

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ
ФЕДЕРАЦИИ**

**Федеральное государственное автономное
образовательное учреждение высшего
образования**

«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Кафедра инфокоммуникаций

«Работа с IPython и Jupyter Notebook»

Отчет по лабораторной работе № 3.1

по дисциплине «Технологии распознавания образов»

Выполнил студент группы ПИЖ-б-о-21-1

Коновалова Виктория Николаевна

« » 2023г.

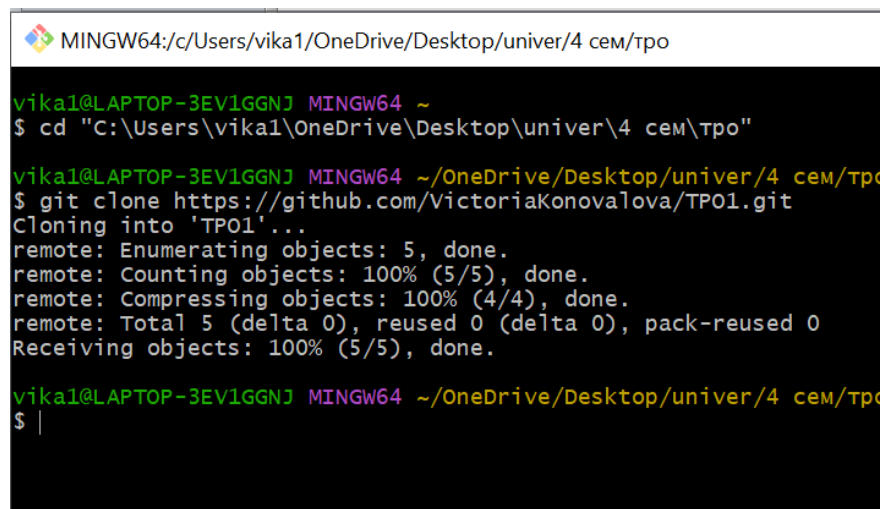
Подпись студента _____

Работа защищена « » _____ 20__ г.

Проверил Воронкин Р.А. _____
(подпись)

Ставрополь 2023

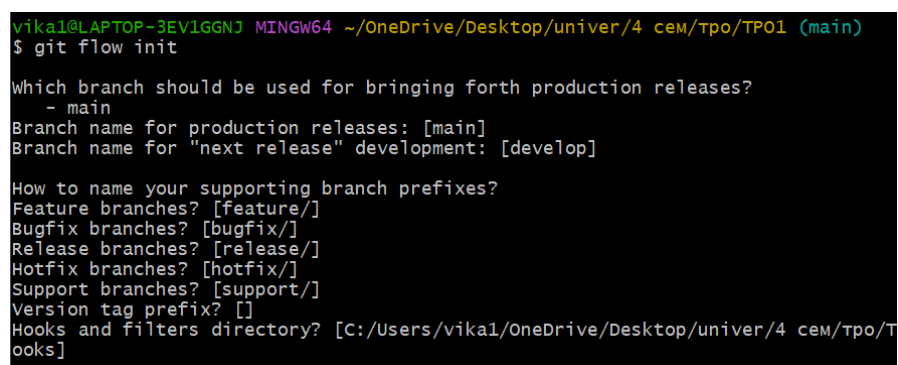
1. Изучить теоретический материал работы.
2. Создать общедоступный репозиторий на GitHub, в котором будет использована лицензия MIT и выбранный Вами язык программирования (выбор языка программирования будет доступен после установки флажка Add .gitignore).
3. Выполните клонирование созданного репозитория на рабочий компьютер.



```
MINGW64:/c:/Users/vika1/OneDrive/Desktop/univer/4 сем/тпо
vika1@LAPTOP-3EV1GGNJ MINGW64 ~
$ cd "C:\Users\vika1\OneDrive\Desktop\univer\4 сем\тпо"
vika1@LAPTOP-3EV1GGNJ MINGW64 ~/OneDrive/Desktop/univer/4 сем/тпо
$ git clone https://github.com/VictoriaKonovalova/TP01.git
Cloning into 'TP01'...
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
Receiving objects: 100% (5/5), done.
vika1@LAPTOP-3EV1GGNJ MINGW64 ~/OneDrive/Desktop/univer/4 сем/тпо
$ |
```

Рисунок 1- Клонирование репозитория

4. Организуйте свой репозиторий в соответствии с моделью ветвления git-flow.



```
vika1@LAPTOP-3EV1GGNJ MINGW64 ~/OneDrive/Desktop/univer/4 сем/тпо/TP01 (main)
$ 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:/Users/vika1/OneDrive/Desktop/univer/4 сем/тпо/TP01/hooks]
```

Рисунок 2 – Организация репозитория в соответствии с моделью ветвления git-flow

5. Дополните файл .gitignore необходимыми правилами для выбранного языка программирования, интерактивной оболочки Jupyter notebook и интегрированной среды разработки.

```

vika1@LAPTOP-3EV1GGNJ MINGW64 ~/OneDrive/Desktop/univer/4 сем/тпо/ТПО1 (develop)
$ git add .

vika1@LAPTOP-3EV1GGNJ MINGW64 ~/OneDrive/Desktop/univer/4 сем/тпо/ТПО1 (develop)
$ git status
On branch develop
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        modified:   .gitignore

```

Рисунок 2 – Дополнение файлф .gitignore

6. Проработать примеры лабораторной работы.

0	Name	Last Modified	File size
<input type="checkbox"/>	VirtualBox VMs	2 месяца назад	
<input type="checkbox"/>	Videos	2 года назад	
<input type="checkbox"/>	Untitled Folder 1	14 дней назад	
<input type="checkbox"/>	source	год назад	
<input type="checkbox"/>	Searches	2 года назад	
<input type="checkbox"/>	Saved Games	2 года назад	
<input type="checkbox"/>	PycharmProjects	2 месяца назад	
<input type="checkbox"/>	PyCharm Community Edition 2020.2.3	2 года назад	
<input type="checkbox"/>	PersonalData	год назад	
<input type="checkbox"/>	opi_1	2 дня назад	
<input type="checkbox"/>	OneDrive	2 месяца назад	
<input type="checkbox"/>	notebooks	минуту назад	

Рисунок 3 – Пример установки и запуска ноутбука

```

Ввод [1]: 2+3
Out[1]: 5

Ввод [2]: a=5
          b=7
          print(a+b)
          12

Ввод [3]: n=7
          for i in range(n):
              print(i*10)
          0
          10
          20
          30
          40
          50
          60

Ввод [4]: i=0
          while True:
              i+=1
              if i>5:
                  break
              print("Test while")
          Test while
          Test while
          Test while
          Test while
          Test while

```

Рисунок 4 – Работа с простыми примерами Jupyter Notebook

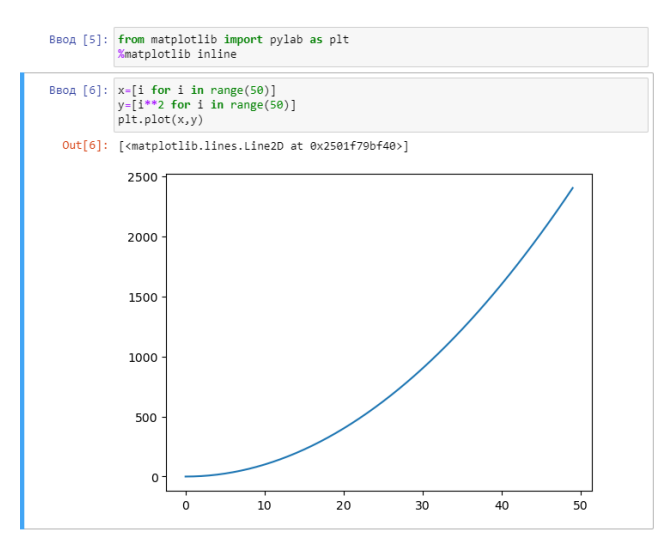


Рисунок 5 – Рисование графиков

```

In [15]: %lsmagic

Out[15]: Available line magics:
%alias %alias_magic %autoawait %autocall %automagic %autosave
%bookmark %cd %clear %cls %colors %conda %config %connect_inf
o %copy %ddir %debug %dhist %dirs %doctest_mode %echo %ed %
edit %env %gui %hist %history %killbgscripts %ldir %less %lo
ad %load_ext %loadpy %logoff %logon %logstart %logstate %logs
top %ls %lsmagic %macro %magic %matplotlib %mkdir %more %not
ebook %page %pastebin %pdb %pdef %pdoc %pfile %pinfo %pinfo2
%pip %popd %pprint %precision %prun %psearch %psource %pushd
%pwd %pycat %pylab %qtconsole %quickref %recall %rehashx %rel
oad_ext %ren %rep %rerun %reset %reset_selective %rmdir %run
%save %sc %set_env %store %sx %system %tb %time %timeit %un
alias %unload_ext %who %who_ls %whos %xdel %xmode

Available cell magics:
%%! %%HTML %%SVG %%bash %%capture %%cmd %%debug %%file %%htm
l %%javascript %%js %%latex %%markdown %%perl %%prun %%pypy
%%python %%python2 %%python3 %%ruby %%script %%sh %%svg %%sx
%%system %%time %%timeit %%writefile

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

```

Рисунок 6 – Пример использования %ismagic

```

In [6]: %env TEST = 5

env: TEST=5

```

Рисунок 7 – Пример использования %env

```

In [9]: %%time
import time
for i in range(50):
    time.sleep(0.1)

Wall time: 5.49 s

In [10]: %timeit X = [(i**10) for i in range(10)]

7.08 µs ± 419 ns per loop (mean ± std. dev. of 7 runs, 100000 loops
each)

```

Рисунок 8 – Пример использования %%timetimeit

7. Решить задания в ноутбуках, выполненных преподавателем.

Задание 1.

Билет считается счастливым, если выполнено следующее условие: сумма первых трёх цифр номера равна сумме последних трёх цифр.

Задание:

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

```
Ввод [1]: ticket_number = int(input("Введите шестизначный номер: "))
```

Введите шестизначный номер: 321231

```
Ввод [7]: s1=ticket_number//100000+ticket_number//10000%10+ticket_number//1000%10
s2=ticket_number%1000//100+ticket_number%100//10+ticket_number%10
if s1==s2:
    print("yes")
else:
    print("no")
```

yes

Рисунок 10 – Решение примера

Задание 2.

Задание:

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

```
Ввод [9]: password = input("Введите шестизначный пароль: ")
name=input("Введите своё имя: ")
```

Введите шестизначный пароль: vika./1
Введите своё имя: vika

```
Ввод [11]: if password.isalpha() or password.isdigit():
            print("weak")
            exit(-1)
            if password.islower() or password.issupper():
                print("weak")
                exit(-1)
```

weak

```
Ввод [14]: unic=set(password)
            if len(unic)<4:
                print("weak")
                exit(-1)
```

```
Ввод [16]: if name.lower() in password.lower():
            print("weak")
            exit(-1)
            else:
                print("strong")
```

weak

Рисунок 11 – Решение примера

Задание 3.

Числа Фибоначчи

Как известно, [числа Фибоначчи](#) — это последовательность чисел, каждое из которых равно сумме двух предыдущих (первые два числа равны 1):

1, 1, 2, 3, 5, 8, 13, ...

Задание:

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

```
Ввод [18]: amount = int(input("Введите кол-во чисел Фибоначчи: "))
```

Введите кол-во чисел Фибоначчи: 8

```
Ввод [19]: a,b=0,1
for i in range(amount):
    sum=a+b
    a=b
    b=sum
    print(sum)
```

1
2
3
5
8
13
21
34

Рисунок 12 – Решение примера

Задача 4.

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.

О данных:

Список содержит лучшие американские колледжи 2022 года и подробную информацию о них, включая финансовую помощь, количество студентов, номер телефона колледжа, веб-сайт и т. д.

```
Ввод [1]: import csv
from math import sqrt
```

```
Ввод [2]: with open('top_colleges_2022.csv', 'r', newline='') as f:
    data=csv.reader(f, delimiter=',')
    faculty_ratio=[]
    students=[]
    for row in data:
        if row[3] == "NY":
            faculty_ratio.append(float(row[19]))
            students.append(float(row[4]))
```

```
Ввод [3]: sr_faculty_ratio = sum(faculty_ratio) / len(faculty_ratio)
sr_students = sum(students) / len(students)
print(f"Среднее значение соотношения преподавателей к студентам: {sr_faculty_ratio:.3f}")
print(f"Среднее значение количества студентов: {sr_students:.3f}")
```

Среднее значение соотношения преподавателей к студентам: 13.000
Среднее значение количества студентов: 11921.254

Рисунок 13 – Решение примера

```

Ввод [4]: std1 = sqrt(sum((elem-sr_faculty_ratio)**2 for elem in faculty_ratio) / len(faculty_ratio))
std2 = sqrt(sum((elem-sr_students)**2 for elem in students) / len(students))
print("Стандартное отклонение преподавателей к студентам: ")
f"{std1:.2f}")
print("Стандартное отклонение количества студентов: ")
f"{std2:.2f}")

Стандартное отклонение преподавателей к студентам: 3.81
Стандартное отклонение количества студентов: 11241.75

Ввод [5]: sum_ab = 0
sum_square = 0
for idx, elem in enumerate(faculty_ratio):
    sum_ab += elem * students[idx]
    sum_square += elem**2
k_lin = (len(faculty_ratio) * sum_ab - sum(faculty_ratio) * sum(students)) / (len(faculty_ratio) * sum_square - sum(faculty_ratio)**2)
b_lin = sr_students - sr_faculty_ratio * k_lin
func_val = []
for elem in faculty_ratio:
    func_val.append(k_lin * elem + b_lin)
print(f"Уравнение линейной зависимости: y = {k_lin}x + {b_lin}")
print("Значения функции на кривой методом наименьших квадратов: ")
for val in func_val:
    print(val)

Уравнение линейной зависимости: y = 725.8100436681223x + 2485.7234005683786
Значения функции на кривой методом наименьших квадратов:
6840.583662577112
9018.013793581478
9018.013793581478
9018.013793581478
8292.203749913357
16276.114230262701
17001.924273930825
9018.013793581478
8292.203749913357
16276.114230262701
12647.06401192209
9743.823837249602
13372.874055590213
8292.203749913357
12277.874055590213

```

Рисунок 14 – Решение примера

```

Ввод [6]: razn = 0
for idx, elem in enumerate(faculty_ratio):
    razn += (elem-sr_faculty_ratio)*(students[idx] - sr_students)
razn_sr_faculty_ratio = sum((elem-sr_faculty_ratio)**2 for elem in faculty_ratio)
razn_sr_students = sum((elem-sr_students)**2 for elem in students)
mark_kor = razn / sqrt(razn_sr_faculty_ratio*razn_sr_students)
print(f"Коэффициент парной корреляции: {mark_kor}")

Коэффициент парной корреляции: 0.2461877136685876

```

Рисунок 15 – Решение примера

8. 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.

Система линейных уравнений

```

Ввод [1]: import sympy as sp
from sympy.solvers.solveset import linsolve
%matplotlib inline
sp.init_printing()

Ввод [2]: x,y=sp.symbols("x y")

eq1=sp.Eq(x+2*y,11)
eq2=sp.Eq(5*x-3*y,3)

display(eq1)
display(eq2)

x + 2y = 11
5x - 3y = 3

Ввод [3]: eqs=[eq1,eq2]
res=linsolve(eqs,(x,y))
print(f"X: {res.args[0][0]}")
print(f"Y: {res.args[0][1]}")

X: 3
Y: 4

```

Рисунок 15 – Решение примера

9. Зафиксируйте сделанные изменения в репозитории.
10. Выполните слияние ветки для разработки с веткой main (master).
11. Отправьте сделанные изменения на сервер GitHub.