ДМИТРИЙ ВОЛЬФ : : 7 МАЯ 2022 ГОДА

Как создать интеллектуальный чатбот в «телеге» за один урок

АВТОР ЛЕКЦИИ

Дмитрий Вольф

Data Scientist,
Assistant of SPBU

YADRO

YADRO — ГРУППА ИНЖЕНЕРНЫХ ТЕХНОЛОГИЧЕСКИХ КОМПАНИЙ

Современные конкурентоспособные рыночные продукты

- Серверы VESNIN и VEGMAN
- Системы хранения данных TATLIN
- Сетевое и коммутационное оборудование KORNEFELD
- Базовые станции
- Клиентские устройства KVADRA

Полный цикл продуктовой разработки и производства внутри группы

Разработка полупроводни-кового IP → Разработка процессоров → Аппаратная разработка → Программная разработка → Промышленный дизайн → Разработка конечных систем → Логистика и цепочка поставок → Производство полного цикла → Сборка и монтаж → Продажи и сервис

Совокупный технологический фундамент и международная кооперация

open invention network · RISC-V · Storage Performance Council · SNIA · OpenPOWER · THE LINUX FOUNDATION · PCI SIG · GEN Z · OPEN Compute Project · OPEN MAINFRAME PROJECT · OpenCAPI · DMTF

НАШ ВКЛАД В РАЗВИТИЕ КАДРОВ

Для школьников и младших студентов

Участвуем в организации мастер-классов по проектированию микропроцессоров в Школе Цифрового Синтеза

Для студентов

Сотрудничаем с вузами в организации учебных программ и проведении исследовательских работ

Лучшим студентам предоставляем возможность стажировки

Для специалистов

Публикуем статьи и видеолекции ведущих практикующих инженеров и ученых на портале «Истовый инженер»

engineer.yadro.com

По всем образовательным вопросам пишите нам: edu@yadro.com

ПЛАН ЛЕКЦИИ

I. Что такое искусственный интеллект:

- Примеры искусственного интеллекта в жизни
- Рождение искусственного интеллекта
- Возможности искусственного интеллекта
- Машинное обучение. Наука о данных
- Машинное обучение. Базовые подходы
- Результаты машинного обучения
- Прогнозы на будущее искусственного интеллекта

II. Как сделать свой ML-чат-бот:

- Разновидности
- Архитектура
- Пример работы
- Реализация
  - Предобработка текста
  - Векторизация текста
    - OHE
    - WOF
    - TF-IDF
  - Уменьшение размерности
  - Метод k-ближайших соседей
- Telegram API

ПЛАН ЛЕКЦИИ

I. Что такое искусственный интеллект:

- Примеры искусственного интеллекта в жизни
- Рождение искусственного интеллекта
- Возможности искусственного интеллекта
- Машинное обучение. Наука о данных
- Машинное обучение. Базовые подходы
- Результаты машинного обучения
- Прогнозы на будущее искусственного интеллекта

II. Как сделать свой ML-чат-бот:

- Разновидности
- Архитектура
- Пример работы
- Реализация
  - Предобработка текста
  - Векторизация текста
    - OHE
    - WOF
    - TF-IDF
  - Уменьшение размерности
  - Метод k-ближайших соседей
- Telegram API

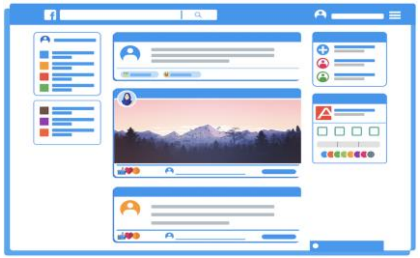
ПРИМЕРЫ ИИ В РЕАЛЬНОЙ ЖИЗНИ

Поиск/переводчик Google, голосовой ассистент

Аудио- и видеостриминговые платформы

Боты для видеоигр

Умная лента, рекомендации в соцсетях



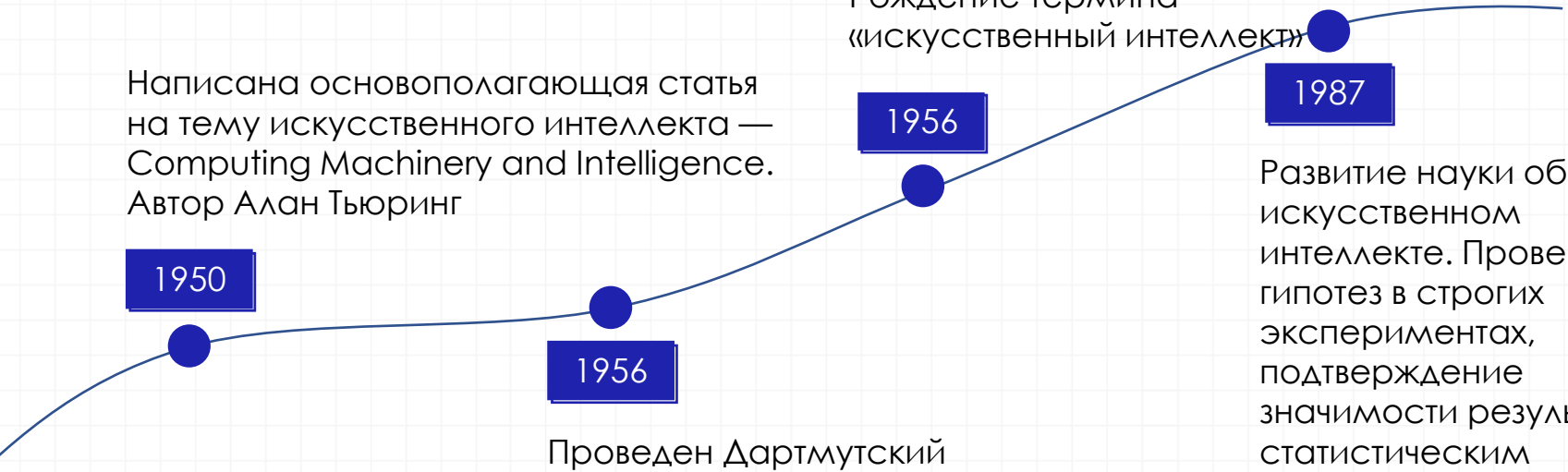
Безопасность

Роботы

Автопилоты, дроны

Умные девайсы

РОЖДЕНИЕ ИСКУССТВЕННОГО ИНТЕЛЛЕКТА (ИИ)



Сильный ИИ

Обладает сознанием, мыслит, **превосходит** все когнитивные способности **человека**

Слабый ИИ

Фокусируется на выполнении конкретной задачи, воспроизводит некоторые аспекты человеческого познания, **полагается на** вмешательство **человека** для определения параметров алгоритмов и предоставления исторических данных

Машинное обучение — это процесс использования математических моделей для определения закономерностей в исторических данных; обучение компьютера без прямых инструкций

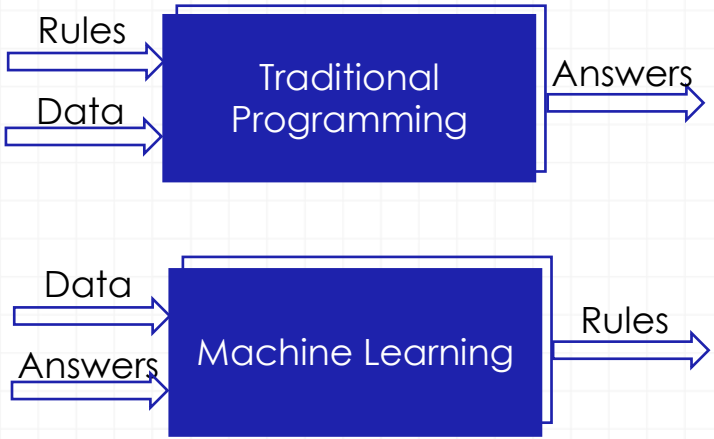
Машинное обучение — это подмножество ИИ

Придуман термин «машинное обучение», или «самообучающийся компьютер»

1959

1960

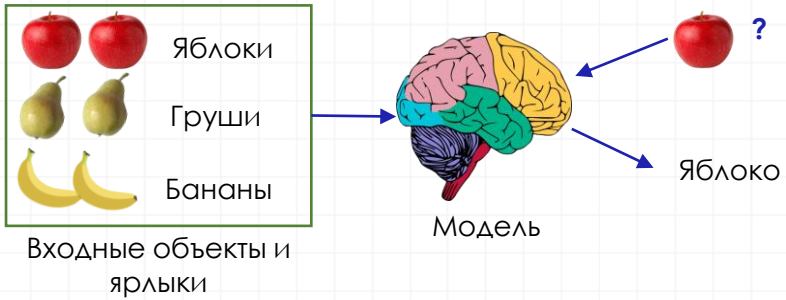
Написана книга Learning Machines, посвященная алгоритмам распознавания образов



С учителем

Алгоритму обучения представлены только примеры объектов. Ярлыки для них неизвестны

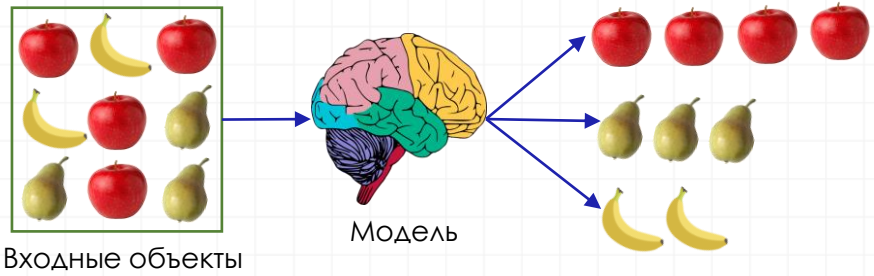
Цель модели — самостоятельно найти скрытую закономерность в объектах



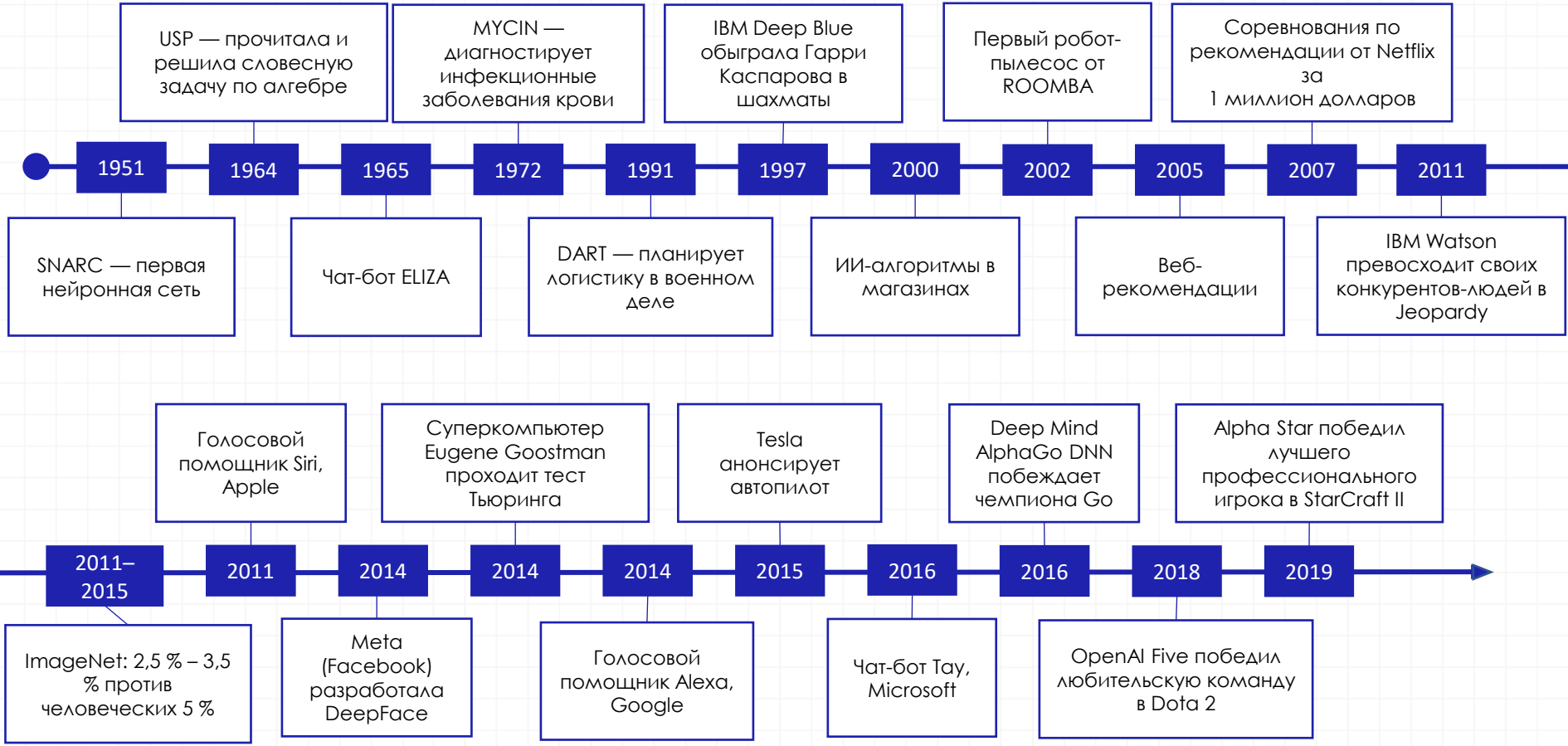
Без учителя

Алгоритму обучения представлены примеры объектов и их ярлыки

Цель модели — найти общее правило, которое на каждом объекте сможет сопоставить корректный ярлык



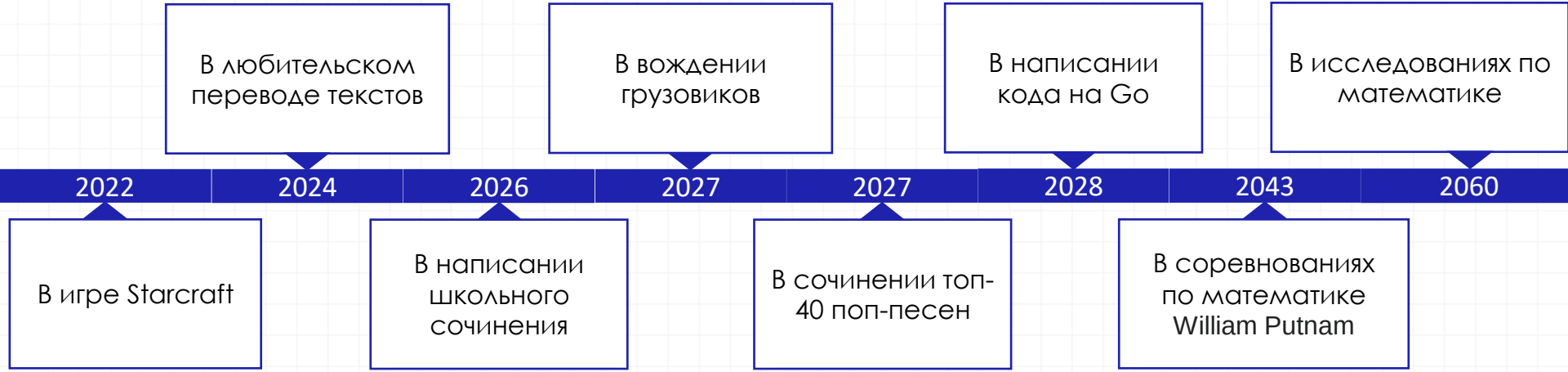
ДОСТИЖЕНИЯ МАШИННОГО ОБУЧЕНИЯ



Когда и в чем ИИ превзойдет человека. Прогноз на будущее

С вероятностью 50 % ИИ превзойдет человека во всех задачах через 45 лет и автоматизирует все вакансии через 120 лет

[When Will AI Exceed Human Performance? Evidence from AI Experts]



ПЛАН ЛЕКЦИИ

I. Что такое искусственный интеллект:

- Примеры искусственного интеллекта в жизни
- Рождение искусственного интеллекта
- Возможности искусственного интеллекта
- Машинное обучение. Наука о данных
- Машинное обучение. Базовые подходы
- Результаты машинного обучения
- Прогнозы на будущее искусственного интеллекта

II. Как сделать свой ML-чат-бот:

- Разновидности
- Архитектура
- Пример работы
- Реализация
  - Предобработка текста
  - Векторизация текста
    - OHE
    - WOF
    - TF-IDF
  - Уменьшение размерности
  - Метод k-ближайших соседей
- Telegram API

КЛАССИФИКАЦИЯ ЧАТБОТОВ

Rule-based (по правилам)

Выдают заранее **подготовленные ответы** на определенные запросы пользователя

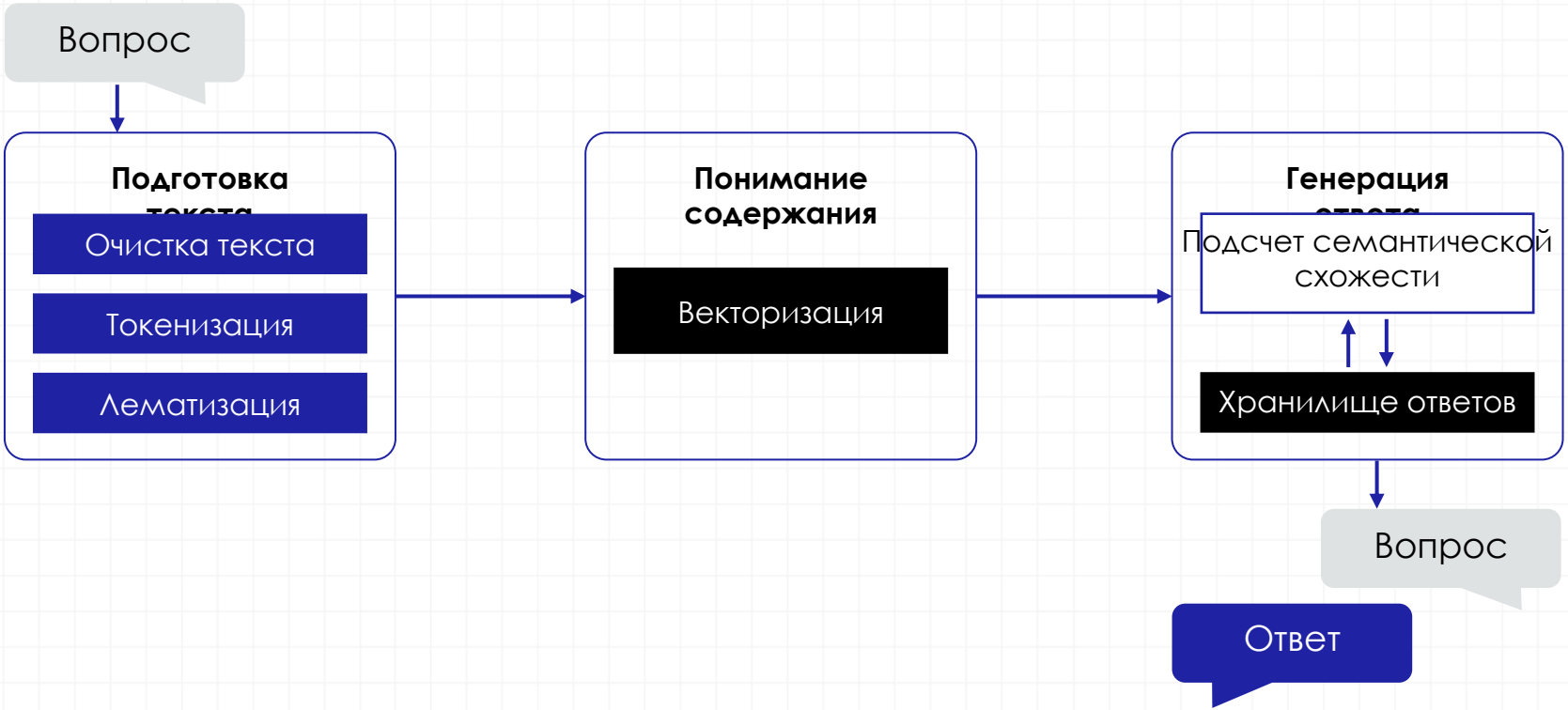
Retrieval-based (основанные на поиске)

Предоставляют наилучший возможный ответ из базы данных предопределенных ответов, не генерируют новые выходные данные

Generative (порождающие)

Генерируют новые ответы на основе больших объемов данных из разговорной речи

АРХИТЕКТУРА RETRIEVAL-BASED-ЧАТ-БОТОВ



I. Отправляем вопрос
II. Вопрос разбивается на слова (токены), убираются окончания, выделяются корни слов, удаляются знаки препинания и стоп-слова
III. Каждый токен представляется в виде чисел (эмбединга) — процесс векторизация текста
IV. Из подготовленных заранее ответов выбирается ближайший к вопросу в векторном пространстве

Ты сделал домашку по матеше?

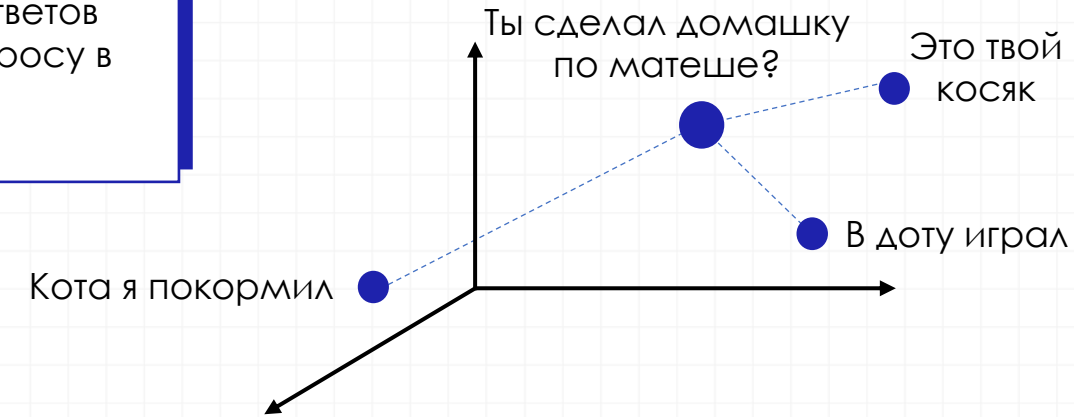
↓

['дел', 'дом', 'матем']

↓

[[1.96579050e-02 4.99110693e-03 6.86838150e-02
1.59690537e-02 2.86113465e-03 -1.19095763e-02]]

↓



RETRIEVAL-BASED CHATBOTS. Data preprocessing

1. Проводим все слова в нижний регистр

```
sentence = sentence.lower()
```

```
"We can build a much brighter future where humans are relieved of menial work using AI capabilities."
```

```
we can build a much brighter future where humans are relieved of menial work using ai capabilities.
```

2. Очищаем данные: удаляем все ненужные символы (знаки пунктуации, специальные

```
tokenizer = RegexpTokenizer(r'\w+')
sentence = ' '.join(tokenizer.tokenize(sentence))
```

```
we can build a much brighter future where humans are relieved of menial work using ai capabilities
```

3. Токенизация (Tokenization)

```
words = word_tokenize(sentence)
```

```
['we', 'can', 'build', 'a', 'much', 'brighter', 'future', 'where', 'humans', 'are', 'relieved', 'of', 'menial', 'work', 'using', 'ai', 'capabilities']
```

4. Удаляем стоп-слова

```
stop_words = set(stopwords.words('russian'))
word_tokens = word_tokenize(sentence)
filtered_sentence = [w for w in word_tokens if not w in stop_words]
```

```
['build', 'much', 'brighter', 'future', 'humans', 'relieved', 'menial', 'work', 'using', 'ai', 'capabilities']
```

5.1 Lemmatization

```
ps = PorterStemmer()
result = []
for word in sentence.split():
    result.append(ps.stem(word))
```

```
['build', 'much', 'brighter', 'futur', 'human', 'reliev', 'menial', 'work', 'use', 'ai', 'capabl']
```

5.2 Stemming

```
lemmatizer = WordNetLemmatizer()
result = []
for word in sentence.split():
    result.append(lemmatizer.lemmatize(word, pos='n'))
```

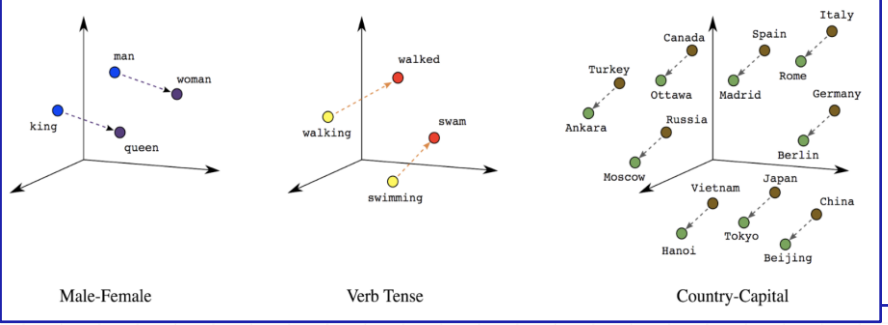
```
['build', 'much', 'brighter', 'future', 'human', 'relieved', 'menial', 'work', 'using', 'ai', 'capability']
```

Свойство векторизации:

«Вложения слов должны представлять токены слов в плотном векторном пространстве, где расположение и расстояние между словами указывают на то, насколько они семантически похожи»

Техники векторизации:

I. One-hot Encoding (OHE)
II. Bag-of-Words (BOW)
III. Term Frequency — Inverse Document Frequency (TF-IDF)
IV. Word2Vec
V. GPT-3
VI. Bert



One-hot Encoding (OHE-вектор) содержит значения для каждого уникального слова
в словаре, устанавливая уникальный токен со значением 1 и 0 в других позициях вектора

Ключевые особенности OHE:

I. Интуитивно понятный
II. N-мерное пространство, N — размер словарной памяти
III. Неэффективное использования памяти
IV. Векторы не передают контекста

Словарь:

[алгебра, русский, биология, химия, физика]

↓

алгебра = [1, 0, 0, 0, 0]

русский = [0, 1, 0, 0, 0]

биология = [0, 0, 1, 0, 0]

химия = [0, 0, 0, 1, 0]

физика = [0, 0, 0, 0, 1]

Значения вектора Bag-of-Words (BOW) представляют частоту, с которой каждое слово встречается в тексте

Ключевые особенности BOW:

I. Интуитивно понятный
II. Не сохраняет порядок слов
III. Работает со всеми словами одинаково
IV. Матрица требует много памяти

Я люблю делать домашку по алгебре, а ты любишь?

↓

1 – алгебра

0 – биология

1 – делать

1 – домашка

2 – любить

↓

[1, 0, 1, 1, 2]

Term Frequency — Inverse Document Frequency (TF-IDF-вектор) учитывает важность слова в зависимости от частоты его использования в документе

Подсчет вектора состоит из трех этапов:

I. Term Frequency — количество раз, когда слово появлялось в документе, деленное на общее количество слов в документе:

$$tf_{ij} = \frac{n_{ij}}{\sum_k n_{ij}}$$

II. Inverse Document Frequency —измеряет, является ли слово распространенным или редким в документе:

$$idf(w) = \log\left(\frac{N}{df_t}\right)$$

III. Финальное значение TF-IDF:

$$W_{ij} = tf_{ij} * idf(w)$$

Ключевые особенности TF-IDF:

I. «Наказывает» часто встречающиеся слова
II. Придает больший вес менее частым или редким словам
III. Требует много памяти для большого набора данных

I. Я люблю делать домашку по алгебре, а ты любишь?
II. А давай вместе решать уравнения по алгебре
III. А я люблю своего кота

TF

	I.	II.	III.
а	1/9	1/6	1/4
алгебра	1/9	1/6	0
любить	2/9	0	1/4
кот	0	0	1/4

*

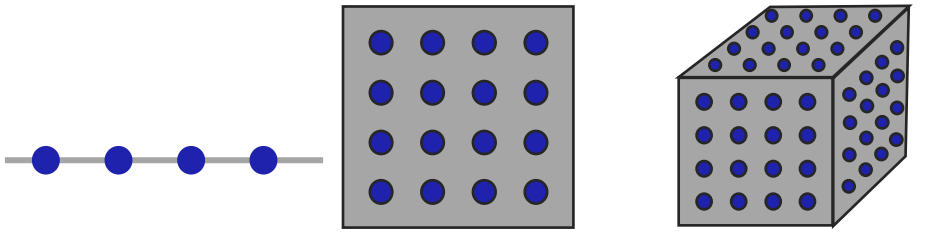
IDF
log(3/3)
log(3/2)
log(3/2)
log(3/1)

«Проклятие размерности» — феномен задач машинного обучения, заключающийся в том, что число возможных комбинаций переменных увеличивается экспоненциально по мере увеличения числа переменных.

Идея решения состоит в том, чтобы уменьшить размерность пространства, а именно спроецировать данные на подпространство меньшей размерности.

Алгоритмы сокращения размерности:

- Principal Component Analysis
- Singular Value Decomposition
- t-Distributed Stochastic Neighbour Embedding
- Uniform Manifold Approximation and Projection (UMAP)
- Linear Discriminant Analysis



1. Стандартизируем данные — приведение признаков (features) к одному масштабу:

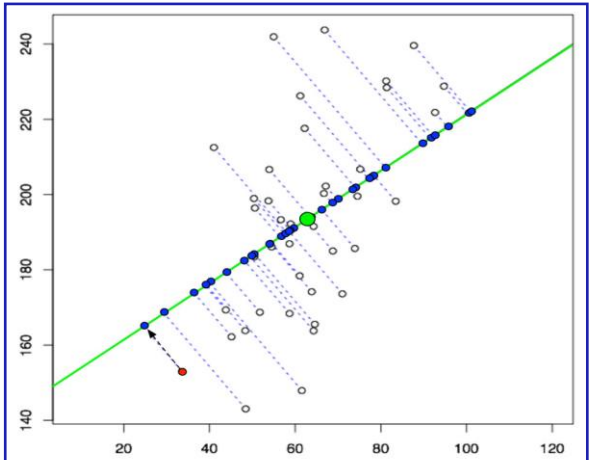
$$X_{new} = \frac{X - X_{min}}{X_{max} - X_{min}};\ X_{new} = \frac{X - X_{mean}}{standart\ deviation}$$

2. Считаем матрицу ковариции:

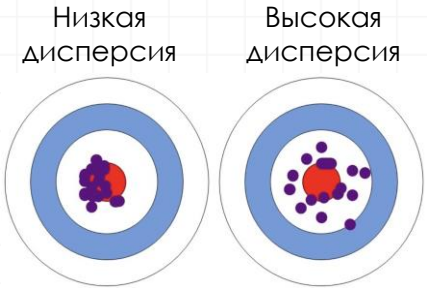
$$\begin{bmatrix} cov(x_1, x_1) & \cdots & cov(x_1, x_n) \\ \vdots & \ddots & \vdots \\ cov(x_n, x_1) & \cdots & cov(x_n, x_n) \end{bmatrix}$$

3. Считаем собственные векторы и собственные значения

4. Строим проекции. Считаем скалярное произведение стандартизированной матрицы и собственного вектора



Проецируем объекты на зеленую ось

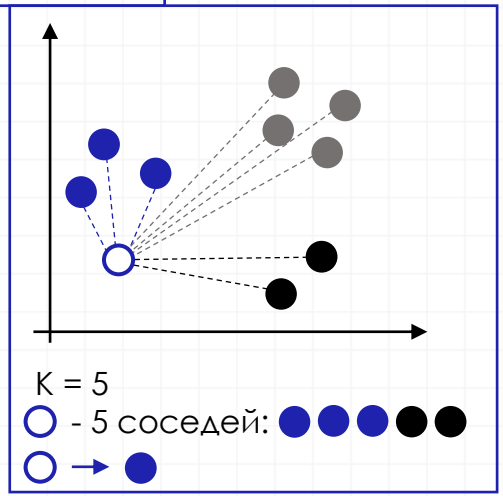


K-ближайших соседей (KNN) — алгоритм обучения с учителем для задач регрессии и классификации:

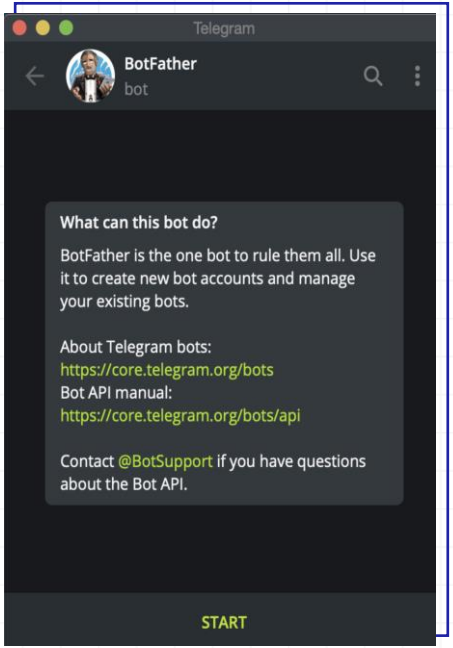
I. Устанавливаем значение K, количество соседей, на которых будем смотреть
II. Для каждого соседа считаем расстояние Евклида
III. Берем K ближайших соседей
IV. Присваиваем тот ярлык текущей точке, который чаще встретился среди ее соседей

Ключевые особенности KNN:

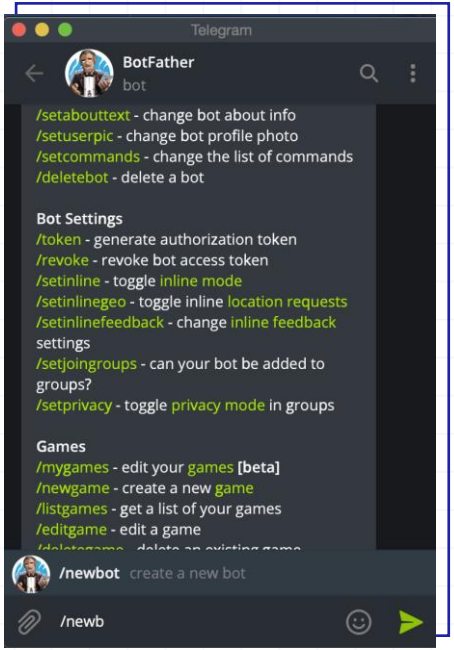
I. Как выбрать оптимальное значение K
II. Для подсчета расстояния можно использовать расстояния Евклида, Манхеттэн
III. Алгоритм ленивого обучения



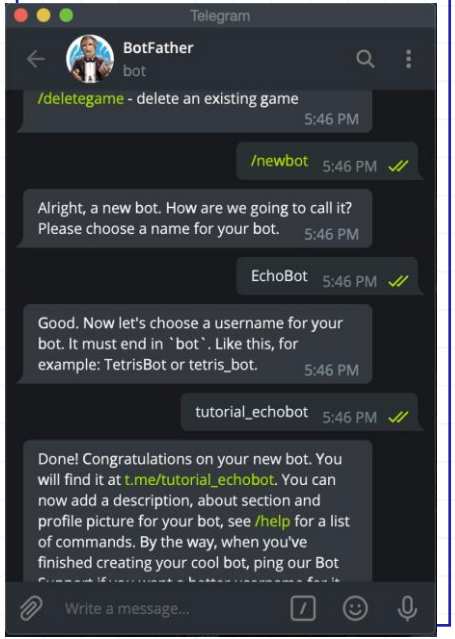
Находим @BotFather и нажимаем «старт»



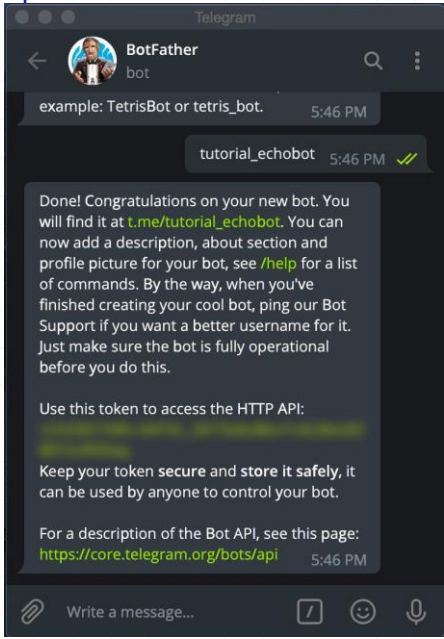
Создаем бот, выполнив команду /newbot



Вводим название бота и имя пользователя для него



Вставляем токен в код



Контакты

telegram: @Volfiik
email: d.volf@yadro.com
Git: https://github.com/answerIII/Chatbot