

**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

Гасанов Г. М. « » 2023 г.

Подпись студента _____

Работа защищена « » _____ 2023 г.

Проверил Воронкин Р.А. _____

(подпись)

Ставрополь 2023

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

Выполнение работы:

1. Изучить теоретический материал работы.
2. Создать общедоступный репозиторий на GitHub, в котором будет использована лицензия IT и язык программирования Python.
3. Выполните клонирование созданного репозитория.
4. Организуйте свой репозиторий в соответствие с моделью ветвления git-flow.
5. Дополните файл .gitignore необходимыми правилами для выбранного языка программирования, интерактивной оболочки Jupyter notebook и интегрированной среды разработки.
6. Проработать примеры лабораторной работы.

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

In [3]: a = 5
        b = 7
        print(a + b)
12

In [4]: n = 7
        for i in range(n):
            print(i*10)
0
10
20
30
40
50
60

In [5]: 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 – Примеры из лабораторной работы

7. Решить задания в ноутбуках, выполненных преподавателем.

```
In [ ]: ticket_number = # any 6-digit number
```

```
In [12]: ticket_number = list(map(int, list(input())))  
123456
```

```
In [13]: if sum(ticket_number[:3]) == sum(ticket_number[3:]):  
         print("Yes")  
         else:  
         print("No")  
No
```

Рисунок 5 – Решение задания 1

```
In [ ]: password = # any password
```

```
In [3]: password = input('Enter password: ')  
name = input('Enter your name: ')  
Enter password: an12dRei  
Enter your name: andrei
```

```
In [5]: if password.isalpha() or password.isdigit():  
         print("weak")  
         exit(-1)  
         if password.islower() or password.isupper():  
         print("weak")  
         exit(-1)
```

```
In [9]: un_symb = set(password)  
if len(un_symb) < 4:  
    print('weak')  
    exit(-1)
```

```
In [10]: if name.lower() in password.lower():  
         print("weak")  
         exit(-1)  
         else:  
         print("strong")  
strong
```

Рисунок 6 – Решение задания 2

```
In [ ]: amount = # any number
```

```
In [15]: amount = int(input()) - 1
```

10

```
In [18]: a, b = 0, 1
for i in range(amount):
    a, b = b, a + b
    print(b)
```

1
2
3
5
8
13
21
34
55

Рисунок 7 – Решение задания 3

```
In [3]: import csv
from math import sqrt

with open('file.csv', 'r', newline='') as csvfile:
    data = csv.reader(csvfile, delimiter=';')
    high = []
    low = []
    next(data)
    for row in data:
        high.append(float(row[2]))
        low.append(float(row[3]))
```

```
In [4]: high_sred = sum(map(int, high)) / len(high)
low_sred = sum(map(int, low)) / len(low)
print('Среднее значение high: ', high_sred)
print('Среднее значение low: ', low_sred)
```

Среднее значение high: 1190.389590702375

Среднее значение low: 1113.521475492673

```
In [6]: v1 = sqrt(sum((elem - high_sred) ** 2 for elem in high) / len(high))
v2 = sqrt(sum((elem - low_sred) ** 2 for elem in low) / len(low))
print("Стандартное отклонение high: ", v1)
print("Стандартное отклонение low: ", v2)
```

Стандартное отклонение high: 1198.3325262366664

Стандартное отклонение low: 1122.629118971339

In [7]:

```
sum_ab = 0
sum_square = 0
for idx, elem in enumerate(high):
    sum_ab += elem * low[idx]
    sum_square += elem ** 2
size = len(high)
k_lin = (size * sum_ab - sum(high) * sum(low)) / (size * sum_square - sum(high) ** 2)
b_lin = low_sred - high_sred * k_lin
func_val = []
for elem in high:
    func_val.append(k_lin * elem + b_lin)
print(f"Уравнение линейной зависимости: y = {k_lin}x + {b_lin}")
print("Значение функции на кривой методом наименьших квадратов: ")
for val in func_val:
    print(val)
```

Уравнение линейной зависимости: y = 0.9350009231387384x + 0.5061092912073946

Значение функции на кривой методом наименьших квадратов:

```
304.11772689596023
299.1949619956496
298.914472938719
307.57444587882446
318.57192019177586
319.259131845269
314.81508180759977
313.6977482244416
1777.1033331102007
1808.563459990067
```

In [9]:

```
cor_chisl = 0
for idx, elem in enumerate(high):
    cor_chisl += (elem - high_sred) * (low[idx] - low_sred)
sqr_diff_high = sum((elem - high_sred) ** 2 for elem in high)
sqr_diff_low = sum((elem - low_sred) ** 2 for elem in low)
r_xy = cor_chisl / sqrt(sqr_diff_high * sqr_diff_low)
print(f"Коэффициент парной корреляции: {r_xy}")
```

Коэффициент парной корреляции: 0.9980518171048023

Рисунок 8 - Решение задания 4

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.

Задача

В июле 2016 года планируется взять кредит в банке на пять лет в размере S тыс. рублей. Условия его возврата таковы: – каждый январь долг возрастает на 25% по сравнению с концом предыдущего года; – с февраля по июнь каждого года необходимо выплатить часть долга; – в июле 2017, 2018 и 2019 долг остаётся равным S тыс. рублей; – выплаты в 2020 и 2021 годах равны по 625 тыс. рублей; – к июлю 2021 долг будет выплачен полностью. Найдите общую сумму выплат за пять лет

```
В [17]: S = 1000 # сумма кредита в тыс. руб.
        = i ((1 + 0.25)**(1/12)) - 1 # ежемесячная процентная ставка
        n = 60 # количество месяцев выплаты кредита

В [18]: # Вычисляем аннуитетный платеж для каждого года
        = платежи []
        ассортимент в году на(2016, 2022):
        в году если [2017, 2018, 2019]:
            платежи.добавить(ы)
        в году элиф [2020, 2021]:
            платежи.добавить(625)
        ещё:
            = m (year - 2016) * 12 # количество месяцев прошло с начала выплат
            оставшиеся_месяцы = n - m
            = a (S * i) / (1 - (1 + i)**(-remaining_months)) # аннуитетный платеж
            платежи.добавить(a)

В [19]: # Вычисляем общую сумму выплат за пять лет
        сумма = общая сумма платежей(платежи)

        print("Общая сумма выплат за пять лет составляет", total_payments, "тыс. руб.")
```

Общая сумма выплат за пять лет составляет 4277.917160164068 тыс. руб.

Рисунок 8 - Решение экономической задачи

9. Зафиксируйте сделанные изменения в репозитории.

Вопросы для защиты работы:

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?

Для запуска ячейки используете команды из меню Cell, либо следующие сочетания клавиш: Ctrl+Enter – выполнить содержимое ячейки. Shift+Enter – выполнить содержимое ячейки и перейти на ячейку ниже. Alt+Enter – выполнить содержимое ячейки и вставить новую ячейку ниже.

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

Magic—это отличные команды, упрощающие нашу жизнь при решении определенных задач. Часто похожи на команды Unix, но реализованы на Python. Магических команд в Python великое множество!

Существует 2 типа магических команд: строчные (применяются к одной строке) и ячейечные (применимы ко всей ячейке). Строчные команды начинаются с символа %, а ячейечные—с двух %%. Просмотреть все доступные команды можно через: `%lsmagic`

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.