Сделали бейзлайн без нормализации.

- **CountVectorizer & LogisticRegression**

	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 - 0.463064212113
Микросредняя F1 мера - 0.638753651412

- **TF-IDF & LogisticRegression**

	precision	recall	f1-score	support
-1	0.70	0.69	0.69	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 - 0.517260400863
Микросредняя F1 мера - 0.670886075949

Теперь добавляем лемматизацию.
Для того, чтобы улучшить результаты, решила очистить выборку от всего "мусора".

- От знаков препинания, которые есть в string.punctuation, с добавлением кавычек-лапок и кавычек-ёлочек, которые встречаются в выборке, а так же добавила троеточие;
- От ссылок. Постаралась рассмотреть все варианты: и http://, и https://, и просто www. ;
- От обращений (начинаются с @);
- От всех английских слов;
- От лишних пробелов/переносов/табуляций.

TF-IDF & LogisticRegression + лемматизация

	precision	recall	f1-score	support
-1	0.75	0.55	0.63	902
0	0.62	0.86	0.72	972
1	0.52	0.13	0.21	180
avg / total	0.67	0.66	0.64	2054

Макросредняя F1 мера - 0.521260884222
Микросредняя F1 мера - 0.658227848101

Показатели у макросредней меры улучшились, а у микросредней меры ухудшились.

Потом решила делать с помощью **стемминга**, потому что с лемметизацией вышло как-то не очень:(Стемминг, по сравнению с лемматизацией, легче, потому что лемматизация опирается на словообразование. То есть она определает часть речи и применяет к слову различные способы нормализации. Стемминг же ищет флективную форму в своей таблице поиска, что значительно упрощает и ускоряет работу алгоритма. Кроме того, стемминг хорошо обрабатывает исключения, что в живой речи нам только на руку.

- В предварительной "очистке" текста ничего не меняла.
- Для твиттера есть отдельный токанайзер (TweetTokenizer), который хорошо распознаёт смайлики и хорошо определяет тон с их помощью и с помощью пунктуации. Используем его!

Те же самые **TF-IDF** и **LogisticRegression + стемминг**:

	precision	recall	f1-score	support
-1	0.72	0.71	0.72	902
0	0.68	0.78	0.73	972
1	0.61	0.15	0.24	180
avg / total	0.69	0.70	0.68	2054

Макросредняя F1 мера - 0.561677964639
Микросредняя F1 мера - 0.696202531646

Для сравнения, **TF-IDF** и **LogisticRegression + лемматизация**:

	precision	recall	f1-score	support
-1	0.75	0.55	0.63	902
0	0.62	0.86	0.72	972
1	0.52	0.13	0.21	180
avg / total	0.67	0.66	0.64	2054

Макросредняя F1 мера - 0.521260884222
Микросредняя F1 мера - 0.658227848101

И просто **TF-IDF** и **LogisticRegression**

	precision	recall	f1-score	support
-1	0.70	0.69	0.69	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 - 0.517260400863
Микросредняя F1 мера - 0.670886075949

Как видно, **TF-IDF** и **LogisticRegression + стемминг** даёт лучшие результаты как по макросредней мере, так и по микросредней.

Теперь попробуем использовать другие классификаторы в комбинации с TF-IDF и стеммингом.

- **DecisionTree** (результаты ухудшились)

	precision	recall	f1-score	support
-1	0.63	0.62	0.62	902
0	0.63	0.65	0.64	972
1	0.20	0.17	0.18	180
avg / total	0.59	0.59	0.59	2054

Макросредняя F1 мера - 0.481050686251
Микросредняя F1 мера - 0.593476144109

- **RandomForest** (результаты ухудшились)

	precision	recall	f1-score	support
-1	0.69	0.48	0.57	902
0	0.58	0.83	0.69	972
1	0.42	0.09	0.15	180
avg / total	0.62	0.61	0.59	2054

Макросредняя F1 мера - 0.467094653922
Микросредняя F1 мера - 0.613437195716

- **SGD** (результаты очень хорошие!)

	precision	recall	f1-score	support
-1	0.71	0.77	0.74	902
0	0.72	0.73	0.73	972
1	0.54	0.26	0.35	180
avg / total	0.70	0.71	0.70	2054

Макросредняя F1 мера - 0.604001833634
Микросредняя F1 мера - 0.707400194742

Посмотрим на GridSearchCV с SVC.

Как показывает практика, лучшим среди параметров *kernel* подойдёт 'rbf' с C = до 1000. Но, давайте проверим (не по максимуму, но и не по минимуму):

params = [{'kernel':['poly'], 'C':[1.e-4, 2, 10, 100, 1000], 'degree':[2, 3, 5, 11]},

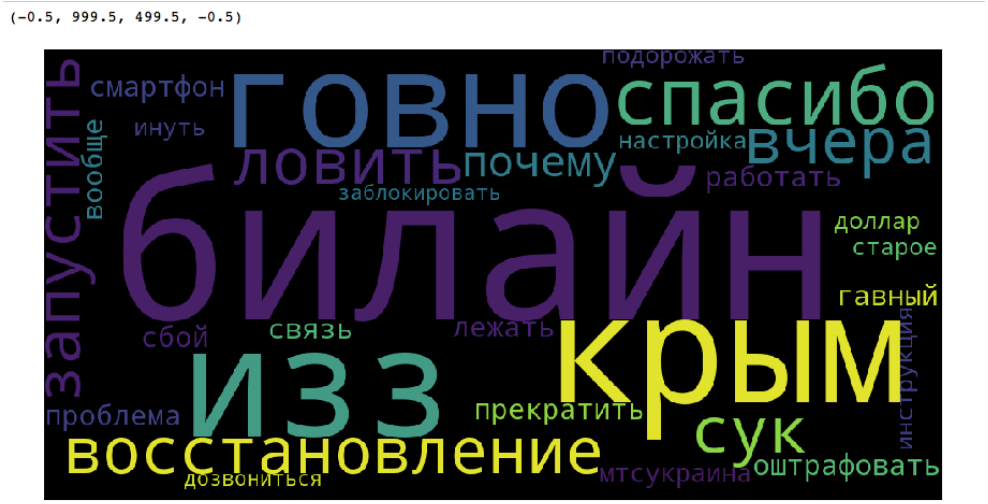
{'kernel':['rbf'], 'C':[1.e-4, 2, 10, 100, 1000], 'gamma':['auto', 1.e-4, 1.e-2]},

{'kernel':['linear'], 'C':[1.e-4, 2, 10, 100, 1000]}]

Так и получается:
`{'C': 10, 'gamma': 0.01, 'kernel': 'rbf'}`

И посмотрим на облака слов, для двух лучших методов:

TF-IDF и **LogisticRegression + стемминг**



TF-IDF и **SGD + стемминг**

```
(-0.5, 999.5, 499.5, -0.5)
```