

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»
Вариант № 4

Выполнил:
Левашев Тимур Рашидович
2 курс, группа ИВТ-б-о-23-2,
09.03.01 «Информатика и
вычислительная техника»,
направленность (профиль)
«Программное обеспечение средств
вычислительной техники и
автоматизированных систем», очная
форма обучения

(подпись)

Проверил:
Доцент департамента цифровых,
робототехнических систем и
электроники института перспективной
инженерии Воронкин В.И

(подпись)

Отчет защищен с оценкой _____ Дата защиты _____

Ставрополь, 2025 г.

Тема работы: “Рабоа с Jupyter Notebook, JupyterLab Google Colab”

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

Порядок выполнения работы:

1. Создание Jupyter файла с помощью JupyterLab.

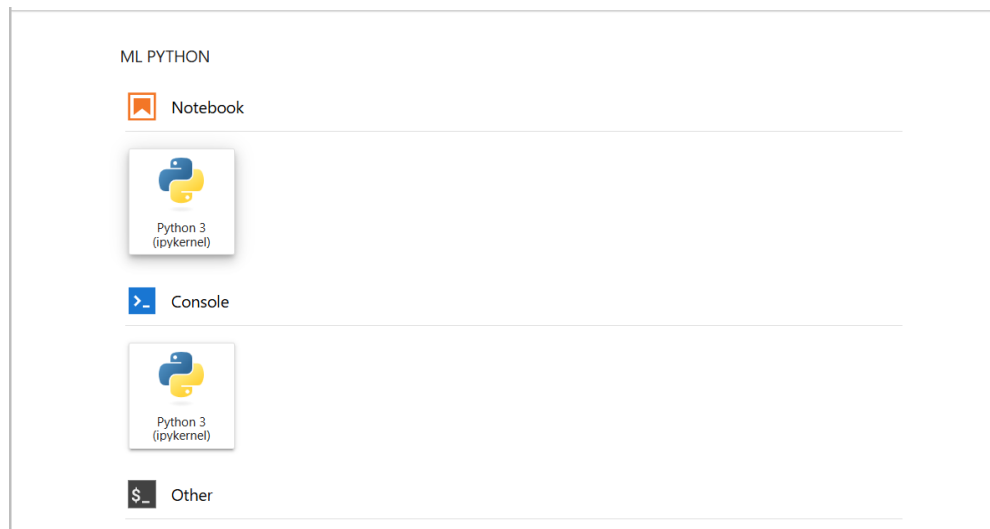


Рисунок 1 – Создание файла

2. Выполнение примеров из методички.

```
[1]: 3+2
[1]: 5
[2]: a=5
     b=7
     print(a+b)
12
[3]: n=7
     for i in range(n):
         print(i*10)
0
10
20
30
40
50
60
[4]: i=0
     while True:
         i+=1
         if i>5:
             break
         print("Test while")
```

Рисунок 2 – Примеры из методички

3. Создание графики в jupyter lab.

```
import matplotlib.pyplot as plt
%matplotlib
```

Using matplotlib backend: module://matplotlib_inline.backend_inline

```
x=[i for i in range(50)]
y=[i**2 for i in range(50)]
plt.plot(x,y)
```

[<matplotlib.lines.Line2D at 0x223a1faf650>]

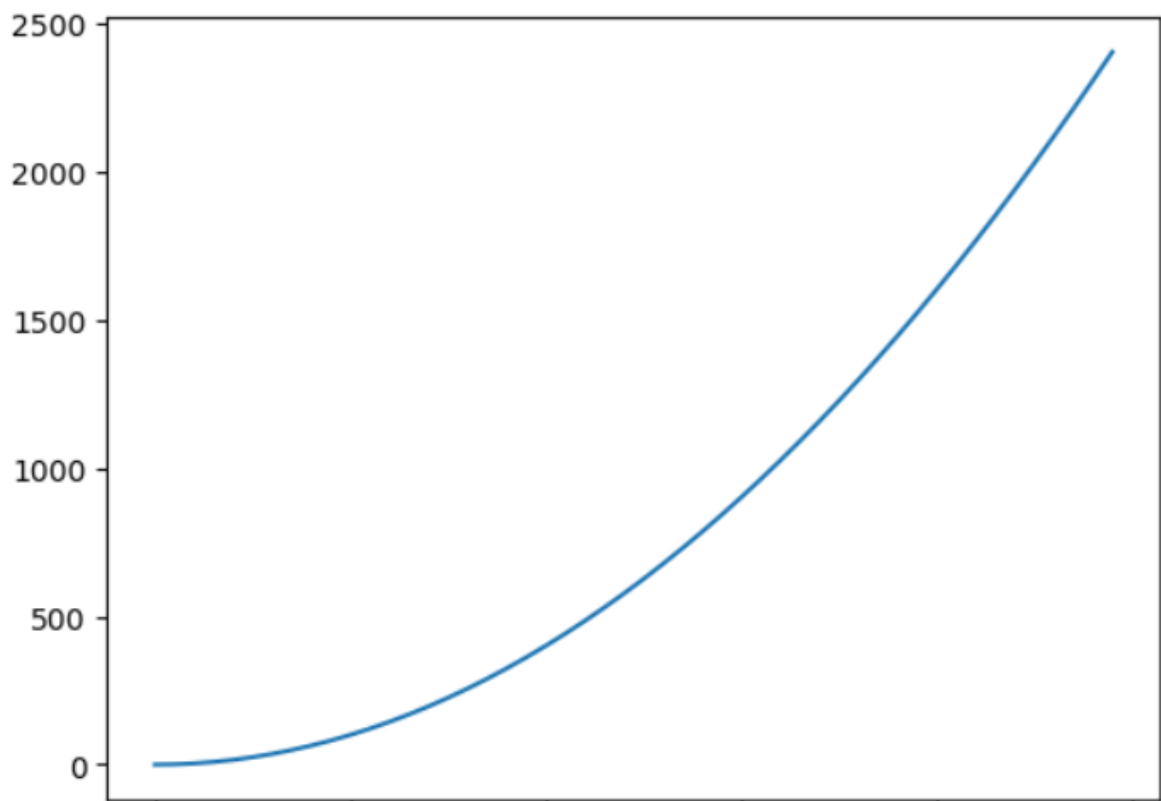


Рисунок 3 – Создание графика функции

4. Магические команды в jupyter lab.

```
1): %lsmagic
```

```
1): root
  cell
    ! "OSMagics"
    HTML "Other"
    SVG "Other"
    bash "Other"
    capture "ExecutionMagics"
    cmd "Other"
    code_wrap "ExecutionMagics"
    debug "ExecutionMagics"
    file "Other"
    html "DisplayMagics"
    javascript "DisplayMagics"
    js "DisplayMagics"
    latex "DisplayMagics"
    markdown "DisplayMagics"
    perl "Other"
    prun "ExecutionMagics"
    python "Other"
```

Рисунок 4 – Магические команды

5. Работа с файлами в jupyter lab.

```
[1]: import os

with open("my_file.txt", "w") as file:
    file.write("\n 1)Первая строка \n 2)Вторая строка")

with open("my_file.txt", "r") as file:
    print("Содержимое файла: ", file.read())

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

os.remove("my_file.txt")
print("Файл удален.")
```

Содержимое файла:
1)Первая строка
2)Вторая строка
Файл существует: True
Файл удален.

Рисунок 5 – Работа с файлами

6. Создание скриптов в jupyter lab.

```
1 for i in range(100):
2     print(f"Итерация {i}")
3
4 !python test_script.py
5
6 %time sum(range(1000000))
7
```

Рисунок 6 – Создание скриптов

7. Создание markdown ячеек и добавление в них формул.

```
***Практическое задание №1***
```

1. Первый пункт нумерованного списка
2. Второй пункт

* Ненумерованный список можно размечать звездочками
- Или минусами
+ Или плюсами

$$\int \cos(x) \, dx = \sin(x) + C$$

Рисунок 7 – Создание md ячеек

8. Создание ячейки с исполняемым кодом и добавление в нее программы.

```
[2]: name = input("Пожалуйста, введите свое имя")
print(f"Привет, {name}! Добро пожаловать в JupyterLab / Google Colab!")

Пожалуйста, введите свое имя Timur
Привет, Timur! Добро пожаловать в JupyterLab / Google Colab!
```

Рисунок 8 – Создание ячеек с кодом

Ссылка на git репозиторий: https://github.com/mazy99/ML_prakt

Ответы на контрольные вопросы:

1. Основные отличия JupyterLab от Jupyter Notebook?

1) **JupyterLab** – это более современная среда разработки, поддерживающая несколько вкладок и окон, тогда как **Jupyter Notebook** работает как последовательность страниц.

2) В JupyterLab можно открывать терминалы, текстовые редакторы и файлы без необходимости запускать отдельные браузерные вкладки.

3) Улучшенный файловый менеджер.

4) Расширяемость с помощью плагинов.

2. Создание новой рабочей среды (ноутбука) в JupyterLab?

1) Открыть JupyterLab.

2) Нажать "File" → "New Notebook" и выбрать нужное ядро (например, Python 3).

3. Типы ячеек в JupyterLab и их переключение:

1) **Code** (код)

2) **Markdown** (разметка)

3) **Raw** (сырой текст)

Переключение: через меню "Cell" → "Cell Type" или сочетания клавиш:

Esc + M – в Markdown

Esc + Y – в код

4. Как выполнить код в ячейке и горячие клавиши:

Shift + Enter – выполнить ячейку и перейти к следующей

Ctrl + Enter – выполнить без перехода

Alt + Enter – выполнить и создать новую ячейку

5. Запуск терминала или текстового редактора в JupyterLab:

В файловом браузере нажать "+" и выбрать **"Terminal"** или **"Text Editor"**.

6. Инструменты работы с файлами и каталогами в JupyterLab:

Файловый менеджер (слева) – можно открывать, перемещать и удалять файлы.

7. Управление ядрами (kernels) в JupyterLab:

Через **"Kernel"** → **"Restart Kernel"**

Панель **"Running Terminals and Kernels"** показывает активные процессы.

8. Система вкладок и окон в JupyterLab:

Позволяет открывать несколько файлов, терминалов и ноутбуков в одном окне.

Окна можно перемещать и группировать.

9. Магические команды для измерения времени выполнения кода:

`%time` – измеряет время выполнения одной строки.

`%%timeit` – измеряет среднее время выполнения блока кода.

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

`%%bash` – выполнение команд Bash

`%%perl` – Perl

`%%ruby` – Ruby

`%%javascript` – JavaScript

11. Какие основные отличия Google Colab от JupyterLab?

Colab – облачный сервис, JupyterLab – локальная среда.

В Colab есть бесплатный доступ к GPU и TPU.

Подключение к Google Drive.

12. Как создать новый ноутбук в Google Colab?

Перейти на Google Colab → "File" → "New Notebook".

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

Code (код)

Text (Markdown)

Переключение через меню или клавишами:

Ctrl + M M – текст

Ctrl + M Y – код

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

Shift + Enter – выполнить и перейти далее

Ctrl + Enter – выполнить без перехода

Alt + Enter – выполнить и создать новую ячейку

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

Файлы можно загружать вручную через панель "Files".

Кодом:

```
from google.colab import files
```

```
uploaded = files.upload()
```

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

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

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

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

```
from google.colab import files
```

```
files.upload()
```

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

```
!ls
```

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

```
%time
```

```
%%timeit
```

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

"Runtime" → "Change runtime type" → выбрать GPU или TPU.

Проверить GPU:

```
!nvidia-smi
```

Вывод: В ходе проделанной работы исследовал базовые возможности интерактивных оболочек Jupyter Notebook, JupyterLab и Google Colab для языка программирования Python