

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное
образовательное учреждение высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»**

**Институт перспективной инженерии
Департамент цифровых, робототехнических систем и электроники**

**Отчет по лабораторной работе № 1
«Работа с Jupyter Notebook, JupyterLab, Google Colab»
по дисциплине «Искусственный интеллект и машинное обучение»**

Выполнил:

Гончаров Серафим Ростиславович 2
курс, группа ИВТ-б-о-23-1, 09.03.01
«Информатика и вычислительная
техника», направленность
(профиль) «Автоматизированные
системы обработки информации и
управления», очная форма
обучения

Руководитель практики:
Воронкин Роман Александрович

Ставрополь 2025

Тема: Рабоа с JupyterNotebook, JupyterLabGoogleColab».

Цель работы: исследовать базовые возможности интерактивных оболочек JupyterNotebook, JupyterLab и Google Colab для языка программирования Python.

Порядок выполнения лабораторной работы:

- 1) Открыли официальный сайт Anaconda:

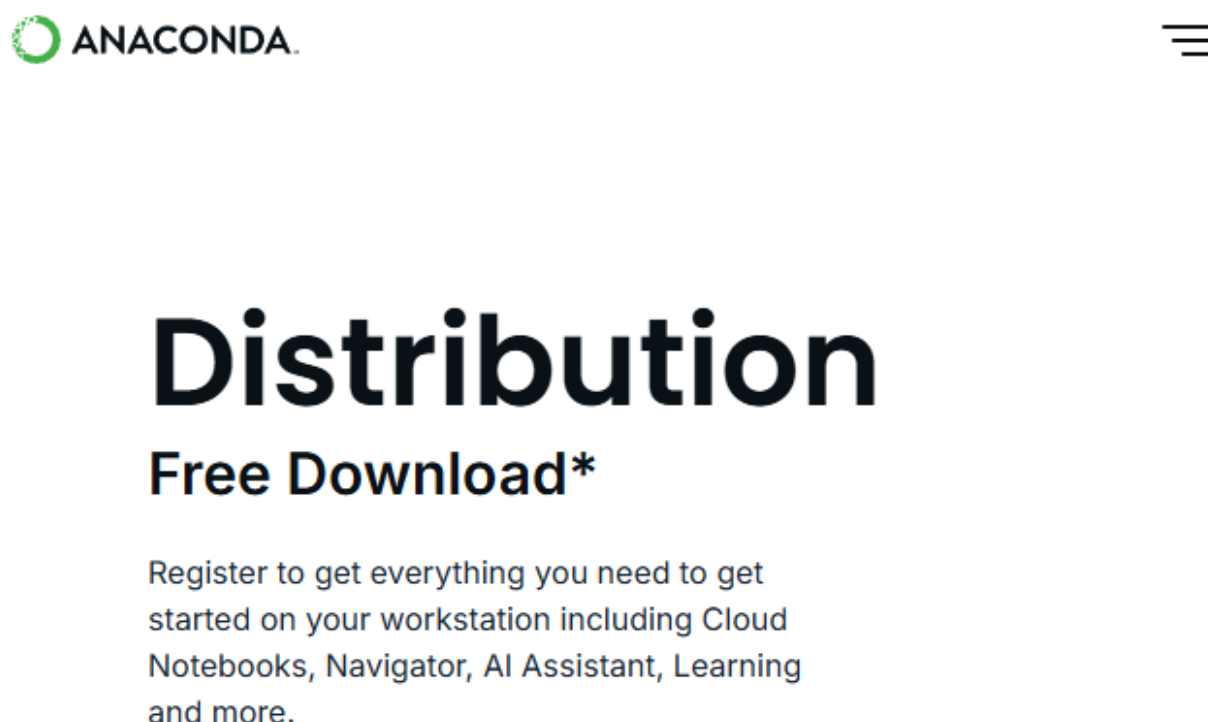


Рисунок 1 – Официальный сайт Anaconda

- 2) Создать общедоступный репозиторий на GitHub:

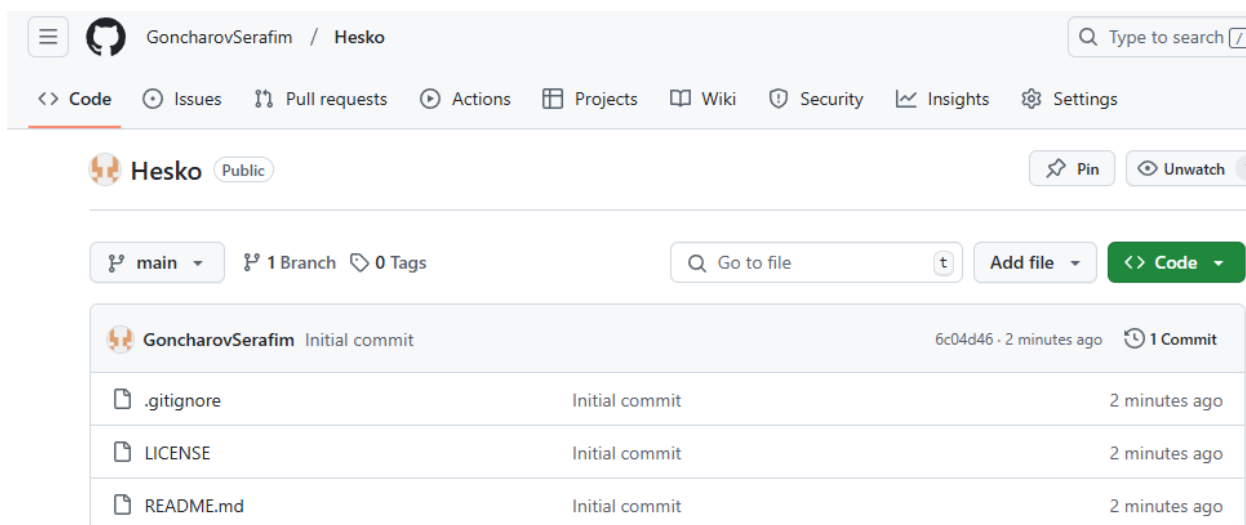
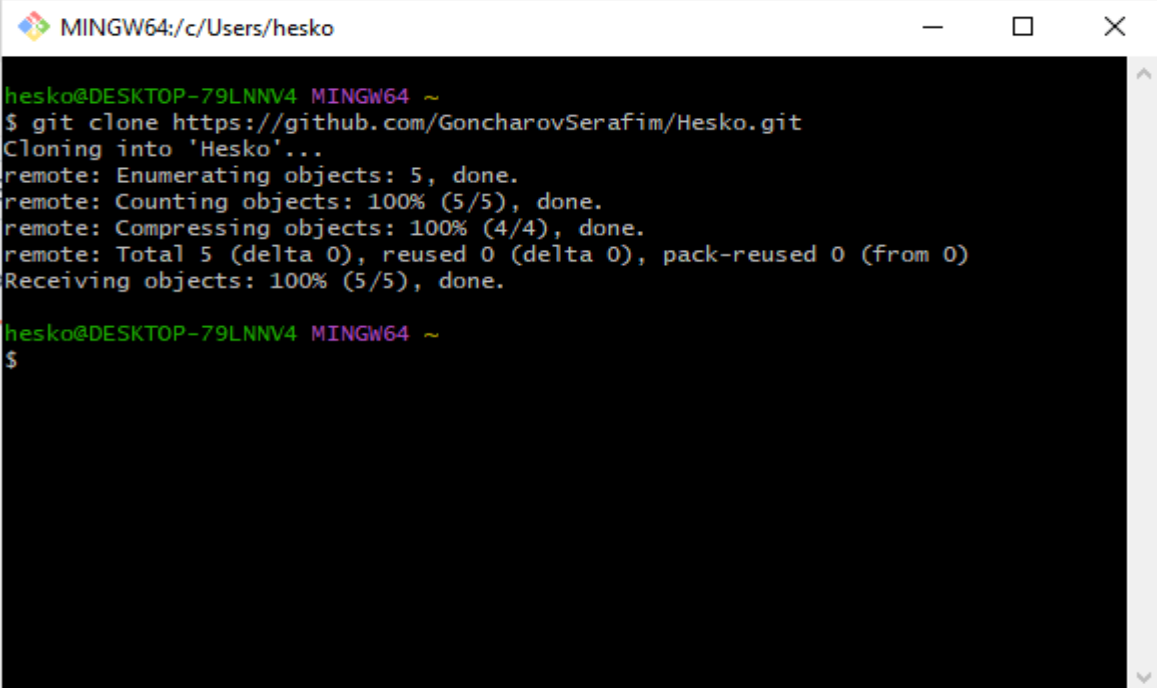


Рисунок 2 – Общедоступный репозиторий

3) Выполните клонирование созданного репозиторий на рабочий компьютер:



```
MINGW64/c/Users/hesko
hesko@DESKTOP-79LNNV4 MINGW64 ~
$ git clone https://github.com/GoncharovSerafim/Hesko.git
Cloning into 'Hesko'...
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 5 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
Receiving objects: 100% (5/5), done.
hesko@DESKTOP-79LNNV4 MINGW64 ~
$
```

Рисунок 3 – Клонирование репозитория

Проработать примеры практической работы:

4) Открывается веб–интерфейс:

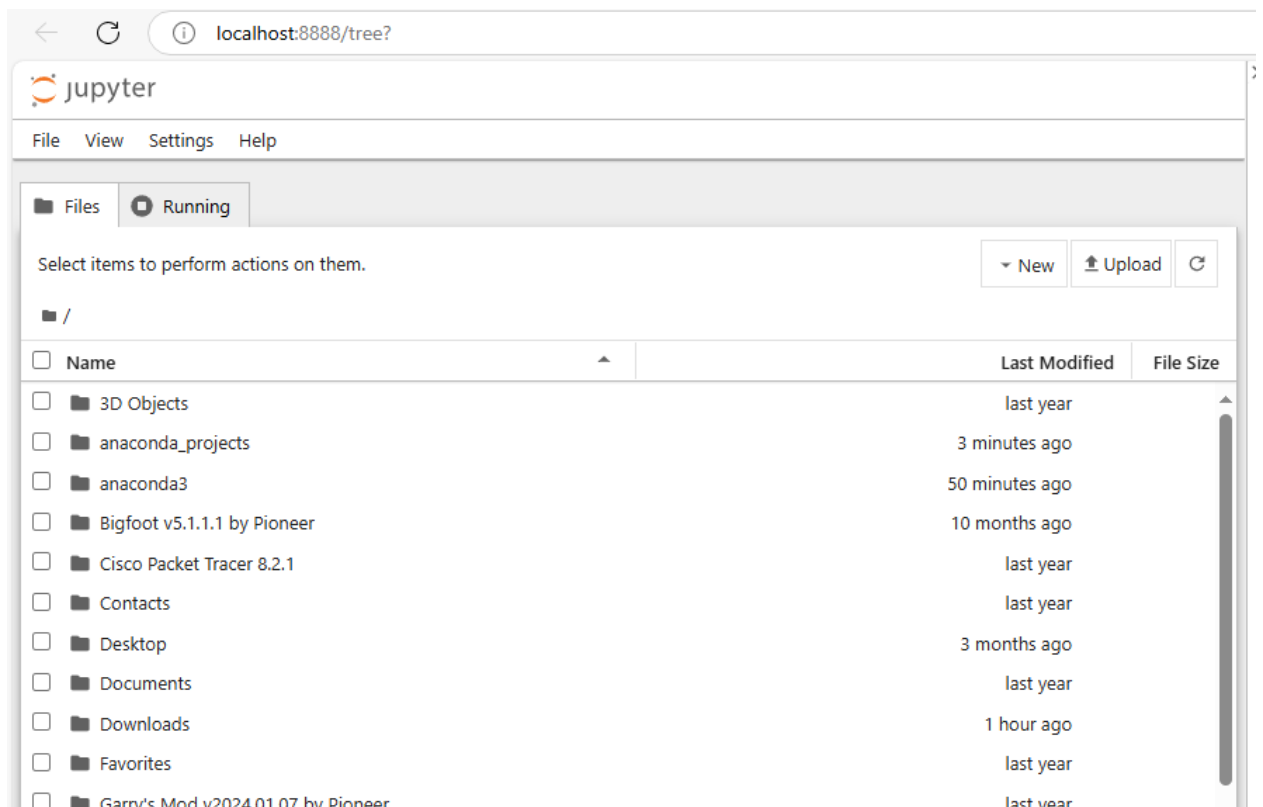


Рисунок 4 – Веб–интерфейс

5) Создаем папку notebooks:

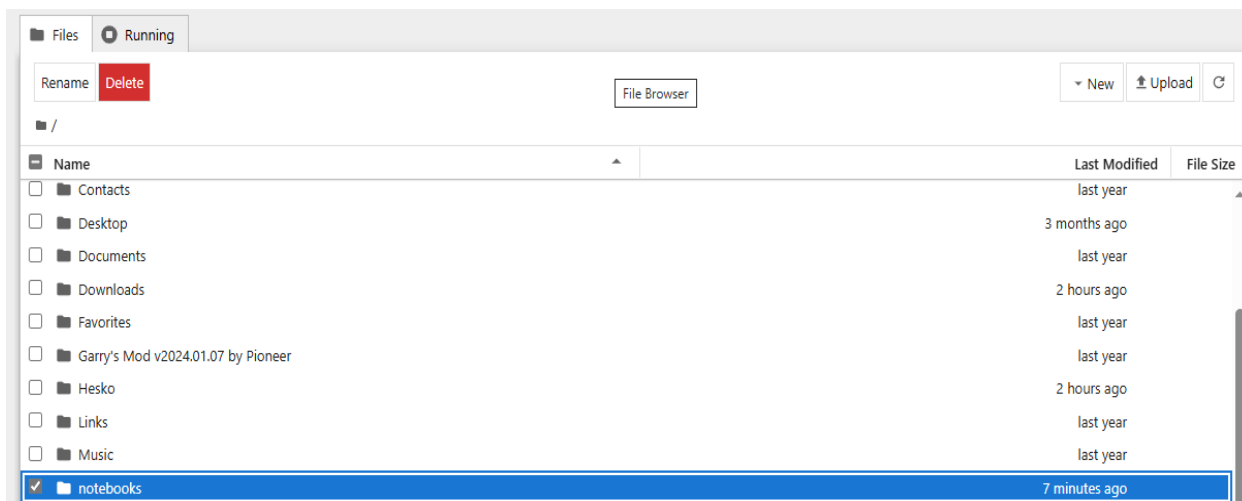


Рисунок 5 – Папка

6) Запускаем ноутбук:

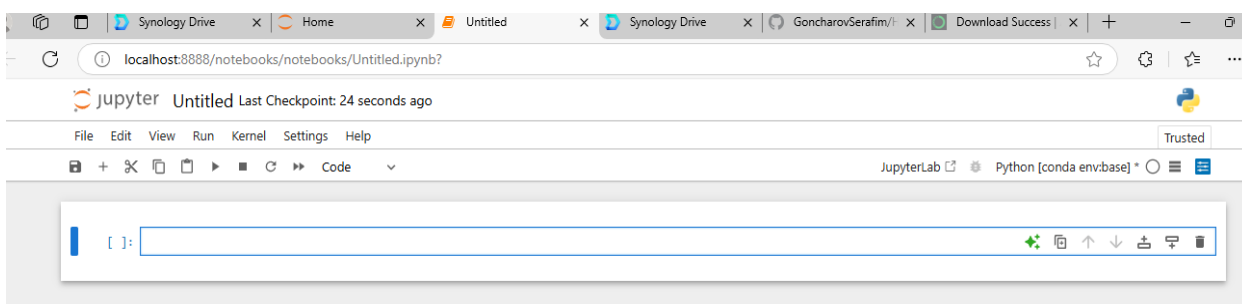


Рисунок6 – Заходим в ноутбук

7) Выполняем пример в интерактивном сеансе:

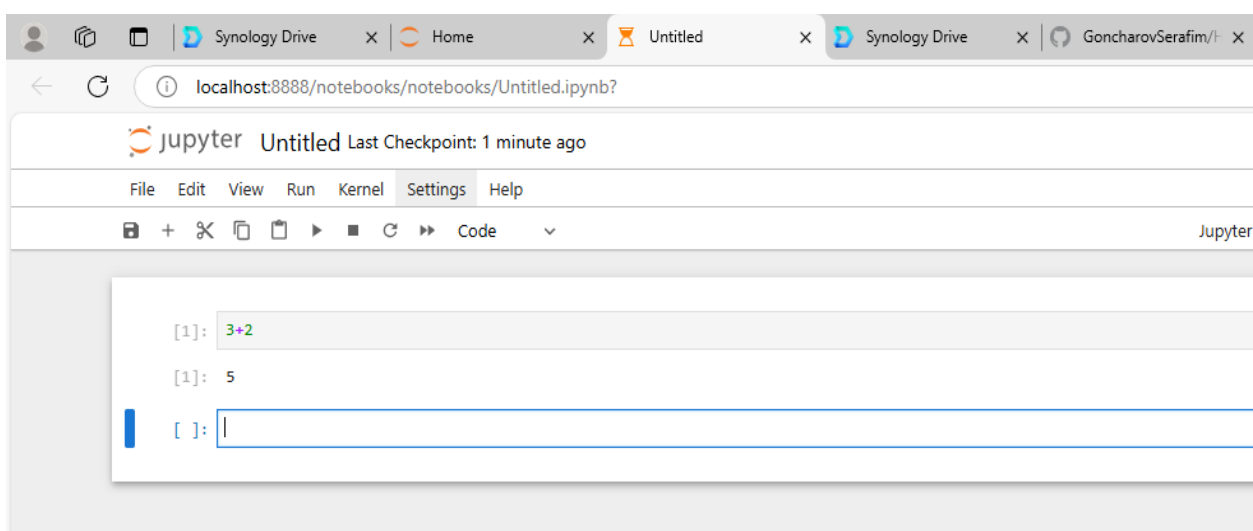
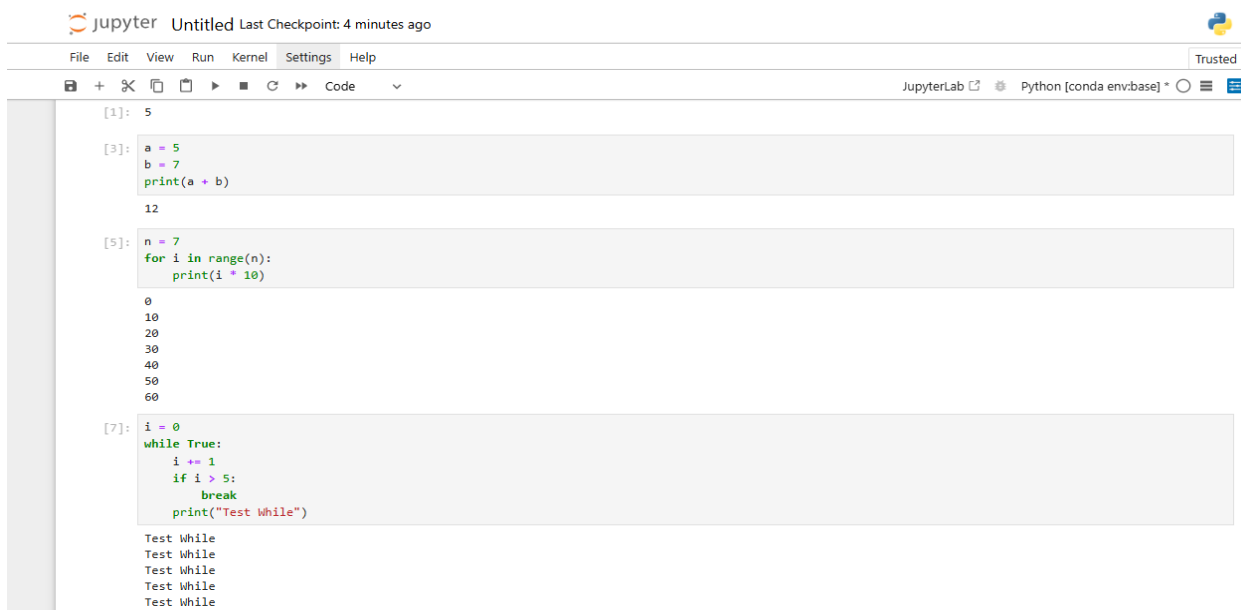


Рисунок7–Выполняем сложение

8) Выполняем пример:



The screenshot shows a JupyterLab window titled 'Untitled' with a last checkpoint 4 minutes ago. The interface includes a menu bar (File, Edit, View, Run, Kernel, Settings, Help) and a toolbar. Three code cells are visible:

```
[1]: 5
```

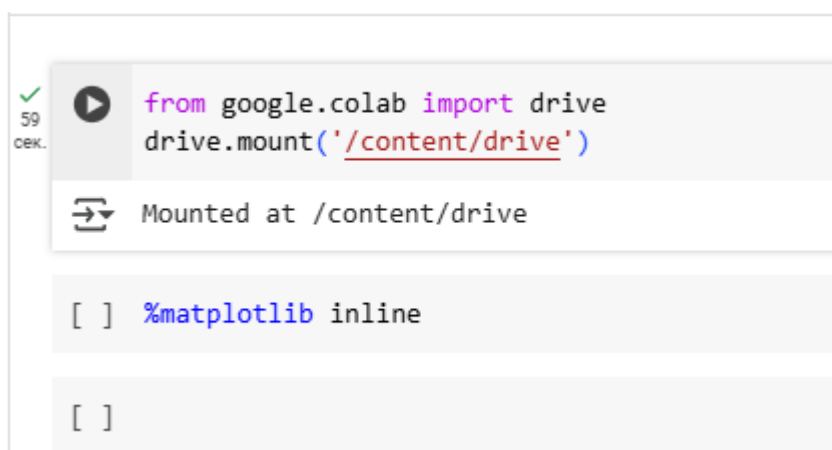
```
[3]: a = 5
      b = 7
      print(a + b)
      12
```

```
[5]: n = 7
      for i in range(n):
          print(i * 10)
      0
      10
      20
      30
      40
      50
      60
```

```
[7]: i = 0
      while True:
          i += 1
          if i > 5:
              break
          print("Test While")
      Test While
      Test While
      Test While
      Test While
      Test While
```

Рисунок 8 – пример

9) Заходим в Google Colab и подключаем Google Диск



The screenshot shows a Google Colab notebook with the following code cells:

```
59 сек. [ ] from google.colab import drive
      drive.mount('/content/drive')
```

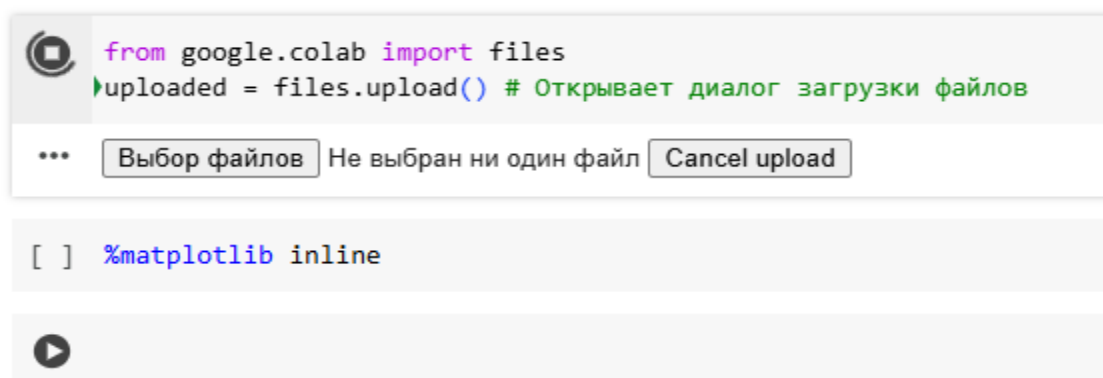
Mounted at /content/drive

```
[ ] %matplotlib inline
```

```
[ ]
```

Рисунок 9 – Google Colab

10) Произвести загрузку файлов:



The screenshot shows a Google Colab notebook with the following code cells:

```
[ ] from google.colab import files
    uploaded = files.upload() # Открывает диалог загрузки файлов
```

... [Выбор файлов] Не выбран ни один файл [Cancel upload]

```
[ ] %matplotlib inline
```

[Run button]

Рисунок 9 – Загрузка файлов

11) Проработать примеры с файловой системой:

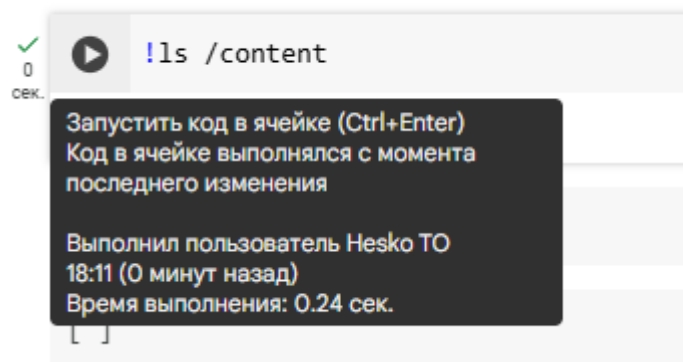


Рисунок 11 - пример

Смотрим все доступные магические команды:

12) Работаем с линейными командами:

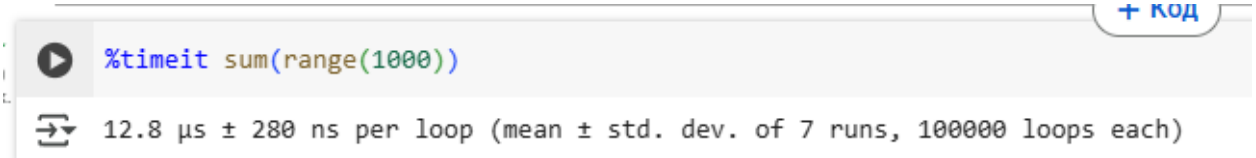


Рисунок 10 – линейные команды

13) Работаем с блочными командами:

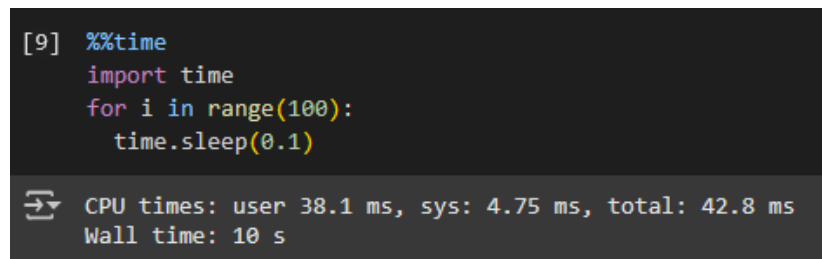


Рисунок 11 – блочные команды

14) Работаем с визуализацией данных:

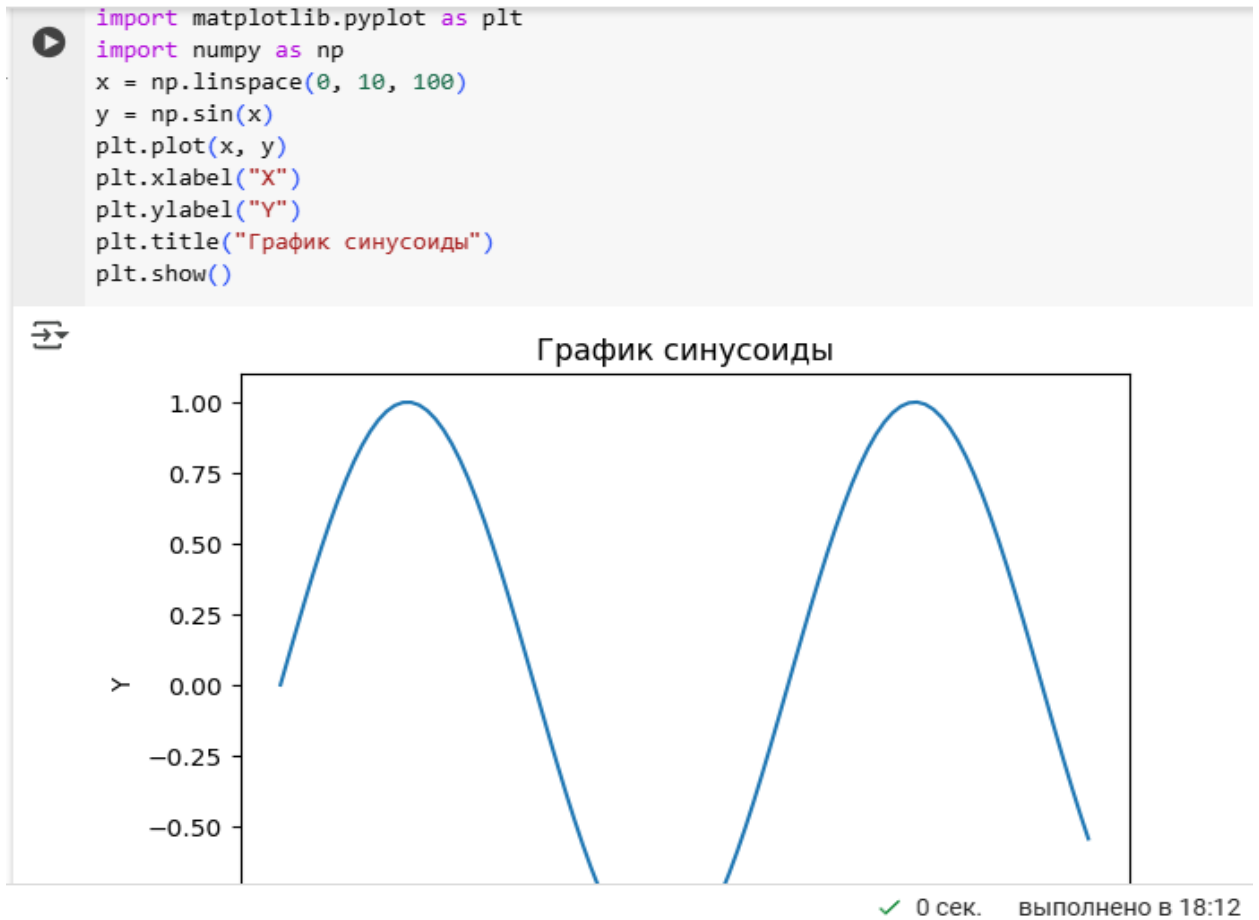


Рисунок12– переменные окружения

15) Устанавливаем стандартные библиотеки:

```
!pip install numpy pandas matplotlib
```

Requirement already satisfied: numpy in /usr/local/lib/python3.11/dist-packages (1.26.4)
Requirement already satisfied: pandas in /usr/local/lib/python3.11/dist-packages (2.2.2)
Requirement already satisfied: matplotlib in /usr/local/lib/python3.11/dist-packages (3.10.0)
Requirement already satisfied: python-dateutil>=2.8.2 in /usr/local/lib/python3.11/dist-packages (from pandas) (2.9.0.post0)
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.11/dist-packages (from pandas) (2025.1)
Requirement already satisfied: tzdata>=2022.7 in /usr/local/lib/python3.11/dist-packages (from pandas) (2025.1)
Requirement already satisfied: contourpy>=1.0.1 in /usr/local/lib/python3.11/dist-packages (from matplotlib) (1.3.1)
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.11/dist-packages (from matplotlib) (0.12.1)
Requirement already satisfied: fonttools>=4.22.0 in /usr/local/lib/python3.11/dist-packages (from matplotlib) (4.56.0)
Requirement already satisfied: kiwisolver>=1.3.1 in /usr/local/lib/python3.11/dist-packages (from matplotlib) (1.4.8)
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.11/dist-packages (from matplotlib) (24.2)
Requirement already satisfied: pillow>=8 in /usr/local/lib/python3.11/dist-packages (from matplotlib) (11.1.0)
Requirement already satisfied: pyparsing>=2.3.1 in /usr/local/lib/python3.11/dist-packages (from matplotlib) (3.2.1)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.11/dist-packages (from python-dateutil>=2.8.2->pandas) (1.17.0)

Рисунок13– стандартные библиотеки

16) Устанавливаем библиотеки из GitHub:

```
!pip install git+https://github.com/GoncharovSerafim/repository.git
```

Collecting git+https://github.com/GoncharovSerafim/repository.git
Cloning https://github.com/GoncharovSerafim/repository.git to /tmp/pip-req-build-z77fu18o
Running command git clone --filter=blob:none --quiet https://github.com/GoncharovSerafim/repository.git /tmp/pip-req-build-z77fu18o
fatal: could not read Username for 'https://github.com': No such device or address

Рисунок14–Время работы

17) Запустили jupyterLab

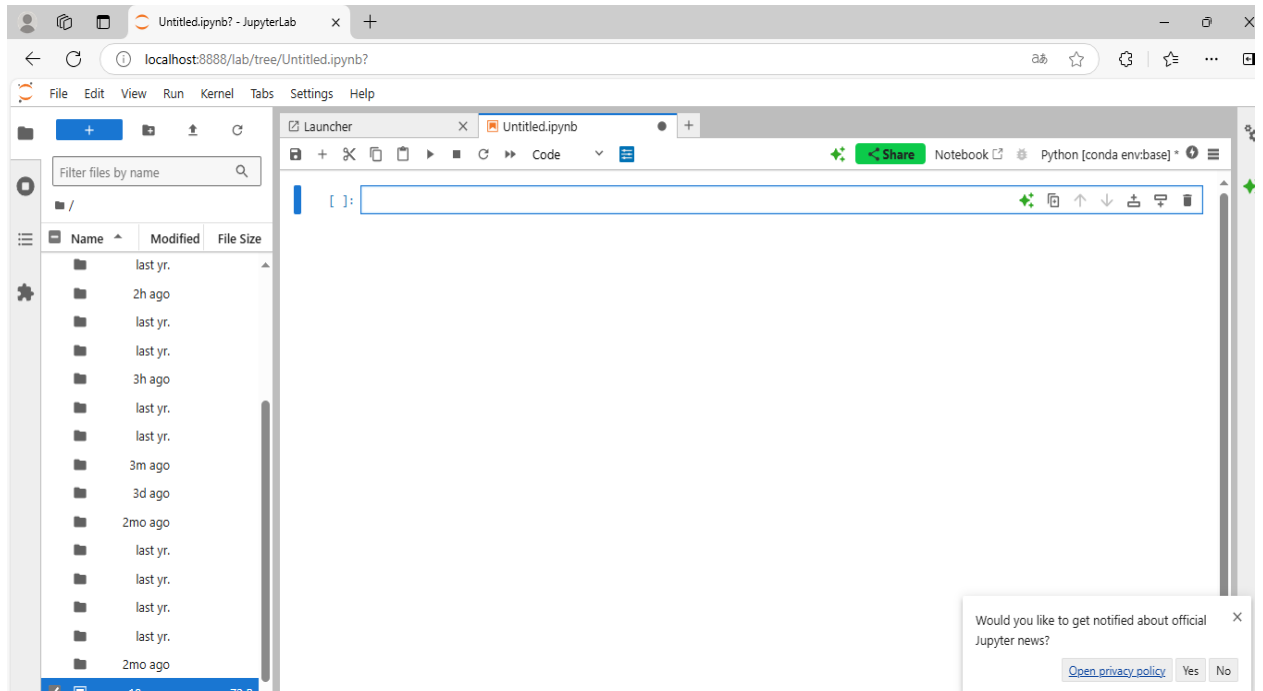


Рисунок17 – JupyterLab

18) Выполнили примеры:

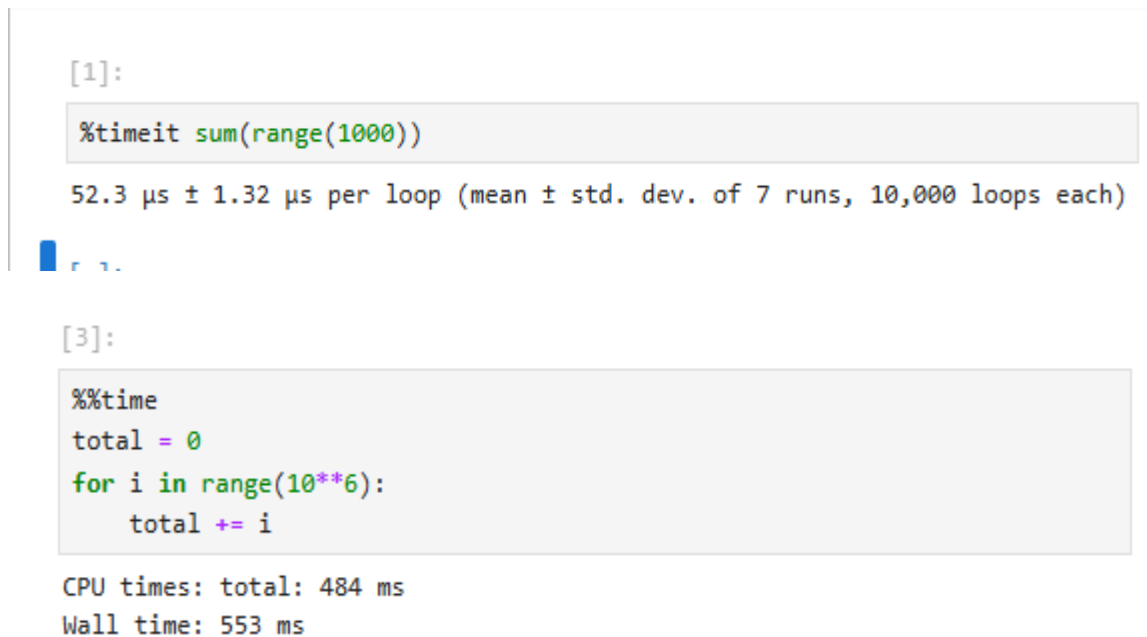


Рисунок18– Примеры

19) Установили библиотеки:

[9]:

```
pip install numpy pandas matplotlib
```

```
Requirement already satisfied: numpy in c:\users\hesko\anaconda3\lib\site-  
packages (1.26.4)  
Requirement already satisfied: pandas in c:\users\hesko\anaconda3\lib\site-  
packages (2.2.2)  
Requirement already satisfied: matplotlib in c:\users\hesko\anaconda3\lib  
\site-packages (3.9.2)  
Requirement already satisfied: python-dateutil>=2.8.2 in c:\users\hesko\an  
aconda3\lib\site-packages (from pandas) (2.9.0.post0)  
Requirement already satisfied: pytz>=2020.1 in c:\users\hesko\anaconda3\li  
b\site-packages (from pandas) (2024.1)  
Requirement already satisfied: tzdata>=2022.7 in c:\users\hesko\anaconda3  
\lib\site-packages (from pandas) (2023.3)  
Requirement already satisfied: contourpy>=1.0.1 in c:\users\hesko\anaconda  
3\lib\site-packages (from matplotlib) (1.2.0)  
Requirement already satisfied: cycler>=0.10 in c:\users\hesko\anaconda3\li  
b\site-packages (from matplotlib) (0.11.0)  
Requirement already satisfied: fonttools>=4.22.0 in c:\users\hesko\anacond  
a3\lib\site-packages (from matplotlib) (4.51.0)  
Requirement already satisfied: kiwisolver>=1.3.1 in c:\users\hesko\anacond  
a3\lib\site-packages (from matplotlib) (1.4.4)  
Requirement already satisfied: packaging>=20.0 in c:\users\hesko\anaconda3  
\lib\site-packages (from matplotlib) (24.1)
```

Рисунок19 – Установили

20) Переключили аппаратные средства:

Сменить среду выполнения

Тип среды выполнения

Python 3

Аппаратный ускоритель ?

☒ ЦП ☐ Графический процессор T4

☐ Графический процессор A100

☐ Графический процессор L4 ☐ TPU v2-8

☐ TPU v5e-1

Нужен доступ к мощным графическим процессорам?
[Купите дополнительные вычислительные единицы](#)

Отмена **Сохранить**

Рисунок20– Аппаратные средства

21) Проверяем наличие процессоров:

```
[11] import torch
      print(torch.cuda.is_available())

False

[12] import tensorflow as tf
      print("TPU доступен:", "Yes" if 'COLAB_TPU_ADDR' in os.environ else "No")

-----
NameError                                Traceback (most recent call last)
<ipython-input-12-89d01adab515> in <cell line: 0>()
      1 import tensorflow as tf
----> 2 print("TPU доступен:", "Yes" if 'COLAB_TPU_ADDR' in os.environ else "No")

NameError: name 'os' is not defined

[13] import os
      import tensorflow as tf
      print("TPU доступен:", "Yes" if 'COLAB_TPU_ADDR' in os.environ else "No")

TPU доступен: No
```

Рисунок21– Проверяем наличие

22) Создаем текстовый файл, записываем строки, читаем и удаляем его:

```
import os

with open("example.txt", "w") as f:
    f.write("Доброе утро!\n")
    f.write("Как настроение?\n")

with open("example.txt", "r") as f:
    content = f.read()
    print("Содержимое файла:\n", content)

print("Файл существует:", os.path.exists("example.txt"))

os.remove("example.txt")
print("Файл удален.")

Содержимое файла:
Доброе утро!
Как настроение?

Файл существует: True
Файл удален.
```

Рисунок22– Задание на редактирование файла

23) Выполняем список заданий:

```
✓ 0 %time sum(range(100))
0 CPU times: user 5 µs, sys: 1 µs, total: 6 µs
сек. Wall time: 10.3 µs
4950

✓ [10] %%time
0 total = sum(range(100**2))
сек. CPU times: user 111 µs, sys: 32 µs, total: 143 µs
Wall time: 145 µs

✓ [15] %%writefile test_script.py
0 for i in range(5):
сек.     print(f"Итерация {i}")

!python test_script.py

Overwriting test_script.py

✓ [16] %ls
0 sample_data/ test_script.py
сек.

✓ %history
0
```

Рисунок23– Задание 3

24) Выполняем список заданий:

Практическое задание №1

Привет, *Серафим*! Добро пожаловать в Google Colab!

1. Google Colab
2. JupyterLab
 - Jupyter Notebook
 - Новый пункт

$$\lim_{x \rightarrow \infty} \frac{1}{x} \sum_{i=1}^n f(x_i) = \int_a^b f(x) dx$$

Рисунок 24– Задание и результат

25) Даем доступ к нашим файлам:

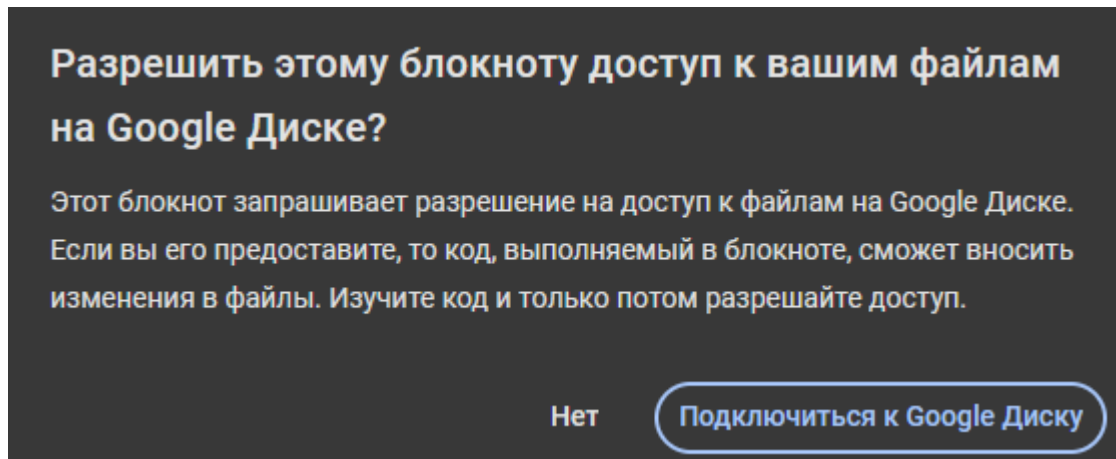


Рисунок25– Доступ к файлам

26) Подключаем GoogleDrive, смотрим список папок в корневой директории и выполняем задания:



Рисунок26– Выполнение заданий

<https://github.com/GoncharovSerafim/Hesko>

Ответы на контрольные вопросы:

1. Какие основные отличия JupyterLab от JupyterNotebook?

JupyterLab - это более современная и расширяемая среда, которая предоставляет интерфейс с вкладками, поддержку работы с несколькими файлами и инструментами в одном окне.

JupyterNotebook - это классический интерфейс, который фокусируется только на работе с ноутбуками и не поддерживает многозадачность в одном окне.

2. Как создать новую рабочую среду (ноутбук) в JupyterLab?

1. Откройте JupyterLab.
2. Нажмите на кнопку "+" в левой панели или выберите File→New→Notebook.
3. Выберите ядро для нового ноутбука.

3. Какие типы ячеек поддерживаются в JupyterLab и как их переключать?

Code - для написания и выполнения кода.

Markdown - для написания текста с форматированием.

Raw - для необработанного текста.

Переключение типов ячеек:

Используйте горячие клавиши: `Esc` для выхода из режима редактирования, затем `Y` для Code, `M` для Markdown, `R` для Raw.

4. Как выполнить код в ячейке и какие горячие клавиши для этого используются?

Нажмите Shift + Enter, чтобы выполнить код в текущей ячейке и перейти к следующей.

Нажмите Ctrl + Enter, чтобы выполнить код в текущей ячейке без перехода к следующей.

5. Как запустить терминал или текстовый редактор внутри JupyterLab?

Терминал: Выберите File → New → Terminal.

Текстовый редактор: Выберите File → New → Text File.

6. Какие инструменты JupyterLab позволяют работать с файлами и структурами каталогов?

Файловый браузер в левой панели позволяет просматривать, создавать, удалять и перемещать файлы и папки.

Терминал позволяет использовать команды командной строки для работы с файлами.

7. Как можно управлять ядрами (kernels) в JupyterLab?

Чтобы перезапустить или остановить ядро, выберите Kernel → RestartKernel или Kernel → ShutdownKernel.

Чтобы сменить ядро, выберите Kernel → Change Kernel.

8. Каковы основные возможности системы вкладок и окон в интерфейсе JupyterLab?

JupyterLab позволяет открывать несколько файлов и инструментов в одном окне с вкладками.

Вы можете перемещать вкладки между окнами, создавая гибкую рабочую среду.

9. Какие магические команды можно использовать в JupyterLab для измерения времени выполнения кода? Приведите примеры.

`%time` - измеряет время выполнения одной строки кода.

`%timesum(range(1000))`

`%%time` - измеряет время выполнения всей ячейки.

`%%time`

`for i in range(1000):`

`print(i*10)`

`%%timeit` - измеряет время выполнения заданного количества повторений кода.

10. Какие магические команды позволяют запускать код на других языках программирования в JupyterLab?

Такие магические команды как `%%bash` для Bash, `%%html` для HTML, `%%javascript` для JavaScript и т.д.

11. Какие основные отличия Google Colab от JupyterLab?

Google Colab - это облачная среда, которая работает в браузере и предоставляет бесплатный доступ к GPU/TPU.

JupyterLab - это локальная среда, которая требует установки и настройки на вашем компьютере.

12. Как создать новый ноутбук в Google Colab?

1. Перейдите на [Google Colab](https://colab.research.google.com/).
2. Нажмите File → New Notebook.

13. Какие типы ячеек доступны в Google Colab, и как их переключать?

Типы ячеек: Code и Text (аналог Markdown).

Переключение: Нажмите на кнопку с иконкой текста (для Text) или кода (для Code) в верхней части ячейки.

14. Как выполнить код в ячейке Google Colab и какие горячие клавиши для этого используются?

Нажмите Shift + Enter для выполнения кода и перехода к следующей ячейке.

Нажмите Ctrl + Enter для выполнения кода без перехода.

15. Какие способы загрузки и сохранения файлов поддерживает Google Colab?

Загрузка: Используйте File → Upload notebook или команды в коде (например, `files.upload()`).

Сохранение: Используйте File → Save или File → Save a copy in Drive.

16. Как можно подключить Google Drive к Google Colab и работать с файлами?

1. Выполните код:

```
from google.colab import drive
drive.mount('/content/drive')
```

2. Авторизуйтесь и получите доступ к файлам в Google Drive.

17. Какие команды используются для загрузки файлов в Google Colab из локального компьютера?

Используйте:

```
from google.colab import files  
uploaded = files.upload()
```

18. Как посмотреть список файлов, хранящихся в среде Google Colab?

Используйте команду: `!ls`

19. Какие магические команды можно использовать в Google Colab для измерения времени выполнения кода? Приведите примеры.

`%time` и `%%time` (аналогично JupyterLab):

```
%time sum(range(1000))
```

20. Как можно изменить аппаратные ресурсы в Google Colab (например, переключиться на GPU)?

1. Перейдите в Runtime → Change runtime type.

2. В разделе Hardware accelerator выберите GPU или TPU.

Вывод: в ходе лабораторной работы мы исследовали базовые возможности интерактивных оболочек Jupyter Notebook, JupyterLab и Google Colab для языка программирования Python.