

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ  
ФЕДЕРАЦИИ**

**Федеральное государственное автономное  
образовательное учреждение высшего  
образования  
«СЕВЕРОКАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ  
УНИВЕРСИТЕТ»**

**Кафедра**

**инфокоммуникаций**

**Институт цифрового**

**развития**

**ОТЧЁТ**

**по лабораторной работе №3.1**

Дисциплина: «Технологии распознавания образов»

Тема: «Работа с IPython и Jupyter Notebook»

Выполнила:

студентка 2 курса

группы Пиж-б-о-21-1

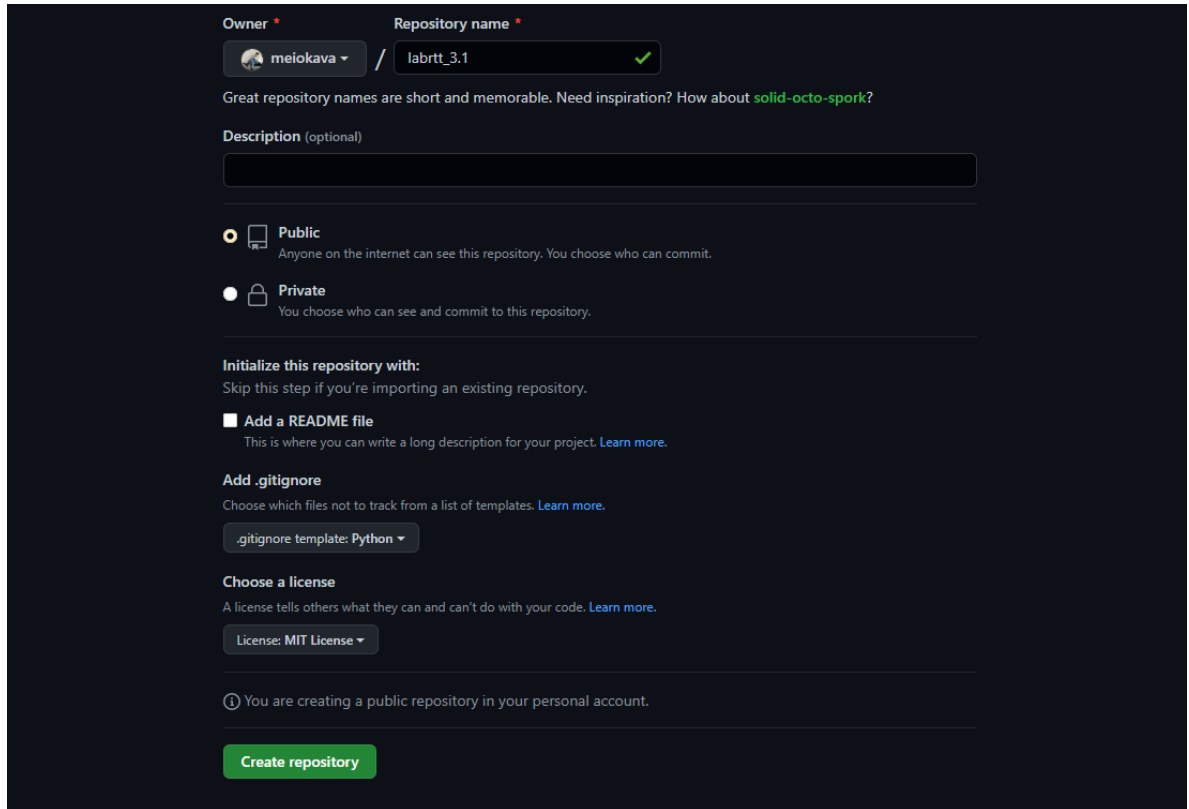
Джолдошова Мээрим

Бекболотовна

Ставрополь 2023

Цель работы: исследовать базовые возможности интерактивных оболочек IPython и Jupyter Notebook для языка программирования Python.

1. Был создан общедоступный репозиторий на GitHub, в котором использована лицензия MIT и выбран язык программирования python.



Owner \* meiokava / Repository name \* labrtt\_3.1 ✓

Great repository names are short and memorable. Need inspiration? How about [solid-octo-spork?](#)

Description (optional)

☒ Public  
Anyone on the internet can see this repository. You choose who can commit.

☐ Private  
You choose who can see and commit to this repository.

Initialize this repository with:  
Skip this step if you're importing an existing repository.

☒ Add a README file  
This is where you can write a long description for your project. [Learn more.](#)

Add .gitignore  
Choose which files not to track from a list of templates. [Learn more.](#)

.gitignore template: Python ▼

Choose a license  
A license tells others what they can and can't do with your code. [Learn more.](#)

License: MIT License ▼

*i* You are creating a public repository in your personal account.

Create repository

Рисунок 1 – Создание репозитория

2. Было выполнено клонирование созданного репозитория на рабочий компьютер.

```
C:\Users\мвидео>cd/d C:\gith\lb_3.1

C:\gith\lb_3.1>git clone https://github.com/meiokava/labrtt_3.1.git
Cloning into 'labrtt_3.1'...
remote: Enumerating objects: 4, done.
remote: Counting objects: 100% (4/4), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 4 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (4/4), done.

C:\gith\lb_3.1>|
```

Рисунок 2 – Клонирование репозитория

3. Организация репозитория в соответствии с моделью ветвления git-flow.

```
C:\gith\lb_3.1>cd/d C:\gith\lb_3.1\labrtt_3.1
C:\gith\lb_3.1\labrtt_3.1>git flow init

Which branch should be used for bringing forth production releases?
- main
Branch name for production releases: [main]
Branch name for "next release" development: [develop]

How to name your supporting branch prefixes?
Feature branches? [feature/]
Bugfix branches? [bugfix/]
Release branches? [release/]
Hotfix branches? [hotfix/]
Support branches? [support/]
Version tag prefix? []
Hooks and filters directory? [C:/gith/lb_3.1/labrtt_3.1/.git/hooks]

C:\gith\lb_3.1\labrtt_3.1>
```

Рисунок 3 – Организация модели ветвления git-flow

4. Был дополнен файл .gitignore правилами для выбранного языка программирования, интерактивной оболочки Jupyter notebook и интегрированной среды разработки.

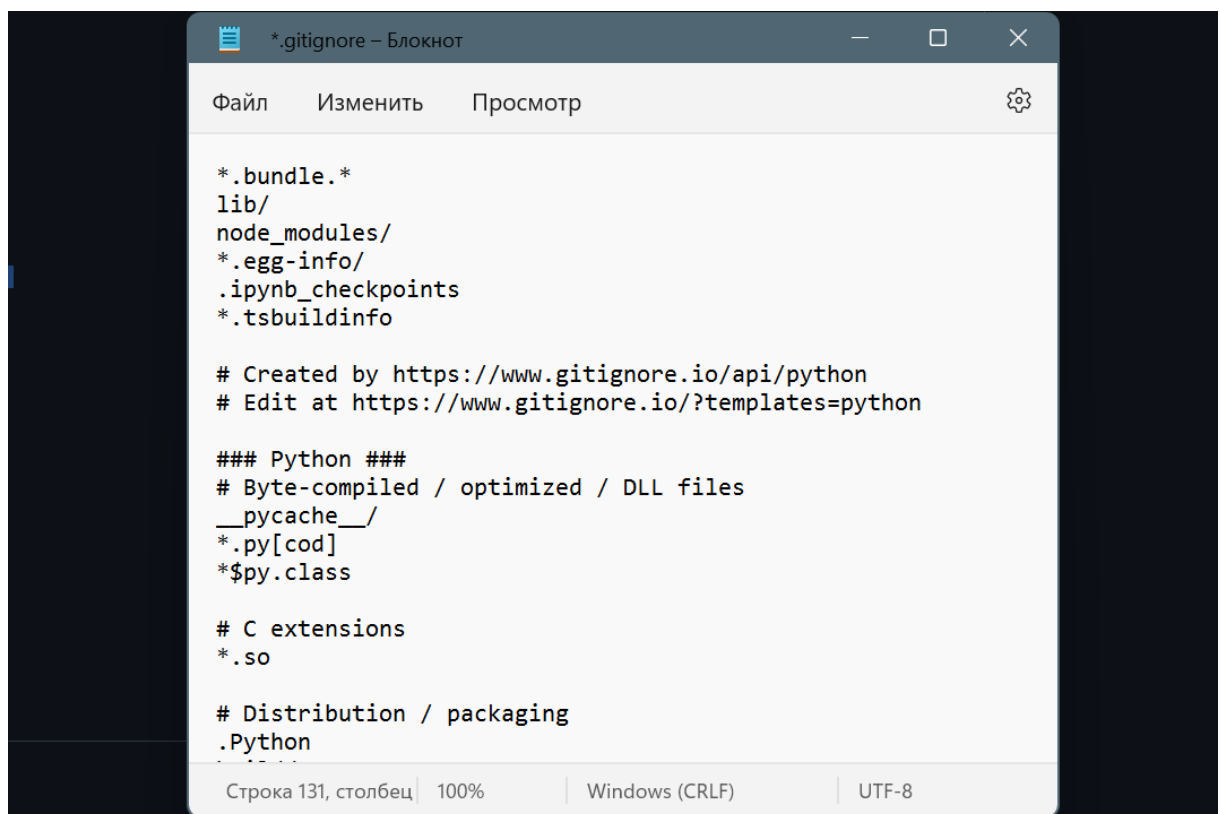


Рисунок 4 – Дополнение файла .gitignore

## 5. Проработать примеры лабораторной работы.

```
In [2]: 2+3
Out[2]: 5

In [4]: a = 5
        b = 7
        print(a+b)
12

In [6]: n = 7
        for i in range(n):
            print(i*10)
0
10
20
30
40
50
60

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

Рисунок 5 – Проработка примеров

```
In [9]: from matplotlib import pylab as plt
        %matplotlib inline

In [10]: x = [i for i in range(50)]
         y = [i**2 for i in range(50)]
         plt.plot(x, y)

Out[10]: [<matplotlib.lines.Line2D at 0x1f3a0fe1340>]
```

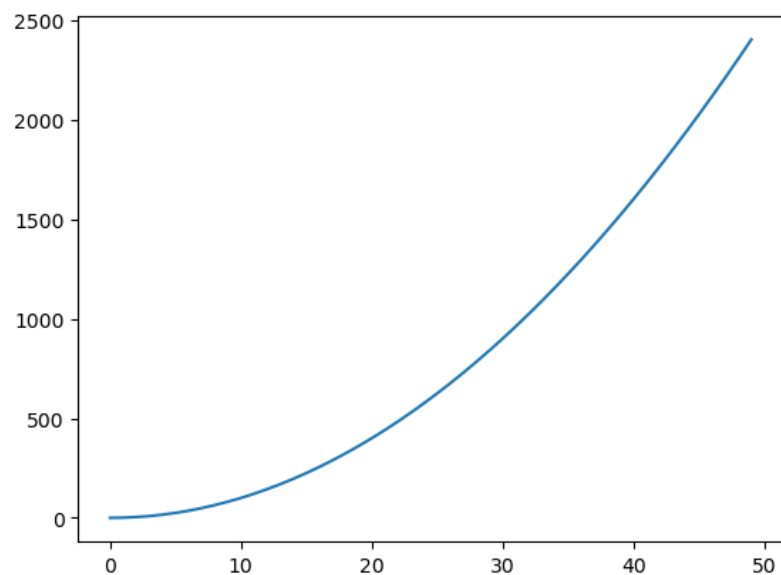


Рисунок 6 – Проработка примеров (продолжение)

```
In [12]: %lsmagic
```

```
Out[12]: Available line magics:
%alias %alias_magic %autoawait %autocall %automagic %autosave %bookmark
%cd %clear %cls %colors %conda %config %connect_info %copy %ddir %debug
%dhist %dirs %doctest_mode %echo %ed %edit %env %gui %hist %history
%killbgscripts %ldir %less %load %load_ext %loadpy %logoff %logon %logs
tart %logstate %logstop %ls %lsmagic %macro %magic %matplotlib %mkdir
%more %notebook %page %pastebin %pdb %pdef %pdoc %pfile %pinfo %pinfo2
%pip %popd %pprint %precision %prun %psearch %psource %pushd %pwd %pyc
at %pylab %qtconsole %quickref %recall %rehashx %reload_ext %ren %rep
%rerun %reset %reset_selective %rmdir %run %save %sc %set_env %store %
sx %system %tb %time %timeit %unalias %unload_ext %who %who_ls %whos
%xdel %xmode
```

Available cell magics:

```
%%! %%HTML %%SVG %%bash %%capture %%cmd %%debug %%file %%html %%javasc
ript %%js %%latex %%markdown %%perl %%prun %%pypy %%python %%python2 %
%python3 %%ruby %script %%sh %%svg %%sx %%system %%time %%timeit %%wr
itefile
```

Automagic is ON, % prefix IS NOT needed for line magics.

```
In [13]: %env TEST = 5
```

```
env: TEST=5
```

Рисунок 7 – Проработка примеров (продолжение)

```
In [15]: %%time
import time
for i in range(50):
    time.sleep(0.1)
```

```
Wall time: 5.44 s
```

```
In [16]: %timeit x = [(i**10) for i in range(10)]
```

```
2.29 µs ± 23.7 ns per loop (mean ± std. dev. of 7 runs, 100000 loops each)
```

```
In [ ]:
```

Рисунок 8 – Проработка примеров (продолжение)

6. Решить задания в ноутбуках, выполненных преподавателем.

**Задание:**

1. Определите число `ticket_number` — шестизначный номер билета;
2. Напишите код, который по шестизначному номеру `ticket_number` билетика проверяет, является ли он счастливым;
3. Если номер счастливый, выведите строку `Yes`, иначе — `No`.

**Пример 1:**

*Input:* 123456

*Output:* No

**Пример 2:**

*Input:* 123042

*Output:* Yes

```
In [1]: ticket_number = int(input("Enter the number on your ticket:"))
        ahead_3 = ticket_number // 1000
        behind_3 = ticket_number % 1000
        if (ahead_3 // 100 + ahead_3 // 10 % 10 + ahead_3 % 10 == behind_3 // 100 + behind_3 // 10 % 10 + behind_3 % 10):
            print("Yes")
        else:
            print("No")
```

Enter the number on your ticket:123321  
Yes

## Рисунок 9 – Решение 1 задания

**Задание:**

1. Определите строку `password` — придуманный вами пароль;
2. Напишите код, который по паролю `password` проверяет, является ли он надёжным;
3. Если пароль надёжный, выведите строку `strong`, иначе — `weak`.

Пусть имя пользователя -- Андрей.

**Пример 1:**

*Input:* Aandrei123

*Output:* weak

**Пример 2:**

*Input:* an12dRei

*Output:* strong

```
In [2]: password = input("Enter your password: ")
        name = input("Enter your name: ")
        if (password == password.upper() or password == password.lower() or password.isalpha()
            or len(set(password)) < 4 or name.lower() in password.lower()):
            print("your password is weak")
        else:
            print("your password is strong")
```

Enter your password: ament24  
Enter your name: min  
the password is weak

## Рисунок 10 – Решение 2 задания

#### Задание:

1. Определите число `amount` — количество чисел Фибоначчи, которые надо вывести;
2. Напишите код, который выводит первые `amount` чисел Фибоначчи.

#### Пример 1:

Input: 3

Output: 1 1 2

#### Пример 2:

Input: 10

Output: 1 1 2 3 5 8 13 21 34 55

```
In [3]: amount = int(input("Enter a digit: "))
a, b = 0, 1
print(b)
for i in range(amount):
    sum = a + b
    a = b
    b = sum
    print(b)
```

Enter a digit: 3

1  
1  
2  
3

## Рисунок 11 – Решение 3 задания

### Время исследований

На сайте <https://www.kaggle.com/> выберите любой набор данных в формате CSV и проведите для него маленькое исследование: загрузите данные из набора с использованием стандартного модуля `csv`, посмотрите средние значения и стандартные отклонения двух выбранных числовых атрибутов, найдите методом наименьших квадратов уравнение линейной зависимости, связывающей один числовой атрибут с другим. Для оценки заданной зависимости найдите коэффициент парной корреляции, сделайте соответствующие выводы.

Результаты надо обязательно прокомментировать и пояснить!

#### Пример 1:

Пусть таблица `bikes.csv` содержит данные по арендам велосипедов за 2 года:

- `datetime`: дата и время аренды
- `season`: время года
- `temp`: температура воздуха по Цельсию
- `windspeed`: скорость ветра
- `registered`: число аренд

Одно из направлений исследования могло бы заключаться в проверке зависимости суммарного числа аренд от температуры воздуха.

Импорт библиотек:

Чтение файла с данными и вывод столбцов:

```
In [14]: import csv
from math import sqrt

with open('train.csv', encoding = 'utf-8') as csvfile:
    f = csv.reader(csvfile, delimiter=',')
    hgt = []
    wdth = []
    for element in f:
        if element[0] == "battery_power":
            continue
        hgt.append(int(element[11]))
        wdth.append(int(element[12]))
```

Подсчет среднего значения в 2-х столбцах

## Рисунок 12 – Решение 4 задания

```
In [16]: v1 = sum((el-hh)**2 for el in hgt) / len(hgt)
st_v = sqrt(v1)
v2 = sum((elem-ww)**2 for elem in wdth) / len(wdth)
st_b = sqrt(v2)
print(sum((el-hh)**2 for el in hgt))
print(len(hgt))
print(f"Стандартное отклонение 1: {st_v}")
print(f"Стандартное отклонение 2: {st_b}")

393685874.6720002
2000
Стандартное отклонение 1: 443.6698517321186
Стандартное отклонение 2: 432.09138357499097
```

```
In [17]: import numpy as np
import pandas as pd
data = pd.read_csv('summer-products-with-rating-and-performance_2020-08.csv')
data.std()
```

C:\Users\vyaza\AppData\Local\Temp\ipykernel\_29724\3721584870.py:4: FutureWarning: Dropping of nuisance columns in DataFrame reductions (with 'numeric\_only=None') is deprecated; in a future version this will raise TypeError. Select only valid columns before calling the reduction.

```
data.std()
```

```
Out[17]: index                454.230301
price                3.932030
retail_price         30.357863
units_sold           9356.539302
uses_ad_boosts        0.495639
rating                0.515374
rating_count         1983.928834
rating_five_count     980.203270
rating_four_count     400.516231
rating_three_count    311.690656
rating_two_count      151.343933
rating_one_count      214.075544
badges_count          0.340709
badge_local_product   0.134565
badge_product_quality 0.262472
badge_fast_shipping   0.112075
product_variation_inventory 21.353137
shipping_option_price  1.024371
shipping_is_express    0.050379
countries_shipped_to  20.301203
inventory_total        2.562799
has_urgency_banner     0.000000
merchant_rating_count 78474.455607
merchant_rating        0.204768
merchant_has_profile_picture 0.350871
dtype: float64
```

Рисунок 13 – Решение 4 задания (продолжение)

```
In [19]: sum_ab = 0
sum_square = 0

for i, el in enumerate(hgt):
    sum_ab += el * wdth[i]
    sum_square += el**2

size = len(hgt)
k_lin = (size * sum_ab - sum(hgt) * sum(wdth)) / (size * sum_square - sum(hgt)**2)
b_lin = hh - ww * k_lin
func_val = []

for el in hgt:
    func_val.append(k_lin * el + b_lin)

print(f"Уравнение линейной зависимости: y = {k_lin}x + {b_lin}")

Уравнение линейной зависимости: y = 0.4973375913350377x + 22.682295711534607
```

```
In [20]: cor_chisl = 0
for i, el in enumerate(hgt):
    cor_chisl += (el - hh) * (wdth[i] - ww)
sqr_diff_vol = sum((el-hh)**2 for el in hgt)
sqr_diff_bag = sum((el-ww)**2 for el in wdth)
r_xy = cor_chisl / sqrt(sqr_diff_vol * sqr_diff_bag)
print(f"Коэффициент парной корреляции: {r_xy}")

Коэффициент парной корреляции: 0.5106644191393129
```

```
In [ ]:
```

Рисунок 14 – Решение 4 задания (продолжение)

7. Создать ноутбук, в котором выполнить решение вычислительной задачи (например, задачи из области физики, экономики, математики, статистики и т. д.), условие которой предварительно необходимо согласовать с преподавателем.



### Решение СЛАУ

$$\begin{cases} x_1 - 2x_2 + 3x_3 = 2 \\ 3x_1 + x_2 - x_3 = 3 \\ 2x_1 + 5x_2 + 2x_3 = 9 \end{cases}$$

```
In [28]: import scipy

In [29]: import numpy as np

In [30]: from scipy.linalg import solve

In [31]: A = np.array(
[
    [1, -2, 3],
    [3, 1, -1],
    [2, 5, 2],
]
)

In [32]: b = np.array([2, 3, 9]).reshape((3, 1))

In [33]: x = solve(A, b)

In [34]: x
Out[34]: array([[1.],
               [1.],
               [1.]])
```

Рисунок 15 – Работа программы

### Контрольные вопросы

1. Как осуществляется запуск Jupyter notebook?

Jupyter Notebook входит в состав Anaconda. Для запуска Jupyter Notebook перейдите в папку Scripts (она находится внутри каталога, в котором установлена Anaconda) и в командной строке наберите: «ipython notebook»

В результате будет запущена оболочка в браузере

2. Какие существуют типы ячеек в Jupyter notebook?

Ячейки в блокноте Jupyter бывают четырех типов – Code, Markdown и Raw и Headings.

Содержимое в ячейке Code обрабатывается как инструкции на языке программирования, по умолчанию используется Python.

Ячейки Markdown содержат текст, отформатированный с

использованием языка markdown. Доступны все виды функций форматирования, такие как выделение текста жирным шрифтом и курсивом, отображение упорядоченного или неупорядоченного списка, отображение табличного содержимого и т.д.

Содержимое Raw ячейки не оценивается ядром notebook.

Headings-ячейка может использоваться для разбивки блокнота на разделы.

3. Как осуществляется работа с ячейками в Jupyter notebook?

4. Что такое "магические" команды Jupyter notebook? Какие "магические" команды Вы знаете?

5. Самостоятельно изучите работу с Jupyter notebook и IDE PyCharm и Visual Studio Code. Приведите основные этапы работы с Jupyter notebook в IDE PyCharm и Visual Studio Code.