

Fundamentals Of Data Science

CSA-0412

Name : Bhakthan Lokeshwaran

Reg No : 192424305

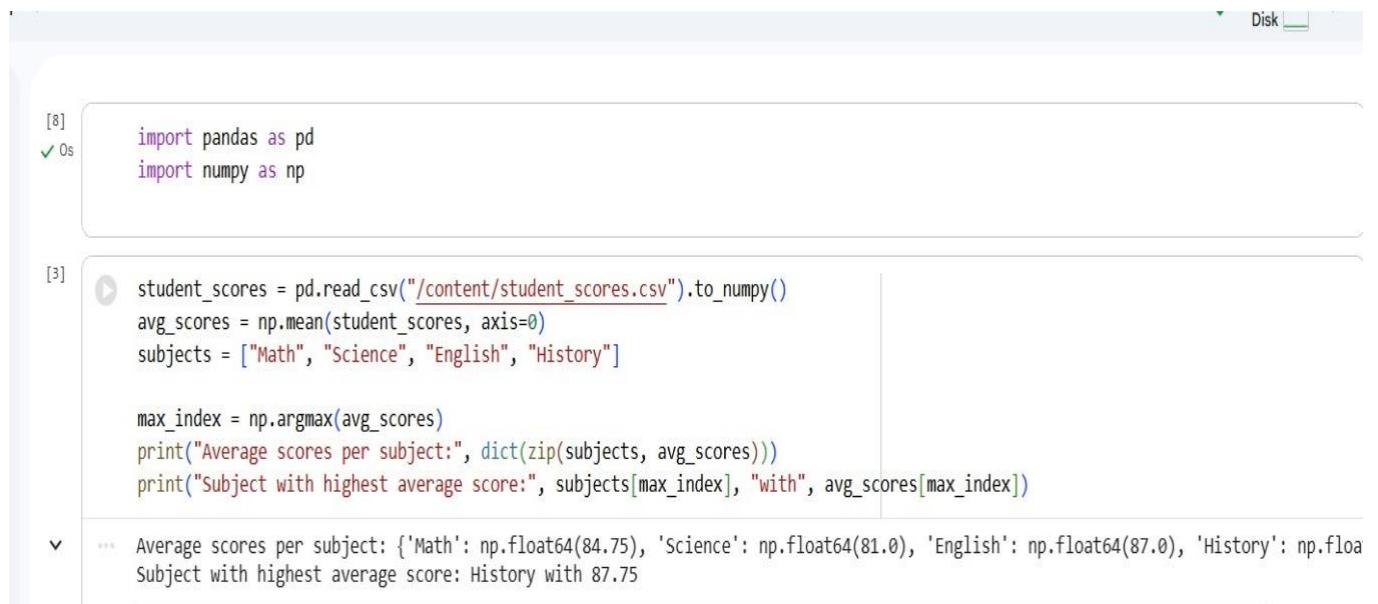
Date : 07-01-2026

1. Student Performance Data

Code:

```
student_scores = pd.read_csv("/content/student_scores.csv").to_numpy()  
  
avg_scores = np.mean(student_scores, axis=0)  
  
subjects = ["Math", "Science", "English", "History"]  
  
  
max_index = np.argmax(avg_scores)  
  
print("Average scores per subject:", dict(zip(subjects, avg_scores)))  
  
print("Subject with highest average score:", subjects[max_index], "with",  
avg_scores[max_index])
```

Output:



The screenshot shows a Jupyter Notebook interface with two code cells and their corresponding outputs.

Cell 1 (In [8]):

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

Cell 2 (In [3]):student_scores = pd.read_csv("/content/student_scores.csv").to_numpy()
avg_scores = np.mean(student_scores, axis=0)
subjects = ["Math", "Science", "English", "History"]

max_index = np.argmax(avg_scores)
print("Average scores per subject:", dict(zip(subjects, avg_scores)))
print("Subject with highest average score:", subjects[max_index], "with", avg_scores[max_index])

*** Average scores per subject: {'Math': np.float64(84.75), 'Science': np.float64(81.0), 'English': np.float64(87.0), 'History': np.float64(87.75)}
Subject with highest average score: History with 87.75

The output shows the average scores for each subject and the subject with the highest average score.

2. Product Sales Data

Code:

```
product_sales = pd.read_csv("/content/product_sales.csv").to_numpy()  
avg_price = np.mean(product_sales)  
print("\nAverage price of all products sold:", avg_price)
```

Output:

▶ product_sales = pd.read_csv("/content/product_sales.csv").to_numpy()

```
avg_price = np.mean(product_sales)  
print("\nAverage price of all products sold:", avg_price)
```

...

Average price of all products sold: 120.55555555555556

3. House Data

Code:

```
house_data = pd.read_csv("/content/house_data.csv").to_numpy()  
houses_5plus = house_data[house_data[:, 0] > 4]  
avg_sale_price = np.mean(houses_5plus[:, 2])  
print("\nAverage sale price of houses with more than 4 bedrooms:", avg_sale_price)
```

Output:

```
▶ house_data = pd.read_csv("/content/house_data.csv").to_numpy()  
  
houses_5plus = house_data[house_data[:, 0] > 4]  
  
avg_sale_price = np.mean(houses_5plus[:, 2])  
print("\nAverage sale price of houses with more than 4 bedrooms:", avg_sale_price)  
  
...  
Average sale price of houses with more than 4 bedrooms: 475000.0
```

4. Quarterly Sales Data

Code:

```
sales_data = pd.read_csv("/content/sales_data.csv").to_numpy().flatten()
total_sales = np.sum(sales_data)
percentage_increase = ((sales_data[3] - sales_data[0]) / sales_data[0]) * 100
print("\nTotal sales for the year:", total_sales)
print("Percentage increase from Q1 to Q4:", percentage_increase, "%")
```

Output:

```
sales_data = pd.read_csv("/content/sales_data.csv").to_numpy().flatten()
total_sales = np.sum(sales_data)
percentage_increase = ((sales_data[3] - sales_data[0]) / sales_data[0]) * 100
print("\nTotal sales for the year:", total_sales)
print("Percentage increase from Q1 to Q4:", percentage_increase, "%")
```

```
Total sales for the year: 66500
Percentage increase from Q1 to Q4: 20.0 %
```

5. Fuel Efficiency Data

Code:

```
fuel_efficiency = pd.read_csv("/content/fuel_efficiency.csv").to_numpy().flatten()
avg_efficiency = np.mean(fuel_efficiency)
percentage_improvement = ((fuel_efficiency[1] - fuel_efficiency[0]) / fuel_efficiency[0]) * 100
print("\nAverage fuel efficiency:", avg_efficiency)
print("Percentage improvement from car 1 to car 2:", percentage_improvement, "%")
```

Output:

```
▶ fuel_efficiency = pd.read_csv("/content/fuel_efficiency.csv").to_numpy().flatten()
avg_efficiency = np.mean(fuel_efficiency)
percentage_improvement = ((fuel_efficiency[1] - fuel_efficiency[0]) / fuel_efficiency[0]) * 100
print("\nAverage fuel efficiency:", avg_efficiency)
print("Percentage improvement from car 1 to car 2:", percentage_improvement, "%")
...
Average fuel efficiency: 28.4
Percentage improvement from car 1 to car 2: 20.0 %
```