**README**

**Нейросеть для автоматического анализа резюме**

**Описание проекта**

Проект направлен на создание нейросети, которая анализирует резюме, выявляет типичные ошибки и предоставляет персонализированные рекомендации для их улучшения. Это решение помогает соискателям создать конкурентоспособное резюме, а HR-специалистам ускоряет процесс отбора кандидатов.

**Бизнес-юнит:**

**VK Education**

**Цель:**

Автоматизировать процесс анализа резюме, выявлять ошибки (грамматические, структурные, содержательные) и предлагать улучшения на основе требований работодателей.

**Состав команды и зоны ответственности**

* **Портной Жанна** отвечает за сбор и обработку данных с сайтов по вакансиям. Она разработала парсер для автоматического сбора информации с hh.ru, настроила фильтрацию и сохранение данных в JSON-формате, а также создала систему для расширения набора данных. Реализовала алгоритмы для анализа текста резюме, включая проверку грамматических и орфографических ошибок, поиск ключевых слов и их соответствие требованиям вакансий.
* **Пономарев Дмитрий Валентинович** отвечает за создание пользовательского интерфейса. Разработал прототип приложения на основе Streamlit, реализовал возможность загрузки файлов резюме, их обработки и отображения результатов анализа, а также добавил функции для скачивания рекомендаций.
* **Бирюков Владислав Михайлович** проводит интеграцию модулей и тестирование системы. Отвечает за улучшение взаимодействия между модулями и подготовку документации.

**Текущее состояние проекта**

**Реализованные модули**

1. **Парсер вакансий с HH.ru (parser\_regions.ipynb)**  
   Сбор данных по вакансиям из городов-миллионников России. Данные сохраняются в формате JSON и используются для обучения модели и проверки ключевых навыков в резюме.
2. **Модуль анализа резюме (Xakaton\_III.ipynb)**
   * Проверяет орфографию и грамматику с использованием библиотеки language\_tool\_python.
   * Анализирует текст резюме, выявляет пропуски в хронологии и отсутствие ключевых навыков.
   * Оценивает соответствие резюме требованиям и генерирует исправленный текст.
3. **Прототип интерфейса (app.py)**  
   Этот файл представляет собой прототип приложения для анализа резюме.  
   Основные функции:
   * Загрузка резюме в форматах PDF и DOCX.
   * Извлечение текста из файлов.
   * Классификация резюме на категории: "студент", "профессионал" или "общее".
   * Генерация персонализированных рекомендаций на основе категории:
     + Для студентов — акцент на учебных проектах и навыках.
     + Для профессионалов — улучшение структуры и достижений.
     + Для общего типа — устранение общих ошибок и улучшение структуры.
   * Возможность скачивания рекомендаций в формате PDF и DOCX.
   * Прогресс-бар для отслеживания обработки файла.
4. **Файл зависимостей (requirements.txt)**  
   Содержит все необходимые библиотеки:
   * streamlit для интерфейса.
   * PyPDF2 и python-docx для работы с текстом.
   * reportlab для генерации PDF-файлов.

**Используемые технологии**

* **Языки**: Python.
* **Библиотеки**: Streamlit, PyPDF2, python-docx, language\_tool\_python, reportlab.
* **Инструменты для сбора данных**: API hh.ru.

**Как запустить проект**

1. **Клонируйте репозиторий**:
2. git clone <ссылка на репозиторий>
3. cd <название\_папки\_репозитория>
4. **Установите зависимости**:
5. pip install -r requirements.txt
6. **Запустите приложение Streamlit**:
7. streamlit run app.py
8. **Загрузите резюме и получите персонализированные рекомендации.**

**Следующие шаги**

1. **Интеграция модели анализа с интерфейсом**: подключение аналитической модели для обработки данных в реальном времени.
2. **Оптимизация модели**: повышение точности классификации и рекомендаций.
3. **Улучшение пользовательского интерфейса**: добавление интерактивных настроек, улучшение визуализации и функциональности.
4. **Тестирование на большем наборе данных**: проверка системы на различных типах резюме и расширение сценариев использования.
5. **Добавление новых функций**: поддержка дополнительных форматов файлов и внедрение расширенной аналитики.