ISSN 2658-5014 (Print)
2023. Т. 5, № 5 (14). С. 16–24
СИИТ
СИСТЕМНАЯ ИНЖЕНЕРИЯ И ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ
ISSN 2686-7044 (Online)
http://siit.ugatu.su

УДК 004.942
DOI 10.54708/2658-5014-SIIT-2023-no5-p16

Анализ производительности автоматизированных систем обработки информации

А. Н. Скитяева • В. Ю. Арьков

Аннотация. В статье рассматривается проблема сравнительного анализа быстродействия автоматизированных систем. В качестве примера рассматривается задача построения регрессии, для которой исходные данные многократно генерируются средствами имитационного моделирования. Обнаружено явление нестабильности результатов в отношении продолжительности вычислений, построена гистограмма распределения.

Ключевые слова: автоматизированные системы; анализ данных; регрессионный анализ; распределение вероятностей; Google Colab; Jupiter Notebook; VS Code; большие данные.

Введение

Большие данные (Big Data) – массивы информации, которые невозможно обработать и анализировать с помощью традиционных методов и инструментов. Они характеризуются высокой скоростью поступления, разнообразием форматов и источников, а также сложностью структурирования и интерпретации [1–3], что создает необходимость в использовании новых технологий и инструментов для их обработки и анализа, а также в обучении и повышении квалификации специалистов в данной области и обеспечении их необходимыми знаниями и навыками для эффективной работы [4, 5].

В области аналитики и статистики это особенно актуально, поскольку большие данные могут предоставить более точные и полные результаты анализа [6]. Для достижения результата необходимо научиться использовать специализированные программные инструменты и технологии, такие как Hadoop, Spark, Python [7] и R.

Задача по обработке больших данных эффективно решается с помощью языка программирования Python, который является одним из наиболее популярных языков программирования [8]. Он предоставляет богатый набор таких библиотек и инструментов для работы с данными, как NumPy, Pandas, SciPy, Matplotlib и т. д. Для начала работы с помощью Python необходимо установить и настроить соответствующее окружение, например, VS Code, Anaconda или Jupyter Notebook [9, 10]. Также можно использовать облачные решения, например, Google Colab. В целом анализ больших данных с помощью данного языка программирования является эффективным инструментом для работы, например, с компьютерными сетями и позволяет выявить проблемы и улучшить их работу [11–13].

Стоит отметить, что задача построения регрессии является одной из ключевых задач анализа больших данных. Регрессионный анализ позволяет определить связь между зависимой переменной и одной или несколькими независимыми переменными, и может быть использован для прогнозирования будущих значений зависимой переменной на основе известных значений независимых. Одним из методов регрессионного анализа, который широко используется в данной области, является линейная регрессия, которая позволяет определить линейную связь между зависимой и независимыми переменными [14].

В данной статье рассматривается задача построения регрессии, для которой можно сгенерировать произвольный объем исходных данных и исследовать характеристики производительности. Эксперимент будет реализован в среде разработки Visual Studio Code в блокноте

Jupiter Notebook. Также будет произведено сравнение производительности в обработке больших данных между Google Colab и VS Code.

Разработка имитационной агент-ориентированной системы функционирования малых предприятий

Ранее для анализа обработки больших данных использовались JupyterLab, Jupyter Notebook и Google Colab [14, 15]. При использовании данных платформ была обнаружена нестабильность в обработке данных, но наиболее стабильно и равномерно они обрабатывали в облачном сервисе Google Colab.

Все вышеперечисленные среды использовали возможности браузера для отображения результатов обработки, то есть происходила нагрузка на браузер при рендеринге. Теперь будет произведена та же самая процедура обработки данных в редакторе Visual Studio Code, который поддерживает файлы Jupyter Notebook, чтобы проверить, насколько эффективно будут обрабатываться данные без дополнительной нагрузки на браузер.

Сравнение Google Colab и VS Code: Google Colab предлагает возможность использовать бесплатные вычислительные ресурсы, включая GPU и TPU, что может значительно ускорить процесс обработки данных [16]. Также Google Colab имеет доступ ко множеству таких библиотек и инструментов для анализа данных, как Pandas, NumPy и Matplotlib. В отличие от Google Colab VS Code не предоставляет вычислительных ресурсов, но позволяет локально установить и использовать множество библиотек для анализа данных. Кроме того, VS Code имеет богатый набор инструментов для отладки кода и автоматического форматирования [17].

Visual Studio Code представляет собой текстовый редактор, разработанный Microsoft для Windows, Linux и macOS. Создать файл с расширением .ipynb (рисунок 1) можно несколькими способами.

[1]
```
import time #импорт библиотеки (ИБ) (получение текущего времени)
import numpy as np #ИБ (работа с числовыми массивами)
import matplotlib.pyplot as plt #ИБ(построение графиков)
from sklearn import linear_model #ИБ(машинное обучение)
#инструмент построения линейной регрессии
reg = linear_model.LinearRegression()
np.random.seed(20130479)
A = [] #пустой список (ПС) оценки свободного члена
B = [] #ПС оценки коэффициента регрессии
T = [] #ПС длительности вычислений
N = 10000 #параметр объема выборки
n = 1200 #параметр количества итераций цикла
```

[2]
```
for i in range(n):
  t = time.time()
  x = np.random.uniform(low = 150, high = 200, size = N)
  e = np.random.normal(loc = 0, scale = 10, size = N)
  y = -100 + x + e
  reg.fit(x.reshape(-1, 1), y.reshape(-1, 1))
  A.append(reg.intercept_[0])
  B.append(reg.coef_[0][0])
  T.append(time.time() - t)
```

[3]
```
plt.figure(figsize = (10, 6)) #размер графика в дюймах
plt.hist(B, bins = 20, edgecolor = 'black', color = 'white') #bins – деление
plt.title ('Гистограмма оценок коэффициента регрессии')
plt.show #вывод графика на экран
```

Рис. 1 Файл с записанным в ячейках текстом кода.

1. Правой кнопкой мыши кликнуть на директиву, в которой необходимо создать файл. Выбрать команду «New File». Написать название файла с расширением, например, Обработка_больших_данных.ipynb.

2. На панели управления кликнуть по вкладке «File». Выбрать команду «New File». В выпадающем окне выбрать тип файла Jupyter Notebook. Далее создастся безымянный файл, который необходимо переименовать и сохранить (Ctrl + S).

При запуске программы можно обнаружить ошибки, связанные с отсутствием необходимых для запуска библиотек. Чтобы написанная программа смогла отработать, необходимо установить библиотеки: numpy, matplotlib, -U scikit-learn scipy matplotlib. Для этого в консоли необходимо написать команды:

pip install numpy;

pip install matplotlib;

pip install -U scikit-learn scipy matplotlib.

В ячейки записан ранее использовавшийся код. Рассмотрим время, затрачиваемое на обработку каждой ячейки, и результат обработки в виде гистограмм и графиков.

Ячейка 1: 12,3 секунд (рисунок 2).

```
import time #импорт библиотеки (ИБ) (получение текущего времени)
import numpy as np #ИБ (работа с числовыми массивами)
import matplotlib.pyplot as plt #ИБ(построение графиков)
from sklearn import linear_model #ИБ(машинное обучение)
#инструмент построения линейной регрессии
reg = linear_model.LinearRegression()
np.random.seed(20130479)
A = [] #пустой список (ПС) оценки свободного члена
B = [] #ПС оценки коэффициента регрессии
T = [] #ПС длительности вычислений
N = 10000 #параметр объема выборки
n = 1200 #параметр количества итераций цикла
```

✓ 12.3s

Рис. 2 Время обработки первой ячейки.

Ячейка 2: 0,8 секунд (рисунок 3).

```
for i in range(n):
    t = time.time()
    x = np.random.uniform(low = 150, high = 200, size = N)
    e = np.random.normal(loc = 0, scale = 10, size = N)
    y = -100 + x + e
    reg.fit(x.reshape(-1, 1), y.reshape(-1, 1))
    A.append(reg.intercept_[0])
    B.append(reg.coef_[0][0])
    T.append(time.time() - t)
```

✓ 0.8s

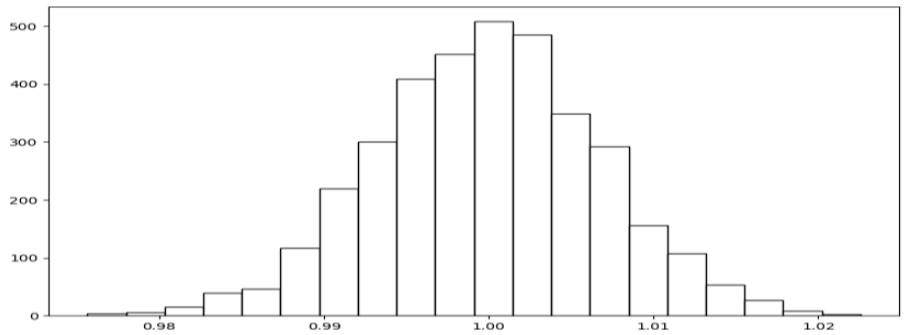
Рис. 3 Время обработки второй ячейки.

Ячейка 3: 0,4 секунды (рисунок 4).

```
plt.figure(figsize = (10, 6)) #размер графика в дюймах
plt.hist(B, bins = 20, edgecolor = 'black', color = 'white') #bins
plt.title ('Гистограмма оценок коэффициента регрессии')
plt.show #вывод графика на экран
```

✓ 0.4s

а



б

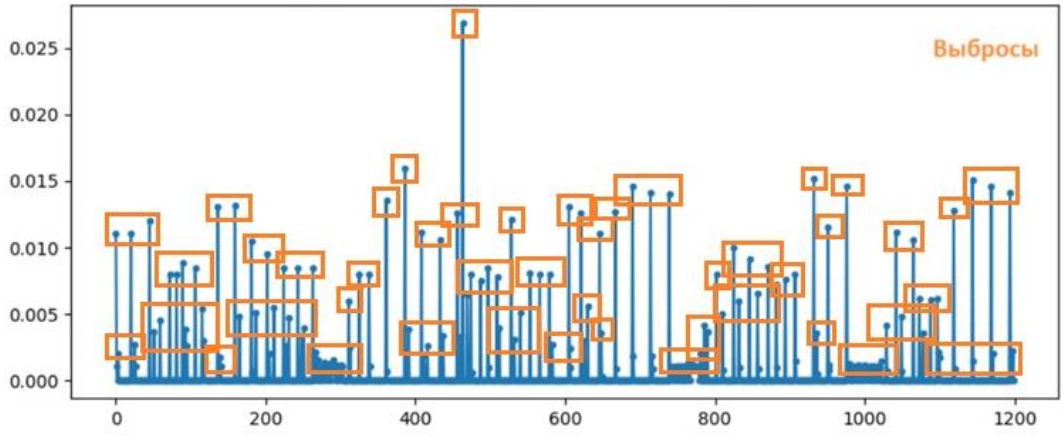
Рис. 4 Результат выполнения третьей ячейки:
а – время обработки третьей ячейки; *б* – гистограмма оценок коэффициента регрессии.

Ячейка 4: 0,1 секунды (рисунок 5).

```
print (min(T), np.median(T), max(T))
plt.figure(figsize = (10, 4))
plt.plot(T, marker = '.')
plt.title ('График изменения продолжительности расчетов')
plt.show()
```

✓ 0.1s

а



б

Рис. 5 Результат выполнения четвертой ячейки:
а – время обработки четвертой ячейки; *б* – график изменения продолжительности расчетов.

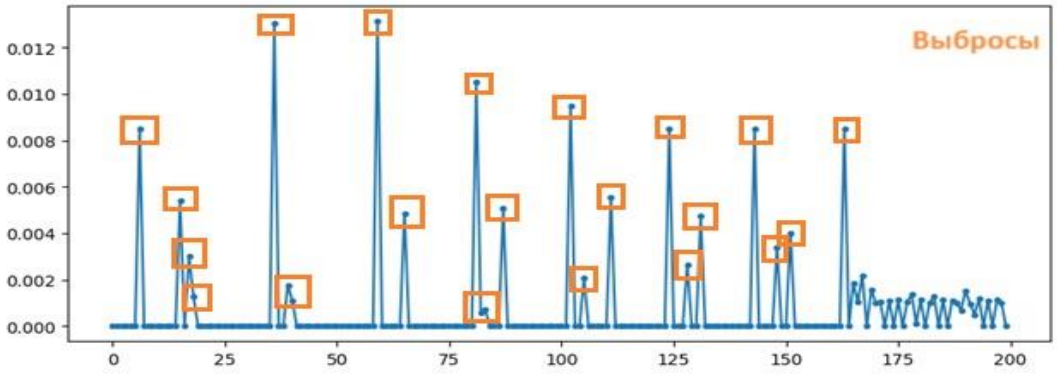
На графике достаточно много выбросов («пиков»), то есть увеличивается продолжительность вычислений.

Необходимо выбрать отрезок с более-менее однородными результатами. Пусть это будет отрезок [100; 300].

Ячейка 5: 0,2 секунды (рисунок 6).

```
plt.figure(figsize = (10, 4))
plt.plot(T[100:300], marker = '.')
plt.show()
```
✓ 0.2s

а



б

Рис. 6 Результат выполнения пятой ячейки:
а – время обработки пятой ячейки; *б* – «однородные» длительности вычислений.

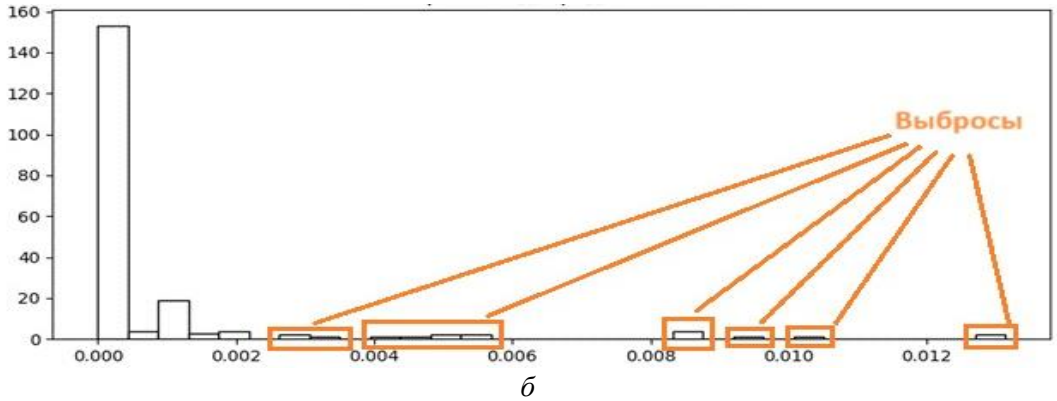
На полученном графике также наблюдается большое число выбросов, что связано с неоднородностью выбранного участка.

Далее необходимо построить гистограмму, отражающую длительность расчётов.

Ячейка 6: 0,1 секунды (рисунок 7).

```
plt.figure(figsize = (10, 4))
plt.hist(T[100:300], bins = 30, edgecolor = 'black', color = 'white')
plt.title ('Гистограмма однородного массива')
plt.show()
```
✓ 0.1s

а

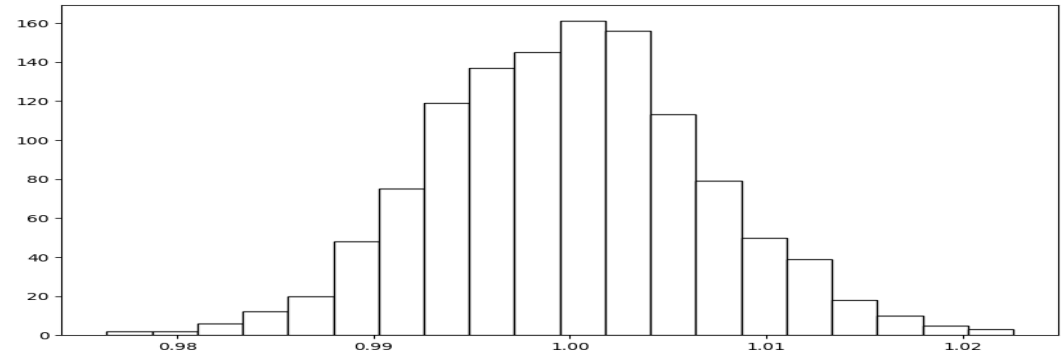


б

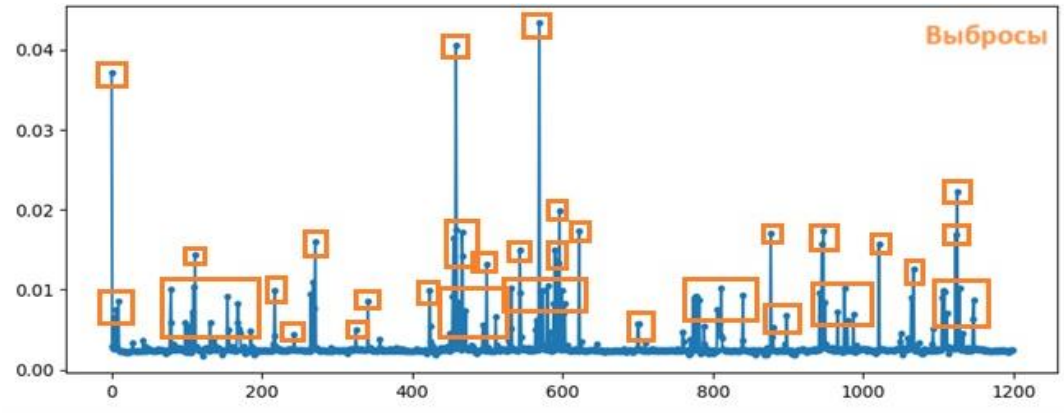
Рис. 7 Результат выполнения шестой ячейки:
а – время обработки шестой ячейки; *б* – гистограмма однородного массива.

Общее время обработки всех ячеек составило 13.9 секунды. Несмотря на работу в среде редактирования кода без вывода результатов в окне браузера, была обнаружена неоднородная обработка данных. Таким образом, можно сделать вывод, что нагрузка на браузер при рендеринге не имеет существенного влияния на равномерность обработки больших данных.

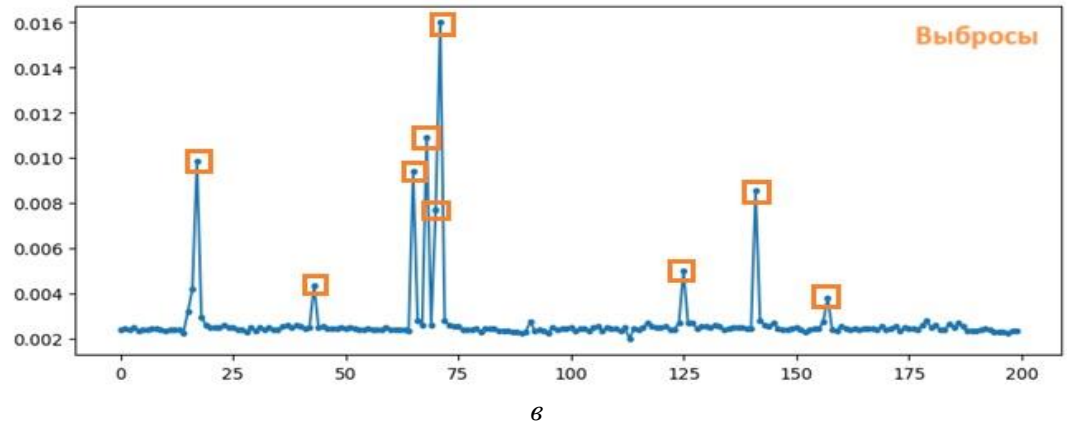
Повторим данный эксперимент силами Google Colab (рисунок 8).



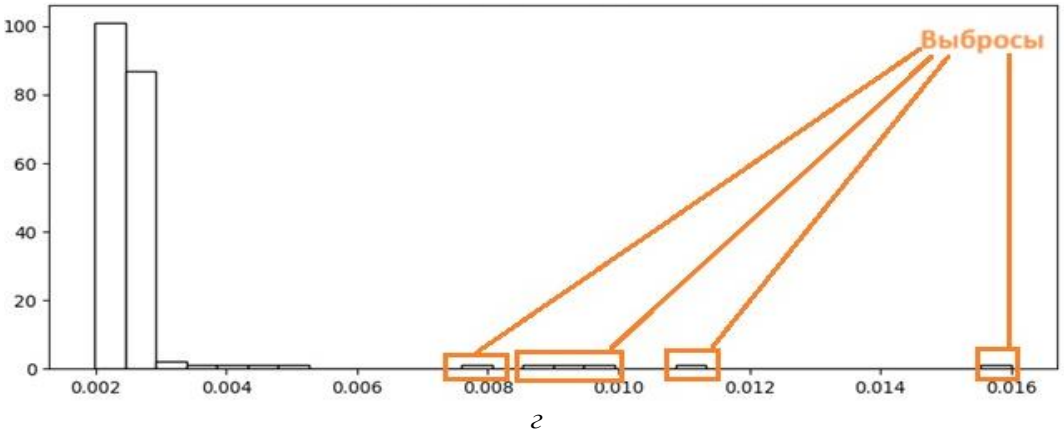
а



б



в



г

Рис. 8 Результат выполнения кода в Google Colab:
а – гистограмма оценок коэффициента регрессии; *б* – график изменения продолжительности расчетов; *в* – «однородные» длительности вычислений; *г* – гистограмма однородного массива.

Получили следующие результаты:
– ячейка 1: 1,025 сек.;
– ячейка 2: 4,055 сек.;
– ячейка 3: 1,011 сек.;
– ячейка 4: 0,716 сек.;
– ячейка 5: 0,818 сек.;
– ячейка 6: 0,615 сек.

Общее время обработки ячеек составило 8.24 сек.

На графике изменения продолжительности расчетов (см. рисунок 8, *б*) также наблюдается большое количество выбросов, но их количество меньше, чем на ранее полученном графике в VS Code (см. рисунок 5, *б*).

Для построения графика «однородных» длительностей вычислений (см. рисунок 8, *в*) было необходимо выбрать отрезок с более-менее однородными результатами. Пусть это будет отрезок [200; 400]. На полученном графике наблюдается число выбросов, которое составляет примерно 30% от среднего. То есть данное число выбросов гораздо меньше (примерно в 2.4 раза), чем в предыдущем результате (см. рисунок 6, *б*).

Также было необходимо построить гистограмму, отражающую длительность расчётов для «однородного» массива (см. рисунок 8, *г*). На полученной гистограмме однородного массива также наблюдаются выбросы, но их количество меньше примерно в 1.5 раза, чем в предыдущем результате (см. рисунок 7, *б*).

Результаты эксперимента

В результате проведение эксперимента в редакторе кода VS Code и в облачном решении Google Colab были получены результаты, отличающиеся между собой в скорости обработки данных и в количестве выбросов (разница в количестве выбросов была рассмотрена после рисунка 8).

Сравним время, затраченное на прогон программы в VS Code и Google Colab (таблица).

Таблица

Время обработки ячеек

Номер ячейки	VS Code	Google Colab
1	12.3 сек	1.025 сек
2	0.8 сек	4.055 сек
3	0.4 сек	1.011 сек
4	0.1 сек	0.716 сек
5	0.2 сек	0.818 сек
6	0.1 сек	0.615 сек
Итого	13.9 сек	8.24 сек

В результате сравнительного анализа многократного прогона одной и той же программы на платформах VS Code и Google Colab была установлена нестабильность производительности. Причем отсутствие нагрузки на браузер при обработке кода и отрисовке результата в редакторе VS Code не повлияло на получение более равномерной обработки данных. Также в VS Code времени на обработку кода ушло примерно в 1,69 раза больше, чем в облачном решении.

Значит, решение повысить производительность за счёт работы в редакторе кода, то есть без применения облачных сервисов, не является эффективным. Необходимо учитывать, что редакторы требуют установки библиотек при их импорте, что не требуется в облачных решениях.

Заключение

Таким образом, можно сделать вывод о том, что нельзя однозначно оценить производительность обработки больших данных с помощью Google Colab и VS Code через Jupiter Notebook, так как это зависит от многих факторов, таких как размер данных, сложность алгоритмов, используемые библиотеки и вычислительные ресурсы. Но стоит отметить, что Google Colab предоставляет доступ к вычислительным ресурсам без необходимости их импорта, что ускоряет процесс обработки данных, в то время как VS Code с Jupiter Notebook позволяет локально установить и использовать множество библиотек для анализа данных. Оба инструмента справляются с задачей обработки больших данных, выбор зависит от целей и предпочтений пользователя.

СПИСОК ЛИТЕРАТУРЫ / REFERENCES

1. Билуха И. Н. Обработка больших данных // Молодой ученый. 2020. № 8 (298). С. 7-9. [[Belukha I. N. “Processing of big data” // Young Scientist, 2020, No. 8 (298), pp. 7-9. (In Russian).]]

2. Джангаров А. И., Сулейманова М. А. Анализ больших данных // Colloquium-journal. 2019. [[Dzhangirov A. I., Suleymanova M. A. “Big data analysis” // Colloquium-journal, 2019. (In Russian).]]

3. Демидов Д. В., Перминев М.А., Пивоваров Г. А. Анализ существующих систем обработки и визуализации больших данных для решения задач бизнес-аналитики // Формообразование в дизайне, рекламе, информационных технологиях: Матлы Всероссийской научно-практической конференции студентов, аспирантов и преподавателей, 2018. С. 34–40. [[Demidov D.V., Perminov M.A., Pivovarov G.A. “Analysis of existing big data processing and visualization systems for solving business intelligence problems” // Shaping in Design, Advertising, Information Technology: Materials of the All-Russian Scientific and Practical Conference of students, postgraduates and teachers, 2018, pp. 34-40. (In Russian).]]

4. Самошкин П. А., Иванов И. А., Фоминов М. А. Анализ информации компьютерной сети на основе больших данных // Славянский форум. 2022. № 4 (38). С. 323–332. [[Samoshkin P. A., Ivanov I. A., Fomina M. A. “Analysis of computer network information based on big data” // Slavic Forum, 2022, No. 4 (38), pp. 323-332. (In Russian).]]

5. Абдирахимов И. Э. Проблемы и решения в Big Data // Sanoatda Raqamli Texnologiyalar / Цифровые технологии в промышленности. 2023. [[Abdirakhimov I. E. “Problems and solutions in Big Data” // Sanoatda Raqamli Texnologiyalar / Digital Technologies in Industry-news, 2023. (In Russian).]]

6. Скитяева А. Н. Анализ производительности автоматизированных систем обработки информации // Мавлютовские чтения: Мат-лы XVI Всероссийской молодежной научной конференции. В 6 т. Уфа, 25–27 октября 2022 года. Т. 5. Уфа: Уфимский государственный авиационный технический университет, 2022. С. 109–114. EDN FFGKVI. [[Skityaeva A. N. “Performance analysis of automated information processing systems” // Materials of the XVI All-Russian Youth Scientific Conference. Vol. 5. Ufa, 2022. 109-114. (In Russian).]]

7. Бухаров Т. А., Нафикова А. Р., Мигранова Е. А. Обзор языка программирования Python и его библиотек // Colloquium-journal. 2019. [[Bukharov T. A., Nafikova A. R., Migranova E. A. “Review of the python programming language and its libraries” // Colloquium-journal, 2019. (In Russian).]]

8. Копытова М. А. Актуальность языка программирования Python // Экономика и социум. 2016. [[Kopytova M. A. “Relevance of the Python programming language” // Economy and Society, 2016. (In Russian).]]

9. Костюченко Ю. А. Анализ подходов к моделированию данных с помощью библиотек языка Python // Альманах научных работ молодых ученых университета ИТМО: Мат-лы XLVII научной и учебно-методической конференции Университета ИТМО по тематикам: экономика; менеджмент, инноватика, 2018. С. 175–178. [[Kostyuchenko Yu. A. “Analysis of approaches to data modeling using python libraries” // Almanac of Scientific Works of Young Scientists of ITMO University. XLVII Scientific and educational-methodical conference of ITMO University on topics: economics; management, Innovation, 2018, pp. 175-178. (In Russian).]]

10. Косьминов Т. Р., Тихонов А. И. Интерактивные веб-приложения Jupiter Notebook для учебного процесса // Электромеханика, электротехнологии, электротехнические материалы и компоненты. Труды МКЭЭЭ-2016, 2016. С. 270–271. [[Kosminov T. R., Tikhonov A. I. “Interactive Jupiter Notebook web applications for the educational process” // Electromechanics, Electrotechnologies, Electrotechnical Materials and components. Proceedings of the ICEE-2016, 2016, pp. 270-271. (In Russian).]]

11. Мельникова В. А., Медведев Д. А. Анализ больших данных с использованием Python // Труды Братского государственного университета. Серия: Естественные и инженерные науки. 2019. Т. 1. С. 46–49. [[Melnikova V. A., Medvedev D. A. “Big data analysis using Python” // Proceedings of Bratsk State University. Series: Natural and Engineering Sciences, 2019, Vol. 1, pp. 46-49. (In Russian).]]

12. Ермаков О. А., Брозгунова Н. П. Python как инструмент для анализа данных // Наука и образование. 2020. Т. 3. № 4. С. 26. [[Ermakov O. A., Brozgunova N. P. “Python as a tool for data analysis” // Science and Education, 2020, Vol. 3, No. 4, p. 26. (In Russian).]]

13. Ельсуков Д. А. Python – язык программирования // Экономика и социум. 2021. [[Zhukov D. A. “Python – programming language” // Economy and Society, 2021. (In Russian).]]

14. Арьков В. Ю., Шарипова А. М., Куликов Г. Г. Оценивание неопределённости в машинном обучении // Вестник Южно-Уральского государственного университета. Серия: Компьютерные технологии, управление, радиоэлектроника. 2023. Т. 23. № 3. С. 48–58. [[Arkov V. Yu., Sharipova A. M., Kulikov G. G. “Estimation of uncertainty in machine learning” // Bulletin of the South Ural State University. Series: Computer Technologies, Control, Radio electronics, 2023, vol. 23, No. 3, pp. 48-58. (In Russian).]]

15. Гусаренко А. С. Производительность запросов к гетерогенным источникам в ситуационно-ориентированных базах данных // Системная инженерия и информационные технологии. 2023. Т. 5. № 3 (12). С. 42–52. [[Gusarenko A. S. “Performance of queries to heterogeneous sources in situation-oriented databases” // System Engineering and Information Technologies, 2023. Vol. 5, No. 3(12), pp. 42-52. (In Russian).]]

16. Волокитина Т. С. Анализ возможностей Google Colab // Современные научные исследования и инновации. 2020. № 12 (116). С. 1. [[Volokitina T. S. “Analysis of the capabilities of Google Colab” // Modern Scientific Research and Innovation, 2020, No. 12 (116), p. 1. (In Russian).]]

17. Иванов И. А., Корнилов Ю. В. Перспективы использования пакета по "neuron - vs code" как замена дистрибутива anaconda для использования в data science // Образование как социокультурный потенциал развития общества: Сб. мат-лов Всероссийской научно-практической конференции с международным участием. 2019. С. 135–138. [[Ivanov I. A., Kornilov Yu. V. “Prospects of using the software package "neuron - vs code" as a replacement for the anaconda distribution kit for use in data science” // Education as a Socio-Cultural Potential for the Development of Society. Collection of materials of the All-Russian Scientific and Practical Conference with international participation, 2019, pp. 135-138. (In Russian).]]

Поступила в редакцию 27 сентября 2023 г.

МЕТАДАННЫЕ / METADATA

Title: Performance analysis of automated information processing systems in the Visual Studio Code development environment.

Abstract: The article deals with the problem of comparative analysis of the performance of automated systems. As an example, the problem of constructing a regression is considered, for which the initial data is repeatedly generated by means of simulation modeling. The phenomenon of instability of the results with respect to the duration of calculations was discovered, a histogram of the distribution was constructed.

Key words: automated systems; data analysis; regression analysis; probability distribution; Google Colab; Jupiter Notebook; VS Code; big data.

Язык статьи / Language: русский / Russian.

Об авторах / About the authors:

СКИТЯЕВА Анастасия Николаевна
ФГБОУ ВО «Уфимский университет науки и технологий», Россия.
Студ. бакалавриата института информатики, математики и робототехники.
E-mail: skityaeva.anastasi@mail.ru
ORCID: https://orcid.org/0009-0005-5105-8506
URL: elibrary.ru/author_profile.asp?authorid=691759

АРЬКОВ Валентин Юльевич
ФГБОУ ВО «Уфимский университет науки и технологий», Россия.
Профессор каф. автоматизированных систем управления. Дипл. инженер электронной техники (Уфимск. авиац. ин-т, 1979). Д-р техн. наук по сист. анализу, управлению и обработке информации (Уфимск. гос. авиац. техн. ун-т, 2002). Иссл. в обл. идентификации и моделирования систем автоматического управления газотурбинными двигателями, управления в организац. системах.
E-mail: arkov.vyu@ugatu.su
ORCID: https://orcid.org/0000-0002-7913-4778
URL: https://elibrary.ru/author_profile.asp?authorid=263152

SKITYAEVA Anastasiya Nikolaevna
Ufa University of Science and Technologies, Russia.
Bachelor student, Institute of Informatics, Mathematics, and Robotics.
E-mail: skityaeva.anastasi@mail.ru
ORCID: https://orcid.org/0009-0005-5105-8506
URL: elibrary.ru/author_profile.asp?authorid=691759

ARKOV Valentin Yulyevich
Ufa University of Science and Technologies, Russia.
Professor of the Department of Automated Control Systems. Dipl. electronics engineer (Ufa Aviation Institute, 1979). Dr. Tech. Sciences in system analysis, management, and information processing (Ufa State Aviation Technical University, 2002). Research in the field of identification and modeling of automatic control systems for gas turbine engines, control in organizational systems.
E-mail: arkov.vyu@ugatu.su
ORCID: https://orcid.org/0000-0002-7913-4778
URL: https://elibrary.ru/author_profile.asp?authorid=263152