

**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ  
ФЕДЕРАЦИИ**

**Федеральное государственное автономное  
образовательное учреждение высшего образования  
«СЕВЕРО-КАВКАЗСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»**

## Кафедра инфокоммуникаций

**Отчет по лабораторной работе № 3.4**  
**по дисциплине «Технологии распознавания образов»**

Выполнил студент группы ПИЖ-б-о-21-1

Трушева В. О. .«\_\_»\_\_ 2023г.

Подпись студента\_\_\_\_\_

Работа защищена «    »                      20    г.

Проверила Воронкин Р.А. \_\_\_\_\_  
(подпись)

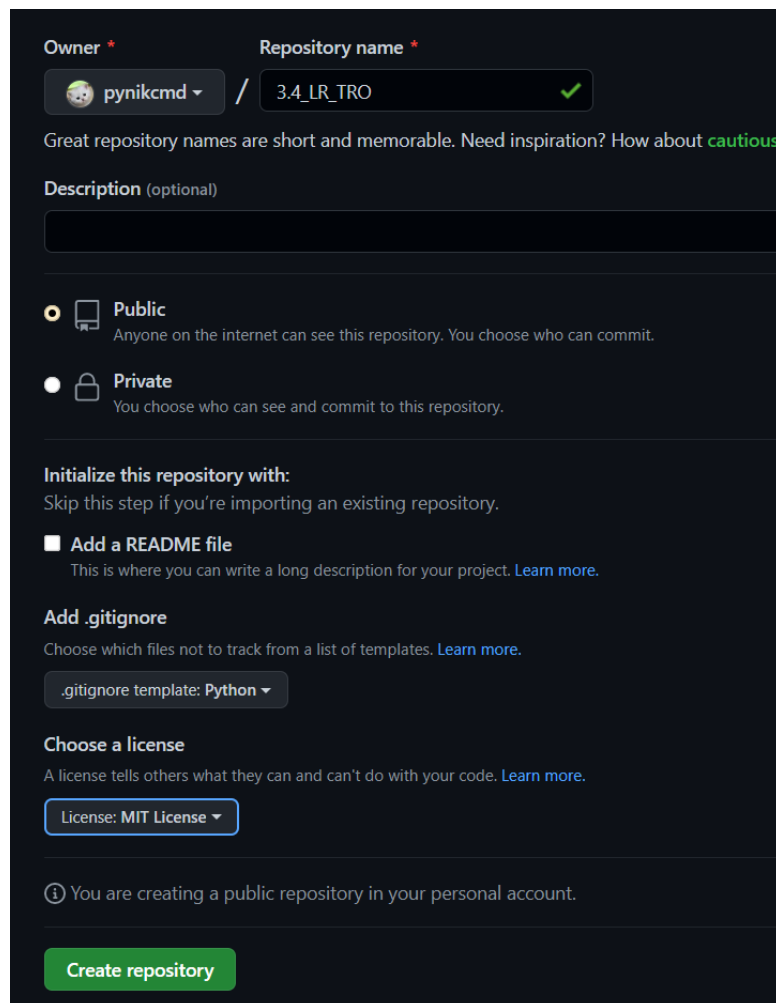
Ставрополь 2023

Цель работы: исследовать базовые возможности библиотеки NumPy языка программирования Python.

Методика и порядок выполнения работы

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).



The image shows the GitHub 'Create repository' form. At the top, the 'Owner' is set to 'pynikcmd' and the 'Repository name' is '3.4\_LR\_TRO', which is marked with a green checkmark. Below this, there is a text input field for the 'Description (optional)'. The 'Public' option is selected with a radio button, and the 'Private' option is unselected. Under the 'Initialize this repository with:' section, the 'Add a README file' checkbox is checked. The 'Add .gitignore' section shows the '.gitignore template: Python' selected. The 'Choose a license' section shows the 'License: MIT License' selected. At the bottom, there is a green 'Create repository' button. A note at the bottom states: 'You are creating a public repository in your personal account.'

Рисунок 1 – Создание репозитория

3. Выполните клонирование созданного репозитория на рабочий компьютер.

```
D:\fgit>git clone https://github.com/pynikcmd/3.4_LR_TRO.git
Cloning into '3.4_LR_TRO'...
remote: Enumerating objects: 7, done.
remote: Counting objects: 100% (7/7), done.
remote: Compressing objects: 100% (6/6), done.
remote: Total 7 (delta 1), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (7/7), done.
Resolving deltas: 100% (1/1), done.
```

Рисунок 2 – Клонирование репозитория

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

```
D:\fgit\3.4_LR_TRO>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? [D:/fgit/3.4_LR_TRO/.git/hooks]
```

Рисунок 3 – Модель 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 в отдельном ноутбуке.

### Примеры лабораторной работы 3.4

```
In [1]: import matplotlib.pyplot as plt  
%matplotlib inline
```

```
In [6]: matplotlib.__version__
```

```
Out[6]: '3.5.2'
```

```
In [13]: plt.plot([1, 2, 3, 4, 5], [1, 2, 3, 4, 5])
```

```
Out[13]: [<matplotlib.lines.Line2D at 0x1c580ae11f0>]
```

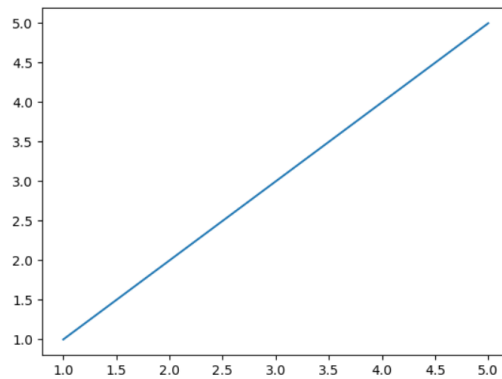


Рисунок 4 – Примеры лабораторной работы

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

### Вопросы для защиты работы

1. Как осуществляется установка пакета matplotlib?

Существует два основных варианта установки этой библиотеки: в первом случае вы устанавливаете пакет Anaconda, в состав которого входит большое количество различных инструментов для работы в области машинного обучения и анализа данных (и не только); во втором – установить Matplotlib самостоятельно, используя менеджер пакетов.

2. Какая "магическая" команда должна присутствовать в ноутбуках Jupyter для корректного отображения графиков matplotlib?

```
%matplotlib inline
```

3. Как отобразить график с помощью функции plot?

Для построения графика используется команда `plot()`. Если в качестве параметра функции `plot()` передать список, то значения из этого списка будут отложены по оси ординат (ось *y*), а по оси абсцисс (ось *x*) будут отложены индексы элементов массива.

Для того, чтобы задать значения по осям *x* и *y* необходимо в `plot()` передать два списка.

```
plt.plot([1, 2, 3, 4, 5], [1, 2, 3, 4, 5])
```

4. Как отобразить несколько графиков на одном поле?

Для того, чтобы вывести несколько графиков на одном поле необходимо передать соответствующие наборы значений в функцию `plot()`.

```
plt.plot(x, y1, x, y2)
```

5. Какой метод Вам известен для построения диаграмм категориальных данных?

Метод `bar()`

6. Какие основные элементы графика Вам известны?

- Заголовок (`title`)
- Легенда (`legend`)
- Основная сетка (`major grid`)
- Линейный график (`plot`)
- Точечный график (`scatter`)
- Дополнительные тики (`minor ticks`) Фигура (`figure`)
- Дополнительная сетка (`minor grid`) Подпись оси *y* (`y label`)
- Основные тики (`major ticks`) Подпись оси *x* (`x label`)

7. Как осуществляется управление текстовыми надписями на графике? Наименование осей: `plt.xlabel()`, `plt.ylabel()`

Заголовок графика: `plt.title()` Текстовое примечание: `plt.text()`  
Легенда: `plt.legend()`

8. Как осуществляется управление легендой графика?

Легенда будет размещена на графике, если вызвать функцию `legend()`

9. Как задать цвет и стиль линий графика?

Задание цвета: `plt.plot(x, y, color='red')`, `plt.setp( color='red', linewidth=1)` Задание цвета линии графика производится через параметр `color` (или `c`, если использовать сокращенный вариант). Значение может быть представлено в одном из следующих форматов:

- RGB или RGBA кортеж значений с плавающей точкой в диапазоне [0, 1] (пример: (0.1, 0.2, 0.3))
- RGB или RGBA значение в hex формате (пример: '#0a0a0a')
- строковое представление числа с плавающей точкой в диапазоне [0, 1] (определяет цвет в шкале серого) (пример: '0.7')
- символ из набора {'b', 'g', 'r', 'c', 'm', 'y', 'k', 'w'}
- имя цвета из палитры X11/CSS4
- цвет из палитры `xkcd`(<https://xkcd.com/color/rgb/>), должен начинаться с префикса 'xkcd:'
- цвет из набора Tableau Color (палитра T10), должен начинаться с префикса 'tab:'

Если цвет задается с помощью символа из набора {'b', 'g', 'r', 'c', 'm', 'y', 'k', 'w'}, то он может быть совмещен со стилем линии в рамках параметра `fmt` функции `plot()`. Например штриховая красная линия будет задаваться так: '—r', а штрих пунктирная зеленая так '—g'

## 10. Как выполнить размещение графика в разных полях?

Существуют три основных подхода к размещению нескольких графиков на разных полях:

- использование функции `subplot()` для указания места размещения поля с графиком;
- использование функции `subplots()` для предварительного задания сетки, в которую будут укладываться поля;
- использование `GridSpec`, для более гибкого задания геометрии размещения полей с графиками в сетке.

Самый простой способ представить графики в отдельных полях – это использовать функцию `subplot()` для задания их мест размещения.

После задания размера, указывается местоположение, куда будет установлено поле с графиком с помощью функции `subplot()`. Чаще всего используют следующие варианты вызова `subplot`:

*`subplot(nrows, ncols, index)`*

- `nrows (int)` – количество строк.
- `ncols (int)` – количество столбцов. `index(int)` – местоположение элемента

*`subplot(pos)`*

- `pos (int)` – позиция, в виде трехзначного числа, содержащего информацию о количестве строк, столбцов и индексе, например 212, означает подготовить разметку с двумя строками и одним столбцов, элемент вывести в первую позицию второй строки. Этот вариант можно использовать, если количество строк и столбцов сетки не более 10, в ином случае лучше обратиться к первому варианту.