

Документация по программному обеспечению для приложения и API
машинного обучения. Вариант 2 –

Приложение, предсказывающее категорию новостей по тексту новости

Таблица, содержащая структуру приложения (Streamlit, интерфейс)
представлена на таблице 1.

Класс / Модуль	Метод / Функция	Назначение
Осново й скрипт Streaml it	check_api_health()	Проверяет доступность API сервера по эндпойнту /health. Кэши рует результат на 5 минут.
	st.set_page_config()	Настройка заголовка страницы, иконки и расположения элементов интерфейса.
	st.text_area()	Поле для ввода описания фильма для кластеризации.
	st.button("🔍 Предсказать кластер")	Обработчик кнопки запуска предсказания кластера по введенному описанию.
	Встроенные блоки st.spinner(), st.success(), st.error(), st.war ning()	Отображение статуса обработки, ошибок и предупреждений для пользователя
	requests.post()	Отправляет POST-запрос к API для получения прогноза кластера.
	st.expander()	Раскрывающиеся блоки с дополнительной

		информацией о прогнозе и примерами описаний.
	st.sidebar	Боковая панель с кнопкой проверки состояния API и отображением информации о сервере.

Таблица 1 – Структура веб-приложения

Структура Flask API (бэкенд) представлена на таблице 2.

Класс / Модуль	Метод / Эндпоинт	Назначение
PredictResource (Flask-RESTful Resource)	post() (POST /pred)	Принимает JSON с полем description, выполняет кластерное прогнозирование с помощью загруженных моделей.
HealthResource (Flask-RESTful Resource)	get() (GET /health)	Возвращает статус загрузки моделей и векторизатора, проверка работоспособности API.
Вспомогательные функции	load_model(model_path)	Загружает модель из файла с помощью joblib.
	safe_predict(model, vectorizer, text)	Выполняет безопасное прогнозирование кластера, возвращает кластер, уверенность, вероятности и ошибки.

Таблица 2 – Структура API

Описание параметров и логики:

`data_path` — путь к CSV с токенизированными текстами и названиями кластеров.

`cluster_names` — словарь соответствия токенизированного текста и названия кластера.

`models_dir` — директория с сохранёнными моделями и векторизатором.

Загружаются несколько моделей классификации (SGD, LinearSVC, LogisticRegression).

В методе POST /pred:

Проверяется валидность входных данных.

Текст нормализуется (нижний регистр, обрезка).

Для каждой модели вызывается `safe_predict`.

Выбирается предсказание с максимальной уверенностью.

Формируется ответ с кластером, названием, моделью, уверенностью и вероятностями.

Если уверенность низкая (<0.5), добавляется предупреждение.

В методе GET /health возвращается статус загрузки моделей и векторизатора.

Подробное описание ключевых методов и параметров представлено в таблице 3.

Метод / Функция	Параметры	Возвращаемое значение / Описание
<code>check_api_health()</code>	Нет	bool — True, если API отвечает статусом 200, иначе False.
<code>requests.post(url, json, timeout)</code>	<code>url</code> (str), <code>json</code> (dict), <code>timeout</code> (int)	Отправляет POST-запрос с JSON-данными, возвращает объект ответа.
<code>load_model(model_path)</code>	<code>model_path</code> (str)	Загружает и возвращает модель из файла, либо None при ошибке.

safe_predict(model, vectorizer, text)	model (sklearn модель), vectorizer (объект), text (str)	Ставит предсказание, возвращает словарь с ключами: cluster (int), confidence (float), probabilities (dict), error (str или None).
PredictResult()	JSON с ключом description (str)	JSON с результатом предсказания: cluster, cluster_name, description, model_used, confidence, probabilities, warning (опционально), error (опционально).
HealthResult()	Нет	JSON с состоянием: status ("OK" или "ERROR"), models_loaded (список), vectorizer_loaded (bool).

Таблица 3 – Подробное описание

Пример структуры ответа API /pred\ представлен на рисунке 1.

```
{
  "cluster": 3,
  "cluster_name": "Action Movies",
  "description": "Описание фильма...",
  "model_used": "LogisticRegression_model",
  "confidence": 0.85,
  "probabilities": {
    "Cluster 0": 0.05,
    "Cluster 1": 0.10,
    "Cluster 3": 0.85
  },
  "warning": "Low confidence prediction" // опционально
}
```

Рисунок 1 – структура ответа