Орищенко Виталий Алексеевич

[Vitosh992@gmail.com](mailto:Vitosh992@gmail.com)

+79585005653

Группа 1

**Отчет к итоговому заданию по курсу «Архитектор в области искусственного интеллекта»**

Тема: Разработка и тестирование приложения для анализа и прогнозирования данных в промышленности с помощью нейронных сетей

Москва, 2023

**Оглавление**

1. Постановка задачи……...…………………………………………………..3
2. Обзор моделей……...………………………………………………………3
3. Общая концепция решения……...………………………………...………4
4. Результаты работы……...………………………………………………….5
5. Заключение……...………………………………………………………….8
6. Список литературы……...…………………………………………………9
7. **Постановка задачи**

В рамках выбранной темы была выбрана отрасль медицинского страхования. Соответственно, в открытом доступе был выбран датасет insurance.csv, в котором имеются построчно данные по 6 входным параметрам (возраст, пол, индекс массы тела, количество детей, наличность табачной зависимости, регион проживания), на основе которых выдается параметр стоимости медицинской страховки для этого человека.

В рамках этих данных была поставлена цель создания и обучения модели нейронной сети прогнозирования стоимости страхования по входным данным, интеграция обученной модели в созданное программной обеспечение, сравнение полученного результата с другими методами и формирование выводов об эффективности использования НС для выполнения данной задачи.

1. **Обзор моделей**

Задача прогнозирования целевого значения на основании множества независимых отлична решается различными инструментами:

* Основной математический метод – регрессионный анализ. Путем формирования модели множественной регрессии можно достаточно эффективно смоделировать определение целевого значения, путем определения значений весов для каждого из независимых переменных. Это наиболее распространенный метод решения данной задачи.

Применим данный метод для решения нашей задачи. Моделирование было осуществлено в программном обеспечении STATISTICA. В результате можно увидеть следующий график зависимости реального значения от прогнозируемого (рис. 1):

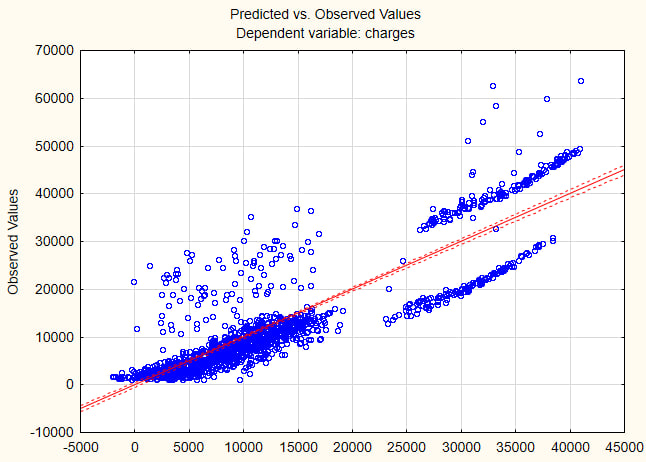


Рисунок 1. График прогнозируемого и реального значения множественной регрессии.

* Использование моделей нейронных сетей. В последнее время набирает все большую популярность, метод показывает высокую эффективность с тем условием, что данных достаточно и предобработка данных выполнена качественным образом. Именно этим способом мы и будем выполнять поставленную задачу.

1. **Общая концепция модели**

Выполнять нашу задачу будем, используя следующий стек технологий:

Python (torch, Flask, sklearn, psycopg2, numpy, pandas), PostgreSQL (DBeaver), Jupyter Notebook, PyCharm.

В рамках нашей задачи в первую очередь создадим и обучим модель нейронной сети на наших данных, используя библиотеку PyTorch в Jupyter Notebook. После подбора оптимальных параметров модели, сохраним модель в виде файлов для будущего использования в другой программе для прогнозирования.

Далее создадим веб-интерфейс в виде страниц типа .html, который будет включать в себя две страницы: первая – страница ввода данных, вторая – страница получения результатов оценки.

После в PyCharm создадим проект, в котором создадим веб-приложения Flask, к которому будут привязаны написанные нами .html страницы.

Далее подключим БД Postgres к нашему проекту. Создадим логику добавления всех вводных данных и получившихся расчетов в таблицу в БД с указанием времени. Также все данные будут отображаться на второй .html странице.

1. **Результаты работы**

На этапе формирования модели ИИ в библиотеки torch я столкнулся с проблемой недостатка исходных данных (было около 1350 строк). С учетом размерности данных прогноза этого мало. Однако у меня получилось создать модель, которая прогнозирует значения на уровне 0.88 коэффициента детерминации. Учитывая малочисленность данных для обучения – это хороший результат. Ниже можно наблюдать график реальных и прогнозируемых значений модели (рис. 2), график значений функции потерь по каждой эпохе (рис. 3), численные значения функции потерь каждые 10 эпох (рис. 4). Стоит заметить, что значения функции потерь принимают достаточно большое значение, но это сумма всех потерь за эпоху, так что результат неплохой.

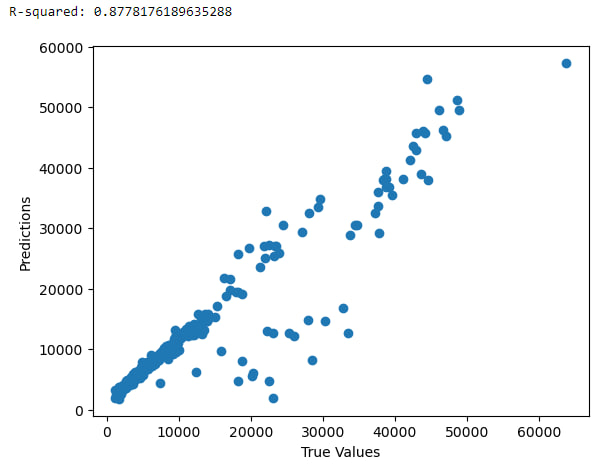


Рисунок 2. График реальных и прогнозируемых значений обученной модели НС.

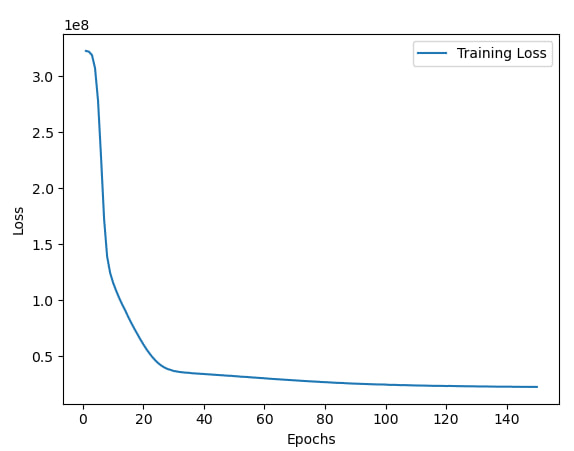


Рисунок 3. График функции потерь по эпохам.

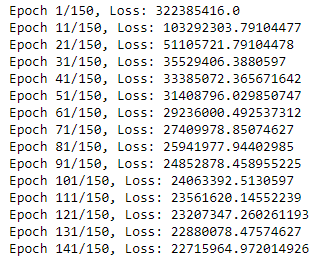


Рисунок 4. Значения функции потерь каждый 10 эпох.

Когда модель была адекватно обучена, были сохранены файлы весов модели и файл трансформатором модели, что обеспечило использование этой модели вне программы. Была создана отдельная программа, в корень которой были загружены файлы для модели. Также были созданы шаблоны HTML-страниц, необходимые для визуального взаимодействия с моделью.

После создания итогового приложения, модель успешно принимает входные значения (стр. 5) и выводит расчет целевой функции вместе со всеми входными параметрами (рис. 6 и 7). Также модель при каждом расчете делает SQL-запрос на БД Postgres со вставкой всех данных (рис. 8).

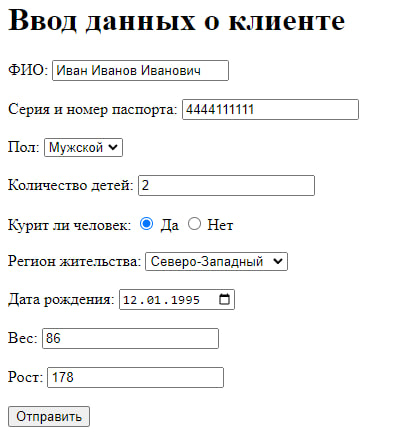


Рисунок 5. Страница для ввода данных.

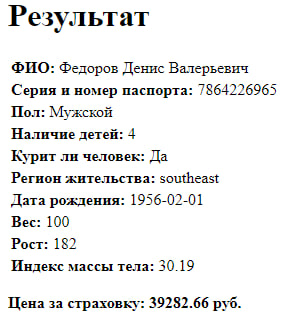


Рисунок 6. Пример вывода данных на странице.

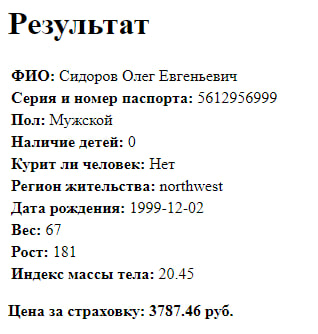


Рисунок 7. Пример вывода данных на странице.

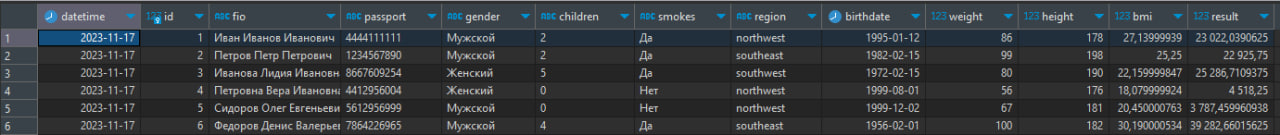


Рисунок 8. Данные в таблице БД Postgres.

Можно заметить, что с увеличением ИМТ или вредных факторов для здоровья, стоимость медицинской страховки существенно увеличивается.

1. **Заключение**

В заключении хочется сравнить два метода решения этой задачи: метод множественной регрессии и создание модели искусственного интеллекта. По графику реального/прогнозируемого значения можно однозначно ответить, что модель ИИ справилась с задачей лучше, но для практического применения модель необходимо улучшить. Для этого необходима большая выборка исходных данных. Предполагаю, что для множественной регрессии количество данных для обучения также оказалось сильно мало, из-за чего и результат оказался хуже. Однако, с точки зрения потенциала, если бы было огромное количество данных для обучения – нейронная сеть безусловно была бы более эффективна.

1. **Список литературы**
2. Орельен Жерон − Прикладное машинное обучение с помощью Scikit-Learn и TensorFlow, 2018 г.
3. Ясер Абу-Мостафа, Малик Магдон-Исмаил, Сюань-Тянь Линь − Learning From Data, 2012 г.
4. Андрей Бурков − The Hundred-Page Machine Learning Book, 2019 г.
5. Кристоф Молнар − Interpretable Machine Learning, 2019 г.
6. Г. Домбровская, Б. Новиков, А. Бейликова − Оптимизация запросов PostgreSQL
7. М. Гринберг – Flask Web Development, 2018 г.