Федеральное государственное образовательное бюджетное

учреждение высшего образования

**«Финансовый университет при Правительстве Российской Федерации»**

**(Финансовый университет)**

**Департамент анализа данных и машинного обучения**

**Отчёт**

по дисциплине «Управление качеством программных систем»

**Лабораторная работа 2**

Выполнил:

студент учебной группы ПИ21-3 Шарипов Николай Ильгизарович

Факультет *«Информационные технологии и анализ больших данных»*

Направление подготовки: *09.03.03 «Прикладная информатика»*

Профиль:

*«ИТ-сервисы и технологии обработки данных в экономике и финансах»*

Научный руководитель:

Доцент, КТН

*Клочков Евгений Юрьевич*

  Москва 2024

**Отчет по лабораторной работе №2**

**Анализ и тестирование нейронной сети для кредитного скоринга**

**a. Цель работы:**

1. Проанализировать готовую нейронную сеть для кредитного скоринга.
2. Определить применимые к ней виды, типы, методы и уровни тестирования.
3. Привести примеры дефектов, характерных для каждого вида тестирования.
4. Сделать выводы о качестве работы нейронной сети.

**b. Описание программного проекта:**

1. Тип проекта: Классификация
2. Задача: Прогнозировать вероятность дефолта заемщиков по кредитам.
3. Модель: Нейросетевая модель, обученная на исторических данных о кредитах.
4. Входы: Данные о заемщиках (возраст, доход, кредитная история и т.д.).
5. Выход: Прогноз вероятности дефолта (бинарная классификация).

**c. Виды, типы, методы и уровни тестирования:**

Виды тестирования:

1. Функциональное тестирование: Проверка соответствия функций нейронной сети требованиям.

import numpy as np

# Загружаем тестовые данные

X\_test, y\_test = load\_test\_data()

# Запускаем нейронную сеть на тестовых данных

y\_pred = model.predict(X\_test)

# Рассчитываем метрики точности

accuracy = accuracy\_score(y\_test, y\_pred)

sensitivity = recall\_score(y\_test, y\_pred)

specificity = specificity\_score(y\_test, y\_pred)

print("Точность:", accuracy)

print("Чувствительность:", sensitivity)

print("Специфичность:", specificity)

1. Нефункциональное тестирование: Проверка нефункциональных характеристик нейронной сети, таких как производительность, точность, надежность и т.д.

import numpy as np

# Загружаем тестовые данные

X\_test, y\_test = load\_test\_data()

# Запускаем нейронную сеть на тестовых данных

y\_pred = model.predict(X\_test)

# Рассчитываем метрики точности

accuracy = accuracy\_score(y\_test, y\_pred)

sensitivity = recall\_score(y\_test, y\_pred)

specificity = specificity\_score(y\_test, y\_pred)

print("Точность:", accuracy)

print("Чувствительность:", sensitivity)

print("Специфичность:", specificity)

Типы тестирования:

1. Черное тестирование: Тестирование нейронной сети без знания ее внутренней структуры.
2. Белое тестирование: Тестирование нейронной сети с знанием ее внутренней структуры.

Методы тестирования:

1. Статическое тестирование: Анализ нейронной сети без ее запуска.
2. Динамическое тестирование: Анализ нейронной сети во время ее выполнения.

Уровни тестирования:

1. Модульное тестирование: Тестирование отдельных модулей нейронной сети.
2. Интеграционное тестирование: Тестирование взаимодействия модулей нейронной сети.
3. Системное тестирование: Тестирование нейронной сети как системы в целом.

**d. Примеры дефектов:**

Функциональное тестирование:

1. Нейронная сеть некорректно классифицирует некоторые типы заемщиков.
2. Нейронная сеть не учитывает все необходимые факторы при прогнозировании вероятности дефолта.

Нефункциональное тестирование:

1. Нейронная сеть работает слишком медленно.
2. Нейронная сеть не является достаточно точной.
3. Нейронная сеть не является достаточно надежной.

**e. Выводы по работе:**

1. Нейронная сеть для кредитного скоринга является сложным программным продуктом, требующим тщательного тестирования.
2. Для тестирования нейронной сети необходимо использовать различные виды, типы, методы и уровни тестирования.
3. Примеры дефектов, приведенные в данном отчете, не являются исчерпывающими.

**f. Список использованных источников:**

* <https://habr.com/ru/companies/alfa/articles/680346/>
* <https://pytorch.org/>
* <https://www.youtube.com/watch?v=319u2OEgUZ8>

**Контрольные вопросы по тестированию ПО**

**1. Определяется ли качество ПО качеством программного кода?**

Нет, качество ПО не определяется исключительно качеством программного кода.

Хотя хорошо написанный код является важной предпосылкой для создания качественного ПО, он не гарантирует,что ПО будет соответствовать всем требованиям пользователей и будет работать без ошибок.

На качество ПО также влияют такие факторы, как:

1. Требования: Четкость, полнота и непротиворечивость требований к ПО.
2. Дизайн: Архитектура ПО, выбор технологий и алгоритмов.
3. Тестирование: Тщательность и полнота тестирования ПО.
4. Документация: Качество документации ПО.
5. Используемость: Удобство использования ПО для пользователей.
6. Производительность: Скорость и эффективность работы ПО.
7. Безопасность: Уровень защиты ПО от несанкционированного доступа, модификации и разрушения.
8. Надежность: Способность ПО работать без сбоев и отказов.

**2. Какие существуют виды тестирования?**

Существует множество видов тестирования, которые можно классифицировать по различным признакам.

По цели тестирования:

1. Функциональное тестирование: Проверка соответствия функций ПО требованиям.
2. Нефункциональное тестирование: Проверка нефункциональных характеристик ПО, таких как производительность, безопасность, надежность и т.д.
3. Приемочное тестирование: Тестирование ПО, проводимое заказчиком для оценки его соответствия своим требованиям.

По уровню тестирования:

1. Модульное тестирование: Тестирование отдельных модулей ПО.
2. Интеграционное тестирование: Тестирование взаимодействия модулей ПО.
3. Системное тестирование: Тестирование ПО как системы в целом.
4. Приемочное тестирование: Тестирование ПО, проводимое заказчиком.

По методу тестирования:

1. Ручное тестирование: Тестирование, проводимое вручную тестировщиками.
2. Автоматизированное тестирование: Тестирование, проводимое с помощью автоматизированных инструментов.
3. Статическое тестирование: Анализ ПО без его запуска.
4. Динамическое тестирование: Анализ ПО во время его выполнения.

По типу тестируемого объекта:

1. Тестирование единичного кода: Тестирование небольших фрагментов кода.
2. Тестирование компонентов: Тестирование отдельных компонентов ПО.
3. Тестирование приложений: Тестирование готовых приложений.
4. Тестирование систем: Тестирование сложных систем, состоящих из нескольких приложений.

**3. Какие существуют типы тестирования?**

Тип тестирования - это более узкая категория, чем вид тестирования.

Некоторые распространенные типы тестирования:

1. Черное тестирование: Тестирование ПО без знания его внутренней структуры.
2. Белое тестирование: Тестирование ПО с знанием его внутренней структуры.
3. Дымовое тестирование: Быстрое тестирование ПО, чтобы выявить критические ошибки.
4. Регрессионное тестирование: Повторное тестирование ПО после внесения изменений, чтобы убедиться,что эти изменения не привели к новым ошибкам.
5. Нагрузочное тестирование: Тестирование ПО под нагрузкой, чтобы оценить его производительность и устойчивость к нагрузкам.
6. Напряженное тестирование: Тестирование ПО на пределе его возможностей, чтобы выявить его слабые места.
7. Уязвимое тестирование: Тестирование ПО на наличие уязвимостей, которые могут быть использованы злоумышленниками.

**4. Какие существуют методы тестирования?**

1. Метод тестирования - это конкретный подход к реализации тестирования.
2. Некоторые распространенные методы тестирования:
3. Эксплорационное тестирование: Непланируемое тестирование, при котором тестировщик свободно исследует ПО.
4. Управляемое тестирование: Тестирование, которое проводится по заранее составленному плану.
5. Сценарное тестирование: Тестирование ПО на основе сценариев, которые описывают типичные действия пользователей.
6. Тестирование на основе ключевых слов: Тестирование ПО на основе ключевых слов, которые извлекаются из требований к ПО.
7. Управляемое тестирование данными: Тестирование ПО с использованием различных тестовых данных.

**5. Какие существуют уровни тестирования?**

1. **Уровень тестирования** - это этап тестирования ПО, на котором оно проверяется на соответствие определенному набору требований.
2. **Существует несколько уровней тестирования:**
3. **Модульное тестирование:** Тестирование отдельных модулей ПО. Проводится на ранних стадиях разработки ПО.
4. **Интеграционное тестирование:** Тестирование взаимодействия модулей ПО. Проводится после завершения модульного тестирования.
5. **Системное тестирование:** Тестирование ПО как системы в целом. Проводится после завершения интеграционного тестирования.
6. **Приемочное тестирование:** Тестирование ПО, проводимое заказчиком для оценки его соответствия своим требованиям. Проводится после завершения системного тестирования.