REPORT

Programming For Artificial Intelligence

Sneha Mini Biju

Student ID: x23323701

1. **Problem 1:**

   i) **Task 1:**
   API Integration: Here I used the requests library to send requests. ‘api_key’ and ‘date’ are passed as parameters.
   Response Standardization:
   Here I used a helper function format_apod_data was implemented to structure the data.
   Error Handling:
   Here I used a try-except to identify errors such as invalid API responses or network issues.

   ii) **Task 2**:
   Here I used a for loop that will iterate from a date to another.
   For each iteration, call the get function to retrieve the APOD data.
   Rate Limiting:
   Time.sleep(1) is used to delay API calls further apart than the rate limit policy.
   Data Storage:
   The fetched data is appended to a list; the format_apod_data() function is used to format it.

   iii) **Task 3:** Saving data to JSON file
   At the end of the range, all data is written to a JSON file. If it exists, it do not get overwritten; instead, new data is appended to it.
   Error Handling:
   Errors while fetching data or saving to a file are caught and logged.

2. **Problem 2:**

   i) Task 1
   Here I copied data from apod_data.json into a Python dictionary. The program shall handle potential file-related errors.

   Error Handling:
   Exceptions such as file not found, permission denied, or invalid JSON format were handled with a try-except block.

In each case, an informative error message was printed out to help in debugging.

ii) **Task 2:**

Here I Counted all the images and videos with the help of for loop and if-else cases
Then I determined which entry has the longest explanation.For each entry, a check was made on its type to increment either the counter of images or videos.
The explanation length was checked to obtain the entry with the longest explanation
Total Images: 314
Total Videos: 27.
The date and contents of the longest explanation were displayed.

iii) **Task 3:**

Here I Extracted specific and wrote these into a CSV file. JSON data read and a subset of fields extracted for CSV output.If the CSV file already existed, new entries were appended; otherwise, headers were included in the new file.The outcome is a CSV file, apod_summary.csv, was created and updated with formatted data from the JSON file, ensuring no duplication or data loss.

3. **Problem 3:**

i) Task 1:

For Generated integers ranging from 10 to 100 using np.random.randint(10, 101).
Iterated through each row to make the sum even.If needed, it changed the first element of the row to have the sum even without letting values get out of the specific range.

Total Sum Normalization:

Check the total sum of all elements. Each row's sum was verified to be even. The total sum of all array values was confirmed as a multiple of 5.

ii) Task 2:

Here I used indexing to access and print elements divisible by 3 and 5
Changed all values larger than 75 with the mean.

iii)Task 3:

I calculated some statistical operations on the array.

Mean: Calculated the mean.

Standard Deviation: Data variability was measured.

Median: Found the middle value in the array.

Column-wise Variance: Calculated the variance for each column to observe data spread across rows.

4. **Problem 4:**

i)Task 1:

Loaded Iris dataset into a Pandas DataFrame from iris.csv file provided.

Performed shallow exploration: Total data points: 150.

Data types: float64 for numeric measurements, object for species name.

Column names: 'sepal.length', 'sepal.width', 'petal.length', 'petal.width', 'variety'.

Number of flower species: 3 (['Setosa', 'Versicolor', 'Virginica'].

ii) Task 2:

The errors in two rows (35th and 38th) were corrected

The corrected rows were displayed to ensure the changes were applied correctly.

iii)Task 3:

New Features mentioned in the question using division mechanism.

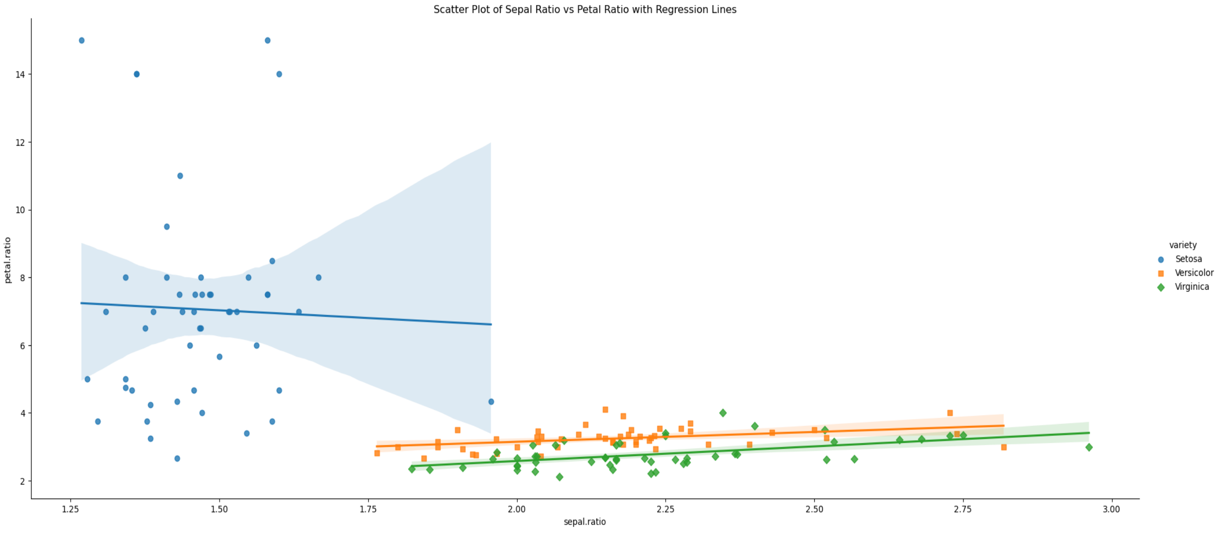
The updated DataFrame was stored in the same folder..

iv) Task 4:

For this I selected numerical columns only.
High Correlation between petal.length and petal.width(correlation: 0.962878).
Highest Negative Correlation between sepal.width and sepal.ratio(correlation: -0.746705).

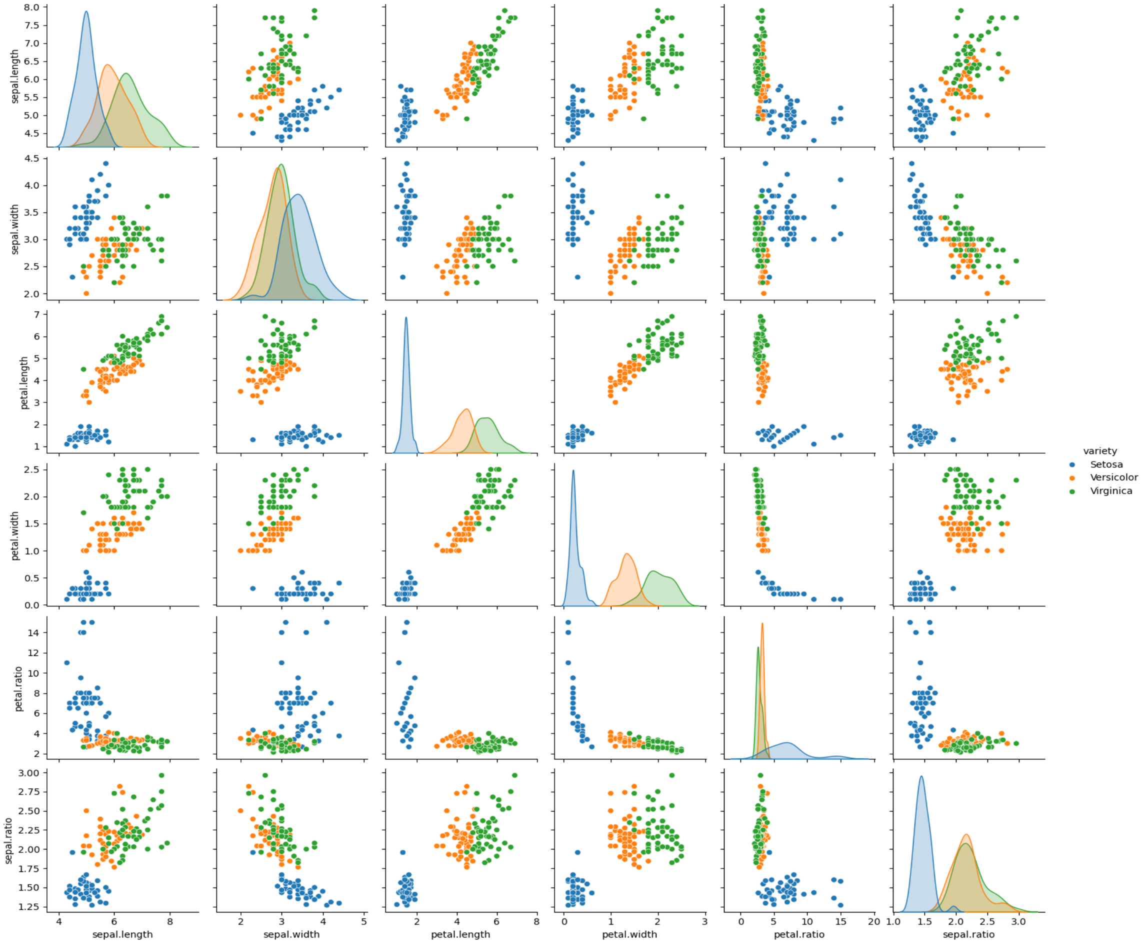
v) Task 5:

I plot a scatterplot with different point colors for each species, which is able to find regression.



vi) Task 6:

A pair plot using Seaborn was created.The points were colored by species for better visualization of feature relationships.


sepal.length
sepal.width
petal.length
petal.width
petal.ratio
sepal.ratio
sepal.length
sepal.width
petal.length
petal.width
petal.ratio
sepal.ratio
variety
Setosa
Versicolor
Virginica