Анна Зуева
БКЛ 151

Отчет по заданию S_Preprocessing

Препроцессинг

1) Параметры проведенных экспериментов:

Подчеркнуты те параметры, которые были изменены по сравнению с предыдущими экспериментами.

(1) **Baseline**
preprocessing: -
vectorizer: CountVectorizer
classifier: LogisticRegression (penalty='l1', C=0.1)
(с семинара)

(2) preprocessing: -
vectorizer: TfidfVectorizer
classifier: LogisticRegression (penalty='l1', C=1)
(с семинара)

(3) preprocessing: лемматизация (pymorphy)
vectorizer: TfidfVectorizer
classifier: LogisticRegression (penalty='l1', C=1)

(4) preprocessing: лемматизация (mystem)
vectorizer: TfidfVectorizer
classifier: LogisticRegression (penalty='l1', C=1)

(5) preprocessing: очистка стоп-слов (nltk.corpus.stopwords.words('russian')
vectorizer: TfidfVectorizer
classifier: LogisticRegression (penalty='l1', C=1)

(6) preprocessing: очистка стоп-слов (nltk.corpus.stopwords.words('russian') + лемматизация (mystem)
vectorizer: TfidfVectorizer
classifier: LogisticRegression (penalty='l1', C=1)

(7) preprocessing: -
vectorizer: TfidfVectorizer (ngram_range=(1,2))
classifier: LogisticRegression (penalty='l1', C=1)

(8) preprocessing: лемматизация (mystem)
vectorizer: TfidfVectorizer (ngram_range=(1,2))
classifier: LogisticRegression (penalty='l1', C=1)

(9) preprocessing: стемминг (nltk.stem.snowball.RussianStemmer)

vectorizer: TfidfVectorizer ()
classifier: LogisticRegression (penalty='l1', C=1)

(10) preprocessing: стемминг (nltk.stem.snowball.RussianStemmer), очистка стоп-слов (nltk.corpus.stopwords.words('russian')
vectorizer: TfidfVectorizer ()
classifier: LogisticRegression (penalty='l1', C=1)

(11) preprocessing: стемминг (nltk.stem.snowball.RussianStemmer), очистка стоп-слов (nltk.corpus.stopwords.words('russian')
vectorizer: TfidfVectorizer (ngram_range=(1,2))
classifier: LogisticRegression (penalty='l1', C=1)

2) Сравнительная таблица качества при прогонах с разными условиями:

В самом левом столбце указан номер проведенного эксперимента.
Во втором столбце указано, что было изменено в эксперименте.
Начиная со второго эксперимента используется TfidfVectorizer и LogisticRegression (penalty='l1', C=1).

		avg. precision	**avg. recall**	**avg. f1-score**	**Макросредняя F1 мера**	**Микросредняя F1 мера**
1	Baseline	0,62	0,64	0,61	0,46306421211	0,63875365141
2	TfidfVectorizer LogisticRegression (penalty='l1', C=1)	0,65	0,67	0,65	0,51726040086	0,67088607595
3	+ лемматизация (pymorphy)	0,65	0,66	0,65	0,53791199699	0,66163583252
4	+ лемматизация (mystem)	0,66	0,67	0,66	0,54773571741	0,66747809153
5	+ очистка стоп-слов	0,65	0,64	0,61	0,48991340627	0,63631937683
6	+ очистка стоп-слов + лемматизация (mystem)	0,64	0,64	0,62	0,5116365877	0,64021421616
7	TfidfVectorizer (ngram_range=(1,2))	0,64	0,66	0,64	0,51169841746	0,65968841285
8	+ лемматизация (mystem) + TfidfVectorizer (ngram_range=(1,2))	0,65	0,66	0,65	0,5431998649	0,65920155794
9	+ стемминг (nltk.stem.snowball.RussianStemmer)	0,65	0,67	0,65	0,54701465976	0,66553067186

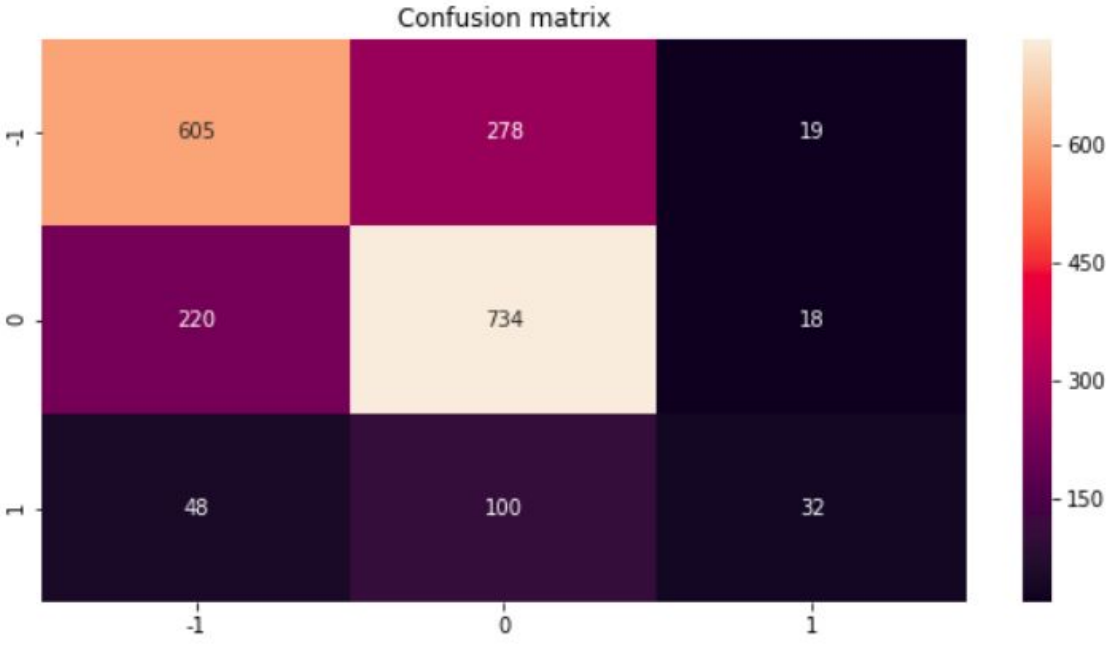
10	+ стемминг (nltk.stem.snowball.RussianStemmer) + очистка стоп-слов	0,65	0,67	0,65	0,54703026288	0,66553067186
11	+ стемминг (nltk.stem.snowball.RussianStemmer) + очистка стоп-слов + TfidfVectorizer (ngram_range=(1,2))	0,65	0,66	0,65	0,53618485066	0,66455696203

Как можно заметить, из всех 11 экспериментов лучше всего показал себя (4) эксперимент (лемматизация с mystem), за исключением микросредней F1 меры, которая была наибольшей на (2) эксперименте (TfidfVectorizer(), LogisticRegression(penalty='l1', C=1)):

(4) эксперимент (лемматизация с mystem) 0,66747809153,
(2) эксперимент (без лемматизации) 0,67088607595.

Для дальнейших целей будет использован вариант с лемматизацией с помощью mystem.

Анализ confusion matrix



Положительные отзывы определяются хуже всего (всего около 18% (32 из 180 отзывов) определились правильно), и это, скорее всего, можно объяснить тем, что они хуже всего представлены в обучающей выборке. Чаще положительные определяются как нейтральные отзывы (около 56% (100 из 180 отзывов)) и как негативные (около 27% (48 из 180 отзывов)).

Что касается негативных и нейтральных отзывов, то лучше всего определяются нейтральные отзывы (около 76% (734 из 972 отзывов)) и чуть менее хорошо находятся негативные (около 67% (605 из 902 отзывов)).

Анализ топ 10 признаков

Значимые слова для класса - -1
['оштрафовать', 'сбой', 'tele2', 'подорожать', 'гавно', 'повышать', 'сука', 'не', 'восстановление', 'заблокировать']

В принципе, все кажется подходящим, кроме 'восстановление'.

Значимые слова для класса - 0
['доллар', 'гавно', 'иа', 'подорожать', 'сбой', 'уточнять', 'восстановление', 'оштрафовать', 'ловить', 'pomogite']

Важные признаки для классов 0 и -1 повторяются (5 слов из 10). Это, скорее всего, и является причиной того, что негативные отзывы чаще всего путаются с нейтральными и наоборот, но не с позитивными. Непонятно, что значит 'иа' и также почему сюда попало 'pomogite'.

Значимые слова для класса - 1
['спасибо', 'защита', 'любить', 'узбекистан', 'подарок', 'расход', 'доллар', 'бесплатный', 'пожалуйста', 'хороший']

В принципе, все кажется подходящим. Только 'доллар' присутствует, как у класса 0, так и у класса 1.

Подбор параметров в классификаторе

Параметры в классификаторе подбирались с помощью sklearn.model_selection.GridSearchCV.

Были переданы следующие значения параметров:

C	1.e-4, 1.e-3, 1.e-2, 1.e-1, 1, 2, 10, 50, 100, 1000;
penalty	l1, l2.

Лучшим оказалось сочетание параметров penalty = l2 и C = 10:
среднее значение по сплитам f1_macro: 0.645807;
среднее значение по сплитам f1_micro: 0.725784.

Анализ топ 10 признаков

Значимые слова для класса - -1
['гавно', 'оштрафовать', 'не', 'tele2', 'сбой', 'сука', 'повышать', 'плохо', 'подорожать', 'заблокировать']

Все кажется подходящим, даже ушло непонятное 'восстановление'.

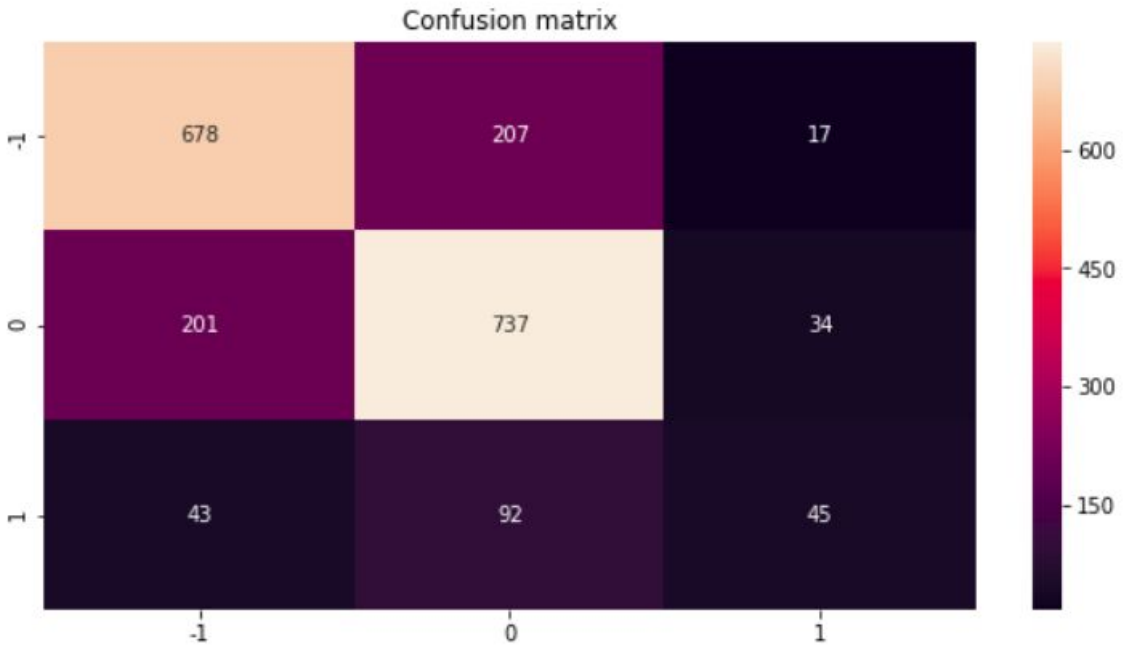
Значимые слова для класса - 0
['гавно', 'доллар', 'любить', 'оштрафовать', 'восстановление', 'иа', 'ловить', 'сбой', 'даже', 'заебывать']

Опять повторяются важные признаки у классов -1 и 0, но теперь уже меньше: 3 слова из 10. Осталось непонятное 'иа', также не очень ясно, почему в класс 0 попало 'гавно' и 'заебывать'.

Значимые слова для класса - 1
['спасибо', 'любить', 'защита', 'узбекистан', 'радовать', 'бесплатный', 'подарок', 'хороший', 'благодарить', 'зарабатывать']

Все кажется подходящим. Но 'любить' повторяется у классов 0 и 1.

Анализ confusion matrix



По сравнению с предыдущей матрицей ошибок значительно улучшилось определение негативных отзывов, также заметно улучшилось определение позитивных и немного увеличилось количество правильно определенных нейтральных отзывов.

Для негативных отзывов уменьшилось как количество причисленных их к нейтральным (можно объяснить тем, что меньше важных признаков стало пересекаться у этих двух классов), так и к позитивным. Тогда как больше нейтральных отзывов стало ошибочно классифицироваться как положительные за счет уменьшения количества ошибочно причисленных к негативным. Для

положительных отзывов уменьшилось как количество причисленных их к негативным, так и к нейтральным.