

**Отчёт по практическому заданию №1**

«Классификация комментариев на ревью»

Автор: Туровец Владислав Юрьевич

## 1. Цель работы

Целью практического задания является разработка решения на языке Python для автоматической классификации комментариев к исходному коду на токсичные и нетоксичные.

Решение включает два этапа:

- предобработку текстовых данных;
- обучение и оценку моделей машинного обучения (классических и на основе трансформеров).

## 2. Ход работы

### 2.1. Исходные данные

В качестве исходных данных использовался набор комментариев к исходному коду, размеченных по признаку токсичности. Каждый элемент представляет собой короткий текст, содержащий оценочные, оскорбительные или нейтральные высказывания, а также бинарную метку `is_toxic`, принимающую значение 1 для токсичных комментариев и 0 для нетоксичных.

Данные имеют табличный вид, где первая колонка хранит текст комментария, а вторая — числовую метку. Пример исходных данных приведён в рисунке 1.

	message ↕	is_toxic ∨
1	it doesn't look right	1
2	this looks like cr@p	1
3	what the f*ck are you talking about	1
4	That is a MOFO	1
5	eat \$hit	1

Рисунок 1 – Исходные данные

### 2.2. Предобработка данных

Перед обучением модели тексты были очищены от шумовых и неинформативных элементов. Для этого был создан отдельный Python-модуль, выполняющий загрузку, фильтрацию и нормализацию данных. Очистка включала удаление ссылок, e-mail адресов, HTML-тегов и специальных символов, а также исправление повторов букв и нестандартных апострофов.

Ключевым шагом стало раскрытие разговорных сокращений, например: doesn't → does not, we're → we are, что повысило однородность текстов и снизило количество редких слов. Для нормализации ругательств использовался словарь profane-words.txt, охватывающий как стандартные, так и искажённые формы (с символами @, \$, \*). После обработки выполнялось удаление дубликатов, строк с пропусками и проверка типов данных.

Результат сохранялся в файлах clean\_train.csv и clean\_test.csv, представляющих собой очищенный и унифицированный корпус, готовый к векторизации и обучению моделей машинного обучения.

### 2.3. Классическая модель (TF-IDF + Logistic Regression)

Для первого подхода использовалась классическая схема на основе TF-IDF и логистической регрессии.

Сначала очищенные тексты были преобразованы в числовое представление с помощью TF-IDF-векторизатора, который учитывал униграммы и биграммы и ограничивал размер признакового пространства до двадцати тысяч элементов. Такая конфигурация позволила сохранить устойчивость модели и одновременно отразить сочетания слов, характерные для токсичных комментариев.

После векторизации обучалась модель логистической регрессии. Параметр class\_weight="balanced" обеспечивал равный вес для обеих меток, чтобы модель не склонялась к преобладающему классу. Обучение выполнялось на обучающем наборе, а затем модель оценивалась на тестовых данных.

По результатам тестирования F1-мера составила **0.91**, что соответствует высокому качеству классификации.

По отдельным классам значения метрик были следующими:

для класса «не токсичный» — precision = 0.930, recall = 0.899, f1 = 0.915;

для класса «токсичный» — precision = 0.894, recall = 0.927, f1 = 0.910.

Таким образом, модель одинаково хорошо справляется с обоими типами комментариев, а общее значение accuracy = 0.912 подтверждает стабильность работы.

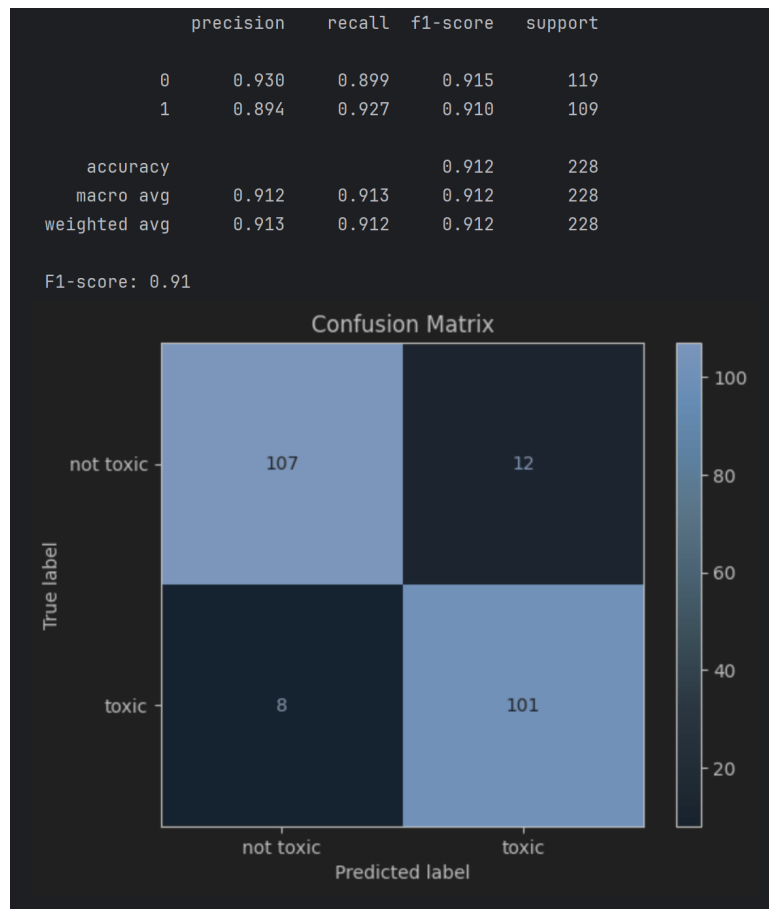


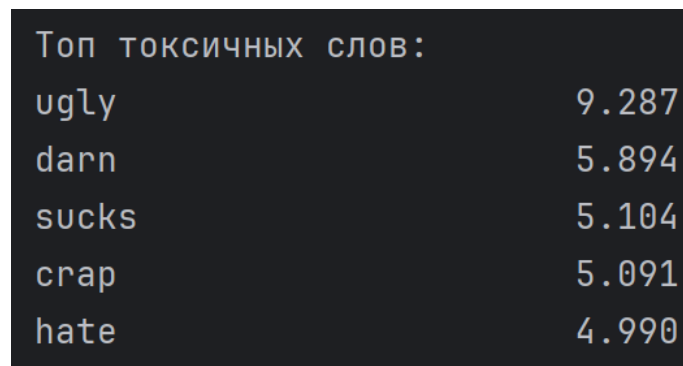
Рисунок 3 – Матрица ошибок и метрики для модели TF-IDF + Logistic Regression

Для проверки интерпретируемости модели были проанализированы веса признаков. Наибольшие положительные коэффициенты соответствуют

словам, увеличивающим вероятность токсичности, а отрицательные — признакам, связанным с нейтральными комментариями.

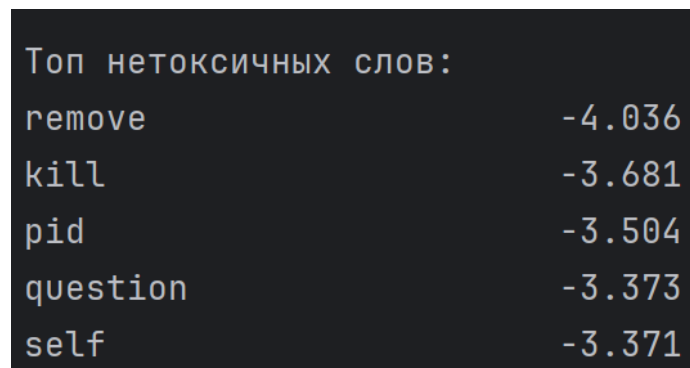
Среди наиболее выраженных токсичных слов оказались ugly, darn, sucks, crap, hate, что подтверждает адекватность модели.

Среди нетоксичных — remove, kill, pid, question, self, что показывает: даже слова с потенциально «агрессивным» звучанием не воспринимаются как оскорбительные вне контекста.



Топ токсичных слов:	
ugly	9.287
darn	5.894
sucks	5.104
crap	5.091
hate	4.990

Рисунок 4 – Топ токсичных слов, выделенных моделью логистической регрессии



Топ нетоксичных слов:	
remove	-4.036
kill	-3.681
pid	-3.504
question	-3.373
self	-3.371

Рисунок 5 – Топ нетоксичных слов, выделенных моделью логистической регрессии

Таким образом, линейный классификатор на основе TF-IDF показал высокую точность и понятную структуру признаков, что делает его надёжной базовой моделью для последующего сравнения.

## 2.4. Модель на основе трансформеров (RoBERTa)

Второй подход основан на контекстной модели RoBERTa, способной учитывать семантику фраз целиком, а не только частоты отдельных слов. Для обучения использовалась предобученная версия roberta-base, на которую была добавлена классификационная голова с двумя выходами для меток токсичности.

Перед обучением все тексты были токенизированы: каждый комментарий разбивался на подслова с помощью токенизатора RoBERTa и дополнялся специальными служебными токенами. Максимальная длина последовательности установлена 128 токенов. Далее данные группировались в батчи по 8 элементов и подавались в модель. Обучение проходило 3 эпохи с использованием оптимизатора AdamW и скорости обучения  $1e-5$ .

Если в каталоге models/roberta-toxic обнаруживались ранее сохранённые веса, модель загружалась повторно, чтобы избежать переобучения. После завершения цикла обучения модель сохранялась и использовалась для инференса на тестовом наборе.

По итогам тестирования F1-мера составила **0.924**, что немного превосходит результат классической модели.

Точность для класса «не токсичный» = 0.955 при полноте 0.899 ( $f1 = 0.926$ ),

а для «токсичного» = 0.897 при полноте 0.954 ( $f1 = 0.924$ ).

Модель показывает более сбалансированные значения по всем метрикам, что свидетельствует о лучшем распознавании контекста.

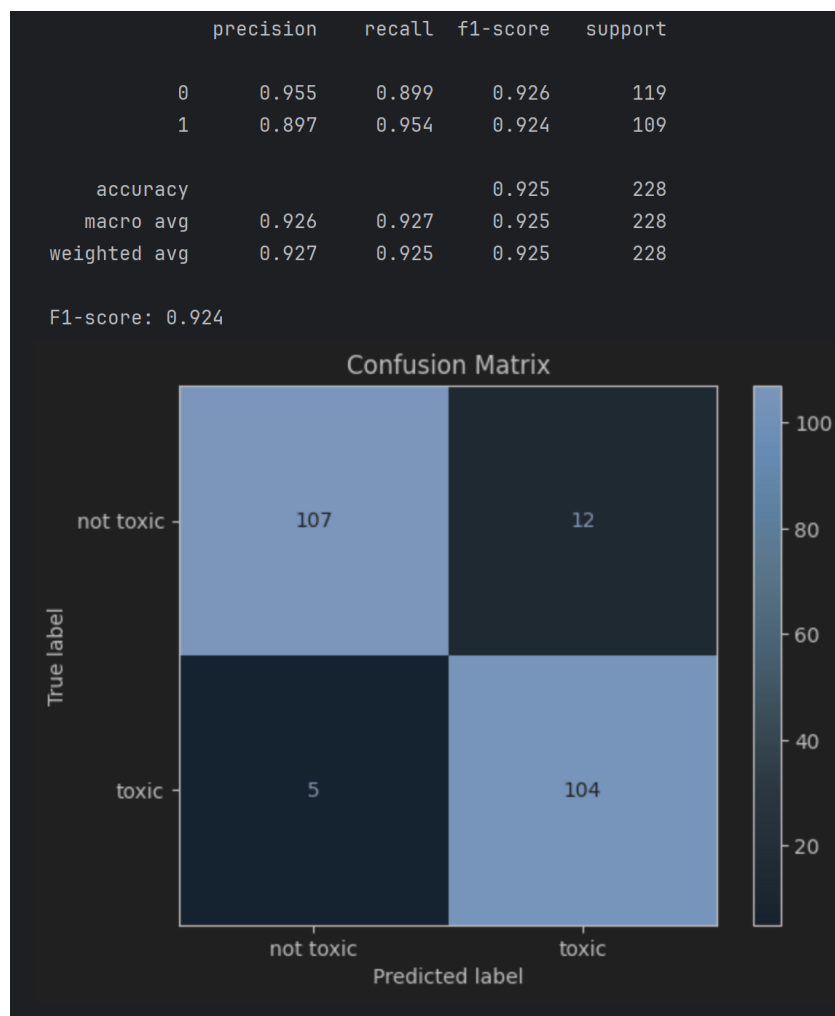


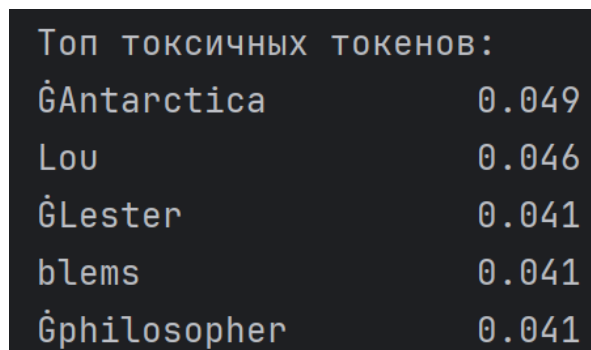
Рисунок 4 – Матрица ошибок и метрики для модели RoBERTa

Для понимания внутренней логики классификатора был выполнен анализ весов выходного слоя. Из них можно извлечь токены, оказывающие наибольшее влияние на решение.

Среди токенов, ассоциированных с токсичностью, встречаются ĠAntarctica, Lou, ĠLester, blems, Ġphilosopher, а среди нетоксичных — Ġyells, ization, gam, Tony, Fax.

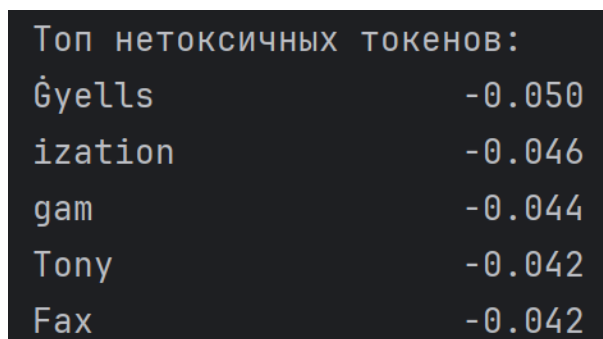
Так как RoBERTa использует контекстную сегментацию, отдельные подслова не всегда несут самостоятельный смысл, поэтому интерпретация таких токенов проводится с осторожностью.





Топ токсичных токенов:	
ĠAntarctica	0.049
Lou	0.046
ĠLester	0.041
blems	0.041
Ġphilosopher	0.041

Рисунок 5 – Топ токсичных токенов, влияющих на решение модели RoBERTa



Топ нетоксичных токенов:	
Ġyells	-0.050
ization	-0.046
gam	-0.044
Tony	-0.042
Fax	-0.042

Рисунок 6 – Топ нетоксичных токенов, влияющих на решение модели RoBERTa

В целом дообученная модель RoBERTa показала чуть более высокие показатели качества по сравнению с TF-IDF-регрессией и лучше справилась с контекстными выражениями, где смысл токсичности определяется не отдельным словом, а сочетанием фраз.

### 3. Результаты работы

В ходе работы были реализованы и сравнены два подхода к классификации токсичных комментариев: классическая модель TF-IDF + логистическая регрессия и нейросетевая модель RoBERTa. Обе показали высокое качество и устойчивость на тестовых данных.

Классическая модель достигла F1-меры 0.91, корректно различая токсичные и нейтральные тексты. Её поведение интерпретируемо, а ошибки возникают в пограничных случаях. Модель RoBERTa показала немного лучший результат —  $F1 = 0.924$ , точнее обрабатывая контекст и сложные выражения, но при этом требует больше вычислительных ресурсов и менее прозрачна в интерпретации.

Таким образом, обе модели справились с задачей: линейная — проще и объяснима, трансформер — точнее и глубже учитывает смысл текста. Итоговые метрики существенно превышают целевой уровень, что подтверждает корректность реализации и выбранных методов.