УДК 004.62

ИССЛЕДОВАНИЕ ИНСТРУМЕНТОВ ДЛЯ АНАЛИЗА ТАБЛИЧНЫХ ДАННЫХ В PYTHON

Коцюбинский К.А.
студент,
Институт сферы обслуживания и предпринимательства (филиал) ДГТУ в г. Шахты,
Шахты, Россия

Кузнецов Д.В.
ассистент,
Институт сферы обслуживания и предпринимательства (филиал) ДГТУ в г. Шахты,
Шахты, Россия

Аннотация

В статье проводится исследование библиотек Python для анализа табличных данных, включая Polars, Pandas и DuckDB, с акцентом на сравнении их производительности и эффективности использования памяти. Рассмотрены ключевые операции, такие как фильтрация, функциональные операции и обработка больших объемов данных, что позволило оценить, какая библиотека быстрее выполняет задачи и расходует меньше ресурсов памяти. Статья выявляет отличия в подходах библиотек к хранению данных и обработке вычислений, а также анализирует, какие методы оказываются наиболее экономичными с точки зрения ресурсов при решении аналитических задач.

Ключевые слова: Python, анализ табличных данных, производительность, Polars, Pandas, DuckDB, эффективность памяти, сравнение библиотек, обработка данных, оптимизация ресурсов.

RESEARCH OF THE TOOLS FOR TABULAR DATA ANALYSIS IN PYTHON

Kotsyubinsky K.A.
student,
Institute of Service and Business (branch) DSTU in Shakhty,
Shakhty, Russia

Kuznetsov D.V.
assistant,
Institute of Service and Business (branch) DSTU in Shakhty,
Shakhty, Russia

Abstract

The article examines Python libraries for analyzing tabular data, including Polars, Pandas and DuckDB, with an emphasis on comparing their performance and memory efficiency. Key operations such as filtering, aggregation and processing of large amounts of data are considered, which made it possible to assess which library performs tasks faster and consumes less memory resources. The article identifies differences in the approaches of libraries to data storage and computing processing, and also analyzes which methods turn out to be the most economical in terms of resources when solving analytical problems.

Key words: Python, tabular data analysis, performance, Polars, Pandas, DuckDB, memory efficiency, library comparison, data processing, resource optimization.

На сегодняшний день анализ данных играет решающую роль в развитии и успехе практически любой отрасли, от бизнеса до научных исследований и государственного управления. Объем данных, с которыми ежедневно работают

организации, постоянно растет, и это предъявляет новые требования к методам и инструментам обработки информации.

График на рисунке 1 демонстрирует рост объема данных в мире. Экспоненциальный характер графика подчеркивает важность применения производительных инструментов для обработки и анализа большого объема данных.

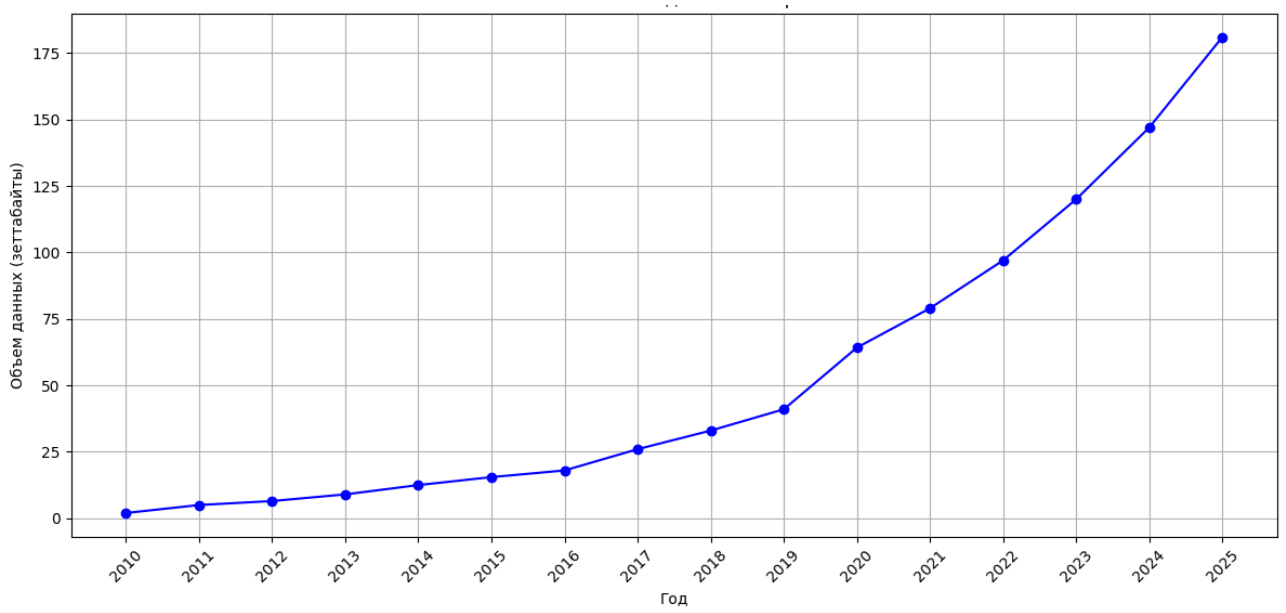

Рис.1 – График роста объема данных в мире [1]

Язык Python один из самых востребованных языков для анализа данных благодаря своей относительной простоте, гибкости, большому количеству подходящих библиотек и удобной экосистеме. Библиотеки, такие как Pandas [2], Polars [3] и DuckDB [4], предоставляют мощные инструменты для работы с данными. Они позволяют эффективно фильтровать, агрегировать и преобразовывать данные, поддерживая работу с различными форматами и увеличивая производительность анализа. Эти библиотеки оптимизированы для обработки больших объемов данных, что особенно важно в условиях постоянного роста информации.

Для сравнения эффективности работы с данными проведем комплексные тесты библиотек Pandas, Polars и DuckDB. В ходе тестирования оценим

различные аспекты работы с большими объемами данных, включая время, затраченное на загрузку и обработку информации, а также скорость выполнения операций, таких как фильтрация, нахождение суммы значений (sum) и среднего значения (mean), максимального (max) и минимального (min) элемента. Учтем и другой важный параметр, такой как потребление оперативной памяти при выполнении этих операций.

Первым экспериментом является измерение производительности загрузки данных с помощью рассматриваемых библиотек. На рисунке 2 представлены графики частотности и рассеяния для анализа скорости загрузки фрейма данных и потребления памяти при этом процессе.

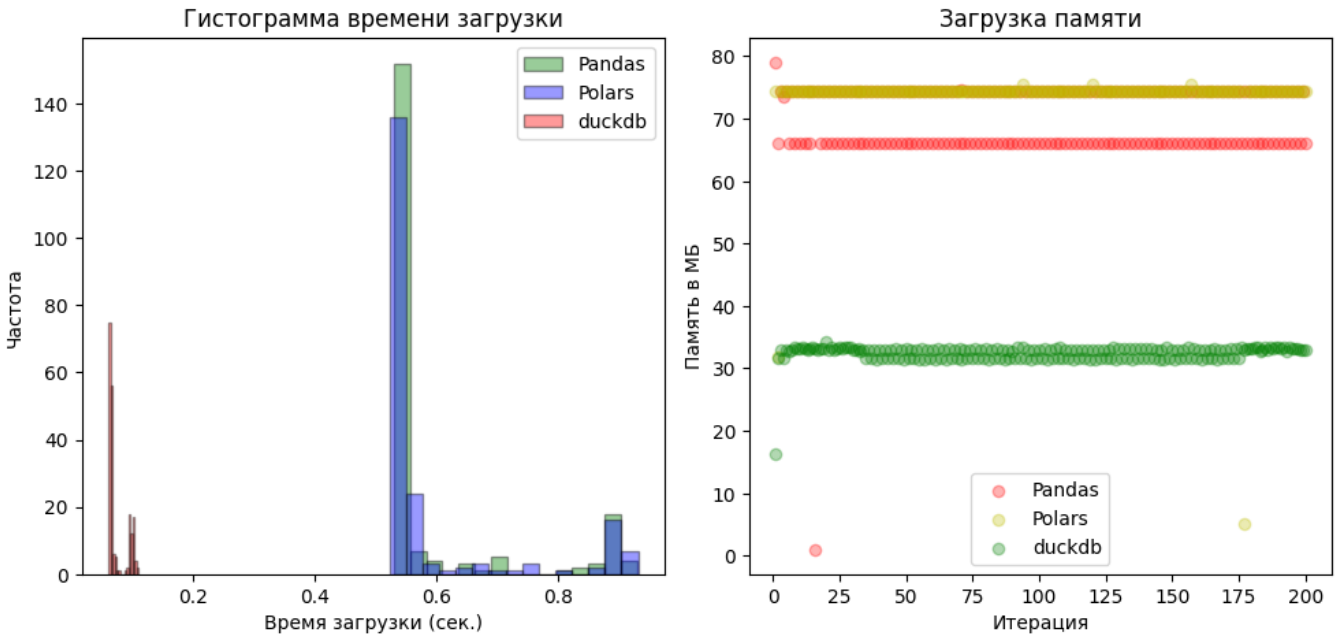

Рис.2 – Графики частотности и рассеяния при работе с фреймом данных [Создано авторами]

Результаты тестирования показали, что библиотека DuckDB занимает лидирующую позицию как по скорости загрузки данных, так и по эффективности использования оперативной памяти. На втором месте по скорости загрузки данных находится Polars, которая обрабатывает данные быстрее, чем Pandas, но при этом требует больше памяти, уступая в этом показателе именно Pandas, которая занимает второе место по экономии

ресурсов памяти. Pandas, в свою очередь, оказалась самым ресурсоемким инструментом, занимая третье место по скорости и второе по потреблению памяти.

Для более детального анализа производительности библиотек DuckDB, Polars и Pandas проведем тестирование их работы при применении различных функций, таких как sum, mean, max, и min, к фрейму данных. Эти функции являются одними из наиболее часто используемых в анализе данных, поскольку позволяют быстро получать сводные статистики по столбцам.

В ходе второго эксперимента будет измерено время выполнения каждой из функций, а также сколько памяти было использовано. На рисунке 3 представлены графики производительности при выполнении функций для каждой из библиотек.

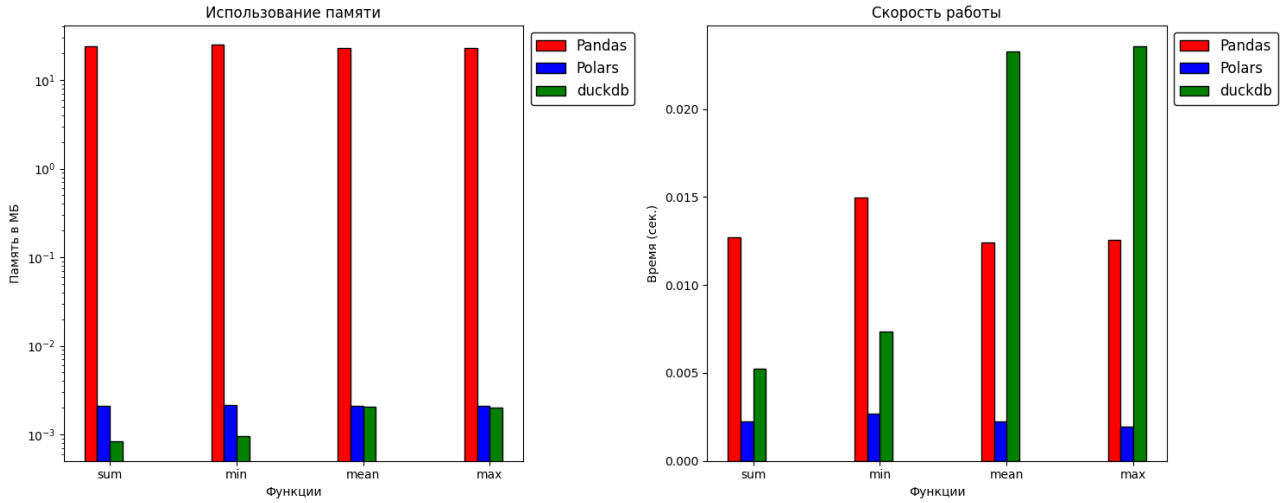

Рис.3 – Графики производительности при выполнении функций для каждой из библиотек [Создано авторами]

Pandas продемонстрировала наибольшее потребление памяти среди всех библиотек. Ее значения существенно превышают результаты Polars и DuckDB. Это свидетельствует о менее эффективной оптимизации Pandas для работы с большими объемами данных. В свою очередь, Polars и DuckDB показали значительно меньшие значения использования памяти, что говорит об их лучшей адаптации для таких задач.

Что касается скорости выполнения, Polars оказалась лидером во всех тестируемых функциях. Ее архитектура, ориентированная на многозадачность, позволяет быстрее справляться с расчетами даже при больших объемах данных. DuckDB заняла второе место по производительности, особенно хорошо справляясь с функциями sum и mean, однако при выполнении операций min и max она уступила Pandas. Интересно, что Pandas, хотя и оказалась медленнее остальных библиотек в большинстве тестов, все же продемонстрировала хорошую скорость выполнения для функций min и max, что может быть связано с особенностями ее оптимизации для таких вычислений.

В третьем и заключительном эксперименте будет измерена производительность библиотек при фильтрации данных с использованием различных критериев. Фильтрация данных является одной из наиболее важных операций в процессе анализа, так как она позволяет эффективно извлекать нужные подмножества данных на основе заданных условий.

Цель тестирования — оценить скорость выполнения фильтрации и использование памяти в каждой библиотеке при применении различных типов условий. Это будет включать в себя проверку фильтрации по отдельным условиям, комбинации нескольких условий, а также работу с диапазонами значений. На рисунке 4 представлен график скорости фильтрации данных для каждой из библиотек.

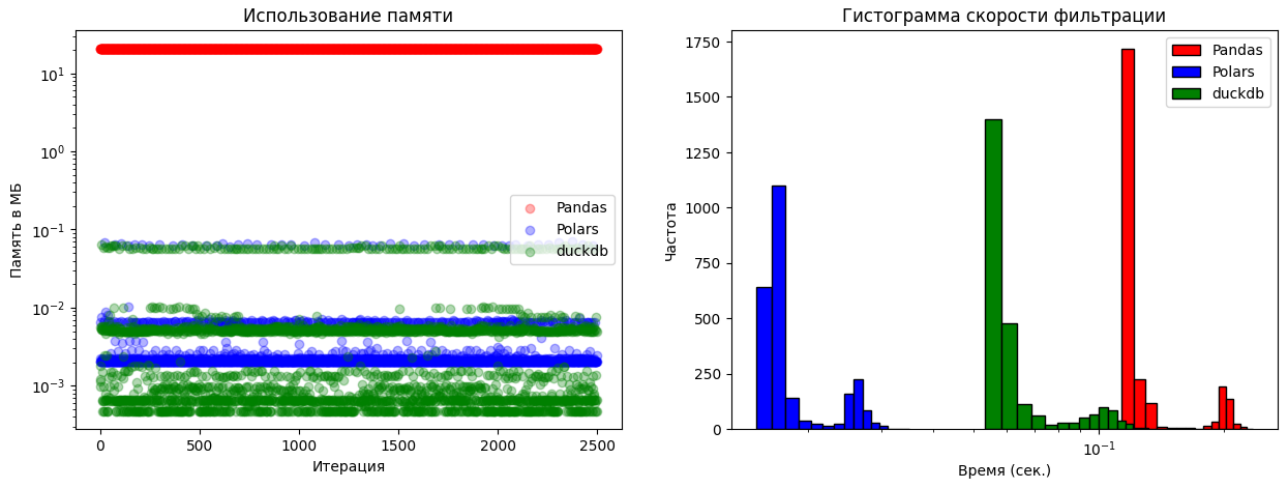

Рис.4 – Графики производительности при выполнении фильтрации данных для каждой из библиотек [Создано авторами]

Результаты анализа показали, что библиотека Pandas значительно обгоняет остальные по объему потребляемой памяти. На графике видно, что ее значения использования памяти стабильно находятся в высоком диапазоне, что может свидетельствовать о низком уровне оптимизации этой библиотеки для работы с большими объемами данных. Polars занимает промежуточную позицию по данному показателю, демонстрируя более эффективное использование памяти, чем Pandas, однако немного уступая DuckDB. Ее результаты расположены ближе к области, характеризующей эффективную оптимизацию. DuckDB, напротив, демонстрирует наименьший объем потребляемой памяти, что подтверждает высокий уровень оптимизации ее архитектуры для экономичного использования ресурсов.

Временные затраты на выполнение операций также выявили значительные различия в производительности библиотек. Pandas продемонстрировала наибольшую продолжительность выполнения операций, что отражает ее низкую эффективность по сравнению с конкурентами. Время выполнения операций Polars оказалось наименьшим, что обусловлено высокой степенью оптимизации данной библиотеки, ориентированной на многопоточность и работу с большими объемами данных. DuckDB показала

средние результаты, уступая Polars, но опережая Pandas. Ее временные показатели находятся в диапазоне, который делает библиотеку эффективным инструментом для выполнения SQL-запросов и обработки больших наборов данных.

Результаты тестирования производительности трех библиотек — Polars, DuckDB и Pandas — в задачах загрузки данных, фильтрации и выполнения агрегатных операций продемонстрировали существенные различия в их эффективности и потреблении ресурсов.

Polars продемонстрировала наилучшую производительность по большинству тестируемых параметров. Её архитектура, ориентированная на многозадачность и эффективное использование системных ресурсов, позволяет достигать высокой скорости выполнения операций, включая фильтрацию данных и расчёт агрегатных функций. Оптимизация для работы с большими объёмами данных минимизирует использование памяти. Это делает Polars оптимальным выбором для задач, где критичны производительность и низкая нагрузка на ресурсы. Однако ограниченная экосистема и меньшая популярность по сравнению с Pandas могут затруднять интеграцию в проекты, требующие разнообразного инструментария.

DuckDB заняла промежуточное положение между Polars и Pandas, демонстрируя хорошую производительность, особенно при выполнении SQL-запросов и фильтрации больших наборов данных. Она стабильно показывает вторые результаты по скорости выполнения операций, оставаясь эффективной при низком уровне потребления памяти, хотя немного уступает Polars. DuckDB подходит для задач, где необходимо сочетание SQL-функционала и высокой производительности, таких как аналитика больших таблиц или работа с реляционными базами данных.

Pandas, несмотря на свою популярность и развитую экосистему, показал наихудшие результаты в тестах производительности. Объём потребляемой

памяти значительно превышает показатели Polars и DuckDB, а время выполнения операций, таких как загрузка данных и фильтрация, оказалось самым высоким. Тем не менее, Pandas остаётся востребованной благодаря интуитивно понятному интерфейсу и широкой поддержке сообщества. Она подходит для задач, где на первый план выходят удобство работы, богатый функционал и доступность инструментов, а не критичная оптимизация для больших данных.

В заключение хотелось бы отметить, что проведенное исследование позволило оценить производительность рассмотренных библиотек при выполнении заданных операций, необходимых при анализе табличных данных в Python. Кроме того, были даны рекомендации по выбору библиотеки под определенные классы задач, основной мерой сложности которых выступает количество данных. Так, библиотеки Polars и DuckDB являются наилучшим выбором для анализа больших массивов данных, потому что показывают сравнимую производительность, а выбор конкретной библиотеки зависит от предпочтений аналитика. Библиотека же Pandas является хорошим выбором при работе со сравнительно небольшими наборами данных, благодаря удобному интерфейсу и удовлетворительной производительности. Для ознакомления с кодом исследования, отражающим процесс тестирования и полученные результаты, вы можете посетить репозиторий на GitHub [5].

Библиографический список:

1. Сколько данных создается каждый день? // Инклиент URL: https://inclient.ru/data-create-stats/ (дата обращения: 30.11.2024).

2. Molin S. Hands-On Data Analysis with Pandas: A Python data science handbook for data collection, wrangling, analysis, and visualization. – 2-е издание. – Бирмингем: Packt, 2021 – 788 с.

3. Nahrstedt F. et al. An Empirical Study on the Energy Usage and Performance of Pandas and Polars Data Analysis Python Libraries // Proceedings of the 28th International Conference on Evaluation and Assessment in Software Engineering. – 2024. – С. 58-68.

4. Needham M., Hunger M., Simons M. DuckDB in Action. – Шелтер Айленд: Manning, 2024 – 312 с.

5. Research-Pandas-Polars-duckdb // GitHub URL: https://github.com/Dywinar/Research-Pandas-Polars-duckdb (дата обращения: 30.11.2024).

Оригинальность 76%