Министерство науки и высшего образования РФ

федеральное государственное автономное  
образовательное учреждение высшего образования

«Омский государственный технический университет»

Факультет информационных технологий и компьютерных систем

Кафедра «Прикладная математика и фундаментальная информатика»

|  |
| --- |
| **Расчетно-графическая работа** |
| по дисциплине **Машинное обучение и большие данные**  **Тема: Разработка Web-приложения (дашборда)**  **для вывода моделей ML и анализа данных** |

|  |  |  |  |
| --- | --- | --- | --- |
| Студента | Абрамов Егор Владимирович | | |
| Курс | 2 | Группа | ФИТ-221 |
| Направление | 02.03.02 Фундаментальная информатика и информационные технологии | | |
| Руководитель | Доц, К.Н. | | |
|  | Моисеева Н.А. | | |
| Выполнил | 30.12.2023 | | |
| Проверил | дата, подпись руководителя | | |

Омск 2023

Оглавление

[Введение 3](#_Toc155516014)

[Основа для приложения. 3](#_Toc155516015)

[Технологии для обучения, сериализации и десериализации моделей, обработки данных. 3](#_Toc155516016)

[Краткое описание моделей и используемых инструментов сериализации. 4](#_Toc155516017)

[Модель обучения с учителем(Задача регрессии) 4](#_Toc155516018)

[Модель обучения с учителем(Задача классификации) 4](#_Toc155516019)

[Модель обучения без учителя(задача кластеризации, датасет регрессии) 4](#_Toc155516020)

[Модель обучения без учителя(задача кластеризации, датасет классификации) 4](#_Toc155516021)

[Ансамблевые модели(Stacking, Bagging, Boosting) 5](#_Toc155516022)

[Полносвязная нейронная сеть(датасет регрессии) 5](#_Toc155516023)

[Полносвязная нейронная сеть(датасет классификации) 5](#_Toc155516024)

[Процесс разработки дашборда 6](#_Toc155516025)

[Начало разработки 6](#_Toc155516026)

[Создание первых страниц 6](#_Toc155516027)

[Создание страницы с выводом моделей 7](#_Toc155516028)

[Сохранение и загрузка моделей 7](#_Toc155516029)

[Работа над страницей 7](#_Toc155516030)

[Развертка проекта 8](#_Toc155516031)

[Заключение 8](#_Toc155516032)

# Введение

## Основа для приложения.

Для разработки дашборда для вывода моделей ML и анализа данных использовался язык python и соответственно библиотеки данного языка. За основу для создания дашборда была взята библиотека streamlit, данная библиотека позволила создать приложение без углубления во frontend разработку, а используя лишь знания языка и основных методов, содержащихся в библиотеке.

## Технологии для обучения, сериализации и десериализации моделей, обработки данных.

Для обучения моделей классического машинного обучения использовалась библиотека sklearn и датасеты используемые при выполнении лабораторных работ. Для сохранения данных моделей использовалась библиотека pickle, она же применялась для последующего считывания. Для полносвязной нейронной сети же на этапе обучения использовалась библиотека tensotrflow, для сохранения и чтения модели также применялись встроенные возможности библиотеки. Для обработки данных, применяемых для всех моделей, чтения датасетов и других манипуляций производимых с данными использовались библиотеки – pandas, numpy, imblearn. Для визуализации данных использовались библиотеки matplotlib и seaborn.

# Краткое описание моделей и используемых инструментов сериализации.

## Модель обучения с учителем(Задача регрессии)

Выбранной моделью обучения с учителем для задачи регрессии стала модель полиномиальной регрессии, так как при обучении она показала лучшие результаты, коэффициент детерминации практически равный 1. Для сериализации и десериализации использовалась библиотека pickle.

## Модель обучения с учителем(Задача классификации)

Выбранной моделью обучения с учителем для задачи классификации стала модель метод k ближайших соседей(KNN), так как при обучении она показала лучшие результаты, рейтинг f1 равный 1 а количество неверных прогнозов порядка тысячной доли процента. Для сериализации и десериализации использовалась библиотека pickle.

## Модель обучения без учителя(задача кластеризации, датасет регрессии)

Выбранной моделью обучения без учителя для задачи кластеризации датасета регрессии стала модель KMeans, в целом для регрессии все из использованных моделей показали крайне низкий результат, около 0.15 – 0.2 для метрики – силуэт. Для сериализации и десериализации использовалась библиотека pickle.

## Модель обучения без учителя(задача кластеризации, датасет классификации)

Выбранной моделью обучения без учителя для задачи кластеризации датасета классификации стала модель иерархической кластеризации показав результат для метрики силуэт – около 0.8. Для сериализации и десериализации использовалась библиотека pickle.

## Ансамблевые модели(Stacking, Bagging, Boosting)

Для обоих датасетов все ансамблевые модели показали почти идеальный результат. Для моделей регрессии коэффициент дискриминации равен почти единице, с различиями между моделями в тысячных единицах. Для моделей же классификации рейтинг f1 равен 1, для всех моделей.

## Полносвязная нейронная сеть(датасет регрессии)

Для модели регрессии была создана сеть состоящая из 3 слоев.

1. 1 слой – 512 нейронов, Dropout(0.5), функция активации – relu.
2. 2 слой – 256 нейронов, Dropout(0.25), функция активации – relu.
3. 3 слой – 1 нейрон, функция активации – relu.

Результат – коэффициент детерминации >0.9.

## Полносвязная нейронная сеть(датасет классификации)

Для модели классификации была создана сеть состоящая из 5 слоев.

1. 1 слой – 10 нейронов, Dropout(0.5), функция активации – tanh.
2. 2 слой – 8 нейронов, Dropout(0.5), функция активации – tanh.
3. 3 слой – 6 нейронов, Dropout(0.5), функция активации – linear.
4. 4 слой – 4 нейрона, Dropout(0.5), функция активации – relu.
5. 5 слой – 1 нейрон, функция активации – sigmoid.

Результат – рейтинг f1 около 0.75

# Процесс разработки дашборда

## Начало разработки

Первым делом во время разработки было установлено все необходимое окружение, создан новый репозиторий на GitHub. Далее определена примерная структура проекта и встал вопрос о его размещении. Для размещения был зарегистрирован аккаунт на Streamlit cloud, на сайте создан проект запускаем из созданного GitHub репозитория. Далее была изучена документация Streamlit.

## Создание первых страниц

После завершения подготовки и чтения необходимой документации, первым делом была создана первая страница с информацией об авторе. Код приведен ниже.

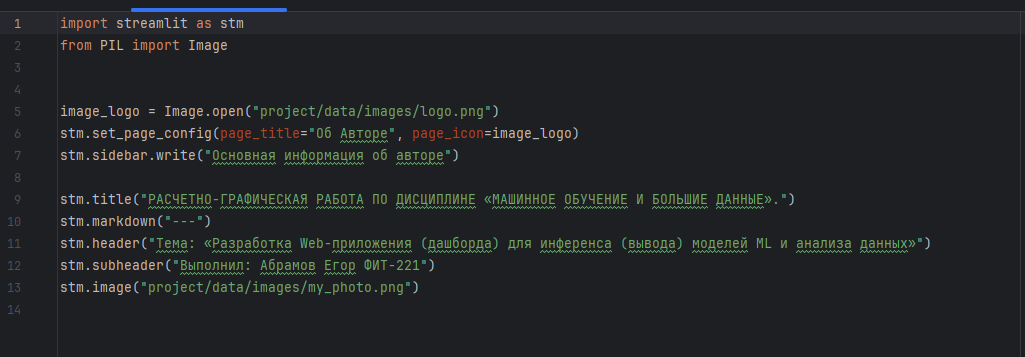


Рисунок 1. Исходный код первой страницы

Далее была создана страница с визуализацией зависимостей в датасетах. Т.к. на обработку многих из визуализации уходит много ресурсов и времени, да и не рационально будет ее проводить при каждом запросе, было решено выполнить их обработку на компьютере и сохранить в виде обычных изображений, соответственно при обращении к странице с визуализациями просто выводить эти изображения.

После создания 2 страницы я перешел к созданию 3 страницы содержащий информацию о датасетах, здесь также ничего интересного, просто вывод самих датасетов и соответствующей информацией о них.

## Создание страницы с выводом моделей

### Сохранение и загрузка моделей

Данная часть оказалось самой трудоемкой. Прежде чем браться за создание страниц было необходимо сохранить сами модели. Для модели нейросети использовались встроенные возможности - save для сохранения модели и loadmodel для загрузки. С моделями sklearn дело обстоит немного иначе, здесь все происходило с помощью модуля pickle где для сохранения использовалась метод dump, а для загрузки метод load.

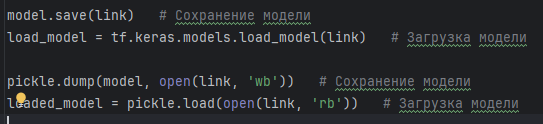


Рисунок 2 сохранение и загрузка моделей

### Работа над страницей

При создании страницы было решено разбить ее на 3 части(получение результатов от модели регрессии, получение результатов от модели классификации, и получение результатов от одной из двух моделей на выбор по строчке данных)

На самой странице в первую очередь была реализована возможность загрузки датасета, с проверкой правильности загруженного файла. Далее для моделей(классификации, регрессии) были созданы подпункты для каждой из моделей обучения(KNN, Kmeans и т.д.), тут при поступлении правильного файла от пользователя, модель выдает результат работы, метрики и визуализирует полученный результат. Весь исходный код хранится в репозитории [GitHub](https://github.com/Otocolobus-manules/RGR_ML_3_SEM).

## Развертка проекта

Для развертки проекта был использован сервис Stream cloud, для этого был создан аккаунт, сопряжен с аккаунтом на гитхаб и запущенно приложение из репозитория с выполненной ргр, с выбором соответствующей версии языка пайтон, для версий ниже 3.10 программа выполняться не будет. [Ссылка на приложение.](https://ishakspace.streamlit.app)

# Заключение

В процессе выполнения работы были изучены способы сериализации и десериализации моделей машинного обучения и нейросетей. Частично дополнены знания о библиотеках pandas, numpy, tensorflow, sklearn, matplotlib и seaborn. Также изучена документация библиотека streamlit, основные принципы построения веб приложений с помощью данной библиотеки. Создано веб приложение для вывода данных о моделях машинного обучения.

# Список используемых источников

1. Сайт с документацией по библиотеке streamlit - <https://docs.streamlit.io>
2. Сервис с вопросами пользователей - <https://stackoverflow.com>

# Приложение А

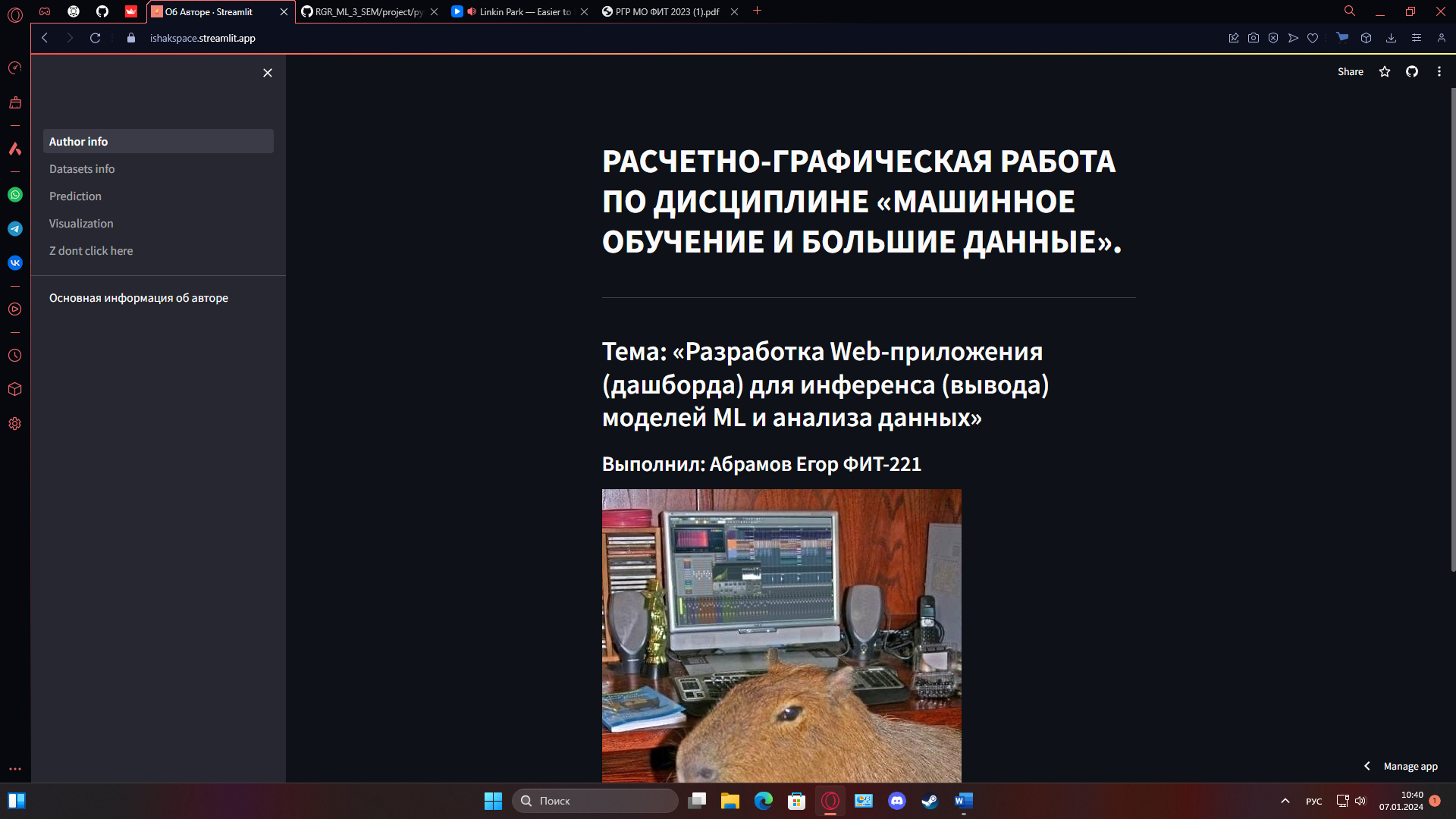


Рисунок 3. Главная страница с информацией об авторе

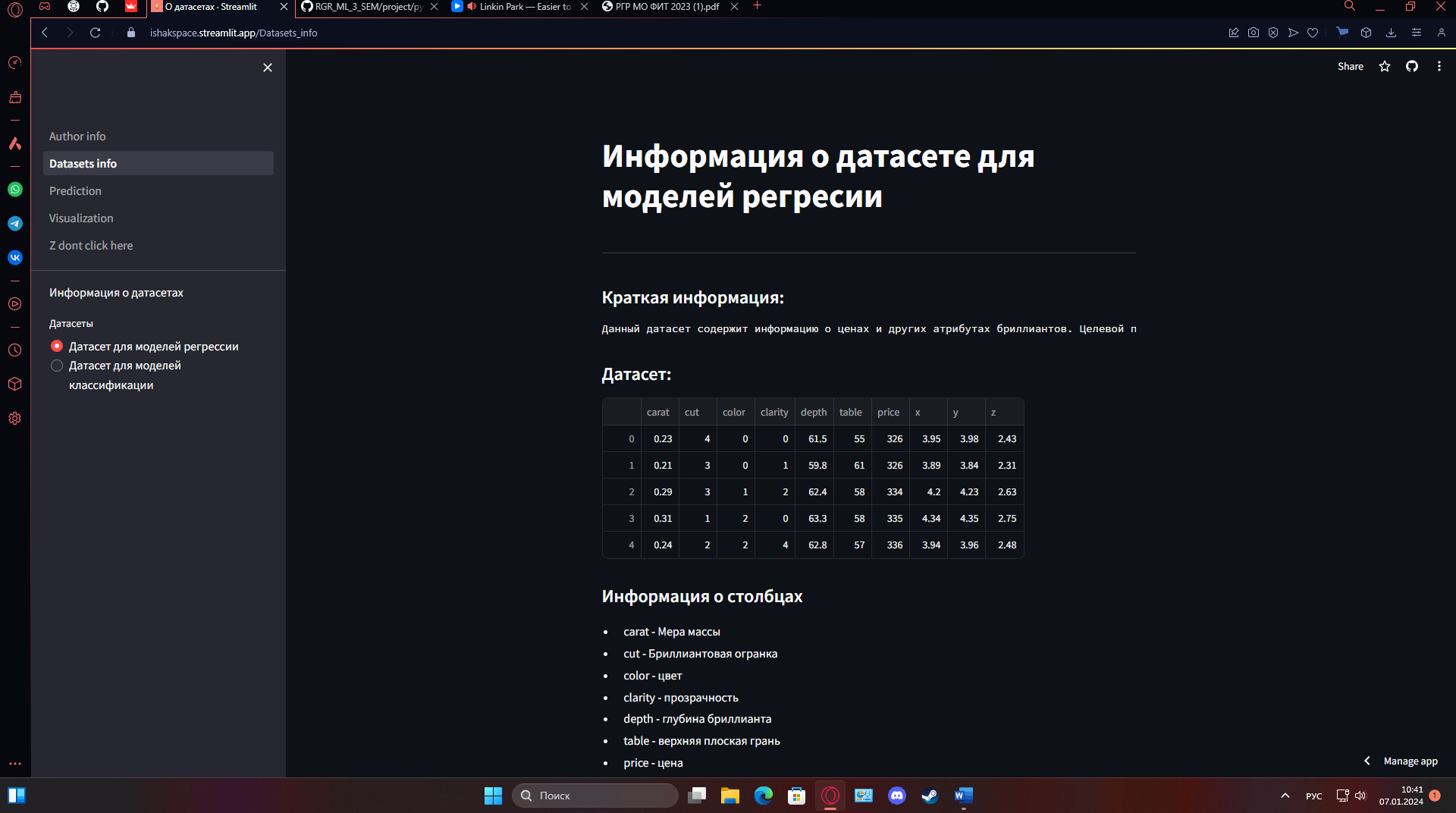


Рисунок 4. Вывод информации о датасете

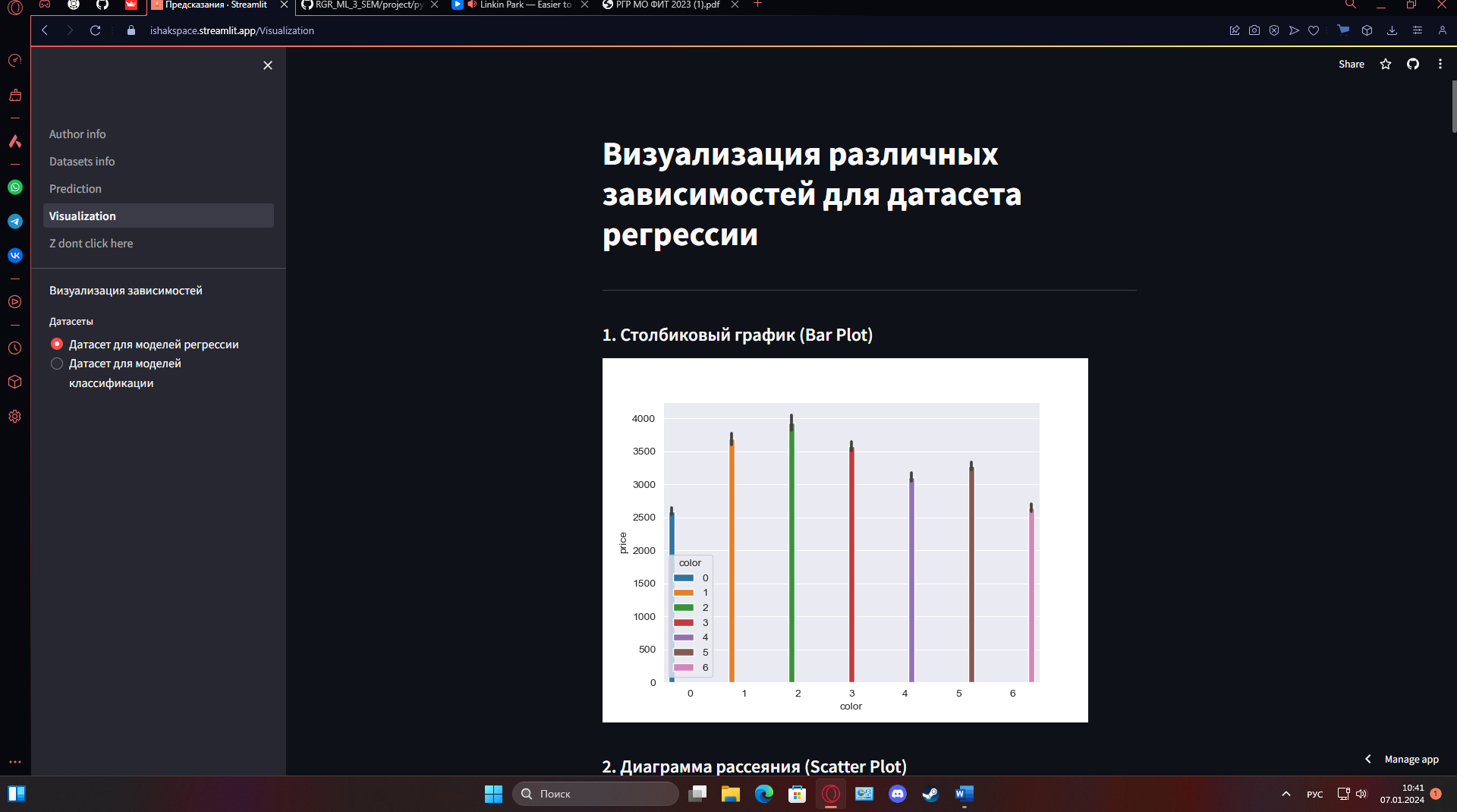


Рисунок 5. Визуализация зависимостей

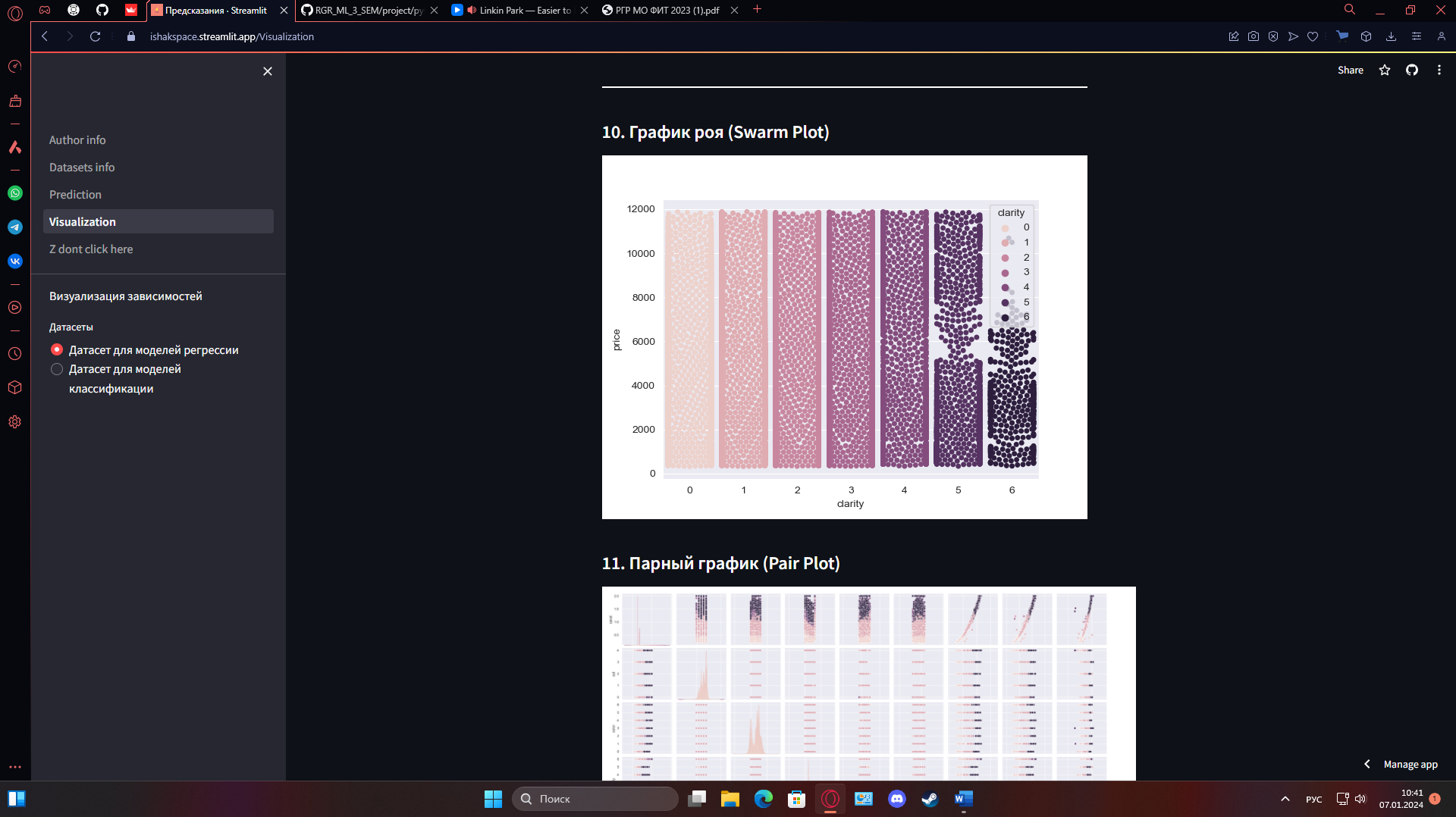


Рисунок 6. Визуализация зависимостей

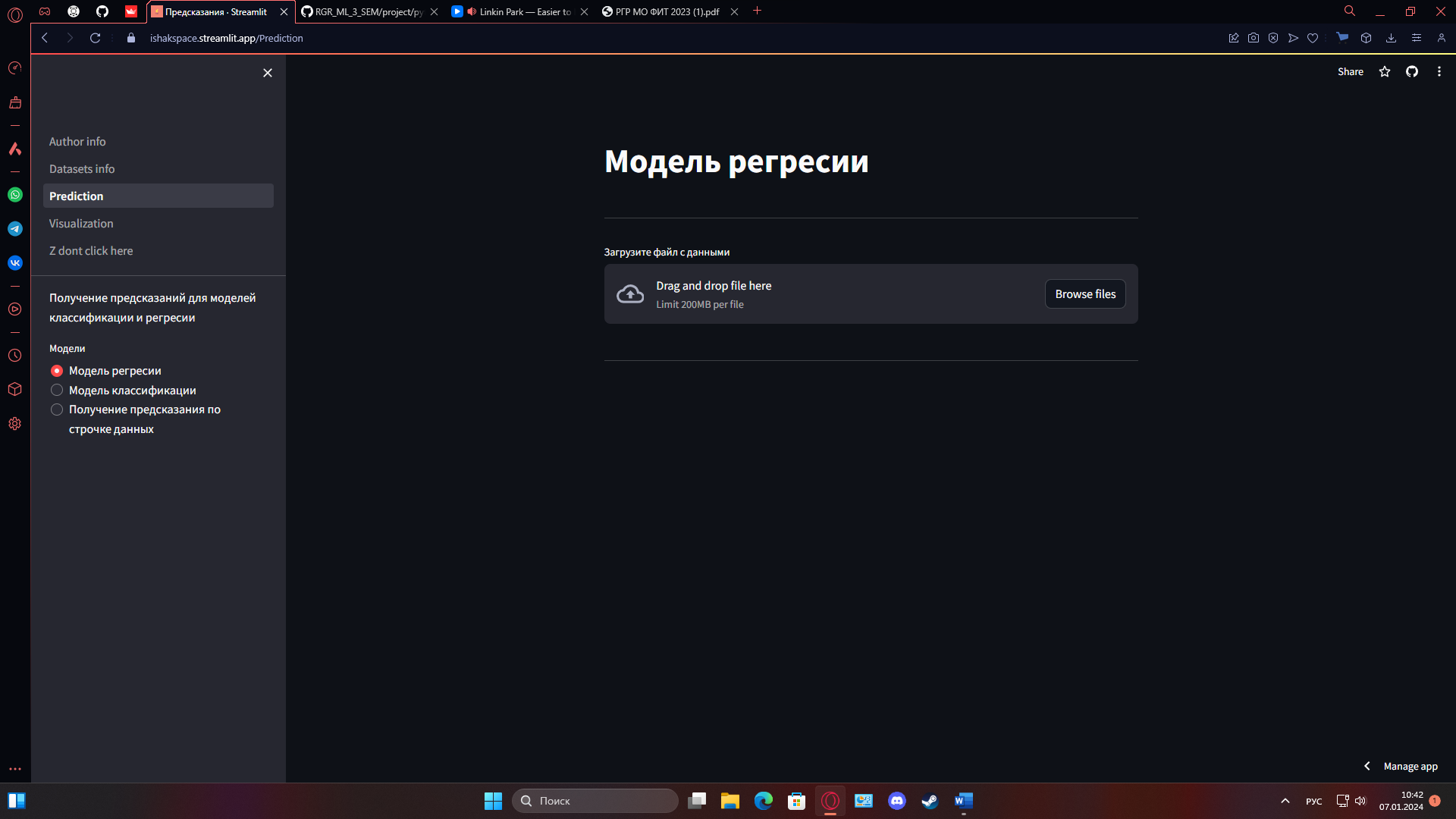


Рисунок 7. Окно с выводом моделей до загрузки датасета

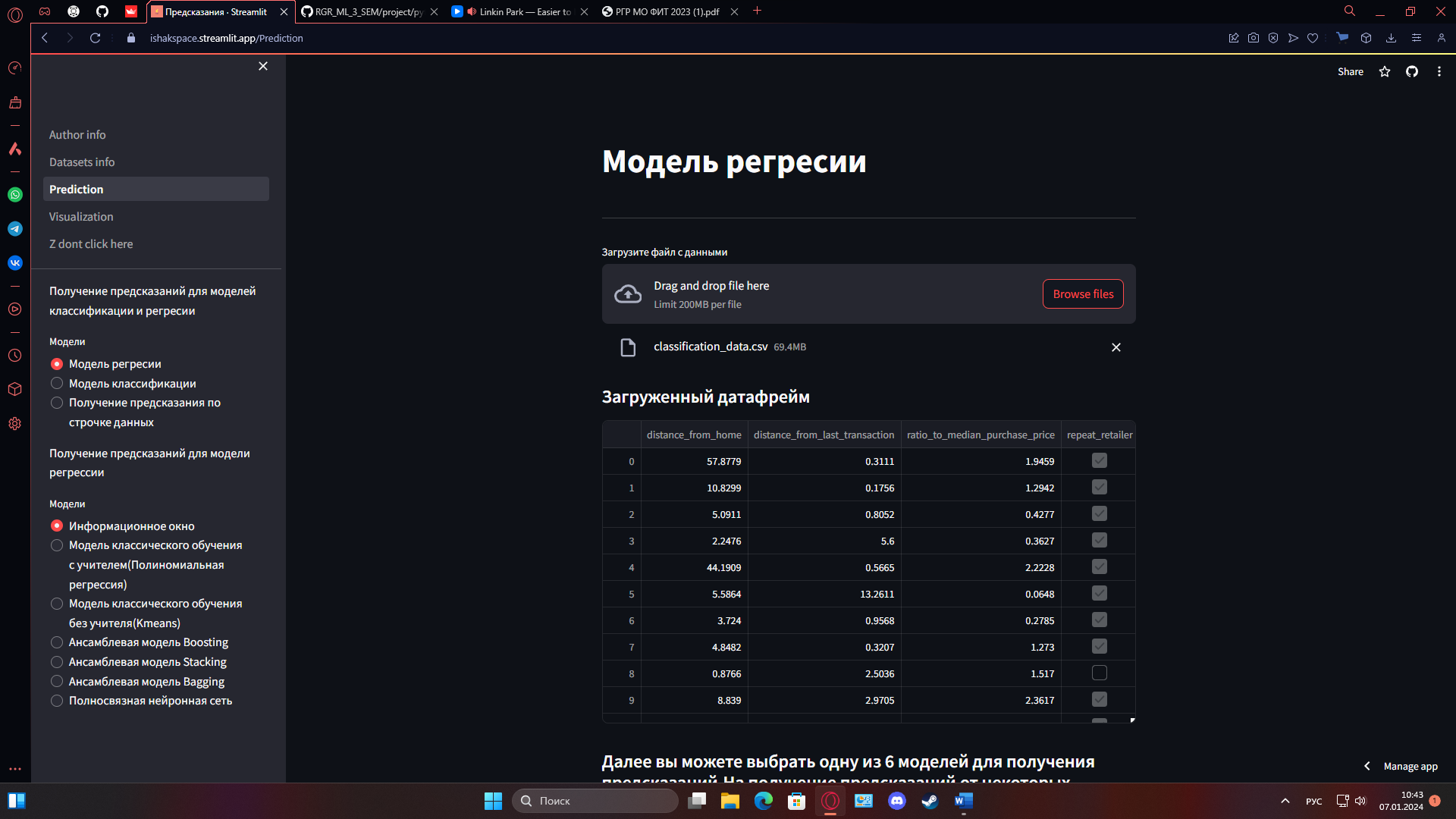


Рисунок 8. Окно с выводом моделей после загрузки датасета

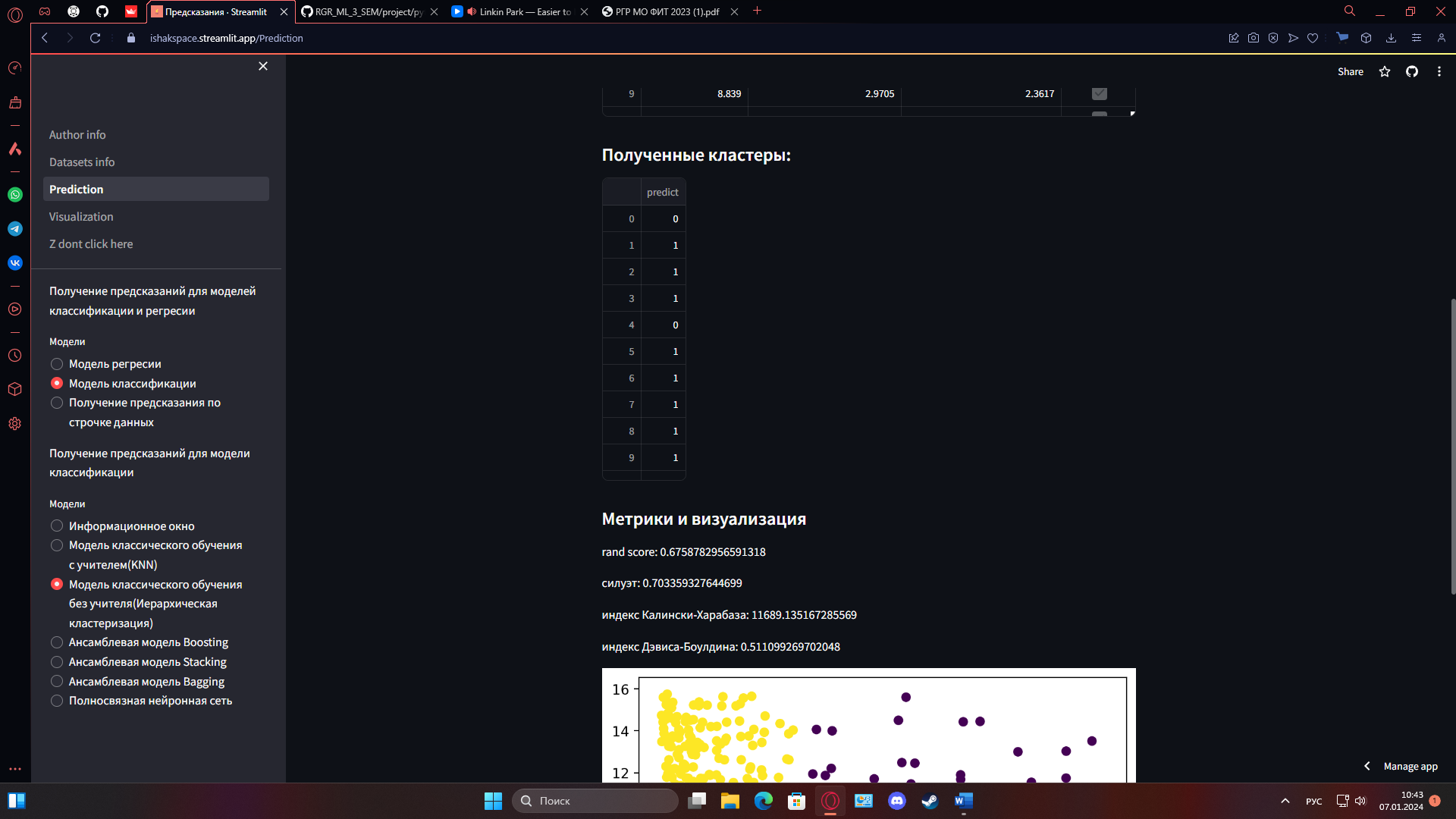


Рисунок 9. Пример вывода модели обучения без учителя

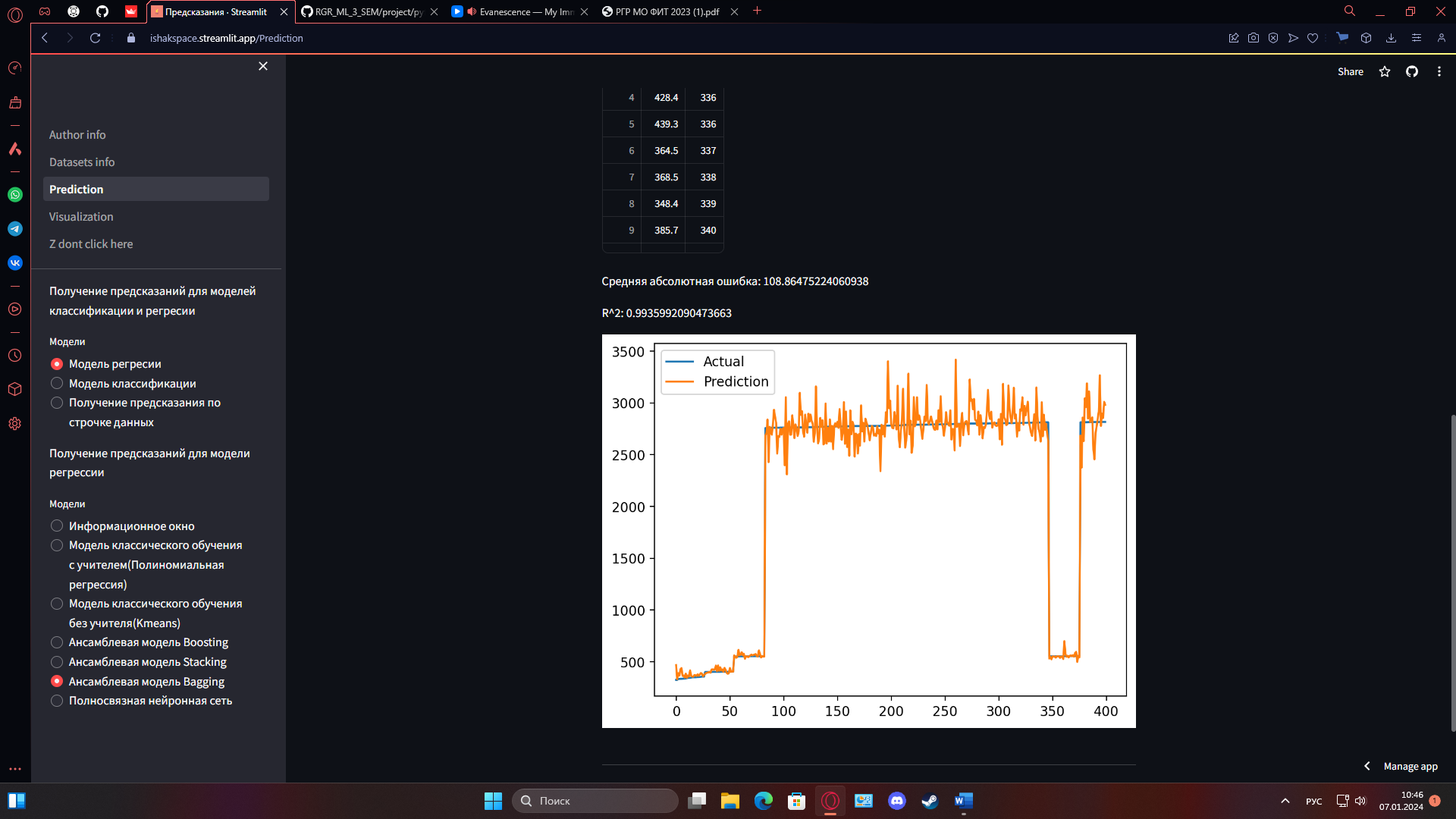


Рисунок 10. Пример вывода ансамблевой модели

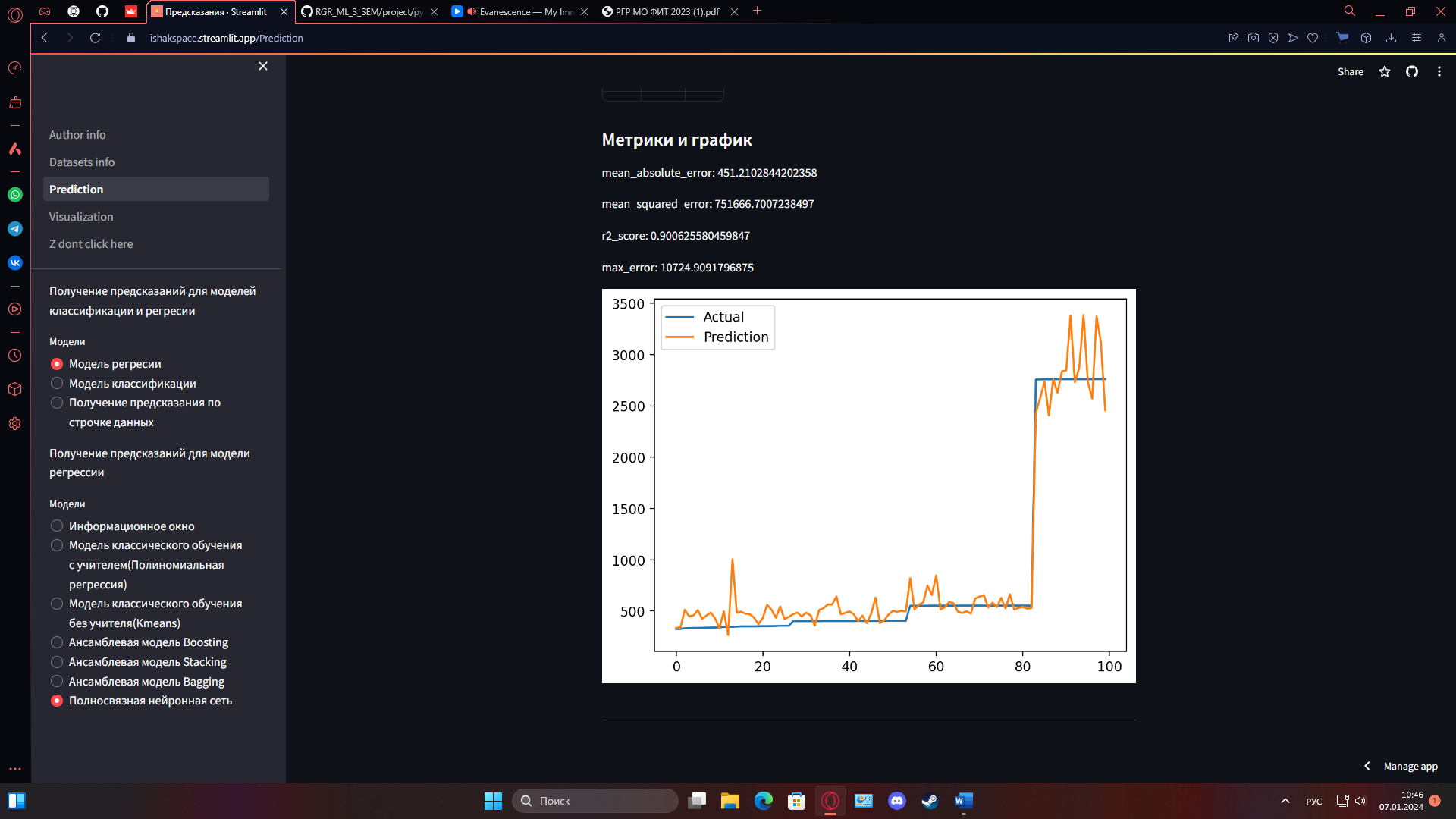


Рисунок 11. Пример вывода для полносвязной нейронной сети

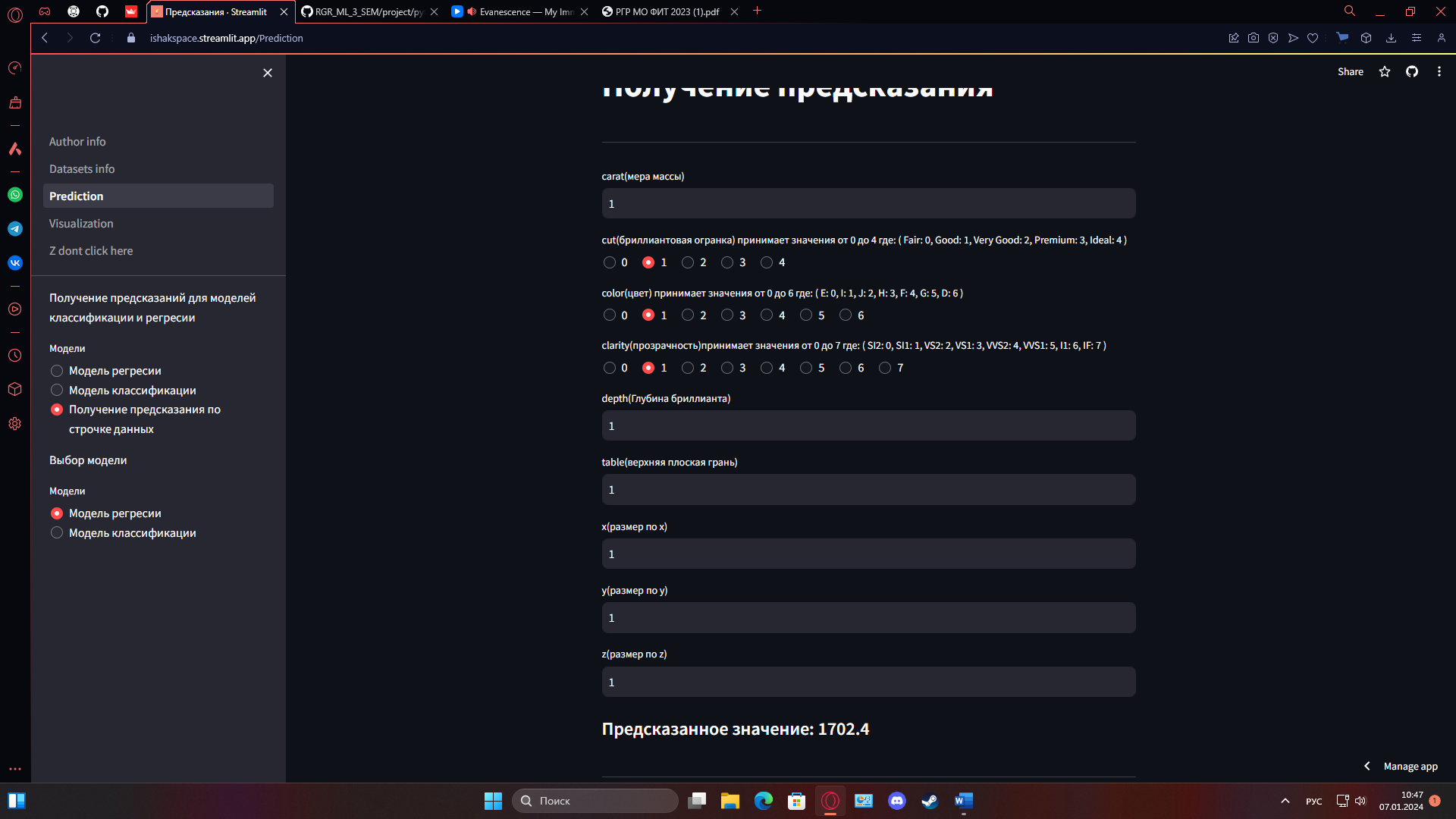


Рисунок 12. Пример получения предсказания по строке данных

# Приложение Б

## Общая структура проекта

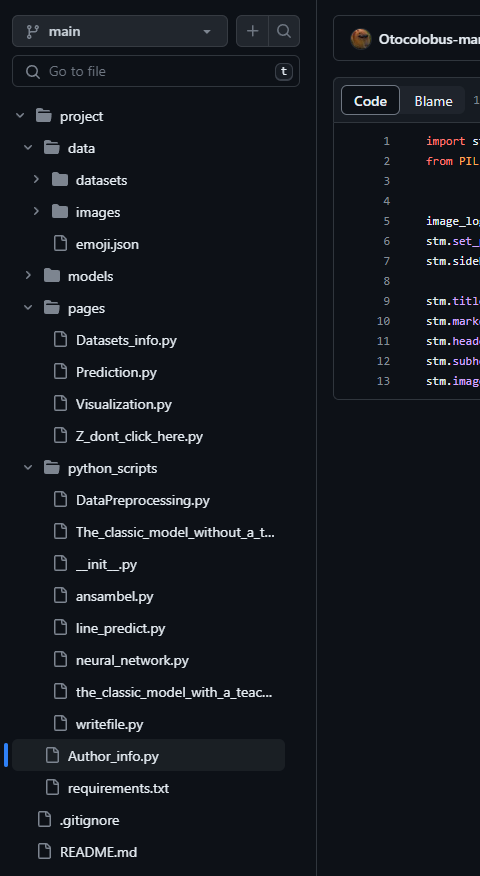


Рисунок 13.Общая структура проекта

Здесь за главную страницу отвечает файл Author\_info.py, остальные страницы хранятся в файле папке pages и в python\_scripts хранятся подпрограммы для выполнения какой либо логики из основной программы, например обратка датасета.

Ссылка на открытый репозиторий где хранится весь исходный код - <https://github.com/Otocolobus-manules/RGR_ML_3_SEM>

Ссылка на само приложение - <https://ishakspace.streamlit.app>