Машинное обучение и предобработка данных

Сегодня

1. Что такое машинное обучение
2. Типы задач машинного обучения
3. Предварительная обработка данных

Методология анализа данных

CRISP-DM (Cross-Industry Standard Process for Data Mining) - наиболее распространенная методология ведения проектов по исследованию данных.

4. Моделирование

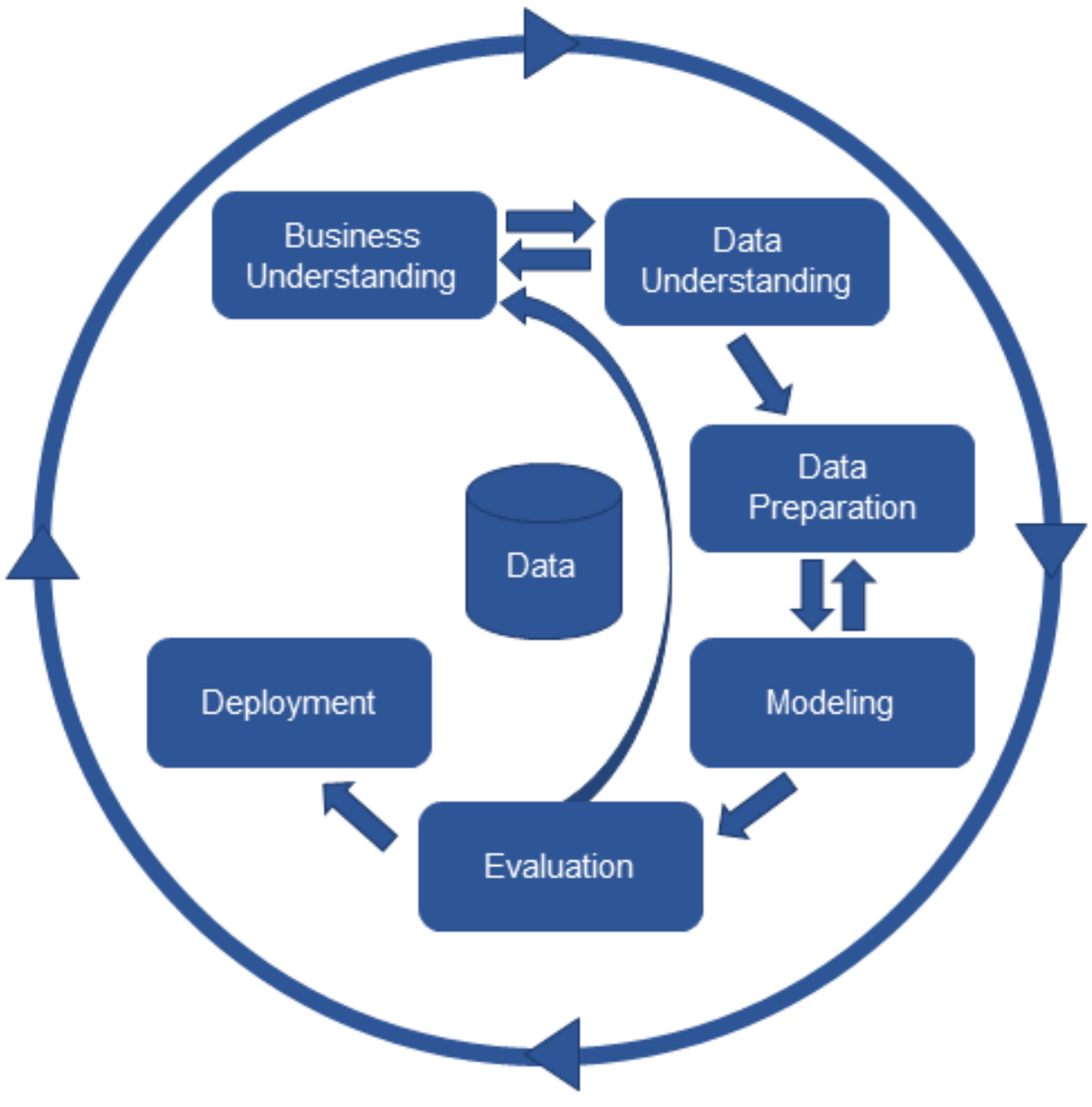
1. Выбрать методику моделирования
2. Сделать тесты для модели
3. Построить модель
4. Оценить модель

5. Оценка

1. Оценить результаты
2. Сделать ревью процесса
3. Определить следующие шаги

6. Развертывание

1. Запланировать развертывание
2. Запланировать поддержку и мониторинг развернутого решения
3. Сделать финальный отчет
4. Сделать ревью проекта

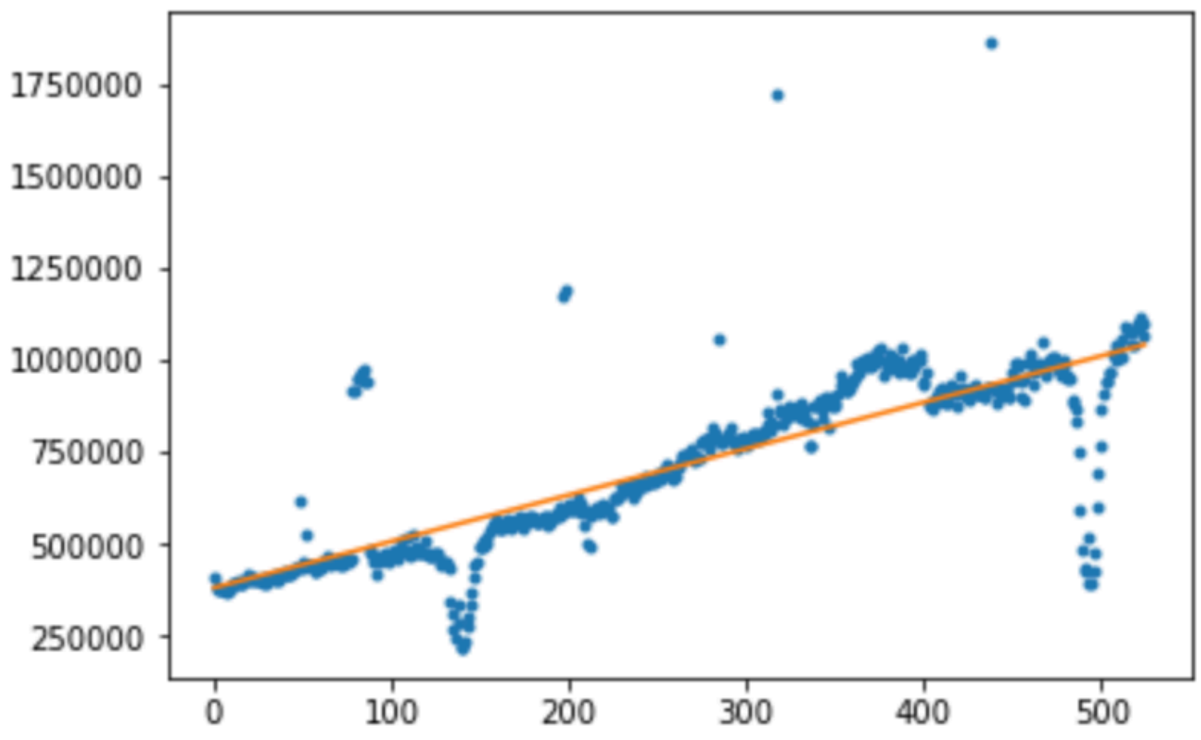


Пример задачи прогнозирования

Найдем тренд по значениям посещаемости сайта в день

```
x = np.arange(0,len(grouped.MAX_USER))
y=np.array(grouped.MAX_USER)
z = np.polyfit(x,y,1)
print("{0}x + {1}".format(*z))

from matplotlib import pyplot as plt

plt.plot(x,y,'.')
trendpoly = np.poly1d(z)
plt.plot(x,trendpoly(x))
plt.show()
```

```
1258.7273731087068x + 381743.1825312329
```



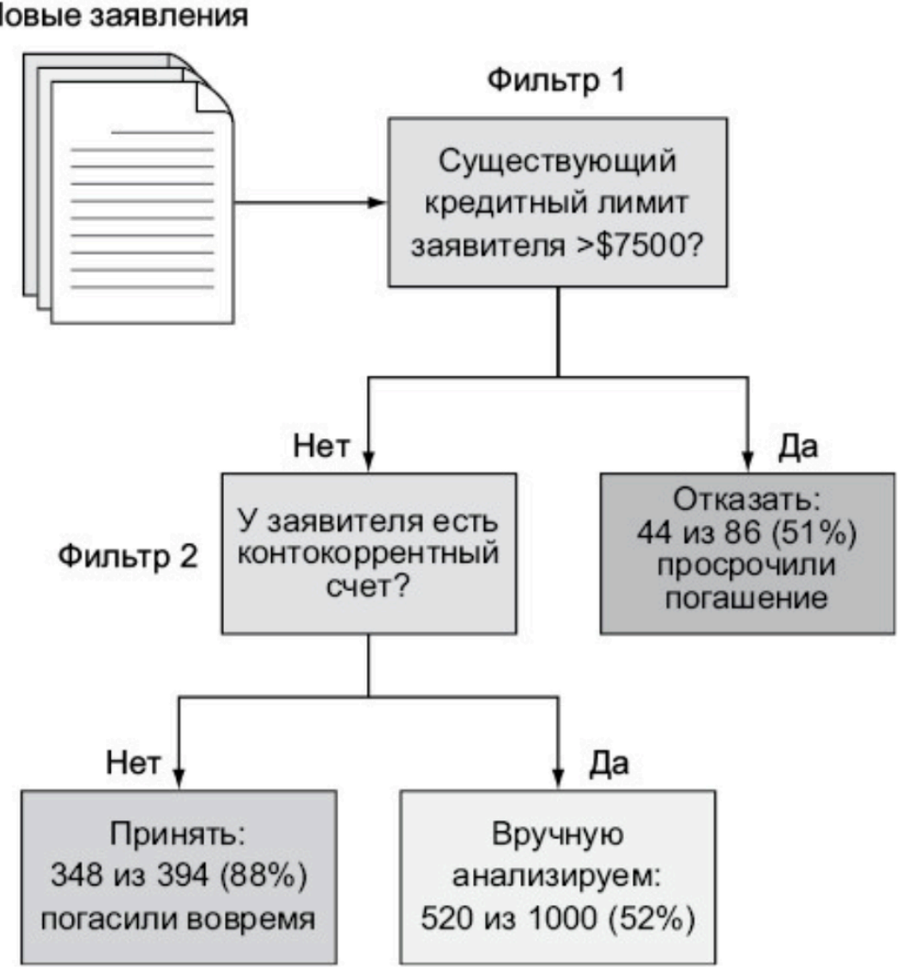
Что такое машинное обучение

Машинное обучение (МО) - наиболее распространенный и мощный метод анализа данных.
Это **процесс**, в ходе которого система обрабатывает **большое число примеров**, выявляет **закономерности** и использует их, чтобы **прогнозировать** характеристики **новых данных**.

Для МО необходимо:

- большой объем данных для обучения;
- возможность представления данных в виде набора признаков.

Распространенный пример задачи МО:
одобрение кредита клиенту банка.



Сложности в работе с МО

К сложным аспектам машинного обучения относятся:

- **распознавание и формулировка задач**, к которым оно применимо;
- **получение данных и преобразование** их в пригодную для использования форму;
- обнаружение **корректных алгоритмов;**
- **проектирование признаков** и **переобучение**.

Постановка задачи МО

Цель машинного обучения — обнаружение закономерностей и взаимосвязей в данных и практическое применение полученной информации.

Общая постановка задачи:

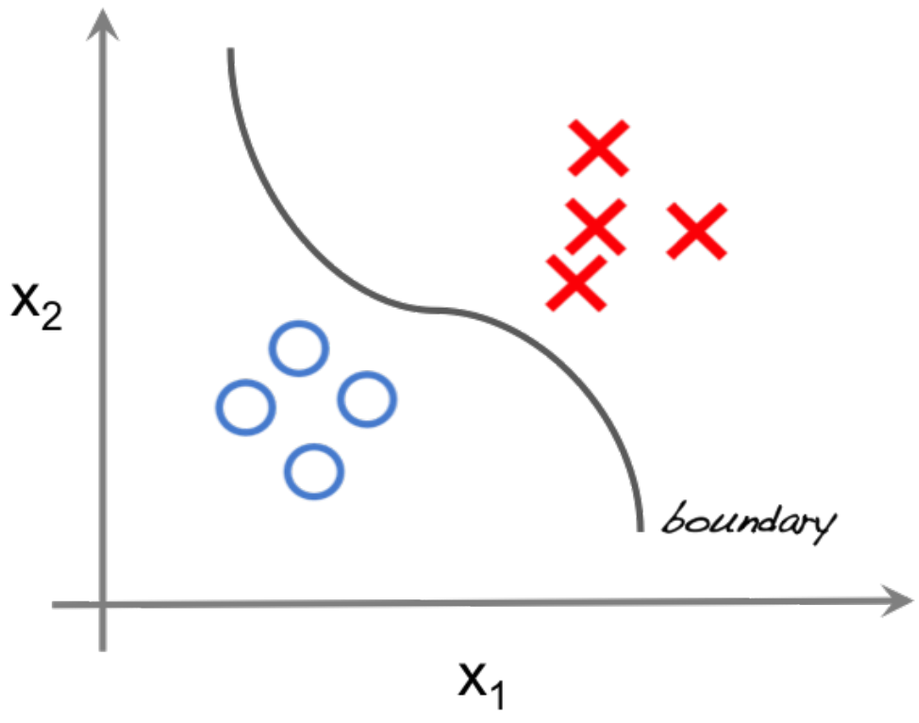
Дано: конечное множество ***прецедентов*** (объектов, ситуаций), по каждому из которых собраны (измерены) некоторые данные. Данные о прецеденте называют также его описанием. Совокупность всех имеющихся описаний прецедентов называется ***обучающей выборкой***.

Требуется: по этим частным данным выявить ***общие зависимости, закономерности, взаимосвязи***, присущие не только этой конкретной выборке, но вообще всем прецедентам, в том числе тем, которые ещё не наблюдались.

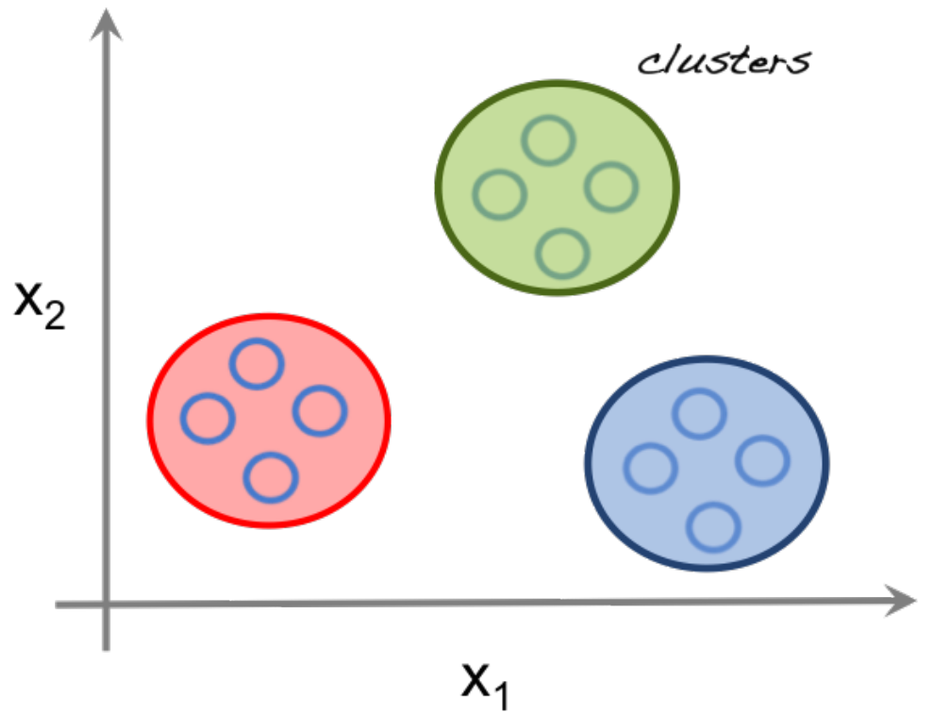
Типы задач МО

Задачи с машинным обучением делятся на два типа — **обучение с учителем (supervised learning)** и **обучение без учителя (unsupervised learning)**.

Supervised learning



Unsupervised learning

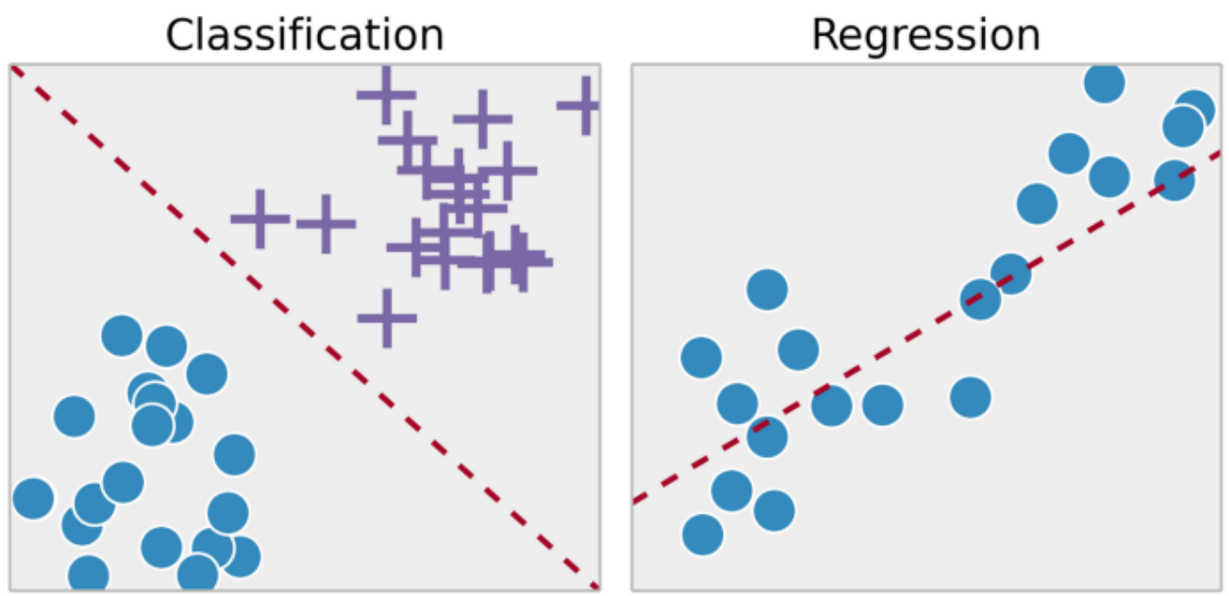


Обучение с учителем

Обучение с учителем (supervised learning) - наиболее распространённый случай. Каждый прецедент представляет собой пару «объект, ответ». Требуется найти функциональную зависимость ответов от описаний объектов и построить **алгоритм**, **принимающий на входе описание объекта** и **выдающий на выходе ответ**. Функционал качества обычно определяется как средняя ошибка ответов, выданных алгоритмом, по всем объектам выборки.

Это методы решения задачи, где определен **правильный ответ**.

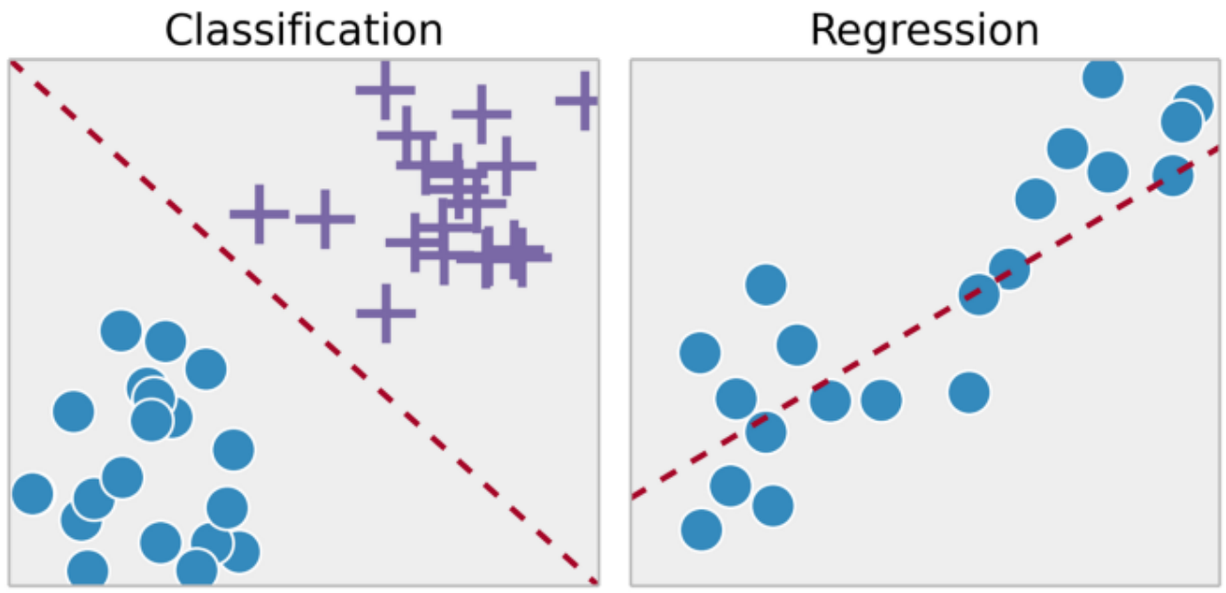
Обучение с учителем направлено на решение задач **классификации** и **регрессии**.



Обучение с учителем

Задача **классификации** отличается тем, что множество допустимых ответов конечно. Их называют метками классов (значение целевой переменной). Класс — это множество всех объектов с данным значением метки. Набор меток заранее известен.

Задача **регрессии** отличается тем, что допустимым ответом является действительное число.



Значение целевой переменной дискретно

Значение целевой переменной непрерывно

Обучение с учителем

Задача **классификации** отличается тем, что множество допустимых ответов конечно. Их называют метками классов (значение целевой переменной). Класс — это множество всех объектов с данным значением метки. Набор меток заранее известен.

Задача **регрессии** отличается тем, что допустимым ответом является действительное число.

Примеры:

1.Определение эмоциональной окраски отзыва?

2.Определение для сообщения - спам или не спам?

3.Определение оценки за сочинение?

4.Определение качества программного кода?

Обучение с учителем

Задача **классификации** отличается тем, что множество допустимых ответов конечно. Их называют метками классов (значение целевой переменной). Класс — это множество всех объектов с данным значением метки. Набор меток заранее известен.

Задача **регрессии** отличается тем, что допустимым ответом является действительное число.

Примеры:

1.Определение эмоциональной окраски отзыва?	классификация
2.Определение для сообщения - спам или не спам?	классификация
3.Определение оценки за сочинение?	классификация
4.Определение качества программного кода?	классификация / регрессия

Достаточный объем обучающих данных

Достаточный объем обучающей выборки зависит от конкретной задачи, универсальных рекомендаций и общих правил не существует.
Факторы, от которых зависит количество необходимых данных:

- **Сложность задачи.** Можно ли описать связь между входными признаками и целевой переменной простым шаблоном или же она запутана и не имеет линейной зависимости?
- **Требования к точности.** Если достаточно всего 60% успешных результатов, можно обойтись меньшей обучающей выборкой, чем в случае, когда необходимо получить 95% успешных результатов.
- **Размерность пространства признаков.** Если доступны всего два входных признака, обучающих данных потребуется меньше, чем при наличии 2000 таких признаков.

Поиск похожего текста

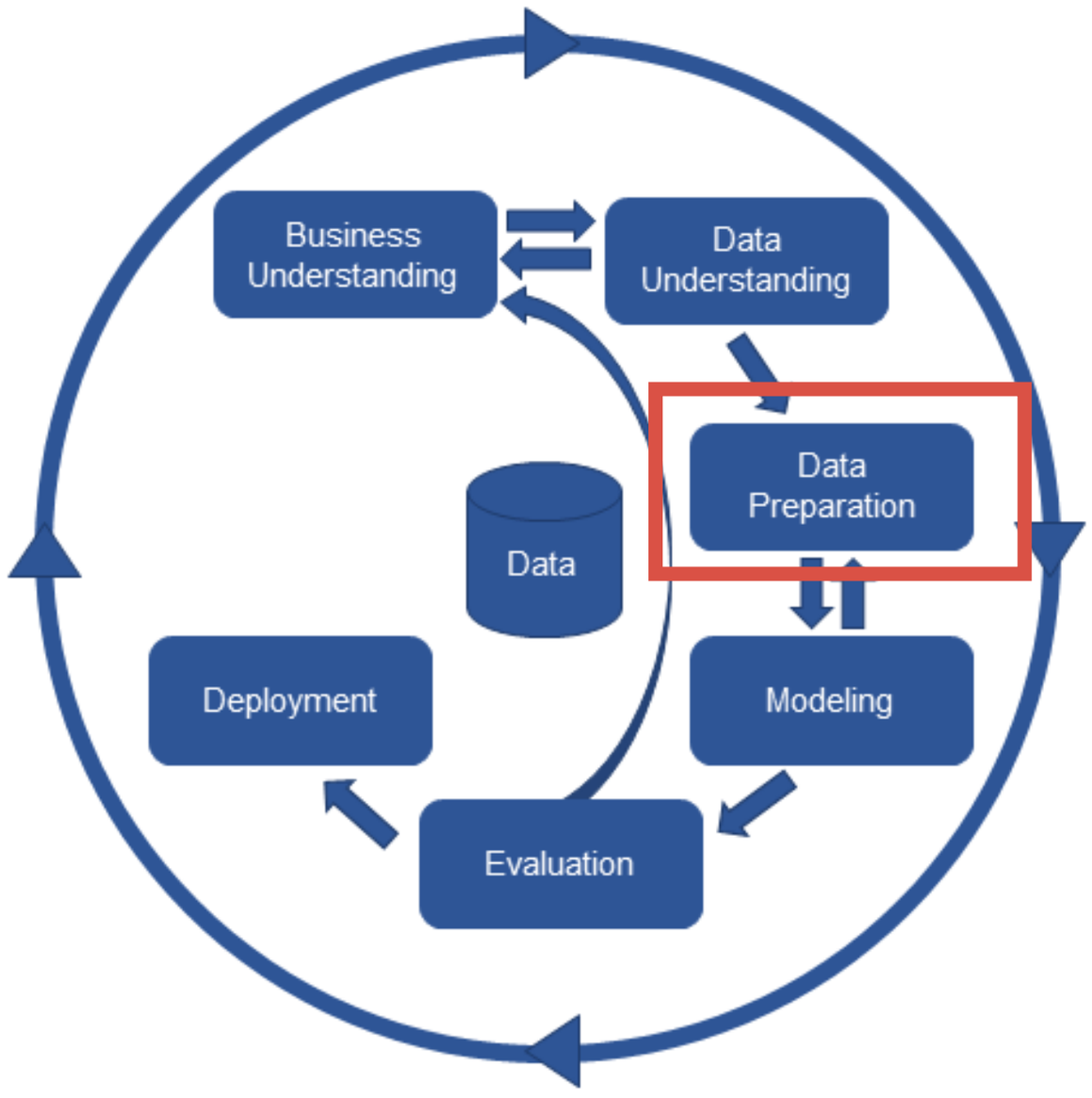
```
#поиск ближайшего нормированного текста
found_publ_n = None
found_dist_n = sys.maxsize
found_i_n = None

for i in range(0, count_samples):
    publ = (list(data))[i]
    if publ == request:
        continue
    publ_vect = X_train.getrow(i)
    cur_dist = dist_norm(publ_vect, request_vect)
    #печать всех расстояний
    #print('publ %i dist=%.2f: %s' % (i, cur_dist, publ))
    if cur_dist<found_dist_n:
        found_dist_n = cur_dist
        found_publ_n = publ
        found_i_n = i

print('\nBest publication is %i with dist=%.2f: %s' % (found_i_n, found_dist_n, found_publ_n))
```

```
Best publication is 65 with dist=1.31: <p>Organizing your travel without agents is almost always cheaper and more fun
.
But does that also apply for Maldives? Can I just travel to the Maldives? Do you have some tips or experiences to rep
ort? </p>
```

Исходное сообщение: «safely travel tour Amazon jungle»

Найден НЕ подходящий текст

Методология анализа данных



Поиск похожего текста

Возьмем для наглядности не содержимое статей, а их заголовки.

Наиболее похожие текст ДО нормирования:

```
Best publication is 98 with dist=2.83: China visa issue
```

Наиболее похожие текст ПОСЛЕ нормирования:

```
Best publication is 1 with dist=1.12: How can I find a guide that will take me safely through the Amazon jungle?
```

Исходное сообщение: «safely travel tour Amazon jungle»

Найден подходящий текст

Предобработка данных

Выполним предобработку текстов:

1. Приведем в нижнему регистру;
2. Удалим HTML-теги;
3. Удалим лишние символы;
4. Удалим стоп-слова;
5. Выполним токенизацию.

Предобработка данных

Выполним предобработку текстов.

```
import regex as re

#удаление HTML-тегов, пунктуации, декодирование (чистка)
def clean_content(content):
    #приведение к нижнему регистру
    content = content.apply(lambda x: x.lower())
    #удаление HTML-тегов
    content = content.apply(lambda x: re.sub(r'\<[^<>]*\>','',x))
    #удаление всех символов кроме букв, цифр и подчеркивания
    content = content.apply(lambda x: re.sub(r'^\W+|\W+$',' ',x))
    #удаление пробелов, переводов строк и табов
    content = content.apply(lambda x: re.sub(r'\s',' ',x))
    #удаление знаков препинания
    content = content.apply(lambda x: re.sub(r'[^a-zA-Z0-9]',' ',x))
    return(content)
```

Предобработка данных

Выполним предобработку текстов.

```
cleaned_data = clean_content(data)
print(data)
```

```
id
1       <p>My fiancée and I are looking for a good Car...
2       <p>This was one of our definition questions, b...
4       <p>Singapore Airlines has an all-business clas...
5       <p>Another definition question that interested...
6       <p>A year ago I was reading some magazine, and...
8       <p>Can anyone suggest the best way to get from...
9       <p>We are considering visiting Argentina for u...
10      <p>Recently my wife and I traveled to Italy an...
11      <p>I'm planning on taking the trans-Siberian /...
13      <p>I need to travel from Cusco, Peru to La Paz...
14      <p>I am aware of travel agencies catering to U...
15      <p>I'm planning my first international trip ou...
16      <p>My wife and I have decided to move across E...
25      <p>I'm looking for data plans I can use while ...
```

```
print('Cleaned')
print(cleaned_data)
```

```
Cleaned
id
1       my fianc e and i are looking for a good caribb...
2       this was one of our definition questions  but ...
4       singapore airlines has an all business class f...
5       another definition question that interested me...
6       a year ago i was reading some magazine  and fo...
8       can anyone suggest the best way to get from se...
9       we are considering visiting argentina for up t...
10      recently my wife and i traveled to italy and b...
11      i m planning on taking the trans siberian   tr...
13      i need to travel from cusco  peru to la paz  b...
14      i am aware of travel agencies catering to us c...
15      i m planning my first international trip out o...
16      my wife and i have decided to move across euro...
25      i m looking for data plans i can use while tou...
```

Предобработка данных

Удалим стоп-слова и проведем токенизацию.

```
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize
#from nltk.stem import WordNetLemmatizer

stops = set(stopwords.words("english"))
```

```
def clean_stopwords_tokenize(content):
    #Токенизация
    content = content.apply(lambda x: word_tokenize(x))
    #удаление стоп-слов
    content = content.apply(lambda x: [i for i in x if i not in stops])
    return(content)
```

Предобработка данных

Удалим стоп-слова и проведем токенизацию.

```
final_data = clean_stopwords_tokenize(cleaned_data)
print(final_data)
```

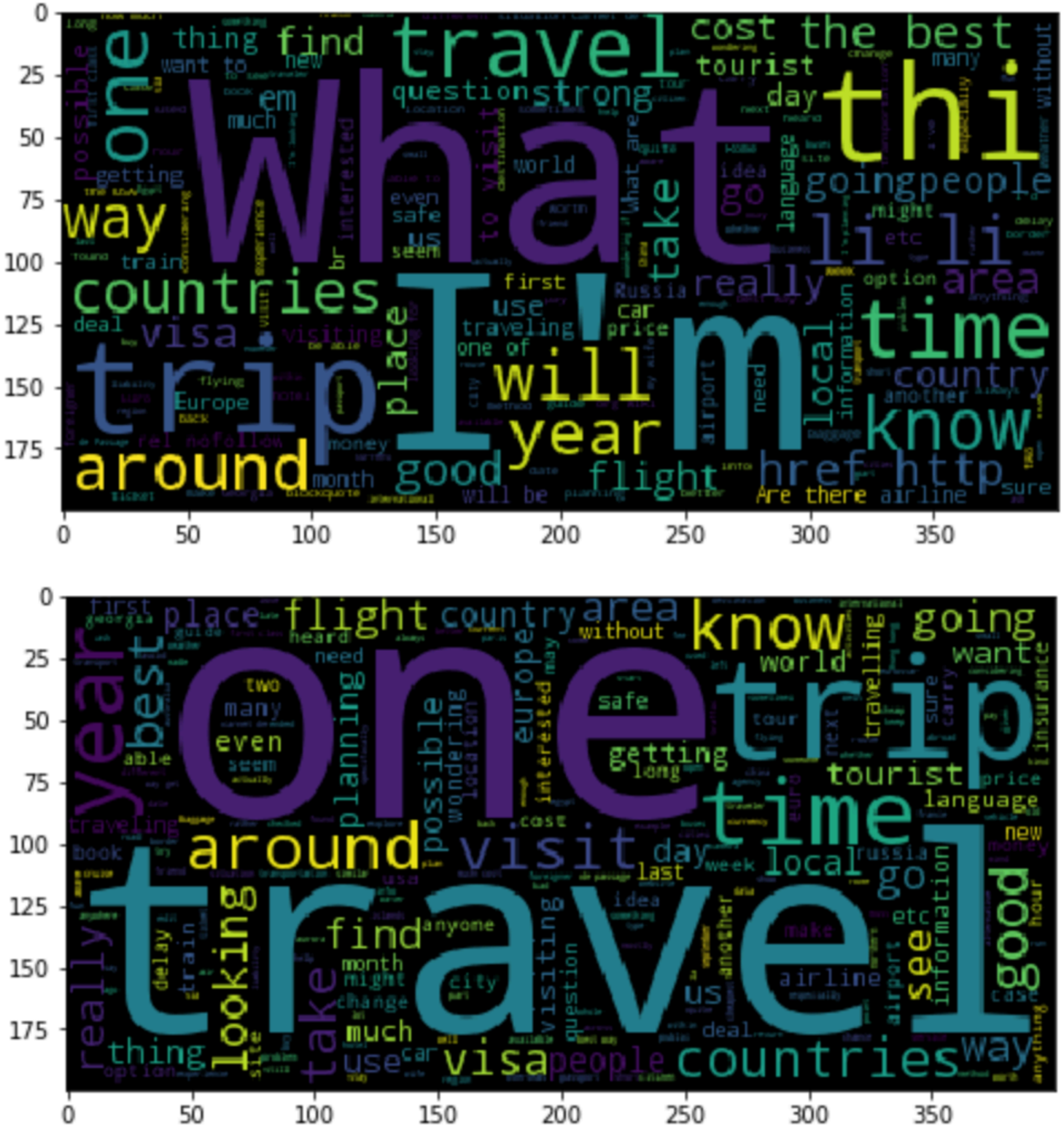
```
id
1       [fianc, e, looking, good, caribbean, cruise, o...
2       [one, definition, questions, also, one, intere...
4       [singapore, airlines, business, class, flight,...
5       [another, definition, question, interested, ea...
6       [year, ago, reading, magazine, found, availabi...
8       [anyone, suggest, best, way, get, seattle, tac...
9       [considering, visiting, argentina, ten, days, ...
10      [recently, wife, traveled, italy, left, got, 8...
11      [planning, taking, trans, siberian, trans, mon...
13      [need, travel, cusco, peru, la, paz, bolivia, ...
14      [aware, travel, agencies, catering, us, citize...
15      [planning, first, international, trip, u, fall...
16      [wife, decided, move, across, europe, train, k...
25      [looking, data, plans, use, touring, different...
26      [heard, rumours, things, difficult, time, find...
27      [traveling, one, favorite, things, simply, exp...
28      [possible, duplicate, best, ways, avoid, data,...
```

Предобработка данных

Построим облако слов по данным после предобработки.

```
#wordcloud
travel_text_cleaned = ' '
for x in final_data:
    for y in x:
        travel_text_cleaned+=' '+y

#print(travel_text_cleaned)
plt.figure(figsize=(8,10))
wc_new = WordCloud(max_words=1000,random_state=1).generate(travel_text_cleaned)
plt.imshow(wc_new)
plt.show()
```

Облако слов



ДО предобработки

ПОСЛЕ предобработки

Предобработка данных

Выполним поиск ближайшего текста после предобработки.

```
#поиск ближайшего нормированного текста после предобработки
flat_list = []
for x in final_data:
    x_new = ''
    for y in x:
        x_new +=' '+y
    x = x_new
    flat_list.append(x)

print(flat_list)
```

```
[' fianc e looking good caribbean cruise october wondering islands best see cruise line take seems like lot cruises r
un month due hurricane season looking good options edit travelling 2012', ' one definition questions also one interes
ts personally find guide take safely amazon jungle love explore amazon would attempt without guide least first time p
refer guide going ambush anything p edit want go anywhere touristy start end points open trip take places likely see
travellers tourists definitely require good guide order safe', ' singapore airlines business class flight ewr sin new
ark singapore seem find reward krisflyer flights dates', ' another definition question interested easiest transportat
ion use throughout romania foreigner plan visit point still relatively ignorant get particularly interested rural are
as mountains difficulty crossing also ignorant', ' year ago reading magazine found availability get trip antarctica u
```

Предобработка данных

Выполним поиск ближайшего текста после предобработки.

```
vectorizer_c = CountVectorizer(min_df=1)
X_train_c = vectorizer_c.fit_transform(list(flat_list))
#число сообщений, число слов
count_samples_c, count_features_c = X_train_c.shape
#print('titles_count=%d, words_count=%d' % (count_samples_c, count_features_c))

#сообщение-вопрос
request_2 = 'safely travel tour Amazon jungle'
request_vect_c = vectorizer_c.transform([request_2])
print(request_vect_c.toarray())
```

Предобработка данных

Выполним поиск ближайшего текста после предобработки.

```
found_publ_n_c = None
found_dist_n_c = sys.maxsize
found_i_n_c = None

for i in range(0, count_samples_c):
    publ = (list(flat_list))[i]
    if publ == request:
        continue
    publ_vect = X_train_c.getrow(i)
    cur_dist = dist_norm(publ_vect, request_vect_c)
    #печать всех расстояний
    #print('publ %i dist=%.2f: %s' % (i, cur_dist, publ))
    if cur_dist<found_dist_n_c:
        found_dist_n_c = cur_dist
        found_publ_n_c = publ
        found_i_n_c = i

print('\nBest publication is %i with dist=%.2f: %s' % (found_i_n_c, found_dist_n_c, found_publ_n_c))
```

```
Best publication is 1 with dist=1.25:  one definition questions also one interests personally find guide take safely amazon jungle love explore amazon would attempt without guide least first time prefer guide going ambush anything p e dit want go anywhere touristy start end points open trip take places likely see travellers tourists definitely requir e good guide order safe
```

Предобработка данных

ДО предобработки и ДО нормализаии векторов

```
Best publication is 44 with dist=4.12: <p>Are there travel sites constantly updated with safety tips and political si
tuation of countries?</p>
```

ДО предобработки и ПОСЛЕ нормализаии векторов

```
Best publication is 65 with dist=1.31: <p>Organizing your travel without agents is almost always cheaper and more fun
.
But does that also apply for Maldives? Can I just travel to the Maldives? Do you have some tips or experiences to rep
ort? </p>
```

ПОСЛЕ предобработки и нормализации векторов

```
Best publication is 1 with dist=1.25:  one definition questions also one interests personally find guide take safely
amazon jungle love explore amazon would attempt without guide least first time prefer guide going ambush anything p e
dit want go anywhere touristy start end points open trip take places likely see travellers tourists definitely requir
e good guide order safe
```

Задание 3

1. Определить необходимые этапы предобработки для ваших данных:
    1. разделение на предложения / токены;
    2. удаление стоп-слов;
    3. удаление знаков препинания, чисел, символов другого алфавита;
    4. удаление коротких / длинных слов;
    5. приведение к одному регистру;
    6. дополнительные методы обработки для ВАШИХ данных.
2. Выполнить поиск похожего текста с данными после предобработки.
3. Сравнить результаты ДО и ПОСЛЕ предобработки.