Задание:

Исследовать функционал одного модуля не из стандартной библиотеки (например, joblib) и создать фрагмент ЭОР с описанием и примерами его использования при работе в Jupyter Notebook и в скриптах. Для выполнения задания использовать Jupyter Notebook, опубликовать результат выполнения задания в портфолио в HTML и PDF формате.

NumPy

NumPy — это библиотека языка Python, добавляющая поддержку больших многомерных массивов и матриц, вместе с большой библиотекой высокоуровневых (и очень быстрых) математических функций для операций с этими массивами.

Функции и методы	**Описание**
array(<список компонентов>,[<тип данных>])	Создаёт массив из списка компонентов. Можно указать тип данных
zeros(<размерность>, [<тип данных>])	Создаёт специальнный массив указонной размерности, заполненный нулями. Можно указать тип данных
ones(<размерность>, [<тип данных>])	Создаёт специальный массив указанной размерности, заполненный еденицами. Можно указать тип данных
identity(<размер>,[<тип данных>])	Создаёт специальный *n* x *n* массив с единицами на главной диагонали
arange(<от>,<до>,<шаг>, [<тип данных>])	Возвращает массив с равномерно разнесенными значения в пределах заданного интервала. Можно указать шаг
linspace(<первый>, <последний>,<размер>)	Аналогичена arange, только указывается не значение шага, а их колличество, шаг высчитывается автоматически
.**size**	Возвращает количество элементов массива
.shape	Возвращает или задаёт размер массива (количество строк и столбцов)
transpose(a)	Транспонирование матрицы a
.**ravel**()	Возвращает flattened массив (в одну строчку)
resize(<массив>, <размерность>)	Ихменяет размерность массива
.**tile**(<массив>, <размерность>)	Повторяет массив указанное на указанное число повторений
sort([<ось>=-1])	Возвращает отсортированую копию массива

Примеры

```
#объявление массива
np.array([1, 2, 3])
array([1, 2, 3])
```

```
#всё элементы объявляемого массива числа типа float
>>> np.array([1, 2, 3.0])
array([ 1.,  2.,  3.])
```

```
>>> np.array([[1, 2], [3, 4]])
array([[1, 2],
       [3, 4]])
```

```
#массив из 5-ти элементов, которые являются нулями (по умолчанию типа float)
>>> np.zeros(5)
array([ 0.,  0.,  0.,  0.,  0.])
```

```
>>> np.zeros((5,), dtype=int)
array([0, 0, 0, 0, 0])
```

```
>>> np.zeros((2, 1))
array([[ 0.],
       [ 0.]])
```

```
>>> s = (2,2)
>>> np.zeros(s)
array([[ 0.,  0.],
       [ 0.,  0.]])
```

```
np.ones(5)
array([ 1.,  1.,  1.,  1.,  1.])
```

```
>>> np.ones((5,), dtype=int)
array([1, 1, 1, 1, 1])
```

```
#создаётся единичная матрица n на n
np.identity(3)
array([[ 1.,  0.,  0.],
       [ 0.,  1.,  0.],
       [ 0.,  0.,  1.]])
```

```
#создаётся массив из указанного диапазона
np.arange(3)
array([0, 1, 2])

>>> np.arange(3.0)
array([ 0.,  1.,  2.])

>>> np.arange(3,7)
array([3, 4, 5, 6])

>>> np.arange(3,7,2)
array([3, 5])
```

```
#колличество элементов массива (на основе матрицы)
>>> a = array([[11,12,13],[21,22,23]])
>>> a.size
6
```

```
#полученные массивы можно перемножать, складывать, вычитать
>>> a = np.array([1.0, 2.0, 3.0])
>>> b = np.array([2.0, 2.0, 2.0])
>>> a * b
array([ 2.,  4.,  6.])
```

```
#sharp возвращает количество строк и столбцов матрицы
>>> a = array([[11,12,13],[21,22,23]])
>>> a.size
>>> 6
>>> a.shape
>>> (2, 3)
>>> a.shape=(3,2)
>>> a
>>> array([[11, 12],
          [13, 21],
          [22, 23]])
```

```
#имеется возможность транспонирования мматрицы
>>> a
>>> array([[11, 12],
>>> [13, 21],
>>> [22, 23]])
>>> transpose(a)
>>> array([[11, 13, 22],
>>> [12, 21, 23]]
```

```
#к примеру, двумерный массив можно представить в виде одной строки
>>> a
>>> array([[11, 12, 13],
>>> [21, 22, 23]])
>>> a.ravel()
>>> array([11, 12, 13, 21, 22, 23])
>>> a
>>> array([[11, 12, 13],
>>> [21, 22, 23]])
```

```
#создание массива из определённого элемента с заданием количества строк и столбцов
tile(pi,(2,3))
array([[ 3.14159265, 3.14159265, 3.14159265],
[ 3.14159265, 3.14159265, 3.14159265]])

>>> tile([1,2],4)
>>> array([1, 2, 1, 2, 1, 2, 1, 2])

>>> a = array([[11,12],[21,22]])
>>> tile(a,(2,3))
>>> array([[11, 12, 11, 12, 11, 12],
>>> [21, 22, 21, 22, 21, 22],
>>> [11, 12, 11, 12, 11, 12],
>>> [21, 22, 21, 22, 21, 22]])
```