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
## дисциплины
## «Искусственный интеллект и машинное обучение»

Выполнил:
Митряшкина Дарина Сергеевна
2 курс, группа ИТС-б-о-23-1,
11.03.02 «Инфокоммуникационные технологии и системы связи»,
направленность (профиль)
«Инфокоммуникационные системы и сети», очная форма обучения

_____
(подпись)

Проверил:
Доцент департамента цифровых,
робототехнических систем и
электроники Воронкин Р.А.

_____
(подпись)

Отчет защищен с оценкой _____ Дата защиты_____

Ставрополь, 2025 г.

**Тема:** Рабоа с Jupyter Notebook, JupyterLab,  Google Colab

**Цель:** исследовать базовые возможности интерактивных оболочек Jupyter Notebook, JupyterLab и Google Colab для языка программирования Python

**Порядок выполнения работы:**

1. Изучен теоретический материал по лабораторной работе
2. Установлена программа Anaconda
3. Запущен Google Colab

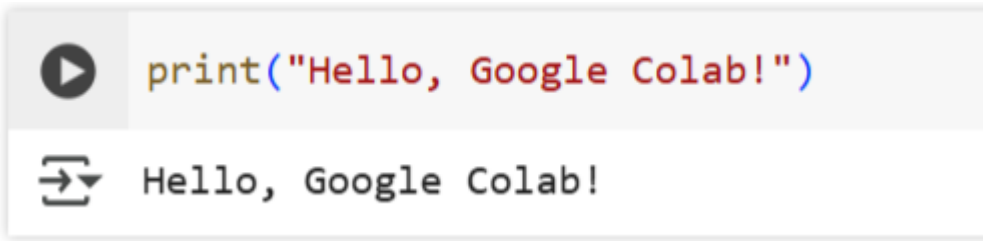


Рисунок 1. Запуск Google Colab

4. Выполнено форматирование в Markdown

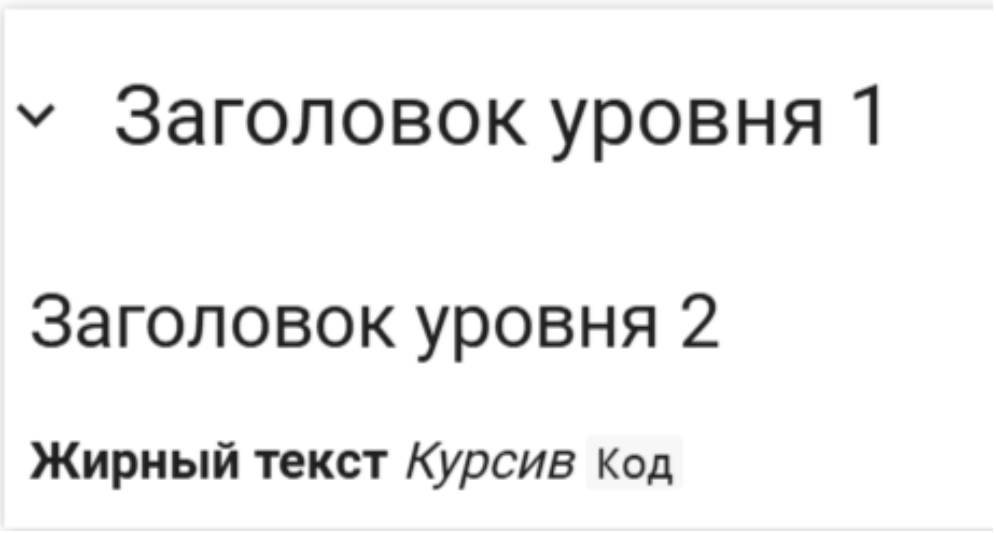


Рисунок 2. Форматирование в Markdown

5. Выполнено построение графика через библиотеку Python

```
plt.show()
```

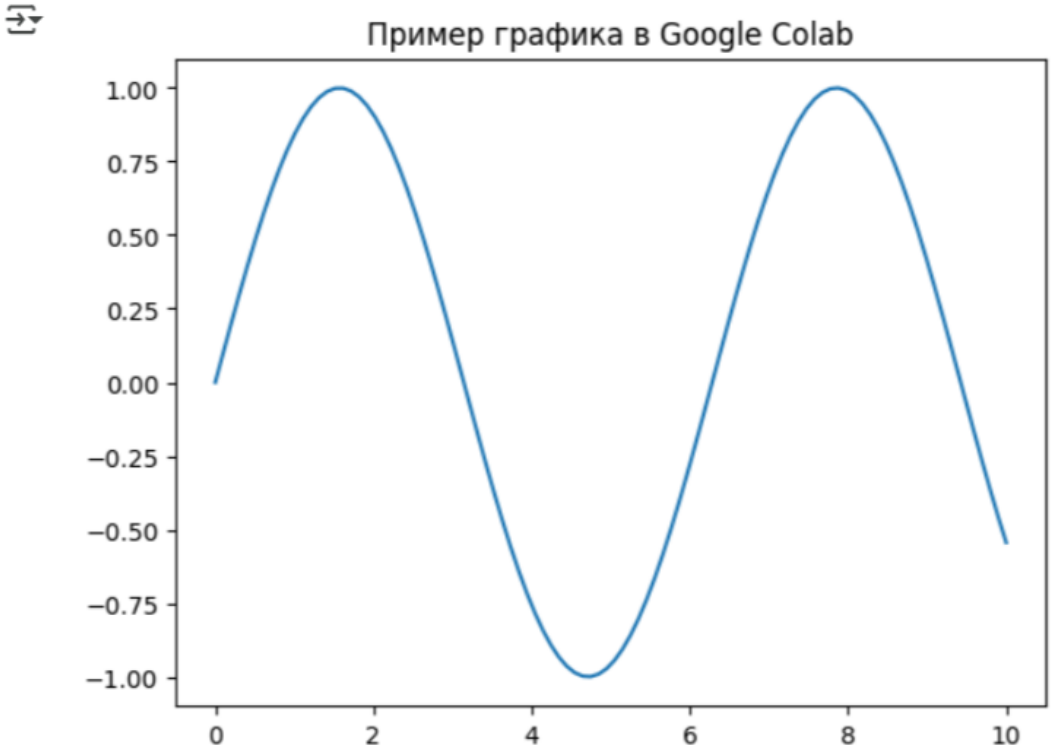


Рисунок 3. Пример построения графика через библиотеку Python

6. Выполнена загрузка файлов в Colab

```
from google.colab import files
uploaded = files.upload()  # Открывает окно загрузки файла
```

```
Выбрать файлы ЭМПиВ №1.docx
• ЭМПиВ №1.docx(application/vnd.openxmlformats-officedocument.wordprocessingml.document)
- 220742 bytes, last modified: 28.02.2025 - 100% done
Saving ЭМПиВ №1.docx to ЭМПиВ №1.docx
```

Рисунок 4. Загрузка файлов в Colab

7. Выполнено сохранение файлов в Colab

```
with open("whatever.txt", "w") as f:
    f.write("Привет, Google Colab!")

files.download("whatever.txt")  # Скачивание файла
```

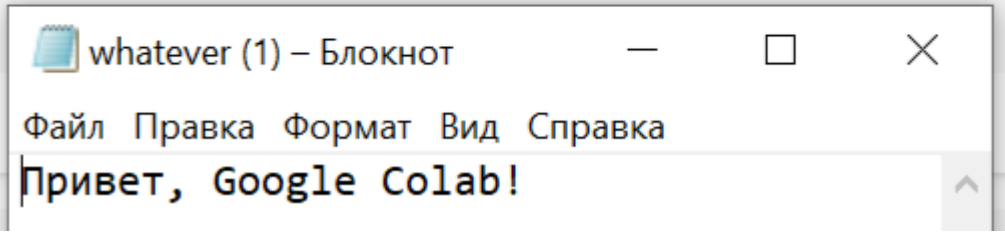


Рисунок 5. Сохранение файлов в Colab

8. Выполнено подключение Google Drive

```
from google.colab import drive
drive.mount('/content/drive')  # Подключение Google Диска
```

```
Mounted at /content/drive
```

Рисунок 6. Подключение Google Drive

9. Скрининг подключения Google Drive

```
[16] !ls "/content/drive/My Drive/"
```

```
'Colab Notebooks'  'Контрольная по литературе 8 класс.gdoc'   РАСПИСАНИЕ.pdf  'Тамара 186.rar'
```

Рисунок 7. Выполнение кода

10. Выполнено использование терминала

```
!ls    # Посмотреть файлы в текущей директории
!pwd   # Узнать путь к текущей папке
!pip list  # Посмотреть установленные библиотеки
```

```
sample_data
/content
Package                           Version
--------------------------------- ------------------
absl-py                           1.4.0
accelerate                        1.3.0
aiohappyeyeballs                  2.5.0
aiohttp                           3.11.13
aiosignal                         1.3.2
alabaster                         1.0.0
```

Рисунок 8. Использование терминала

11. Пример магической команды Jupyter

```
[3]  %timeit sum(range(1000))  # Замер времени выполнения
```

```
18 µs ± 4.63 µs per loop (mean ± std. dev. of 7 runs, 10000 loops each)
```

Рисунок 9. Магическая команды Jupyter

12.Выполнено взаимодействие с конкретным файлом

```
[19] import os
     os.chdir("/content/drive/My Drive/Colab Notebooks")
```

```
!pwd
```

```
/content/drive/MyDrive/Colab Notebooks
```

Рисунок 10. Взаимодействие с конкретным файлом

13. Работа с Jupyter lab

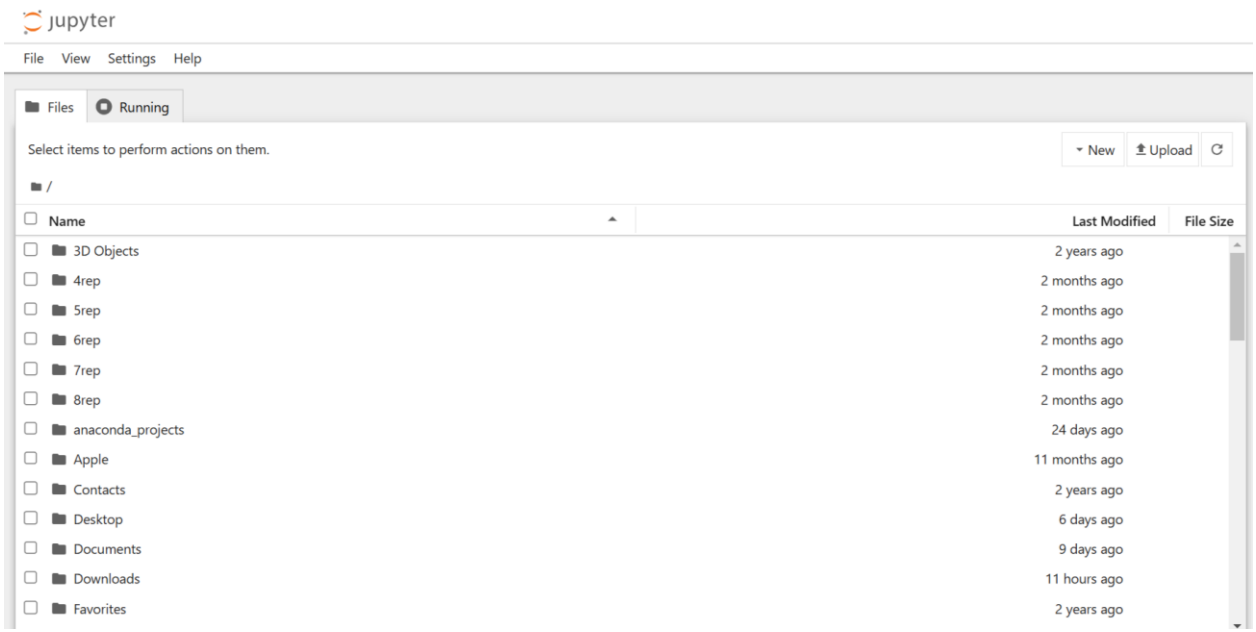


Рисунок 11. Jupyter lab

14. Создание ноутбука

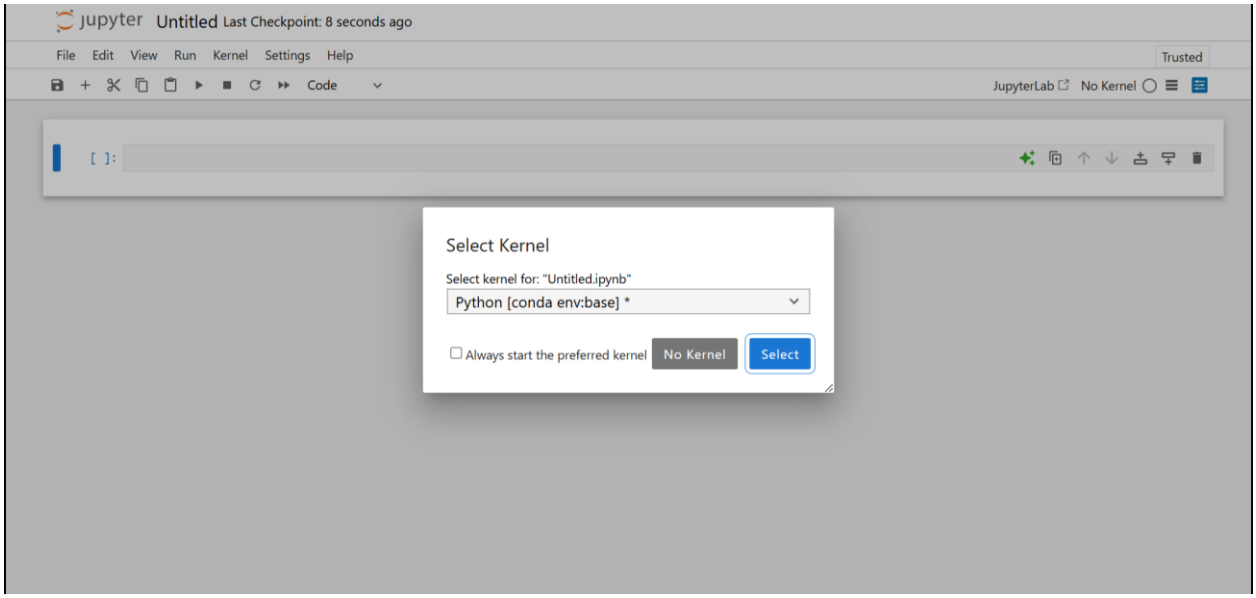


Рисунок 12. Новый ноутбук

15. Выполнено индивидуальное задание

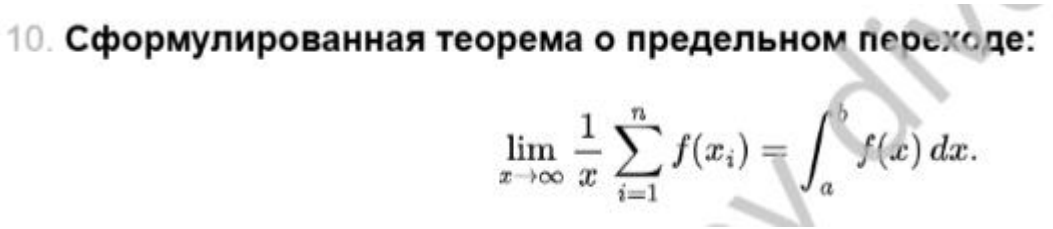


Рисунок 13. Индивидуальное задание

```
Приближение суммы: 0.33350016683350014
Значение интеграла: 0.33333333333333337
```

```
[6]  x = np.linspace(a, b, 1000)
     y = f(x)

     plt.plot(x, y, label="f(x)", color="blue")
     plt.fill_between(x_values, f(x_values), alpha=0.3, color="red", label="Сумма Римана")
     plt.axhline(y=integral_value, color="green", linestyle="--", label="Интеграл")

     plt.legend()
     plt.xlabel("x")
     plt.ylabel("f(x)")
     plt.title("Приближение интеграла суммой")
     plt.show()
```

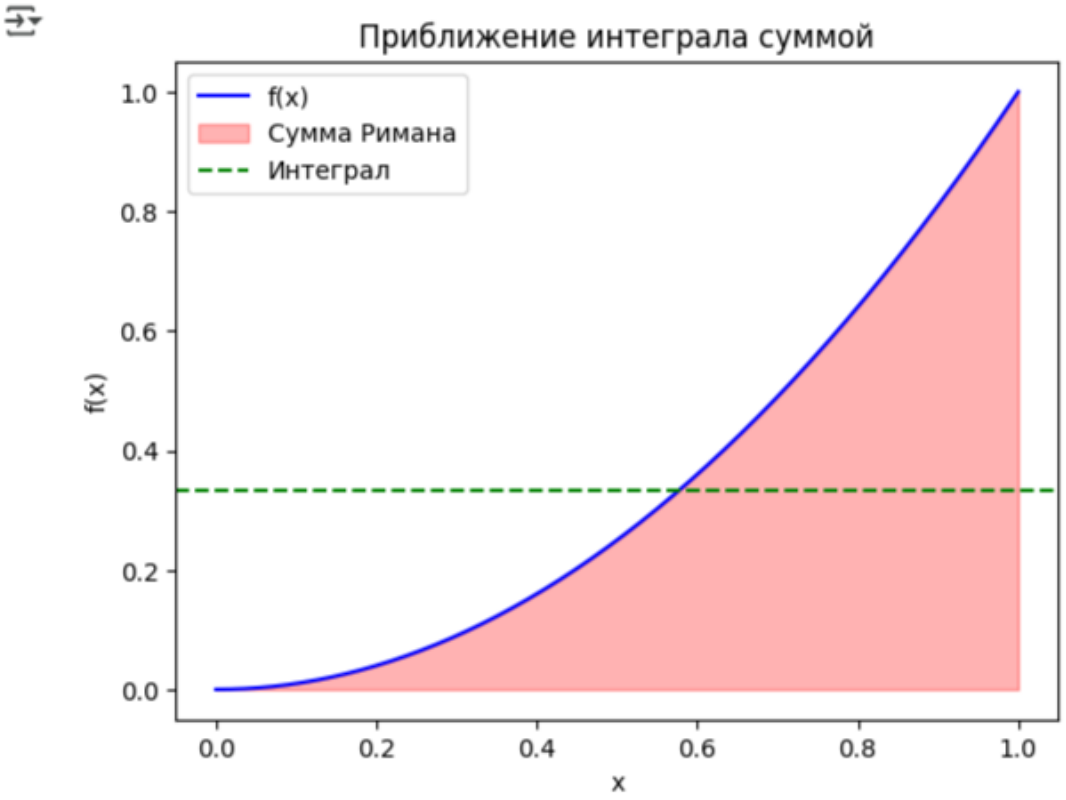


Рисунок 14. Выполненное задание

**Ответы на контрольные вопросы:**

1. Какие основные отличия JupyterLab от Jupyter Notebook?

- JupyterLab — это более мощная среда, чем Jupyter Notebook.

- Поддерживает многовкладочный интерфейс, в отличие от классического Jupyter Notebook.

- Включает терминал, редактор кода, работу с файлами.

- Можно открывать несколько файлов одновременно и переключаться между ними.

- Гибкое управление вкладками и панелями для удобства работы.

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

- Открыть JupyterLab.

- Нажать "File" → "New Notebook".

- Выбрать нужное ядро (Kernel), например Python.

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

- Code (Код) — для выполнения кода Python.

- Markdown — для текстов и формул.

- Raw — для необработанных данных.

- Переключение: "Cell" → "Cell Type" или с помощью горячих клавиш (Esc → M для Markdown, Esc → Y для Code).

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

- Shift + Enter — выполнить код и перейти к следующей ячейке.

- Ctrl + Enter — выполнить код и остаться в текущей ячейке.

- Alt + Enter — выполнить код и создать новую ячейку.

5. Как запустить терминал или текстовый редактор внутри JupyterLab?

- Открыть "File" → "New Launcher".

- Выбрать "Terminal" для командной строки.

- Для текстового редактора: "File" → "New Text File".

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

- Файловый браузер (File Browser) слева.

- Возможность создавать, перемещать, удалять файлы.

- Поддержка drag-and-drop для удобной работы.

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

- Через "Kernel" → "Restart Kernel" (Перезапустить).

- "Kernel" → "Interrupt Kernel" (Остановить выполнение).

- Можно управлять из "Running Terminals and Kernels".

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

- Можно открывать несколько вкладок одновременно.

- Размещать окна рядом или друг над другом.

- Удобное управление через drag-and-drop.

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

- `%time` — измерить время выполнения одной строки.

- `%timeit` — выполнить код несколько раз и измерить среднее время.

- `%%time` — измерить время выполнения всей ячейки.

- `%%timeit` — многократное выполнение всей ячейки.

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

- `%%bash` — выполнение команд Bash.

- `%%ruby` — выполнение кода на Ruby.

- `%%perl` — выполнение кода на Perl.

- `%%javascript` — выполнение JavaScript.

11. Какие основные отличия Google Colab от JupyterLab?

- Работает в браузере без установки на ПК.

- Поддерживает облачные GPU и TPU.

- Автоматическое сохранение в Google Drive.

- Позволяет совместную работу.

12. Как создать новый ноутбук в Google Colab?

- Перейти на [Google Colab](https://colab.research.google.com).

- Нажать "Файл" → "Создать новый ноутбук".

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

- Код (`Code`) — для выполнения Python.

- Markdown — для текста и формул.

- Переключение: Меню "Ячейка" → "Тип ячейки".

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

- Shift + Enter — выполнить код и перейти к следующей ячейке.

- Ctrl + Enter — выполнить код без перехода.

- Alt + Enter — выполнить код и создать новую ячейку.

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

- Файлы можно загружать через Файловый браузер.

- Использовать Google Drive.

- Загружать файлы через команду:

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

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

- Подключение Google Диска:

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

- После этого файлы доступны в `/content/drive/My Drive/`.

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

- Загрузить файл:

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

- Сохранить файл:

```
python
with open("example.txt", "w") as f:
    f.write("Пример сохранения файла")
files.download("example.txt")
```

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

- Через файловый браузер в левом меню.
- Вызвать команду:

```
sh
!ls –lh
```

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

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

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

- `%timeit` и `%%timeit` — выполняют код несколько раз для точного измерения.

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

- Перейти в меню "Среда выполнения" → "Сменить среду выполнения".

- В разделе "Аппаратный ускоритель" выбрать:

- GPU — для ускорения вычислений.

- TPU — для задач машинного обучения.

**Вывод:** JupyterLab предоставляет локальную среду с расширенными возможностями управления файлами, терминалом и поддержкой множества языков программирования. Он удобен для работы с большими проектами и требует установки на компьютер.

Google Colab, в отличие от JupyterLab, работает в облаке, не требует установки и предоставляет доступ к GPU/TPU. Он особенно полезен для работы с нейросетями, анализа данных и совместных проектов.

Оба инструмента поддерживают магические команды, позволяют выполнять код с горячими клавишами и работать с файлами. В Google Colab есть интеграция с Google Drive, что упрощает доступ к данным.

Ссылка на Git Hub: https://github.com/darina-rtm/first-laba-ai.git

Ссылка на Google Colab: https://colab.research.google.com/drive/1Yyz2q-XILNyJc_BUqTsS1eELEJ6M8_0u#scrollTo=vm7MMcEFMecw