

Машинное обучение.

Начало

ML: приложения



ML: приложения

≡ Google Translate ⋮

Text Documents

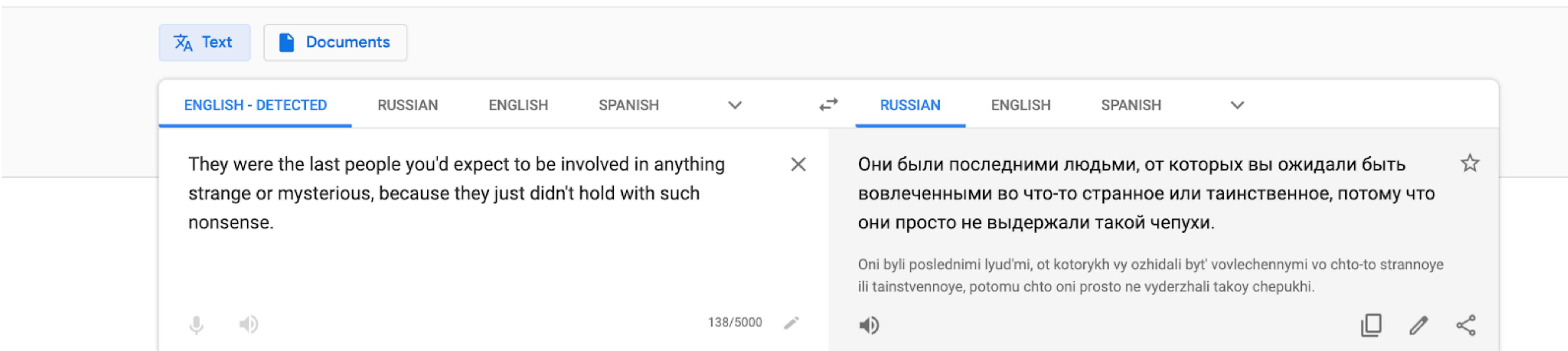
ENGLISH - DETECTED RUSSIAN ENGLISH SPANISH RUSSIAN ENGLISH SPANISH

They were the last people you'd expect to be involved in anything strange or mysterious, because they just didn't hold with such nonsense.

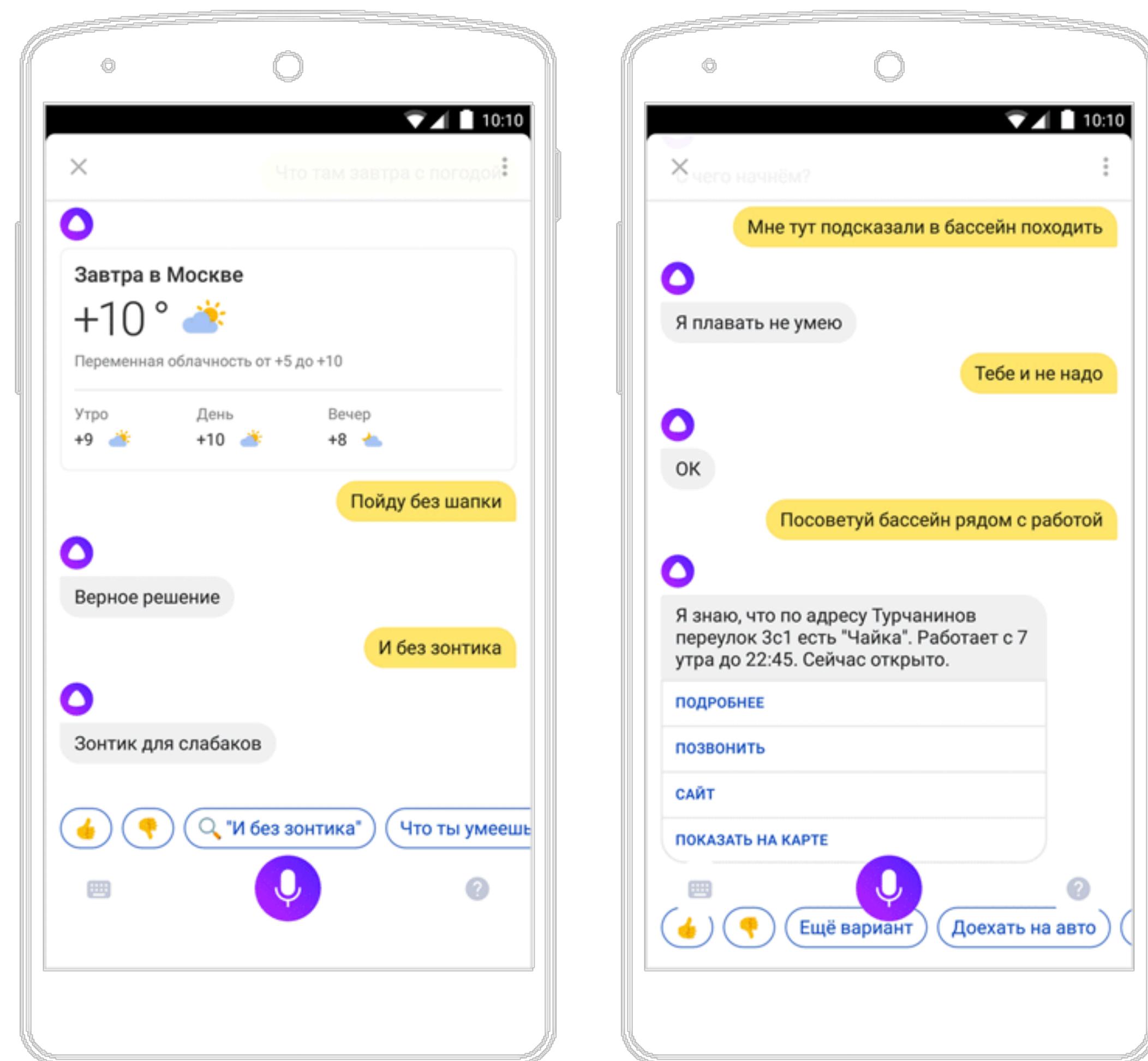
Они были последними людьми, от которых вы ожидали быть вовлеченными во что-то странное или таинственное, потому что они просто не выдержали такой чепухи.

Oni byli poslednimi lyud'mi, ot kotorikh vy ozhidali byt' vovlechennymi vo chto-to strannoye ili tainstvennoye, potomu chto oni prosto ne vyderzhali takoy chepukhi.

Send feedback



ML: приложения



Анализ тональности текста

- Какой эмоциональный окрас имеет текст?
- Варианты: позитивный, нейтральный, негативный
- Применение: автоматический анализ отзывов от пользователей

Анализ тональности текста

«Большое спасибо! Сюда по всему, это как раз то, чего не хватает всем зарубежным курсам по Machine Learning и Knowledge Discovery. Это теория, математика, объяснение того, как оно устроено “в кишках”.»

Какой окрас?

Анализ тональности текста

«Я вижу очень большой минус, что курс будет на готовой библиотеке sci-kit. Курс от Andrew лучше тем, что ученик сам пишет алгоритм и видит изнутри, как он работает.»

Какой окрас?

Анализ тональности текста

- x — текст на русском языке
 - $f(x)$ — его окрас (принимает значения -1, 0, 1)
 - Можно ли выписать формулу для $f(x)$?
-
- На входе — вовсе не числа
 - Точная зависимость может не существовать

Признаковые описания документов

- Обычно в ML данные представляют собой матрицу «объекты-признаки»:

Номер автомобиль	Тип топлива	Мощность двигателя	...	Масса
1	Бензин	120	...	1700
...
N	Дизель	160	...	2100

- Для текстов тоже нужно как-то получить такую матрицу

Модель мешка слов

- Можно проверять наличие всех возможных слов из некоторых словаря:

Номер текста	Содержит «абрикос»	...	Содержит «яблоко»
1	0	...	1
...
N	1	...	0

- Пусть значением признака будет не наличие слова, а число его вхождений в документ («мешок слов»):

Номер текста	Вхождений «абрикос»	...	Вхождений «яблоко»
1	0	...	23
...
N	2	...	0

TF-IDF

- Представление «мешка слов» часто используется при обработке текстов, но частота встречаемости слов не самый информативный признак
- Идея: хотим выделить слова, которые часто встречаются в данном тексте, и редко — в других текстах - используем значения TF-IDF:

$$v_{wd} = tf_{wd} \times \log \frac{N}{df_w}$$

- ▶ tf_{wd} — доля слова w в словах документа d
- ▶ df_w — число документов, содержащих w
- ▶ N — общее число документов

Представления текста

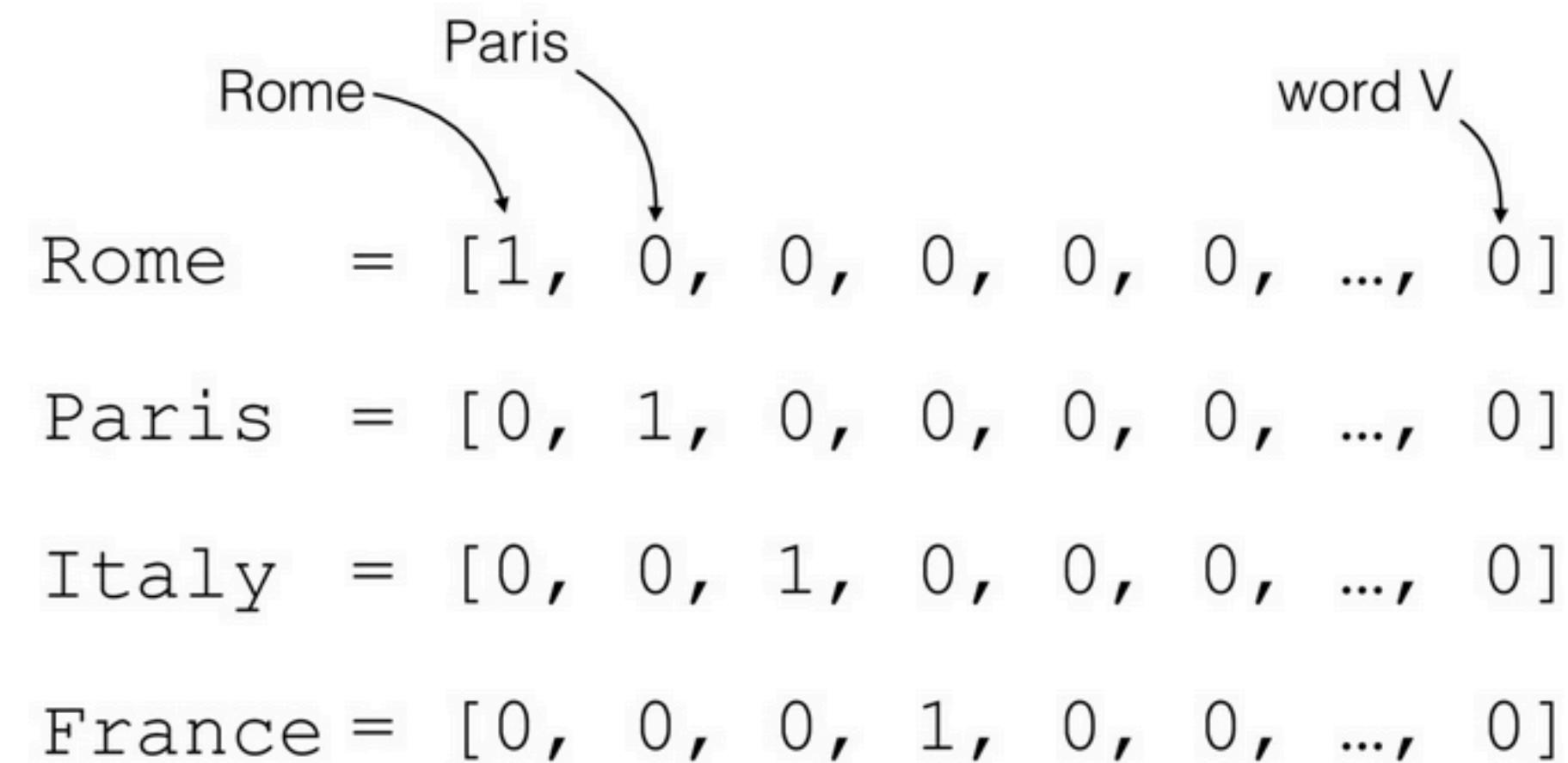
Хотим: перевести слова в числовой вид

Представления текста

Хотим: перевести слова в числовой вид

One-hot encoding

Создаем словарь всех слов (размера V) и нумеруем их



Представления текста

Хотим: перевести слова в числовой вид

One-hot encoding

Создаем словарь всех слов (размера V) и нумеруем их

The diagram illustrates the one-hot encoding of four words: Rome, Paris, Italy, and France. Above each word, an arrow points to its corresponding index in a vector. The word 'Rome' is at index 1, 'Paris' is at index 2, 'Italy' is at index 3, and 'France' is at index 4. To the right, an arrow labeled 'word V' points to the final index of the vector, indicating it represents a word from a vocabulary of size V.

$$\begin{aligned} \text{Rome} &= [1, 0, 0, 0, 0, 0, \dots, 0] \\ \text{Paris} &= [0, 1, 0, 0, 0, 0, \dots, 0] \\ \text{Italy} &= [0, 0, 1, 0, 0, 0, \dots, 0] \\ \text{France} &= [0, 0, 0, 1, 0, 0, \dots, 0] \end{aligned}$$

Все вектора ортогональны друг другу $\cos_{\text{sim}}(w_1, w_2) = \frac{w_1^T w_2}{\|w_1\|^2 \|w_2\|^2} = 0$

Представления текста

Хотим: перевести слова в числовой вид

One-hot encoding

Создаем словарь всех слов (размера V) и нумеруем их

Ортогональные вектора — это такие вектора, угол между которыми равен 90 градусов, что значит, что их скалярное произведение равно нулю.

Rome	=	[1, 0, 0, 0, 0, 0, ..., 0]
Paris	=	[0, 1, 0, 0, 0, 0, ..., 0]
Italy	=	[0, 0, 1, 0, 0, 0, ..., 0]
France	=	[0, 0, 0, 1, 0, 0, ..., 0]

Все вектора ортогональны друг другу

$$\cos_{\text{sim}}(w_1, w_2) = \frac{w_1^T w_2}{\|w_1\|^2 \|w_2\|^2} = 0$$

Косинусное сходство (cosine similarity) — это мера, используемая для вычисления схожести между двумя векторами в многомерном пространстве. Оно определяется как косинус угла между векторами (числитель — скалярное произведение векторов, знаменатель — нормы (длина) векторов). Основное преимущество косинусного сходства в том, что оно позволяет сравнивать векторы, игнорируя их длину, и фокусируется исключительно на их направлениях.

Неправильная интерпретация: One-hot encoding не учитывает семантические связи между словами (например, "кот" и "собака" рассматриваются как равные, хотя они имеют разное значение).

Представления текста

Хотим: перевести слова в числовой вид

Идея: вектора слов, которые встречаются в одном контексте, должны быть близки $\text{cos_sim}(w_1, w_2) \approx w_1^T w_2$

_____ *is the most beautiful capital city in Europe*

Rome? Paris?

Представления текста

Хотим: перевести слова в числовой вид

Идея: вектора слов, которые встречаются в одном контексте, должны быть близки $\cos_{\text{sim}}(w_1, w_2) \approx w_1^T w_2$

$$\mathbf{expect} = \begin{pmatrix} 0.286 \\ 0.792 \\ -0.177 \\ -0.107 \\ 0.109 \\ -0.542 \\ 0.349 \\ 0.271 \\ 0.487 \end{pmatrix}$$



Больше сложных задач!

- Какой будет спрос на товар в следующем месяце?
- Сколько денег заработает магазин за год?
- Вернет ли клиент кредит?
- Заболеет ли пациент раком?
- Сдаст ли студент следующую сессию?
- На фотографии гуманитарий или технарь?
- Кто выиграет битву в онлайн-игре?

Больше сложных задач!

- Везде — очень сложные неявные зависимости
- Нельзя выразить их формулой
- Но есть некоторое число примеров
 - Тексты с известным окрасом
- Будем приближать зависимости, используя примеры

Анализ данных и машинное обучение

— это про то, как восстановить сложные зависимости
по конечному числу примеров

Основные термины

Пример задачи

- Сеть ресторанов
- Хотим открыть еще один
- Несколько вариантов размещения
- Какой из вариантов принесет максимальную прибыль?

* см. kaggle.com, TFI Restaurant Revenue Prediction

Обозначения

- x — объект, sample — для чего хотим делать предсказания
 - Конкретное расположение ресторана
- \mathbb{X} — пространство всех возможных объектов
 - Все возможные расположения ресторанов
- y — ответ, целевая переменная, target — что предсказываем
 - Прибыль в течение первого года работы
- \mathbb{Y} — пространство ответов — все возможные значения ответа
 - Все вещественные числа

Обучающая выборка

- Мы ничего не понимаем в экономике
- Зато имеем много объектов с известными ответами
- $X = (x_i, y_i)_{i=1}^{\ell}$ — обучающая выборка
- ℓ — размер выборки

Признаки

- Объекты — абстрактные сущности
- Компьютеры работают только с числами
- Признаки, факторы, features — числовые характеристики объектов
- d — количество признаков
- $x = (x^1, \dots, x^d)$ — признаковое описание

Признаки

- Объекты — абстрактные сущности
- Компьютеры работают только с числами
- Признаки, факторы, features — числовые характеристики объектов
- d — количество признаков
- $x = (x_1, \dots, x_d)$ — признаковое описание



Вектор

Признаки

- Объекты — абстрактные сущности
- Компьютеры работают только с числами
- Признаки, факторы, features — числовые характеристики объектов
- d — количество признаков
- $x = (x_1, \dots, x_d)$ — признаковое описание



Признаки

- Про демографию:
 - Средний возраст жителей ближайших кварталов
 - Динамика количества жителей
- Про недвижимость:
 - Средняя стоимость квадратного метра жилья поблизости
 - Количество школ, банков, магазинов, заправок
 - Расстояние до ближайшего конкурента
- Про дороги:
 - Среднее количество машин, проезжающих мимо за день

Алгоритм

- $a(x)$ — алгоритм, модель — функция, предсказывающая ответ для любого объекта
- Отображает \mathbb{X} в \mathbb{Y}
- Линейная модель: $a(x) = w_0 + w_1x_1 + \dots + w_dx_d$
- Например:

$$a(x) = 1.000.000 + 100.000 * (\text{расстояние до конкурента}) - 100.000 * (\text{расстояние до метро})$$

ФУНКЦИЯ ПОТЕРЬ

- Не все алгоритмы полезны
- $a(x) = 0$ — не принесет никакой выгоды
- Функция потерь — мера корректности ответа алгоритма
- Предсказали \$10000 прибыли, на самом деле \$5000 — хорошо или плохо?
- Квадратичное отклонение: $(a(x) - y)^2$

Функционал качества

- Функционал качества, метрика качества — мера качества работы алгоритма на выборке
- Среднеквадратичная ошибка (Mean Squared Error, MSE):

$$\frac{1}{\ell} \sum_{i=1}^{\ell} (a(x_i) - y_i)^2$$

- Чем меньше, тем лучше

Функционал качества

- Должен соответствовать бизнес-требованиям
- Одна из самых важных составляющих анализа данных

Обучение алгоритма

- Есть обучающая выборка и функционал качества
- Семейство алгоритмов \mathcal{A}
 - Из чего выбираем алгоритм
 - Пример: все линейные модели
 - $\mathcal{A} = \{w_0 + w_1x_1 + \dots + w_dx_d \mid w_0, w_1, \dots, w_d \in \mathbb{R}\}$
- Обучение: поиск оптимального алгоритма с точки зрения функционала качества

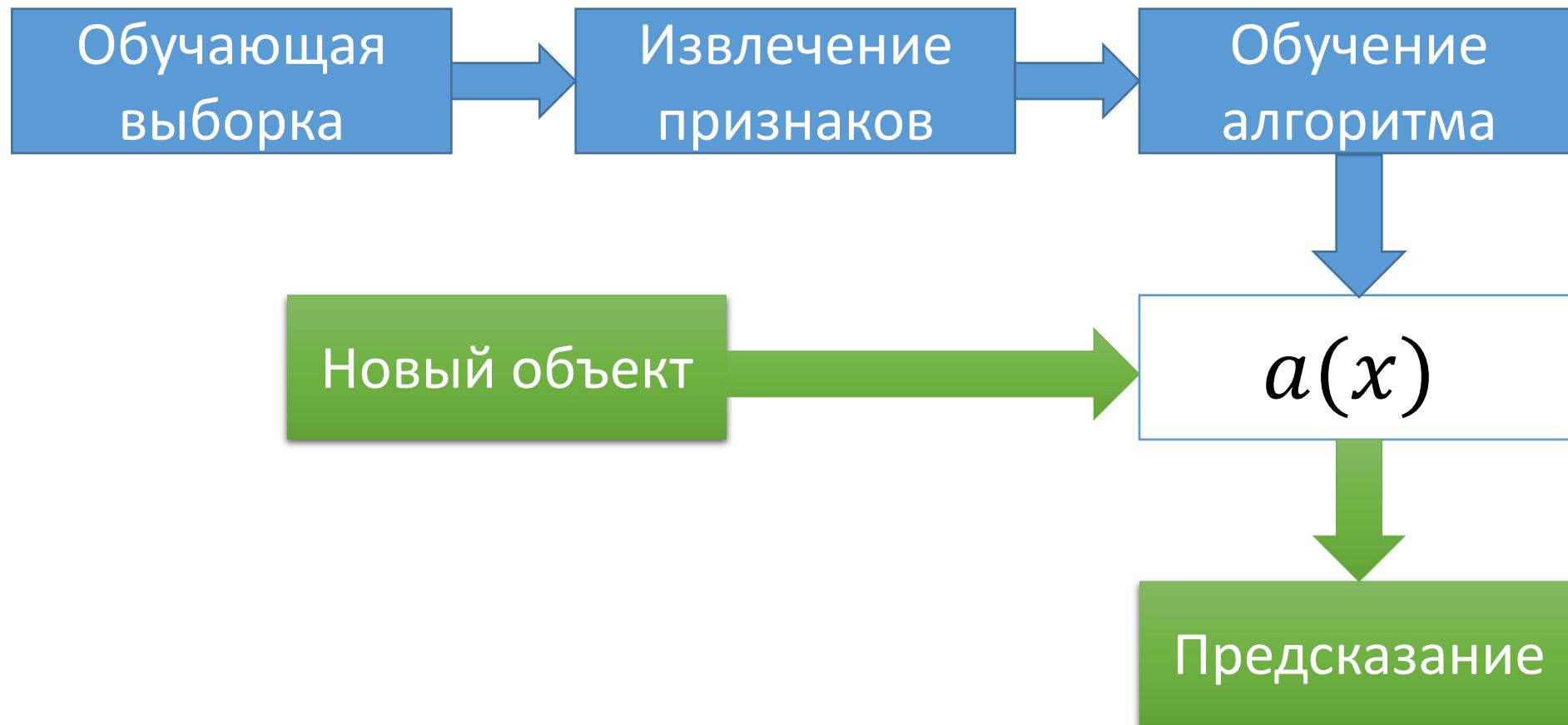
Что нужно знать

1. Как сформулировать задачу?
2. Какие признаки использовать?
3. Откуда взять обучающую выборку?
4. Как выбрать метрику качества?
5. Как обучить алгоритм?
6. Как оценить качество алгоритма?

Машинное обучение

- Не все задачи имеют такую формулировку!
- Обучение без учителя
- Обучение с подкреплением
- И т.д.

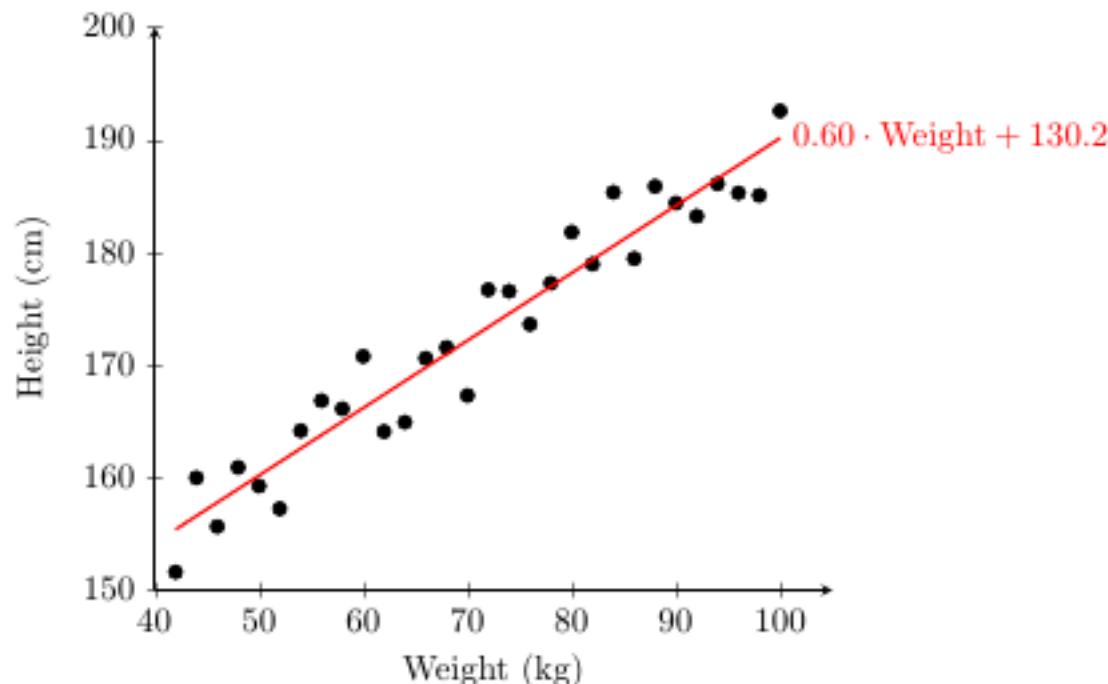
Машинное обучение



Типы ответов

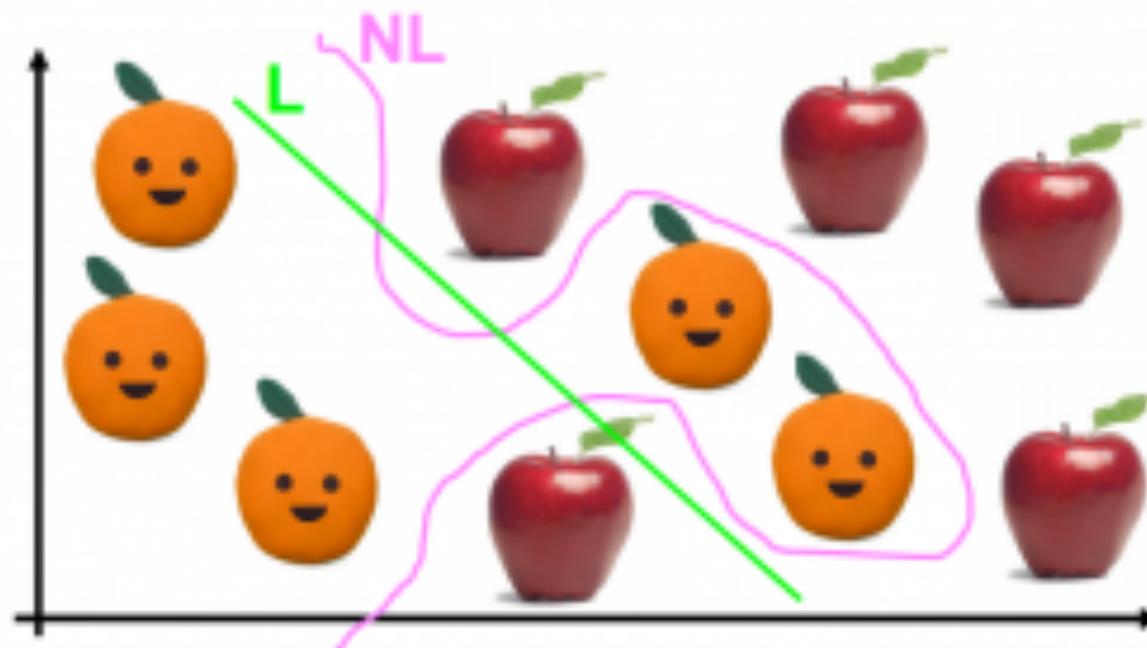
Регрессия

- Вещественные ответы: $\mathbb{Y} = \mathbb{R}$
- (вещественные числа — числа с любой дробной частью)
- Пример: предсказание роста по весу



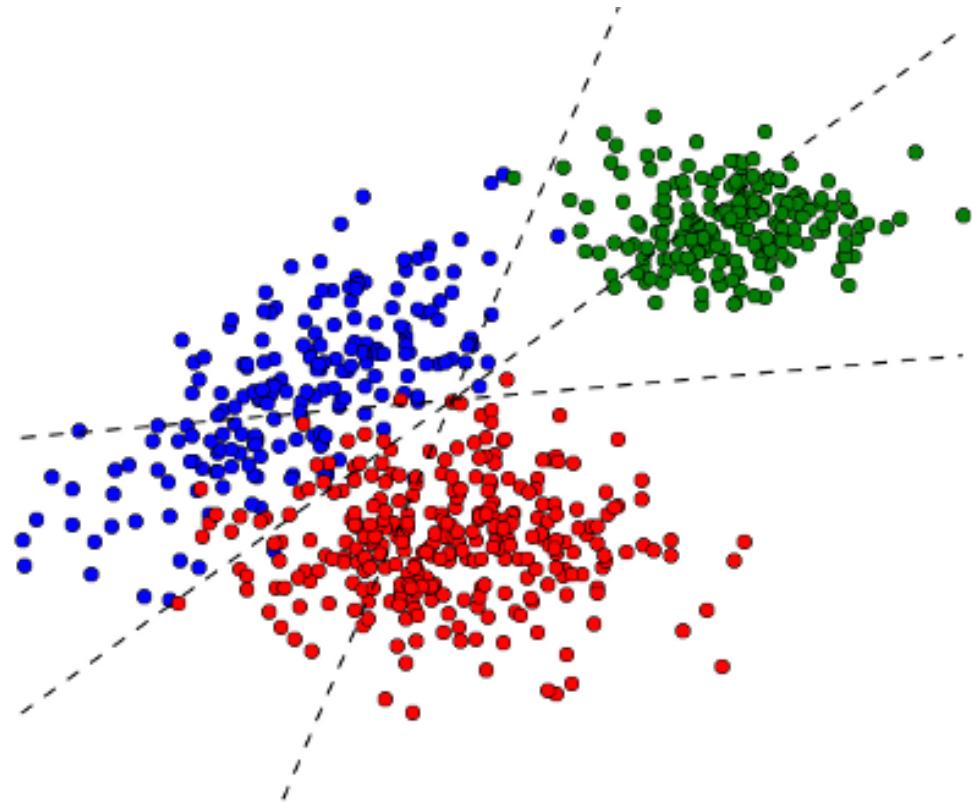
Классификация

- Конечное число ответов: $|\mathbb{Y}| < \infty$
- Бинарная классификация: $\mathbb{Y} = \{-1, +1\}$



Классификация

- Многоклассовая классификация: $\mathbb{Y} = \{1, 2, \dots, K\}$



Классификация

- Классификация с пересекающимися классами: $\mathbb{Y} = \{0, 1\}^K$
 - (multi-label classification)
- Ответ — набор из K нулей и единиц
- i -й элемент ответа — принадлежит ли объект i -му классу
- Какие темы присутствуют в статье?
- (математика, биология, экономика)

Ранжирование

- Набор документов d_1, \dots, d_n
- Запрос q
- Задача: отсортировать документы по *релевантности* запросу
- $a(q, d)$ — оценка релевантности

Ранжирование

Яндекс

картинки с котиками — 5 млн ответов



Найти

Поиск

[Картинки с кошками | Fun Cats — Забавные коты](#)

[funcats.by > pictures/](#) ▾

Картинки с кошками. Прикольные коты. 777 изображений. ... 32 изображения. Кошки Стамбула. 41 изображение. Веселые котята.

Картинки

Видео

[Уморные котики \(57 фото\) » Бяки.нет | Картинки](#)

[byaki.net > Картинки > 14026-umornye-kotiki-57...](#) ▾

Бяки нет! . NET. Уморные котики (57 фото). 223. Коментариев:9Автор:4ertonok Просмотров:161 395 Картинки28-10-2008, 00:03.

Карты

Маркет

Ещё

[Смешные картинки кошек с надписями | Лолкот.Ру](#)

[lolkot.ru](#) ▾

Смешные картинки для новых приколов! Сделать свой прикол очень просто. ... Котик верит в чудеса. Он в носке подарок ищет...

[Красивые картинки и фото кошек, котят и котов](#)

[foto-zverey.ru > Кошки](#) ▾

Фото и картинки кошек и котят потрясающей красоты и нежности. Здесь мы собрали такие изображения, которые всегда вызывают море положительных эмоций...

[Обои для рабочего стола Котята | картинки на стол Котята](#)

[7fon.ru > Чёрные обои и картинки > Обои котята](#) ▾

Картинки Котята с 1 по 15. Обои для рабочего стола Котята. ... Скачать Картинки Котята на рабочий стол бесплатно.

Кластеризация

- \mathbb{Y} — отсутствует
- Нужно найти группы похожих объектов
- Сколько таких групп?
- Как измерить качество?
- Пример: сегментация пользователей мобильного оператора

Метод k-ближайших соседей (KNN)

Представим, что мы проводим классификацию объектов на два класса — красный или жёлтый. Нам дана некоторая обучающая выборка и целевой объект (серый):

Хотим предсказать класс



$K=3$

Среди трёх
ближайших соседей:
— 0 класса x
— 3 класса ○

Предсказываем
класс ○

kNN: обучение

- Дано: обучающая выборка $X = (x_i, y_i)_{i=1}^\ell$
- Задача классификация (ответы из множества $\mathbb{Y} = \{1, \dots, K\}$)
- Обучение модели:
 - Запоминаем обучающую выборку X

kNN: применение

Дано: новый объект x

Применение модели:

- Сортируем объекты обучающей выборки по расстоянию до нового объекта:
 $\rho(x, x_{(1)}) \leq \rho(x, x_{(2)}) \leq \dots \leq \rho(x, x_{(\ell)})$
- Выбираем k ближайших объектов: $x_{(1)}, \dots, x_{(k)}$
- Выдаём наиболее популярный среди них класс:

$$a(x) = \arg \max_{y \in \mathbb{Y}} \sum_{i=1}^k [y_{(i)} = y]$$

kNN: применение

Дано: новый объект x

Применение модели:

- Сортируем объекты обучающей выборки по расстоянию до нового объекта:
 $\rho(x, x_{(1)}) \leq \rho(x, x_{(2)}) \leq \dots \leq \rho(x, x_{(\ell)})$
- Выбираем k ближайших объектов: $x_{(1)}, \dots, x_{(k)}$
- Выдаём наиболее популярный среди них класс:

$$a(x) = \arg \max_{y \in \mathbb{Y}} \sum_{i=1}^k [y_{(i)} = y]$$

kNN: применение

Дано: новый объект x

Применение модели:

- Сортируем объекты обучающей выборки по расстоянию до нового объекта:
 $\rho(x, x_{(1)}) \leq \rho(x, x_{(2)}) \leq \dots \leq \rho(x, x_{(\ell)})$
- Выбираем k ближайших объектов: $x_{(1)}, \dots, x_{(k)}$
- Выдаём наиболее популярный среди них класс:

$$a(x) = \arg \max_{y \in \mathbb{Y}} \sum_{i=1}^k [y_{(i)} = y]$$

kNN: применение

Дано: новый объект x

Применение модели:

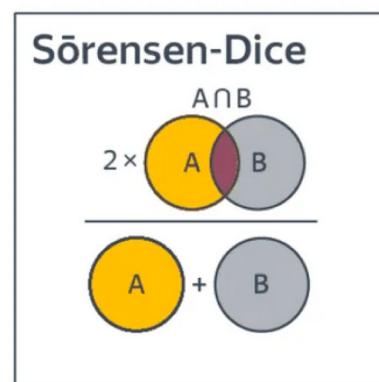
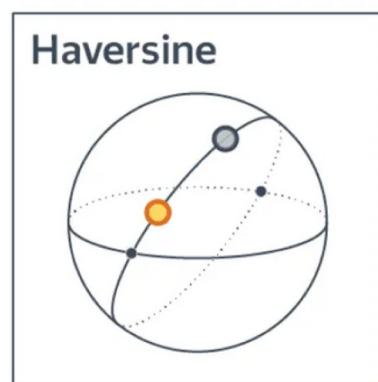
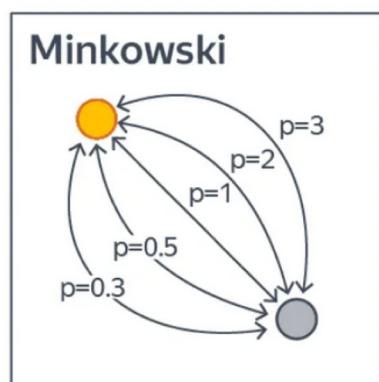
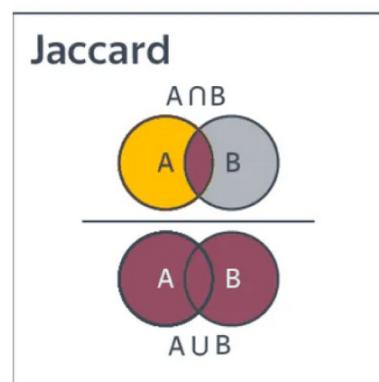
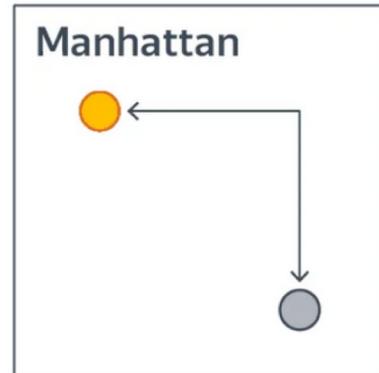
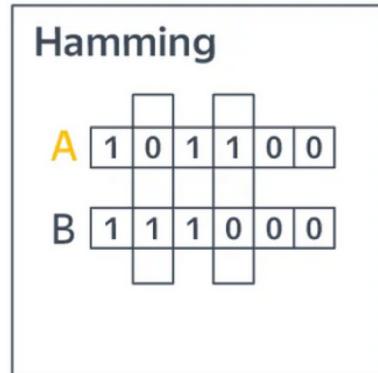
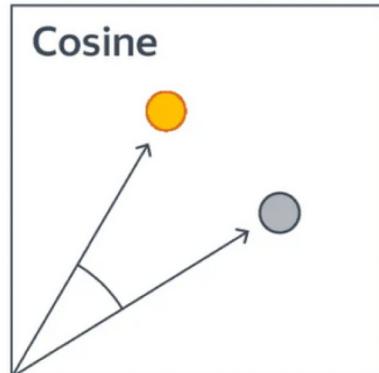
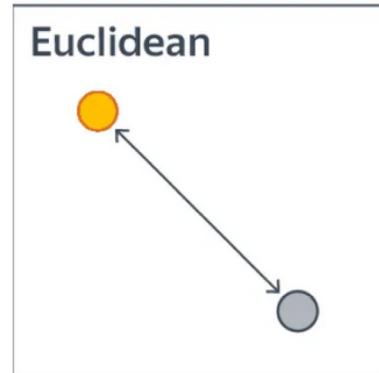
- Сортируем объекты обучающей выборки по расстоянию до нового объекта:
 $\rho(x, x_{(1)}) \leq \rho(x, x_{(2)}) \leq \dots \leq \rho(x, x_{(\ell)})$
- Выбираем k ближайших объектов: $x_{(1)}, \dots, x_{(k)}$
- Выдаём наиболее популярный среди них класс:

$$a(x) = \arg \max_{y \in \mathbb{Y}} \sum_{i=1}^k [y_{(i)} = y]$$

Выбор метрики

Может возникнуть закономерный вопрос, как же правильно выбрать функцию расстояния ρ . В

подавляющем большинстве случаев обычное евклидово расстояние $\rho(x, y) = \sqrt{\sum_i(x_i - y_i)^2}$ будет хорошим выбором. Однако в некоторых случаях другие функции будут подходить лучше, поэтому давайте разберём ещё несколько функций, наиболее используемых на практике.



Манхэттенская метрика

$$\rho(x, y) = \sum_i |x_i - y_i|$$

Часто используется в высокоразмерных пространствах из-за лучшей устойчивости к выбросам.

Представим, что два объекта в 1000-размерном пространстве почти идентичны, но сильно отличаются по одному из признаков. Это почти наверняка свидетельствует о выбросе в этом признаке, и объекты, скорее всего, очень близки. Однако евклидово расстояние усилит различие в единственном признаке и сделает их более далёкими друг от друга. Этого недостатка лишена манхэттенская метрика — в ней вместо квадрата используется модуль.

Метрика Минковского

$$\rho(x, y) = \left(\sum_i |x_i - y_i|^p \right)^{1/p}$$

Является обобщением евклидовой ($p = 2$) и манхэттенской ($p = 1$) метрик.

Косинусное расстояние

$$\rho(x, y) = 1 - \cos \theta = 1 - \frac{x \cdot y}{\|x\| \|y\|}$$

Эта метрика хороша тем, что не зависит от норм векторов. Такое поведение бывает полезно в некоторых задачах, например при поиске похожих документов. В качестве признаков там часто используются количества слов. При этом интуитивно кажется, что если в тексте использовать каждое слово в два раза больше, то тема этого текста поменяться не должна. Поэтому как раз в этом случае нам не важна норма вектор-признака, и в задачах, связанных с текстами, часто применяется именно косинусное расстояние.

kNN: применение

