

# Самое необходимое о классах: Д3

## **Условие**

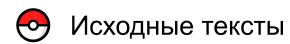
Peaлизуйте класс CountVectorizer, имеющий

- метод <u>fit transform</u>
- метод <u>get feature names</u>

#### Условия:

- пользоваться внешними пакетами запрещено
- решение должно быть в \*.ру файлах
- не должно быть замечаний РЕР8
- если умеешь, то напиши проверки/тесты

Далее на слайдах пример получения терм-документной матрицы



#### Crock Pot Pasta

Never boil pasta again



#### Pasta Pomodoro

Fresh ingredients Parmesan to taste





## Подсчитываем кол-во слов

#### Crock Pot Pasta

Never boil pasta again

#### Список слов

0	crock	7	fresh
1	pot	8	ingredients
2	pasta	9	parmesan
3	never	10	to
4	boil	11	taste
5	again		
6	pomodoro		

#### Pasta Pomodoro

Fresh ingredients Parmesan to taste



## Строим матрицу

#### Crock Pot Pasta

Never boil pasta again

#### Список слов

0	crock	7	fresh
1	pot	8	ingredients
2	pasta	9	parmesan
3	never	10	to
4	boil	11	taste
5	again		
6	pomodoro		

#### Pasta Pomodoro

Fresh ingredients Parmesan to taste

#### Вектора

1 [1, 1, 2, 1, 1, 1, 0, 0, 0, 0, 0, 0] 2 [0, 0, 1, 0, 0, 0, 1, 1, 1, 1, 1]

# Пример использования

Peanusyйте класс CountVectorizer, имеющий метод fit\_transform

```
corpus = [
        'Crock Pot Pasta Never boil pasta again',
        'Pasta Pomodoro Fresh ingredients Parmesan to taste'
4
5
    vectorizer = CountVectorizer()
    count matrix = vectorizer.fit transform(corpus)
    print(vectorizer.get feature names())
    Out: ['crock', 'pot', 'pasta', 'never', 'boil', 'again', 'pomodoro',
10
           'fresh', 'ingredients', 'parmesan', 'to', 'taste']
11
12
    print(count matrix)
13
    Out: [[1, 1, 2, 1, 1, 1, 0, 0, 0, 0, 0, 0],
14
           [0. 0. 1. 0. 0. 0. 1. 1. 1. 1. 1. 1]
```



Спасибо за проделанную работу!