Анастасия Костяницына
БКЛ -151

Отчет
Препроцессинг

№1

1) В ходе работы над программой для определения тональности твитов использовалась LogisticRegression, одним из параметров которой является C - коэффициент регуляризации. Был произведен перебор возможных значений данного параметра с помощью GridSearchCV и определено наилучшее значение: C = 10, которое и использовалось в дальнейшем анализе.

2) Далее был определен baseline для программы: на вход CountVectorizer() подавался чистый текст без всяких изменений. Получились следующие значения:

	precision	recall	f1-score	support
-1	0.73	0.67	0.70	902
0	0.66	0.75	0.71	972
1	0.35	0.26	0.30	180
avg / total	0.67	0.67	0.67	2054

Макросредняя F1 мера - 0.56662297778040 13
Микросредняя F1 мера - 0.6713729308666018

3) Для улучшения качества работы программы была произведена нормализация: токинизация, удаление пунктуации, лемматизация, удаление стоп-слов. Стоп-слова были взяты из nltk.corpus. Обучили CountVectorizer и LogisticRegression, получили следующие результаты:

	precision	recall	f1-score	support
-1	0.75	0.61	0.67	902
0	0.65	0.77	0.71	972
1	0.35	0.32	0.33	180
avg / total	0.67	0.66	0.66	2054

Макросредняя F1 мера - 0.5718232190860312
Микросредняя F1 мера - 0.6626095423563778

Результат ухудшился. Это может быть связано с тем, что стоп слова, которые используются по умолчанию включают в себя такие слова, которые могут повлиять на определение тональности твита. Например, не, которое в сочетании с глаголом, может кардинально поменять смысл всего сообщения (понравилось vs не понравилось). Попробуем посчитать без удаления стоп-слов.

4) Без удаления стоп-слов:

	precision	recall	f1-score	support
-1	0.73	0.67	0.70	902
0	0.68	0.76	0.71	972
1	0.42	0.33	0.37	180
avg / total	0.68	0.68	0.68	2054

Макросредняя F1 мера - 0.5945907742000814
Микросредняя F1 мера - 0.6801363193768257

Нам удалось увеличить качество работы программы.

№ 2

Посмотрим, какие результаты получатся на таких же данных для TfidfVectorizer и сравним их с CountVectorizer.

Data	CountV (accuracy)	TfidfV (accuracy)
Без изменений	0.671373	0.663583
Нормализация - стоп-слова	0.662610	0.647517
Нормализация + стоп-слова	0.680136	0.671373

Как видно из таблицы, наилучший результат получился при Нормализация с стоп-словами и использованием CountVectorizer.

№ 3

Для лучшего алгоритма взглянем на топ 10 признаков и на confusion_matrix.

Список важных признаков:

Значимые слова для класса - -1

['задолженность', 'amaranth815', 'оштрафовать', 'атаковать', 'сбой', 'турбокнопка', 'уезжать', 'добиться', 'расторгнуть', 'испытывать']

Значимые слова для класса - 0

['650p', 'топливо', 'расторгнуть', 'гавный', 'испытывать', 'вносить', 'достоверно', 'задолженность', 'слогана', 'инноватор']

Значимые слова для класса - 1

['lizinastusha', 'здорово', 'адекватный', '6j3mfzcy5u', 'мило', 'топливо', 'расширяться', 'инноватор', 'youdicuudv', ‘эфирный']

Достаточно низкий accuracy (0,68) может быть объяснен тем, что, как видно из списка важных признаков, что некоторые слова являются важными сразу для нескольких классов. Например, *испытывать, расторгнуть* для -1 и 0; *инноватор, топливо* для 0 и 1. Помимо этого, в список значимых признаков для отрицательного класса входят такие отрицательно маркированные слова, как *атаковать, сбой, оштрафовать, задолженность*, которые вряд могут встретиться в положительных твитах, что помогает хорошо отделить этот класс. Для нейтрального и положительного таких слов практически нет, кроме как здорово и мило у положительного. Это может повлиять на работу программы. Проверим свои догадки с помощью матрицы ошибок.

Confusion_matrix:

```
array([[602, 269,  31],
       [186, 735,  51],
       [ 37,  83,  60]])
```

На основе результатов confusion_matrix видно, что у программы действительно возникают трудности в разграничивании нейтрального класса от положительного.

В списке ключевых слов кажется мусорным слово топливо, но все зависит от контекста.

№ 4

Произведем отбор признаков:

Удалим ненужные признаки, общие для всех классов. Для этого для каждого класса были найдены 2500 слов самых незначимых признаков, найдено пересечение этих множеств. Это пересечение - и есть новые стоп-слова, которые были удалены при нормализации.

Результаты:
accuracy - 0.6815968841285297

	precision	recall	f1-score	support
-1	0.73	0.67	0.70	902
0	0.68	0.76	0.72	972
1	0.43	0.33	0.37	180
avg / total	0.68	0.68	0.68	2054

Макросредняя F1 мера - 0.5804887254021792
Микросредняя F1 мера - 0.6713729308666018

Нам удалось еще увеличить качество работы программы.