```
!pip install pymystem3==0.1.10
#а то гугл коллаб отказывается лемматизировать
```

```
Collecting pymystem3==0.1.10
  Downloading pymystem3-0.1.10-py3-none-any.whl (10 kB)
Requirement already satisfied: requests in /usr/local/lib/python3.7/dist-packages (fr
Requirement already satisfied: urllib3!=1.25.0,!=1.25.1,<1.26,>=1.21.1 in /usr/local/
Requirement already satisfied: chardet<4,>=3.0.2 in /usr/local/lib/python3.7/dist-pac
Requirement already satisfied: idna<3,>=2.5 in /usr/local/lib/python3.7/dist-packages
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.7/dist-pa
Installing collected packages: pymystem3
  Attempting uninstall: pymystem3
    Found existing installation: pymystem3 0.2.0
    Uninstalling pymystem3-0.2.0:
      Successfully uninstalled pymystem3-0.2.0
Successfully installed pymystem3-0.1.10
```

```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
import seaborn as sns
from tqdm.auto import tqdm, trange
from pymystem3 import Mystem
import nltk
from nltk.stem import *
from nltk.corpus import stopwords
from string import punctuation
from nltk.tokenize import word_tokenize
from nltk.stem.snowball import SnowballStemmer
stemmer = SnowballStemmer("russian")
nltk.download('stopwords')
nltk.download('punkt')
from sklearn.naive_bayes import MultinomialNB
from sklearn.pipeline import Pipeline
from sklearn.feature_extraction.text import TfidfTransformer
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.metrics import accuracy_score
```

```
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Unzipping corpora/stopwords.zip.
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Unzipping tokenizers/punkt.zip.
```

```
#монтируем с гугл диска чтоб не закачивать постоянно
from google.colab import drive
drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

```
df=pd.read_csv('/content/drive/MyDrive/Colab Notebooks/20170301.csv')
```

1. изучение датасета новостей

Далее познакомимся с данными, со структурой датасета, посмотрим пропуски

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 576383 entries, 0 to 576382
Data columns (total 8 columns):
 #   Column       Non-Null Count   Dtype 
---  ------       --------------   ----- 
 0   u_id         576383 non-null  int64 
 1   provider     576383 non-null  object
 2   date_time    576383 non-null  object
 3   title        576383 non-null  object
 4   description  542805 non-null  object
 5   link         576292 non-null  object
 6   pubdate      576383 non-null  object
 7   numfield     576383 non-null  int64 
dtypes: int64(2), object(6)
memory usage: 35.2+ MB
```

в поле описания имеется очень много пропусков, здесь логично встсавить туда хотя бы краткое описание title

```
df.head()
```

	u_id	provider	date_time	title	description	
0	103451	washingtonpostcom_world[eng]	2017-01-17 20:15:49	"$10,000 stuffed in a diplomat-s car." Moscow ...	Russia-s foreign minister said that United Sta...	htt
1	211367	vsesmiru_business	2017-02-02 06:00:33	"100 друзей" Гродненского мясокомбината	Дизайнеры агентства Fabula Branding (Минск) пр...	htt
2	13559	mailru_common	2016-11-28 13:30:23	"12-я партия - игра жизни не только Карякина, ...	<p>Победитель шахматной олимпиады 1998 гроссме...	
			2016-11-	"37 мне только	<p>Боксёры	

```
df['description']=df['description'].fillna(df['title'])
```

```
df=df[df['link'].str.contains("business") | df['link'].str.contains("finance") | df['link'
```

```
def detect_ru(row):
  alphabet = {"а","б","в","г","д","е","ё","ж","з","и","й","к","л","м","н","о",
              "п","р","с","т","у","ф","х","ц","ч","ш","щ","ъ","ы","ь","э","ю","я"}

  text = row['description']
  return bool(alphabet.intersection(set(text.lower())))



df['Languagereveiw'] = df.apply(detect_ru, axis=1)


df=df[df['Languagereveiw']==True·].reset_index(drop=True)
df.head()
```

	u_id	provider	date_time	title	description	
0	211367	vsesmiru_business	2017-02-02 06:00:33	"100 друзей" Гродненского мясокомбината	Дизайнеры агентства Fabula Branding (Минск) пр...	ht
1	353361	vsesmiru_business	2017-02-24 09:00:34	"ArcelorMittal Кривой Рог" инициировал антидем...	В течение 30 дней, с даты публикации сообщения...	ht
2	65369	newrucom_common	2016-12-05 20:30:06	"Абсолютно никчемный аргумент": Путин прокомме...	Глава государства назвал совершенно не имеющей...	htt
3	58879	vsesmiru_business	2017-01-11 07:45:35	"Аврора" со следующей недели открывает новый р...	Интерфакс-Россия, Новость:\nАвиакомпания "Авро...	ht

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 37243 entries, 0 to 37242
Data columns (total 9 columns):
 #   Column          Non-Null Count  Dtype 
---  ------          --------------  ----- 
 0   u_id            37243 non-null  int64 
 1   provider        37243 non-null  object
 2   date_time       37243 non-null  object
 3   title           37243 non-null  object
 4   description     37243 non-null  object
 5   link            37243 non-null  object
 6   pubdate         37243 non-null  object
 7   numfield        37243 non-null  int64 
 8   Languagereveiw  37243 non-null  bool  
dtypes: bool(1), int64(2), object(6)
memory usage: 2.3+ MB
```

```
df.duplicated().sum()
```

```
0
```

```
#df=df[:2150]
```

```
dout=df['description']
```

```
dout.duplicated().sum()
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-45-efae4ec9cc04> in <module>()
----> 1 dout.duplicated().sum()

NameError: name 'dout' is not defined

 SEARCH STACK OVERFLOW
```

```
dout = dout.drop_duplicates()
```

```
dout.describe()
```

```
dout.to_csv("descr.csv", index=False)
```

```
dout.head(15)
```

```
from google.colab import files
files.download("descr.csv")
```

!!!СЕЙЧАС ТУТ и РАЗМЕЧАЮ вручную 2000 примеров, беру только русские новости, иначе нереал

---

2. Подготовка обучающего датасета

```
df_train=pd.read_csv('/content/drive/MyDrive/Colab Notebooks/descr_labeled.csv',sep='I',er
```

```
df_train.head(10)
```

	description	label
0	Дизайнеры агентства Fabula Branding (Минск) пр...	1
1	В течение 30 дней, с даты публикации сообщения...	2
2	Глава государства назвал совершенно не имеющей...	2
3	Интерфакс-Россия, Новость: Авиакомпания "Аврор...	3
4	Отечественный концерн по производству автомоби...	4
5	"АвтоВАЗ" начал продажи автомобилей Lada Vesta...	3
6	Совет директоров "АвтоВАЗа" принял стратегичес...	4
7	"АвтоВАЗ" завершил размещение дополнительных а...	3

```
df_train.tail(10)
```

	description	label
2059	Шведский концерн IKEA, которому принадлежит кр...	3
2060	Шведская компания IKEA выплатит $50 млн в каче...	2
2061	Компания IRI Investments Lietuva, контролируем...	3
2062	Краснинский суд Смоленской области арестовал с...	2
2063	Шведская компания IKEA не согласна с решением ...	2
2064	Шведская IKEA не будет строить торговый центр ...	3
2065	Шведская IKEA планирует выставить права на дол...	3
2066	Шведская компания IKEA намерена трудоустроить ...	3
2067	Шведский ритейлер IKEA направил обращение упол...	2
2068	Каждый пятый товар сети подешевеет на 15–20%	5

разметили вручную новости в программе CSVpad

метки ставятся следующим образом

1 - реклама, позитивное ожидание чего-либо

2 - судебные дела, иски, претензии

3 - информационное сообщение нейтральное по смыслу

4 - рост продаж, производства, поставок - позитив по сути

5 - уменьшение чего - либо, продаж, поставок и т.д. - негатив короче

```
df_train['label'].value_counts()
```

```
3    1125
```

```
1     464
2     213
4     155
5     112
Name: label, dtype: int64
```

ок, теперь есть столбец текста новости и столбец метки новости! можно двигаться дальше

---

3. Очищение текста новости

удаляем стоп слова, обрабатываем текст

```
import string
def remove_punctuation(text):
    return "".join([ch if ch not in string.punctuation else ' ' for ch in text])

def remove_numbers(text):
    return ''.join([i if not i.isdigit() else ' ' for i in text])

import re
def remove_multiple_spaces(text):
  return re.sub(r'\s+', ' ', text, flags=re.I)


mystem = Mystem()

russian_stopwords = stopwords.words("russian")
russian_stopwords.extend(['…', '«', '»', '...'])
def lemmatize_text(text):
    tokens = mystem.lemmatize(text.lower())
    tokens = [token for token in tokens if token not in russian_stopwords and token != " "
    text = " ".join(tokens)
    return text
```

```
 Installing mystem to /root/.local/bin/mystem from http://download.cdn.yandex.net/mys
```

```
#df_train=df.reset_index(drop = True)


preproccessing = lambda text: (remove_multiple_spaces(remove_numbers(remove_punctuation(te
df_train['preproccessed'] = list(map(preproccessing, df_train['description']))
```

очистили текст от пунктуации, от пробелов, от цифр, и переведем в нижний регистр

```
prep_text = [remove_multiple_spaces(remove_numbers(remove_punctuation(text.lower()))) for
```

```
100%                                                2069/2069 [00:00<00:00, 4249.64it/s]
```

```
len(prep_text)
prep_text[0]
```

```
'дизайнеры агентства fabula branding минск провели комплексную разработку торговой м
арки колбасных изделий « друзей» нейминг логотип дизайн упаковки для оао «гродненски
й мясокомбинат» продукт – колбасные изделия среднего ценового сегмента сырокопченые
сыровяленые вареные колбасы сосиски и сардельки регионы продаж – беларусь и россия с
итуации потребления дружеские и семейные застолья пикник гости быстрый перекус целев
ая аудитория – мужчины и женщины лет решением стал теплый и яркий желтый цвет которы
```

```
df_train['text_prep'] = prep_text
```

```
df_train.head()
```

	description	label	preproccessed	text_prep
0	Дизайнеры агентства Fabula Branding (Минск) пр...	1	Дизайнеры агентства Fabula Branding Минск пров...	дизайнеры агентства fabula branding минск пров...
1	В течение 30 дней, с даты публикации сообщения...	2	В течение дней с даты публикации сообщения в У...	в течение дней с даты публикации сообщения в у...
2	Глава государства назвал	2	Глава государства назвал совершенно не	глава государства назвал совершенно не

4. Стэмминг

```
russian_stopwords = stopwords.words("russian")
russian_stopwords.extend(['…', '«', '»', '...', 'т.д.', 'т', 'д'])
```

```
text = df_train['text_prep'][1]
word_tokenize(text)
```

```
stemmed_texts_list = []
for text in tqdm(df_train['text_prep']):
    tokens = word_tokenize(text)
    stemmed_tokens = [stemmer.stem(token) for token in tokens if token not in russian_stop
    text = " ".join(stemmed_tokens)
    stemmed_texts_list.append(text)

df_train['text_stem'] = stemmed_texts_list
```

```
100%                                                2069/2069 [00:04<00:00, 523.81it/s]
```

```
def remove_stop_words(text):
    tokens = word_tokenize(text)
```

```
    tokens = [token for token in tokens if token not in russian_stopwords and token != ' '
    return " ".join(tokens)


sw_texts_list = []
for text in tqdm(df_train['text_prep']):
    tokens = word_tokenize(text)
    tokens = [token for token in tokens if token not in russian_stopwords and token != ' '
    text = " ".join(tokens)
    sw_texts_list.append(text)

df_train['text_sw'] = sw_texts_list


df_train.head()
```

	description	label	preproccessed	text_prep	text_stem	text_sw
0	Дизайнеры агентства Fabula Branding (Минск) пр...	1	Дизайнеры агентства Fabula Branding Минск пров...	дизайнеры агентства fabula branding минск пров...	дизайнер агентств fabul branding минск провел ...	дизайнеры агентства fabula branding минск пров...
1	В течение 30 дней, с даты публикации сообщения...	2	В течение дней с даты публикации сообщения в У...	в течение дней с даты публикации сообщения в у...	течен дне дат публикац сообщен урядов курьер м...	течение дней даты публикации сообщения урядово...
	Глава		Глава	глава	глав государств	глава

5. Лемматизация

```
df_train.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2069 entries, 0 to 2068
Data columns (total 6 columns):
 #   Column         Non-Null Count  Dtype 
---  ------         --------------  ----- 
 0   description    2069 non-null   object
 1   label          2069 non-null   int64 
 2   preproccessed  2069 non-null   object
 3   text_prep      2069 non-null   object
 4   text_stem      2069 non-null   object
 5   text_sw        2069 non-null   object
dtypes: int64(1), object(5)
memory usage: 97.1+ KB
```

```
lemm_texts_list = []
for text in tqdm(df_train['text_sw']):
    #print(text)
    try:
```

```
            text_lem = mystem.lemmatize(text)
            tokens = [token for token in text_lem if token != ' ' and token not in russian_sto
            text = " ".join(tokens)
            lemm_texts_list.append(text)
        except Exception as e:
            print(e)

df_train['text_lemm'] = lemm_texts_list
```

```
100%                                                2069/2069 [00:03<00:00, 578.80it/s]
```

```
df_train.tail(20)
```

	description	label	preprocessed	text_prep	text_s
2049	От Ассоциации предприятий информационных техно...	3	От Ассоциации предприятий информационных техно...	от ассоциации предприятий информационных техно...	accou предпр информац техно укра
2050	IBU временно отстранил от соревнований российс...	2	IBU временно отстранил от соревнований российс...	ibu временно отстранил от соревнований российс...	ib вре отст соревно росси биатло
2051	АйСиБиСи Банк (100% "дочка" ...	3	АйСиБиСи Банк amp quot дочка amp quot крупнейш...	айсибиси банк amp quot дочка amp quot крупнейш...	айсибис б amp quot д amp крупн ба
2052	В феврале 2017 года состоится первая встреча с...	3	В феврале года состоится первая встреча с поте...	в феврале года состоится первая встреча с поте...	феврал состо г вст потенциа инв

```
df_train.to_csv('df_train_prep.csv', index=False)
```

```
from google.colab import files
files.download("df_train_prep.csv")
```

Чтобы изменить содержимое ячейки, дважды нажмите на нее (или выберите "Ввод")

```
X = df_train['text_sw']
y = df_train['label']
```

разобьем обучающий датасет на тренировку и тестирование

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state = 42
```

```
X[0]
```

```
'дизайнеры агентства fabula branding минск провели комплексную разработку торговой м
арки колбасных изделий друзей нейминг логотип дизайн упаковки оао гродненский мясоко
мбинат продукт – колбасные изделия среднего ценового сегмента сырокопченые сыровялен
ые вареные колбасы сосиски сардельки регионы продаж – беларусь россия ситуации потре
бления дружеские семейные застолья пикник гости быстрый перекус целевая аудитория –
мужчины женщины лет решением стал теплый яркий желтый цвет который отстраивает проду
```

```
y[0]
```

```
1
```

кр... принадлежит принадлежит принад

6. Байесовский классификатор

2060 выплатит $50 млн в каче 2 выплатит млн в выплатит млн в выплат

```
nb = Pipeline([('vect', CountVectorizer()),
               ('tfidf', TfidfTransformer()),
               ('clf', MultinomialNB()),
              ])
```

```
%%time
nb.fit(X_train, y_train)
```

```
CPU times: user 81.4 ms, sys: 1.83 ms, total: 83.2 ms
Wall time: 87.3 ms
Pipeline(steps=[('vect', CountVectorizer()), ('tfidf', TfidfTransformer()),
                ('clf', MultinomialNB())])
```

```
%%time
from sklearn.metrics import classification_report
y_pred = nb.predict(X_test)
```

```
CPU times: user 24.5 ms, sys: 0 ns, total: 24.5 ms
Wall time: 24.7 ms
```

```
y_pred[0]
```

```
3
```

```
my_tags = df_train['label'].unique().astype('str')
my_tags
```

```
array(['1', '2', '3', '4', '5'], dtype='<U21')
```

```
print('accuracy %s' % accuracy_score(y_pred, y_test))
print(classification_report(y_test, y_pred, target_names=my_tags))
```

```
accuracy 0.5684380032206119
              precision    recall  f1-score   support

           1       0.88      0.10      0.19       143
           2       1.00      0.03      0.06        68
           3       0.56      1.00      0.72       335
           4       0.33      0.03      0.05        38
           5       0.00      0.00      0.00        37

    accuracy                           0.57       621
   macro avg       0.55      0.23      0.20       621
weighted avg       0.63      0.57      0.44       621

/usr/local/lib/python3.7/dist-packages/sklearn/metrics/_classification.py:1318: Undef
```

```
  _warn_prf(average, modifier, msg_start, len(result))
/usr/local/lib/python3.7/dist-packages/sklearn/metrics/_classification.py:1318: Undef
  _warn_prf(average, modifier, msg_start, len(result))
/usr/local/lib/python3.7/dist-packages/sklearn/metrics/_classification.py:1318: Undef
  _warn_prf(average, modifier, msg_start, len(result))
```

Linear Support Vector Machine

```
from sklearn.linear_model import SGDClassifier

sgd = Pipeline([('vect', CountVectorizer()),
                ('tfidf', TfidfTransformer()),
                ('sgd', SGDClassifier(loss='hinge', penalty='l2',alpha=1e-3, random_state=
               ])
```

```
%%time
sgd.fit(X_train, y_train)
```

```
CPU times: user 145 ms, sys: 35.1 ms, total: 180 ms
Wall time: 233 ms
Pipeline(steps=[('vect', CountVectorizer()), ('tfidf', TfidfTransformer()),
                ('sgd',
                 SGDClassifier(alpha=0.001, max_iter=5, random_state=42,
                               tol=None))])
```

```
%%time
y_pred = sgd.predict(X_test)
```

```
CPU times: user 33.7 ms, sys: 1.97 ms, total: 35.7 ms
Wall time: 105 ms
```

```
print('accuracy %s' % accuracy_score(y_pred, y_test))
print(classification_report(y_test, y_pred,target_names=my_tags))
```

```
accuracy 0.6280193236714976
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-39-41d3cdcf3700> in <module>()
      1 print('accuracy %s' % accuracy_score(y_pred, y_test))
----> 2 print(classification_report(y_test, y_pred,target_names=my_tags))

NameError: name 'classification_report' is not defined
```

SEARCH STACK OVERFLOW

Случайный лес

```
from sklearn.ensemble import RandomForestClassifier
```

```
rf_model = Pipeline([('vect', CountVectorizer()),
                ('tfidf', TfidfTransformer()),
                ('rf', RandomForestClassifier()),
               ])

%%time
rf_model.fit(X_train, y_train)
```

```
CPU times: user 1.07 s, sys: 9.08 ms, total: 1.08 s
Wall time: 1.08 s
Pipeline(steps=[('vect', CountVectorizer()), ('tfidf', TfidfTransformer()),
                ('rf', RandomForestClassifier())])
```

```
print('accuracy %s' % accuracy_score(y_pred, y_test))
print(classification_report(y_test, y_pred,target_names=my_tags))
```

```
accuracy 0.6280193236714976
              precision    recall  f1-score   support

           1       0.59      0.22      0.32       143
           2       0.84      0.38      0.53        68
           3       0.63      0.96      0.76       335
           4       0.38      0.24      0.29        38
           5       0.60      0.08      0.14        37

    accuracy                           0.63       621
   macro avg       0.61      0.38      0.41       621
weighted avg       0.63      0.63      0.57       621
```

Logistic Regression

```
from sklearn.linear_model import LogisticRegression

logreg = Pipeline([('vect', CountVectorizer()),
                ('tfidf', TfidfTransformer()),
                ('logreg', LogisticRegression(n_jobs=1, C=1e5)),
               ])
```

```
%%time
logreg.fit(X_train, y_train)
```

```
CPU times: user 1.53 s, sys: 1.67 s, total: 3.2 s
Wall time: 1.74 s
/usr/local/lib/python3.7/dist-packages/sklearn/linear_model/_logistic.py:818: Converg
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression
  extra_warning_msg=_LOGISTIC_SOLVER_CONVERGENCE_MSG,
```

```
Pipeline(steps=[('vect', CountVectorizer()), ('tfidf', TfidfTransformer()),
                ('logreg', LogisticRegression(C=100000.0, n_jobs=1))])
```

```
%%time
y_pred = logreg.predict(X_test)
```

```
CPU times: user 33.8 ms, sys: 25.5 ms, total: 59.3 ms
Wall time: 29.9 ms
```

```
print('accuracy·%s'·%·accuracy_score(y_pred,·y_test))
print(classification_report(y_test,·y_pred,target_names=my_tags))
```

```
accuracy 0.642512077294686
              precision    recall  f1-score   support

           1       0.55      0.36      0.43       143
           2       0.79      0.46      0.58        68
           3       0.67      0.87      0.76       335
           4       0.42      0.50      0.46        38
           5       0.67      0.16      0.26        37

    accuracy                           0.64       621
   macro avg       0.62      0.47      0.50       621
weighted avg       0.64      0.64      0.62       621
```

Чтобы изменить содержимое ячейки, дважды нажмите на нее (или выберите "Ввод")

```
X_train[0]
```

```
'дизайнеры агентства fabula branding минск провели комплексную разработку торговой м
арки колбасных изделий друзей нейминг логотип дизайн упаковки оао гродненский мясоко
мбинат продукт – колбасные изделия среднего ценового сегмента сырокопченые сыровялен
ые вареные колбасы сосиски сардельки регионы продаж – беларусь россия ситуации потре
бления дружеские семейные застолья пикник гости быстрый перекус целевая аудитория –
мужчины женщины лет решением стал теплый яркий желтый цвет который отстраивает проду
```

```
y_pred[0]
```

```
3
```

```
travel_text = '''отечественный концерн производству автомобилей отчитался росте продаж янв
```

```
grow_up = remove_multiple_spaces(remove_numbers(remove_punctuation(travel_text.lower())))
grow_up = remove_stop_words(travel_text)
```

```
pred = logreg.predict([grow_up])
pred
```

```
array([4])
```

было верно предсказана категория 4 - то есть новости об увеличении экономических показателей

```
sud_text='течение дней даты публикации сообщения урядовом курьере министерство проводить р


sud = remove_multiple_spaces(remove_numbers(remove_punctuation(sud_text.lower())))
sud = remove_stop_words(travel_text)


pred = logreg.predict([sud])
pred
```

```
array([4])
```

Кластеризация

```
texts = df_train['description']
type(texts)
```

```
pandas.core.series.Series
```

```
def token_and_stem(text):
    tokens = [word for sent in nltk.sent_tokenize(text) for word in nltk.word_tokenize(sen
    filtered_tokens = []
    for token in tokens:
        if re.search('[а-яА-Я]', token):
            filtered_tokens.append(token)
    stems = [stemmer.stem(t) for t in filtered_tokens]
    return stems


stopwords = nltk.corpus.stopwords.words('russian')
#можно расширить список стоп-слов
stopwords.extend(['что', 'это', 'так', 'вот', 'быть', 'как', 'в', 'к', 'на'])

from sklearn.feature_extraction.text import TfidfVectorizer, CountVectorizer
from sklearn.feature_extraction.text import TfidfVectorizer, CountVectorizer

n_featur=200000
tfidf_vectorizer = TfidfVectorizer(max_df=0.8, max_features=10000,
                                   min_df=0.01, stop_words=stopwords,
                                   use_idf=True, tokenizer=token_and_stem, ngram_range=(1,3)


%%time
tfidf_matrix = tfidf_vectorizer.fit_transform(texts)
```

```
/usr/local/lib/python3.7/dist-packages/sklearn/feature_extraction/text.py:401: UserWa
  % sorted(inconsistent)
CPU times: user 5.07 s, sys: 25.7 ms, total: 5.09 s
Wall time: 5.1 s
```

```
type(tfidf_matrix)
tfidf_matrix
```

```
<2069x462 sparse matrix of type '<class 'numpy.float64'>'
        with 22658 stored elements in Compressed Sparse Row format>
```

```
num_clusters = 5

# Метод к-средних - KMeans
from sklearn.cluster import KMeans
km = KMeans(n_clusters=num_clusters)
```

```
%%time
km.fit(tfidf_matrix)
```

```
CPU times: user 573 ms, sys: 15.8 ms, total: 589 ms
Wall time: 322 ms
KMeans(n_clusters=5)
```

```
%%time
idx = km.fit(tfidf_matrix)
clusters = km.labels_.tolist()
```

```
CPU times: user 628 ms, sys: 9.28 ms, total: 638 ms
Wall time: 338 ms
```

```
len(km.labels_)
clusters[:10]
```

```
[1, 1, 1, 1, 1, 1, 0, 1, 0, 1]
```

```
clusterkm = km.labels_.tolist()
frame = pd.DataFrame(texts)

#k-means
out = { 'text': texts, 'cluster': clusterkm, 'topic': df_train['label'] }
frame1 = pd.DataFrame(out, columns = ['text', 'cluster', 'topic'])
```

```
frame1.head(10)
```

	text	cluster	topic
0	Дизайнеры агентства Fabula Branding (Минск) пр...	1	1
1	В течение 30 дней, с даты публикации сообщения...	1	2
2	Глава государства назвал совершенно не имеющей...	1	2
3	Интерфакс-Россия, Новость: Авиакомпания "Аврор...	1	3
4	Отечественный концерн по производству автомоби...	1	4

```
frame1.tail(10)
```

	text	cluster	topic
2059	Шведский концерн IKEA, которому принадлежит кр...	1	3
2060	Шведская компания IKEA выплатит $50 млн в каче...	0	2
2061	Компания IRI Investments Lietuva, контролируем...	0	3
2062	Краснинский суд Смоленской области арестовал с...	3	2
2063	Шведская компания IKEA не согласна с решением ...	3	2
2064	Шведская IKEA не будет строить торговый центр ...	1	3
2065	Шведская IKEA планирует выставить права на дол...	1	3
2066	Шведская компания IKEA намерена трудоустроить ...	0	3
2067	Шведский ритейлер IKEA направил обращение упол...	1	2
2068	Каждый пятый товар сети подешевеет на 15–20%	1	5

```
frame1.describe()
```

	cluster	topic
count	2069.000000	2069.000000
mean	1.085549	2.631706
std	0.942203	1.075344
min	0.000000	1.000000
25%	1.000000	2.000000
50%	1.000000	3.000000
75%	1.000000	3.000000
max	4.000000	5.000000

Биграммы, триграммы и прочее

```
from __future__ import unicode_literals
import nltk
from nltk import word_tokenize
from nltk.util import ngrams
from collections import Counter


def main_words1(row):
      return nltk.word_tokenize(row['description'])# токинезация текста i-го документа


words1=df_train.apply(main_words1,axis=1)
words1
```

```
0       [Дизайнеры, агентства, Fabula, Branding, (, Ми...
1       [В, течение, 30, дней, ,, с, даты, публикации,...
2       [Глава, государства, назвал, совершенно, не, и...
3       [Интерфакс-Россия, ,, Новость, :, Авиакомпания...
4       [Отечественный, концерн, по, производству, авт...
                              ...
2145    [Шведская, IKEA, не, будет, строить, торговый,...
2146    [Шведская, IKEA, планирует, выставить, права, ...
2147    [Шведская, компания, IKEA, намерена, трудоустр...
2148    [Шведский, ритейлер, IKEA, направил, обращение...
2149    [Каждый, пятый, товар, сети, подешевеет, на, 1...
Length: 2150, dtype: object
```

```
ww=[]
for i in range(len(words1)):
  if type(words1[i])==list:
     ww=ww+words1[i]
word = list(filter(lambda x: x  != 'quot', ww))
word_ws=[w.lower()  for w in   word if w.isalpha() ]#исключение слов и символов
token=[w for w in word_ws if w not in russian_stopwords ]#нижний регистр

bigrams = ngrams(token,2)
trigrams = ngrams(token,3)
fourgrams = ngrams(token,4)
fivegrams = ngrams(token,5)


Counter(bigrams).most_common(5)
```

```
[(('говорится', 'сообщении'), 65),
 (('млрд', 'рублей'), 63),
 (('уровне', 'баррель'), 62),
 (('deutsche', 'bank'), 53),
 (('млрд', 'руб'), 47)]
```

```
Counter(trigrams).most_common(5)
```

```
[(('фьючерсы', 'brent', 'торговались'), 35),
 (('сообщает', 'rns', 'ссылкой'), 25),
 (('бирже', 'ice', 'futures'), 24),
 (('рейтинговое', 'агентство', 'fitch'), 24),
 (('сша', 'дональда', 'трампа'), 23)]
```

```
Counter(fourgrams).most_common(5)
```

```
[(('лондонской', 'бирже', 'ice', 'futures'), 23),
 (('президента', 'сша', 'дональда', 'трампа'), 22),
 (('международное', 'рейтинговое', 'агентство', 'fitch'), 21),
 (('фьючерсы', 'brent', 'торговались', 'лондоне'), 20),
 (('brent', 'торговались', 'лондоне', 'уровне'), 20)]
```

```
Counter(fivegrams).most_common(5)
```

```
[(('фьючерсы', 'brent', 'торговались', 'лондоне', 'уровне'), 20),
 (('brent', 'торговались', 'лондоне', 'уровне', 'баррель'), 20),
 (('электронных', 'торгах', 'товарной', 'биржи', 'nymex'), 15),
 (('общий', 'объем', 'продажи', 'иностранной', 'валюты'), 14),
 (('объем', 'продажи', 'иностранной', 'валюты', 'долларовом'), 14)]
```

BigARTM

```
text=df_train['description']
freq=nltk.FreqDist(text)
```

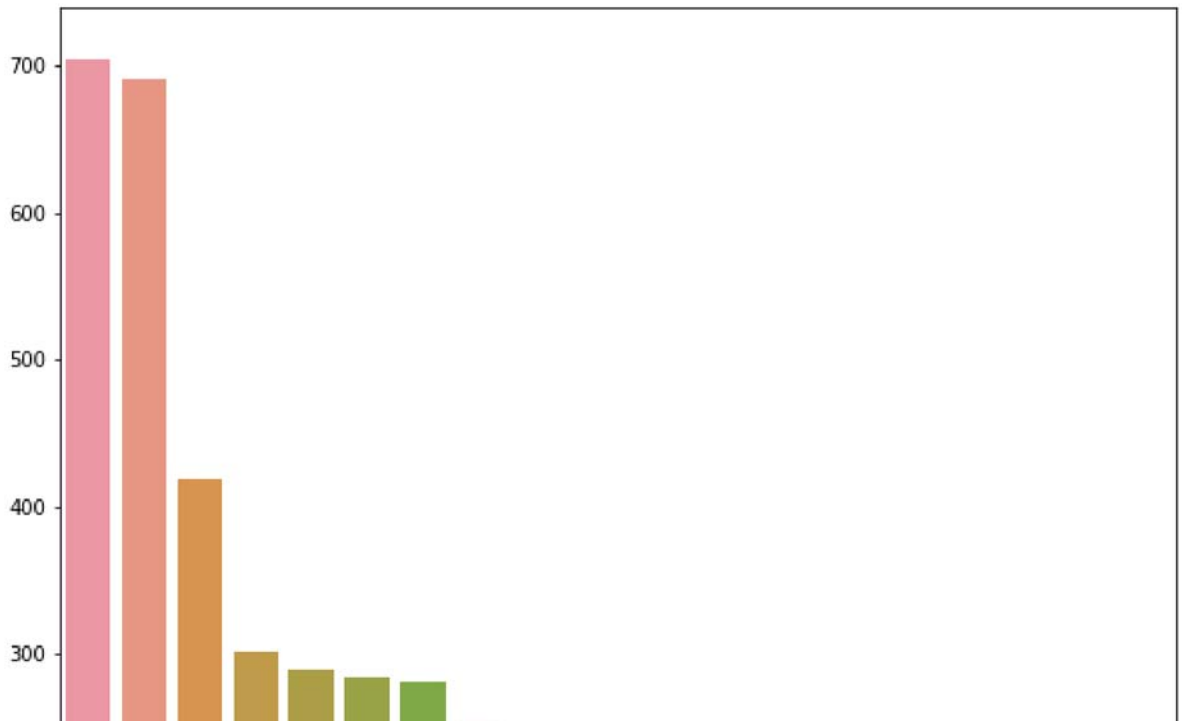
посмотрим наиболее часто встречающиеся слова, возможно, отсюда уже можно будет ориентироваться на категории новостей

```
## Creating FreqDist for whole BoW, keeping the 20 most common tokens
all_fdist = freq.most_common(20)

## Conversion to Pandas series via Python Dictionary for easier plotting
all_fdist = pd.Series(dict(all_fdist))

## Setting figure, ax into variables
fig, ax = plt.subplots(figsize=(10,10))

## Seaborn plotting using Pandas attributes + xtick rotation for ease of viewing
all_plot = sns.barplot(x=all_fdist.index, y=all_fdist.values, ax=ax)
plt.xticks(rotation=30);
```



нет, это все общеупотребительные слова

```
text
```

```
0       Дизайнеры агентства Fabula Branding (Минск) пр...
1       В течение 30 дней, с даты публикации сообщения...
2       Глава государства назвал совершенно не имеющей...
3       Интерфакс-Россия, Новость:\nАвиакомпания "Авро...
4       Отечественный концерн по производству автомоби...
                              ...
2145    Шведская IKEA не будет строить торговый центр ...
2146    Шведская IKEA планирует выставить права на дол...
2147    Шведская компания IKEA намерена трудоустроить ...
2148    Шведский ритейлер IKEA направил обращение упол...
2149         Каждый пятый товар сети подешевеет на 15–20%
Name: description, Length: 2150, dtype: object
```

```
def main_words1(row):
      return nltk.word_tokenize(row['description'])# токинезация текста i-го документа


words1=df_train.apply(main_words1,axis=1)


words1
```

```
0       [Дизайнеры, агентства, Fabula, Branding, (, Ми...
1       [В, течение, 30, дней, ,, с, даты, публикации,...
2       [Глава, государства, назвал, совершенно, не, и...
3       [Интерфакс-Россия, ,, Новость, :, Авиакомпания...
4       [Отечественный, концерн, по, производству, авт...
                              ...
2145    [Шведская, IKEA, не, будет, строить, торговый,...
2146    [Шведская, IKEA, планирует, выставить, права, ...
2147    [Шведская, компания, IKEA, намерена, трудоустр...
2148    [Шведский, ритейлер, IKEA, направил, обращение...
```

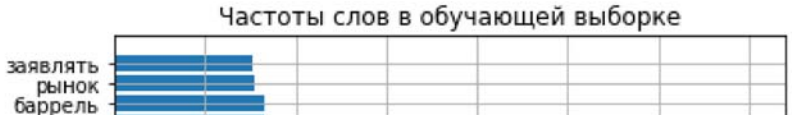
```
2149    [Каждый, пятый, товар, сети, подешевеет, на, 1...
Length: 2150, dtype: object
```

```
count=0
ww=[]
for i in range(len(words1)):
  if type(words1[i])==list:
     ww=ww+words1[i]
     count+=1
word = list(filter(lambda x: x  != 'quot', ww))
word_ws=[w.lower()  for w in   word if w.isalpha() ]#исключение слов и символов
word_w=[w for w in word_ws if w not in russian_stopwords ]#нижний регистр
lem = mystem.lemmatize ((" ").join(word_w))# лемматизация i -го документа
lem=[w for w in lem if w.isalpha() and len(w)>1]
freq=nltk.FreqDist(lem)# распределение слов в i -м документе по частоте
z=[]# обновление списка для нового документа
z=[(key+":"+str(val)) for key,val in freq.items() if val>1] # частота упоминания через : о

    #text=text+"|text" +" "+str((" ").join(z))+'\n'# запись в мешок слов с меткой |text
    #text=text+"|text" +" "+str((" ").join(z).encode('utf-8'))+'\n'# запись в мешок слов с
c=[];d=[]
for key,val in freq.items():#подготовка к сортировке слов по убыванию частоты в i -м докум
        if val>1:
            c.append(val); d.append(key)
a=[];b=[]
for k in np.arange(0,len(c),1):#сортировка слов по убыванию частоты в i -м документе
                ind=c.index(max(c));  a.append(c[ind])
                b.append(d[ind]); del c[ind]; del d[ind]

a=a[0:20];b=b[0:20]# TOP-10 для частот a  и слов b в i -м документе
y_pos = np.arange(1,len(a)+1,1)#построение TOP-10 диаграмм
performance =a
plt.barh(y_pos, a)
plt.yticks(y_pos, b)
plt.xlabel(u'Количество слов')
plt.title(u'Частоты слов в обучающей выборке', size=12)
plt.grid(True)
plt.show()
```

```
Exception ignored in: <function BatchVectorizer.__del__ at 0x7fc435d63170>
Traceback (most recent call last):
  File "/usr/local/lib/python3.7/dist-packages/artm/batches_utils.py", line 137, in _
    self.__dispose()
  File "/usr/local/lib/python3.7/dist-packages/artm/batches_utils.py", line 130, in _
    shutil.rmtree(self._target_folder)
  File "/usr/lib/python3.7/shutil.py", line 485, in rmtree
    onerror(os.lstat, path, sys.exc_info())
  File "/usr/lib/python3.7/shutil.py", line 483, in rmtree
    orig_st = os.lstat(path)
FileNotFoundError: [Errno 2] No such file or directory: 'urnuuidaf904b00-7ceb-11ec-86
```




```
sw_texts_list = []
for text in tqdm(df_train['text_prep']):
    tokens = word_tokenize(text)
    tokens = [token for token in tokens if token not in russian_stopwords and token != ' '
    text = " ".join(tokens)
    sw_texts_list.append(text)

df_train['text_sw'] = sw_texts_list
```

```
  0%|          | 0/2069 [00:00<?, ?it/s]
```

```
count
```

```
2150
```

```
#здесь теперь word содержит все слова с обучающей выборки
#далее это пригодится для bigARTM
len(word)
```

```
82472
```

```
!pip install bigartm
```

```
Collecting bigartm
  Downloading bigartm-0.9.2-cp37-cp37m-manylinux1_x86_64.whl (1.9 MB)
     |████████████████████████████████| 1.9 MB 14.5 MB/s
Requirement already satisfied: numpy in /usr/local/lib/python3.7/dist-packages (from
Requirement already satisfied: pandas in /usr/local/lib/python3.7/dist-packages (fro
Requirement already satisfied: protobuf>=3.0 in /usr/local/lib/python3.7/dist-package
Requirement already satisfied: tqdm in /usr/local/lib/python3.7/dist-packages (from t
Requirement already satisfied: six>=1.9 in /usr/local/lib/python3.7/dist-packages (fr
Requirement already satisfied: pytz>=2017.2 in /usr/local/lib/python3.7/dist-packages
Requirement already satisfied: python-dateutil>=2.7.3 in /usr/local/lib/python3.7/dis
Installing collected packages: bigartm
Successfully installed bigartm-0.9.2
```

```
import artm
```

```
from sklearn.feature_extraction.text import CountVectorizer

from numpy import array

cv = CountVectorizer(max_features=1000, stop_words='english')
n_wd = array(cv.fit_transform(word).todense()).T
vocabulary = cv.get_feature_names_out()

bv = artm.BatchVectorizer(data_format='bow_n_wd',
                          n_wd=n_wd,
                          vocabulary=vocabulary)



lda = artm.LDA(num_topics=5, alpha=0.01, beta=0.001,
               num_document_passes=5, dictionary=bv.dictionary,
               cache_theta=True)


lda.fit_offline(batch_vectorizer=bv, num_collection_passes=10)


top_tokens = lda.get_top_tokens(num_tokens=10)
for i, token_list in enumerate(top_tokens):
    print ('Topic #{0}: {1}'.format(i, token_list))
```

```
Topic #0: ['на', 'по', 'года', 'млрд', 'ссылкой', 'будет', 'заявил', 'передает', 'мож
Topic #1: ['компания', 'млн', '2017', 'как', 'баррель', 'газа', 'год', 'при', 'уровне
Topic #2: ['что', 'до', 'для', 'россии', '2016', 'роснефть', 'пишет', 'после', 'руб',
Topic #3: ['за', 'об', 'не', 'сообщает', 'со', 'сша', 'от', 'рублей', 'говорится', 'а
Topic #4: ['этом', 'компании', 'из', 'году', 'газпром', 'нефти', 'января', '10', 'дол
```

что мы видим? что по мнению BigARTM следует сгруппировать новости так:

0 - новости касательно новостей заявительного характера, то есть по сути совпадает с рекламным характером

1 - новости касательно биржевой оптовой торговли нефтью и газом, возможно 2017 года

2 - новости относительно 2016 года касательно россии и роснефти

3 - новости относительно США и акций

4 - новости относительно газпрома, нефти, и очевидно долларов

на мой взгляд, это неудовлетворительное разделение на категории, будем работать с моими предложенными категориями.

обучение через FastText

```
!pip3 install fasttext
```

```
Collecting fasttext
  Downloading fasttext-0.9.2.tar.gz (68 kB)
     |████████████████████████████████| 68 kB 4.2 MB/s
Collecting pybind11>=2.2
  Using cached pybind11-2.9.0-py2.py3-none-any.whl (210 kB)
Requirement already satisfied: setuptools>=0.7.0 in /usr/local/lib/python3.7/dist-pac
Requirement already satisfied: numpy in /usr/local/lib/python3.7/dist-packages (from
Building wheels for collected packages: fasttext
  Building wheel for fasttext (setup.py) ... done
  Created wheel for fasttext: filename=fasttext-0.9.2-cp37-cp37m-linux_x86_64.whl siz
  Stored in directory: /root/.cache/pip/wheels/4e/ca/bf/b020d2be95f7641801a6597a29c81
Successfully built fasttext
Installing collected packages: pybind11, fasttext
Successfully installed fasttext-0.9.2 pybind11-2.9.0
```

```
import pandas as pd
import fasttext


# А теперь одно из разачарований имплементации именно этой библиотеки:
# Для обучения придется сделать файл, где целевой класс должен начинаться с __label__

df_train['target'] = df_train['label'].apply(lambda x: '__label__' + str(x))
df_train[['target', 'description']].to_csv('train_data.txt', header=False, index=False, se

# обучаем на 20 эпохах, полный набор гиперпараметров можно взглянуть на официальном сайте
model = fasttext.train_supervised(input='train_data.txt', epoch=20)


p = model.predict('Дизайнеры агентства Fabula Branding (Минск) провели комплексную разрабо


p
```

```
(('__label__1', '__label__3', '__label__2', '__label__4', '__label__5'),
 array([0.48090658, 0.3290841 , 0.08553353, 0.05949963, 0.04502626]))
```

Чтобы изменить содержимое ячейки, дважды нажмите на нее (или выберите "Ввод")

```
model_fasttext = fasttext.train_supervised(input="train_data.txt", lr=0.5, epoch=25, wordN


p = model_fasttext.predict('Дизайнеры агентства Fabula Branding (Минск) провели комплексну
p
```

```
(('__label__1', '__label__4', '__label__3', '__label__2', '__label__5'),
 array([0.98795623, 0.04604391, 0.01641303, 0.00942259, 0.00171072]))
```

с помощью библиотеки fasttext успешно произвели обучение с высокой точностью, будем с ее помощью оценивать успешность работы других моделей

```
model_fasttext.test("train_data.txt", k=5)
```

```
(2069, 0.2, 1.0)
```

```
p = model_fasttext.predict('Краснинский суд Смоленской области арестовал счет с 9,3 млрд р
p
```

```
(('__label__2', '__label__1', '__label__5', '__label__3', '__label__4'),
 array([9.96416390e-01, 8.85735452e-03, 2.89958040e-03, 8.65900831e-04,
        5.62778732e-04]))
```

здесь увидели что два примера распознаны были верно

Многослойный перцептрон

```
from sklearn.neural_network import MLPClassifier

MP = Pipeline([('vect', CountVectorizer()),
               ('tfidf', TfidfTransformer()),
               ('mp', MLPClassifier(hidden_layer_sizes=(8,8,8), activation='relu', solver
              ])
```

Чтобы изменить содержимое ячейки, дважды нажмите на нее (или выберите "Ввод")

```
MP.fit(X_train, y_train)
```

```
Pipeline(steps=[('vect', CountVectorizer()), ('tfidf', TfidfTransformer()),
                ('mp',
                 MLPClassifier(hidden_layer_sizes=(8, 8, 8), max_iter=500))])
```

```
%%time
y_pred = sgd.predict(X_test)
```

```
CPU times: user 25.7 ms, sys: 711 µs, total: 26.4 ms
Wall time: 31 ms
```

```
print('accuracy %s' % accuracy_score(y_pred, y_test))
print(classification_report(y_test, y_pred,target_names=my_tags))
```

```
accuracy 0.6280193236714976
              precision    recall  f1-score   support

           1       0.59      0.22      0.32       143
           2       0.84      0.38      0.53        68
           3       0.63      0.96      0.76       335
           4       0.38      0.24      0.29        38
           5       0.60      0.08      0.14        37
```

```
          accuracy                           0.63       621
         macro avg       0.61      0.38      0.41       621
      weighted avg       0.63      0.63      0.57       621
```

RNN (LTSM)

```
from keras.preprocessing.text import Tokenizer
from keras.preprocessing.sequence import pad_sequences
from keras.models import Sequential
from keras.layers import Dense, Embedding, LSTM, SpatialDropout1D
from sklearn.model_selection import train_test_split
from keras.utils.np_utils import to_categorical
from keras.callbacks import EarlyStopping
from keras.layers import Dropout


# The maximum number of words to be used. (most frequent)
MAX_NB_WORDS = 50000
# Max number of words in each complaint.
MAX_SEQUENCE_LENGTH = 450
# This is fixed.
EMBEDDING_DIM = 100

tokenizer = Tokenizer(num_words=MAX_NB_WORDS, filters='!"#$%&()*+,-./:;<=>?@[\]^_`{|}~', l
tokenizer.fit_on_texts(df_train['text_sw'].values)
word_index = tokenizer.word_index
print('Found %s unique tokens.' % len(word_index))
```

```
     Found 14253 unique tokens.
```

```
X = tokenizer.texts_to_sequences(df_train['text_sw'].values)
X = pad_sequences(X, maxlen=MAX_SEQUENCE_LENGTH)
print('Shape of data tensor:', X.shape)
```

```
     Shape of data tensor: (2069, 450)
```

```
Y = pd.get_dummies(df_train['label']).values
print('Shape of label tensor:', Y.shape)
```

```
     Shape of label tensor: (2069, 5)
```

```
X_train, X_test, Y_train, Y_test = train_test_split(X,Y, test_size = 0.10, random_state =
print(X_train.shape,Y_train.shape)
print(X_test.shape,Y_test.shape)
```

```
     (1862, 450) (1862, 5)
     (207, 450) (207, 5)
```

```
model = Sequential()
model.add(Embedding(MAX_NB_WORDS, EMBEDDING_DIM, input_length=X.shape[1]))
model.add(SpatialDropout1D(0.2))
```

```
model.add(LSTM(100, dropout=0.2, recurrent_dropout=0.2))
model.add(Dense(5, activation='softmax'))
model.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['accuracy'])
print(model.summary())
```

```
Model: "sequential_2"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 embedding_2 (Embedding)     (None, 450, 100)          5000000

 spatial_dropout1d_2 (Spatia  (None, 450, 100)         0
 lDropout1D)

 lstm_2 (LSTM)               (None, 100)               80400

 dense_2 (Dense)             (None, 5)                 505

=================================================================
Total params: 5,080,905
Trainable params: 5,080,905
Non-trainable params: 0
_________________________________________________________________
None
```

```
epochs = 5
batch_size = 64

history = model.fit(X_train, Y_train, epochs=epochs, batch_size=batch_size,validation_spli
```

```
Epoch 1/5
27/27 [==============================] - 36s 1s/step - loss: 1.3863 - accuracy: 0.526
Epoch 2/5
27/27 [==============================] - 33s 1s/step - loss: 1.2363 - accuracy: 0.545
Epoch 3/5
27/27 [==============================] - 33s 1s/step - loss: 1.0624 - accuracy: 0.574
Epoch 4/5
27/27 [==============================] - 33s 1s/step - loss: 0.6938 - accuracy: 0.753
Epoch 5/5
27/27 [==============================] - 33s 1s/step - loss: 0.4359 - accuracy: 0.828
```

```
accr = model.evaluate(X_test,Y_test)
print('Test set\n  Loss: {:0.3f}\n  Accuracy: {:0.3f}'.format(accr[0],accr[1]))
```

```
7/7 [==============================] - 1s 106ms/step - loss: 1.1139 - accuracy: 0.628
Test set
  Loss: 1.114
  Accuracy: 0.628
```

```
texts=df['description'][33]
X = tokenizer.texts_to_sequences(texts)
X = pad_sequences(X, maxlen=MAX_SEQUENCE_LENGTH)
```

```
texts
```

```
'"Аэрофлот" отменил ряд рейсов на внутрироссийских направлениях из-за приостановки э
ксплуатации шести'
```

```
model.predict(np.array(X))[0]
```

```
array([0.24125358, 0.17394969, 0.3706453 , 0.11501416, 0.09913718],
      dtype=float32)
```

Частичное обучение с учителем

```
#когда загружаем работу заново, то сразу грузим датасет с метками и лемматизацией и прочее
df_train=pd.read_csv('/content/drive/MyDrive/Colab Notebooks/df_train_prep.csv')
```

```
from numpy import concatenate
import sklearn
from sklearn.datasets import make_classification
from sklearn.semi_supervised import LabelPropagation
```

```
#пробуем fasttext на неразмеченных примерах
p = model_fasttext.predict(df['description'][3133], k=5)
p[0][0][-1]
```

```
'3'
```

```
logreg.predict([df['description'][3133]])
```

```
array([1])
```

новость нейтрального характера, пример распознан успешно

```
def insert(df, row):
    insert_loc = df.index.max()

    if pd.isna(insert_loc):
        df.loc[0] = row
    else:
        df.loc[insert_loc + 1] = row
```

будем дообучать logreg

```
prep_text = [remove_multiple_spaces(remove_numbers(remove_punctuation(text.lower()))) for
```

```
100%                                                  37243/37243 [00:04<00:00, 10137.78it/s]
```

```
df['text_prep'] = prep_text
```

```
df['text_prep'] = prep_text
sw_texts_list = []
for text in tqdm(df['text_prep']):
    tokens = word_tokenize(text)
    tokens = [token for token in tokens if token not in russian_stopwords and token != ' '
    text = " ".join(tokens)
    sw_texts_list.append(text)

df['text_sw'] = sw_texts_list
```

```
100%                                                  37243/37243 [00:13<00:00, 1514.16it/s]
```

```
df['labeled']=False
```

```
df.head()
```

	provider	date_time	title	description	
	miru_business	2017-02-02 06:00:33	"100 друзей" Гродненского мясокомбината	Дизайнеры агентства Fabula Branding (Минск) пр...	http://www.vsesmi.ru/bu
	miru_business	2017-02-24 09:00:34	"ArcelorMittal Кривой Рог" инициировал антидем...	В течение 30 дней, с даты публикации сообщения...	http://www.vsesmi.ru/bu
	ıcom_common	2016-12-05 20:30:06	"Абсолютно никчемный аргумент": Путин прокомме...	Глава государства назвал совершенно не имеющей...	http://www.newsru.com/
	miru_business	2017-01-11 07:45:35	"Аврора" со следующей недели открывает новый р...	Интерфакс-Россия, Новость:\nАвиакомпания "Авро...	http://www.vsesmi.ru/bu
	ance_common	2017-02-07 11:15:54	"АвтоВАЗ" наращивает продажи. Главное	Отечественный концерн по производству автомоби...	http://www.\

```
data_add=[]
for i in range(2151,len(df)):
  if not df['labeled'][i]:
    x=np.array_str ( logreg.predict( [df['text_sw'][i]]  ))[1]
    y=model_fasttext.predict(df['text_sw'][i], k=5)[0][0][-1]
    if x==y:
      data_add.append([df['description'][i] ,df['text_sw'][i],x ])
```

```
len(data_add)
```

```
29098
```

это успех! было на второй итерации успешно распознано более 29000 примеров! анализ содержимого показал, что это соответствует истине

```
data_add[:-30]
```

```
df_train2=pd.DataFrame(data=data_add,columns=['description','text_sw','label'])
```

```
df_train2.head()
```

1 to 5 of 5 entries Filter

index	description	text_sw	label
0	Гендиректора сети IKEA в России Вальтер Каднар сообщил, что ритейлер планирует снизить цены на 15-20% на 1,8 тысячи видов продаваемых товаров, отметив, что на некоторые товары цена будет снижена на 40%, передает РИА "Новости" со ссылкой на печатные СМИ.	гендиректора сети ikea россии вальтер каднар сообщил ритейлер планирует снизить цены тысячи видов продаваемых товаров отметив некоторые товары цена снижена передает риа amp quot новости amp quot ссылкой печатные сми	3
1	IKEA не хочет платить российскому бизнесмену Константину Пономарёву 507 млн рублей, которые ему присудил Краснинский суд, и	ikea хочет платить российскому бизнесмену константину пономарёву млн рублей которые присудил краснинский суд	2
2	Производитель сигарет Imperial Tobacco решил закрыть одну из двух своих российских фабрик из-за падения табачного рынка и непростой	производитель сигарет imperial tobacco решил закрыть одну двух своих российских фабрик падения табачного рынка непростой ситуации	3

```
X = df_train2['text_sw']
y = df_train2['label']
```

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state = 42
```

```
len(X_train)
```

```
23278
```

```
logreg.fit(X_train, y_train)
```

```
/usr/local/lib/python3.7/dist-packages/sklearn/linear_model/_logistic.py:818: Converg
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression
  extra_warning_msg=_LOGISTIC_SOLVER_CONVERGENCE_MSG,
Pipeline(steps=[('vect', CountVectorizer()), ('tfidf', TfidfTransformer()),
                ('logreg', LogisticRegression(C=100000.0, n_jobs=1))])
```

```
y_pred = logreg.predict(X_test)
```

```
print('accuracy %s' % accuracy_score(y_pred, y_test))
print(classification_report(y_test, y_pred,target_names=my_tags))
```

```
accuracy 0.9750859106529209
              precision    recall  f1-score   support

           1       0.91      0.75      0.82       290
           2       0.92      0.90      0.91       136
           3       0.98      0.99      0.99      5324
           4       0.86      0.71      0.78        59
           5       0.75      0.27      0.40        11

    accuracy                           0.98      5820
   macro avg       0.88      0.72      0.78      5820
weighted avg       0.97      0.98      0.97      5820
```

точность составила порядка 98 %

загрузка оставшихся датасетов, еще одна итерация

```
df_all=pd.read_csv('/content/drive/MyDrive/Colab Notebooks/20170601.csv')
```

```
df_all.append(pd.read_csv('/content/drive/MyDrive/Colab Notebooks/20170901.csv'),sort=Fals
```

	u_id	provider	date_time	title	desc
0	407986	lifenewsru_common	2017-03-05 15:00:50	!!! АВИТКЕПСРЕП	
1	836706	mailru_common	2017-05-11 14:00:28	""Вашингтону" нужно расставаться с Овечкиным"....	<p>Североамери пресса пс неу
2	819187	mailru_common	2017-05-08 15:45:20	""Краснодар" рискует потерять лицо". Ловчев - ...	<p>Обозр «Советского Евге
3	864467	mailru_common	2017-05-16 11:00:44	"11 Аспиликуэт выиграют Лигу чемпионов". Самый...	<p>И незамен в нынешнем
4	724875	lifenewsru_common	2017-04-22 11:30:51	"13 причин почему". Американская история про "...	Ученица старшей городка Либ
...	...	...	...	...	
617985	1491950	infoxru_business	2017-08-15	... Плюс монополизация	Почему правитель

```
df_all.append(pd.read_csv('/content/drive/MyDrive/Colab Notebooks/20171201.csv'),sort=Fals
```

	u_id	provider	date_time	title	de
0	407986	lifenewsru_common	2017-03-05 15:00:50	!!! АВИТКЕПСРЕП	
1	836706	mailru_common	2017-05-11 14:00:28	""Вашингтону" нужно расставаться с Овечкиным"....	<p>Североаме пресса н

```
df_all.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 584369 entries, 0 to 584368
Data columns (total 8 columns):
 #   Column       Non-Null Count   Dtype 
---  ------       --------------   ----- 
 0   u_id         584369 non-null  int64 
 1   provider     584369 non-null  object
 2   date_time    584369 non-null  object
 3   title        584369 non-null  object
 4   description  551672 non-null  object
 5   link         584301 non-null  object
 6   pubdate      584369 non-null  object
 7   numfield     584369 non-null  int64 
dtypes: int64(2), object(6)
memory usage: 35.7+ MB
```

```
df_all['description']=df_all['description'].fillna(df_all['title'])
```

```
df_all=df_all[df_all['link'].str.contains("business") | df_all['link'].str.contains("finan
```

```
df_all['Languagereveiw'] = df_all.apply(detect_ru, axis=1)
```

```
df_all=df_all[df_all['Languagereveiw']==True ].reset_index(drop=True)
df_all.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 41555 entries, 0 to 41554
Data columns (total 9 columns):
 #   Column          Non-Null Count  Dtype 
---  ------          --------------  ----- 
 0   u_id            41555 non-null  int64 
 1   provider        41555 non-null  object
 2   date_time       41555 non-null  object
 3   title           41555 non-null  object
 4   description     41555 non-null  object
 5   link            41555 non-null  object
 6   pubdate         41555 non-null  object
 7   numfield        41555 non-null  int64 
 8   Languagereveiw  41555 non-null  bool  
dtypes: bool(1), int64(2), object(6)
memory usage: 2.6+ MB
```

```
prep_text = [remove_multiple_spaces(remove_numbers(remove_punctuation(text.lower()))) for
df_all['text_prep'] = prep_text
sw_texts_list = []
for text in tqdm(df_all['text_prep']):
    tokens = word_tokenize(text)
    tokens = [token for token in tokens if token not in russian_stopwords and token != ' '
    text = " ".join(tokens)
    sw_texts_list.append(text)

df_all['text_sw'] = sw_texts_list
```

```
100%                                                  41555/41555 [00:14<00:00, 2542.53it/s]
100%                                                  41555/41555 [00:20<00:00, 1095.70it/s]
```

```
df_all['labeled']=False
```

```
df_all.head()
```

provider	date_time	title	description	
_common	2017-03-29 19:00:47	"I-ll be back": история искусственного интеллекта	Об искусственном разуме человечество задумалос...	http://www.vestifinance
_business	2017-05-10 04:15:30	"А почему бы и нет?" Как парень из алтайской д...	Предприятие Анатолия Вытоптова одним из первых...	http://www.vsesmi.ru/business/201
u_finance	2017-04-25 02:45:39	"Авишка" возвращается в новом образе	Агропромышленный холдинг «Авида» является прои...	http://www.vsesmi.ru/economy/201
_common	2017-03-21 18:45:37	"АвтоВАЗ" отзывает 106 тыс. автомобилей "Лада"	Компания "АвтоВАЗ" объявила об отзыве 106,7 ты...	http://www.vestifinance
_common	2017-03-21 22:00:44	"АвтоВАЗ" отзывает 106 тысяч автомобилей "Лада"	Компания "АвтоВАЗ" объявила об отзыве 106,7 ты...	http://www.vestifinance

```
df_all['u_id'].value_counts
```

```
<bound method IndexOpsMixin.value_counts of 0        567759
1        825987
2        738680
3        512853
4        513963
          ...
41550    393643
41551    552159
41552    645421
41553    575493
41554    488716
Name: u_id, Length: 41555, dtype: int64>
```

```
df_all.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 41555 entries, 0 to 41554
Data columns (total 12 columns):
 #   Column          Non-Null Count  Dtype 
---  ------          --------------  ----- 
 0   u_id            41555 non-null  int64 
 1   provider        41555 non-null  object
 2   date_time       41555 non-null  object
 3   title           41555 non-null  object
 4   description     41555 non-null  object
 5   link            41555 non-null  object
 6   pubdate         41555 non-null  object
 7   numfield        41555 non-null  int64 
 8   Languagereveiw  41555 non-null  bool  
 9   text_prep       41555 non-null  object
 10  text_sw         41555 non-null  object
 11  labeled         41555 non-null  bool  
dtypes: bool(2), int64(2), object(8)
memory usage: 3.2+ MB
```

```
logreg.predict( [df_all['text_sw'][555]])
```

```
array(['3'], dtype=object)
```

```
len(df_all)
```

```
41555
```

```
x=logreg.predict( [df_all['text_sw'][100]]  )[0]
x
```

```
'1'
```

```
type(x)
```

```
str
```

```
df_all['labeled']=False
```

```
data_add=[]
for i in range(len(df_all)):
  if not df_all['labeled'][i]:
    x=logreg.predict( [df_all['text_sw'][i]]  )[0]
    y=model_fasttext.predict(df_all['text_sw'][i], k=5)[0][0][-1]
    #print(df_all['text_sw'][i],x,y)
    if x==y:
      data_add.append([df_all['description'][i] ,df_all['text_sw'][i],x ])
     # df_all['labeled'][i]=True


len(data_add)
```

```
37503
```

```
df_train3=pd.DataFrame(data=data_add,columns=['description','text_sw','label'])


df_train2=df_train2.append(df_train3, sort=False)


df_train2.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 66601 entries, 0 to 37502
Data columns (total 3 columns):
 #   Column       Non-Null Count  Dtype 
---  ------       --------------  ----- 
 0   description  66601 non-null  object
 1   text_sw      66601 non-null  object
 2   label        66601 non-null  object
dtypes: object(3)
memory usage: 2.0+ MB
```

```
df_train2.reset_index(drop=True)
```

	description	text_sw	label
0	Гендиректора сети IKEA в России Вальтер Каднар...	гендиректора сети ikea россии вальтер каднар с...	3

```
X = df_train2['text_sw']
y = df_train2['label']
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state = 42
logreg.fit(X_train, y_train)
y_pred = logreg.predict(X_test)
print('accuracy %s' % accuracy_score(y_pred, y_test))
print(classification_report(y_test, y_pred,target_names=my_tags))
```

```
/usr/local/lib/python3.7/dist-packages/sklearn/linear_model/_logistic.py:818: Converg
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression
  extra_warning_msg=_LOGISTIC_SOLVER_CONVERGENCE_MSG,
accuracy 0.9864124314991367
              precision    recall  f1-score   support

           1       0.94      0.86      0.90       679
           2       0.96      0.94      0.95       323
           3       0.99      1.00      0.99     12149
           4       0.93      0.89      0.91       141
           5       1.00      0.72      0.84        29

    accuracy                           0.99     13321
   macro avg       0.96      0.88      0.92     13321
weighted avg       0.99      0.99      0.99     13321
```

точность повысилась еще на один пункт!

точность повысилась еще на один пункт!