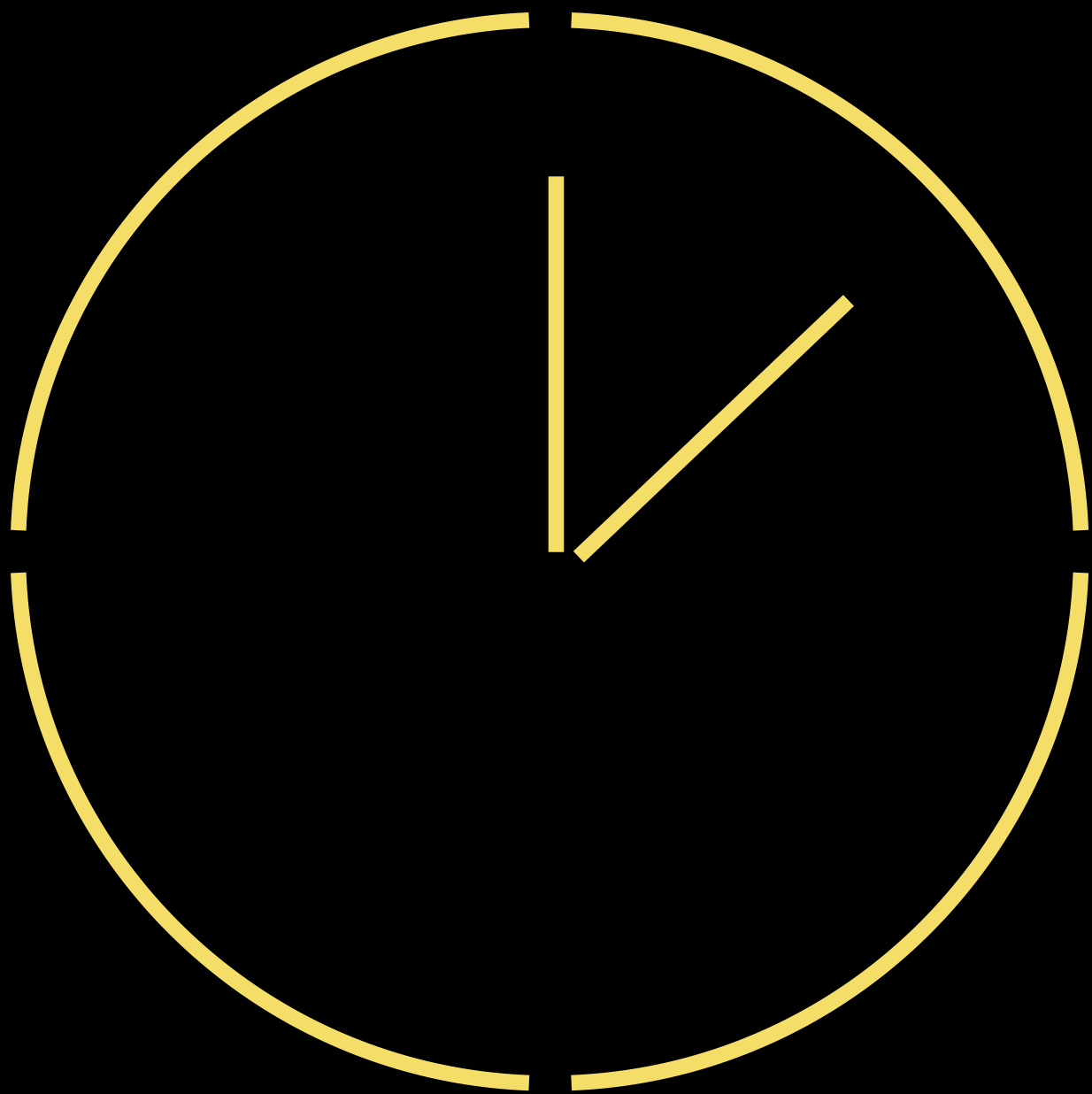


Немного о нестандартных применениях статистики и регулярных выражениях

Александр Ольферук,
наставник

Яндекс Практикум

Цели на консультацию



Первая часть – 40 минут

- Трюк с кодированием переменных
- Сортировка с затухающим средним



Перерыв – 10 минут



Вторая часть – 30 минут

- Регулярные выражения

Кодирование переменных с помощью статистики

Регрессия как задача взвешивания

Площадь квартиры, м²

Наличие балкона

Удаленность от центра, км

Высота потолков, м

Наличие панорамных окон

Регрессия как задача взвешивания

Площадь квартиры, м ²	65
Наличие балкона	1
Удаленность от центра, км	3.2
Высота потолков, м	2.5
Наличие панорамных окон	0

Регрессия как задача взвешивания

Площадь квартиры, м ²	65	x 80 000
Наличие балкона	1	x 150 000
Удаленность от центра, км	3.2	x (-10 000)
Высота потолков, м	2.5	x 70 000
Наличие панорамных окон	0	x 100 000

Регрессия как задача взвешивания

Площадь квартиры, м ²	65	x 80 000	
Наличие балкона	1	x 150 000	
Удаленность от центра, км	3.2	x (-10 000)	5 493 000
Высота потолков, м	2.5	x 70 000	
Наличие панорамных окон	0	x 100 000	

Кодирование переменных

- Label Encoding

	student	grade
0	John	A
1	John	B
2	John	C
3	John	D
4	John	E
5	John	F
6	Mary	A
7	Mary	A
8	Mary	B

Кодирование переменных

- Label Encoding

```
from sklearn.preprocessing import LabelEncoder
```

```
le = LabelEncoder()
```

```
df['grade'].values
```

```
array(['A', 'B', 'C', 'D', 'E', 'F', 'A', 'A', 'B'], dtype=object)
```

```
le.fit_transform(df['grade'])
```

```
array([0, 1, 2, 3, 4, 5, 0, 0, 1])
```

Кодирование переменных

- Label Encoding

```
from sklearn.preprocessing import LabelEncoder
```

```
le = LabelEncoder()
```

```
df['grade'].values
```

```
array(['Excellent', 'Very good', 'Average', 'Mediocre', 'Bad', 'Very bad',  
      'Excellent', 'Excellent', 'Excellent'], dtype=object)
```

```
le.fit_transform(df['grade'])
```

```
array([2, 5, 0, 3, 1, 4, 2, 2, 2])
```

Кодирование переменных

- Label Encoding

```
label_binarize(df['grade'], classes=['Excellent', 'Very good', 'Average', 'Mediocre', 'Bad', 'Very bad'])\
    .argmax(axis=1)

array([0, 1, 2, 3, 4, 5, 0, 0, 0])
```

Кодирование переменных

- One-Hot Encoding

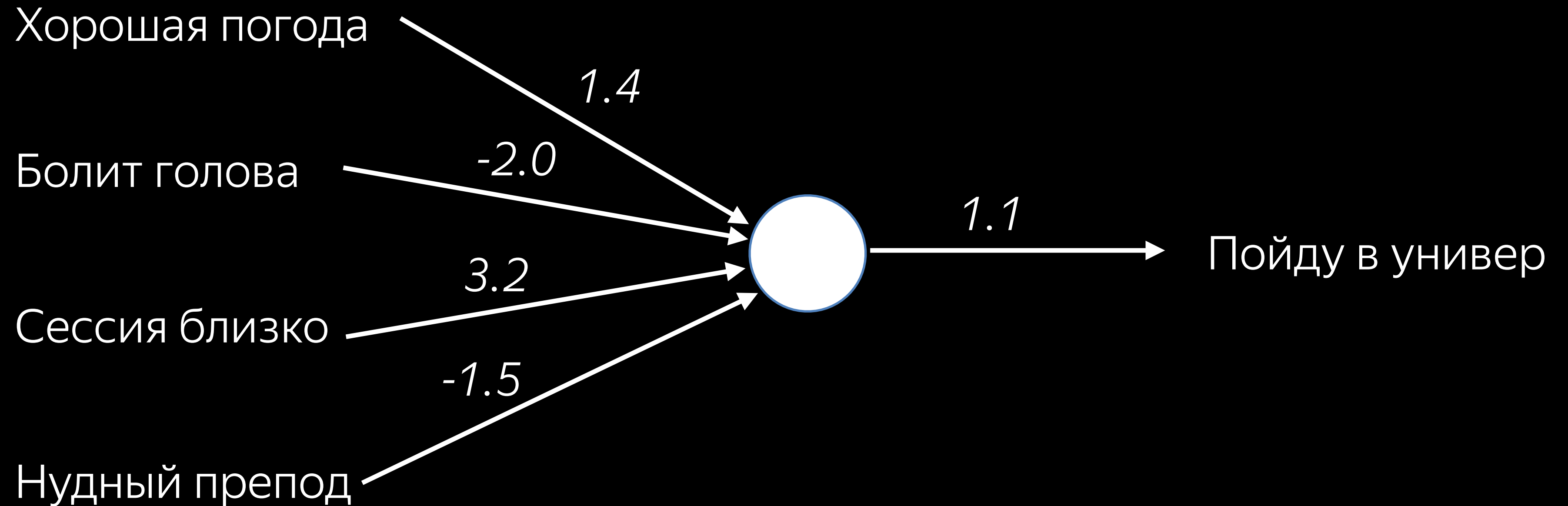
```
label_binarize(df['grade'], classes=['Excellent', 'Very good', 'Average', 'Mediocre', 'Bad', 'Very bad'])  
  
array([[1, 0, 0, 0, 0, 0],  
       [0, 1, 0, 0, 0, 0],  
       [0, 0, 1, 0, 0, 0],  
       [0, 0, 0, 1, 0, 0],  
       [0, 0, 0, 0, 1, 0],  
       [0, 0, 0, 0, 0, 1],  
       [1, 0, 0, 0, 0, 0],  
       [1, 0, 0, 0, 0, 0],  
       [1, 0, 0, 0, 0, 0]])
```

Как мы принимаем решения

За

Против

Как мы принимаем решения



Отталкиваемся от статистики

50% всех вылетов из аэропорта **А**, были задержаны

13% всех вылетов из аэропорта **Б**, были задержаны

Можем заменить каждое **А** на 0.5, а каждое **Б** - на 0.13

Отталкиваемся от статистики

50% всех вылетов из аэропорта **А**, были задержаны

13% всех вылетов из аэропорта **Б**, были задержаны

Можем заменить каждое **А** на 0.5, а каждое **Б** - на 0.13

Но на деле все не так просто

Отталкиваемся от статистики

Если в нашей выборке из аэропорта **Б** было 1000 вылетов (строчек), то задержке в 13% случаев доверять в целом можно

А если из аэропорта **А** было 2 вылета, и один из них задержался, то заменять **А** на 50% неразумно

Отталкиваемся от статистики

Можем посчитать среднюю задержку по всем вылетам и добавить некоторое количество таких вылетов к каждой группе, отдельно к вылетам из **А**, отдельно к вылетам из **Б**

Это называется *регуляризацией* (в терминах данного подхода),
или *regularization term*

Отталкиваемся от статистики

Можем посчитать среднюю задержку по всем вылетам и добавить некоторое количество таких вылетов к каждой группе, отдельно к вылетам из **А**, отдельно к вылетам из **Б**

Было:

$$\frac{\sum_{i=1}^N target_i}{N}$$

Стало:

$$\frac{\sum_{i=1}^N target_i + \mu N_r}{N + N_r}$$

Отталкиваемся от статистики

Представим, что средняя задержка у нас по всем данным возникает в 10% случаев, а размер группы-регуляризатора - 100.

Тогда **Б** мы будем заменять на $(1000*0.13 + 100*0.1)/1100 = 0.127$

В то же время **А** мы будем заменять на $(2*0.5 + 100*0.1)/102 = 0.108$

Обратите внимание, как хорошо стало в обоих случаях: **Б** почти не сдвинулась, а аномально маленькая группа **А** не портит картину.

О сортировке

[Ссылка на эту страницу](#)

Количество оценивших
дает устойчивую оценку,
но, с другой стороны, нельзя
отсортировать по рейтингу,
от лучших - к худшим.

[illegible]

О сортировке

[Ссылка на эту страницу](#)

В то же время сортируя по рейтингу, находим сотни игр, оцененные одним-двумя людьми. В итоге такой оценке тоже доверять нельзя.


Linked Items > Board Games

Average Rating ▾Category ▾Mechanic ▾

+ Add

Showing 1-25 of 4,374

<>




10

8-Cylinder Overlords (2019)

Ratings	1	Own	1	Want in Trade	0
Weight	0.00	Prev. Own	0	Wishlist	0
Comments	1	For Trade	0		

Add To




10

AEROSTAR (2014)

Ratings	2	Own	3	Want in Trade	1
Weight	5.00	Prev. Own	0	Wishlist	3
Comments	3	For Trade	0		

Add To




10

Auction Lords (2017)

Ratings	1	Own	1	Want in Trade	0
Weight	0.00	Prev. Own	0	Wishlist	3
Comments	0	For Trade	0		

Add To



10

Battle of Capua (2015)

Ratings	1	Own	1	Want in Trade	0
Weight	0.00	Prev. Own	0	Wishlist	1
Comments	0	For Trade	0		

Add To

Что делать?

Тот же принцип приходит на помощь и здесь: добавляем некоторое количество оцененных средне, на 5/10, игр, невидимых и несуществующих.

Взвешиваем это с имеющимися оценками, и сортируем по данному критерию.

$$\frac{\sum_{i=1}^N target_i + \mu N_r}{N + N_r}$$

Перерыв

10 минут

Регулярные выражения

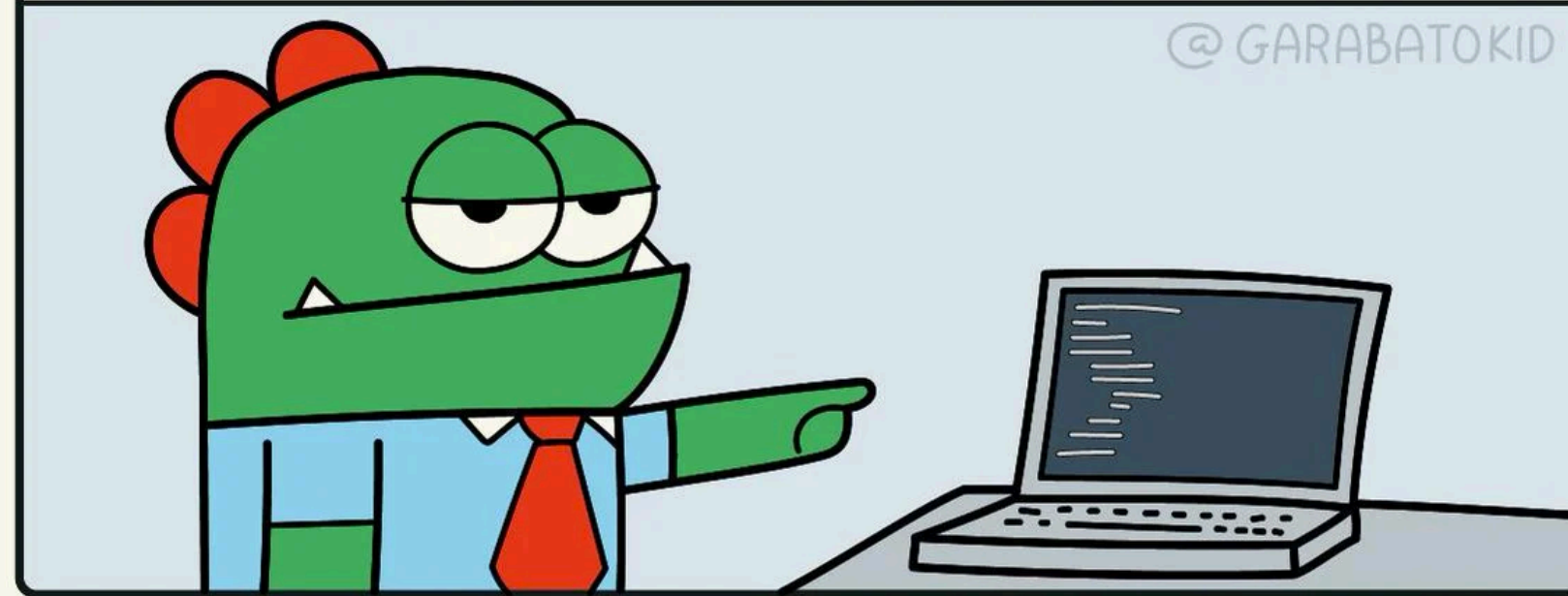
Регулярные выражения



Регулярные выражения

HOW TO REGEX

STEP 1: OPEN YOUR FAVORITE EDITOR



STEP 2: LET YOUR CAT PLAY ON YOUR KEYBOARD



Что это такое?

Регулярные выражения – грамматика поиска подстрок в строках.

Другими словами, это набор правил, описывающий искомую строку в тексте.

С помощью регулярных строк можно выцепить из текста все, что похоже на телефонные номера, имейлы, веб-адреса и все, на что у вас хватит фантазии :)

Пример

Ищем телефонные номера: все вида **+7 (951) 123-23-01**.

Как мы можем описать эту структуру?

- Плюс семь. **Плюса может и не быть.**
- **Опционально пробел**
- Дальше скобка, **но ее может и не быть**
- Три цифры
- Закрывающая скобка, **но и ее может не быть**
- .. и так далее

Какие есть служебные символы?

\d – любая цифра (от слова digit)

\D – любая **не**-цифра

\w – любая буква (от слова word)

\W – любая **не**-буква

\s – любой пробел (от слова space): пробелы, табы, переносы строк

\S – любой **не**-пробел

. – любой символ

\b – конец слова

\B – **не**-конец слова

^ – начало строки

\$ – конец строки

Какие есть служебные символы?

[a-z] – любая буква от **a** до **z**

[^a-z] – все, что угодно, но не буква от **a** до **z**

[0-9] – любая цифра. Эквивалент **\d**

[123] – символ 1, 2 или 3. Не 123!

[A-Za-z0-9_] – эквивалент **\w**

Какие есть квантификаторы?

? – 0 или 1 раз (может быть, а может и не быть)

+ – 1 и более

+? – 1 и более (не жадный вариант)

***** – 0 и более

***?** – 0 и более (не жадный вариант)

{1,4} – от 1 до 4 раз

{2,} – от 2 раз

| – аналог ИЛИ. Требует скобок: **(http|www)**

Квантификаторы и жадность

Задача: найти все, что написано в скобках.

Представим, что работаем с таким текстом:

Я люблю (хотя иногда когда как) есть омлет на завтрак (с сосисочками и помидорчиками (грибы я не очень люблю)).

Квантификаторы и жадность

Задача: найти все, что написано в скобках.

Представим, что работаем с таким текстом:

Я люблю (хотя иногда когда как) есть омлет на завтрак (с сосисочками и помидорчиками (грибы я не очень люблю)).

Если написать `\([a-яA-Я\s\(\)]+\)`, то заключена будет эта область:

Я люблю (хотя иногда когда как) есть омлет на завтрак (с сосисочками и помидорчиками (грибы я не очень люблю)).

Квантификаторы и жадность

Задача: найти все, что написано в скобках.

Представим, что работаем с таким текстом:

Я люблю (хотя иногда когда как) есть омлет на завтрак (с сосисочками и помидорчиками (грибы я не очень люблю)).

Если написать `\([a-яA-Я\s\(\)]+\)`, то заключена будет эта область:

Я люблю (хотя иногда когда как) есть омлет на завтрак (с сосисочками и помидорчиками (грибы я не очень люблю)).

Сделаем выражение не жадным: `\([a-яA-Я\s\(\)]+?\)`

Я люблю (хотя иногда когда как) есть омлет на завтрак (с сосисочками и помидорчиками (грибы я не очень люблю)).

Экранирование

Именно поэтому, если вы хотите использовать символы `.`, `{`, `}`, `?`, `+`, `*` и так далее как символы, а не как команды, используем экранирование: `\.` описывает точку.

Верно и обратное: символы `\d`, `\w` используются с `\`, чтобы не быть перепутанными с символами `d` или `w`.

Флажки

/i – ignore case, то есть регистронезависимый поиск

/m – multiline, то есть **^** и **\$** подходят к началу и концу одной строки или текста в целом

/g – Эти

/u – вам

/s – не

/y – нужны.

Python

```
import re                                # от regular expressions
re.sub(r'\d', '~', '123 abc')           # ~~~ abc
match = re.search(r'иии', 'пиииво')     # нашел, match.group() == "иии"
all_tokens = re.findall(r'майонез', f.read()) # находит весь
                                              "майонез" в файле
```

Домашняя работа

Ваших знаний на текущий момент должно быть достаточно, чтобы сделать:

- программу-фильтр шуток для КВН: на вход принимает имя текстового файла и имя файла-результата. Проходит по нему, если в предложении есть слово **“Путин”**, вырезает шутку. Отфильтрованные предложения записывает в файл-результат.
- программу, считающую количество **“Так, стоп!”** в сценарии. На вход – имя файла-сценария. Сделать регистронезависимый поиск и не считать те **“так, стоп”**, которые имеют другой смысл: *“а он прост**так, стоп**орился об косяк”*.



Домашняя работа

Решать:

- [здесь](#) есть несколько задач по регуляркам
- и [еще здесь](#)
- [regex-кроссворд!](#)

Читать:

- Резник А.Д. "Книга для тех, кто не любит статистику, но вынужден ею пользоваться. Непараметрическая статистика в примерах, упражнениях и рисунках"
- Гмурман В.Е. "Теория вероятностей и математическая статистика"

Для профессионалов

- Виленкин Н.Я. “Комбинаторика”
- Ширяев А.Н. “Вероятность”

Немного о нестандартных применениях статистики и регулярных выражениях

Александр Ольферук,
наставник

Яндекс Практикум