XIII Международная молодежная научно-практическая конференция с элементами научной школы «Прикладная математика и фундаментальная информатика»

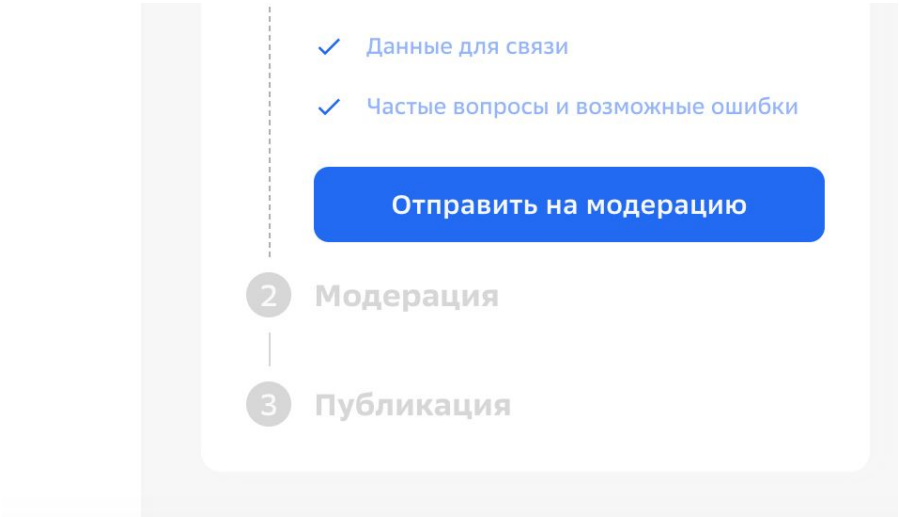
РАЗРАБОТКА ИНСТРУМЕНТАРИЯ ДЛЯ МОДЕРАЦИИ КОММЕНТАРИЕВ НА ОСНОВЕ АНАЛИЗА ТОНАЛЬНОСТИ

Е.И. Рыбакова, И.В. Шарун, старший преподаватель

Актуальность модерации текста на веб-сайтах

Инструментарий может использоваться на образовательных веб-сервисах, где можно фильтровать негативные комментарии и предотвращать их появление на сайте, что в свою очередь улучшает качество контента и безопасность для пользователей.

Однако, задача модерации комментариев является сложной и требует постоянного совершенствования.



Прошлые разработки

Похожая задача решалась в рамках соревнования Toxic Comment Classification Challenge на Kaggle.

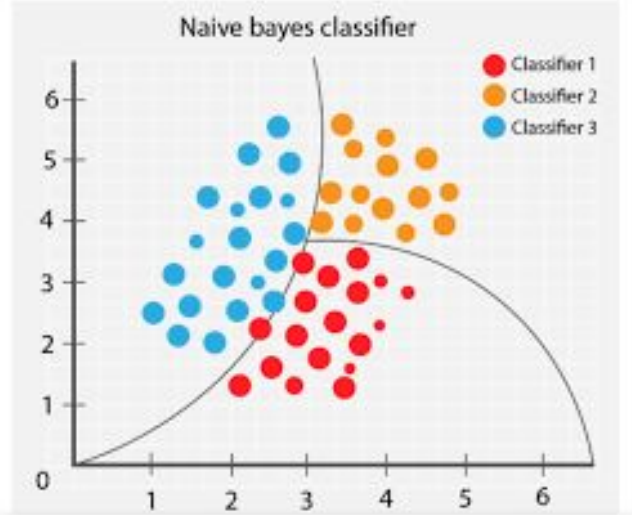
- Toxic Comment Classification - Logistic Regression
- Toxic comment classification - NLP using PyTorch - In this problem we will use DistilBERT model for classification which is lightweight version of BERT transformer
- Pretrained Embedding Keras LSTM Classification
- Toxic Comments Vectorizer Lightning Linear

Для русского языка решение должно получаться аналогичным образом, но качество модели может оказаться ниже из-за того, что русский язык структурно сложнее английского.

Naive Bayes classifier

Алгоритм Наивного Байеса (Naive Bayes) – это метод классификации, основанный на теореме Байеса, который используется для определения вероятности принадлежности объекта к определенному классу на основе его характеристик.

Наивный Байесовский классификатор считается "наивным", потому что он предполагает, что все характеристики объекта независимы друг от друга.

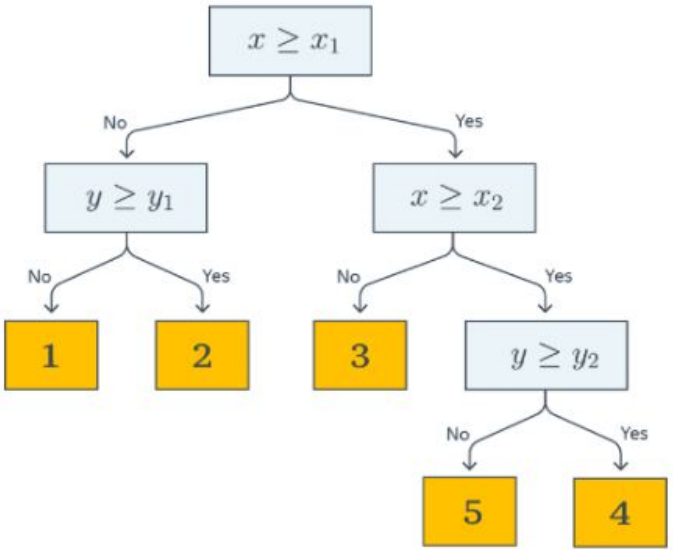


Naive Bayes:

$$P(class|x_1, x_2, \ldots, x_n) = P(x_1|class) \cdot P(x_2|class) \cdot \ldots \cdot P(x_n|class) \cdot P(class)$$

В сравнении с DecisionTreeClassifier

Решающее дерево предсказывает значение целевой переменной с помощью применения последовательности простых решающих правил (которые называются предикатами). Алгоритм представляет собой древовидную структуру, где каждый узел представляет собой тест на значение одного из признаков, а каждая ветвь представляет собой возможный результат этого теста.



Тренировочный набор данных

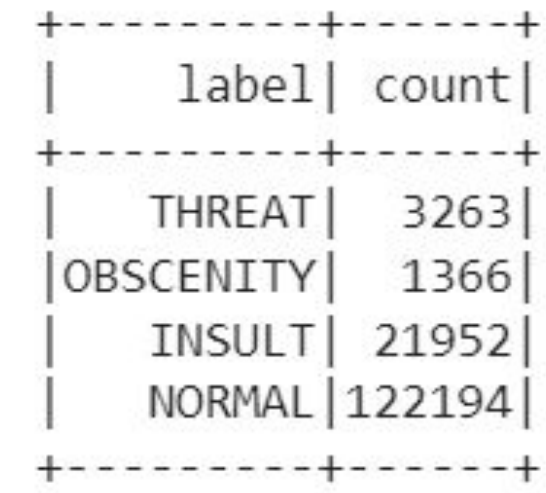
41127	__label__THREAT	дворника надо тоже уничтожить!
6812	__label__NORMAL	моя старшая неделю шипела, не принимала подкидыша, которого я принесла. китя такой славный, потерпите немного, м
6256	__label__NORMAL	полностью с вами согласна!
189636	__label__NORMAL	хоть ногу вверх, ничего не изменится
99053	__label__NORMAL	а что значит - левого ребенка?
98418	__label__NORMAL	вечер добрый! а, что он у вас уже постарел?!
3619	__label__NORMAL	какая порода .?
176463	__label__INSULT	спасатель? просто петух чванливый, взял наших пацанов в заложники и торгуется. позор! как бы сам в щи не попал.
13520	__label__NORMAL	с замечательным юбилеем!!! голос- чудо, заслушаешься 😍 и про чтение все правильно- ничто не сравнится с бумажной
133613	__label__NORMAL	еще бы .такой красавец.
77333	__label__NORMAL	есть вот такое фото культурно-массового мероприятия.
201554	__label__NORMAL	можно бесконечно искать недостатки! мудрак это достойный кандидат для главы города красномакенска.
167393	__label__NORMAL	..... 100% (охрана) + 5 по 100% (всё остальное). итог понятен! полный кирдык!!!
210414	__label__INSULT	бедненький,давайте скинемся,а то сдохнет мразь!
91190	__label__NORMAL	организация противопожарного контроля. но вот денег на неё нет:-(
40268	__label__NORMAL	ещё версии есть или только эта?
79556	__label__NORMAL	я всегда говорила -ума нет вот и бодаются с матерками !!! какие слова -такие и мысли !!! о какие мысли ???? !! глупый ин
193973	__label__NORMAL	гать метлой

Предобработка данных

Далее необходимо структурировать данные, определяя название и тип столбцов: *id*, *label*, *comment*. Очищаем столбец *label* от лишнего, используя регулярное выражение. При структурировании данных важно учитывать их целостность и соответствие заданным требованиям.

Код представлен ниже.

```
from pyspark.sql import functions as F
data = data.withColumn('label',
F.regexp_replace('label', '^__label__', ''))
```



label	count
THREAT	3263
OBSCENITY	1366
INSULT	21952
NORMAL	122194

Предобработка данных

Для улучшения структурированности данных было принято решение объединить метки THREAT, OBSCENITY, INSULT под общим обозначением INSULT. Это позволит упростить процесс анализа комментариев и сократить количество категорий до двух: "оскорбление" и "норма".

label	count
INSULT	26581
NORMAL	122194

Предобработка данных

Для обучения модели данные необходимо представить в виде понятном для модели, а именно в виде числовых векторов. Чтобы правильно провести векторизацию необходимо очистить сообщения от стоп-слов, лишних символов и сокращений, не несущих полезную нагрузку.

```
tokenizer = Tokenizer(inputCol='comment',outputCol='words_token')
stopwordList = nltk.corpus.stopwords.words('russian')
stopwordList.extend([',', '.', '%', '-', '!', '?', ';', ':', '...', '(', ')', '«', '»', '"', '/', '|', '–', '—', '•', ')))', '):'])
stopwords_remover = StopWordsRemover(inputCol='words_token',outputCol='filtered_words', stopWords=stopwordList)
vectorizer = CountVectorizer(inputCol='filtered_words',outputCol='rawFeatures')
idf = IDF(inputCol='rawFeatures',outputCol='vectorizedFeatures')
```

Обучение модели

После разделения на обучающую и тестовую выборки, приступим к построению и обучению модели. Код, показывающий данный этап работы представлен ниже.

```
nb = NaiveBayes(smoothing=1.0, modelType="multinomial", featuresCol='vectorizedFeatures', labelCol='prepared_label')
pipeline = Pipeline(stages=[tokenizer,stopwords_remover,vectorizer,idf,nb])
```

```
dt = DecisionTreeClassifier(labelCol="prepared_label", featuresCol="vectorizedFeatures", predictionCol="dt_prediction")
pipeline = Pipeline(stages=[tokenizer,stopwords_remover,vectorizer,idf,dt])
```

Результат и сравнение

Результаты представлены на рисунке. Первая модель оказалась более эффективной и продуктивной, чем вторая модель, если сравнивать по данным метрикам качеств. Например, такие как доля правильных ответов алгоритма, точность и скорость обработки.

Метрика \ Metric	Значение (округленное до сотых) \Value
Accuracy	0.88
precision	0.84
recall	0.63

Метрика \ Metric	Значение (округленное до сотых) \Value
Accuracy	0.86
precision	0.23
recall	0.99

Заключение

Таким образом, использование инструментария для модерации комментариев является важным шагом для создания безопасной и продуктивной общественной среды на сайтах и платформах. Он помогает бороться с нежелательным контентом, повышает качество взаимодействия между пользователями и улучшает процесс модерации, сокращая время, затрачиваемое на его выполнение.

В качестве дополнительных направлений развития исследования можно рассмотреть следующие:

1)улучшение качества модели путем дополнительной обработки и анализа данных,

2)исследование возможности использования других алгоритмов машинного обучения для модерации комментариев,

3)разработка функционала для автоматической классификации комментариев на основе тематики.

XIII Международная молодежная научно-практическая конференция с элементами научной школы «Прикладная математика и фундаментальная информатика»

Спасибо за внимание!

Е.И. Рыбакова, И.В. Шарун, старший преподаватель