

Министерство науки и высшего образования Российской Федерации
Федеральное государственное автономное образовательное учреждение
высшего образования
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Департамент цифровых, робототехнических
систем и электроники
Институт перспективной инженерии

ОТЧЕТ
ПО ЛАБАРАТОРНОЙ РАБОТЕ №1
дисциплины «Искусственный интеллект и машинное обучение»

Выполнил:

Стародубцев Дмитрий Андреевич 2
курс, группа ИВТ-б-о-23-2,09.03.01
«Информатика и вычислительная
техника», очная форма обучения.

(подпись)

Руководитель практики:

Доцент департамента цифровых,
робототехнических систем и
электроники института перспективной
инженерии Воронкин Роман
Александрович

(подпись)

Отчет защищен с оценкой _____ Дата защиты _____

Ставрополь, 2025 г.

Тема: работа с Jupyter Notebook, JupyterLab Google Colab.

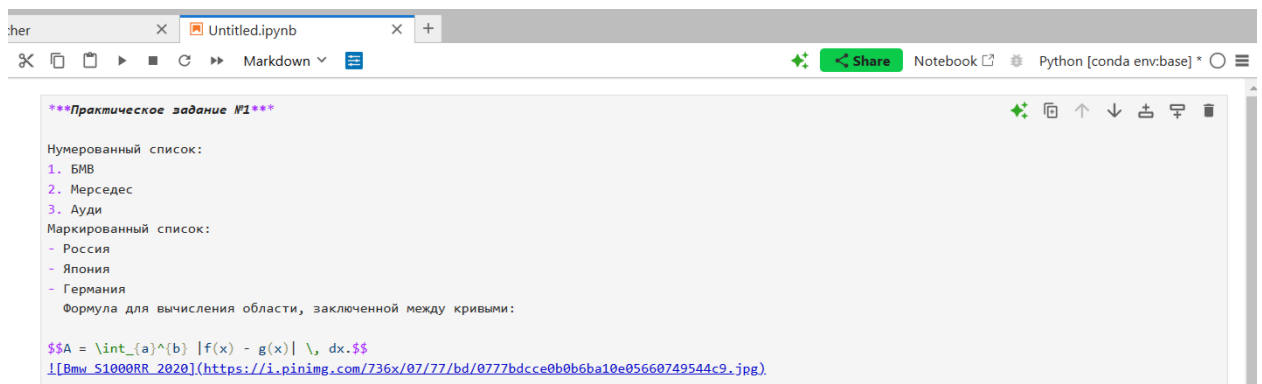
Цель: исследовать базовые возможности интерактивных оболочек Jupyter Notebook, JupyterLab и Google Colab для языка программирования Python.

Ссылка на репозиторий: <https://github.com/bearncfu/OLD>

Ход работы

Задание 1. Освоить работу с различными типами ячеек (код, Markdown) и форматированием текста.

Создал новую Markdown-ячейку и: написал заголовок "Практическое задание №1". Добавил жирный и курсивный текст. Создал нумерованный и маркированный списки. Вставил согласно индивидуального задания пример. Вставил изображение через ![Описание](URL) .



The screenshot shows a Jupyter Notebook window with a tab titled 'Untitled.ipynb'. The interface includes a toolbar with icons for undo, redo, and other actions. The main area contains a Markdown cell with the following content:

```
***Практическое задание №1***

Нумерованный список:
1. БМВ
2. Мерседес
3. Ауди
Маркированный список:
- Россия
- Япония
- Германия
Формула для вычисления области, заключенной между кривыми:


$$A = \int_a^b |f(x) - g(x)| \, dx$$


![Bmw_S1000RR 2020](https://i.pinimg.com/736x/07/77/bd/0777bdcce0b0b6ba10e05660749544c9.jpg)
```

Рисунок 1 – Код первой половины 1 задания

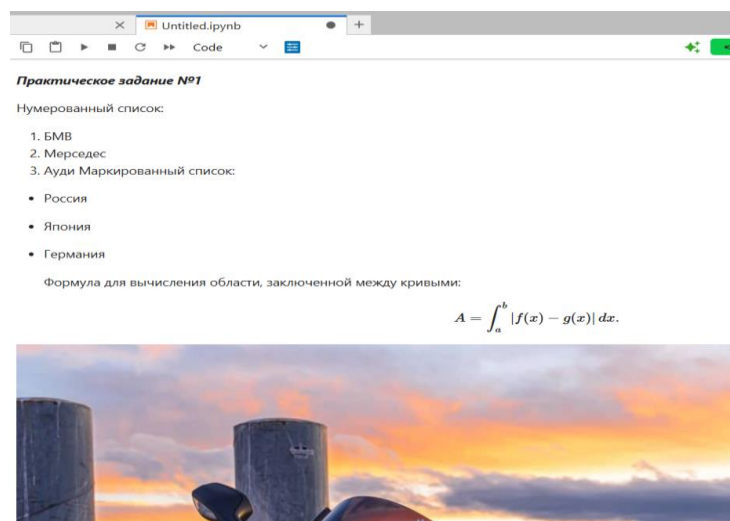


Рисунок 2 – Результат кода



Рисунок 3 – Продолжение результата

Создал ячейку Python-кода и: Запросил у пользователя его имя с помощью `input()`. Вывел приветствие: "Привет, <имя>! Добро пожаловать в JupyterLab!".

```
name = input("Пожалуйста, введите ваше имя: ")  
  
print(f"Привет, {name}! Добро пожаловать в JupyterLab!")
```

Рисунок 4 – Код нового задания

Пожалуйста, введите ваше имя: Дмитрий
Привет, Дмитрий! Добро пожаловать в JupyterLab!

Рисунок 5 – Результат нового задания

Задание 2. Изучить загрузку, создание и сохранение файлов в JupyterLab.

1. Создал и сохранил текстовый файл с помощью open().
2. Записал в него несколько строк текста.
3. Закрыв файл и затем открыл его снова, считав содержимое и выведя на экран.
4. Проверил, существует ли файл, используя os.path.exists().
5. Удалил файл с помощью модуля os.

```
import os

# 1. Создал файл и записал в него текст
with open("example.txt", "w", encoding="utf-8") as file:
    file.write("Первая строка текста.\n")
    file.write("Вторая строка текста.\n")
    file.write("Третья строка текста.\n")

# 2. Прочитал и вывел содержимое файла
with open("example.txt", "r", encoding="utf-8") as file:
    content = file.read()
    print("Содержимое файла:")
    print(content)

# 3. Проверил существование файла
if os.path.exists("example.txt"):
    print("Файл 'example.txt' существует.")
else:
    print("Файл 'example.txt' не существует.")

# 4. Удалил файл
if os.path.exists("example.txt"):
    os.remove("example.txt")
    print("Файл 'example.txt' удален.")
else:
    print("Файл 'example.txt' не существует, удаление не требуется.")
```

Рисунок 6 – Код 2 задания

Содержимое файла:

Первая строка текста.

Вторая строка текста.

Третья строка текста.

Файл 'example.txt' существует.

Файл 'example.txt' удален.

Рисунок 7 – Результат 2 задания

Задание 3. Изучить магические команды для удобной работы в JupyterLab.

1. Вывел список всех доступных магических команд (`%lsmagic`).
2. Использовал `%time` и `%%timeit` для измерения времени выполнения кода.
3. Создал Python-скрипт в Jupyter (`%%writefile script.py`) и выполнил его через `!python script.py` .
4. Вывел список файлов в текущей директории с помощью `%ls` .
5. Использовал `%history` для просмотра истории команд.

```
# 1. Вывел список магических команд
print("1. Список магических команд:")
%lsmagic

# 2. Измерил время выполнения кода
print("\n2. Измерение времени выполнения:")
print("Время выполнения sum(range(1000)):")
%time sum(range(1000))
```

Рисунок 8 – Список магических команд и время их выполнения

```
1. Список магических команд:
root
line
cell
```

Рисунок 9 – Результат выполнения кода с магическими командами

```
2. Измерение времени выполнения:
Время выполнения sum(range(1000)):
CPU times: total: 0 ns
Wall time: 0 ns
499500
```

Рисунок 10 – Результат выполнение кода с временем выполнения

```
%%timeit
total = 0
for i in range(1000):
    total += i
134 µs ± 2.37 µs per loop (mean ± std. dev. of 7 runs, 10,000 loops each)
```

Рисунок 11 – Код 2 задания

134 µs ± 2.37 µs per loop (mean ± std. dev. of 7 runs, 10,000 loops each)

Рисунок 12 – Результат кода 2 задания

```
%%writefile script.py
print("Привет, это мой скрипт!")
for i in range(5):
    print(f"Итерация {i}")
```

Overwriting script.py

```
!python script.py
```

Рисунок 12 – Код 3 задания

```
Привет, это мой скрипт!
Итерация 0
Итерация 1
Итерация 2
Итерация 3
Итерация 4
```

Рисунок 13 – Результат 3 задания

```
# 4. Вывел список файлов в текущей директории
print("\n4. Список файлов в текущей директории:")
%ls
```

Рисунок 14 – Код 4 задания

4. Список файлов в текущей директории:
Том в устройстве C не имеет метки.
Серийный номер тома: F404-D238

Содержимое папки C:\Users\Admin

27.02.2025	23:00	<DIR>	.
27.02.2025	23:00	<DIR>	..
14.02.2025	14:19	<DIR>	.anaconda
27.02.2025	23:02	<DIR>	.conda
14.02.2025	14:20		146 .condarc
14.02.2025	14:19	<DIR>	.continuum
08.02.2024	13:02	<DIR>	.dotnet
02.06.2023	13:20	<DIR>	.idlerc
27.02.2025	22:32	<DIR>	.ipynb_checkpoints
27.02.2025	20:32	<DIR>	.ipython

Рисунок 15 – Результат 4 задания

```
# 5. Просмотрел историю команд
print("\n5. История команд:")
%history
```

Рисунок 16 – Код 5 задания

```
5. История команд:
# 1. Вывод списка магических команд
print("1. Список магических команд:")
%lsmagic

import json
import getpass
import hashlib

def import_pandas_safely():
    try:
        return __import__('pandas')
    except ImportError:
        return False

__pandas = import_pandas_safely()
```

Рисунок 17 – Результат 5 задания

Задание 4. Изучить выполнение команд терминала прямо из JupyterLab.

1. Вывел список файлов в текущей директории с помощью !ls.
2. Проверил, какой Python используется (!which python).

3. Создал папку test_folder (!mkdir test_folder) и убедился, что она появилась.
4. Переместил файл в новую папку и удалил его.
5. Очистил вывод в ячейке (!clear).

```
# 1. Создал папку test_folder
!mkdir test_folder
print("Папка 'test_folder' создана.")

# 2. Проверил, что папка появилась
print("\nСодержимое текущей директории:")
!dir

# 3. Создал файл для перемещения
file_name = "temp_file.txt"
with open(file_name, "w") as f:
    f.write("Это временный файл.")
print(f"\nФайл '{file_name}' создан.")

# 4. Переместил файл в папку test_folder
!move {file_name} test_folder/
print(f"\nФайл '{file_name}' перемещён в папку 'test_folder'.")

# 5. Проверил содержимое папки test_folder
print("\nСодержимое папки 'test_folder':")
!dir test_folder

import os

# Указал путь к файлу
file_path = "C:\\Users\\Admin\\test_folder\\temp_file.txt"

# Удалил файл с помощью команды del и проверил, удалась ли она
if os.path.exists(file_path):
    os.system(f'del "{file_path}"')
    print(f"Файл '{file_path}' успешно удалён.")
else:
    print(f"Файл '{file_path}' не найден.")
!dir test_folder
```

Рисунок 18 – Код 4 задания


```

Папка 'test_folder' создана.
Содержимое текущей директории:
Подпапка или файл test_folder уже существует.
Том в устройстве C не имеет метки.
Серийный номер тома: F404-D238

Содержимое папки C:\Users\Admin
06.03.2025 21:28 <DIR> .
06.03.2025 21:28 <DIR> ..
01.10.2023 00:13 6 904 -1.14-windows.xml
14.02.2025 14:19 <DIR> .anaconda
06.03.2025 21:28 <DIR> .conda
14.02.2025 14:20 146 .condarc
14.02.2025 14:19 <DIR> .continuum
08.02.2024 13:02 <DIR> .dotnet
02.06.2023 13:20 <DIR> .idlerc
06.03.2025 20:44 <DIR> .ipynb_checkpoints
27.02.2025 20:32 <DIR> .ipython
14.02.2025 14:21 <DIR> .jupyter
15.10.2023 19:36 <DIR> .ld9VirtualBox
24.12.2024 22:17 <DIR> .matplotlib
06.07.2023 10:20 <DIR> .ms-ad
28.02.2025 15:59 176 .packettracer
25.04.2024 23:48 <DIR> .pencil
08.02.2024 13:55 <DIR> .templateengine
18.02.2024 00:16 <DIR> .thumbnails
20.12.2024 01:44 <DIR> .VirtualBox
06.03.2025 20:44 <DIR> .virtual_documents
23.09.2024 15:19 <DIR> .vscode
06.03.2025 20:37 6 230 1 laba.ipynb
28.02.2025 14:46 2 468 1_2 laba.ipynb
28.02.2025 14:46 95 532 1_3 laba.ipynb
06.03.2025 21:03 15 467 1_4 laba.ipynb
31.05.2023 16:10 <DIR> 3D Objects
14.02.2025 14:21 <DIR> anaconda3

16 файлов 144 900 байт
43 папок 44 217 782 272 байт свободно

Файл 'temp_file.txt' создан.
Перемещено файлов: 1.

Файл 'temp_file.txt' перемещен в папку 'test_folder'.

Содержимое папки 'test_folder':
Том в устройстве C не имеет метки.
Серийный номер тома: F404-D238

Содержимое папки C:\Users\Admin\test_folder
06.03.2025 21:28 <DIR> .
06.03.2025 21:28 <DIR> ..
06.03.2025 21:28 19 temp_file.txt
1 файл 19 байт
2 папок 44 217 782 272 байт свободно
Файл 'C:\Users\Admin\test_folder\temp_file.txt' успешно удалён.
Том в устройстве C не имеет метки.
Серийный номер тома: F404-D238

Содержимое папки C:\Users\Admin\test_folder
06.03.2025 21:28 <DIR> .
06.03.2025 21:28 <DIR> ..
0 файлов 0 байт
2 папок 44 217 782 272 байт свободно

```

Рисунок 19 – Результат 4 задания

Задание 5. Работа с Google Drive Google Colab.

1. Подключил Google Drive к Colab с помощью команды:

```
from google.colab import drive
```

```
drive.mount('/content/drive').
```

Проверил, что мой диск успешно подключился, используя /content/drive/MyDrive.

2. Создал и сохранил текстовый файл в Google Drive:

Откройте файл my_text_file.txt в режиме записи.

Запишите в него несколько строк текста.

Закройте файл и убедитесь, что он появился в папке MyDrive .

3. Прочитал файл из Google Drive:

Откройте ранее созданный файл.

Считайте его содержимое и выведите в ячейке.

4. Создал и сохранил CSV-файл вручную, используя Microsoft Excel или Libre Office Calc:

Создайте список, в котором будут строки с данными о студентах (например, ФИО, возраст, группа).

Запишите этот список в CSV-файл в Google Drive вручную, используя стандартные методы записи в файл.

```
# 1. Подключил Google Drive
from google.colab import drive
drive.mount('/content/drive')

# Проверил подключение
!ls /content/drive/MyDrive

# 2. Создал текстовый файл
file_1 = "/content/drive/MyDrive/my_text_file.txt"
with open(file_1, "w", encoding="utf-8") as file:
    file.write("Это первая строка текста.\n")
    file.write("Это вторая строка текста.\n")
print(f"Файл '{file_1}' успешно создан.")

# 3. Чтение текстового файла
with open(file_1, "r", encoding="utf-8") as file:
    content = file.read()
print(f"Содержимое файла:")
print(content)

# 4. Чтение CSV-файла
students = [
    ["Иванов И.И.", 20, "ИВТ-101"],
    ["Петров П.П.", 22, "ИВТ-102"],
    ["Сидорова А.А.", 21, "ИВТ-103"]
]

# Путь к CSV-файлу в Google Drive
csv_path = "/content/drive/MyDrive/students.csv"

# Записал список в CSV-файл
with open(csv_path, "w", encoding="utf-8") as file:
    for student in students:
        # Преобразовал каждый элемент списка в строку и объединил через запятую
        line = ",".join(map(str, student))
        file.write(line + "\n") # Записал строку в файл
print(f"Файл '{csv_path}' успешно создан и заполнен.")
```

Рисунок 20 – Код 5 задания

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force_remount=True).
'Colab Notebooks' my_text_file.txt students.csv
Файл '/content/drive/MyDrive/my_text_file.txt' успешно создан.
Содержимое файла:
Это первая строка текста.
Это вторая строка текста.

Файл '/content/drive/MyDrive/students.csv' успешно создан и заполнен.
```

Рисунок 21 – Результат 5 задания

Контрольные вопросы:

1) Какие основные отличия JupyterLab от Jupyter Notebook?

JupyterLab и Jupyter Notebook — это инструменты для работы с интерактивными тетрадями, но JupyterLab является их более продвинутой версией. Вот ключевые различия:

1. Интерфейс и организация работы: о Jupyter Notebook: представляет собой отдельные веб-страницы с линейной последовательностью ячеек. Окружение ограничено одной тетрадью. о JupyterLab: это полноценная интегрированная среда (IDE) с вкладками, панелями, файловым менеджером, терминалом и редактором кода. Можно открывать и редактировать несколько файлов одновременно.

2. Работа с файлами:

В Jupyter Notebook можно работать только с .ipynb-файлами.

В JupyterLab можно работать с разными типами файлов: .ipynb, .py, .csv, .md и даже .json, .yaml и .txt.

3. Многозадачность: о В Jupyter Notebook приходится открывать несколько вкладок браузера для работы с разными тетрадями. о В JupyterLab можно работать с несколькими тетрадями в одном окне благодаря системе вкладок и разделению экрана.

4. Поддержка расширений: о Jupyter Notebook поддерживает расширения, но их сложнее настраивать. о JupyterLab имеет встроенный менеджер расширений, что делает добавление новых функций (например, поддержки дополнительных языков программирования, тем и плагинов) более удобным.

5. Гибкость интерфейса: о В Jupyter Notebook фиксированный интерфейс. о В JupyterLab можно изменять расположение панелей, настраивать вид и работать в нескольких окнах.

6. Поддержка терминала: о В Jupyter Notebook запуск терминала требует отдельных шагов. о В JupyterLab есть встроенный терминал, позволяющий выполнять команды прямо в интерфейсе.

7. Производительность: о JupyterLab может потреблять больше ресурсов, особенно если открыто много вкладок. о Jupyter Notebook проще и легче, но менее функционален.

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

1. В меню выбрать File → New → Notebook

2. Выбрать доступно ядро (обычно Python)

3. Откроется новая тетрадь, состоящая из ячеек (Cells)

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

Код (Code) – для написания и выполнения программного кода.

Текст (Markdown) – используется для оформления пояснений, форматированного текста и математических формул на основе LaTeX.

Вывод (Raw) – предназначен для хранения необработанного текста, например, для экспорта в другие форматы.

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

Ctrl+Enter или Shift+Enter. В первом случае введенный вами код будет выполнен интерпретатором Python, во втором – будет выполнен код и создана новая ячейка, которая расположится уровнем ниже.

5) Как запустить терминал или текстовый редактор внутри JupyterLab?

Терминал доступен через File → New → Terminal, используется для выполнения команд оболочки.

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

Используется файловый браузер для навигации по каталогам и управления файлами. Также доступны команды терминала для работы с файловой системой.

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

В меню "Kernel" можно перезапустить, остановить или сменить ядро. Также можно управлять ядрами через терминал.

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

JupyterLab позволяет открывать несколько вкладок и окон для одновременной работы с разными файлами и инструментами. Вкладки можно перетаскивать и организовывать по своему усмотрению.

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

Можно использовать `%%time` и `%timeit`. `%%time` позволяет получить информацию о времени работы кода в рамках одной ячейки.

`%timeit` запускает переданный ей код 100000 раз (по умолчанию) и выводит информацию о среднем значении трех наиболее быстрых прогонов.

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

Можно использовать `%%` с указанием языка, например, `%%javascript`.

11) Какие основные отличия Google Colab от JupyterLab?

Google Colab работает в облаке и предоставляет доступ к GPU и TPU бесплатно.

Colab интегрирован с Google Drive и другими сервисами Google.

JupyterLab требует локальной установки или сервера.

12) Как создать новый ноутбук в Google Colab?

Перейти в Файл → Новый ноутбук. Откроется рабочая область с первой ячейкой.

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

Код (Code) – для написания и выполнения Python-кода.

Текст (Markdown) – используется для оформления документации, пояснений и формул (LaTeX).

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

Запустить ячейку: Shift + Enter.

Добавить новую ячейку ниже: Ctrl + M B.

Удалить текущую ячейку: Ctrl + M D.

Изменить тип ячейки на Markdown: Ctrl + M N.

Изменить тип ячейки на код: Ctrl + M Y.

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

Файлы можно загружать вручную через боковую панель (Файлы → Загрузить файлы) или с помощью Python. Команда `files.upload()` для загрузки файлов с локального компьютера. Для сохранения файлов можно использовать команду `files.download()`.

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

Google Colab позволяет работать с файлами на Google Диске и загружать файлы в локальное окружение.

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

После выполнения появится ссылка, по которой нужно авторизоваться.

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

Файлы можно загружать вручную через боковую панель `from google.colab import files` `uploaded = files.upload()`

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

Можно использовать команду `!ls` в ячейке кода для просмотра файлов в текущем каталоге.

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

Можно использовать `%timeit` для измерения времени выполнения кода и `%%time` для измерения времени выполнения всей ячейки. `%timeit sum(range(1000))` `%%time total = sum(range(10**6))`.

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

Перейти в Среда выполнения → Изменить среду выполнения.

В поле Аппаратный ускоритель выбрать:

GPU (для графического ускорения)

TPU (для ускорения в TensorFlow) Нажать Сохранить.

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