МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ
РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»
ИНСТИТУТ ЦИФРОВОГО РАЗВИТИЯ

Отчет о лабораторной работе № 1 по дисциплине «Технологии распознавания образов»

Выполнил студент
2 курса, группы ПИЖ-б-о-20-1
Тотубалина С.С.
Проверил:
Доцент кафедры инфокоммуникаций,
Воронкин Р.А.

Ставрополь, 2022 г.

1. Отработка примеров работы с Jupyter Notebook

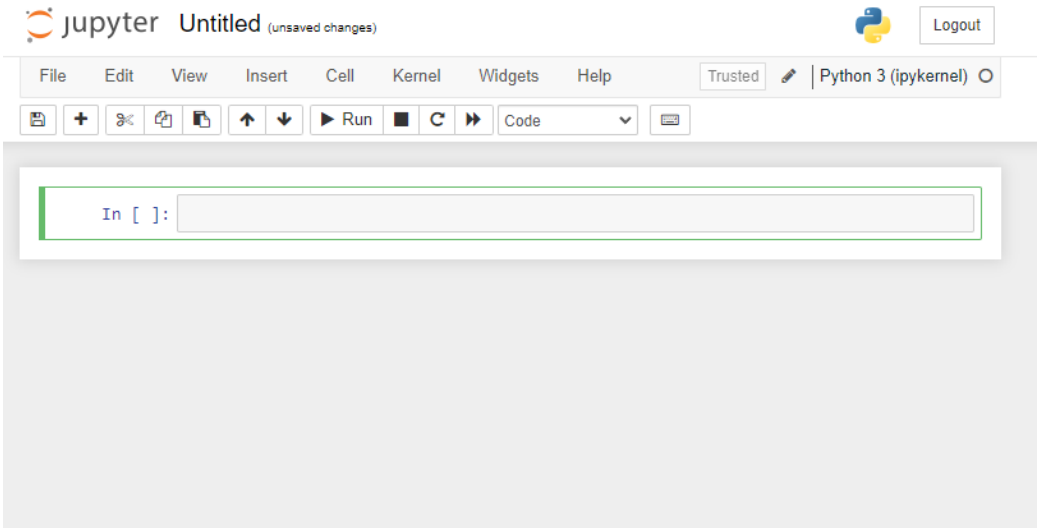


Рис. 1 – результат создания ноутбука

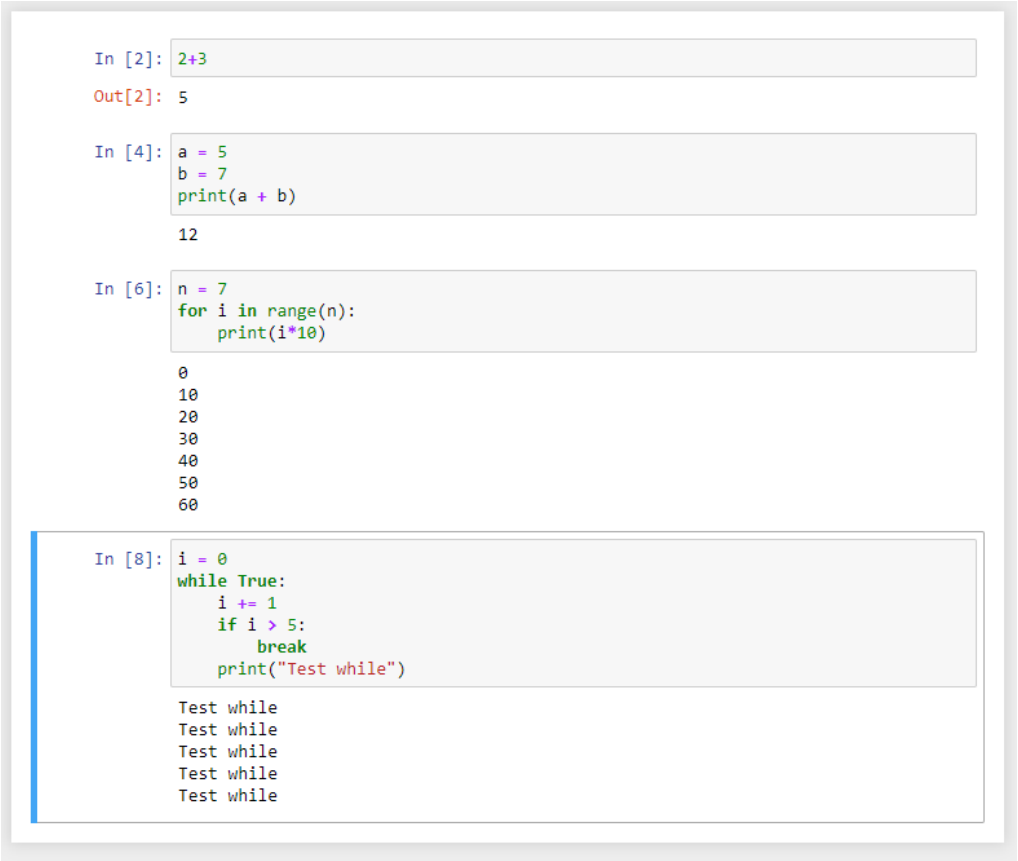


Рис. 2 – проработка примеров кода

2. Вывод изображений в ноутбуке

```
In [1]: from matplotlib import pylab as plt
        %matplotlib inline

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

Out[2]: [<matplotlib.lines.Line2D at 0x212455aee20>]
```

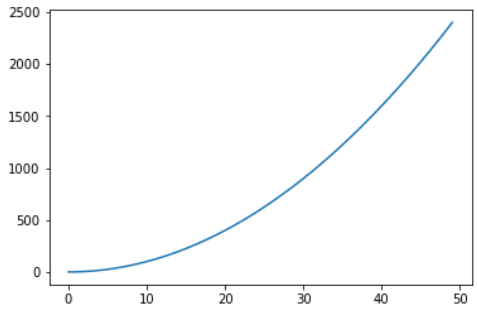


Рис. 3 – вывод графика

3. Магия – дополнительные команды, выполняемые в рамках оболочки, которые облегчают процесс разработки и расширяют возможности для программирования.

```
In [1]: %lsmagic

Out[1]: Available line magics:
        %alias  %alias_magic  %autoawait  %autocall  %automagic  %autosave  %bookmark
        %cd  %clear  %cls  %colors  %conda  %config  %connect_info  %copy  %ddir  %debu
        g  %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 [5]: %env TEST = 5

        env: TEST=5

In [7]: %%time
        import time
        for i in range(50):
            time.sleep(0.1)

        Wall time: 5.46 s

In [8]: %timeit x = [(i**10) for i in range(10)]

        4.26 µs ± 208 ns per loop (mean ± std. dev. of 7 runs, 100000 loops each)
```

Рис. 4 – проработка дополнительных команд (магия)

4. Выполнение заданий в ноутбуках.

```
In [4]: ticket_number = int(input("Enter your ticket"))
        Enter your ticket888888

In [7]: sum1 = ticket_number // 100000 + ticket_number // 10000 % 10 + ticket_number //

In [8]: print(sum1)
        24

In [9]: 2 = ticket_number % 1000 // 100 + ticket_number % 100 // 10 + ticket_number % 10

In [10]: print(sum2)
         24

In [11]: if sum1 == sum2:
             print("Yes")
         else:
             print("No")
         Yes
```

Рис. 5 – решение задачи «Счастливый билетик»

```
In [12]: password = input("Enter your pasword")
         name = input("What is your name?")
         Enter your paswordAandrei123
         What is your name?Aandrei

In [14]: if password.isalpha() or password.isdigit():
             print("weak")
             exit(-1)
         if password.islower() or password.isupper():
             print("weak")
             exit(-1)

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

In [15]: if name.lower() in password.lower():
             print("weak")
             exit(-1)
         else:
             print("strong")
         weak
```

Рис. 6 – решение задачи «Пароль»

```
In [20]: 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
         55
         89
```

Рис. 7 – решение задачи «Числа Фибоначчи»

Для анализа выбран dataset, в котором представлены данные о проценте образованных мужчин и женщин в топ-10 стран мира, в строках кода ниже представлено открытие файла .csv и считывание данных с него. После прохождения по файлу формируются списки прокомментированные ниже.

In [2]:
```
import pandas as pd
import seaborn as sns
import numpy as np
import matplotlib.pyplot as plt
import math

from sklearn.linear_model import LinearRegression
```

In [34]:
```
df = pd.read_csv('Lvl_development.csv')
```

In [35]:
```
df
```

Out[35]:

	Country or Area	Year	Male	Female	Unit
0	Bangladesh	1991	44.3	25.8	Percent
1	Bangladesh	2001	53.9	40.8	Percent
2	Brazil	2000	86.2	86.5	Percent
3	Brazil	2004	88.4	88.8	Percent
4	China	1990	87.0	68.1	Percent
5	China	2000	95.1	86.5	Percent
6	Egypt	1986	57.0	31.4	Percent
7	Egypt	1996	67.2	43.6	Percent
8	Egypt	2005	83.0	59.4	Percent
9	Ethiopia	1994	36.0	18.5	Percent
10	Ethiopia	2004	50.0	22.8	Percent
11	India	1991	61.6	33.7	Percent
12	India	2001	73.4	47.8	Percent
13	Indonesia	1990	88.0	75.3	Percent

Рис. 8 – решение задачи «Время исследований»

В данной таблице представлены, как уже говорилось выше, данные о проценте образованных мужчин и женщин в топ-10 стран мира.

In [36]:
```
df.describe()
```

Out[36]:

	Year	Male	Female
count	30.000000	30.000000	30.000000
mean	1997.933333	78.403333	66.090000
std	6.191949	19.510130	28.641053
min	1986.000000	36.000000	15.100000
25%	1991.000000	62.225000	36.750000
50%	2000.000000	88.200000	83.900000
75%	2003.750000	92.900000	88.775000
max	2005.000000	99.700000	99.200000

Исходя из данных вычислений мы обратим внимание на среднее арифметическое наших данных, которое указано в строке mean, а так же на среднее квадратичное отклонение, также указаное в строке std.

In [38]:
```
df.corr()
```

Out[38]:

	Year	Male	Female
Year	1.000000	0.060173	0.077442
Male	0.060173	1.000000	0.970204
Female	0.077442	0.970204	1.000000

In [42]:
```
sns.pairplot(df);
```

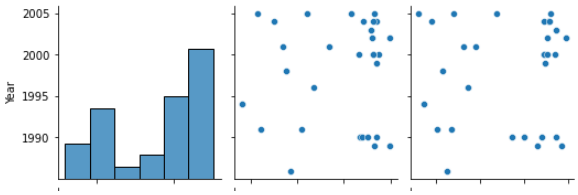


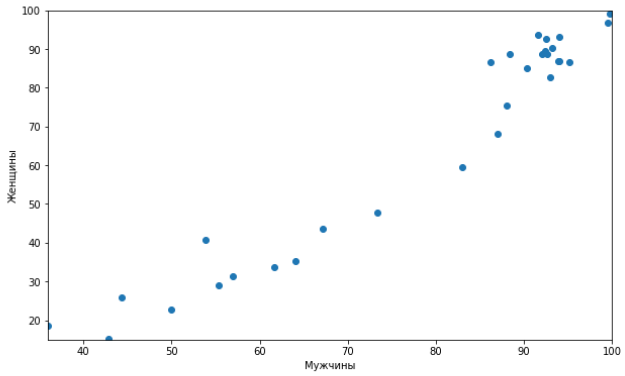
Рис. 9 – решение задачи «Время исследований»

Далее найдём уравнение линейной регрессии для каждого из полученных валидных коэффициентов корреляции.

```
In [56]:
1 data= pd.read_csv('Lvl_development.csv')
2
```

```
In [47]:
1 plt.figure(figsize=(10, 6))
2 plt.scatter(data.Male, data.Female)
3 plt.xlabel('Мужчины')
4 plt.ylabel('Женщины')
5 plt.xlim(36, 100)
6 plt.ylim(15, 100)
```

```
Out[47]: (15.0, 100.0)
```




```
In [3]:
1 model = LinearRegression()
```

```
In [48]:
1 x = pd.DataFrame(data.Male)
2 y = pd.DataFrame(data.Female)
```

```
In [50]:
1 model.fit(x, y)
```

```
Out[50]: LinearRegression()
```

Рис. 10 – решение задачи «Время исследований»

```
In [51]:
1 model.coef_
```

```
Out[51]: array([[1.42426891]])
```

```
In [52]:
1 model.intercept_
```

```
Out[52]: array([-45.57743022])
```

```
In [54]:
1 plt.figure(figsize=(10, 6))
2 plt.scatter(data.Male, data.Female)
3
4 plt.plot(x, model.predict(x), color = 'red')
5
6 plt.xlabel('Мужчины')
7 plt.ylabel('Женщины')
8 plt.xlim(36, 100)
9 plt.ylim(15, 100)
10
```

```
Out[54]: (15.0, 100.0)
```

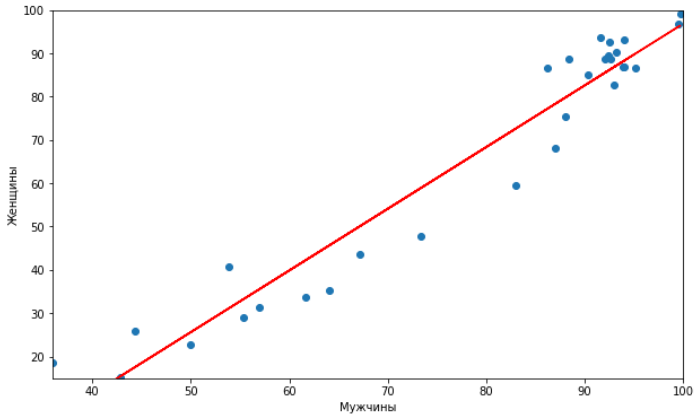


Рис. 11 – решение задачи «Время исследований»

Ответы на вопросы:

1. Как осуществляется запуск Jupyter notebook?

Для запуска Jupyter Notebook следует перейти в папку Scripts и в командной строке набрать:

> ipython notebook

В результате будет запущена оболочка в браузере.

2. Какие существуют типы ячеек в Jupyter notebook?

Ячейки с шрифтом Code содержат в себе информацию кода программы, а также результат выполнения кода. Ячейки с шрифтом Markdown содержат в себе текст.

3. Как осуществляется работа с ячейками в Jupyter notebook?

После выбора ячейки с шрифтом Code в неё можно записать код на языке Python и нажать Ctrl+Enter или Shift+Enter, в первом случае введённый код будет выполнен интерпретатором Python, во втором – будет выполнен код и создана новая ячейка, которая расположится уровнем ниже.

Alt+Enter – выполнить содержимое ячейки и вставить новую ячейку ниже.

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

Важной частью функционала Jupyter Notebook является поддержка магии. Под магией в IPython понимаются дополнительные команды, выполняемые в рамках оболочки, которые облегчают процесс разработки и расширяют ваши возможности. Список доступных магических команд можно получить с помощью команды «%lsmagic».

Для работы с переменными окружения используется команда %env. (%env a = 5)

Запуск Python кода из “.py” файлов, а также из других ноутбуков – файлов с расширением “.ipynb”, осуществляется с помощью команды %run. (%run ./test.py)

Для измерения времени работы кода используйте %%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 среднем значении трех наиболее быстрых прогонах.

%matplotlib – используется для отображения объектов графиков на экране, ключ после него указывает каким способом отображать график. При запуске без ключей он использует сторонний backend и отображает график в отдельном окне. При вводе команды %matplotlib notebook график появляется не в отдельном окне, а прямо внутри вашего блокнота, при этом он так же интерактивен. Команда %matplotlib inline указывает, что график необходимо построить все в той же оболочке Jupyter, но теперь он выводится как обычная картинка. Данный способ удобен тем, что позволяет проводить очень много экспериментов в рамках одного окна (точнее web-страницы). В этом статическом режиме, никакие изменения не отобразятся до тех пор пока не будет выполнена команда plt.show().

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.

PyCharm:

1. Сначала вы должны создать новый проект.

2. В этом проекте создайте новый файл ipynb, выбрав File> New…> Jupyter Notebook. Это должно открыть новый файл записной книжки.

3. Если у вас не установлен пакет Jupyter Notebook, над вновь открытым файлом ipynb появится сообщение об ошибке. Сообщение об ошибке гласит: «Пакет Jupyter не установлен», и у вас будет опция «Установить пакет jupyter» рядом с ним.

4. Нажмите «Установить пакет jupyter». Это запустит процесс установки, который вы можете просмотреть, щелкнув запущенные процессы в правом нижнем углу окна PyCharm.

5. Чтобы начать изучение Jupyter Notebook в PyCharm, создайте ячейки кода и выполните их.

6. Выполните ячейку кода, чтобы запустить сервер Jupyter. По умолчанию сервер Jupyter использует порт 8888 на локальном хосте. Эти конфигурации доступны в окне инструментов сервера. После запуска вы можете просмотреть сервер над окном исходного кода, а рядом с ним вы можете просмотреть ядро, созданное как «Python 2» или «Python 3».

7. Теперь вы можете получить доступ к вкладке переменных в PyCharm, чтобы увидеть, как значения ваших переменных меняются при выполнении ячеек кода. Это помогает при отладке. Вы также можете установить точки останова в строках кода, а затем щелкнуть значок «Выполнить» и выбрать

«Debug Cell» (или использовать сочетание клавиш Alt+Shift+Enter), чтобы начать отладку.

Visual Studio Code:

Чтобы создать новый Jupyter Notebook, запустите палитру команд (Ctrl+Shift+P) и выполните поиск new notebook. Первым результатом должен быть Jupyter: Create New Blank Jupyter Notebook. Вы также можете создать его, нажав на новый файл .ipynb, но горячие клавиши значительно повышают эффективность работы.

Обратите внимание, что блокноты, созданные VS Code, по умолчанию являются доверенными (trusted). С другой стороны, любые импортированные помечаются как not trusted, чтобы защитить пользователя от выполнения вредоносного кода. Таким образом, ставить пометку trust нужно вручную по запросу редактора перед выполнением.

После создания блокнота нажмите на иконку save на верхней части панели инструментов, чтобы сохранить его в рабочем пространстве. Теперь можно экспортировать созданный блокнот как скрипт Python или файл HTML/PDF, используя соответствующую иконку.

По умолчанию в новом блокноте появится пустая ячейка. Добавьте в нее код и выполните его с помощью ctrl+enter. Эта команда запустит выделенную ячейку. shift+enter выполняет то же действие, но при этом создает и выделяет новую ячейку ниже, а alt+enter выполняет выделенную, создает еще одну ниже, но при этом сохраняет метку на предыдущей.

Иконка + добавляет новую ячейку для кода, а bin удаляет ее. Чтобы перемещать фрагменты вверх и вниз, воспользуйтесь соответствующими стрелками.

Изменить тип ячейки на markdown довольно просто: просто нажмите на иконку M, расположенную над кодом. Чтобы снова установить значение code, выберите значок {}. Выполнить эти действия также можно с помощью клавиш M и Y.

Теперь рассмотрим опции отладки. Во-первых, расширение Jupyter для VS Code поддерживает построчное выполнение кода в ячейке. Просто нажмите на кнопку, расположенную рядом с иконкой play.

Вторая возможность для отладки дает весомую причину использовать Jupyter в VS Code. Вы можете просто экспортировать блокнот как скрипт Python и работать с ним прямо в отладчике VS Code, не переходя в другую среду.