ЛАБОРАТОРНАЯ РАБОТА №2

Тема лабораторной работы: работа с классификацией видов тестирования.

Цель работы

Цель данной работы заключается в проведении детального анализа и составлении всестороннего отчета по различным аспектам тестирования программного обеспечения, выбранного в качестве объекта исследования. В рамках этого анализа будет рассмотрена классификация тестирования, определены его виды и типы, а также описаны используемые методы и уровни тестирования, применимые к конкретному проекту. b. Описание программного проекта Проект представляет собой систему машинного обучения для визуализации данных, написанную на языке Python. Используемые библиотеки включают NumPy для работы с массивами, Pandas для обработки данных, Scikit-learn для машинного обучения, Matplotlib и Seaborn для визуализации данных. Основная модель проекта — t-SNE (t-distributed Stochastic Neighbor Embedding), реализованная с помощью библиотеки Scikit-learn. Программа выполняет сбор и подготовку данных, обработку данных (нормализация, масштабирование), обучение моделей, оценку их производительности и визуализацию результатов. Визуализация включает создание различных графиков и диаграмм, что позволяет наглядно представить результаты работы моделей и улучшить процесс анализа данных. c. Классификация тестирования для проекта

Для анализа классификации тестирования используется следующая структура:

1. Виды тестирования:
   * Функциональное тестирование: Проверка основных функций приложения, таких как предобработка данных, обучение модели и визуализация результатов.
   * Нефункциональное тестирование: Проверка производительности, надежности и удобства использования программы.
   * Регрессионное тестирование: Проверка корректности работы программы после внесения изменений в код.
2. Типы тестирования:
   * Модульное тестирование: Тестирование отдельных функций и методов, таких как нормализация данных и построение графиков.
   * Интеграционное тестирование: Тестирование взаимодействия между модулями, например, проверка взаимодействия между этапами предобработки, обучения и визуализации.
   * Системное тестирование: Тестирование всей системы в целом.
3. Методы тестирования:
   * Черный ящик: Тестирование функциональности приложения без знания внутренней структуры кода.
   * Белый ящик: Тестирование внутренних структур и алгоритмов приложения, например, проверка корректности работы алгоритма t-SNE.
   * Серый ящик: Комбинация методов черного и белого ящика для комплексного тестирования.
4. Уровни тестирования:
   * Низкоуровневое тестирование: Модульное тестирование отдельных функций и методов.
   * Среднеуровневое тестирование: Интеграционное тестирование взаимодействия между модулями.
   * Высокоуровневое тестирование: Системное тестирование всего приложения.

d. Примеры дефектов, характерных для каждого вида тестирования

1. Функциональное тестирование:
   * Ошибка в предобработке данных: данные обрабатываются некорректно или частично.
   * Ошибка в визуализации данных: графики и диаграммы не отображают данные корректно или имеют ошибки в построении.
   * Ошибка в обучении модели: модель не обучается или выдает неверные результаты при визуализации.
2. Нефункциональное тестирование:
   * Ошибка производительности: приложение работает медленно при обработке больших объемов данных.
   * Ошибка надежности: приложение падает при обработке некорректных данных.
   * Ошибка удобства использования: интерфейс сложен для понимания или использования.
3. Регрессионное тестирование:
   * Ошибка после изменения кода: внесенные изменения ломают ранее работавшие функции или алгоритмы.

e. Выводы по работе

Проведенный анализ позволяет выделить ключевые направления для тестирования выбранного программного проекта. Особое внимание следует уделить функциональному тестированию основных операций, таких как предобработка данных и визуализация результатов, а также нефункциональным аспектам, включая производительность и удобство использования. Регрессионное тестирование играет важную роль в обеспечении стабильности приложения после внесения изменений в код. Применение различных типов и уровней тестирования позволит достичь более полного охвата тестами и выявить потенциальные дефекты на ранних этапах разработки.

f. Список использованных источников

1. Документация по работе с библиотекой Pandas. URL: <https://pandas.pydata.org/>
2. Документация по работе с библиотекой NumPy. URL: <https://numpy.org/>
3. Документация по работе с библиотекой Matplotlib. URL: https://matplotlib.org/stable/contents.html
4. Документация по работе с библиотекой Seaborn. URL: https://seaborn.pydata.org/
5. Документация по работе с библиотекой Scikit-learn. URL: https://scikit-learn.org/stable/