Домашнее задание №10

Асланов А.Б., ИУ9-21М

Имеются данные для задачи анализа тональности:

- цитаты;
- обучающая коллекция с ответами;
- тестовая коллекция с ответами.
  Нужно обучиться на обучающей коллекции, затем провериться на тестовой.
  Классификация на три класса (позитивный, негативный, нейтральный)

In [1]:

```python
from bs4 import BeautifulSoup
from lxml import etree
import pandas as pd


train = '/Users/user/Desktop/ir/ir10/train/news_eval_train.xml'
test = '/Users/user/Desktop/ir/ir10/test/news_eval_test.xml'
```

Предобработка

1. Парсинг XML
2. Замена +-/-/+ на числовые метки
3. Получение дополнительных признаков
4. Лемматизация цитат
5. Сохранение итогового датафрейма

1. Парсинг XML

Парсим xml: получаем на выходе массивы с цитатами из твиттера, их разметкой (анализом тональностей) и ссылкой на источник.

In [2]:

```python
def get_data(set_type, metric):
    """
    set_type: 'train' or 'test' data;
    metric: evaluation, quote, url
    """

    page = open(set_type, encoding="cp1251")
    soup = BeautifulSoup(page.read())
    metrics = soup.findAll(metric)
    if metric == 'speech':
        quotes = [quote.string.strip() for quote in soup.findAll('speech')]
        return quotes
    else:
        metrics_list = []
        for metric in metrics:
            metrics_list.append(metric.text.split())
        metrics = []
        for metric in metrics_list:
            for m in metric:
                metrics.append(m)
        return metrics



train_quotes = get_data(train, 'speech')
train_evals = get_data(train, 'evaluation')
train_urls = get_data(train, 'url')


test_quotes = get_data(test, 'speech')
test_evals = get_data(test, 'evaluation')
test_urls = get_data(test, 'url')

# В тестовой выборке почему-то после парсинга взяли знаки переноса строки. Удаляем их.
test_quotes = [quote.replace("\n", " ") for quote in test_quotes];
```

Предобработка 2: заменяем оценочные символы числовыми оценками

Заменяем позитивный, негативный и нейтральный отзывы на оценки:
1) 0 - непонятно;
2) 1 - "-";
3) 2 - "+-";
4) 3 - "+".

In [3]:

```python
import pandas as pd

# We code evaluations as numbers
train_marks = []
for mark in train_evals:
    if mark == '+':
        train_marks.append(3)
    elif mark == '+-':
        train_marks.append(2)
    elif mark == '-':
        train_marks.append(1)
    else:
        train_marks.append(0)

test_marks = []
for mark in test_evals:
    if mark == '+':
        test_marks.append(3)
    elif mark == '+-':
        test_marks.append(2)
    elif mark == '-':
        test_marks.append(1)
    else:
        test_marks.append(0)
```

Предобработка 3: получаем дополнительные признаки

Дополнительными признаками могут являться: домен, количество обратных slash'ей (это будет показывать, насколько "далеко" сайт расположен относительно главной страницы), само название сайта (его главная страница).

In [4]:

```python
def get_slash_nums(urls):
    """Дополнительный признак: количество слешей"""

    slash_nums = []
    for url in urls:
        slash_nums.append(len(url.split('/')))
    return slash_nums


def get_domens(urls):
    """Дополнительный признак: домен сайта"""

    domens = []
    for url in urls:
        domens.append(url.split('/')[2].split('.')[-1])
    return domens


def get_base_site_name(urls):
    """Дополнительный признак: название главного сайта"""

    base_site_names = []
    for url in urls:
        base_site_names.append(url.split('/')[2].split('.')[-2])
    return base_site_names


train_slash_nums = get_slash_nums(train_urls);
test_slash_nums = get_slash_nums(test_urls);
train_domens = get_domens(train_urls);
test_domens = get_domens(test_urls);
train_base_site_names = get_base_site_name(train_urls);
test_base_site_names = get_base_site_name(test_urls);


# Датасет с обучающей выборкой
df_train = pd.DataFrame({'quotes':train_quotes,
                         'evals':train_marks,
                         'urls':train_urls,
                         'base_site_names':train_base_site_names,
                         'slash_nums':train_slash_nums,
                         'domens':train_domens})

# Датасет с тестовой выборкой
df_test = pd.DataFrame({'quotes':test_quotes,
                        'evals':test_marks,
                        'urls':test_urls,
                        'base_site_names':test_base_site_names,
                        'slash_nums':test_slash_nums,
                        'domens':test_domens})
```

In [5]:

```python
# Объединяем всё в один датасет
df = pd.concat([df_train, df_test], axis=0, ignore_index=True)

# Удаляем столбец с url, так как он теперь не нужен
df = df.drop(['urls'], axis=1)
```

Предобработка 4: в датафрейме кодируем полученные дополнительные признаки

In [6]:

```python
from sklearn import preprocessing
import numpy as np
le = preprocessing.LabelEncoder()


def make_label_encoding(column_name):
    base_site_names =  df[column_name].values
    le.fit(base_site_names)
    df['{}_le'.format(column_name)] = le.transform(base_site_names)


make_label_encoding('base_site_names')
make_label_encoding('domens')
df = df[['quotes', 'evals', 'slash_nums', 'base_site_names_le', 'domens_le']]
df[:20] # Показать первые 20 цитат
```

Out[6]:

	quotes	evals	slash_nums	base_site_names_le	domens_le
0	Далее в своей проповеди он напомнил, что по би...	0	4	72	19
1	"Меня отпустили. У Коли @nlyaskin забирают вещ...	1	6	42	19
2	"Мои игроки не разочаровали меня, даже слегка ...	2	5	396	19
3	В интервью РИА "Новости" уполномоченный по пра...	0	8	361	19
4	"Психологи начнут работать с двумя девушками, ...	2	8	361	19
5	"Когда мы общались с ними, то видели, что деву...	2	8	361	19
6	"На комиссии мы должны будем оценить степень и...	2	8	361	19
7	Бывший главный тренер сборной Англии Грэм Тэйл...	0	5	409	19
8	"На телах жертв были обнаружены многочисленные...	1	5	243	19
9	"По результатам доследственной проверки следст...	1	5	243	19
10	"Не понимают и не принимают горожане и позицию...	1	5	243	19
11	"Каждый гражданин Южной Осетии имел возможност...	3	6	357	19
12	Лучшая мужская роль второго плана была, по мне...	3	6	404	3
13	Помнится, один мой товарищ прибежал как-то с у...	1	5	87	22
14	Председатель совета директоров московского "Сп...	0	5	79	3
15	Разработчик препарата - фармацевтическая компа...	3	9	61	19
16	Известный российский тренер Михаил Гершкович з...	3	5	409	19
17	Сам игрок заявил, что пока не думает о переход...	0	5	396	19
18	"Меня отпустили. У Коли забирают вещи и сажают...	1	6	225	16
19	По мнению Каспарова, Запад допустил грубейшую ...	1	6	174	19

Лемматизация исходных цитат, удаление спец. символов, сохранение датафрейма

In [8]:

```python
from nltk.tokenize import word_tokenize
from pymystem3 import Mystem
mystem = Mystem()


def remove_stop_words(df):

    symbols = "[ \n«»;,@'```—.%()?!-:''...']"
    preproc_quotes = []
    for quote in df.quotes.values:
        new_quote = []

        word_tokens = word_tokenize(quote)
        for w in word_tokens:
            if w not in symbols:
                new_quote.append(w)

        new_quote = [mystem.lemmatize(word) for word in new_quote]
        new_quote = [word[0] for word in new_quote]
        new_quote = ' '.join(new_quote)
        preproc_quotes.append(new_quote)
    df['preproc_quotes'] = preproc_quotes
    df.to_csv('df.csv')
    return df

remove_stop_words(df)
df = remove_stop_words(df)
df[:30]
```

Out[8]:

	quotes	evals	slash_nums	base_site_names_le	domens_le	preproc_quotes
0	Далее в своей проповеди он напомнил, что по би...	0	4	72	19	далее в свой проповедь он напоминать что по би...
1	"Меня отпустили. У Коли @nlyaskin забирают вещ...	1	6	42	19	я отпускать у коли nlyaskin забирать вещь и са...
2	"Мои игроки не разочаровали меня, даже слегка ...	2	5	396	19	мой игрок не разочаровывать я даже слегка наоб...
3	В интервью РИА "Новости" уполномоченный по пра...	0	8	361	19	в интервью риа новость уполномоченный по право...
4	"Психологи начнут работать с двумя девушками, ...	2	8	361	19	психолог начинать работать с два девушка даша ...
5	"Когда мы общались с ними, то видели, что деву...	2	8	361	19	когда мы общаться с они то видеть что девушка ...
6	"На комиссии мы должны будем оценить степень и...	2	8	361	19	на комиссия мы должный быть оценивать степень ...
7	Бывший главный тренер сборной Англии Грэм Тэйл...	0	5	409	19	бывший главный тренер сборная англия грэм тэйл...
8	"На телах жертв были обнаружены многочисленные...	1	5	243	19	на тело жертва быть обнаруживать многочисленны...
9	"По результатам доследственной проверки следст...	1	5	243	19	по результат доследственная проверка следствен...
10	"Не понимают и не принимают горожане и позицию...	1	5	243	19	не понимать и не принимать горожанин и позиция...
11	"Каждый гражданин Южной Осетии имел возможност...	3	6	357	19	каждый гражданин южный осетия иметь возможност...
12	Лучшая мужская роль второго плана была, по мне...	3	6	404	3	хороший мужской роль второй план быть по мнени...
13	Помнится, один мой товарищ прибежал как-то с у...	1	5	87	22	помниться один мой товарищ прибегать как-то с ...
14	Председатель совета директоров московского "Сп...	0	5	79	3	председатель совет директор московский спартак...
15	Разработчик препарата - фармацевтическая компа...	3	9	61	19	разработчик препарат фармацевтический компания...
16	Известный российский тренер Михаил Гершкович з...	3	5	409	19	известный российский тренер михаил гершкович з...
17	Сам игрок заявил, что пока не думает о переход...	0	5	396	19	сам игрок заявлять что пока не думать о перехо...
18	"Меня отпустили. У Коли забирают вещи и сажают...	1	6	225	16	я отпускать у коли забирать вещь и сажать хрен...
19	По мнению Каспарова, Запад допустил грубейшую ...	1	6	174	19	по мнение каспаров запад допускать грубый ошиб...
20	По мнению Каспарова, также и внешнюю политику ...	1	6	174	19	по мнение каспаров также и внешний политика пу...
21	Работа не была принята: по мнению принимавшего...	1	5	281	19	работа не быть принимать по мнение принимать з...
22	С именем ребенка звездные родители пока не опр...	0	5	326	22	с имя ребенок звездный родитель пока не опреде...
23	Он признался, что не понимает, почему ему став...	1	5	81	19	он признаваться что не понимать почему он став...
24	Он заявил, что речь идет о прискорбном недораз...	1	5	81	19	он заявлять что речь идти о прискорбный недора...
25	В результате, по мнению г-на Этманова, ставшег...	3	8	351	19	в результат по мнение г-н этманова становиться...
26	"А это все надо за счет чего-то строить и соде...	0	7	346	19	а это все надо за счет что-то строить и содерж...
27	ОНЭКСИМ уже заявил, что готов выкупить 51% доп...	0	8	351	19	онэксим уже заявлять что готовый выкупать 51 д...
28	По мнению эксперта, есть внутренние резервы и ...	0	8	351	19	по мнение эксперт быть внутренний резерв и вну...
29	По мнению гендиректора Auto-Dealer.Ru Олега Да...	3	8	351	19	по мнение гендиректор Auto олег дацкива перено...
...	...	...	...	...	...	...
9730	"Поэтому Майдана у нас не будет, и я надеюсь, ...	3	6	466	3	поэтому майдан у мы не быть и я надеяться что ...
9731	"Нам было сложно против "Арсенала", против "Ре...	2	5	397	3	мы быть сложно против арсенал против реймс мы ...
9732	"Мы же видим, с какими проблемами сейчас сталк...	1	7	350	22	мы же видеть с какой проблема сейчас сталкиват...
9733	"Особо подчеркну, что эта процедура касается н...	1	7	350	22	особо подчеркивать что этот процедура касаться...
9734	"Ножевое ранение 19-летний учащийся колледжа н...	1	6	425	12	ножевой ранение 19 учащийся колледж наносить с...
9735	"Я считаю, что разделение Косово является един...	0	6	87	22	я считать что разделение косово являться единс...
9736	"Некоторые возражают, заявляя, что это вызовет...	1	6	87	22	некоторые возражать заявлять что это вызывать ...
9737	"У нас не было большого опыта в изготовлении с...	2	6	357	19	у мы не быть большой опыт в изготовление самол...
9738	"Очень переживала за Дашу Домрачеву в той борь...	3	6	347	19	очень переживать за даша домрачев в тот борьба...
9739	"Мазерати" доказала победой в прошлом году с А...	3	6	347	19	мазерать доказывать победа в прошлое год с анд...
9740	"Отношение руководителя региона к развитию физ...	3	6	347	19	отношение руководитель регион к развитие физич...
9741	"После аварии на Games в начале июля я проходи...	2	6	347	19	после авария на Games в начало июль я проходит...
9742	"Вы - граждане России". Направляю их на честны...	3	6	347	19	вы гражданин россия направлять они на честный ...
9743	"Нам надо продолжать подчищать настройки, пото...	0	6	347	19	мы надо продолжать подчищать настройка потому ...
9744	"Если меня не беспокоит травма, то я могу мног...	3	6	347	19	если я не беспокоить травма то я мочь много вы...
9745	"Участников мало, потому что руководство Между...	1	6	347	19	участник мало потому что руководство междунаро...
9746	"Мытищи" - крепкая, добротная команда, с ними ...	2	6	347	19	мытищи крепкий добротный команда с они быть оч...
9747	"Правительство и Министерство спорта Российско...	3	6	347	19	правительство и министерство спорт российский ...
9748	"Сегодня все сложилось хорошо, хотя можно было...	2	6	347	19	сегодня все складываться хорошо хотя можно быт...
9749	"Льюису очень некомфортно в нынешней ситуации,...	2	6	347	19	льюис очень некомфортно в нынешний ситуация ко...
9750	"Мы воспользовались той паузой, которая образо...	2	6	347	19	мы воспользоваться тот пауза который образовыв...
9751	"Я сейчас нахожусь в некой эйфории, потому что...	3	6	347	19	я сейчас находиться в некий эйфория потому что...
9752	"Я думаю, что "Маруся" в состоянии бороться с ...	2	6	347	19	я думать что маруся в состояние бороться с кэт...
9753	"Нас ждет тяжелая игра с "Новой Генерацией". В...	2	6	347	19	мы ждать тяжелый игра с новый генерация во-пер...
9754	"Машина практически та же, что была у нас в Мо...	2	6	347	19	машина практически тот же что быть у мы в мона...
9755	"Мое мнение по этому вопросу таково - Магдален...	2	6	347	19	мой мнение по это вопрос таков магдалена уходи...
9756	"Кроме моего напарника Льюиса Хэмилтона, Себас...	3	6	347	19	кроме мой напарник льюис хэмилтон себастьян фе...
9757	"Вы никогда не знаете, что случится. В Формуле...	0	6	347	19	вы никогда не знать что случаться в формула уж...
9758	"Рада, что выиграла медаль. Немного не получил...	2	6	347	19	рад что выигрывать медаль немного не получатьс...
9759	"Авиабилет журналисту поменяли, он может улете...	1	5	210	16	авиабилет журналист поменять он может улетать ...

9760 rows × 6 columns

Тип векторизации: частоты

Векторизация встроенным типом

CountVectorizer строит матрицу вхождений слов в документ. Если мы векторизуем цитаты без удаления стоп-слов (`stop_words=False`), на выходе получим матрицу размерностью `9760x23793` . Если векторизуем цитаты с удалением стоп-слов (`stop_words=True` , по умолчанию), на выходе размерность: `9760x23708` .

Каждый элемент этой матрицы - частотность слова в соответствующем предложении (TF).

In [2]:

```
from sklearn.feature_extraction.text import CountVectorizer


def vectorize_quotes(df, stop_words=True):
    """
    Векторизуем цитаты. На выходе получаем полную матрицу
    Если stop_words=True, то мы подаём на вход цитаты *без* стоп-слов.
    """

    vectorizer = CountVectorizer()
    if stop_words:
        vec_quotes = vectorizer.fit_transform(df.quotes_without_stopwords)
    else:
        vec_quotes = vectorizer.fit_transform(df.quotes)
    vec_quotes = vec_quotes.toarray()
    num_of_rows = vec_quotes.shape[0]
    num_of_cols = vec_quotes.shape[1]

    headers = []
    for num in range(1, num_of_cols):
        header = 'N{}'.format(num)
        headers.append(header)
    vec_quotes_ = dict(zip(headers, vec_quotes))
    df_vec = pd.DataFrame(vec_quotes)
    #print(df_vec.astype(bool).sum(axis=0))   # перепроверка, что в матрице не одни нули. upd: всё в порядке

    # Добавляем дополнительные признаки
    df_vec['slash_nums'] = df.slash_nums
    df_vec['base_site_names_le'] = df.base_site_names_le
    df_vec['domens_le'] = df.domens_le
    df_vec['evals'] = df.evals
    return df_vec


df_vec = vectorize_quotes(df)
df_vec_without_stopwords = vectorize_quotes(df, stop_words=False)
df_vec = df_vec.drop(['slash_nums', 'base_site_names_le', 'domens_le'], axis=1)   # удалить доп. признаки
```

In [3]:

```
df_vec.head()
```

Out[3]:

	0	1	2	3	4	5	6	7	8	9	...	23699	23700	23701	23702	23703	23704	23705	23706	23707	evals
0	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	1
2	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	2
3	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	0	2

5 rows × 23709 columns

Применение модели классификации

В качестве модели для классификации был выбран Random Forest (случайный лес). Случайный лес - это ансамбль решающих деревьев, выдающий высокое качество классификации (а также кластеризации и регрессии) засчет независимой друг от друга тренировки небольших по глубине деревьев. Это снижает переобучение и повышает точность в сравнении с одним деревом. В RF используется модель бэггинга - независимого обучения алгоритмов классификации, где результат определяется голосованием. Во время классификации финальным результатом будет тот класс, за который проголосовало большинство деревьев, при условии, что одно дерево обладает одним голосом.

In [4]:

```
from sklearn import model_selection, metrics
from sklearn.ensemble import RandomForestClassifier, AdaBoostClassifier
from sklearn.model_selection import KFold, cross_val_score, train_test_split, GridSearchCV
from sklearn.tree import DecisionTreeClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import accuracy_score
from sklearn.pipeline import Pipeline
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import f1_score


def RFClassifier(X, y, cv):

    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3)
    forest = RandomForestClassifier(n_estimators=10, n_jobs=-1)
    forest_params = {'max_depth': range(50, 55)}
    forest_grid = GridSearchCV(forest, forest_params, cv=5, n_jobs=-1)
    forest_grid.fit(X_train, y_train)

    f1_macro = f1_score(y_test, forest_grid.predict(X_test), average='macro')
    f1_micro = f1_score(y_test, forest_grid.predict(X_test), average='micro')
    print('f1-macro: ', f1_macro)
    print('f1-micro: ', f1_micro)
    return None


def classify_quote(df):
    """Разделяем данные на признаки и целевую переменную"""
    y = df.evals.values
    df.iloc[:, :].drop(['evals'], axis=1, inplace=True)
    X = df

    """Кросс-валидация"""
    cv = KFold(n_splits=5, shuffle=True, random_state=1)
    RFClassifier(X, y, cv)


#classify_quote(df_vec_without_stopwords);
```

Результаты

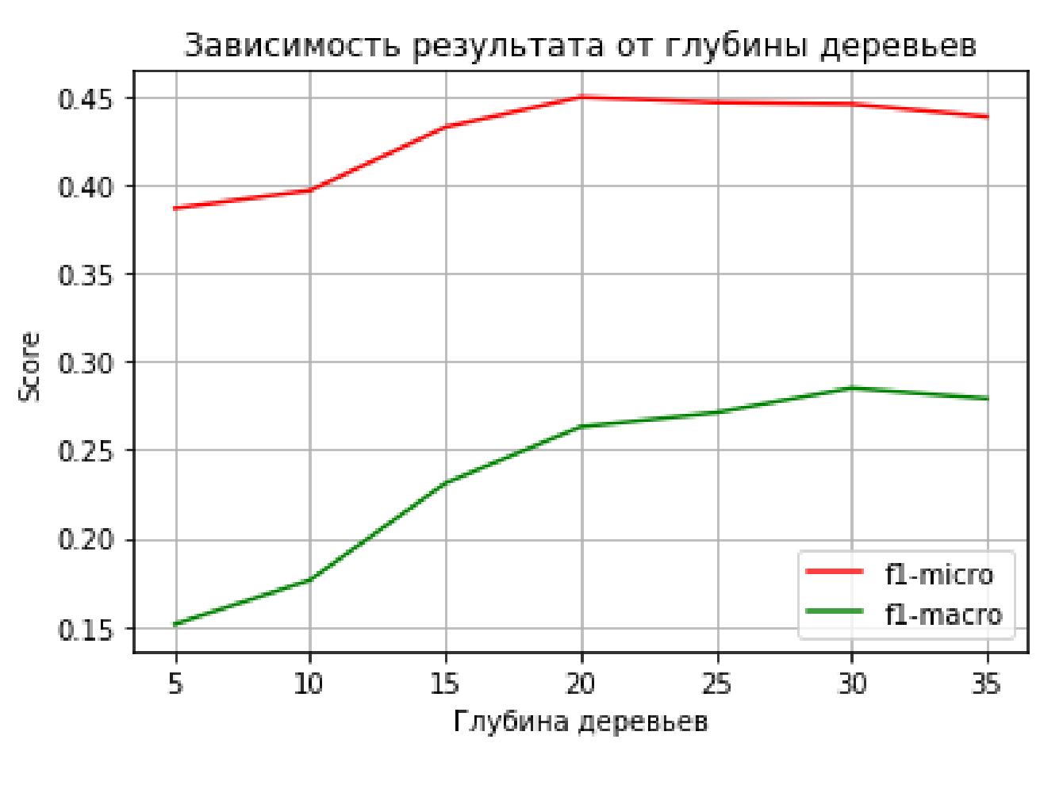
Методом поиска параметров по сетке перебираем разную глубину деревьев и смотрим, при каком значении качество классификации будет выше. Была использована постоянная кросс-валидация на 5 fold'ов.

In [24]:

```
import matplotlib.pyplot as plt

trees_depth = [5, 10, 15, 20, 25, 30, 35]
f_micros = [0.387, 0.397, 0.433, 0.450, 0.447, 0.446, 0.439]
f_macros = [0.151, 0.176, 0.231, 0.263, 0.271, 0.285, 0.279]

plt.figure()
plt.plot(trees_depth, f_micros, c='r', label='f1-micro')
plt.plot(trees_depth, f_macros, c='g', label='f1-macro')
plt.title('Зависимость результата от глубины деревьев')
plt.xlabel('Глубина деревьев')
plt.ylabel('Score')
plt.legend()
plt.grid()
plt.show()
```

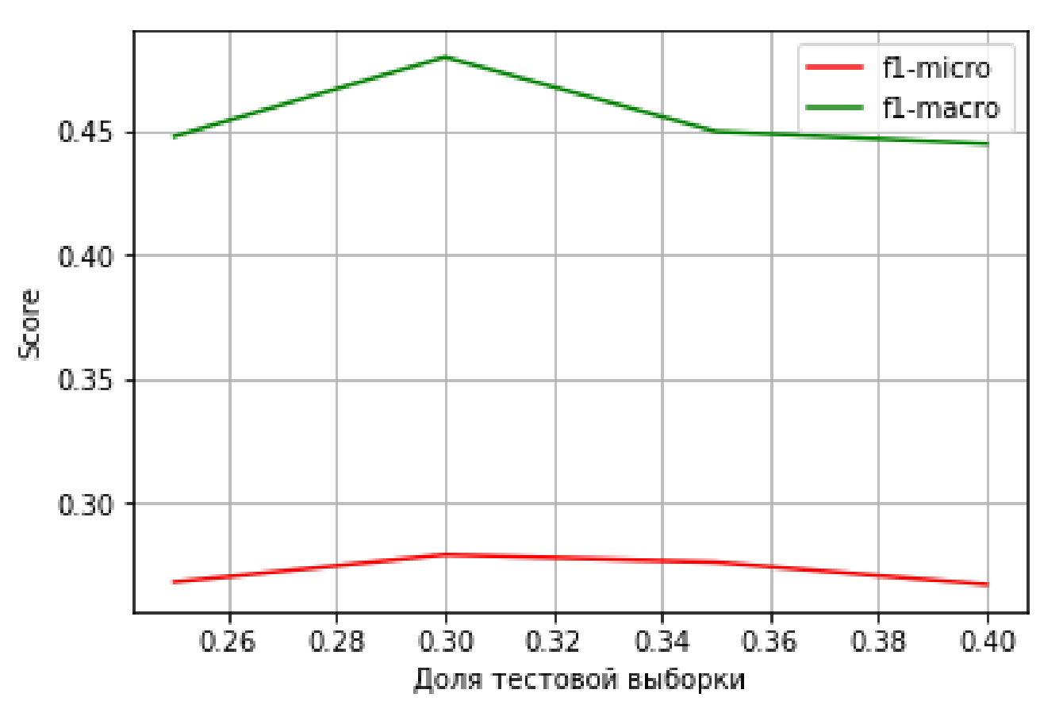


Фиксируем глубину деревьев на одном из наиболее высоких результатов (30) и пробуем изменить соотношение в разбиении выборки на обучающую и тестовую.

In [6]:

```
test_sizes = [0.25, 0.3, 0.35, 0.4]
f_micros = [0.268, 0.279, 0.276, 0.267]
f_macros = [0.448, 0.480, 0.45, 0.445]

plt.figure()
plt.plot(test_sizes, f_micros, c='r', label='f1-micro')
plt.plot(test_sizes, f_macros, c='g', label='f1-macro')
plt.xlabel('Доля тестовой выборки')
plt.ylabel('Score')
plt.legend()
plt.grid()
plt.show()
```



Наиболее хорошие результаты были показаны при доле тестовой выборки от всей равной 30%. Зафиксируем этот результат и оценим, на сколько повысится качество, если взять цитаты с удаленными стоп-словами.

Также оценим вклад дополнительных признаков в выборку (есть ли он вообще или, может, их стоит удалить из датафрейма для повышения качества). В качестве дополнительных признаков были использованы:

- Закодированные названия сайтов;
- Закодированные домены (`.ru` / `.com` / `.ch` /...);
- Количество слэшей в адресе (= удалённость от главной страницы).

	С доп. признаками		**Без доп. признаков**	
	f1-макро	f1-микро	f1-макро	f1-микро
Со стоп-словами	0.254	0.451	0.296	0.461
Без стоп-слов	0.290	0.445	***0.321***	***0.470***

Тип векторизации: TF-IDF

In [7]:

```
# Считаем df как кол-во предложений, в которых встретилось i-ое слово запроса.
# Его проще вычислить по датафрейму: df будет равен количеству ненулевых столбцов для i-го слова.

df_vec_without_stopwords['df'] = pd.Series((df_vec_without_stopwords != 0).astype(int).sum(axis=1))
df_vec_without_stopwords.head()
```

Out[7]:

	0	1	2	3	4	5	6	7	8	9	...	23788	23789	23790	23791	23792	slash_nums	base_site_names_le	domens_le	evals	df
0	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	4	72	19	0	22
1	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	6	42	19	1	42
2	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	5	396	19	2	47
3	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	8	361	19	0	26
4	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	8	361	19	2	34

5 rows × 23798 columns

In [8]:

```
import math
import numpy as np


N = df.shape[0] # количество документов == количество строк в датафрейме == количество цитат
idf = np.log10(N / df_vec_without_stopwords['df'])
df_vec_without_stopwords['idf'] = idf
df_vec_without_stopwords.head()
```

Out[8]:

	0	1	2	3	4	5	6	7	8	9	...	23789	23790	23791	23792	slash_nums	base_site_names_le	domens_le	evals	df	idf
0	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	4	72	19	0	22	2.647027
1	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	6	42	19	1	42	2.366201
2	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	5	396	19	2	47	2.317352
3	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	8	361	19	0	26	2.574476
4	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	8	361	19	2	34	2.457971

5 rows × 23799 columns

In [9]:

```
cols = [i for i in range(0, 23708)]

df_tfidf = df_vec_without_stopwords[cols].multiply(df_vec_without_stopwords.idf.values, axis=0)
df_tfidf['evals'] = df_vec_without_stopwords.evals.values
df_tfidf.head()
```

Out[9]:

	0	1	2	3	4	5	6	7	8	9	...	23699	23700	23701	23702	23703	23704	23705	23706	23707	evals
0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0
1	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	2
3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0
4	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	2

5 rows × 23709 columns

In [ ]:

```
# classify_quote(df_tfidf)
```

Тип векторизации: булевские векторы

Результатом булевской модели ожидается датафрейм с единицами в случае появления слова в цитате

In [22]:

```
df_vec = df_vec_without_stopwords.iloc[:, :-2]
df_vec_without_evals = df_vec.iloc[:, :-1]
df_boolvec = df_vec_without_evals.mask(df_vec_without_evals > 0, 1) # здесь заменяем в датафрейме на единички все значения выш
е 0
df_boolvec['evals'] = df_vec_without_stopwords.evals.values
df_boolvec.head()
```

Out[22]:

	0	1	2	3	4	5	6	7	8	9	...	23787	23788	23789	23790	23791	23792	slash_nums	base_site_names_le	domens_le	evals
0	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	1	1	1	0
1	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	1	1	1	1
2	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	1	1	1	2
3	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	1	1	1	0
4	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	1	1	1	2

5 rows × 23797 columns

In [23]:

```
classify_quote(df_boolvec)
```

```
f1-macro:  0.3089296236452318
f1-micro:  0.4562841530054645
```

Выводы

1) Алгоритм Random Forest выдаёт низкие результаты классификации на больших текстовых коллекицях и не рекомендуется к использованию в задаче анализа тональностей. Тем не менее, если сравнивать результаты при трёх разных типах векторизации, наиболее высокие из них показывает TF-IDF без использования стоп-слов.
2) Дополнительные признаки не вносят существенный вклад в повышение качества классификации;
3) С увеличением глубины деревьев качество повышается;
4) Наиболее оптимальным является разбиение обучающей и тестовой выборок в соотношении 70/30.

		f1-макро	**f1-микро**
Частотная модель	**Со стоп-словами**	0.296	0.461
	Без стоп-слов	***0.321***	***0.470***
TF-IDF	**Со стоп-словами**	0.311	0.469
	Без стоп-слов	***0.324***	***0.475***
Булевские векторы	**Со стоп-словами**	0.298	0.445
	Без стоп-слов	***0.308***	***0.456***