1. **Ссылка на репозиторий (уровень 2):** [ap3814887/movies\_reviews\_project\_lvl2](https://github.com/ap3814887/movies_reviews_project_lvl2/tree/main)
2. **Описание выбранной NLP-модели:**

* **Название модели**: blanchefort/rubert-base-cased-sentiment
* **Ссылка на модель**: [Бланшфор/Руберт-база-падеж-сентиментальность · Обнимающееся лицо](https://huggingface.co/blanchefort/rubert-base-cased-sentiment)
* **Архитектура**: BERT-based (конкретно — RuBERT, адаптация BERT для русского языка)
* **Тип**: модель классификации тональности
* **Обучающий корпус**: OpenSubtitles, комментарии с сайтов, Twitter, и другие русскоязычные тексты (специализированная разметка для задачи Sentiment Analysis)
* **Классы**:
  + positive
  + neutral
  + negative

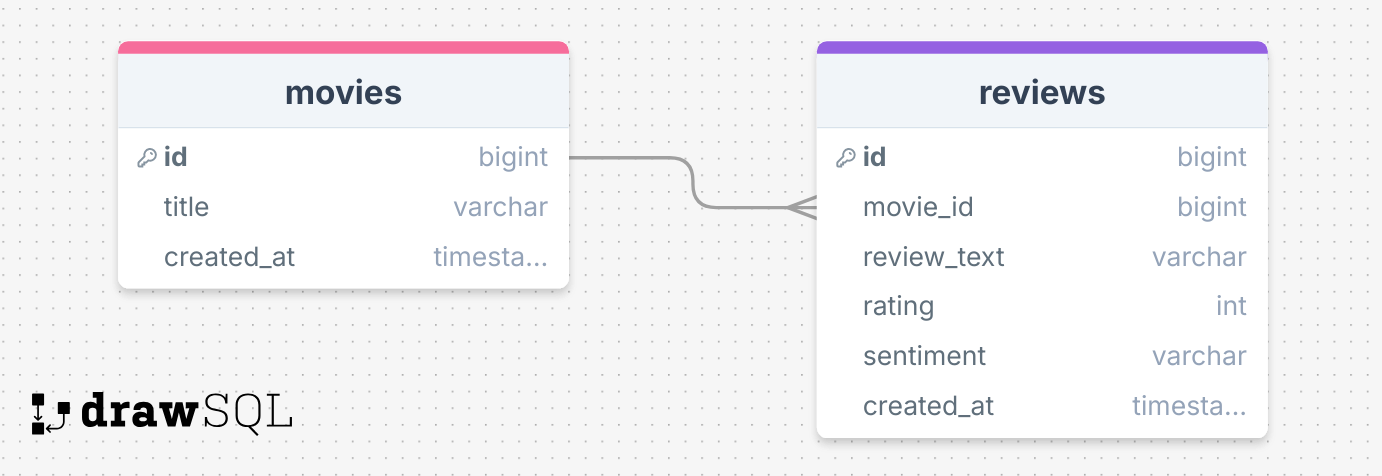
1. **Обоснование выбора модели:**

Модель blanchefort/rubert-base-cased-sentiment была выбрана по следующим причинам:

* Она построена на основе **RuBERT**, который специально адаптирован под русский язык.
* Предобучена на **реальных русскоязычных текстах**, включая неформальные комментарии.
* Поддерживает **многоклассовую классификацию** (3 класса), что идеально подходит для тонального анализа отзывов.
* Проверена в сообществе: более 100K+ загрузок, активно используется.
* Простота интеграции с библиотекой **Transformers**.

Это одна из лучших открытых моделей для анализа тональности на русском языке, особенно если нужен баланс между качеством, производительностью и лёгкостью использования.

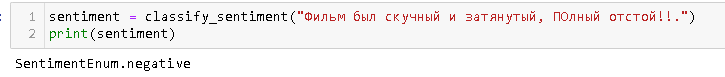
1. **Обновленная ER-диаграмма (в таблицу reviews добавлен столбец sentiment):**



1. **Пример запроса к модели (код):**

Само подключение модели и функция (classify\_sentiment(text: str) -> str:), которая принимает текст и возвращает строковое название предсказанного класса прописаны в скрипте sentiment.py.

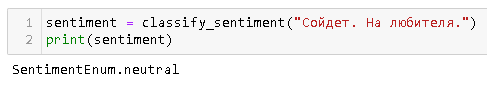
Пример запроса к модели:

1) 

2)



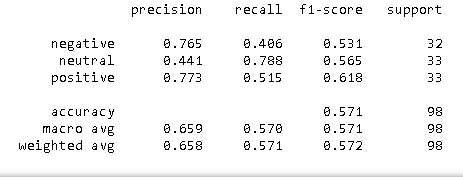
3)



1. **Оценка точности (accuracy, F1-score):**

Для оценки точности использовался искусственно созданный размеченный датасет на 98 различных отзывов.

В результате были получены следующие результаты:



* **Support** — количество примеров каждого класса (всего 98 отзывов).
* **Precision (точность)** — из всех отзывов, которые модель предсказала как этот класс, сколько оказалось правильных.
* **Recall (полнота)** — из всех реальных отзывов этого класса, сколько модель смогла правильно определить.
* **F1-score** — гармоническое среднее точности и полноты (сводный показатель качества).

**Анализ по классам:**

**negative (отрицательные отзывы):**

Precision 0.765 — модель, когда предсказывает негатив, обычно права.

Recall 0.406 — но модель пропускает много настоящих негативных, то есть часто не распознает отрицательные отзывы.

**Итог**: модель "осторожна" с классом negative — мало ошибок в предсказании, но много упущенных случаев.

**neutral (нейтральные отзывы):**

Precision 0.441 — модель часто ошибается, когда предсказывает neutral, то есть много ложных срабатываний.

Recall 0.788 — но при этом большинство реальных нейтральных отзывов она находит.

**Итог**: модель склонна "переклассифицировать" другие отзывы в neutral, что снижает точность.

**positive (положительные отзывы):**

Precision 0.773 — предсказания положительных довольно точны.

Recall 0.515 — половину реальных положительных отзывов модель находит.

**Итог**: модель лучше узнаёт positive, чем negative, но тоже много пропусков.

**Ощая точность:**

**Accuracy = 0.571 (57.1%)** — модель правильно предсказала около 57% всех отзывов.

**Вывод:**

Модель неплохо умеет выделять положительные и отрицательные отзывы, но много пропускает (низкий recall).

Нейтральные отзывы распознаёт лучше по полноте, но при этом много ошибается в них (низкая точность).

1. **Примеры запросов к API (в curl через http://127.0.0.1:8000/docs)**

POST(1):

**curl -X 'POST' \**

**'http://127.0.0.1:8000/reviews' \**

**-H 'accept: application/json' \**

**-H 'Content-Type: application/json' \**

**-d '{**

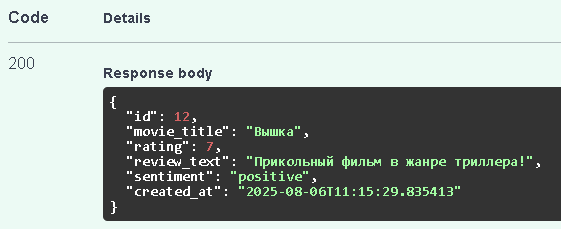
**"movie\_title": "Вышка",**

**"review\_text": "Прикольный фильм в жанре триллера!",**

**"rating": 7**

**}'**

Результат классификации:



POST(2):

**curl -X 'POST' \**

**'http://127.0.0.1:8000/reviews' \**

**-H 'accept: application/json' \**

**-H 'Content-Type: application/json' \**

**-d '{**

**"movie\_title": "Посейдон",**

**"review\_text": "Фильм вообще не понравился! Это какой-то ужас.. Не рекомендую!!",**

**"rating": 7**

**}'**

Результат классификации:

