COL290: DESIGN PRACTICES

ASSIGNMENT 1 – SUBTASK 1

OBJECTIVE –

Given a stock symbol (say SYM) and number of years (say x) as input from make, need to extract the SYM stock's data (specific columns) for the last x years and save it in different file formats such as csv, txt etc. Then, compare the write time and space used by each file types using a graph to plot the points as space used vs time taken.

LIBRARIES/MODULES USED –

1. jugaad-data – for extracting data for the stock from NSE

2. Pandas – for storing the data extracted into different file formats and filtering the columns.

3. datetime & dateutil.relativedelta – for using x and calculating start date and end date of the data to be written

4. matplotlib – for plotting, storing the graph and labelling axes, points and graph.

5. sys – for taking sys args

6. os – for calculating space used by each file

7. time – for calculating time taken to write in the file

DESIGN CHOICES -

- Maintained 2 lists – sizes and times
  Used these 2 lists to store the size and time for each file type in a ordered way so that it can be retrieved using index
- For the project to remain scalable for multiple file types, did not hardcode the file names and plotted values, maintained a list of file types and used it to create file names
- Handled erroneous inputs of wrong value for number of years and terminated program with the message to input correct value
- For each file type, calculate time taken in ms to write in file using start time and end time difference and stored it in times list
- For each file type calculate space required by it in megabytes and stored it in sizes list.
- Used these 2 lists as axes for the graph to be plotted.

Plotted a point in time vs size space for each file type and marked it with a different color stored in color list.

INSIGHT ABOUT FILE TYPES

1. csv and txt – they have almost same write speeds and sizes , they occupy low space and low time relatively

2. html – time taken and space used is very high relative to others

3. json – time taken is least but size used is higher than csv, txt, and parquet

4. parquet – space used is second least but time taken is higher than csv, json and txt

5. feather - size used was the least and time taken is higher than json only.

1. **txt (Text File):**
   Write Speed: Generally fast.
   Space Used: Text files are relatively space-efficient, especially for simple data without additional formatting.

2. **json (JSON - JavaScript Object Notation):**
   Write Speed: Moderate.
   Space Used: JSON files can be human-readable but may use more space compared to binary formats due to the textual representation of data.

3. **csv (Comma-Separated Values):**
   Write Speed: Fast.
   Space Used: CSV files are compact and efficient for tabular data. However, they may not support complex nested structures like JSON.

4. **feather:**
   Write Speed: Fast.
   Space Used: Feather is a binary columnar data format designed for high performance. It is generally more space-efficient than text-based formats.

5. **parquet:**
   Write Speed: Moderate to fast.
   Space Used: Parquet is a columnar storage format optimized for use with big data processing frameworks. It can offer good compression and is suitable for analytics workloads.

6. **html (Hypertext Markup Language):**
   Write Speed: Moderate.

Space Used: HTML is primarily used for representing structured documents on the web. While not optimized for data storage, it can include various multimedia elements and might not be as space-efficient as specialized data formats.

LEARNINGS

- Learnt about Makefile for the first time.
- Learnt to architect the assignment and plan about what needs to be done for the first time since col106 assignments were heavily detailed with little to no flexibility for implementations.
- Learnt to use modules such as matplotlib and pandas for the first time.
- Learnt about os, datetime and time modules.
- Understood the importance of version management.