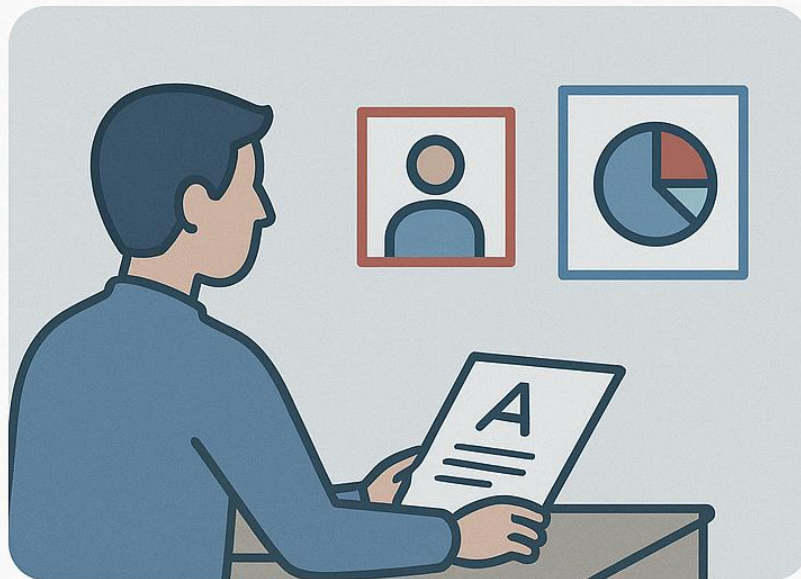


PROJECT: STUDENT PERFORMANCE ANALYSIS (DATA LABELING)



MOHAMED RAAIZ

Project: Student Performance Analysis

Introduction

Student Performance Analysis is a data-driven approach to **understand academic trends, evaluate student strengths, and identify areas for improvement**. By analyzing student scores, attendance, and subject-wise performance, educators can **make informed decisions to enhance learning outcomes**. This type of analysis is critical for **personalized interventions, curriculum improvements, and monitoring overall class performance**.

Key Points

1. Dataset Features:

- Student_ID: Unique identifier for each student
- Name: Student name
- Age: Age of the student
- Gender: Male/Female/Other
- Subject: Subject name (Math, Science, English, etc.)
- Score: Marks obtained
- Attendance (%): Percentage of classes attended

```
[1] # Step 1: Import Libraries
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

```
# Step 2: Create Dummy Dataset
data = {
    "Student_ID": [1,2,3,4,5,6,7,8,9,10],
    "Name": ["Alice", "Bob", "Charlie", "David", "Eva", "Frank", "Grace", "Hannah", "Ian", "Julia"],
    "Age": [15,16,15,17,16,15,16,17,15,16],
    "Gender": ["F", "M", "M", "M", "F", "M", "F", "F", "M", "F"],
    "Subject": ["Math", "Science", "English", "Math", "Science", "English", "Math", "Science", "English", "Math"],
    "Score": [88,92,75,85,95,80,78,89,70,90],
    "Attendance": [95,88,80,92,97,85,90,93,78,96]
}
df = pd.DataFrame(data)
```

```
[4] # Step 3: Average Scores per Subject
avg_scores = df.groupby("Subject")["Score"].mean()
print("Average Scores per Subject:\n", avg_scores)
```

```
Average Scores per Subject:
Subject
English    75.00
Math       85.25
Science    92.00
Name: Score, dtype: float64
```

```
[5] # Step 4: Top Performing Students
top_students = df.sort_values(by="Score", ascending=False)[["Name", "Subject", "Score"]].head(5)
print("\nTop Performing Students:\n", top_students)
```



Top Performing Students:

	Name	Subject	Score
4	Eva	Science	95
1	Bob	Science	92
9	Julia	Math	90
7	Hannah	Science	89
0	Alice	Math	88

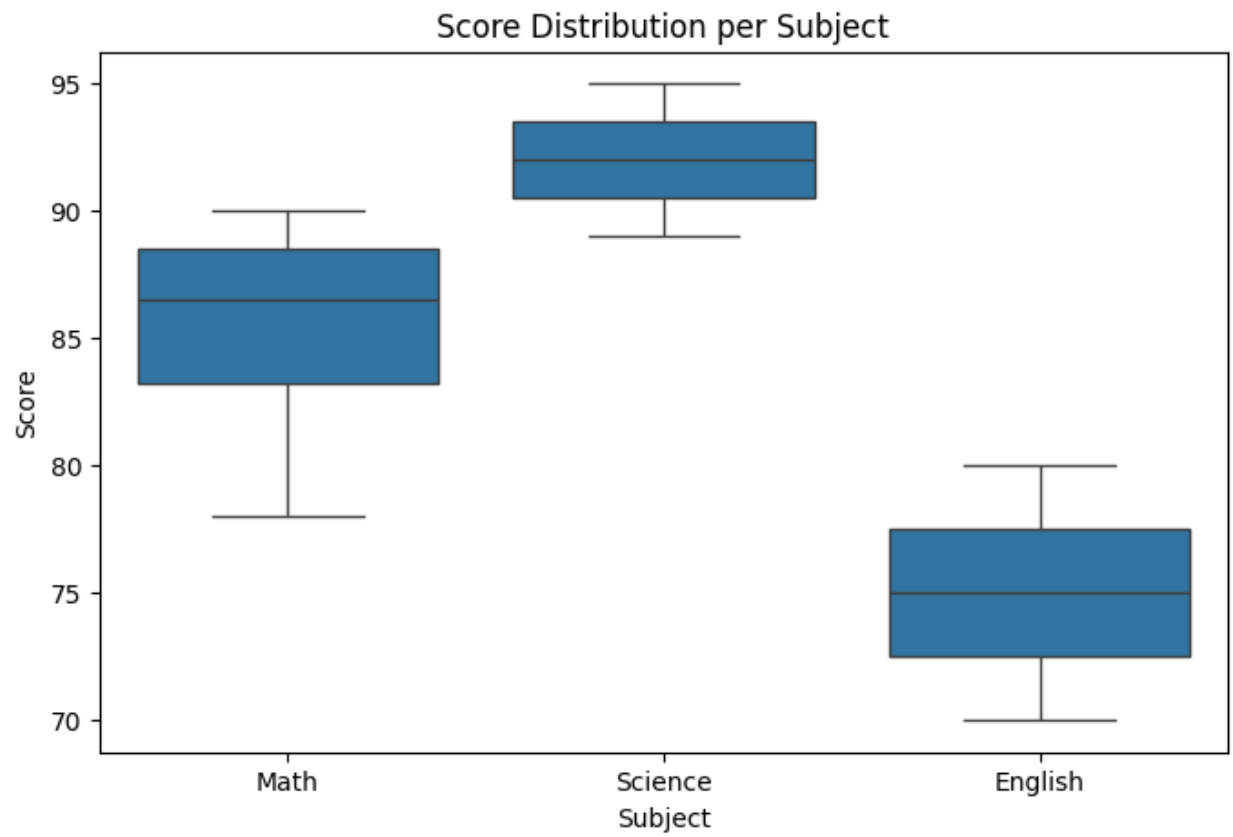
```
[6] # Step 5: Correlation between Attendance and Score
correlation = df["Attendance"].corr(df["Score"])
print("\nCorrelation between Attendance and Score:", correlation)
```

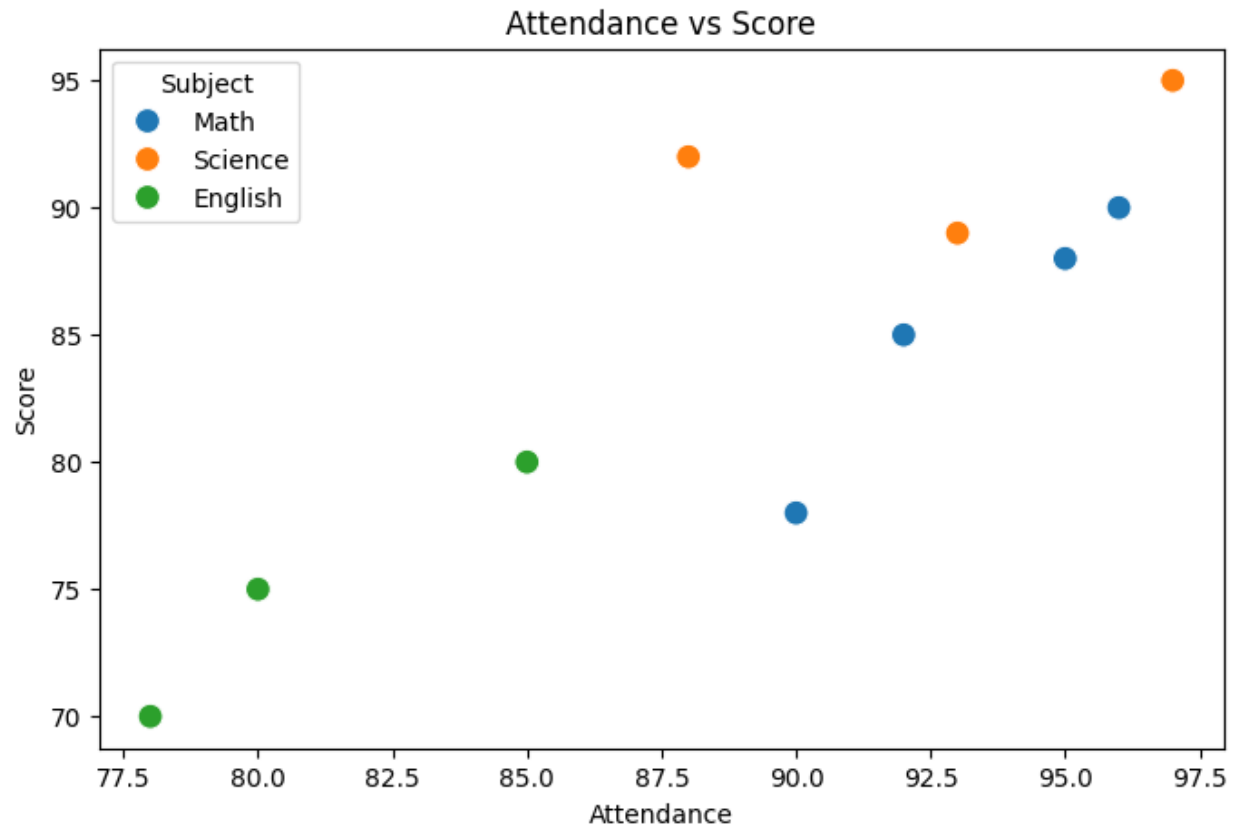


Correlation between Attendance and Score: 0.8575939939456327

```
[7] # Step 6: Visualize Score Distribution
plt.figure(figsize=(8,5))
sns.boxplot(x="Subject", y="Score", data=df)
plt.title("Score Distribution per Subject")
plt.show()

plt.figure(figsize=(8,5))
sns.scatterplot(x="Attendance", y="Score", hue="Subject", data=df, s=100)
plt.title("Attendance vs Score")
plt.show()
```





```
# Step 7: Predict Pass/Fail based on Score Threshold (e.g., 75)
df["Pass/Fail"] = df["Score"].apply(lambda x: "Pass" if x>=75 else "Fail")
print("\nPass/Fail Status:\n", df[["Name", "Score", "Pass/Fail"]])
```

Pass/Fail Status:

	Name	Score	Pass/Fail
0	Alice	88	Pass
1	Bob	92	Pass
2	Charlie	75	Pass
3	David	85	Pass
4	Eva	95	Pass
5	Frank	80	Pass
6	Grace	78	Pass
7	Hannah	89	Pass
8	Ian	70	Fail
9	Julia	90	Pass

2. Analysis Tasks:

- Calculate **average scores per subject**.
- Identify **top-performing students**.

- Check **correlation between attendance and scores**.
- Visualize **score distributions** using histograms or boxplots.
- Predict **pass/fail likelihood** based on attendance and scores.

3. **Key Keywords:**

- Data Cleaning, Aggregation, Correlation, Visualization, Predictive Analysis, Performance Metrics

Conclusion

Analyzing student performance provides **actionable insights for teachers and administrators**.

Key benefits include:

- Identifying **top-performing and struggling students**
- Understanding **subject-wise trends and patterns**
- Linking **attendance to academic performance**
- Supporting **data-driven interventions** to improve student outcomes

Outcome: Helps schools **enhance teaching strategies, monitor learning effectiveness, and improve overall academic results**.