

Machine learning lab2

Группа: MLP Lovers

Выполнили: Булатов Дмитрий, Кабанов Денис



1. Первичная обработка данных

Датасет - коллекции новостей AG. Он содержит 4 класса новостей: world, sport, business and sci-tech.

В датасете присутствует 120 тысяч образцов данных.



Колонки:

- Class Index - содержит идентификатор класса, к которому принадлежит новость
- Title - название новости
- Description - текст новости

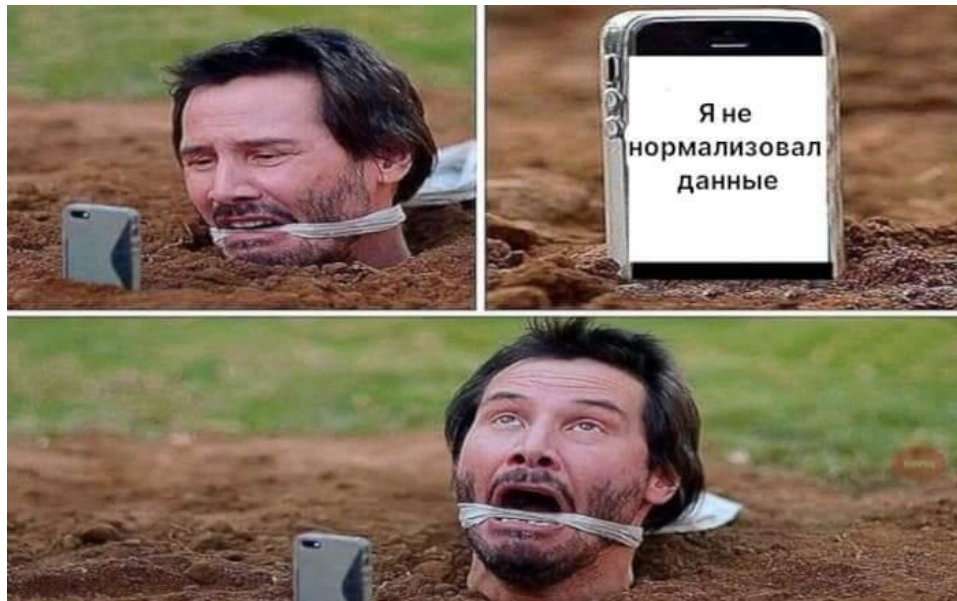
1. Первичная обработка данных

- Так как столбец Title не содержит явного типа новости, но может включать информацию для помощи в его определении, было решено объединить его с Description.
- Выполнен анализ встреченных символов в датасетах для дальнейшей очистки.
- Проведено 6 вариантов очистки датасета с последующей токенизацией:
 - без какой либо очистки
 - удаление стоп-слов, так как они часто встречаются и могут вносить “шум” в данные.
 - удаление знаков препинания, так как они не несут особой смысловой нагрузки и также встречаются во всех новостях
 - удаление чисел
 - удаление мусорных значений, таких как специальные символы и ссылки
 - полная очистка (все варианты удалений вместе)



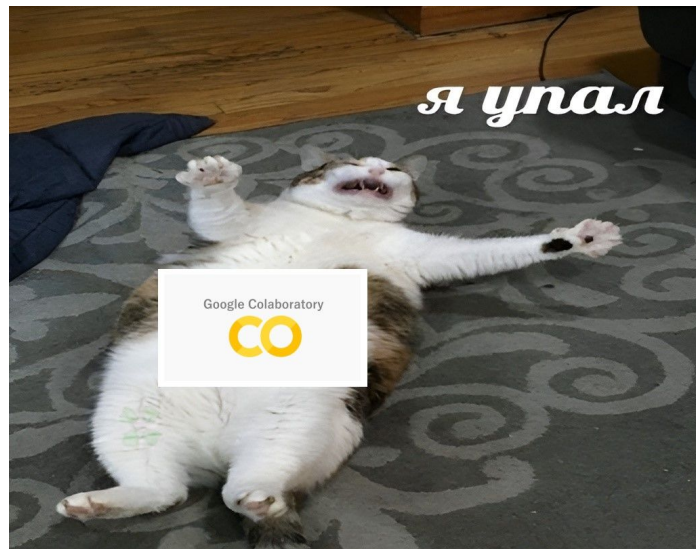
2. Нормализация данных

- Для каждого из предыдущих вариантов очистки датасета были рассмотрены следующие типы нормализации токенов:
 - без нормализации
 - стемминг (stemming)
 - лемматизация (lemmatization)



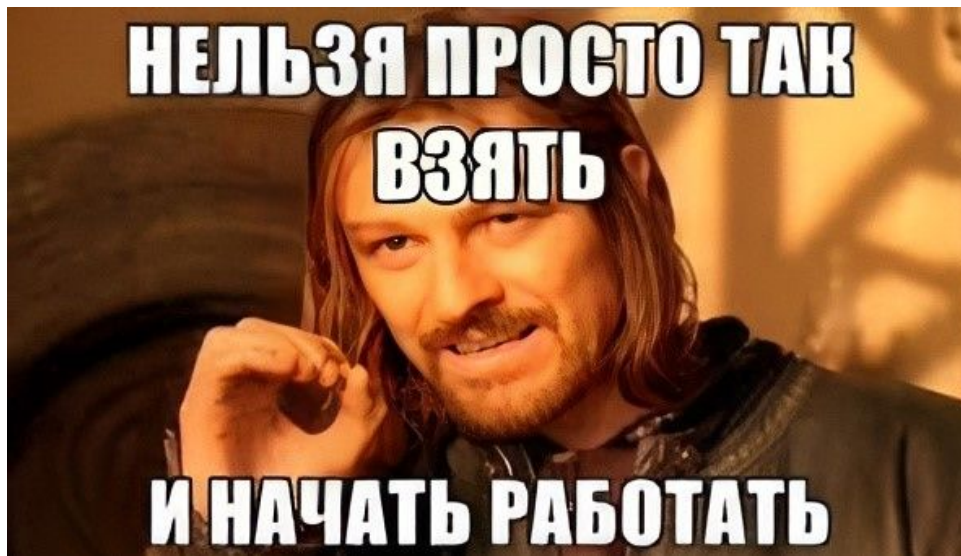
3. Векторизация данных

- Для векторизации токенов были использованы 4 варианта:
 - Count Vectorizer, просто считающий частоту встречи токена в новости
 - TF-IDF, дающий вес токену пропорционально частоте употребления его в документе и обратно пропорционален частоте употребления токена во всех документах коллекции
 - Word2Vec, модель, обучающаяся на датасете и создающая векторное представление токенов, основываясь на их позициях относительно других токенов
 - GloVe, словарь с уже заготовленными векторными представлениями слов



4. Проведение экспериментов

- В качестве модели для обучения был выбран Gradient Boosting Classifier, так как он хорошо работает с табличными данными и задачей multi-class classification.
- Общее количество проведённых экспериментов - 72:
6 вариантов очистки * 3 варианта нормализации * 4 варианта векторизации



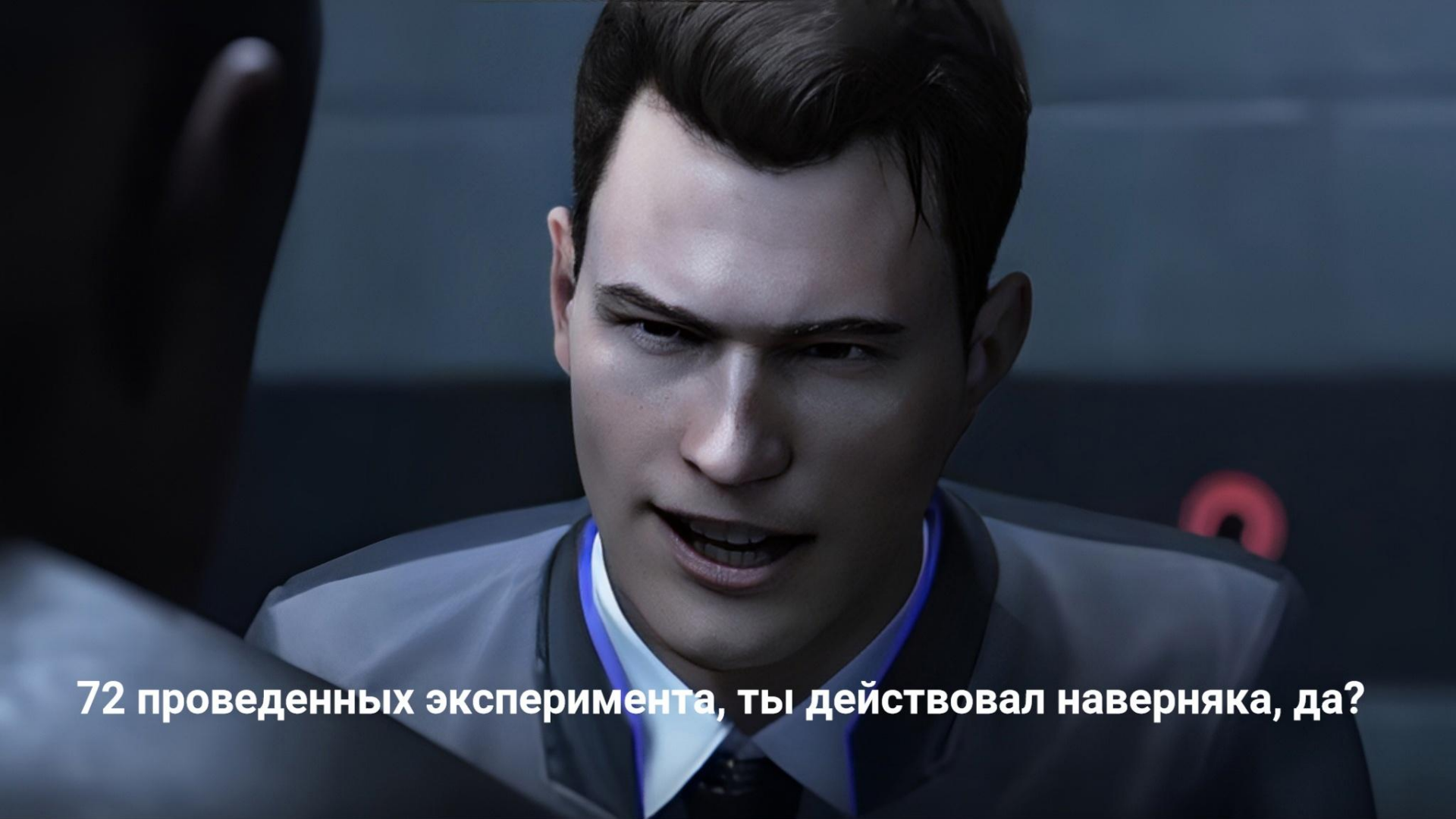
4. Проведение экспериментов

Получившийся F1 score:

	F1 score	Очистка	Нормализация	Векторизация
0	0.87921	digits	none	glove
1	0.87904	stop_words	none	glove
2	0.87896	all	none	glove
3	0.87842	stop_words	lem	count
4	0.87767	stop_words	stem	count
5	0.87746	trash	none	glove
6	0.87692	stop_words	lem	glove
7	0.87692	punctuation	none	glove
8	0.87692	all	lem	glove
9	0.87667	none	stem	count
10	0.87658	none	none	glove
11	0.87558	digits	stem	count
12	0.87546	trash	stem	count
13	0.87546	trash	lem	glove
14	0.87538	digits	lem	count
15	0.87479	punctuation	lem	glove
16	0.87404	none	lem	count
17	0.87321	none	lem	glove
18	0.87308	trash	lem	count
19	0.87292	all	stem	count
20	0.87258	digits	lem	glove
21	0.87204	stop_words	none	count
22	0.87129	all	lem	count
23	0.87125	punctuation	stem	count

24	0.87117	stop_words	stem	tf_idf
25	0.87117	trash	none	count
26	0.87050	none	none	count
27	0.87021	punctuation	lem	count
28	0.86962	digits	none	count
29	0.86887	stop_words	lem	tf_idf
30	0.86879	none	stem	tf_idf
31	0.86817	trash	lem	tf_idf
32	0.86750	digits	stem	tf_idf
33	0.86738	trash	stem	tf_idf
34	0.86700	all	stem	tf_idf
35	0.86696	all	none	count
36	0.86683	digits	lem	tf_idf
37	0.86642	none	lem	tf_idf
38	0.86613	punctuation	stem	tf_idf
39	0.86492	punctuation	none	count
40	0.86429	all	lem	tf_idf
41	0.86296	stop_words	none	tf_idf
42	0.86246	trash	none	tf_idf
43	0.86167	punctuation	lem	tf_idf
44	0.86008	digits	none	tf_idf
45	0.85975	all	none	tf_idf
46	0.85967	none	none	tf_idf
47	0.85521	all	stem	glove

48	0.85475	trash	stem	glove
49	0.85442	punctuation	none	tf_idf
50	0.85404	none	stem	glove
51	0.85221	stop_words	stem	glove
52	0.85142	punctuation	stem	glove
53	0.84971	digits	stem	glove
54	0.79425	all	stem	word2vec
55	0.78371	all	lem	word2vec
56	0.76729	stop_words	stem	word2vec
57	0.76642	all	none	word2vec
58	0.75875	trash	stem	word2vec
59	0.74663	stop_words	lem	word2vec
60	0.74242	punctuation	stem	word2vec
61	0.74221	trash	lem	word2vec
62	0.73929	digits	stem	word2vec
63	0.73638	none	stem	word2vec
64	0.72625	punctuation	lem	word2vec
65	0.72308	digits	lem	word2vec
66	0.71171	stop_words	none	word2vec
67	0.71000	trash	none	word2vec
68	0.70917	none	lem	word2vec
69	0.69208	punctuation	none	word2vec
70	0.68963	digits	none	word2vec
71	0.67808	none	none	word2vec

A close-up shot of a man with dark hair, wearing a grey suit jacket over a white shirt and a dark tie. He has a serious, intense expression on his face, looking slightly to the left. The background is dark and out of focus, with a red circular light visible on the right side.

72 проведенных эксперимента, ты действовал наверняка, да?

4. Проведение экспериментов

Вариант очистки	Средний F1 score	Средняя позиция в таблице
Без очистки	0.828	36.916
Удаление пунктуации	0.829	40.500
Удаление чисел	0.830	36.250
Удаление мусора	0.836	34.416
Удаление стоп-слов	0.838	30.083
Полная очистка	0.846	34.833



- Лучшая очистка — очистка всего (по F1), за ней идут стоп-слова (по положению в таблице), мусор и числа (первое место в таблице).
- Хуже всего себя показало отсутствие очистки (по F1) или удаление только пунктуации (по позиции в таблице).

4. Проведение экспериментов

Вариант нормализации	Средний F1 score	Средняя позиция в таблице
Без нормализации	0.8287	36.083
Стемминг	0.8380	38.458
Лемматизация	0.8383	31.958

- Лучшая нормализация — лемматизация и по среднему F1 score и по средней позиции в таблице.
- Худшая нормализация — её отсутствие (по F1) или стемминг (по позиции в таблице).



4. Проведение экспериментов

Вариант векторизации	Средний F1 score	Средняя позиция в таблице
Count Vectorizer	0.872	19.555
TF-IDF	0.864	37.444
Word2Vec	0.734	62.500
GloVe	0.868	22.500



- Лучший по F1 и позиции в таблице — Count. Однако GloVe занимает больше позиций в топ-10 (8 из 10, в том числе и первая тройка). Можно также заметить, что GloVe получил низкие результаты только при комбинации с стеммингом.
- TF-IDF показал средние результаты.
- Худшим вариантом векторизации оказался Word2Vec, он занял последние позиции при любом варианте очистки+нормализации. Это может быть связано с малым временем обучения Word2Vec модели.

4. Проведение экспериментов

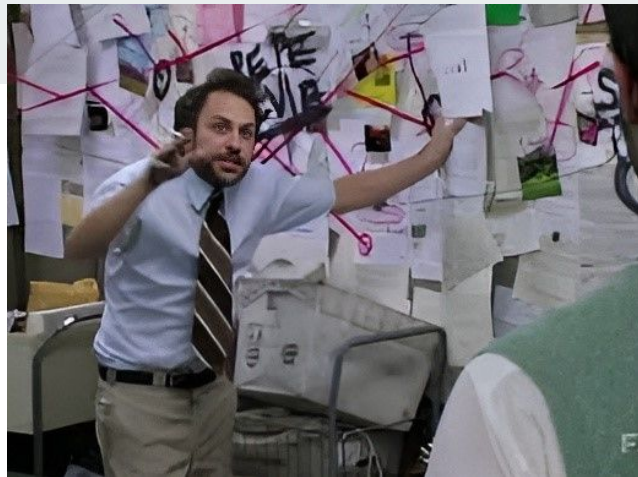
Лучшей субъективной комбинацией оказалась:

- полная очистка + лемматизация + GloVe векторизация

По таблице:

- очистка только + отсутствие нормализации + GloVe векторизация

Однако разница между ними небольшая, только начиная с третьего знака после запятой.

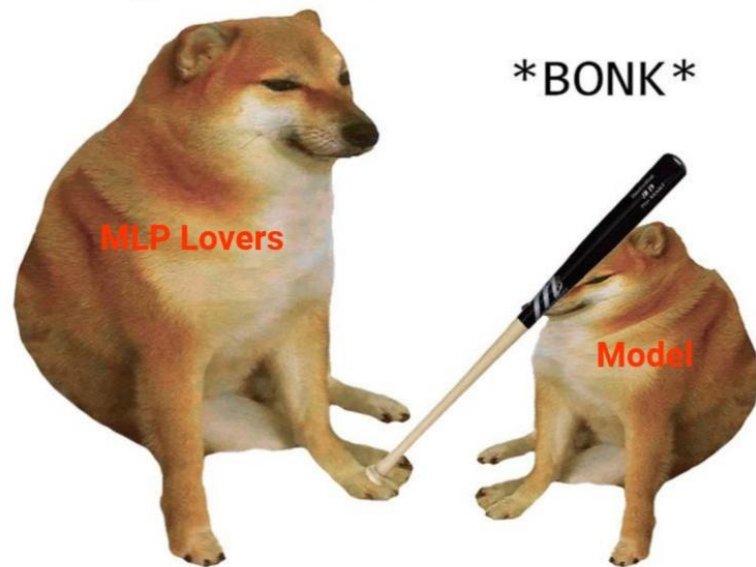


5. Анализ ошибок

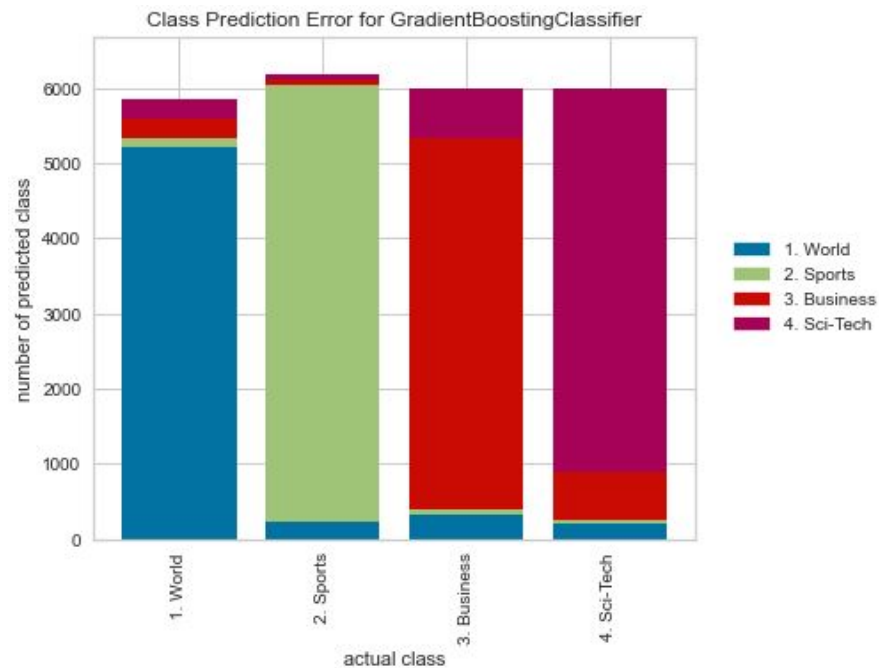
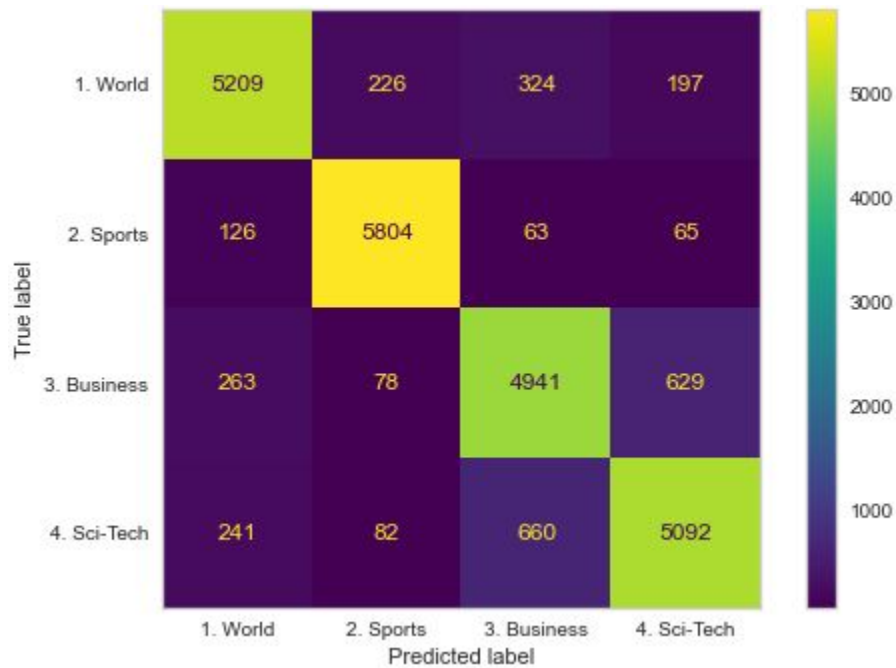
Анализ ошибок проводился у выбранной лучшей комбинации вариантов предобработки:

полная очистка + лемматизация + GloVe векторизация

Почему так много ошибок?



5. Анализ ошибок



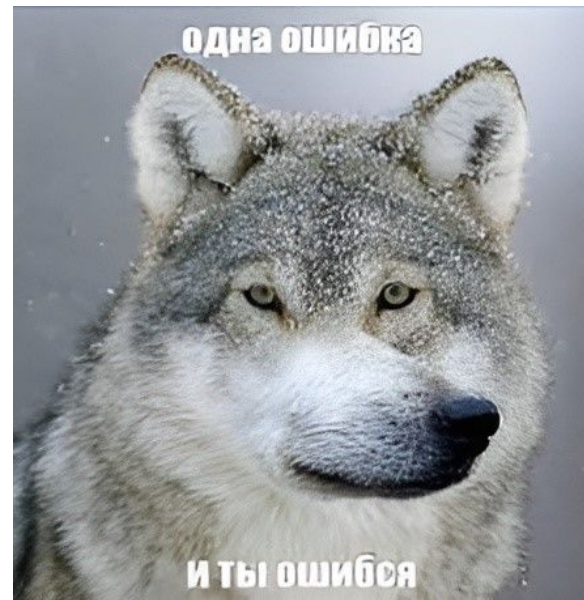
5. Анализ ошибок



- Для категории "World" ошибка $\sim 12.5\%$, из которых 26% - "Sci-Tech", 31% - "Sports", 43% - "Business".
- Лучше всего модель справилась с предсказанием новостей о "Sports", ошибка для данной категории составляет всего $\sim 5\%$, из которых половина приходится на "World" и по четверти на "Business" с "Sci-Tech".
- Новости из "Business" предсказывались с довольно большой ошибкой в 16.5%, основная масса которой пришлась на категорию "Sci-Tech" $\sim 65\%$, 27% на "World", 8% на "Sports".
- Категория "Sci-Tech" имеет ошибку, близкую к 16%. Четверть ошибочных предсказаний приходится на "World", 8% - "Sports", 67% - "Business".

5. Анализ ошибок

- Самая большая ошибка наблюдается для категорий "Business" и "Sci-Tech".
- Можно предположить, что векторы некоторых токенов в новостях "Business", "Sci-Tech", "World" довольно схожи или, что в этих категориях присутствует много одинаковых токенов, отчего общий вектор новости зашумляется и предсказания получаются неточными.



6. Собственная модель

- Модель состоит из четырёх линейных слоёв:
 - 1) получает на вход вектор и увеличивает его размер в 2 раза
 - 2) простое линейное преобразование без изменения размера
 - 3) сжимает размер в 4 раза
 - 4) снижает размер до 4 (=число категорий новостей)
- и сигмоиды для приведения чисел, полученных после линейных слоёв в область от 0 до 1 (в вероятность каждой категории)



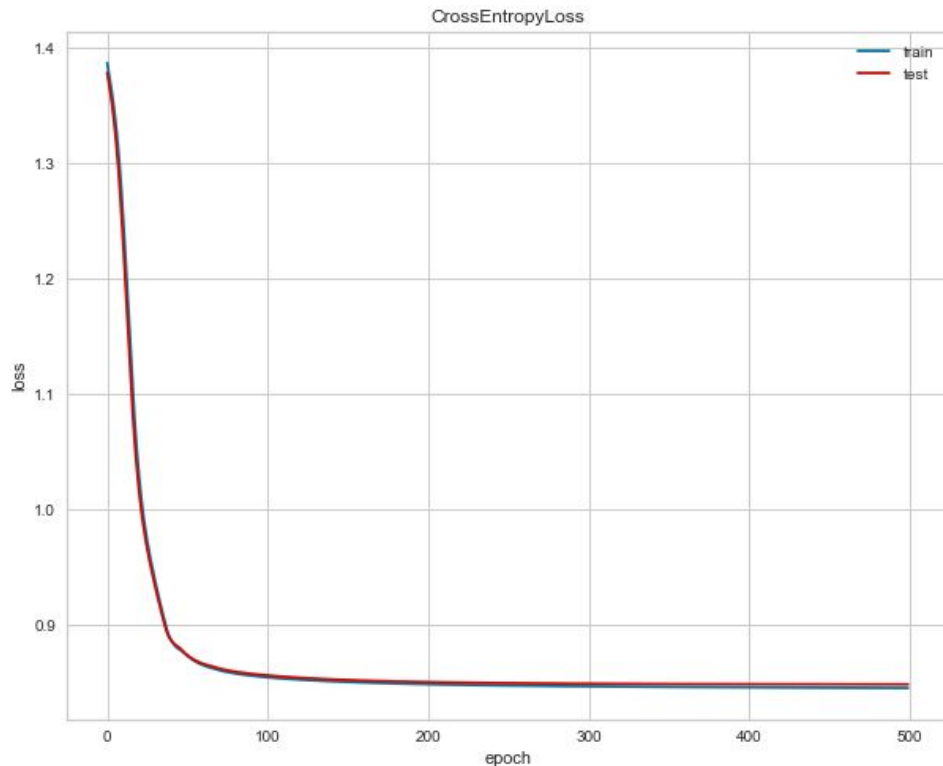
6. Собственная модель

- Обучение проводилось 500 эпох.
- Для подсчёта loss использовался CrossEntropyLoss, что на вход ожидает вероятность класса для всех 4 категорий новостей.
- F1 score на тестовых данных - 0.891

500 эпох, да?



А смысл....?



Полученный F1 score

При использовании Gradient Boosting Classifier:

- на тестовых данных - 0.87692
- на kaggle - 0.87234

При использовании “hand-made” модели:

- на тестовых данных - 0.89120
- на kaggle - 0.88938

P.S. предобработка = полная очистка
+ лемматизация + GloVe векторизация



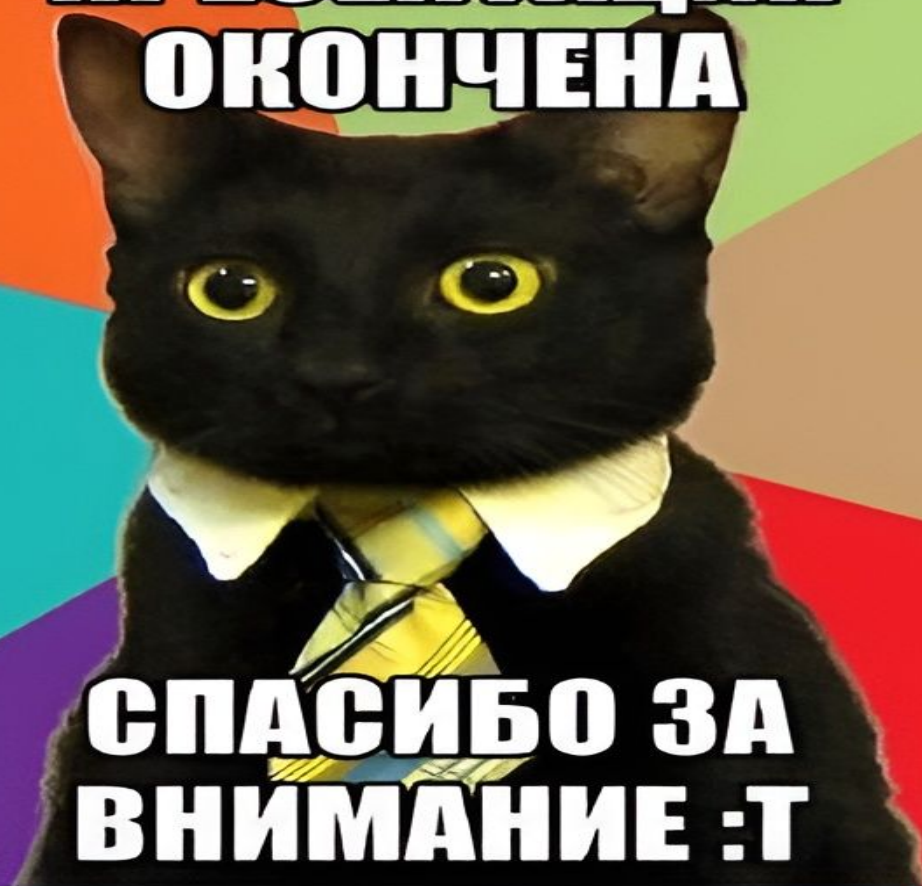
Выводы



Было проведено 72 эксперимента, по которым можно сделать следующие выводы:

- Очистку данных стоит проводить хоть какую-нибудь (лучше всего — полную), так как при её отсутствии были получены худшие метрики.
- Для нормализации токенов лучше всего использовать lemmatization. С ним, в среднем, модели имели больший score.
- В качестве векторизации можно использовать Count Vectorizer, TF-IDF и GloVe (чем больше словарь/размерность вектора — тем качественнее получаются модели, но и обучение их начинает занимать больше времени). Кроме того, в Count Vectorizer и TF-IDF координаты вектора соответствуют реальным токенам, а в Word2Vec и Glove можно считать расстояние (схожесть) между токенами. Для использования Word2Vec, возможно, нужно было его больше обучать на корпусе данных.
- Лучшей субъективной комбинацией оказалась: полная очистка текста + лемматизация + glove. (По таблице: очистка только чисел + отсутствие нормализации + glove, однако разница между ними не большая, только начиная с третьего знака после запятой).

**ПРЕЗЕНТАЦИЯ
ОКОНЧЕНА**



**СПАСИБО ЗА
ВНИМАНИЕ :T**