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 №2**

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

Направление подготовки - 09.03.09 «Прикладная информатика»

на тему: Работа с классификацией видов

тестирования.

**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:**

Объектом исследования текущей лабораторной работы выступает классификация разновидностей тестирования. Работа ориентирована на проведение всестороннего анализа и составление доклада по выбранному для изучения программному продукту. В рамках аналитической работы будет осуществлено изучение классификации тестирования, его подтипов и разновидностей, а также подробно представлены методология и уровни тестирования, применяемые в данном проекте.

# **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.

# **Список видов, типов, методов, уровней тестирования, применимых для выбранного проекта:**

1. **Виды тестирования:**
   1. Функциональное тестирование
   2. Нефункциональное тестирование (такое как производительность, безопасность, доступность)
   3. Регрессионное тестирование
   4. Бета-тестирование
2. **Типы тестирования:**
   1. Черный ящик (тестирование без знания внутренней структуры системы)
   2. Белый ящик (тестирование с полным пониманием внутренней структуры системы)
   3. Серый ящик (комбинация черного и белого ящика)
3. **Методы тестирования:**
   1. Ручное тестирование
   2. Автоматизированное тестирование
   3. Тестирование на основе моделей
4. **Уровни тестирования:**
   1. Модульное тестирование
   2. Интеграционное тестирование
   3. Системное тестирование
   4. Тестирование приемки

# **Примеры дефектов, характерных для каждого вида тестирования:**

1. **Функциональное тестирование** - проблемы с функциональностью: при

запуске приложения на устройствах с другой версией Python (отличные от 3.10) наблюдаются сбои в работе (приложение не запускается или непроизвольно прекращает работу), а также ошибки в обучении модели: модель не обучается или выдает неверные результаты при предсказании.

1. **Нефункциональное тестирование**:
   1. Ошибка производительности: приложение работает медленно при обработке больших объемов текста.
   2. Ошибка надежности: приложение падает при обработке некорректных данных.
   3. Ошибка удобства использования: интерфейс сложен для понимания или использования.
2. **Регрессионное тестирование** - проблемы, которые возникли после внесения изменений в код – ломают ранее работавшие функции или алгоритмы.

# **Выводы по работе:**

После выполненного анализа, можно определить основные области, которые будут в приоритете при тестировании данного программного проекта.

Основное внимание стоит придавать функциональной проверке критически важных операций, вроде начальной предобработки текста, а также рассмотрению нефункциональных параметров, таких как производительность и удобство применения.

Осознание разнообразия типов тестирования является существенным этапом в подготовке программного продукта. Каждый метод тестирования способствует идентификации уникальных ошибок и дефектов внутри системы. Учитывая результаты тестирования, можно заключить, что глубоко проведенное, всестороннее анализирование является фундаментом для создания качественного и надежного программного решения. Обнаруженные в ходе тестирования баги и несоответствия способствуют усовершенствованию системы и позволяют избежать связанных с ними трудностей в будущем.

# **Список используемых источников:**

1. Учебное пособие «Основы управления качеством программных средств»
2. В. Н. Пероцкая, Д. А. Градусов «Качество и надёжность программных систем»