Анализ тональности отзывов на фильмы

Задача:

Путем анализа корпуса отзывов определить к какой группе относятся отзывы из тестовой выборки:

0 - негативный

1 - слегка негативный

2 - нейтральный

3 - слегка положительный

4 - положительный

Проект сделан на основе соревнования Kaggle Movie Review Sentiment Analysis(Kernels Only).

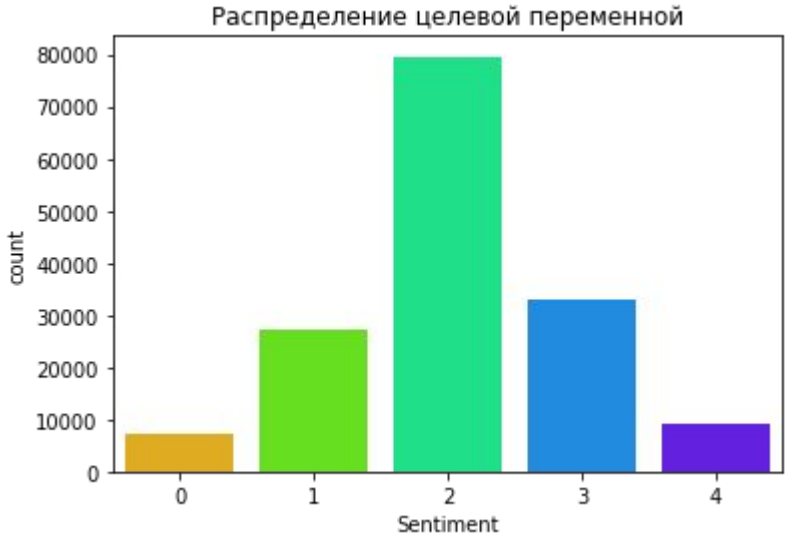
Данные

Данные для соревнования взяты из Rotten Tomatoes movie review dataset.

Каждое предложение было разбито на отдельные фразы с помощью Стэнфордского парсера(Stanford Parser).

Стэнфордский парсер - это синтаксический анализатор естественного языка, это программа, которая определяет грамматическую структуру предложений, например, какие группы слов идут вместе (как «фразы») и какие слова являются предметом или объектом глагола.

EDA



	PhraseId	SentenceId	Phrase	Sentiment
0	1	1	A series of escapades demonstrating the adage ...	1
1	2	1	A series of escapades demonstrating the adage ...	2
2	3	1	A series	2
3	4	1	A	2
4	5	1	series	2
5	6	1	of escapades demonstrating the adage that what...	2
6	7	1	of	2
7	8	1	escapades demonstrating the adage that what is...	2
8	9	1	escapades	2
9	10	1	demonstrating the adage that what is good for ...	2

Препроцессинг данных

1. lowercase
2. Лемматизация с помощью WordNetLemmatizer()
3. Удаление стоп-слов

```
1 print('До препроцессинга:\n' + train["Phrase"][0])
2 print('После: \n' + TextPreprocessor()(train["Phrase"][0]))
```

```
До препроцессинга:
A series of escapades demonstrating the adage that what is good for the goose is also good for the gander , so
После:
series escapade demonstrate adage good goose also good gander occasionally amuses none amount much story
```

Алгоритмы

1. Логистическая регрессия
2. DecisionTreeClassifier
3. GradientBoostingClassifier
4. MultinominalNB

Алгоритмы

Наилучшее качество показал метод логистической регрессии, наихудшее классификатор на основе решающего дерева.

	precision	recall	f1-score	support
0	0.55	0.29	0.38	1084
1	0.54	0.39	0.45	4107
2	0.69	0.87	0.77	11877
3	0.58	0.47	0.52	4926
4	0.61	0.34	0.44	1415
accuracy			0.64	23409
macro avg	0.59	0.47	0.51	23409
weighted avg	0.63	0.64	0.62	23409

Логистическая регрессия

	precision	recall	f1-score	support
0	0.33	0.00	0.01	1084
1	0.36	0.07	0.12	4107
2	0.52	0.97	0.68	11877
3	0.43	0.04	0.07	4926
4	0.41	0.07	0.13	1415
accuracy			0.52	23409
macro avg	0.41	0.23	0.20	23409
weighted avg	0.46	0.52	0.39	23409

DecisionTreeClassifier

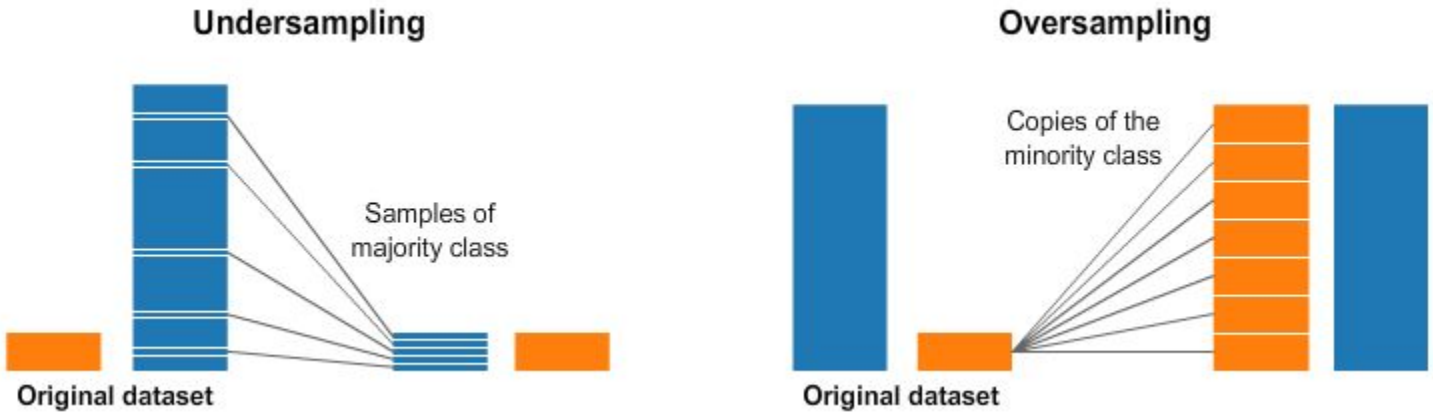
submission (2).csv 0.60965 0.60965

2 days ago by Polina Nikitina

logreg

Результат логистической регрессии на тестовой выборке

Проблема: несбалансированные классы



Источник

Несбалансированные классы: пути решения

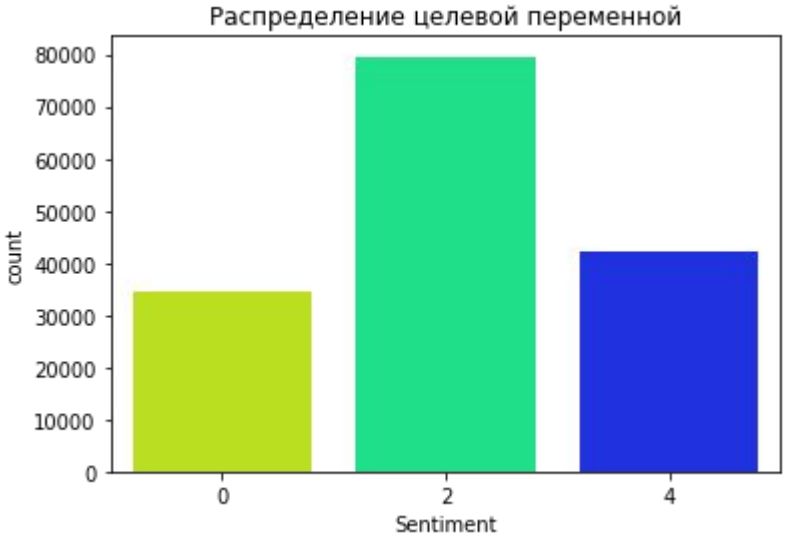
1. Соединить классы (негативные и слегка негативные, положительные и слегка положительные)
2. Найти больше данных
3. Oversampling
4. SMOTE

submission (6).csv a day ago by Polina Nikitina SMOTE	0.49907	0.49907
submission (5).csv a day ago by Polina Nikitina oversampling + logreg	0.59088	0.59088
submission (4).csv 2 days ago by Polina Nikitina add more data	0.60622	0.60622
submission (3).csv 2 days ago by Polina Nikitina 3 lables	0.52902	0.52902

Несбалансированные классы: выводы

Лучше всего себя снова показала логистическая регрессия: при уменьшении количества классов до трех (негативные, нейтральные и положительные фразы), **НО** на тестовой выборке на Kaggle результат проверить невозможно

	precision	recall	f1-score	support
0	0.74	0.56	0.64	5191
2	0.71	0.85	0.77	11877
4	0.78	0.65	0.71	6341
accuracy			0.73	23409
macro avg	0.74	0.69	0.71	23409
weighted avg	0.74	0.73	0.73	23409



Почему же хотя бы без Word2Vec?

```
1 model.most_similar("good")
```

```
/usr/local/lib/python3.7/dist-packages/ipyker
  """Entry point for launching an IPython ker
[('frat', 0.7747133374214172),
 ('skirt', 0.7629051804542542),
 ('reyes', 0.7603274583816528),
 ('earthly', 0.7403990626335144),
 ('sensationalize', 0.7352858781814575),
 ('caliber', 0.7283518314361572),
 ('clam', 0.7224575281143188),
 ('list', 0.7160661220550537),
 ('edit', 0.7105578184127808),
 ('idea', 0.7070690393447876)]
```

```
1 model.most_similar("excellent")
```

```
/usr/local/lib/python3.7/dist-packages/ipyker
  """Entry point for launching an IPython ker
[('receives', 0.9733031988143921),
 ('denzel', 0.9723120927810669),
 ('vanessa', 0.9651637077331543),
 ('washington', 0.9650337100028992),
 ('piccoli', 0.9630882143974304),
 ('redgrave', 0.9627469182014465),
 ('ferrera', 0.9613833427429199),
 ('workshop', 0.9608651995658875),
 ('ontiveros', 0.9590384364128113),
 ('enhances', 0.9577794671058655)]
```

Выводы

1. Методы без глубокого обучения не дают достаточно высокого качества метрик
2. Метод Word2Vec оказался не показателен на данной выборке
3. Балансировка классов путем уменьшения количества классов дала наилучшее качество на валидационной выборке, однако проверить качество на тестовой выборки не представляется возможным
4. **Для дальнейшего улучшения результатов не помешают опыт, насмотренность и чутье, и я на пути к этому!**