

Python

Лекция 5: Алгоритмы на строках. Классификация

Отметься на портале!



План занятия

- Префиксное дерево
- Суффиксное дерево
- Алгоритм КМП
- Классификация
- Метрики классификации
- Логистическая регрессия
- Векторное представление текстов

Подготовимся к лекции

1. Переходим в директорию вашего форка
2. Открываем в ней терминал
3. `git checkout master`
4. `git pull upstream master`
5. `git push -u origin master`

Типы данных. Строки

Неизменяемые !

Строка представляет собой последовательность символов. Мы можем использовать одинарные или двойные кавычки для создания строки.

```
s = "строка1"
print("Для строки {}, адрес: {}".format(s, id(s)))
s = s.replace("с", "С")
print("Для строки {}, адрес: {}".format(s, id(s)))
```

Для строки строка1, адрес: 140368660461024
Для строки Строка1, адрес: 140368660410768

```
s = "строка1"
s[0] = "С"
```

```
-----
TypeError                                 Traceback (most recent call last)
<ipython-input-17-0d4e365a2eba> in <module>()
      1 s = "строка1"
----> 2 s[0] = "С"
```

TypeError: 'str' object does not support item assignment

Все - объект!

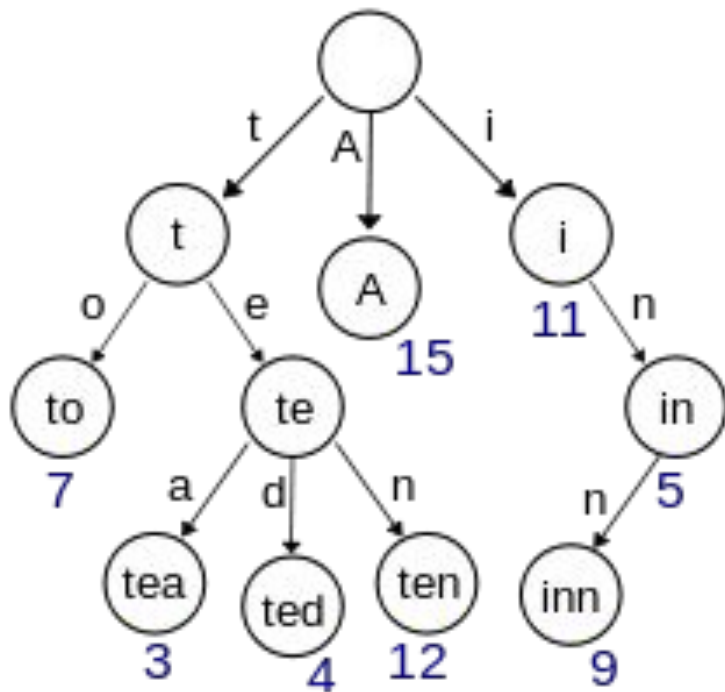


Задачи на строках:

1. Найти наибольшую общую подстроку
2. Проверить вхождение строки в строку
3. Саджесты
4. ...

Давайте подумаем, какая функциональность может быть связана со строками?

Префиксное дерево



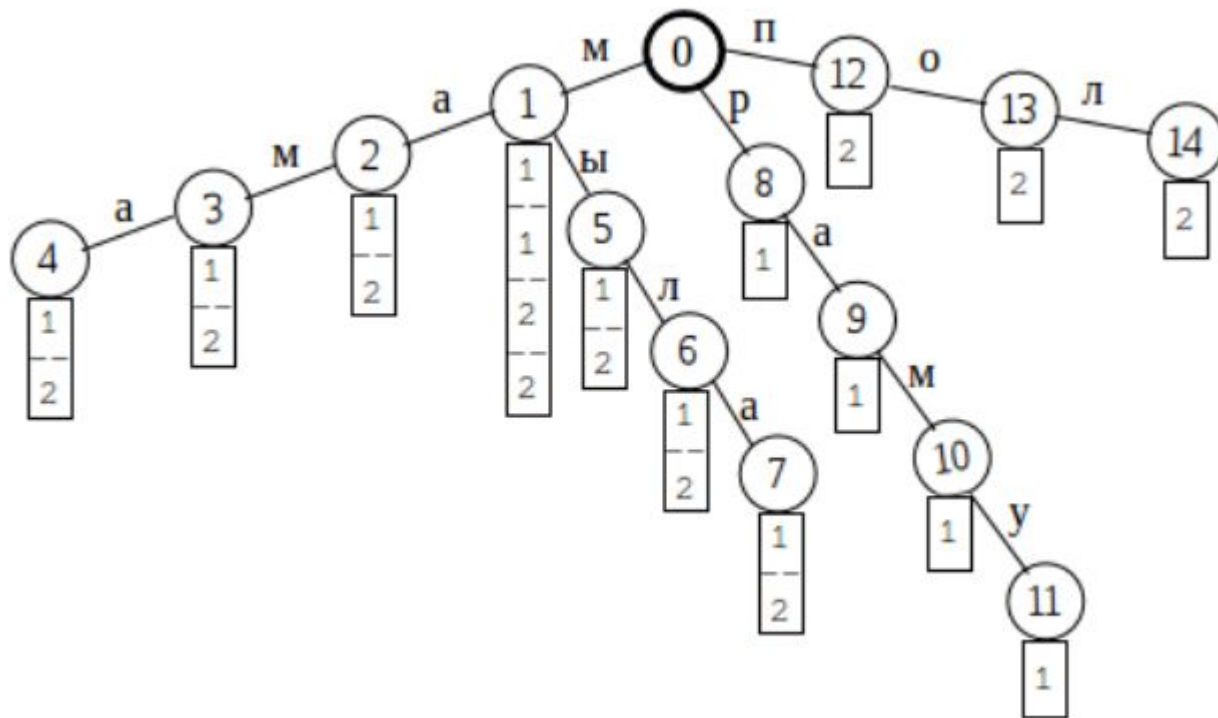
```
import string
import datatree
trie = datatree.Trie(string.ascii_lowercase)

trie[u'foo'] = 5
trie[u'foobar'] = 10
trie[u'bar'] = 'bar value'
trie.setdefault(u'foobar', 15)
```

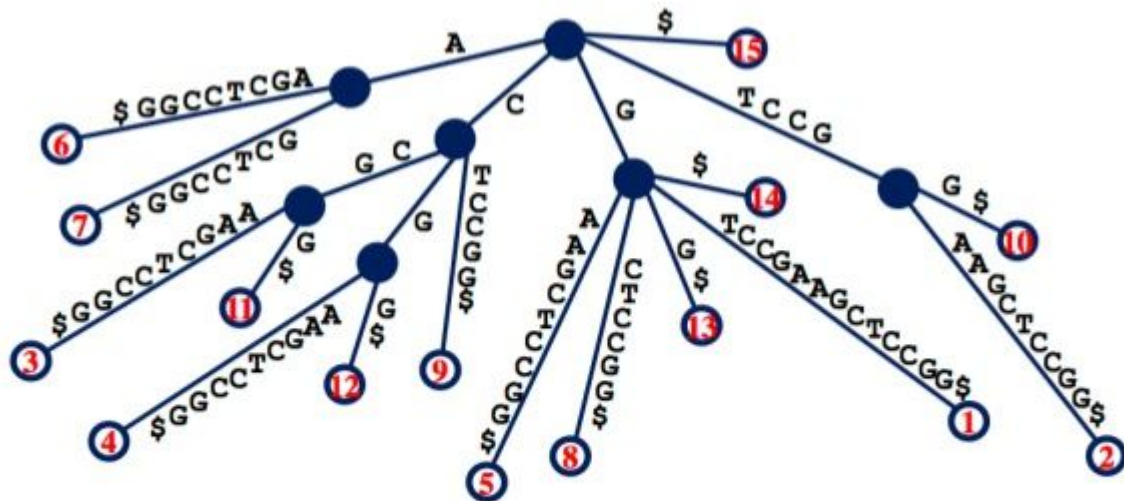
Зачем? - **Поисковые саджесты**



Префиксное дерево



Суффиксное дерево



Зачем? - Поиск вхождения
подстроки в строку

GTCCGAAGCTCCGG\$

Задача классификации

	text	name_user	date	rating	review_positive
0	Приложение не обновляется и не скачивается за...	Sasha Pershina	9 июля 2017 г.	1.0	0
1	Хорошее приложение облегчает платежи	Oleg S	9 июля 2017 г.	5.0	1
2	Нет предела совершенству. Практичное приложен...	Евгения Ключникова	9 июля 2017 г.	4.0	1
3	Нормас работает		9 июля 2017 г.	4.0	1
4	Удобное приложение, основные функции отработы...		9 июля 2017 г.	5.0	1

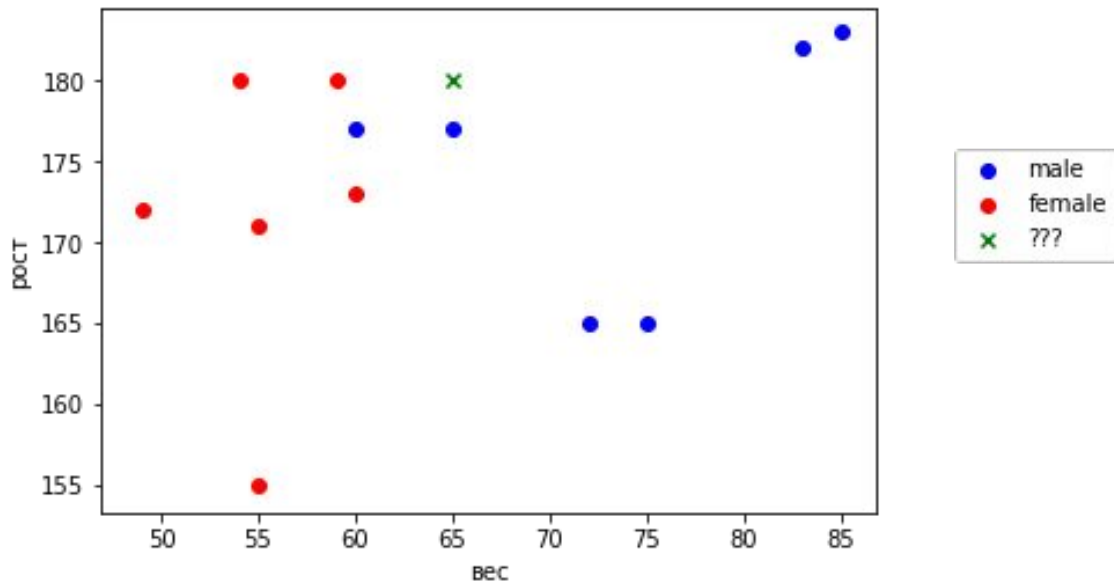
1. Хороший фильм, игра актеров очень понравилась
 2. Фильм на один раз, в целом неплохо, но кот плохо изображал сочувствие
 3. Ну вообще я не любитель ужасов, но теперь всей семьей ходим в костюме клоуна
 4. Это ПРОСТА ТРЕШ!!!
-



1. Настоящий украинский борщ готовится из
 2. Сало - неотъемлемая часть украинской кухни
 3. В этом тексте мы кратко рассмотрим особенности японской кухни
-
1. Собаки предпочитают педигри, вообще они прикольные домашние животные
 2. Африканская гончая такса предпочитает есть из миски
 3. Коты презируют людей, тогда как собаки очень добры. Коты - это животное из семейства кошачьих....

Постановка задачи классификации

Пусть X — множество описаний объектов, Y — множество номеров (или наименований) классов. Существует неизвестная *целевая зависимость* — отображение из X в Y , значения которой известны только на объектах конечной обучающей выборки.



Какой классификатор мы уже знаем?

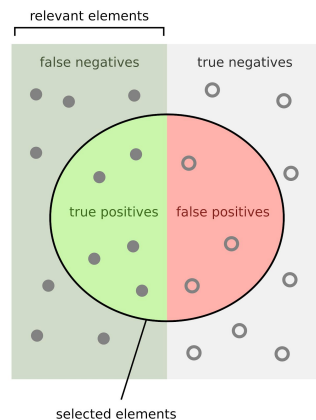


Классификация. Метрики.

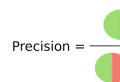
$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

$$\text{Recall} = \frac{TP}{TP + FN}$$

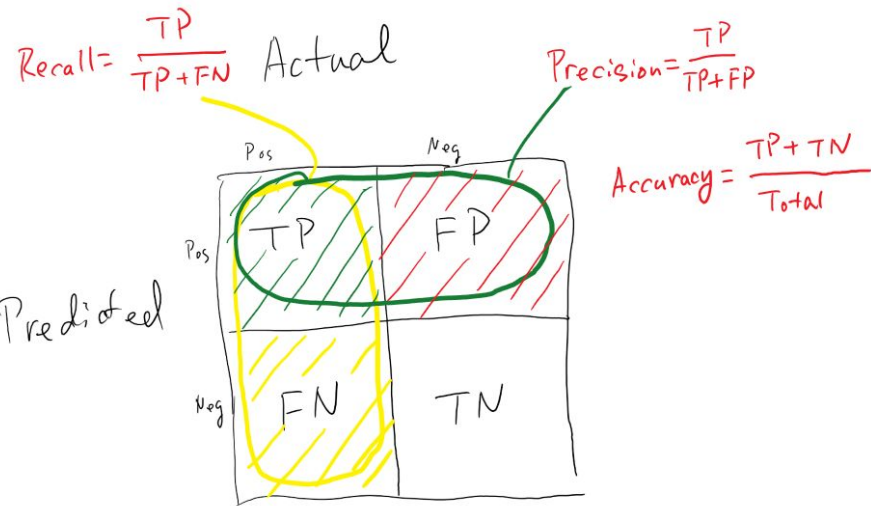
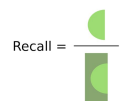
$$\text{Precision} = \frac{TP}{TP + FP}$$



How many selected items are relevant?

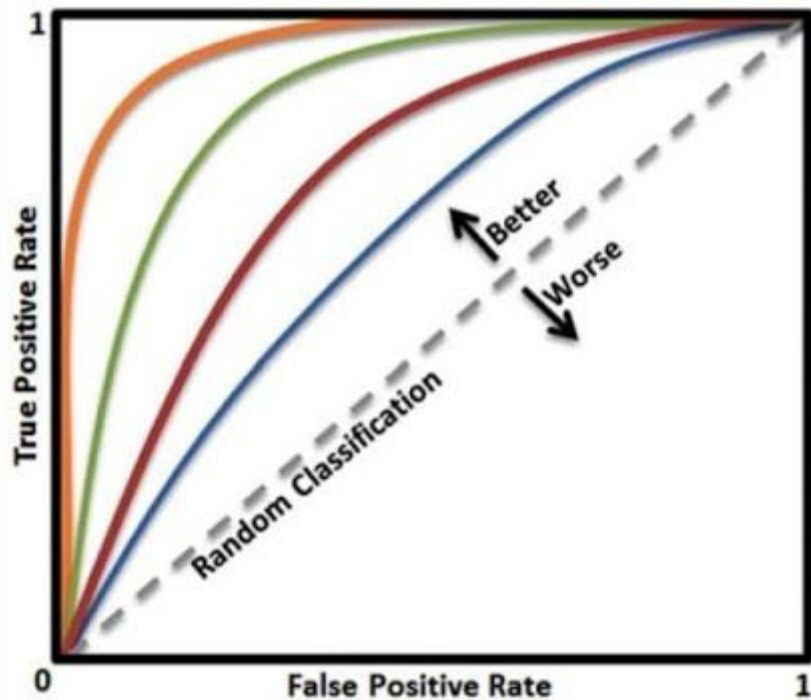


How many relevant items are selected?



Придумаем случай, когда они не работают

Классификация. Метрики



predicted→ real↓	<i>Class_pos</i>	<i>Class_neg</i>
<i>Class_pos</i>	TP	FN
<i>Class_neg</i>	FP	TN

$$\text{TPR (sensitivity)} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

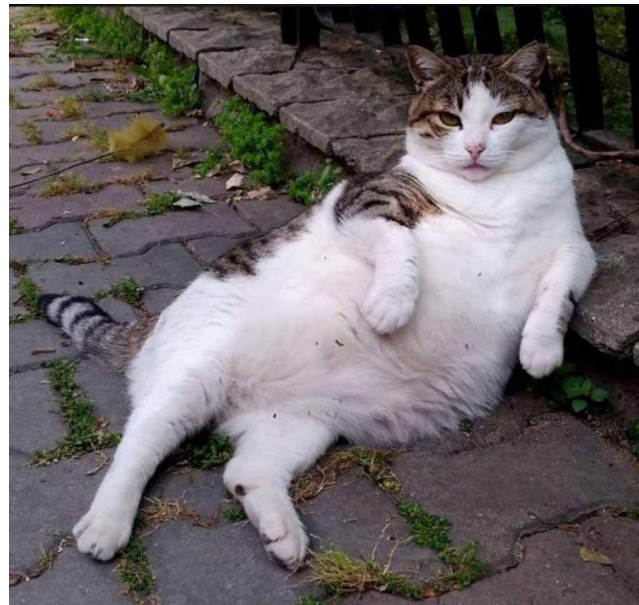
$$\text{FPR (1-specificity)} = \frac{\text{FP}}{\text{TN} + \text{FP}}$$

Придумаем случай, когда они не работают

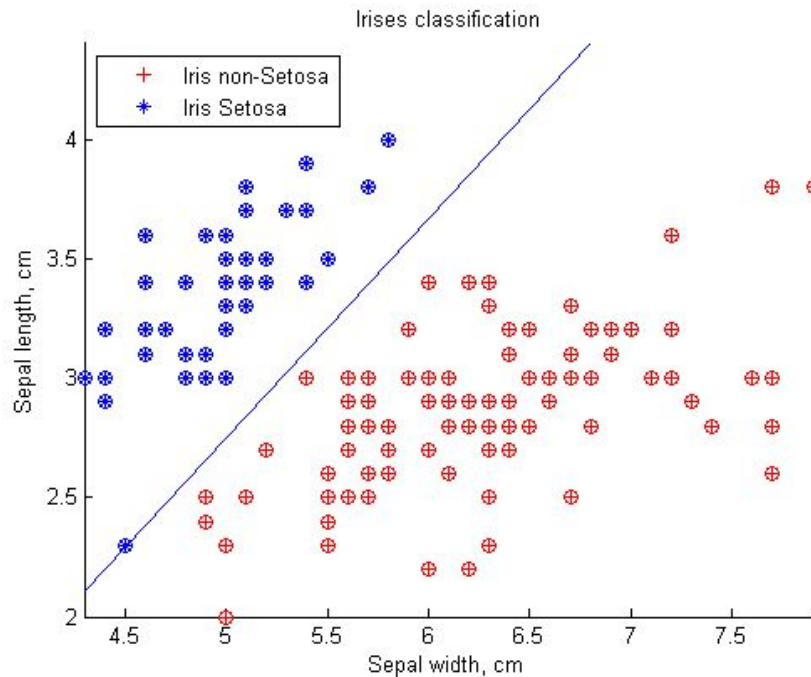
Поиграемся с метриками.

Придумаем случай, когда они не работают

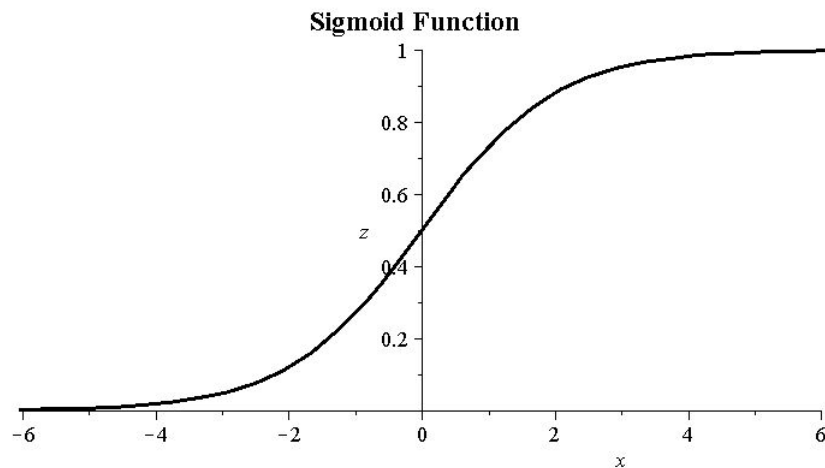
Сгенерировать в ноутбуке выборки, которые пройдут тесты



Логистическая регрессия



$$\log \frac{p}{1-p} = \alpha + \sum_{j=1}^d \beta_j x_j$$



Логистическая регрессия. Логлосс

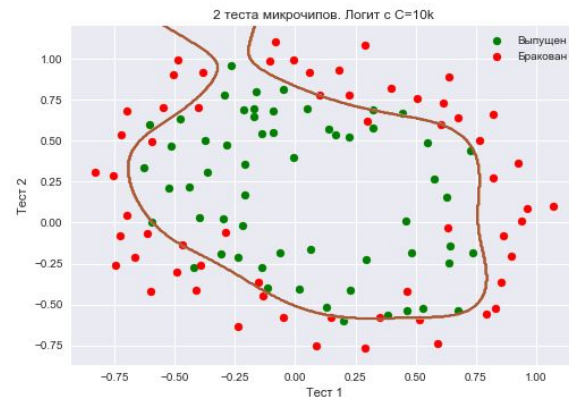
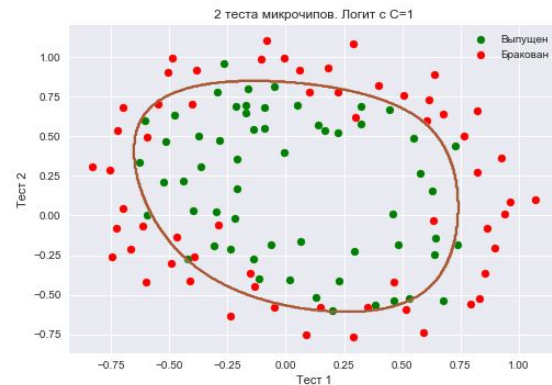
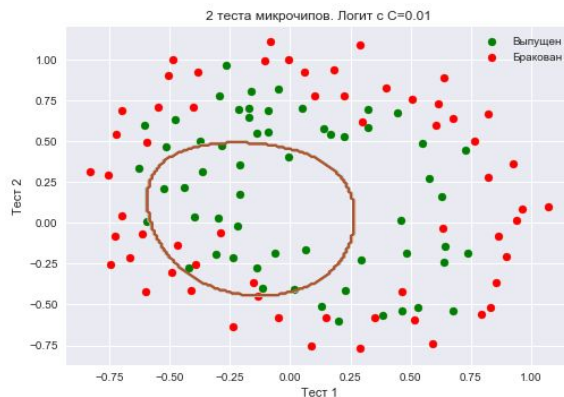
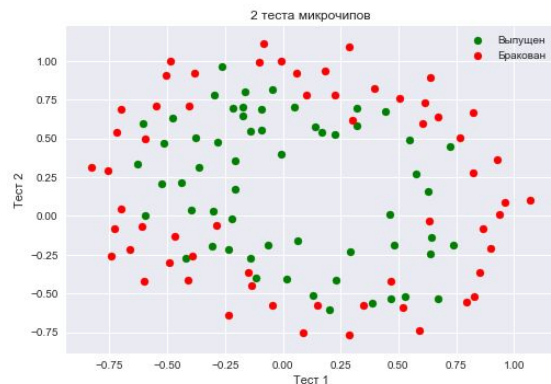
$$\text{Log Loss} = -\frac{1}{N} \sum_{i=1}^N [y_i \log p_i + (1-y_i) \log (1-p_i)]$$

Выведем!



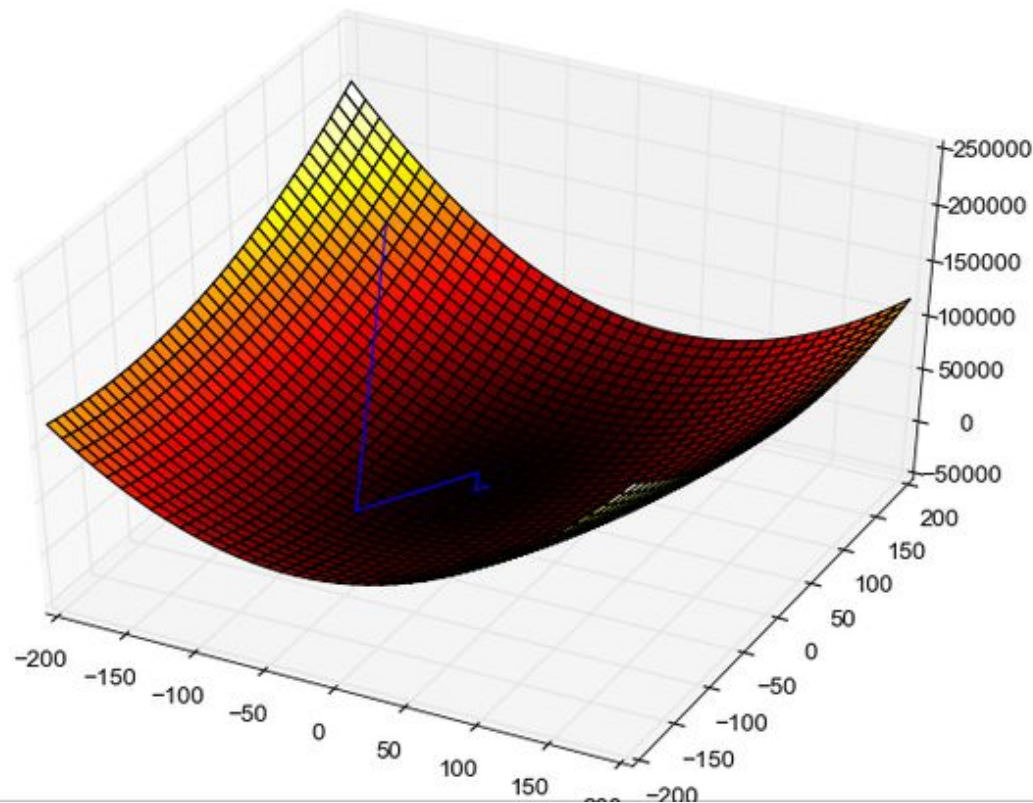
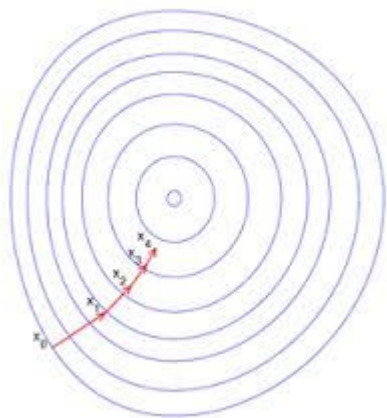
Логистическая регрессия. Регуляризация

$$J(X, \vec{y}, \vec{w}) = \mathcal{L}_{\log}(X, \vec{y}, \vec{w}) + \lambda |\vec{w}|^2$$



$$J(X, y, w) = \mathcal{L} + \frac{1}{C} ||w||^2$$

Градиентный спуск



Наивный Байесовский классификатор

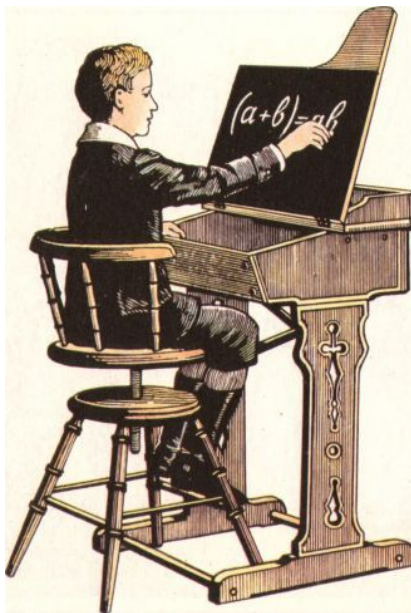
$$P(c|x) = \frac{P(x|c)P(c)}{P(x)}$$

Diagram labels:

- Likelihood: $P(x|c)$
- Class Prior Probability: $P(c)$
- Posterior Probability: $P(c|x)$
- Predictor Prior Probability: $P(x)$

$$P(c|X) = P(x_1|c) \times P(x_2|c) \times \dots \times P(x_n|c) \times P(c)$$

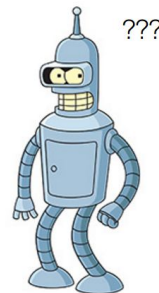
$$\hat{\theta}_{yi} = \frac{N_{yi} + \alpha}{N_y + \alpha n}$$



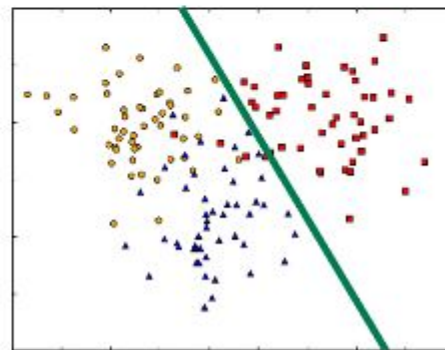
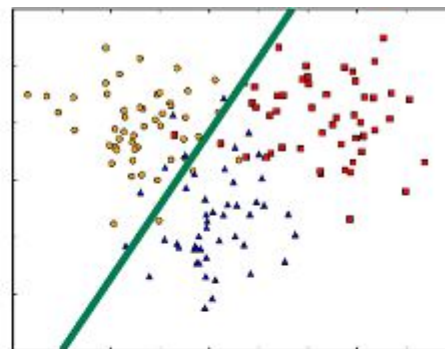
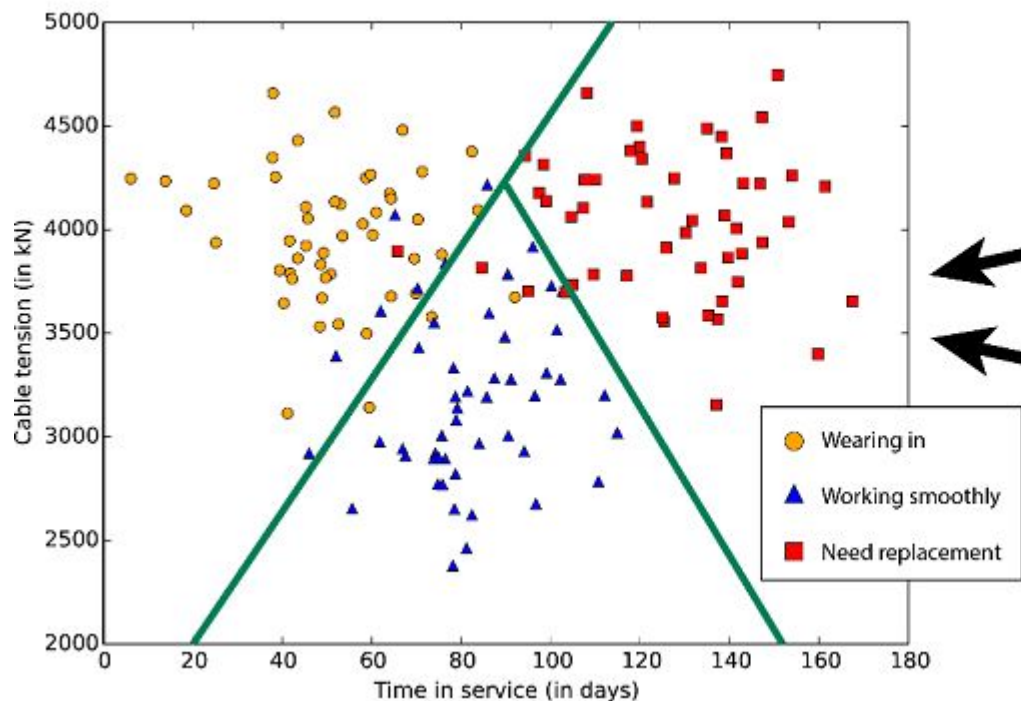
Обработка текстов

- 1) Как сделать многоклассовую классификацию тем же алгоритмом?
- 2) Как объяснить компьютеру смысл текста?
- 3) Как можно классифицировать тексты?
- 4) Как закодировать классы?
- 5) Какие вы видите проблемы?

Текст



Классификация. Многоклассовая



Обработка текстов

Эмбеddинг

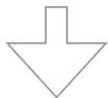
{корова, капуста, Agile, саранча}

Корова - [1, 0, 0, 0]

Капуста - [0, 1, 0, 0]

Agile - [0, 0, 1, 0]

Саранча - [0, 0, 0, 1]



Корова - [0, 0]

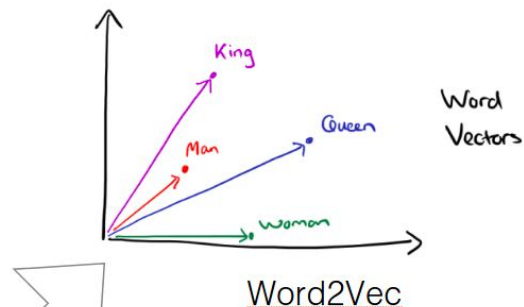
Капуста - [1, 0]

Agile - [0, 1]

Саранча - [1, 1]



Саранча - Agile - Капуста = Корова




Обработка текстов

Представление текста. Embedding.

(1) *John likes to watch movies. Mary likes movies too.*

(2) *John also likes to watch football games.*

Embedding: слово \rightarrow вектор, текст \rightarrow матрица.

- | | | |
|----------|---|---|
| • John | | $[-0.06, 0.92, -0.02, -0.82, \dots, -0.82, 1.06]$ |
| • likes |  | $[-0.32, 0.41, -0.11, 0.02, \dots, -0.34, 0.88]$ |
| • watch | | $[0.06, 0.00, 0.24, 0.21, \dots, -0.32, 0.39]$ |
| • movies | | $[0.11, 0.06, 0.28, -0.17, \dots, -0.67, 0.65]$ |

Обработка текстов

Стемминг и лемматизация

Сте́мминг — это процесс нахождения основы слова для заданного исходного слова. Основа слова необязательно совпадает с морфологическим корнем слова

Лемматиза́ция — процесс приведения словоформы к её нормальной (словарной) форме

Стемминг

Кошечк

Бежал

Боязненн

Лемматизация

Кошка

Бегать

Боязненный

Кошечка

Бежал

Боязненных

Еду

Обработка текстов

Векторизация текстов

Лемматизация

Мама мыла раму мылом



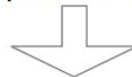
мама мыть рама мыло



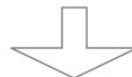
CountVectorizer

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

Мама мама рыла яму мылом



мама мама рыть яма мыло



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

Что еще?

TfidfVectorizer, NN, n-grams, stop words

Чему мы научились?

- 1) Привести все слова к стандартной форме в каждом тексте
- 2) Удалить частотные слова
- 3) Закодировать каждый текст
- 4) PROFIT компьютер понимает числа



Источники

[Логистическая регрессия](#)

[Ворд2век](#)

[О процессинге текстов](#)

[Еще про классификацию текстов](#)

[Суффиксное дерево](#)

Appendix. КМП-алгоритм

