International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 08 Issue: 10 | Oct - 2024 SJIF Rating: 8.448 ISSN: 2582-3930

Comparative Study of Java and Python Languages with Reference to Data Science

Asst. Prof. Pratik Adhav
Department of Computer Science and Applications, Padmashri Manibhai Desai Mahavidyalaya, Uruli Kanchan, Pune 412202,Maharashtra, India.
adhavpratik77@gmail.com

Abstract

This research provides a comprehensive comparison & analysis between Java and Python, focusing on their application in data science. Key aspects such as encapsulation, AI capabilities, software complexity metrics (Halstead), data statistics, and algorithmic performance (Bubble Sort) are analyzed. This study aims to guide developers in choosing the appropriate language for their data science projects by evaluating performance, memory management, multithreading, and ecosystem richness. The findings are supported by empirical data, benchmarks, and detailed analysis, emphasizing when and how each language excels.

1. Introduction

Choosing the right programming language for data science applications is a critical decision that impacts efficiency, scalability, and productivity.(Sarda, 2015) While Python has gained popularity due to its simplicity and robust libraries, Java offers significant advantages in performance, particularly in resource-intensive tasks.(Sarda, 2015) This research aims to compare Java and Python in various data science contexts, providing insights into their strengths and weaknesses. The analysis is based on benchmarks, calculations, and real-world scenarios to determine the most suitable language for different data science tasks.

2. Background

Python is widely adopted in data science due to its ease of use, extensive libraries (such as NumPy, pandas, and TensorFlow), and active community support. On the other hand, Java is known for its performance, stability, and ability to handle large-scale, concurrent tasks effectively.

While both languages are popular, they have distinct differences when it comes to memory management, multithreading capabilities, and execution speed. Previous research has primarily focused on comparing their usability and availability of libraries in data science, many of them have analyzed using various aspects of programming like bubble sort, AI, Encapsulation, but there is limited research on their performance efficiency under real-world data processing conditions, particularly when handling large datasets. Understanding these performance characteristics is essential for selecting the appropriate language for specific data science applications.

Given this context, this research seeks to address the gap by conducting an empirical analysis of Java and Python's performance in handling large datasets. The study specifically focuses on memory management, multithreading efficiency, and execution speed—key factors that determine the overall performance of a data science application. By analyzing these aspects, the research aims to determine which programming language

between Java and Python works better in the field of data science and to provide practical insights for data scientists and developers, helping them make informed decisions based on the requirements of their projects.

3. Literature Review

The role of programming languages in data science is crucial, with Python and Java standing out due to their distinct advantages. (Shaw, 2023) explores Python's adaptability and efficiency in modern computing architectures, highlighting its suitability for various data science tasks. This is complemented by (Gupta, 2021), who provides a comparative review of Java and Python, illustrating their relative strengths and weaknesses in data science applications. (Omer et al., 2021) delve into encapsulation in both languages, noting Java's strict encapsulation compared to Python's more flexible approach.

Performance aspects are further detailed by (Acevedo, 2021), who contrasts Java and Python in AI contexts, underscoring Python's edge in rapid prototyping. (Abdulkareem and Abboud, 2021) evaluate Python and Java using software complexity metrics, revealing Python's ease of use and Java's superior performance in complex systems. (Siebers and Baydag, 2021) provide insights into how Java and Python manage arrays, contributing to a better understanding of their performance differences.

Practical comparisons are highlighted by (Insanudin, 2019), who examines sorting algorithms in Python and Java, demonstrating Python's simplicity and Java's efficiency.(DZone, 2024) offers contemporary data on the statistical and analytical performance of both languages, reflecting current industry trends.

Foundational texts such as (Guttag, 2016) and (Bloch, 2017) offer essential knowledge on Python and Java programming. Further, (Ramalho, 2015) and (Onks, 2014) provide detailed insights into Python's performance and Java's optimization techniques, enriching the overall understanding of these languages in the context of data science.

4. Methodology of Study

A) Study of Existing Material

I) Java vs Python in Encapsulation

This section explores encapsulation in Java and Python, highlighting its significance in data integrity, code maintainability, and access to class attributes.(Omer, 2021) Despite its complexity and code overhead, Java's encapsulation mechanism is superior due to its strict control over data access, making it a better choice for projects requiring data protection and code stability.(Omer, 2021),(Joshua, 2017)

II) Java vs Python in AI

Java is a strong contender for AI applications due to its robust performance metrics and ability to handle large-scale applications.(Acevedo, 2021) Python, with its extensive library support, is ideal for prototyping and smaller-scale AI models. Python's simplicity and easy-to-use code make it less time-consuming to develop AI applications.(Luciano, 2015) Although Java has a memory consumption issue, its performance is generally better than Python.(Acevedo, 2021) Despite these strengths, it is difficult to consider Java as a better programming tool in AI research. (Acevedo, 2021), (Luciano, 2015).

III) Evaluating Java and Python Using Halstead Metrics

A study evaluating the performance of Halstead metrics, a complexity measure for programming languages like Python and Java, found that Python is user-friendly and the simplest, while Java is powerful, modern, quicker, and more complex.(Abdulkareem & Abboud, 2021),(Scott, 2014)

IV) Data Statistics and Analysis with Java and Python

Data statistics and analysis are central to data science, and both Python and Java offer powerful tools for this.(Dzone, 2024),(Micha & Ozsvald, 2014) Python's Pandas library simplifies data manipulation, while Java's Apache Spark is designed for handling large datasets in distributed environments. (Dzone, 2024),(Micha & Ozsvald, 2014).

Java's Approach

Java Streams is a robust data manipulation framework with a declarative programming style, support for selection, filtering, grouping, and aggregations, and efficient operation through parallelism, laziness, and short-circuit operations. (Dzone, 2024) Its key advantage is its simple mechanism for parallel processing using multithreading. (Dzone, 2024),(Micha & Ozsvald, 2014)

Python's Approach

Python is a powerful tool for data statistics and analysis, with the Pandas library being particularly effective for processing large amounts of tabular data. (Micha & Ozsvald, 2014) Running code serially in Python is faster than in Java, but multiprocessing can degrade performance. Java's Streams and Python's Pandas offer solid frameworks for performance and scaling, making them ideal for complex calculations and quick results. (Dzone, 2024),(Micha & Ozsvald, 2014)

V) Java vs Python in Data Science Using Bubble Sort

Python is preferred in data science due to its simplicity, readability, and library ecosystem, while Java is known for its performance and scalability, making it ideal for experimentation and large-scale applications.(Sarda, 2015),(Insanudin, 2019) Python's concise syntax simplifies algorithm implementation in educational settings, while Java's statically-typed nature enhances performance for larger datasets or performance-critical applications. (Insanudin, 2019) Python is ideal for Bubble Sort in data science due to its ease of use and quick implementation, but Java may be the preferred choice for performance-critical scenarios. (Sarda, 2015),(Insanudin, 2019)

B) Experimental Analysis

I) In-depth Memory Management Evaluation

Memory management plays a critical role in data science, particularly with large datasets. We profiled memory usage in both Python and Java during the processing of a 5 GB dataset. The results showed significant differences between the two languages.

Table 1: Detailed Memory Usage Table

Java	**Python**
Initial Memory Usage: **5.483** GB	Initial Memory Usage: **4.602** GB
Memory Usage after processing: **5.492** GB	Memory Usage after processing: **4.769** GB

II) Multithreading Efficiency and Resource Utilization

Given the importance of multithreading in optimizing CPU-bound tasks, we compared the performance of Python and Java in multithreaded data processing and machine learning tasks. We evaluated their efficiency by performing operations on 5GB Dataset. The result revealed that Java outperforms Python in multithreading.

Table 2: Multithreading Efficiency Table

Task	**Python**	**Java**
Multithreading Efficiency Test	**Completed in 3.398 seconds**	**Completed in 0.054 seconds**

III) Performance Benchmarking: Execution Speed

We focused on evaluating the execution speed of Java and Python. The tests were conducted on 5GB dataset, and the results showed that Java consistently outperformed Python in execution speed, particularly in scenarios that required intensive computation.

Table 3: Execution Speed Comparison and Multithreading Efficiency Test (in seconds)

Task	**Python**	**Java**
Execution Speed Test	**Completed in 30.828 seconds**	**Completed in 4.031 seconds**

5. Discussion

The study compared the performance of Java and Python in handling key data science tasks, revealing their strengths and limitations. Python showed better memory efficiency during data processing, while Java significantly outperformed Python in multithreading tasks. Java emerged as the more efficient language in terms of execution speed, completing intensive computational tasks faster than Python. This research has important implications for practitioners in data science and related fields, as Python's popularity is primarily due to its

simplicity, rich ecosystem of libraries, and community support. However, Java remains a powerful alternative for tasks requiring high-speed processing and efficient multithreading. Future research could focus on larger datasets, hardware specifications, and optimization techniques to understand performance differences. Future research could also compare Python and Java in other critical areas of data science, such as machine learning model training and deployment, database connectivity, or distributed computing. Additionally, exploring optimizations for Python could provide more insight into improving its performance for high-performance tasks.

6. Results

The results from our experiments show that Java at run time consumed more memory compared to python. This indicates that python is slightly more memory-efficient during data processing. Java outperformed python in multithreading tasks. This shows Java's superior handling of CPU bound multithreading operations. In terms of execution speed, Java consistently demonstrated faster performance than python, especially in computationally intensive tasks. Java's advantage in execution speed further strengthens its case for scenarios requiring rapid data processing. Overall, Java appears to be more efficient in multithreading and execution speed, while python demonstrates better memory management.

7. Conclusion

The study compares Java and Python in data science. Python is more memory-efficient, suitable for limited memory resources, but Java outperforms it in multithreading and execution speed, making it ideal for high-performance tasks and concurrent processing. While Python is popular due to its simplicity, Java should be considered for computational efficiency and speed. Future research should explore larger datasets, efficiency in machine learning model training, and optimize Python for multithreading and execution speed to narrow the performance gap.

8. References

1. Sarda, D. (2015). Python for the busy Java developer. Apress.

2. Shaw, A. P. J. (2023). Towards evaluating Python as a suitable data science programming language for modern computing architecture. Macquarie University.

3. Gupta, A. (2021). Comparison in Java and Python: A review paper. RIMT University.

4. Omer, S. K., Kareem, A. M., & Muhammad, H. H. (2021). Comparative analysis: Exploring encapsulation in Java and Python programming languages. University of Kurdistan Hewler.

5. Acevedo, M. M. A. (2021). Java vs. Python in AI. Universitat Politècnica de Catalunya.

6. Abdulkareem, S. A., & Abboud, A. J. (2021). Evaluating Python, C++, JavaScript, and Java programming languages based on software complexity calculator (Halstead metrics).

7. Siebers, A., & Baydag, R. M. (2021). A micro-research on the materialities of programming languages: Using the example of an array in Java versus Python.

8. Cutting V., & Stephen N. (2021). Comparative review of Java and Python. International Journal of Research and Development in Applied Science and Engineering (IJRDASE).

9. Insanudin, E. (2019). Implementation of Python source code comparison results with Java using bubble sort method. Journal of Physics: Conference Series.

10. DZone. (2024). Data statistics and analysis with Java and Python. DZone.

11. Lutz, M. (2013). Learning Python (5th ed.). O'Reilly Media.

12. Bloch, J. (2017). Effective Java (3rd ed.). Addison-Wesley.

13. Guttag, J. V. (2016). Introduction to computation and programming using Python (2nd ed.). MIT Press.

14. Horstmann, C. S., & Cornell, G. (2018). Core Java volume I: Fundamentals (11th ed.). Prentice Hall.

15. Shaw, Z. A. (2017). Learn Python 3 the hard way: A very simple introduction to the terrifyingly beautiful world of computers and code. Addison-Wesley.

16. Ramalho, L. (2015). Fluent Python: Clear, concise, and effective programming. O'Reilly Media.

17. Eckel, B. (2006). Thinking in Java (4th ed.). Prentice Hall.

18. Subramaniam, V. (2011). Programming concurrency on the JVM: Mastering synchronization, STM, and actors. Pragmatic Bookshelf.

19. Gorelick, M., & Ozsvald, I. (2014). High performance Python. O'Reilly Media.

20. Oaks, S. (2014). Java performance: The definitive guide. O'Reilly Media.

21. Journal of Physics: Conference Series (2023).