Кружок по искусственному интеллекту

Семинар 4

Организатор: Зубрихина Мария

План занятия

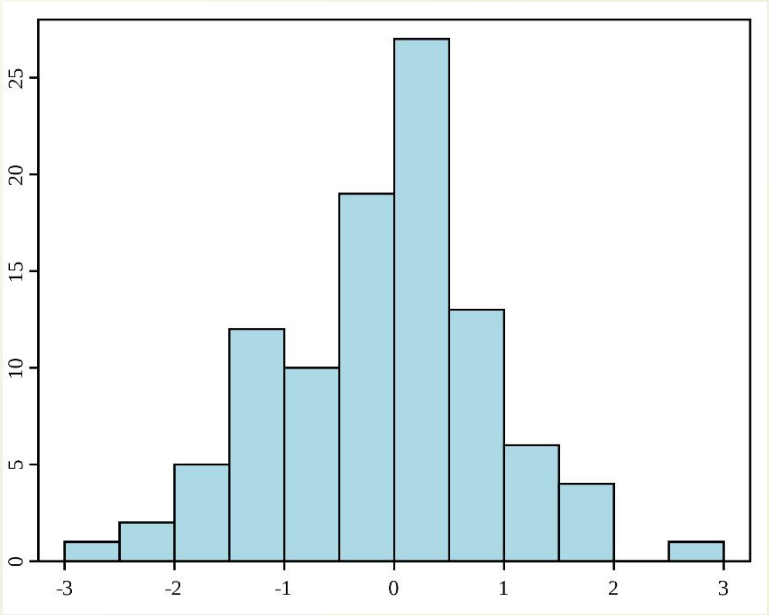
- Общие рекомендации по анализу данных
- Работа с текстовыми данными
- Анализ результатов

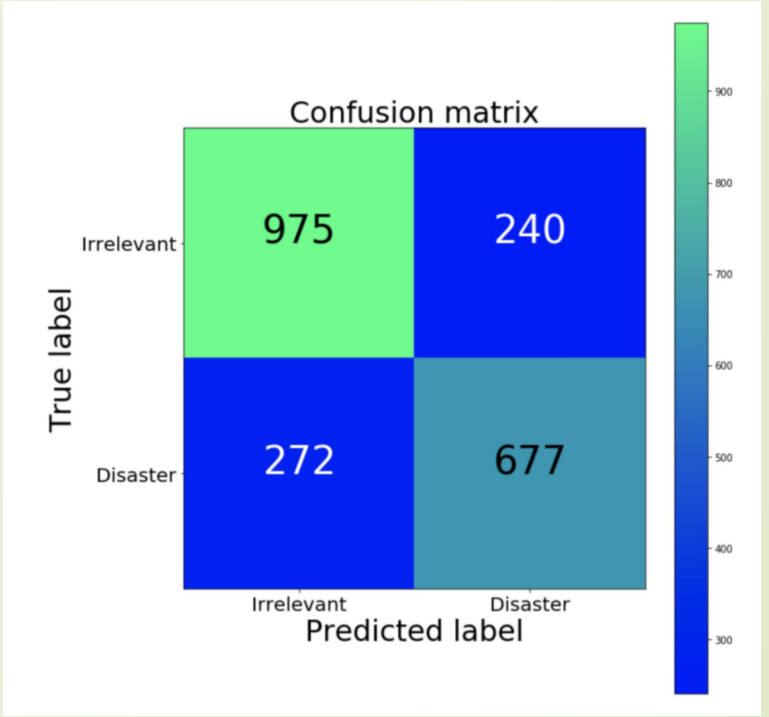
Обработка и анализ данных

- feature extraction and feature engineering – превращение данных, специфических для предметной области, в понятные для модели векторы
- feature transformation – трансформация данных для повышения точности алгоритма;
- feature selection – отсечение ненужных признаков

Обработка и анализ данных

- Построение матриц ошибок
- Построение гистограмм, анализ коррелирующих признаков,





Признаки

- Вещественные (Возраст, площадь квартиры)
- Бинарные (Доход клиента больше среднего по городу?)
- Порядковые (тип населенного пункта,размер одежды,образование)
- Категориальные (цвет глаз, город)

Label Encoder

	city	class	degree	income	city_le
0	Moscow	A	1	10.2	2
1	London	B	1	11.6	1
2	London	A	2	8.8	1
3	Kiev	A	2	9.0	0
4	Moscow	B	3	6.6	2
5	Moscow	B	3	10.0	2
6	Kiev	A	1	9.0	0
7	Moscow	A	1	7.2	2

```
from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()
le.fit(data.city)
data['city_le'] = le.transform(data.city)
data
```

```
# ручная альтернатива
# словарь для кодировки
dct = {'Kiev': 0, 'London': 1, 'Moscow': 2}
data['city_le'] = data['city'].map(dct)
data
```

One-hot-кодирование

```
from sklearn.preprocessing import OneHotEncoder
ohe = OneHotEncoder(sparse=False)
new_ohe_features = ohe.fit_transform(data.city_le.values.reshape(-1, 1))
tmp = pd.DataFrame(new_ohe_features, columns=['city=' + str(i) for i in range(new_ohe_features.shape[1])])
data = pd.concat([data, tmp], axis=1)
data
```

	city	class	degree	income	city=0	city=1	city=2
0	Moscow	A	1	10.2	0	0	1
1	London	B	1	11.6	0	1	0
2	London	A	2	8.8	0	1	0
3	Kiev	A	2	9.0	1	0	0
4	Moscow	B	3	6.6	0	0	1
5	Moscow	B	3	10.0	0	0	1
6	Kiev	A	1	9.0	1	0	0
7	Moscow	A	1	7.2	0	0	1

```
# ручной способ
def code_myohe(data, feature):
    for i in data[feature].unique():
        data[feature + '=' + i] =
            (data[feature] == i).astype(float)
code_myohe(data, 'city')
data
```

Другие способы кодирования

```
# функция возвращает значения нового признака
def code_mean(data, cat_feature, real_feature):
    return (data[cat_feature].map(data.groupby(cat_feature)[real_feature].mean()))

data['city_mean_income'] = code_mean(data, 'city', 'income')
data
```

	city	class	degree	income	city_mean_income
0	Moscow	A	1	10.2	8.5
1	London	B	1	11.6	10.2
2	London	A	2	8.8	10.2
3	Kiev	A	2	9.0	9.0
4	Moscow	B	3	6.6	8.5
5	Moscow	B	3	10.0	8.5
6	Kiev	A	1	9.0	9.0
7	Moscow	A	1	7.2	8.5

Обработка и анализ текстовых данных

- токенизация (nltk)
- приведение к одному регистру
- лемматизация (nltk, pymorphy)
- удаление нерелевантных слов (re)

Векторизация текста

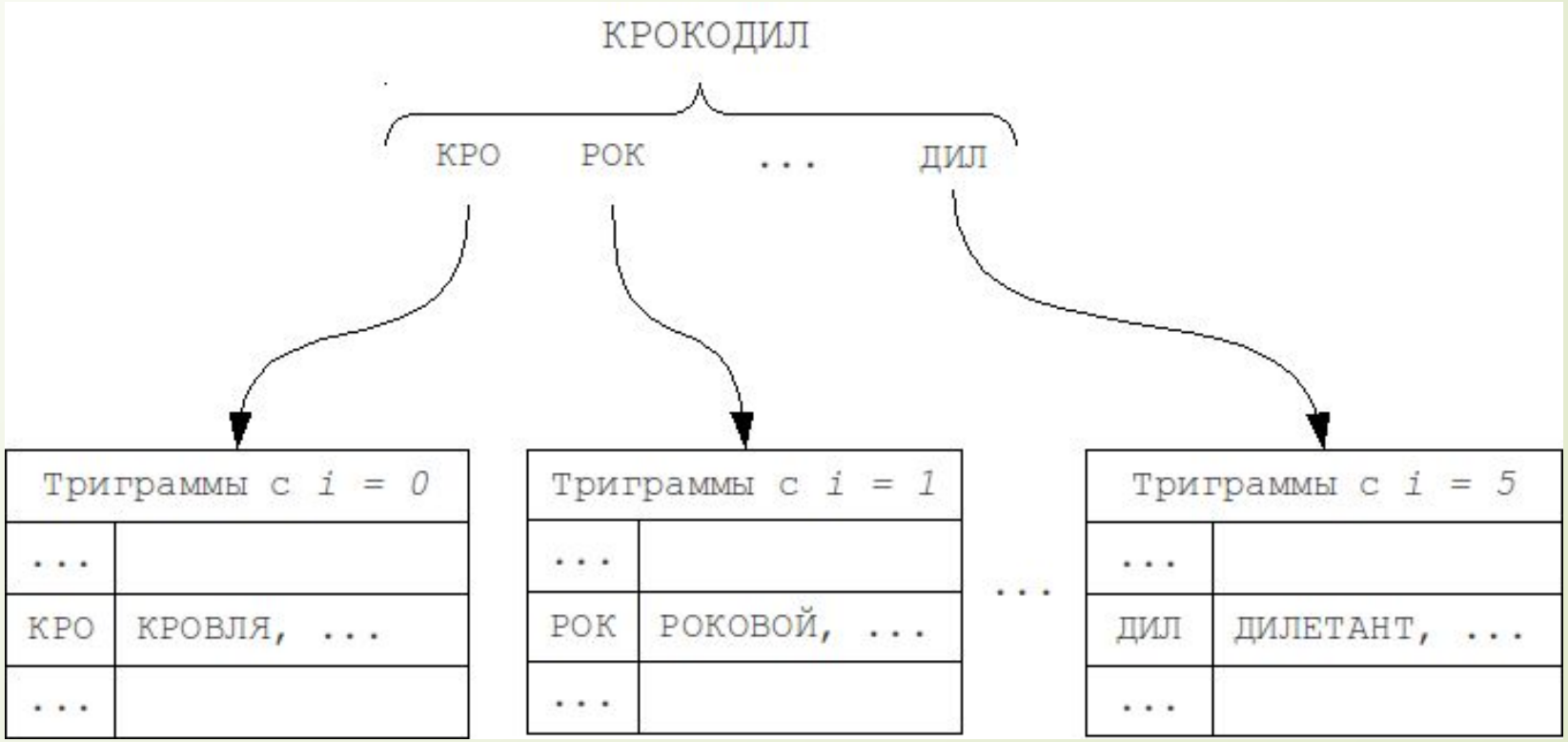
- Разбиение текста на слова и преобразование каждого слова в вектор
- Разбиение текста на символы и преобразование каждого символа в вектор
- Извлечение N-грамм и их преобразование в вектор

Преобазование токенов в векторы

- One-hot encoding(прямое кодирование слов и символов)
- One-hot hashing trick (прямое хеширование признаков)
- Embeddings (векторное представление слов) (Word2vec, Glove, Fasttext)

Bag of words («Мешок слов»)

N-граммы


КРОКОДИЛ
КРО
РОК
...
ДИЛ
Триграммы с i = 0
...
КРО
КРОВЛЯ, ...
...
Триграммы с i = 1
...
РОК
РОКОВОЙ, ...
...
...
Триграммы с i = 5
...
ДИЛ
ДИЛЕТАНТ, ...
...

TF-IDF

$$\mathrm{tf}(t,d) = \frac{n_t}{\sum_k n_k},$$

TF (*term frequency* — частота слова) — отношение числа вхождений некоторого слова к общему числу слов документа.

$$\mathrm{idf}(t,D) = \log \frac{|D|}{|\{\, d_i \in D \mid t \in d_i \,\}|},$$ [2]

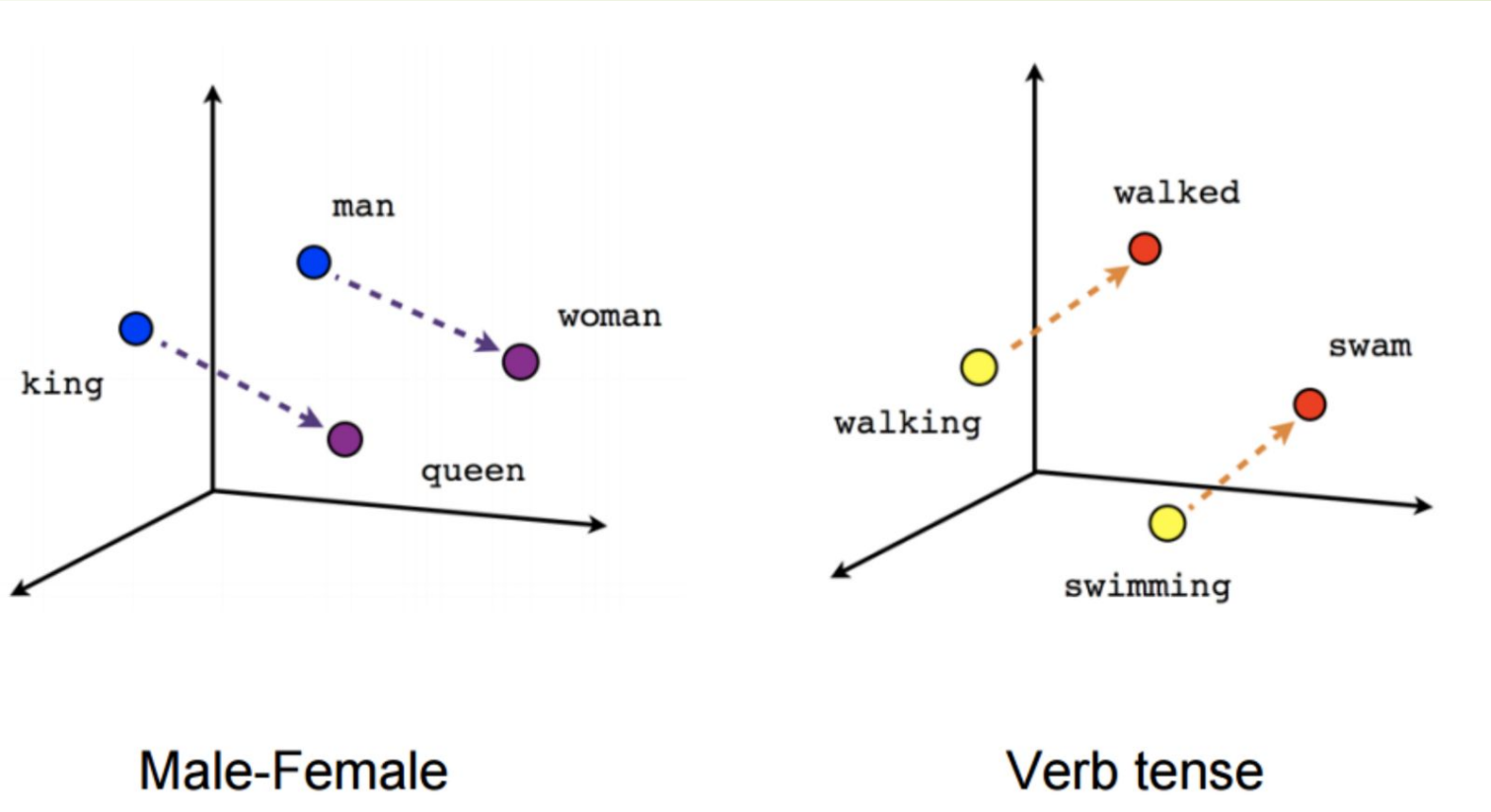
IDF (*inverse document frequency* — обратная частота документа) — инверсия частоты, с которой некоторое слово встречается в документах коллекции.

$$\text{tf-idf}(t,d,D) = \mathrm{tf}(t,d) \times \mathrm{idf}(t,D)$$

Embeddings

- Малоразмерные представления
- Геометрические отношения между векторами отражают семантические связи

Embeddings


man
woman
king
queen
Male-Female
walked
swam
walking
swimming
Verb tense

Спасибо за внимание!