Министерство образования и науки Республики Башкортостан

Государственное автономное профессиональное образовательное учреждение

Уфимский колледж статистики, информатики и вычислительной техники

|  |  |
| --- | --- |
|  | УТВЕРЖДАЮ  Заместитель директора  по учебной работе  З.З. Курмашева  « » 2025г. |

"Приложение для определения языка"

Программная документация и руководство пользователя

|  |  |
| --- | --- |
|  | Руководитель проекта  Р.Ф.Каримова  «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2025 г. |
|  | Студенты гр. 22П-1  К.И.Ахкамова,  \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Я.А.Петрова  «\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2025 г. |

Уфа, 2025

1. **Введение**

### ****1.1. Назначение системы****

Веб-приложение с API для автоматического определения языка введенного текста с использованием модели машинного обучения (LogisticRegression).

**Функционал**:

* Определение языка текста
* Просмотр статистики обучающего датасета
* Интерактивная визуализация результатов

**Технологии**:

* **Backend**: FastAPI (Python)
* **Frontend**: Streamlit (Python)
* **ML**: Scikit-learn (Random Forest + TF-IDF)
* **Дополнительно**: NLTK, Pymorphy3 (для предобработки текста)

1. **Руководство пользователя**

**2.1. Запуск приложения**

**Требования**

* Python 3.8+
* Установленные зависимости (pip install -r requirements.txt)

**Запуск сервера (API)**

Uvicorn main:app --reload

После запуска API будет доступен по адресу:  
🔗 [**http://localhost:8000**](http://localhost:8000/)

**Запуск клиента (Streamlit)**

streamlit run streamlit\_language.py

Приложение откроется в браузере:  
🌐 [**http://localhost:8501**](http://localhost:8501/)

### ****2.2. Использование приложения****

#### ****Вкладка "Определение языка"****

1. Введите текст в поле.
2. Нажмите кнопку **"Определить язык"**.
3. Результат отобразится в виде:
   * Название языка
   * Визуализация вероятностей (топ-5 языков)

#### ****Вкладка "Статистика датасета"****

* Общее количество текстов
* Средняя длина текста
* Распределение языков (график)

#### ****Вкладка "Справка"****

* Инструкция по использованию
* Примеры API-запросов

1. **API Документация**

## **Методы**

## POST (/predict ) – Определение языка

## GET (/status) – Статистика датасета

## GET (/help) – Справка API

## 

### ****3.1. Таблица функций / методов****

|  |  |
| --- | --- |
| Серверная часть | |
| Функции / методы | Описание |
| app = FastAPI() | Создает экземпляр FastAPI приложения |
| if not os.path.exists('lr\_model.pkl') or not os.path.exists('vectorizer.pkl'):# Обучение новой моделиelse:# Загрузка существующей модели | **Инициализация модели и векторизатора**Проверяет наличие сохраненных моделей, при необходимости обучает новые |
| vectorizer = TfidfVectorizer(\*\*VECTORIZER\_PARAMS)model = LogisticRegression(...)model.fit(X, df['language'])joblib.dump()... | **Обучение модели** Создает TF-IDF векторизатор и логистическую регрессию, обучает и сохраняет модель |
| class TextInput(BaseModel) | Модель для валидации входных данных (содержит поле text) |
| @app.post("/predict")async def predict\_language(item: TextInput):# Предсказание языка | **API Endpoint** Принимает текст, возвращает предсказанный язык и вероятности |
| vectorizer.transform() | Преобразует текст в TF-IDF вектор |
| model.predict() | Предсказывает язык текста |
| model.predict\_proba() | Возвращает вероятности для каждого класса |
| joblib.load()/dump() | Сохраняет/загружает модель |
| logger.info()/error() | Логирование информации и ошибок |

|  |  |
| --- | --- |
| Интерфейс приложения | |
| Функции / методы | Описание |
| st.set\_page\_config() | Настраивает базовые параметры страницы (заголовок, иконку, макет) |
| st.markdown("<style>...</style>", unsafe\_allow\_html=True) | Применение кастомных CSS-стилей для всех компонентов интерфейса |
| def load():df = pd.read\_csv('language.csv')# Обработка данных...return df, stats | Загружает датасет из CSV, вычисляет статистику (количество текстов, среднюю длину) и возвращает DataFrame с данными |
| response = requests.post(f"{API\_URL}/predict", json={"text": text\_input}) | Отправляет текст на FastAPI-сервер для предсказания языка и получает JSON-ответ |
| Вкладка 1 **Определение языка**text\_input = st.text\_area(...)st.button("Определить язык") | Поле для ввода текста и кнопка запуска предсказания |
| st.success(f"\*\*Язык:\*\* {result['predicted\_language']}")st.dataframe(prob\_df)st.pyplot(fig) | Отображает результаты:   * + Определенный язык   + Таблицу вероятностей   + График распределения вероятностей |
| **Вкладка 2: Статистика**st.metric() # Для показателейst.dataframe() # Для таблицыst.pyplot() # Для диаграмм sns.barplot()  ax.pie() | Визуализирует статистику датасета:   * + Круговая и столбчатая диаграммы   + Топ-5 языков   + Основные метрики   Круговую диаграмму (распределение языков)  Столбчатую диаграмму (количество текстов) |
| **Вкладка 3: Справка**st.expander() | Содержит инструкции, техническую информацию и описание датасета |