**Rakshitha Ramachandra K00302101**

### **What I Learned About Applying Python to Data Analysis**

Python has emerged as one of the most powerful tools for data analysis. Thus, it can power through real-life datasets, compute key statistics of datasets, make efficiency improvements, and reuse code. This project tries to explore the use of Python for the analysis of datasets, starting from its inbuilt features to the use of libraries like Pandas.

### **Learning Fundamentals of Data Analysis in Python**

First of all, I applied the inbuilt Python features for data analysis, loops, and lists. Loading data with the csv module, I computed key metrics such as averages, minimums, maximums, and percentiles on numeric data. I also performed occurrence counts on at least categorical data in order to find trends inside the customer's preferences. Hands-on work has brought me to understand how it is processed and structured.

Although this could be done manually for small datasets, it offered a critical disadvantage in the fact that, for larger datasets, this became entirely unfeasible. This inefficiency showed the necessity for higher tooling to increase speed and productivity.

**Performance and Efficiency with Pandas**

The second task introduced Pandas, a Python optimization for data analysis. Switching to Pandas greatly improved performance. For example:

* Task 1 (Raw Python): Execution time was 0.53 seconds on the dataset.
* Task 2 (Pandas): Execution time reduced to 0.21 seconds on the very same data set.
* Larger Dataset: Pandas performed very well on this larger dataset, processing it in 0.24 seconds.

Pandas is efficient because of its inbuilt functions, for instance, describe() and value\_counts(), which compute complex things with ease, and column-wise computation of data without the need for explicit-for loops. This transition also reduced the quantity of code and showed the power of third-party libraries for handling big data.

**Reusability, Readaptation of Code**

The third task was to apply the same code to the new e-commerce dataset. The underlying logic didn't change much; I had only to change column name and identification of data as numerical or categorial. Such reusability witnessed the strength of the code but at the same time showed one point of modification for it: automation of column type detection. Implementation of this feature would allow the script to easily adapt to all kinds of datasets, speeding up the analysis process.

**How ChatGPT Improved My Learning**

I combined the first task-consulting with ChatGPT-to further my learning about Python and statistical concepts. It gave clear guidance on:

* How to use loops to process data files line by line.
* Loading and manipulating datasets by utilizing Python's csv module.
* In this regard, statistical formulae encompass computation of averages, variability measures, and percentiles of distribution.

Such assistance made my code accurate, further building my confidence in understanding and providing solutions with Python related to data analysis.

**Key Takeaways**

* Python’s Strengths: Python is beginner-friendly yet powerful, making it ideal for analyzing datasets of varying complexity.
* The Value of Libraries: Pandas accelerates your analysis by providing efficient, prebuilt functions and optimizing memory use.
* Efficiency Gains: Pandas went from 0.53 seconds down from raw Python, to 0.21 seconds, and held longer for bigger datasets.
* Reusability: The initial code I wrote was adaptable to a new dataset; however, automation in column detection can be taken further for flexibility.

**Conclusion**

The project really shows how Python is efficiently and scaleably doing data analysis. Writing raw code gave further insight into the basics, while leveraging Pandas showed the importance of using specialized tools. Programming skills combined with libraries and AI tools such as ChatGPT mean analysts can approach real-world data challenges with assurance.

.