Федеральное государственное образовательное бюджетное учреждение

высшего профессионального образования

**Финансовый университет при Правительстве Российской Федерации**

**(Финансовый университет)**

**Департамент анализа данных и машинного обучения**

**Факультета информационных технологий и анализа больших данных**

**Лабораторная работа №3**

**по дисциплине «Управление качеством программных систем»**

Направление подготовки - 09.03.09 «Прикладная информатика»

на тему: Создание тестовой документации

**Подготовил**

студент группы ЗБ-ПИ20-2

Миловидов Владислав Игоревич

(Ф.И.О.)

**Проверил**:

ассистент ДАДиМО ФИТиАБД,

Клочков Евгений Юрьевич

(Ф.И.О.)

Москва 2024

# **Цель работы:**

Целью данной лабораторной работы является создание тестовой документации для выбранного приложения. К тестовой документации относятся: план тестирования (тест-план), набор тестов (тест-сьют), тестовый случай (тест-кейс), отчет о дефектах (баг-репорт).

# **Описание программного проекта:**

Программный проект представляет собой приложение для демонстрации применения машинного обучения для категоризации текстовых данных, написанный на языке Python. Используемые библиотеки включают NumPy для работы с массивами, Pandas для обработки данных, Scikit-learn для машинного обучения, NLTK для задач обработки естественного языка, а также Matplotlib и Seaborn для визуализации данных. Программное обеспечение проводит сбор и подготовку данных, процесс обработки текста (включая лемматизацию и стемминг), обучение моделей, оценку их эффективности и визуализацию результатов.

Частью визуализации является сравнение производительных метрик, что обеспечивает наглядное представление итоговых результатов моделей и помогает оптимизировать процесс обработки текста сквозь категоризацию.

**Тестовая документация:**

**1. Тест-план:**

1.1 Эффективность алгоритма классификации

* **Цель**: проверить точность алгоритма категоризации текстовых данных.
* **Методы тестирования**: функциональное тестирование, тестирование на базе реальных примеров текстов, тестирование регрессии.
* **Метрики**: точность классификации, количество ложных срабатываний.
* **Ожидаемый результат**: ложных срабатываний минимум, высокая точность в категоризации различных типов текстов.

1.2 Производительность системы

* **Цель**: проверить скорость обработки текста и классификации.
* **Методы тестирования**: тестирование производительности, стресс-тестирование.
* **Метрики**: время обработки, время ответа модели, количество обработанных текстов за фиксированный период времени.
* **Ожидаемый результат**: модель быстро и эффективно обрабатывает большие объемы текста.

1.3 Практичность использования

* **Цель**: проверить удобство и понятность интерфейса, удобство применения системы.
* **Методы тестирования**: использование системы типовыми пользователями, тестирование пользовательского опыта.
* **Метрики**: обратная связь от пользователей, время на изучение системы и начало работы с ней.
* **Ожидаемый результат**: система удобна и понятна для пользователей без специальной подготовки.

1.4 Интеграция и безопасность системы

* **Цель**: проверить защиту данных, корректность интеграции с другими системами и сервисами
* **Методы тестирования**: тестирование на интеграцию, тестирование на безопасность.
* **Метрики**: успешность интеграции, отсутствие утечек данных.
* **Ожидаемый результат**: система успешно интегрируется с другими сервисами, безопасность данных на высоком уровне.

**2. Тест-сюьт**

2.1 **Проверка функциональности**

2.1.1 Проверка работы алгоритма классификации

* **Цель**: убедиться в том, что алгоритм классификации текста работает корректно.
* **Предпосылки**: есть текст для тестирования.
* **Действия**: проанализировать заданный текст через систему.
* **Ожидаемый результат**: текст должен быть классифицирован корректно.

2.1.2 Проверка предобработки текста

* **Цель**: убедиться в корректности предобработки текста.
* **Предпосылки**: есть текст с ошибками и сложными структурами.
* **Действия**: запустить текст через систему предобработки.
* **Ожидаемый результат**: текст быстро и корректно обрабатывается, ошибки исправлены.

2.2 **Проверка производительности**

2.2.1 Обработка больших объемов данных

* **Цель**: убедиться, что система может обрабатывать большие объемы данных.
* **Предпосылки**: есть большой набор текстовых данных.
* **Действия**: запустить обработку больших объемов текстовых данных.
* **Ожидаемый результат**: система обрабатывает данные без сбоев и задержек.

2.2.2 Устойчивость системы

* **Цель**: убедиться, что система работает стабильно под условием высокой нагрузки.
* **Предпосылки**: есть большой объем текстовых данных для обработки.
* **Действия**: запустить систему под высокой нагрузкой.
* **Ожидаемый результат**: система работает стабильно без сбоев.

2.3 **Проверка пользовательского опыта**

2.3.1 Интуитивность интерфейса

* **Цель**: проверить, насколько интуитивно понятен интерфейс.
* **Предпосылки**: система готова к использованию.
* **Действия**: взаимодействовать с интерфейсом системы.
* **Ожидаемый результат**: интерфейс легко понимается и интуитивно воспринимается пользователями.

3. **Тест-кейсы**

3.1 **Позитивные тест-кейсы:**

3.1.2 Верное распознавание и категоризация текста

* **Предпосылки**: на входе имеется текст известной категории.
* **Действия**: ввести текст в систему.
* **Ожидаемый результат**: система должна верно категоризировать введенный текст, соответствуя его исходной категории.

3.1.2 Быстрое время ответа системы

* **Предпосылки**: подготовлен текст для обработки.
* **Действия**: скормить текст системе и засечь время ответа.
* **Ожидаемый результат**: система обрабатывает и категоризирует текст в пределах определенного времени без значительных задержек.

3.2 **Отрицательные тест-кейсы**

3.2.1 Ввод незначительного количества данных для обработки

* **Предпосылки**: имеется очень малый объем текста для обработки.
* **Действия**: ввод небольшого текста в систему.
* **Ожидаемый результат**: система выдает ошибку об недостаточности данных для анализа и категоризации.

3.2.2 Ввод текста на неизвестном системе языке

* **Предпосылки**: имеется текст на языке, который система не способна обрабатывать.
* **Действия**: Ввод текста на неизвестном языке.
* **Ожидаемый результат**: Система выдает ошибку об неизвестном языке и невозможности категоризировать текст.

4. **Баг-репорт**

4.1 Неработоспособность алгоритма предварительной обработки текста

* **Описание**: алгоритм предварительной обработки текста не справляется с комплексными структурами текста и приводит к ошибкам в результате обработки.
* **Приоритет**: высокий
* **Шаги для воспроизведения**:
  + Введите комплексные структуры текста в систему.
  + Пронаблюдайте за процессом обработки.
* **Результат**: система выдает неправильные результаты после обработки.
* **Ожидаемый результат**: система должна правильно обрабатывать все структуры текста и выдавать правильные результаты.

4.2 Превышение времени ответа системы

* **Описание**: система не справляется с большими объемами данных и время ответа значительно превышает установленные рамки.
* **Приоритет**: средний
* **Шаги для воспроизведения**:
  + Загрузите большой объем текстовых данных для обработки.
  + Засеките время ответа системы.
* **Результат**: время ответа системы превышает установленный лимит.
* **Ожидаемый результат**: система должна справляться с большими объемами данных в установленные рамки времени.

# **Выводы по работе:**

Тестирование системы было основано на следующих критериях: эффективность алгоритма классификации, производительность системы, удобство использования и надежность интеграции и безопасности данных. В ходе функционального и производственного тестирования были проверены работа алгоритма классификации, способность системы обрабатывать большие объемы данных, ее устойчивость при высокой нагрузке и возможность оперативного реагирования

Во время тест-кейсов были проведены положительные и отрицательные тесты. Система хорошо классифицировала текст известной категории и обрабатывала данные в заданные сроки. Однако, когда вводился либо слишком малый объем текста для обработки, либо текст на неподдерживаемом языке, система выдавала ошибки.

Также были обнаружены два бага: проблема с предварительной обработкой текста, когда система не справлялась с комплексными структурами текста, и превышение времени ответа системы при работе с большими объемами данных.

Этот проект обеспечил создание критически важных элементов тестовой документации для гарантирования высокого уровня качества приложения. В ходе тестирования было выявлено несколько аспектов для доработки, которые будут приняты во внимание в последующих этапах разработки.

# **Список используемых источников:**

1. Учебное пособие «Основы управления качеством программных средств»
2. А. А. Романов «ТЕСТИРОВАНИЕ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ»

# **Приложения:**

1. Приложение 1 к ЛР2.ipynb