System Requirements Specification Index

For

Create a NumPy array representing daily sales for a week and perform indexing, slicing, and reshaping

(Topic: NumPy Arrays)

Version 1.0

Sales Data Analysis Console

Project Abstract

The Sales Data Analysis Console is a Python-based application that analyzes sales data for a retail company.
This tool helps the company evaluate and gain insights into product sales performance, providing essential metrics like total revenue,
top-selling products, and normalized sales. The system uses NumPy and pandas to process daily product sales data, and provides an easy-to-use
interface for calculating various sales-related metrics. It also reshapes sales data into weekly format for reporting purposes.
This application plays a crucial role in making informed business decisions based on sales performance.

Business Requirements:

1. The system must calculate the total revenue for each product by multiplying units sold by their respective prices.
2. The system must identify the top N selling products based on units sold.
3. The system must reshape daily sales data into a weekly format (7 days per week).
4. The system must retrieve sales data for specific products upon request.
5. The system must normalize sales data to a 0-1 scale for easier comparison.

Constraints

Input Requirements
- Product IDs: Must be a list of unique product identifiers (integer values).
- Units Sold: Must be a list of sales figures (integer or float) representing units sold for each product.
- Prices: Must be a list of product prices (float values), one for each product in the product IDs list.

Data Format:
- Sales data must be inputted as three lists: product_ids, units_sold, and prices.

Function Requirements

1. Total Revenue Calculation:

- Calculate the total revenue per product using the formula: total_revenue = units_sold * prices.
- Return a list of total revenue values for each product.

2. Top Selling Products:
- Retrieve the top N selling products based on units sold.
- Sort the products in descending order by the number of units sold and return the IDs of the top N products.

3. Weekly Sales Reshaping:
- Reshape the daily sales data into a weekly format. The data must be in multiples of 7 for this operation to work.
- Return a 2D array of weekly sales data (7 days per week).

4. Product Sales Retrieval:
- Retrieve sales data for a specific product using its product ID.

5. Sales Normalization:
- Normalize the sales data between 0 and 1, where the minimum sales value is 0 and the maximum is 1.
- Return the normalized sales data.

Output Constraints

- The output for total revenue must show revenue for each product.
- The output for top-selling products must show a list of the top N product IDs.
- The output for weekly sales must display the reshaped weekly sales data in a 2D format.
- The output for product sales must display the units sold for the queried product.
- The output for normalized sales must show the normalized sales data, where all values are between 0 and 1.

Required Output Format:

1. Total Revenue:
- Display: Product ID: {product_id}, Total Revenue: {revenue}

2. Top Selling Products:
- Display: Top N Selling Products: {product_id1, product_id2, ...}

3. Weekly Sales:
- Display the reshaped weekly sales data in a 2D format (each row

representing sales for a week).

4. Product Sales:
- Display: Product ID: {product_id}, Sales: {sales}

5. Normalized Sales:
- Display: Normalized Sales: {normalized_sales}

Template Code Structure:

1. Initialization of Sales Data:
- Initialize the product IDs, units sold, and prices as NumPy arrays.

2. Methods for Sales Analysis:
- total_revenue(): Calculate the total revenue for each product.
- top_selling_products(): Get the top N selling products based on units sold.
- weekly_sales(): Reshape daily sales data into a weekly format.
- get_product_sales(): Retrieve the sales data for a specific product.
- normalize_sales(): Normalize sales data.
- apply_normalize_sales(): Use the apply method to normalize sales.

3. Execution Flow:
- Call the functions to analyze sales data, such as calculating total revenue, normalizing sales, reshaping data, and identifying top-selling products.

Execution Steps to Follow:

- All actions like build, compile, running application, running test cases will be through Command Terminal.
- To open the command terminal the test takers, need to go to Application menu (Three horizontal lines at left top) -> Terminal -> New Terminal
- This editor Auto Saves the code
- If you want to exit(logout) and continue the coding later anytime (using Save & Exit option on Assessment Landing Page) then you need to use **CTRL+Shift+B** -command compulsorily on code IDE. This will push or save the updated contents in the internal git/repository. Else the code will not be available in the next login.
- These are time bound assessments the timer would stop if you logout and while logging in back using the same credentials the timer would resume from the same time it was stopped from the previous logout.
- To setup environment:

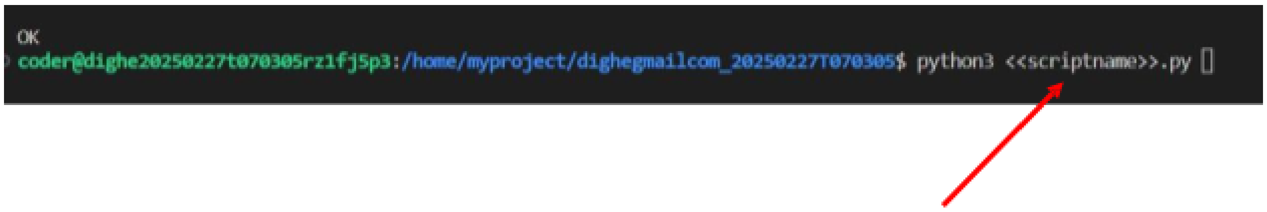
  You can run the application without importing any packages
- To launch application:
  python3 mainclass.py
- To run Test cases:

  python3 -m unittest

- Before Final Submission also, you need to use **CTRL+Shift+B** - command compulsorily on code IDE, before final submission as well. This will push or save the updated contents in the internal git/repository, and will be used to evaluate the code quality.

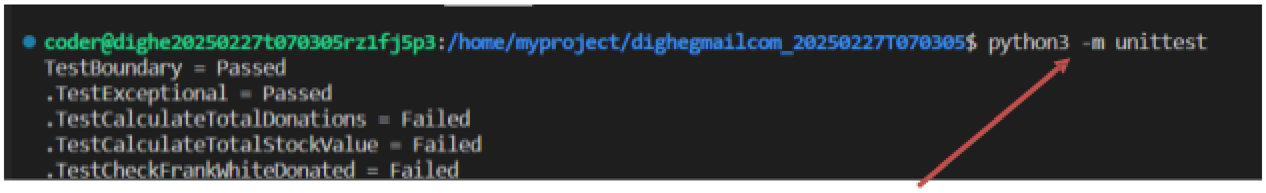
Screen shot to run the program

To run the application



```
OK
coder@dighe20250227t070305rz1fj5p3:/home/myproject/dighegmailcom_20250227T070305$ python3 <<scriptname>>.py
```

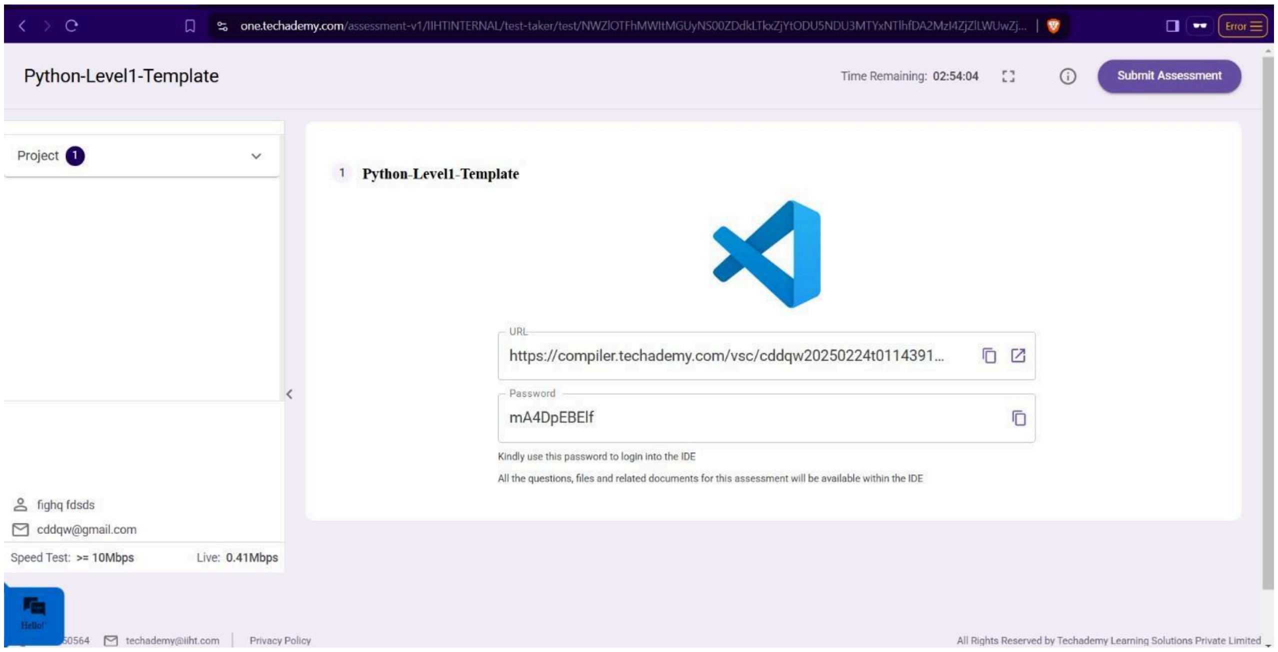
python3 mainclass.py



```
● coder@dighe20250227t070305rz1fj5p3:/home/myproject/dighegmailcom_20250227T070305$ python3 -m unittest
  TestBoundary = Passed
  .TestExceptional = Passed
  .TestCalculateTotalDonations = Failed
  .TestCalculateTotalStockValue = Failed
  .TestCheckFrankWhiteDonated = Failed
```

To run the testcase

python3 -m unittest


one.techademy.com/assessment-v1/IIHTINTERNAL/test-taker/test/NWZlOTFhMWItMGUyNS00ZDdkLTkxZjYtODU5NDU3MTYxNTlhfDA2Mzl4ZjZiLWUwZj...
Python-Level1-Template
Time Remaining: 02:54:04
Submit Assessment
Project 1
1 Python-Level1-Template
URL
https://compiler.techademy.com/vsc/cddqw20250224t0114391...
Password
mA4DpEBElf
Kindly use this password to login into the IDE
All the questions, files and related documents for this assessment will be available within the IDE
fighq fdsds
cddqw@gmail.com
Speed Test: >= 10Mbps
Live: 0.41Mbps
Hello!
50564
techademy@iiht.com
Privacy Policy
All Rights Reserved by Techademy Learning Solutions Private Limited

- **Once you are done with development and ready with submission, you may navigate to the previous tab and submit the workspace. It is mandatory to click on "Submit Assessment" after you are done with code.**