

```
data2["result"].value_counts()
```

```
[9]: data2["result"] = data2["average_marks"].apply(
         lambda x: "Pass" if x >= 40 else "Fail"
     )

[10]: data2["result"].value_counts()

[10]: result
      Pass    21827
      Fail     3173
      Name: count, dtype: int64
```

14. PASS VS FAIL ANALYSIS

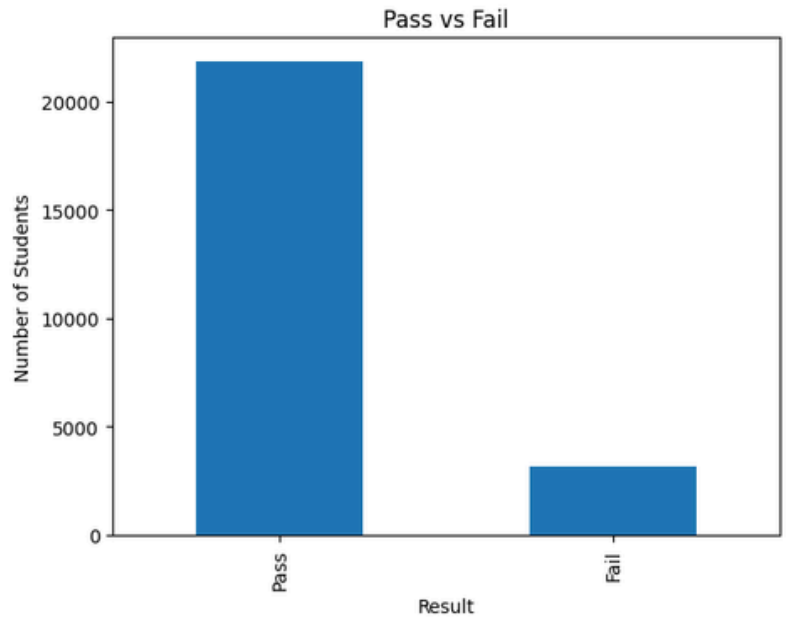
- Pass vs Fail Analysis

This bar chart shows the number of students who passed and failed based on the calculated average marks.

The number of students who passed is significantly higher than those who failed.

```
data2["result"].value_counts().plot(kind="bar")
plt.title("Pass vs Fail")
plt.xlabel("Result")
plt.ylabel("Number of Students")
plt.show()
```

```
[11]: data2["result"].value_counts().plot(kind="bar")
      plt.title("Pass vs Fail")
      plt.xlabel("Result")
      plt.ylabel("Number of Students")
      plt.show()
```

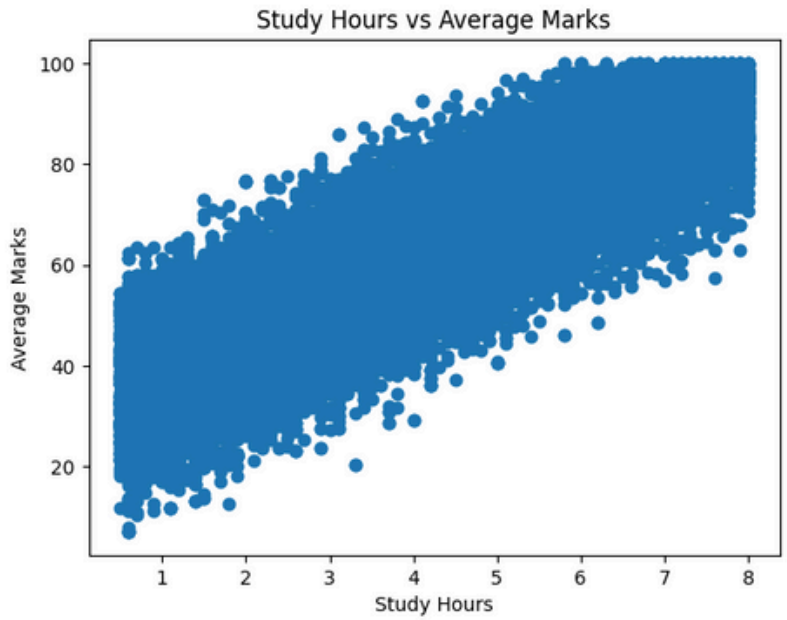


15. STUDY HOURS VS AVERAGE MARKS

- Relationship Between Study Hours and Performance

This scatter plot represents the relationship between study hours and average marks of students.

```
[12]: plt.scatter(data2["study_hours"], data2["average_marks"])
      plt.xlabel("Study Hours")
      plt.ylabel("Average Marks")
      plt.title("Study Hours vs Average Marks")
      plt.show()
```



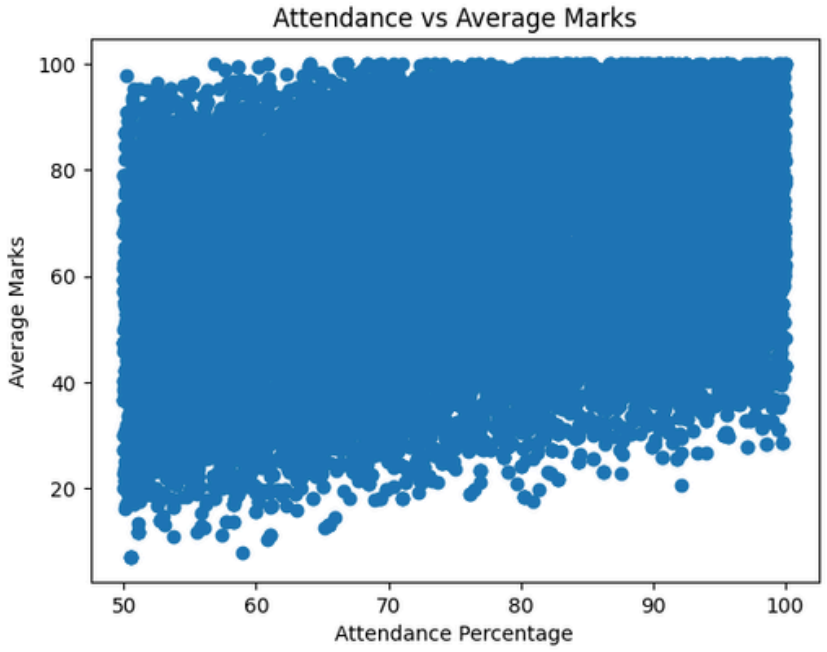
16. ATTENDANCE PERCENTAGE VS AVERAGE MARKS

- Impact of Attendance on Academic Performance

This scatter plot shows the relationship between attendance percentage and average marks of students.

```
plt.scatter(data2["attendance_percentage"], data2["average_marks"])
plt.xlabel("Attendance Percentage")
plt.ylabel("Average Marks")
plt.title("Attendance vs Average Marks")
plt.show()
```

```
[13]: plt.scatter(data2["attendance_percentage"], data2["average_marks"])
      plt.xlabel("Attendance Percentage")
      plt.ylabel("Average Marks")
      plt.title("Attendance vs Average Marks")
      plt.show()
```




```
[*]: data2.to_csv("Student_Performance_Analysis_Result.csv", index=False)
```

17. EXPORT FINAL RESULTS

- Saving Analysis Results

The final processed dataset is saved as a CSV file for future reference and reporting

```
data2.to_csv("Student_Performance_Analysis_Result.csv", index=False)
```

18. CONCLUSION

- Student performance is influenced by study hours and attendance
- Data analysis helped identify performance patterns
- Visualizations made insights easy to understand

19. FUTURE SCOPE

- Include more subjects and behavioral factors
- Apply machine learning models for prediction
- Use real-time student data

20. REFERENCES

- Kaggle Dataset
- Python Documentation
- Pandas Library Documentation

github - https://github.com/nishyavikakopparthi/student-performance-analysis.git

Thankyou

kopparthi nishyavika