Федеральное государственное бюджетное образовательное  
 учреждение высшего профессионального образования   
**«Финансовый университет при Правительстве   
Российской Федерации»**

**Лабораторная работа №2**по дисциплине:   
**«Управление качеством программных систем»**

на тему:

**«Работа с классификацией видов тестирования»**

Факультет «Институт онлайн образования»

Направление «Прикладная информатика»

**Выполнил:**   
студент группы ЗБ-ПИ20-2  
Мамедов Р. Х.

**Преподаватель:**

Клочков Е.Ю.

Москва – 2024

**a. Цель работы**

Цель данной работы — провести анализ и составить отчет по видам тестирования, применимым для 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**

В качестве проекта будет выбран также курсовой проект «Автоматизированная самообучающаяся система логистики», представляющая собой методы машинного обучения в целях прогнозирования времени доставки товара.

Проект написан с помощью языке программирования Python.

В том числе его библиотеки:

- pandas (библиотека для работы с данными, фильтрация, сортировка и подготовка данных);

- numpy (в основном работа с массивами);

- seaborn и matplotlib (визуализационные посредники: создание графиков статистики);

- sklearn, keras и tensorflow (мощные инструменты для предиктивного анализа данных и машинного обучения.

Основная модель – RandomForestRegressor – метод случайных лесов.

Визуализация включает сравнение метрик производительности, что позволяет наглядно представить результаты работы моделей и улучшить процесс нормализации текста.

**c. Классификация тестирования для проекта**

Для анализа классификации тестирования используется следующая структура:

* Виды тестирования:
* Функциональное тестирование: проверка основных функций реализованного решения: предсказание времени доставки продуктов питания и сравнение результатов.
* Нефункциональное тестирование: проверка производительности, надежности и удобства использования программы
* Регрессионное тестирование: проверка корректности работы программы после внесения изменений в код.
* Типы тестирования:
* Модульное тестирование: тестирование отдельных функций и методов, таких как представление созависимости некоторых признаков с временем доставки и различными условиями: погода, трафик, состояние транспортного средства, тип заказа и населенного пункта.
* Интеграционное тестирование: тестирование взаимодействия между модулями, например, проверка взаимодействия между этапами предобработки, нормализации и обучения модели.
* Системное тестирование: тестирование всей системы в целом.
* 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 случайных лесов.
* Серый ящик: комбинация методов черного и белого ящика для комплексного тестирования.
* 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 во времени: выходной формат данных не в формате datetime.
* Ошибка представлении визуальных данных: не удалены отсутствующие значения и не проведена EDA, в результате чего возможны «выбросы» и «пустоты» в графиках.
* Ошибка в обучении модели: модель не обучается или выдает неверные результаты при предсказании.

1. Нефункциональное тестирование:

* Ошибка производительности: приложение работает медленно при загрузке больших датафреймов.
* Ошибка надежности: не обработаны исключения.

**e. Выводы по работе**

В ходе анализа были рассмотрены различные виды тестирования, которые можно применять в отношении самообучающейся системы логистики продуктов питания. Каждому виду тестирования присущи свои характеристики и решаемые задачи (в данном случае поиск несоответствий и проблем). Результатом комплексного тестирования могут являться неоспоримые огрехи проекта, требующие незамедлительного вмешательства. Также возможны некритичные дефекты, однако исправление которых выведет проект на новый уровень качества программного продукта.

**f. Список использованных источников**

1. Баханов, А.Г. Роль больших данных и имитационного моделирования [Текст] // Социологический альманах. – 2017. – №8. – С. 504-510
2. Голицына, О. Л. Базы данных / О.Л. Голицына, Н.В. Максимов, И.И. Попов. - М.: Форум, Инфра-М, 2021. - 303 c.
3. Дай, З. Двухэтапный метод решения многоуровневых задач определения местоположения и маршрутизации в цепочках поставок [Электронный ресурс] / З. Дай, Ф. Аклан, К. Гао, Ю. Джоу. – 2019. С. 618-634. – Режим доступа: <https://doi.org/10.1016/j.eswa.2018.06.050>
4. Плоткин, Б. К. Экономико-математические методы и модели в коммерческой деятельности и логистике / Плоткин Б.К., Делюкин Л.А. — М.:ИЦ РИОР, НИЦ ИНФРА-М, 2018. — 346 с.