

Машинное обучение

Ключевые понятия

Искусственный интеллект (ИИ) — это разум, встроенный в машину.

Машинное обучение — это процесс внедрения интеллекта в систему или машину **без явного программирования**.

Глубокое обучение — алгоритмы основаны на биологической структуре и функционировании мозга и призваны наделить машины интеллектом.



Машинное обучение

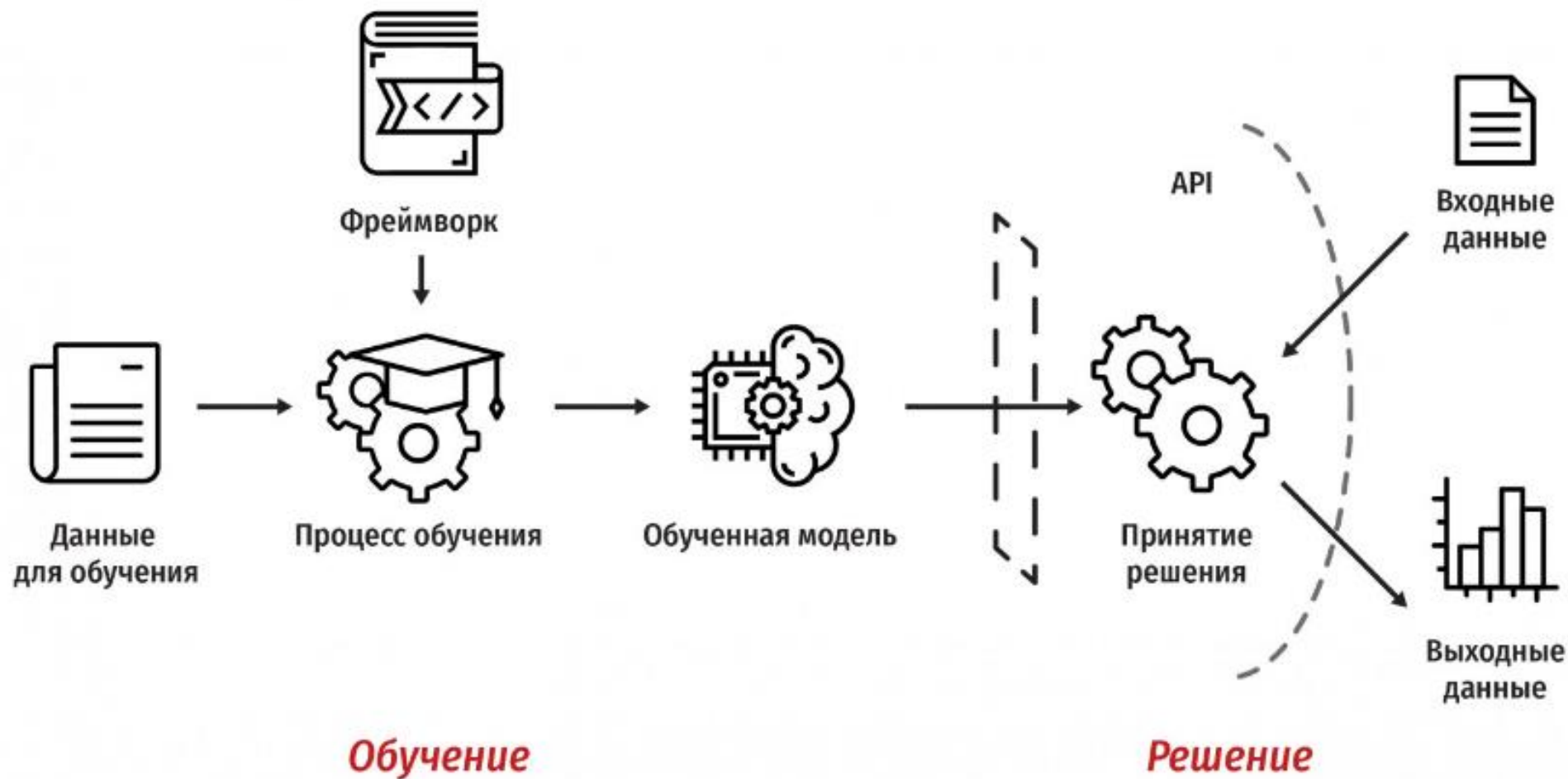
Артур Сэмюэл (Arthur Samuel), американский пионер в области компьютерных игр и искусственного интеллекта, придумал термин «Машинное обучение» в 1959 году, когда работал в IBM.

Способность компьютера с системой ИИ **принимать решения по результатам обработки данных**, не придерживаясь четких схем и правил.

Машина становится **способна к поиску закономерностей** в сложных для человека задачах с большим числом параметров, к обнаружению точных ответов и верному прогнозированию.

В компьютер (или отдельную программу) закладывают **алгоритм самостоятельного нахождения решений** путём комплексного использования статистических данных, из которых выводятся **закономерности** и на основе которых делаются **прогнозы**.

Общий процесс



Типы задач машинного обучения

ЗАДАЧА РЕГРЕССИИ

прогноз на основе выборки объектов с различными признаками: цена квартиры, стоимость ценной бумаги по прошествии полугода, ожидаемый доход магазина на следующий месяц, качество вина при слепом тестировании.

ЗАДАЧА КЛАССИФИКАЦИИ

- получение категориального ответа на основе набора признаков: есть ли на фотографии кот, является ли изображение человеческим лицом, болен ли пациент раком

ЗАДАЧА КЛАСТЕРИЗАЦИИ

- распределение данных на группы: разделение всех клиентов мобильного оператора по уровню платёжеспособности, отнесение космических объектов к той или иной категории

ЗАДАЧА ВЫЯВЛЕНИЯ АНОМАЛИЙ

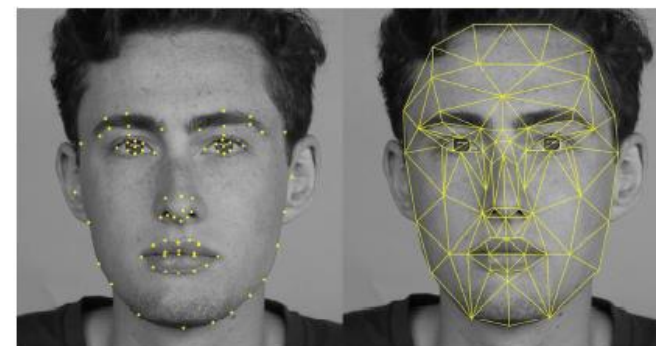
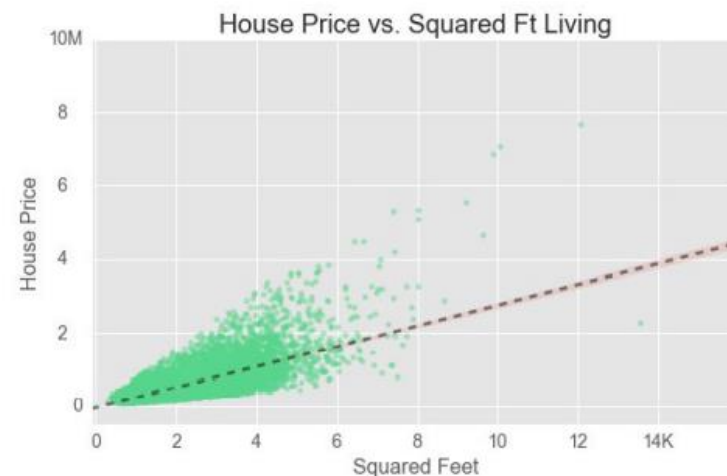
поиск отличий в наборах данных от стандартной информации: выявление мошеннических действий с банковскими картами.

ЗАДАЧА УМЕНЬШЕНИЯ РАЗМЕРНОСТИ

- сведение большого числа признаков к меньшему: сжатие данных

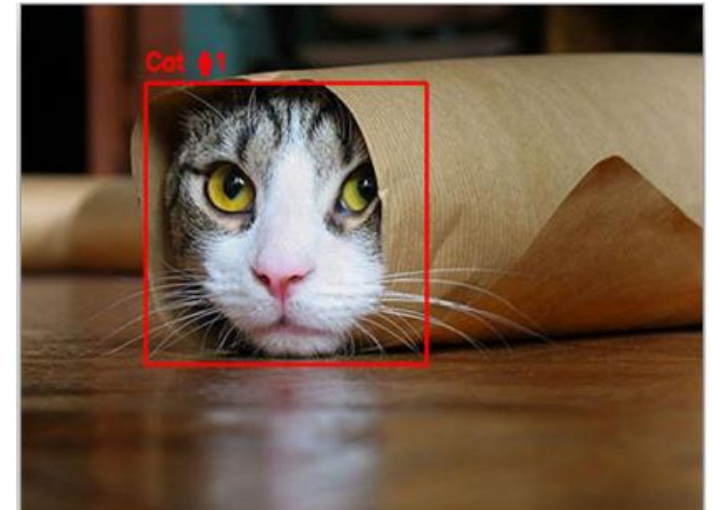
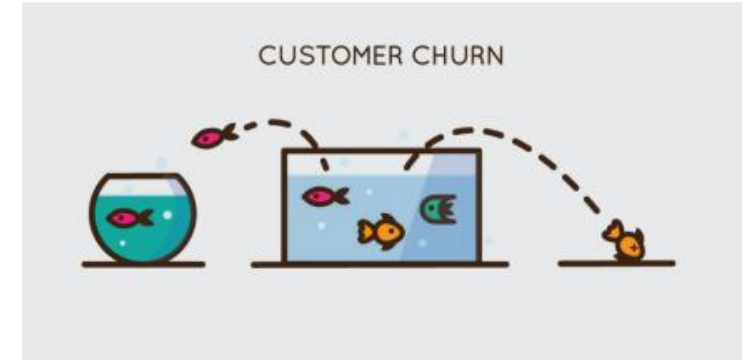
Примеры задач регрессии

- **Предсказание стоимости жилья** для риэлторской компании
- **Предсказание времени доставки**
- **Предсказание спроса на такси** в конкретном районе в конкретный час завтрашнего дня.
- **Предсказание ключевых точек лица**
— т.н. дескрипторов



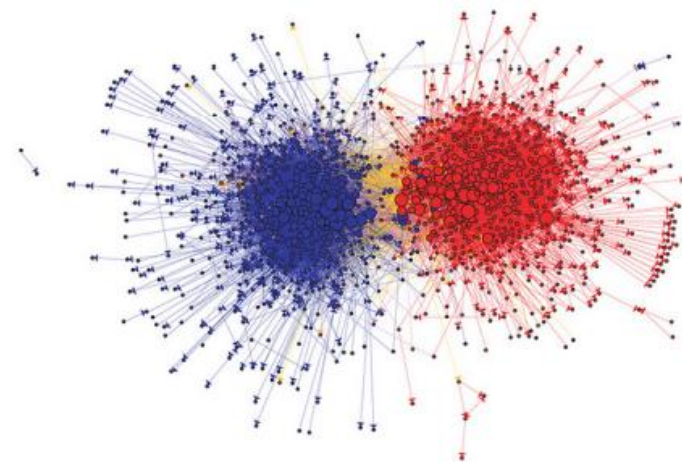
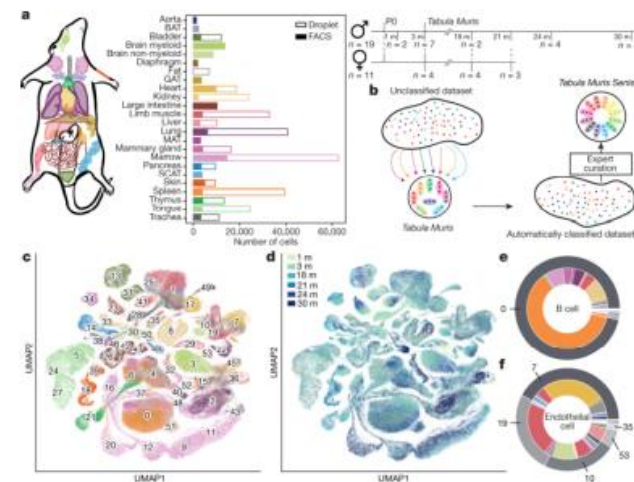
Примеры задач классификации

- **Предсказание оттока клиентов / сотрудников** на основе их поведения.
- **Классификация клеток ткани** на здоровые и опухолевые
- **Детекция объектов на фото.**

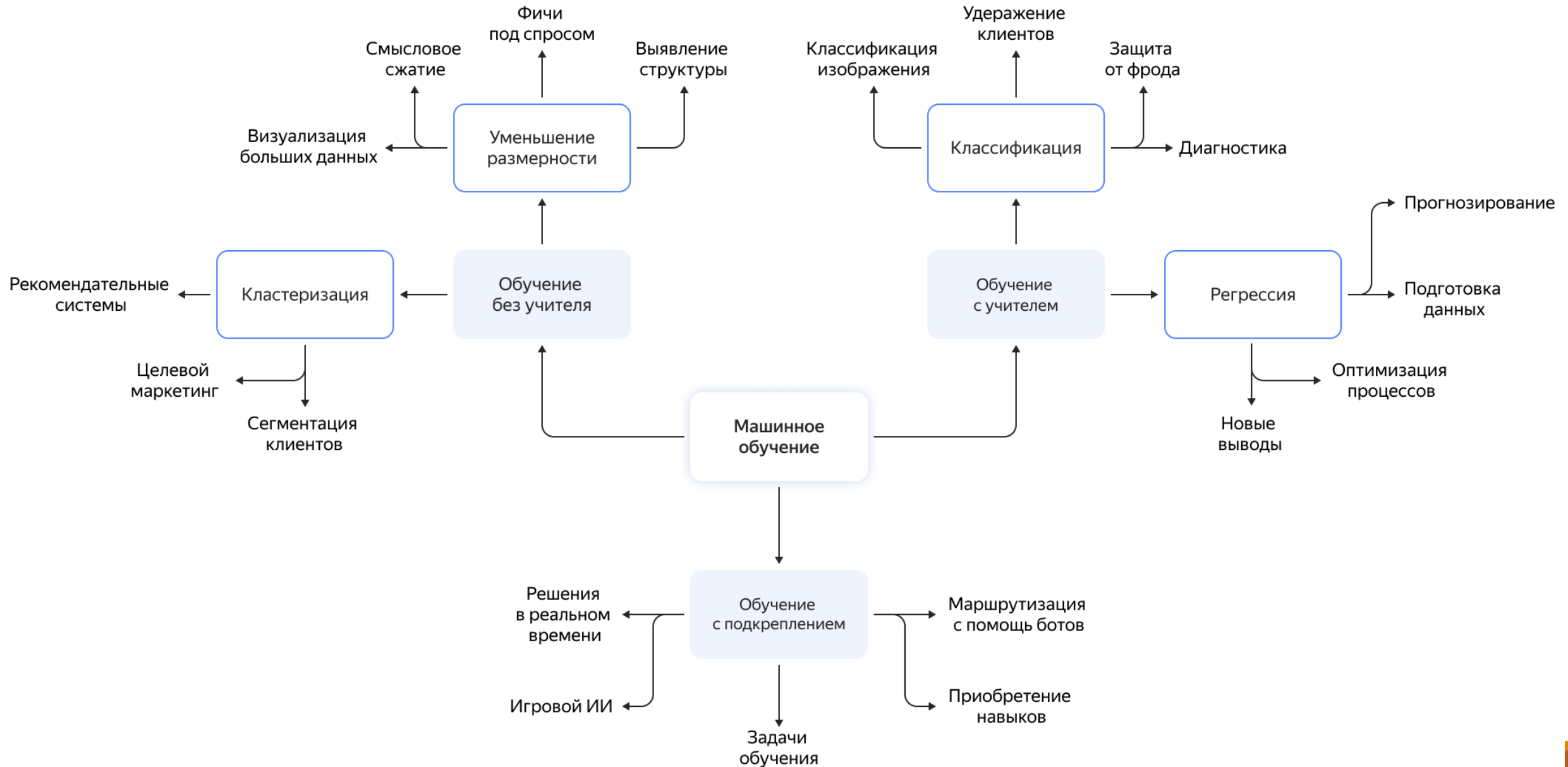


Примеры задач кластеризации

- **Сегментация аудитории** для таргетирования рекламы
- **Идентификация типов клеток** в образце данных секвенирования
- **Поиск сообществ в социальном графе** (из соцсети или из инсайдерской информации о структуре организации)
- **Задача разделения смеси распределений**

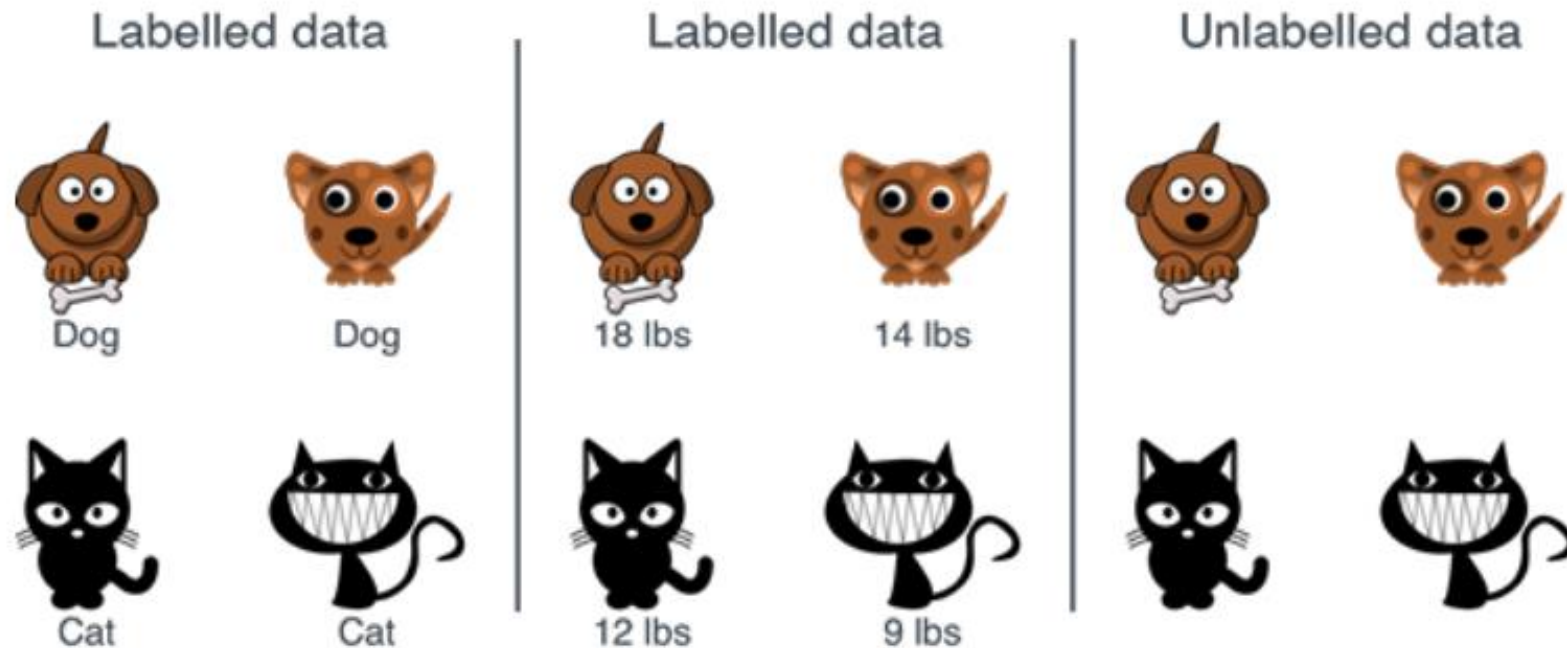


Основные виды машинного обучения





Размеченные (labelled) и неразмеченные (unlabelled) данные





Размеченные (labelled) и неразмеченные (unlabelled) данные

Размеченные

ID	Clump	UnifSize	UnifShape	MargAdh	SingEpiSize	BareNuc	BlandChrom	NormNucl	Mit	Class
1000025	5	1	1	1	2	1	3	1	1	benign
1002945	5	4	4	5	7	10	3	2	1	benign
1015425	3	1	1	1	2	2	3	1	1	malignant
1016277	6	8	8	1	3	4	3	7	1	benign
1017023	4	1	1	3	2	1	3	1	1	benign
1017122	8	10	10	8	7	10		7	1	malignant
1018099	1	1	1	1	2	10	3	1	1	benign
1018561	2	1	2	H	2	1	3	1	1	benign
1033078	2	1	1	1	2	1	1	1	5	benign
1033078	4	2	1	1	2	1	2	1	1	benign

labels

Неразмеченные

Customer Id	Age	Edu	Years Employed	Income	Card Debt	Other Debt	Address	DebtIncomeRatio
1	41	2		6	19	0.124	1.073 NBA001	6.3
2	47	1		26	100	4.582	8.218 NBA021	12.8
3	33	2		10	57	6.111	5.802 NBA013	20.9
4	29	2		4	19	0.681	0.516 NBA009	6.3
5	47	1		31	253	9.308	8.908 NBA008	7.2
6	40	1		23	81	0.998	7.831 NBA016	10.9
7	38	2		4	56	0.442	0.454 NBA013	1.6
8	42	3		0	64	0.279	3.945 NBA009	6.6
9	26	1		5	18	0.575	2.215 NBA006	15.5
10	47	3		23	115	0.653	3.947 NBA011	4
11	44	3		8	88	0.285	5.083 NBA010	6.1
12	34	2		9	40	0.374	0.266 NBA003	1.6

unlabeled

Обучение с учителем

на вход: набор тренировочных примеров (обучающий или тренировочный набор данных (training set или training sample — тренировочная выборка))

задача: продолжить уже известные ответы на новый опыт, выраженный обычно в виде тестового набора данных (test set, test sample)

Пример:

Датасет о квартирах, предсказание рыночной стоимости квартиры по ее параметрам (*задача регрессии*).

Подтвердить рак у пациента по медицинским показателям (*задача классификации*)

Пример: Линейная классификация текстов

Анализ тональности отзывов



Выборка:

Text

Label

Хочу оставить положительный отзыв при посещении банка на улице Менжинского 21 (ДО "Бабушкинский")

1

Очередь не соблюдается! На табло номера которых скорее всего и нет(сидел один....

0

Сегодня в банке была впервые, и осталась очень довольна.
Качество обслуживания на высоком уровне...

1

Позитивный

1

Негативный

0

Пример: Линейная классификация текстов

Задача бинарной классификации

Постановка задачи:

- Объект: текст отзыва
- Целевая переменная: {0,1}
- Метрика качества: доля верных ответов

$$\begin{aligned} & \text{[спасибо]} * 10 \\ + & \text{[реклама]} * (-30) \\ + & \text{[повторно]} * (-5) \\ + & \text{[плохо]} * (-10) \\ + & \text{[прекрасно]} * 10 \\ \hline \end{aligned}$$

$$\Sigma = b(x)$$

1. *Подготовка текстов:* разбиение на слова, приведение к начальной форме, удаление частых и редких слов ...
2. *Векторизация (эмбеддинг):* выделение признаков
3. *Построение линейной модели:* $b(x) > 0 \rightarrow 1$, $b(x) < 0 \rightarrow 0$
4. *Проверка модели:* доля ошибок на обучении и на тесте

Обучение без учителя

Машина самостоятельно выявляет закономерности, определяет признаки и классифицирует полученные данные. Минимизируют “энтропию” системы.

Задача кластеризации

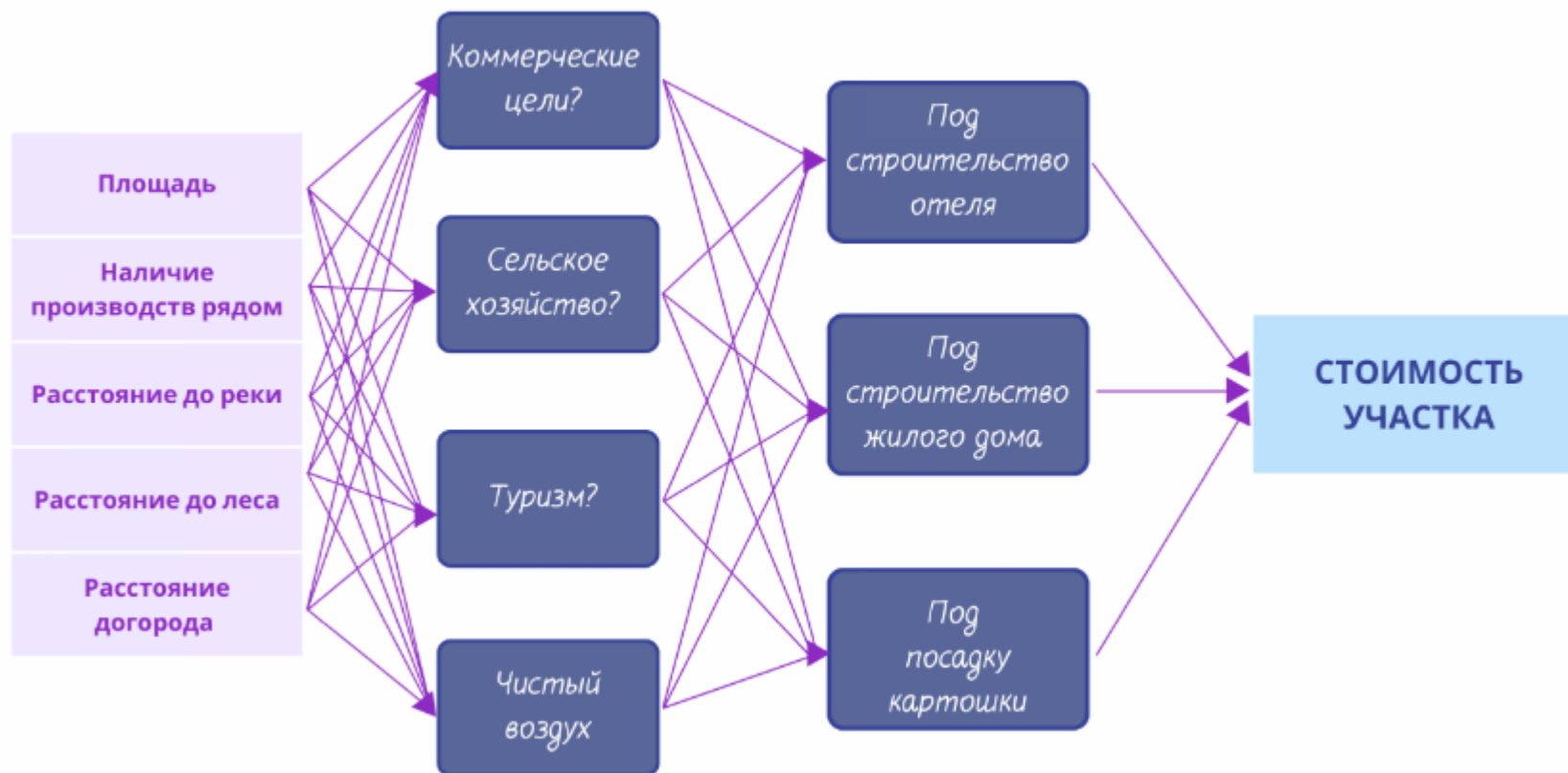
Задача снижение размерности

Пример

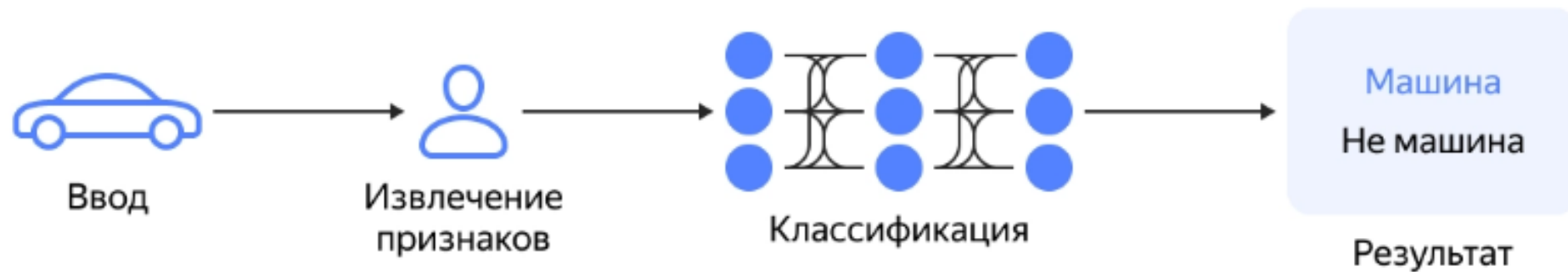
Каждый из объектов в выборке обладает сотней различных признаков, то основной трудностью будет графическое отображение такой выборки. Поэтому количество признаков уменьшают до двух или трёх, и становится возможным визуализировать их на плоскости или в 3D.

Глубокое обучение

вид машинного обучения с использованием **многослойных нейронных сетей**, которые самообучаются на большом наборе данных: обработка изображений, машинный перевод, синтез и распознавание человеческой речи.



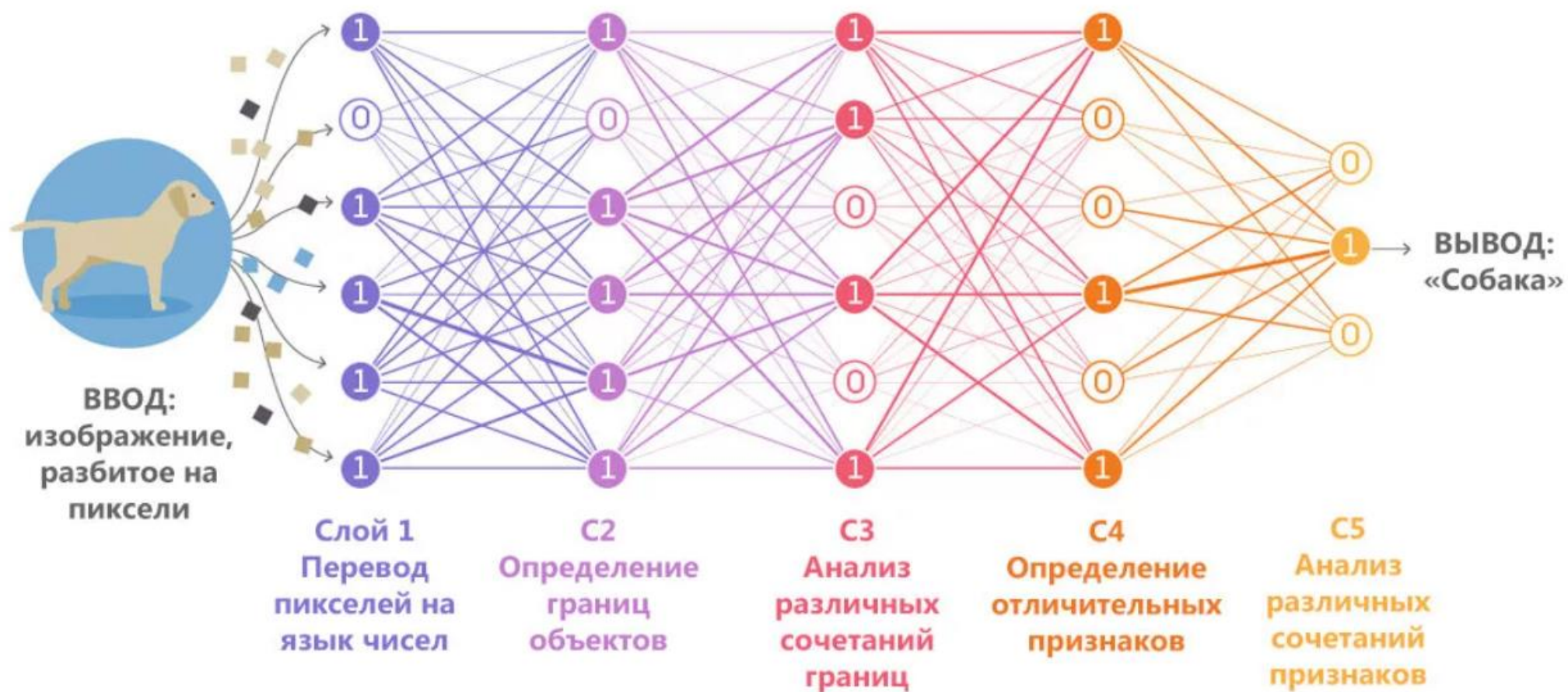
Машинное обучение



Глубокое обучение



Глубокое обучение



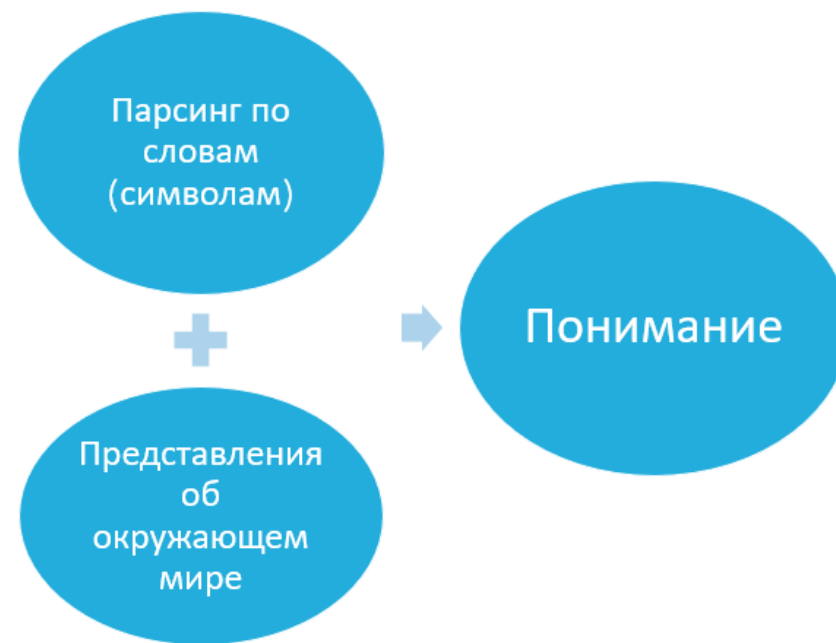
NLP и машинное обучение

Natural Language Processing (NLP)

Обработка естественного языка

Применение алгоритмов машинного обучения для обработки текста и речи:

- системы распознавания речи
- обобщения документов
- машинный перевод
- выявление спама
- распознавание именованных сущностей
- ответы на вопросы
- генерация текста и речи
- автоматический пересказ



Примеры использования NLP

Анализ тональности постов может предоставить много полезной информации о выборе клиентов и факторах, влияющих на их решения

NLP позволяет распознавать и **прогнозировать заболевания** на основе электронной медицинской документации и устной речи пациента.

Фильтрация и классификация электронных писем

Выявлении фейковых новостей

Автоматизация рутинных **судебных задач**

Интеллектуальные голосовые интерфейсы

История NLP

Обработка текста: **Eliza** - чат-бот, разработанный в 1960-х годах и пародирующий диалог с психотерапевтом.

Распознавание речи: в 1970-х годах в Университете Карнеги-Меллона была разработана **Harpy**. Первая компьютерная программа, которая понимала 1000 слов.

Синтез речи: Первым в мире устройством для синтеза речи считается **VODER** (англ. *Voice Operating Demonstrator* — модель голосового аппарата). Оно было разработано Гомером Дадли из компании Bell Labs в 1930-х годах.



Cortana

Виртуальный помощник Windows

Siri

Виртуальный помощник от Apple

Gmail

Почтовый сервис определяет спам

Dialogflow



Платформа от Google, которая позволяет создавать NLP-ботов.
Например, можно сделать бота для заказа пиццы

NLTK

Natural Language Toolkit – ведущая платформа для создания
NLP-программ на Python

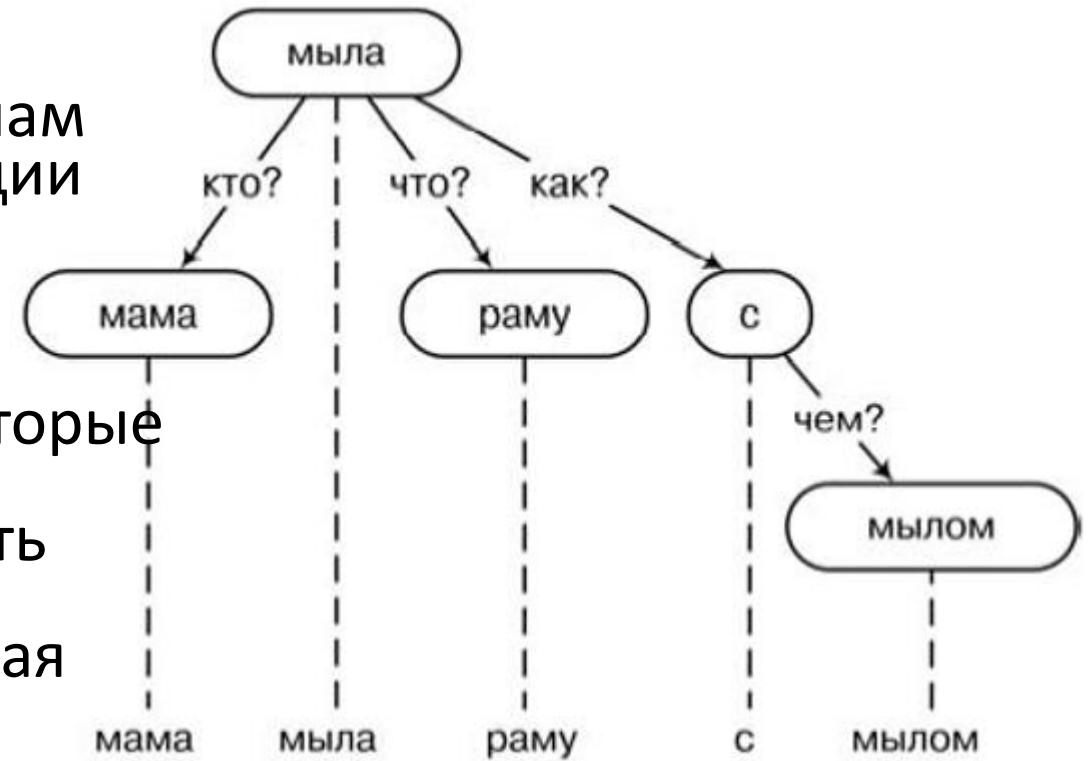
Развитие NLP

Алгоритмы, основанные на правилах и словарях, которые по заданным синтаксическим и семантическим шаблонам искали в текстах определенные конструкции или слова

Дерево синтаксического анализа

Языковые модели - это такие модели, которые для фрагмента текста (в NLP текст это последовательность слов) умеет оценивать вероятность встретить такую последовательность слов в языке. Языковая модель находит наиболее вероятное продолжение для текста.

Моделируется язык при помощи вероятности встретить одни слова рядом с другими



Задачи интеллектуальной обработки текстов

Задачи первого класса (синтаксические)

- Разметка по частям речи, по морфемам, распознавание именованных сущностей

Задачи второго класса (понимание текста)

- Языковые модели, информационный поиск, анализ тональности

Задачи третьего класса (понимание текста и порождение нового)

- Генерация текста, автоматическое реферирование (краткое содержание), диалоговые модели

Основные операции над текстом

Задачи интеллектуальной обработки текстов

Задачи первого класса (синтаксические)

- Разметка по частям речи, по морфемам, распознавание именованных сущностей

Задачи второго класса (понимание текста)

- Языковые модели, информационный поиск, анализ тональности

Задачи третьего класса (понимание текста и порождение нового)

- Генерация текста, автоматическое реферирование (краткое содержание), диалоговые модели

Регулярные выражения

это последовательность символов, которая определяет шаблон поиска

Модуль re

- `.` – любой символ, кроме перевода строки;
- `\w` – один символ;
- `\d` – одна цифра;
- `\s` – один пробел;
- `\W` – один НЕсимвол;
- `\D` – одна НЕцифра;
- `\S` – один НЕпробел;
- `[abc]` – находит любой из указанных символов match any of a, b, or c;
- `[^abc]` – находит любой символ, кроме указанных;
- `[a-g]` – находит символ в промежутке от a до g.

Примеры регулярных выражений

Шаблон	Описание
<code>[a-zA-Z0-9._%+-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,}</code>	Проверяет наличие символов перед @, после @ и после точки.

```
import re
text = "Привет! Меня зовут Денис, мой email - denis@mail.ru"
match = re.search(r'(\w+)@', text)
print(match.group(0))
if match:
    print("Имя пользователя:", match.group(1))
```

denis@

Имя пользователя: denis

Библиотека NLTK

NLTK (Natural Language Toolkit) – ведущая платформа для создания NLP-программ на Python. У нее есть легкие в использовании интерфейсы для многих [языковых корпусов](#), а также библиотеки для обработки текстов для классификации, токенизации, [стемминга](#), [разметки](#), фильтрации и [семантических рассуждений](#). Ну и еще это бесплатный опенсорсный проект, который развивается с помощью коммьюнити.

import nltk

Токенизация по предложениям `nltk.sent_tokenize`

Токенизация по словам `nltk.word_tokenize`

Чтение данных

In [3]:

```
1 import pandas as pd
2 # Reading Tab separated Value
3 data = pd.read_csv('SMSSpamCollection.tsv', sep='\t', names=['label', 'body_text'], header=None)
4 # Print first 5 data
5 data.head()
```

Out[3]:

	label	body_text
0	ham	I've been searching for the right words to tha...
1	spam	Free entry in 2 a wkly comp to win FA Cup fina...
2	ham	Nah I don't think he goes to usf, he lives aro...
3	ham	Even my brother is not like to speak with me. ...
4	ham	I HAVE A DATE ON SUNDAY WITH WILL!!

Удаление пунктуации

```
1 import string
2 string.punctuation
```

```
'!"#$%&\'()*+,-./:;<=>?@[\\]^_`{|}~'
```

```
1 #Function to remove Punctuation
2 def remove_punct(text):
3     text_nopunct = "".join([char for char in text if char not in string.punctuation])# It will discard all punctuations
4     return text_nopunct
5
6 data['body_text_clean'] = data['body_text'].apply(lambda x: remove_punct(x))
7
8 data.head()
```

	label	body_text	body_text_clean
0	ham	I've been searching for the right words to tha...	Ive been searching for the right words to than...
1	spam	Free entry in 2 a wkly comp to win FA Cup fina...	Free entry in 2 a wkly comp to win FA Cup fina...
2	ham	Nah I don't think he goes to usf, he lives aro...	Nah I dont think he goes to usf he lives aroun...
3	ham	Even my brother is not like to speak with me. ...	Even my brother is not like to speak with me T...
4	ham	I HAVE A DATE ON SUNDAY WITH WILL!!	I HAVE A DATE ON SUNDAY WITH WILL

Токенизация по предложениям

процесс разделения письменного языка на предложения-компоненты

по знаку пунктуации - точка

`nltk.sent_tokenize`

Токенизация по словам

- процесс разделения предложений на слова-компоненты
- по знаку пробела
- `nltk.word_tokenize`

Токенизация

```
1 import re
2
3 # Function to Tokenize words
4 def tokenize(text):
5     tokens = re.split('\W+', text) #W+ means that either a word character (A-Za-z0-9_) or a dash (-) can go there.
6     return tokens
7
8 data['body_text_tokenized'] = data['body_text_clean'].apply(lambda x: tokenize(x.lower()))
9 #We convert to lower as Python is case-sensitive.
10
11 data.head()
```

	label	body_text	body_text_clean	body_text_tokenized
0	ham	I've been searching for the right words to tha...	Ive been searching for the right words to than...	[ive, been, searching, for, the, right, words, ...
1	spam	Free entry in 2 a wkly comp to win FA Cup fina...	Free entry in 2 a wkly comp to win FA Cup fina...	[free, entry, in, 2, a, wkly, comp, to, win, f...
2	ham	Nah I don't think he goes to usf, he lives aro...	Nah I dont think he goes to usf he lives aroun...	[nah, i, dont, think, he, goes, to, usf, he, l...
3	ham	Even my brother is not like to speak with me. ...	Even my brother is not like to speak with me T...	[even, my, brother, is, not, like, to, speak, ...
4	ham	I HAVE A DATE ON SUNDAY WITH WILL!!	I HAVE A DATE ON SUNDAY WITH WILL	[i, have, a, date, on, sunday, with, will]

Стоп - слова

Это слова, которые выкидываются из текста до/после обработки текста.

Такие слова могут добавить много шума, поэтому необходимо избавляться от нерелевантных слов.

Стоп-слова это обычно понимают артикли, междометия, союзы и т.д., которые не несут смысловой нагрузки.

`nltk.download("stopwords")`

Удаление стоп-слов

```
1 import nltk
2
3 stopword = nltk.corpus.stopwords.words('english')# All English Stopwords

1 # Function to remove Stopwords
2 def remove_stopwords(tokenized_list):
3     text = [word for word in tokenized_list if word not in stopword]# To remove all stopwords
4     return text
5
6 data['body_text_nostop'] = data['body_text_tokenized'].apply(lambda x: remove_stopwords(x))
7
8 data.head()
```

	label	body_text	body_text_clean	body_text_tokenized	body_text_nostop
0	ham	I've been searching for the right words to tha...	Ive been searching for the right words to than...	[ive, been, searching, for, the, right, words,...	[ive, searching, right, words, thank, breather...
1	spam	Free entry in 2 a wkly comp to win FA Cup fina...	Free entry in 2 a wkly comp to win FA Cup fina...	[free, entry, in, 2, a, wkly, comp, to, win, f...	[free, entry, 2, wkly, comp, win, fa, cup, fin...
2	ham	Nah I don't think he goes to usf, he lives aro...	Nah I dont think he goes to usf he lives around...	[nah, i, dont, think, he, goes, to, usf, he, l...	[nah, dont, think, goes, usf, lives, around, t...
3	ham	Even my brother is not like to speak with me. ...	Even my brother is not like to speak with me T...	[even, my, brother, is, not, like, to, speak, ...	[even, brother, like, speak, treat, like, aids...
4	ham	I HAVE A DATE ON SUNDAY WITH WILL!!	I HAVE A DATE ON SUNDAY WITH WILL	[i, have, a, date, on, sunday, with, will]	[date, sunday]

Лемматизация и стемминг текста

Привести все встречающиеся словоформы к одной, нормальной словарной форме.

Частные случаи нормализации

```
dog, dogs, dog's, dogs' => dog
```

Стемминг – это грубый эвристический процесс, который отрезает «лишнее» от корня слов, часто это приводит к потере словообразовательных суффиксов.

Лемматизация – это более тонкий процесс, который использует словарь и морфологический анализ, чтобы в итоге привести слово к его канонической форме – лемме.

Стемминг

```
In [12]: 1 ps = nltk.PorterStemmer()
2
3 def stemming(tokenized_text):
4     text = [ps.stem(word) for word in tokenized_text]
5     return text
6
7 data['body_text_stemmed'] = data['body_text_nostop'].apply(lambda x: stemming(x))
8
9 data.head()
```

```
Out[12]:
```

	label	body_text	body_text_clean	body_text_tokenized	body_text_nostop	body_text_stemmed
0	ham	I've been searching for the right words to tha...	Ive been searching for the right words to than...	[ive, been, searching, for, the, right, words,...	[ive, searching, right, words, thank, breather...	[ive, search, right, word, thank, breather, pr...
1	spam	Free entry in 2 a wkly comp to win FA Cup fina...	Free entry in 2 a wkly comp to win FA Cup fina...	[free, entry, in, 2, a, wkly, comp, to, win, f...	[free, entry, 2, wkly, comp, win, fa, cup, fin...	[free, entri, 2, wkli, comp, win, fa, cup, fin...
2	ham	Nah I don't think he goes to usf, he lives aro...	Nah I dont think he goes to usf he lives aroun...	[nah, i, dont, think, he, goes, to, usf, he, l...	[nah, dont, think, goes, usf, lives, around, t...	[nah, dont, think, goe, usf, live, around, tho...
3	ham	Even my brother is not like to speak with me. ...	Even my brother is not like to speak with me T...	[even, my, brother, is, not, like, to, speak, ...	[even, brother, like, speak, treat, like, aids...	[even, brother, like, speak, treat, like, aid...
4	ham	I HAVE A DATE ON SUNDAY WITH WILL!!	I HAVE A DATE ON SUNDAY WITH WILL	[i, have, a, date, on, sunday, with, will]	[date, sunday]	[date, sunday]

Лемматизация

In [13]:

```
1 wn = nltk.WordNetLemmatizer()
2
3 def lemmatizing(tokenized_text):
4     text = [wn.lemmatize(word) for word in tokenized_text]
5     return text
6
7 data['body_text_lemmatized'] = data['body_text_nostop'].apply(lambda x: lemmatizing(x))
8
9 data.head(10)
```

Out[13]:

	label	body_text	body_text_clean	body_text_tokenized	body_text_nostop	body_text_stemmed	body_text_lemmatized
0	ham	I've been searching for the right words to tha...	Ive been searching for the right words to than...	[ive, been, searching, for, the, right, words,...	[ive, searching, right, words, thank, breather...	[ive, search, right, word, thank, breather, pr...	[ive, searching, right, word, thank, breather,...
1	spam	Free entry in 2 a wkly comp to win FA Cup fina...	Free entry in 2 a wkly comp to win FA Cup fina...	[free, entry, in, 2, a, wkly, comp, to, win, f...	[free, entry, 2, wkly, comp, win, fa, cup, fin...	[free, entri, 2, wkli, comp, win, fa, cup, fin...	[free, entry, 2, wkly, comp, win, fa, cup, fin...
2	ham	Nah I don't think he goes to usf, he lives aro...	Nah I dont think he goes to usf he lives aroun...	[nah, i, dont, think, he, goes, to, usf, he, l...	[nah, dont, think, goes, usf, lives, around, t...	[nah, dont, think, goe, usf, live, around, tho...	[nah, dont, think, go, usf, life, around, though]
3	ham	Even my brother is not like to speak with me. ...	Even my brother is not like to speak with me T...	[even, my, brother, is, not, like, to, speak, ...	[even, brother, like, speak, treat, like, aids...	[even, brother, like, speak, treat, like, aid,...	[even, brother, like, speak, treat, like, aid,...
4	ham	I HAVE A DATE ON SUNDAY WITH WILL!!	I HAVE A DATE ON SUNDAY WITH WILL	[i, have, a, date, on, sunday, with, will]	[date, sunday]	[date, sunday]	[date, sunday]
5	ham	As per your request 'Melle Melle (Oru Minnamin...	As per your request Melle Melle Oru Minnaminun...	[as, per, your, request, melle, melle, oru, mi...	[per, request, melle, melle, oru, minnaminungi...	[per, request, mell, mell, oru, minnaminungint...	[per, request, melle, melle, oru, minnaminungi...

Мешок слов (bag of words)

Выделяет вектору весь документ, и каждый элемент кодируется 1 по порядку следования слов в словаре

Метод не учитывает важность того или иного токена, ведь одно слово может повторяться по

Теряется вся информация о взаимном расположении слов внутри текста

```
{Пес, Кот, ель, на, сел, пенъ} # Словарь
# Корпус:
[[1, 0, 0, 1, 1, 1] # Первый документ
 [0, 1, 1, 1, 1, 0]] # Второй документ
```

```
[Пес, сел, на, пенъ] # Первый документ
[Кот, сел, на, ель] # Второй документ
[[Пес, сел, на, пенъ], [Кот, сел, на, ель]] # Корпус
```

Word embeddings – векторное представление слов

Word2vec - 2013

Слова, которые встречаются в одинаковых окружениях, имеют близкие значения.

Предсказывать вероятность слова по его окружению (контексту).

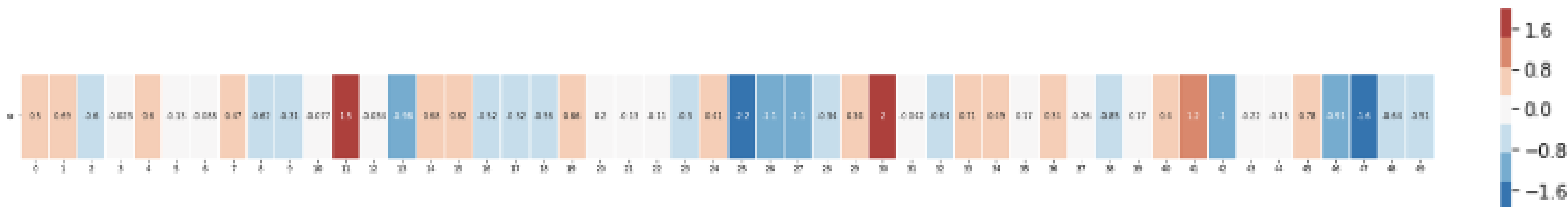
Вероятность, присваиваемая моделью слову близка к вероятности встретить это слово в этом окружении в реальном тексте.



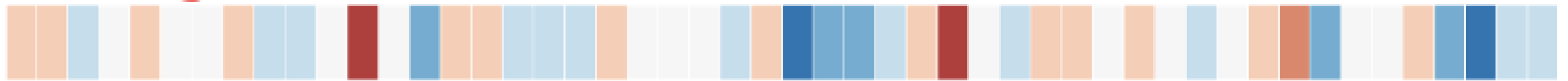
Модель обработки естественного языка на основе вложений

Слово «Король»

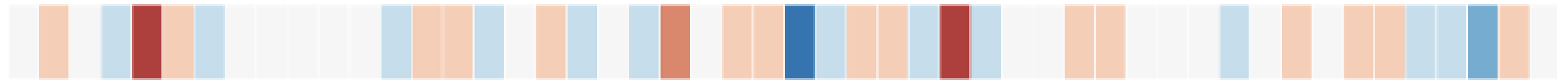
[0.50451 , 0.68607 , -0.59517 , -0.022801, 0.60046 , -0.13498 , -0.08813 , 0.47377 ,
-0.61798 , -0.31012 , -0.076666, 1.493 , -0.034189, -0.98173 , 0.68229 , 0.81722 ,
-0.51874 , -0.31503 , -0.55809 , 0.66421 , 0.1961 , -0.13495 , -0.11476 , -0.30344 ,
0.41177 , -2.223 , -1.0756 , -1.0783 , -0.34354 , 0.33505 , 1.9927 , -0.04234 ,
-0.64319 , 0.71125 , 0.49159 , 0.16754 , 0.34344 , -0.25663 , -0.8523 , 0.1661 ,
0.40102 , 1.1685 , -1.0137 , -0.21585 , -0.15155 , 0.78321 , -0.91241 , -1.6106 ,



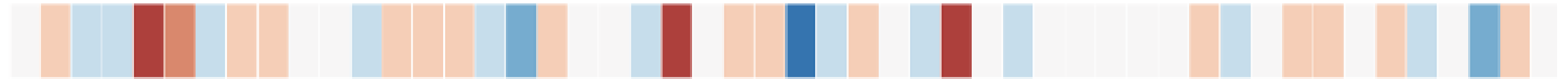
“king”



“Man”



“Woman”



Определение смысла крупных участков текста: предложений, абзацев, статей

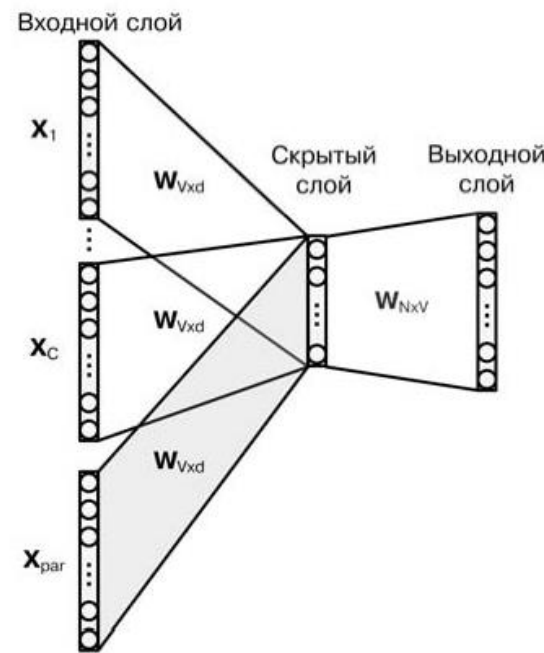
Объединение векторов отдельных слов в векторы нескольких слов:

Сумма или среднее (резюмирование документов)

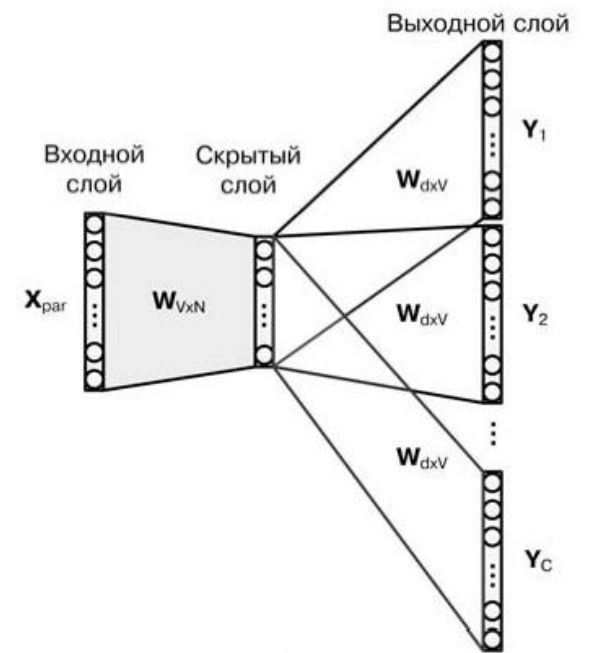
Обучение векторов слов

Distributed Memory Model of Paragraph Vectors (вектор, представляющий абзац или предложение, определяется как еще один дополнительный вектор весов)

Distributed Bag of Words Model of Paragraph Vectors (слова контекста просто полностью игнорируются, и модель использует вектор абзаца, чтобы предсказать слова из окна в том же абзаце)



а



б