Параметры проведенных экспериментов (лемматизация +/-, что использовалось при лемматизации, откуда стоп-список, какие типы обработки применялись)

Наша задача: избавиться от разных ссылок, включая ники людей, так как они не дают никакой информации о тональности текста (удалены адреса, ники). Стоп-список взят из nltk (туда входят всякие служебные части речи, которые не дают никакой информации о тональности текста. Использовался pymorphy, так как при незнакомых словах он предугадывает их начальную форму, mystem в этом случае проигрывает (для твитов это важно – там много разных новых слов, не вошедшие ещё в словарь, также mystem давно не обновлялся). Пользуюсь специальным токенайзером для твиттера, так как он учитывает "разговорный" язык твитов (в том числе считает смайлики, не разделяет слова с апострофами). При стемминге использовался SnowballStemmer.

Сравнительная таблица качества при прогонах с разными условиями

CountVectorizer

	precision	recall	f1-score	support
-1	0.69	0.59	0.64	902
0	0.61	0.80	0.69	972
1	0.30	0.03	0.06	180
avg / total	0.62	0.64	0.61	2054

Макросредняя F1 мера - 0.46306421211286786
Микросредняя F1 мера - 0.6387536514118792

TfidfVectorizer

	precision	recall	f1-score	support
-1	0.70	0.69	0.70	902
0	0.66	0.76	0.71	972
1	0.37	0.09	0.15	180
avg / total	0.65	0.67	0.65	2054

Макросредняя F1 мера - 0.5176166888561017
Микросредняя F1 мера - 0.6713729308666018

TfidfVectorizer с нормализацией (лемматизация)

	precision	recall	f1-score	support
-1	0.76	0.56	0.64	902
0	0.62	0.86	0.72	972
1	0.49	0.12	0.19	180
avg / total	0.67	0.66	0.64	2054

Макросредняя F1 мера - 0.51797162790908
Микросредняя F1 мера - 0.6635832521908471

TfidfVectorizer с нормализацией (лемматизация) и занулением неважных признаков

	precision	recall	f1-score	support
-1	0.72	0.61	0.66	902
0	0.65	0.76	0.70	972
1	0.36	0.31	0.33	180
avg / total	0.65	0.65	0.65	2054

Макросредняя F1 мера - 0.564560794955486
Микросредняя F1 мера - 0.6538461538461539

TfidfVectorizer с нормализацией (стемминг) и занулением неважных признаков

	precision	recall	f1-score	support
-1	0.74	0.62	0.68	902
0	0.67	0.77	0.72	972
1	0.30	0.29	0.29	180
avg / total	0.67	0.66	0.66	2054

Макросредняя F1 мера - 0.5615447238186427
Микросредняя F1 мера - 0.6630963972736125

Лучшая модель

	precision	recall	f1-score	support
-1	0.74	0.66	0.70	902
0	0.70	0.76	0.73	972
1	0.31	0.34	0.32	180
avg / total	0.68	0.68	0.68	2054

Макросредняя F1 мера - 0.5826810202668083
Микросредняя F1 мера - 0.6777020447906524

Итог:

	CountVectorizer	TfidfVectorizer	TfidfVectorizer с нормализацией (лемматизация)	Зануление неважных признаков	Лучшая модель	Стемминг
0	0.4631	0.5176	0.5180	0.5654	0.5814	0.5618
1	0.6388	0.6714	0.6636	0.6553	0.6767	0.6631

Анализ топ 10 признаков: привести списки и прокомментировать, что кажется правильным, что мусорным

Значимые слова для класса - -1

['задолженность', 'оштрафовать', 'неужели', 'говно', 'tele', 'расценка', '#сбербанк', 'хуй', 'атаковать', 'угроза']

Значимые слова для класса - 0

['задолженность', 'гавный', 'восстановление', 'доллар', 'вспомнить', 'заебал', 'ловить', 'иа', 'расторгнуть', 'оштрафовать']

Значимые слова для класса - 1

['свести', 'понравиться', 'топ', 'подарочек', 'вброса', 'мтс-россия', 'интернет-магазин', 'интернетом-доступ', 'защита', 'ёмкость']

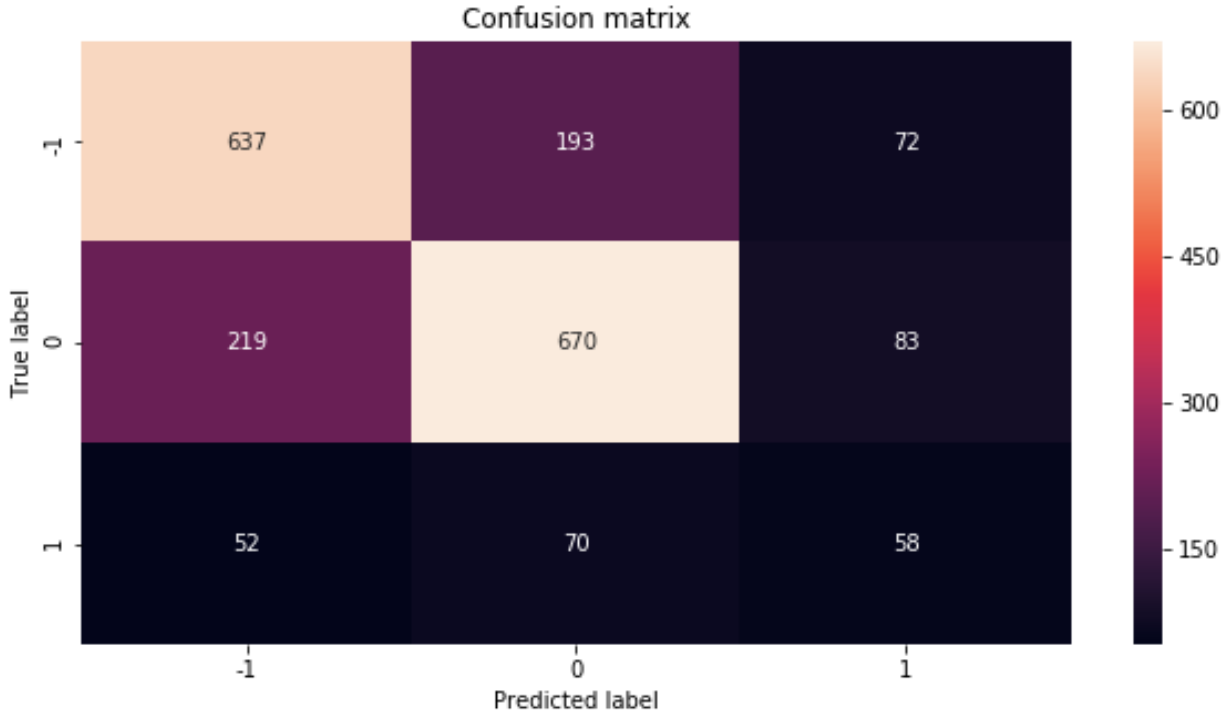
Для классов «-1» и «1» ключевые слова, действительно, являются отрицательного/положительного оттенка. Не понятно, почему для класса «0» выбраны такие ключевые слова, но, кажется, что туда вошли и слова и отрицательной, и положительной тональности.

Для п. 2.3. - указать, какие эвристики применялись для улучшения

TfidfVectorizer: ngram_range (учитывание n-грамм (в итоге выбор пал на минимальный размах n-грамм); token_pattern='\S+' (учитывает знаки препинания, смайлики: для нас это важно, так как учитывает, когда много, например, знаков восклицания)

LogisticRegression: class_weight=balanced (у нас несбалансированные классы, как было раньше указано, это сбалансирует выборку)

Анализ fp и fn - предложить объяснение ошибочного попадания твита в класс, предложить какие-то улучшения алгоритма для борьбы с ошибками, из-за которых, вам кажется, что твит был расклассифицирован ошибочно.



Это confusion matrix для лучшей модели. Видим, что лучше отделяется нейтральный класс. Ошибка может возникнуть из-за того, что, во-первых, изначально на тестовой выборке классы у нас были несбалансированные. Попытки сбалансировать классы гиперпараметром не всегда улучшают результат метрики. Решение: увеличить классы и сбалансировать их. Во-вторых, возможно, когда мы использовали параметр, классификатор в качестве токена брал любую последовательность символов, что привело к ошибках (некоторые знаки препинания можно было удалить изначально). В-третьих, можно использовать другие классификаторы (например, RandomForestClassifier или какой-нибудь линейный классификатор).