

МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра МО ЭВМ

ОТЧЕТ
по практической работе №3
по дисциплине «Искусственные нейронные сети»
Тема: Представление данных и библиотека NumPy

Студентка гр. 8383

Максимова А.А.

Преподаватель

Жангиров Т.Р.

Санкт-Петербург

2021

Цель работы.

Задача 3. Написать функцию, которая возвращает все уникальные строки матрицы.

Примечания к задачам.

- Необходимо использовать модуль NumPy
- Все данные должны считываться из файла в виде массива NumPy
- Результаты необходимо сохранять в файл

Выполнение.

Была подключена библиотека NumPy: `import numpy as np`

Были написаны 3 функции:

- `def read_data(type_data)` - функция, отвечающая за считывание данных из файла в виде массива NumPy, с помощью функции `np.genfromtxt`. Название файла, из которого происходит считывание, и тип данных запрашиваются у пользователя;
- `def save_data(matr, type_data)` - функция, отвечающая за сохранение результата работы программы, в случае, если он был получен (на вход была подана матрица). Название файла для сохранения и тип данных запрашиваются у пользователя. Сохранение реализуется с помощью функции `np.savetxt`;
- `def unique_strings(matr)` - функция, возвращающая матрицу, состоящую только из уникальных строк, если они есть. Под уникальностью подразумевается то, что строка встречается лишь один раз. При выполнении работы функции происходит вывод промежуточных результатов.

Реализация def unique_strings(matr).

На вход функции поступает считанная из файла матрица. Их исходной матрицы, в цикле, поочередно вычитаются составляющие ее строки, начиная с нулевой. Пример, представлен на рисунке ниже:

$$\begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \\ 6 & 7 & 8 & 9 & 10 \\ 11 & 12 & 13 & 14 & 15 \\ 1 & 2 & 3 & 4 & 5 \end{array} - \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array}$$

Так как из матрицы вычитается вектор, то вектор будет преобразован для использования автоматически (повторен несколько раз). После чего будет выполнено поэлементное вычитание.

$$\begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \\ 6 & 7 & 8 & 9 & 10 \\ 11 & 12 & 13 & 14 & 15 \\ 1 & 2 & 3 & 4 & 5 \end{array} - \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \\ 1 & 2 & 3 & 4 & 5 \\ 1 & 2 & 3 & 4 & 5 \\ 1 & 2 & 3 & 4 & 5 \end{array} = \begin{array}{ccccc} 0 & 0 & 0 & 0 & 0 \\ 5 & 5 & 5 & 5 & 5 \\ 10 & 10 & 10 & 10 & 10 \\ 0 & 0 & 0 & 0 & 0 \end{array}$$

В результате во временной переменной будет храниться матрица с одной или более нулевыми строками. С помощью следующей конструкции мы получаем матрицу, из которой будут удалены все нулевые строки:

```
temp_matr = matr[~((matr-matr[cur_row]) == 0).all(axis=1)],  
где (matr-matr[cur_row]) - результат вычитания текущей строки из матрицы;  
ndarray.all() - возвращает True, если все элементы равны True;  
axis - ось, по которой выполняется логическое сокращение AND;  
~ - отрицание.
```

После выполненного преобразования проверяется насколько меньше строк во временной переменной по сравнению с исходной: если удалили только одну строку, значит она уникальна, а следовательно, оставляем ее в исходной матрице и переходим к проверке следующих строк.

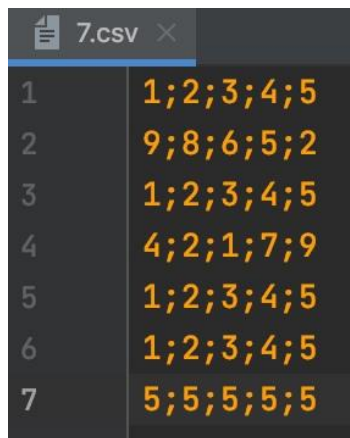
Иначе, удаляем повторяющиеся строки, и проверяем следующую строку, переместившуюся на позицию удаленной строки.

Цикл работает до тех пор, пока индекс текущей проверяемой строки меньше количества строк в матрице, то есть пока не дошли до последней строки.

Тестирование.

Подробное.

Содержимое файла 7.csv из директории tests:



1	1;2;3;4;5
2	9;8;6;5;2
3	1;2;3;4;5
4	4;2;1;7;9
5	1;2;3;4;5
6	1;2;3;4;5
7	5;5;5;5;5

Промежуточный вывод программы:

```
Исходная матрица:
[[1 2 3 4 5]
 [9 8 6 5 2]
 [1 2 3 4 5]
 [4 2 1 7 9]
 [1 2 3 4 5]
 [1 2 3 4 5]
 [5 5 5 5 5]]

Проверка уникальности для строки:
[1 2 3 4 5]

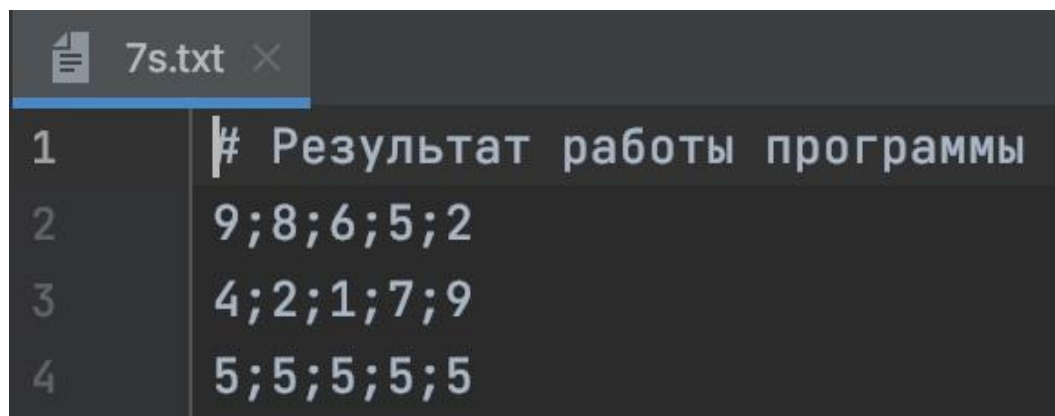
Удаление повторяющийся строки!
Количество повторов строки равно:
4

Проверка уникальности для строки:
[9 8 6 5 2]

Строка уникальна!
```

```
Проверка уникальности для строки:  
[4 2 1 7 9]  
  
Строка уникальна!  
  
Проверка уникальности для строки:  
[5 5 5 5 5]  
  
Строка уникальна!  
  
Матрица, состоящая только из уникальных строк:  
[[9 8 6 5 2]  
 [4 2 1 7 9]  
 [5 5 5 5 5]]  
Введите название файла для сохранения результата работы программы: 7s.txt
```

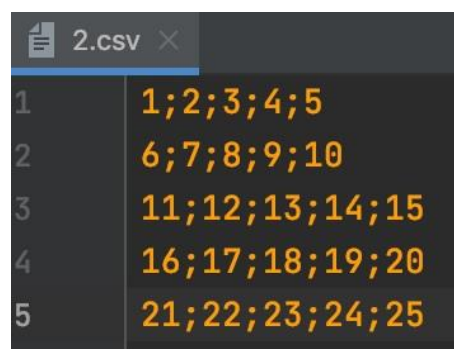
Результаты работы программы в новом файле:



```
7s.txt ×  
1 # Результат работы программы  
2 9;8;6;5;2  
3 4;2;1;7;9  
4 5;5;5;5;5
```

Краткое.

1. Вход:



```
2.csv ×  
1 1;2;3;4;5  
2 6;7;8;9;10  
3 11;12;13;14;15  
4 16;17;18;19;20  
5 21;22;23;24;25
```

Выход:

```
2s.txt x
1 # Результат работы программы
2 1;2;3;4;5
3 6;7;8;9;10
4 11;12;13;14;15
5 16;17;18;19;20
6 21;22;23;24;25
```

2. Вход:

```
1.csv x
1 1;1;1;1;1;1;1;1;1;1
2 1;1;1;1;1;1;1;1;1;1
3 1;1;1;1;1;1;1;1;1;1
4 1;1;1;1;1;1;1;1;1;1
5 1;1;1;1;1;1;1;1;1;1
6 0;0;0;0;0;0;0;0;0;0
7 0;0;0;0;0;0;0;0;0;0
8 0;0;0;0;0;0;0;0;0;0
9 0;0;0;0;0;0;0;0;0;0
10 0;0;0;0;0;0;0;0;0;0
```

Выход:

```
1s.csv x
1 # Результат работы программы
2
```

3. Вход:

```
8.csv x
1 5;5;5;5;5
2 5;5;5;5;5
3 -1;-1;-1;-1;-1
4 1;1;1;1;1
5 0;0;0;0;0
6 1;1;1;1;1
```

Выход:

```
8s.csv ×
1  # Результат работы программы
2  -1;-1;-1;-1;-1
3  0;0;0;0;0
```

4. Вход:

```
float.csv ×
1  1.1;1.2;1.3;1.4;1.5
2  1.1;1.2;1.3;1.4;-1.5
3  1.2;1.2;1.3;1.4;1.5
4  1.1;1.2;1.3;1.4;1.5
5  1.1;32.2;1.3;1.4;1.5
6  1.1;1.1;1.3;1.4;1.5
```

Выход:

```
float_save.txt ×
1  # Результат работы программы
2  1.100000;1.200000;1.300000;1.400000;-1.500000
3  1.200000;1.200000;1.300000;1.400000;1.500000
4  1.100000;32.200000;1.300000;1.400000;1.500000
5  1.100000;1.100000;1.300000;1.400000;1.500000
```

Все тесты содержатся в папке tests (11 тестов).