

Создание и развёртывание интерактивного веб-приложения для анализа данных с использованием Streamlit

Андриенко Иван Сергеевич

Приамурский государственный университет имени Шолом-Алейхема

Студент

Аннотация

В статье описан процесс создания и развёртывания интерактивного веб-приложения для анализа данных из CSV-файлов с использованием библиотеки Streamlit на языке программирования Python. Рассматриваются ключевые этапы разработки приложения, включая создание пользовательского интерфейса, визуализацию данных, а также публикацию приложения на платформе Streamlit community cloud для обеспечения публичного доступа. Приводится пример реализации приложения, позволяющего загружать данные, выполнять базовый статистический анализ и строить графики на основе выбранных параметров. Освещены преимущества использования Streamlit и Streamlit community cloud для быстрого создания и развёртывания веб-приложений.

Ключевые слова: веб-приложение, Streamlit, Python, анализ данных, визуализация данных, Streamlit community cloud.

Creating and deploying an interactive web application for data analysis using Streamlet

Andrienko Ivan Sergeevich

Sholom-Aleichem Priamursky State University

Student

Abstract

The article describes the process of creating and deploying an interactive web application for analyzing data from CSV files using the Streamlit library in the Python programming language. The key stages of application development are considered, including the creation of a user interface, data visualization, as well as publishing the application on the Streamlit Community Cloud platform for public access. An example of an application implementation is provided that allows you to download data, perform basic statistical analysis and plot graphs based on selected parameters. The advantages of using Streamlet and Streamlet Community Cloud for fast creation and deployment of web applications are highlighted.

Keywords: web application, Streamlet, Python, Data analysis, Data visualization, Streamlet Community Cloud.

1 Введение

1.1 Актуальность

В современных условиях быстрой цифровизации и расширяющегося объёма данных актуальность средств для их анализа и визуализации неуклонно растёт. Библиотека Streamlit стала важным инструментом, позволяя в короткие сроки создавать интерактивные веб-приложения для анализа и визуализации данных без необходимости глубоких знаний в области веб-разработки.

Помимо создания веб-приложений, важным становится вопрос быстрого и удобного развёртывания этих решений в общедоступный сетевой формат. Streamlit Community Cloud предоставляет простое и бесплатное решение для хостинга интерактивных приложений, сохраняя главные принципы простоты и доступности. Платформа позволяет за счёт оптимизированного процесса развёртывания сразу превращать локальные скрипты в публично-доступные приложения. Данное исследование рассматривает не только процесс создания интерактивных элементов в приложении, но и реализацию процесса его развёртывания, что делает данную работу актуальной для широкой аудитории разработчиков и исследователей.

1.2 Обзор исследований

В исследовании Р.Р. Абдулвелеева и Е.М. Иванина рассматриваются особенности быстрого развёртывания веб-приложений с использованием Streamlit, где особое внимание уделяется его применению для создания дашбордов и визуализации данных, что значительно облегчает адаптацию данных под нужды пользователей [1]. П.С. Конова исследует возможности Streamlit при работе с большими данными, акцентируя внимание на его гибкости и легкости интеграции с инструментами для анализа. В её работе рассматривается, как Streamlit упрощает визуализацию больших объемов данных, обеспечивая более доступное и наглядное представление результатов [2]. В.В. Демичев, Е.А. Алехина и Б.Ш. Дашиева исследуют использование Streamlit в сельском хозяйстве, применяя его для автоматизации статистического анализа урожайности кормовых культур. Авторы показывают, что Streamlit способствует эффективному управлению и прогнозированию данных [3]. О.Ю. Кузнецова, Р.Н. Кузнецов и А.В. Кузьмин в своём исследовании анализируют возможности Streamlit при разработке модели машинного обучения для прогнозирования послеоперационных осложнений. Они отмечают, что библиотека позволяет интегрировать сложные модели в удобные интерфейсы для медицинских приложений [4]. Д.Н. Хамимова в своей работе фокусируется на развёртывании моделей машинного обучения с помощью Streamlit, подчеркивая удобство библиотеки для тестирования и визуализации результатов машинного обучения, что делает её важным инструментом для специалистов в этой области [5]. Ms.P. Keerthi, G.S. Reddy, V.S. Raghava и др. описали применение Streamlit для создания интерфейсов диагностики заболеваний, демонстрируя простоту интеграции с алгоритмами анализа данных [6]. Ya. Akkem, B.S. Kumar, A.

Varanasi рассмотрели применение Streamlit для реализации методов ансамблевого обучения в системах рекомендаций сельскохозяйственных культур, подчёркивая его возможности для построения интерактивных приложений [7].

1.3 Цель исследования

Цель исследования – разработать интерактивное веб-приложение для анализа данных из CSV-файлов с использованием библиотеки Streamlit и продемонстрировать процесс его развёртывания на платформе Streamlit Community Cloud. Приложение должно позволять пользователям загружать данные, проводить статистический анализ и визуализировать результаты, обеспечивая доступность через облачный хостинг.

2 Материалы и методы

Для реализации веб-приложения использовались следующие инструменты:

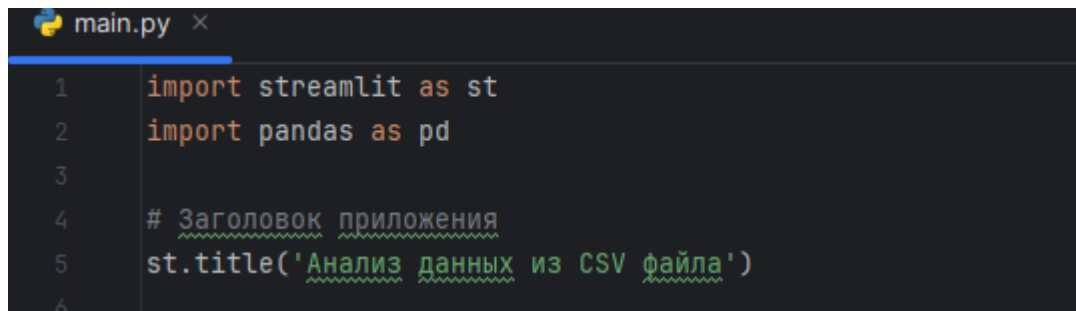
- Язык программирования: Python – основной инструмент для обработки данных и создания логики приложения.
- Библиотека: Streamlit – для построения пользовательского интерфейса и взаимодействия с данными.
- Дополнительные библиотеки: Pandas – для работы с табличными данными; Matplotlib и Seaborn – для визуализации.
- Платформа для хостинга: Streamlit Community Cloud – для развёртывания приложения и обеспечения публичного доступа.

Процесс разработки включал несколько этапов:

- Создание пользовательского интерфейса с элементами для загрузки данных, выбора параметров анализа и отображения результатов.
- Реализация базового функционала анализа данных с использованием Pandas.
- Визуализация данных в виде интерактивных графиков с помощью Matplotlib и Seaborn.
- Настройка и развёртывание приложения на платформе Streamlit Community Cloud через github.

3 Результаты и обсуждения

На первом этапе происходит импорт необходимых библиотек и задание заголовка для приложения (рис. 1). Streamlit предоставляет функции для создания элементов интерфейса, таких как заголовки, загрузчики файлов и графики. В данном случае с помощью функции `st.title()` задается заголовок приложения, который будет отображаться в верхней части веб-страницы. Pandas используется для работы с данными, которые будут загружаться в формате CSV. Эта библиотека позволяет считывать, анализировать и преобразовывать данные в удобные для визуализации форматы.



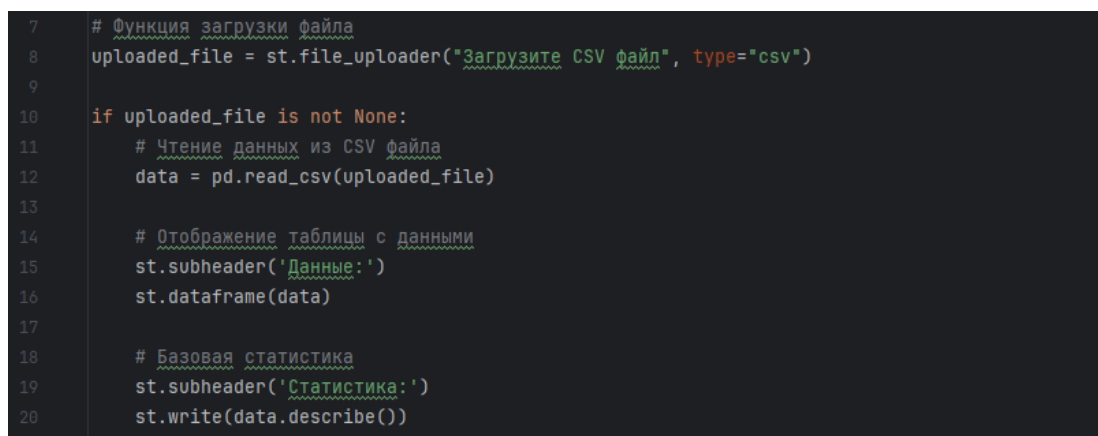
```
main.py x
1 import streamlit as st
2 import pandas as pd
3
4 # Заголовок приложения
5 st.title('Анализ данных из CSV файла')
6
```

Рис. 1. Импорт библиотек и задание заголовка

Для того чтобы приложение могло принимать данные от пользователя, реализована функция загрузки CSV файла. Она вызывается через компонент интерфейса `st.file_uploader()`, который отображает кнопку для выбора файла. Функция `file_uploader()` позволяет пользователю загрузить файл, который затем будет обработан программой. После того как файл загружен, приложение считывает его содержимое с помощью библиотеки `pandas`. Для наглядного отображения данных используется функция `st.dataframe()`. Этот метод выводит таблицу данных, которая интерактивно отображает содержимое загруженного CSV файла, позволяя пользователю прокручивать таблицу и взаимодействовать с ней.

После того как данные были успешно загружены и отображены, приложение вычисляет базовую статистику по загруженным данным. Функция `data.describe()` из `pandas` вычисляет основные статистические характеристики для всех числовых столбцов в наборе данных. Это полезно для быстрого анализа структуры данных. Результаты включают количество значений, среднее, стандартное отклонение, минимальные и максимальные значения, а также квартильные значения (25%, 50%, 75%).

Функция `st.write()` отображает результаты на странице приложения, а заголовок "Статистика:" помогает пользователю четко понимать, где начинается аналитическая часть данных (рис. 2).



```
7 # Функция загрузки файла
8 uploaded_file = st.file_uploader("Загрузите CSV файл", type="csv")
9
10 if uploaded_file is not None:
11     # Чтение данных из CSV файла
12     data = pd.read_csv(uploaded_file)
13
14     # Отображение таблицы с данными
15     st.subheader('Данные:')
16     st.dataframe(data)
17
18     # Базовая статистика
19     st.subheader('Статистика:')
20     st.write(data.describe())
21
```

Рис. 2. Работа с данными

На следующем этапе пользователю предлагается выбрать столбец для построения графика, после чего данные будут визуализированы в виде линейного графика. Функция `st.selectbox()` позволяет выбрать один из

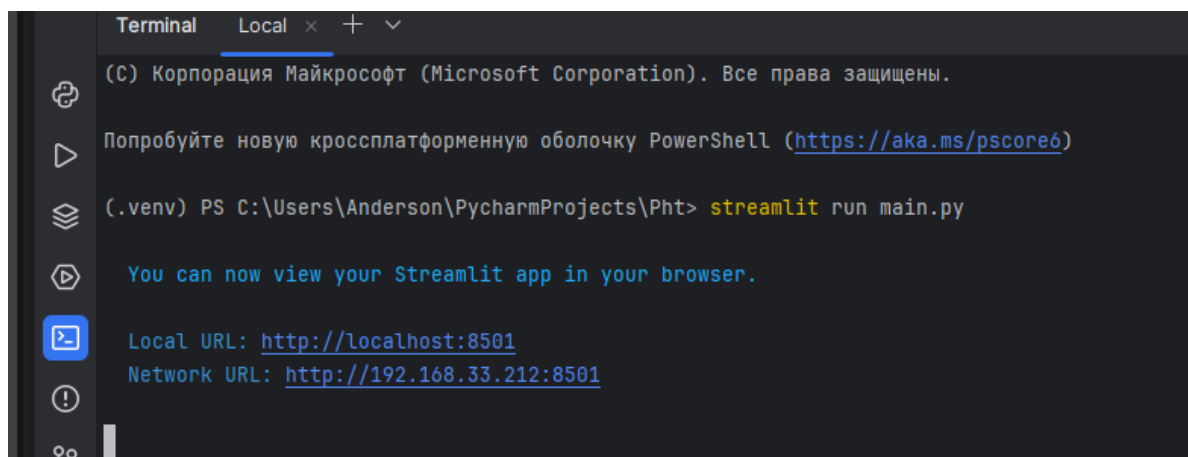
столбцов загруженных данных для дальнейшего построения графика. Все доступные столбцы выводятся в выпадающем списке, а пользователь выбирает нужный для анализа. После выбора столбца, приложение автоматически строит линейный график для выбранного столбца данных и отображает его на странице. Дополнительно выводится информация о типе данных выбранного столбца.

Кроме того, программа выводит тип данных выбранного столбца с помощью конструкции `data[column].dtype`, которая указывает, какие типы данных (например, `int64`, `float64` или `object`) присутствуют в выбранном столбце. Это может быть полезным для понимания структуры данных и их дальнейшей обработки. Если файл не загружен, приложение выводит сообщение, информирующее пользователя о необходимости загрузить файл для начала анализа (рис. 3).

```
22     # Выбор столбца для построения графика
23     st.subheader('График:')
24     column = st.selectbox('Выберите столбец для построения графика', data.columns)
25
26     # Построение графика
27     st.line_chart(data[column])
28
29     # Дополнительная настройка: отображение типа столбца
30     st.write(f'Тип данных выбранного столбца: {data[column].dtype}')
31 else:
32     st.write('Загрузите файл для начала анализа.')
33
```

Рис. 3. Вывод данных

Тестируем приложение. Для его запуска необходимо в терминал вписать команду «`streamlit run 'название скрипта'`». После выполнения команды Streamlit автоматически запускает локальный веб-сервер, который будет обрабатывать запросы к приложению. В терминале отобразится ссылка на адрес сервера, по которой можно открыть веб-интерфейс программы в браузере (рис. 4).



```
Terminal Local x + v
(C) Корпорация Майкрософт (Microsoft Corporation). Все права защищены.
Попробуйте новую кроссплатформенную оболочку PowerShell (https://aka.ms/pscore6)
(.venv) PS C:\Users\Anderson\PycharmProjects\Pht> streamlit run main.py
You can now view your Streamlit app in your browser.
Local URL: http://localhost:8501
Network URL: http://192.168.33.212:8501
```

Рис. 4. Запуск локального веб-сервера

На экране виден заголовок страницы "Анализ данных из CSV файла", а также кнопка для загрузки файла. Пользователь может загрузить CSV файл, который будет анализироваться в реальном времени. После загрузки файла программа читает данные с помощью pandas, проводит базовый анализ, отображает статистику и строит графики по выбранным пользователем столбцам. Благодаря встроенной функции динамического обновления, графики и данные автоматически изменяются в зависимости от действий пользователя, что делает анализ данных наглядным (рис. 5, 6).

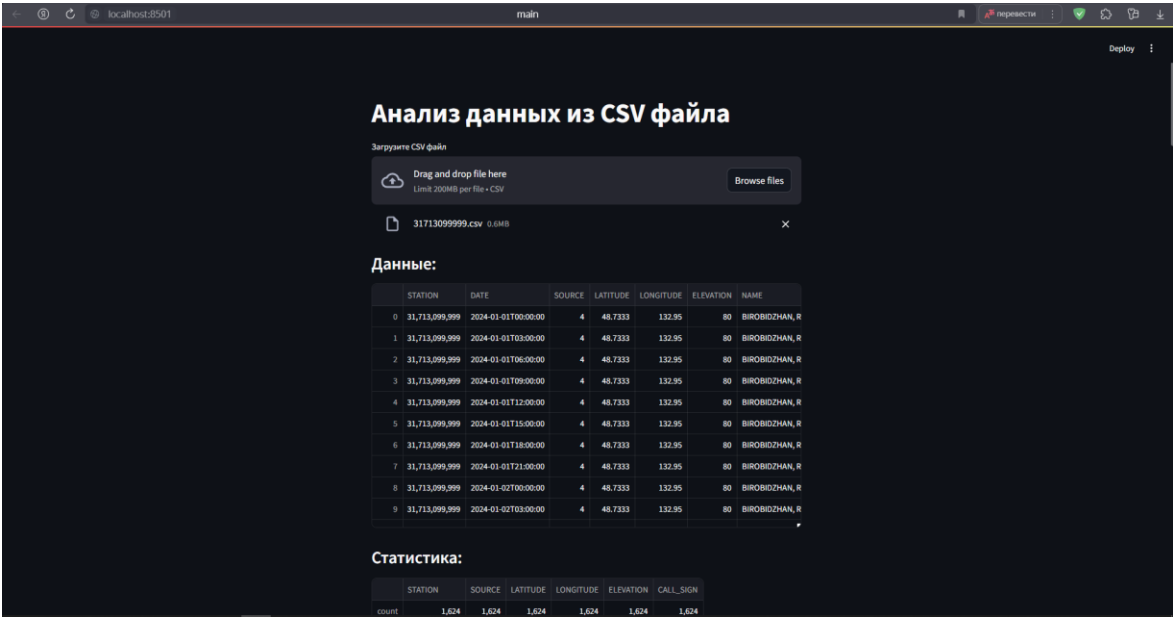


Рис. 5. Результат программы

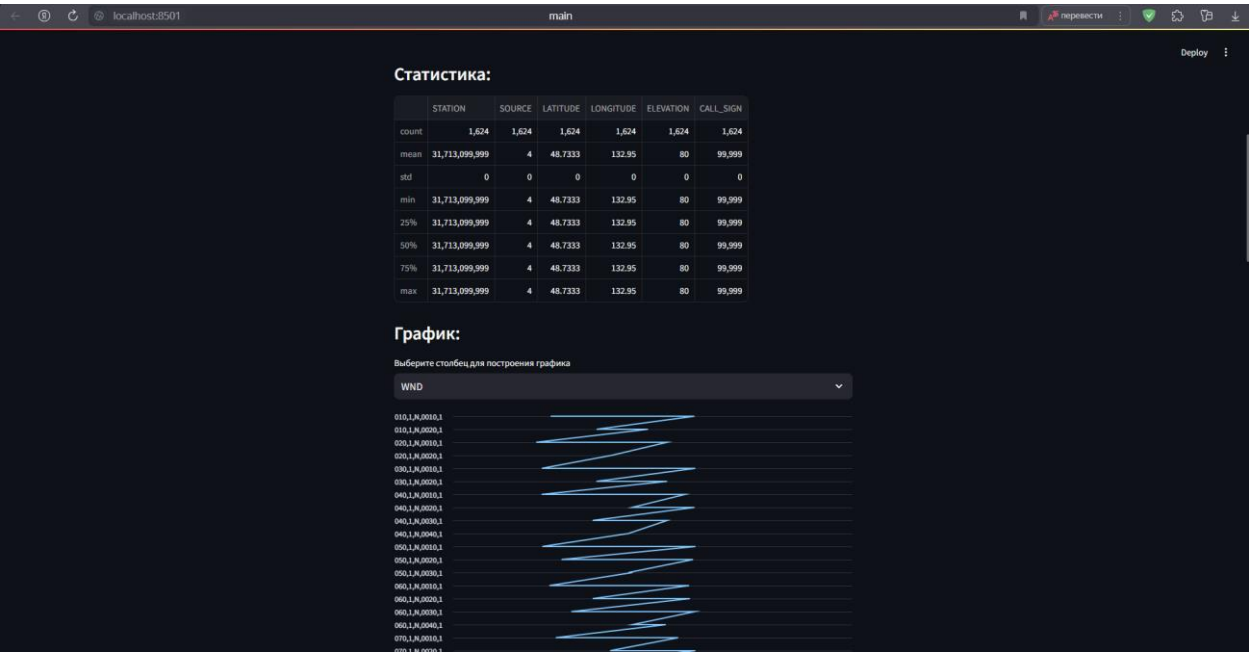


Рис. 6. Результат программы

Streamlit community cloud представляет собой облачную платформу, которая позволяет разработчикам быстро и удобно разворачивать свои

Streamlit-приложения. Основное преимущество платформы заключается в её простоте: для запуска приложения достаточно иметь аккаунт GitHub и загрузить скрипт с кодом. Важной особенностью Streamlit community cloud является автоматическое создание серверного окружения. Это означает, что разработчику не нужно вручную настраивать инфраструктуру или управлять зависимостями. Платформа предоставляет уникальную ссылку для каждого развернутого приложения, что делает его доступным для широкого круга пользователей.

Для развёртывания приложения использовался аккаунт GitHub. На первом этапе создавался репозиторий, содержащий файл с кодом приложения `main.py`. Данный скрипт включал основную логику работы приложения, а также элементы интерфейса, такие как загрузка файла и визуализация данных. На рисунке 7 представлен интерфейс созданного репозитория с загруженным скриптом.

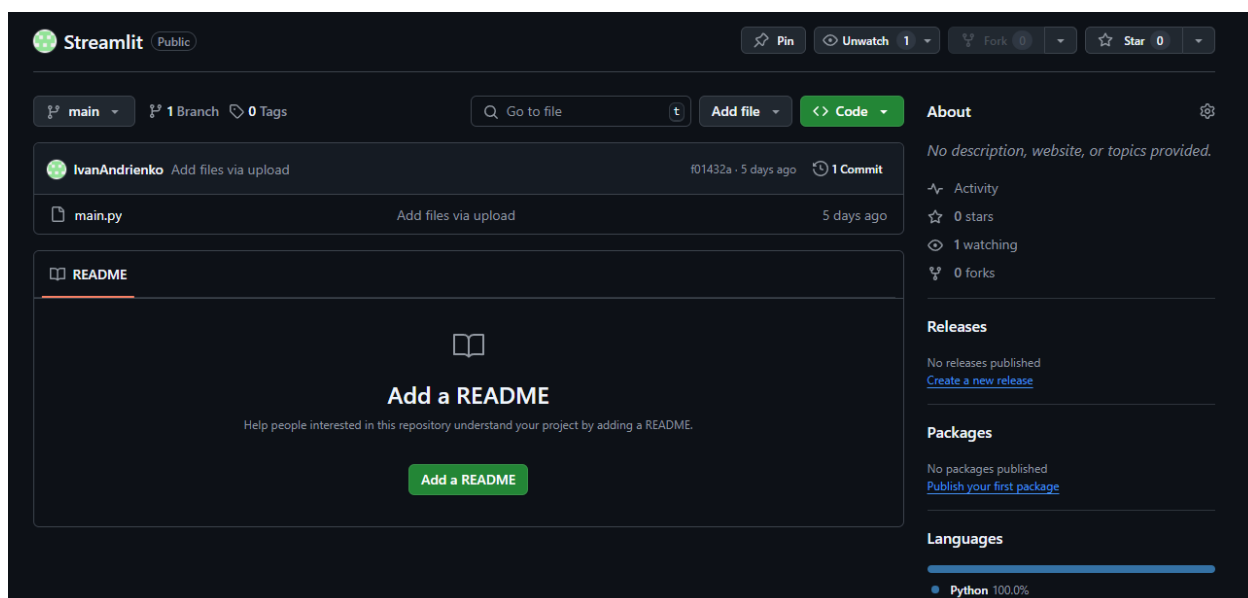


Рис. 7. Загруженный файл на github

После подготовки репозитория необходимо авторизоваться на платформе Streamlit Community Cloud через учётную запись GitHub. Это обеспечивало синхронизацию между локальным кодом и облачным сервисом.

Для развёртывания приложения на главной странице платформы Streamlit создаем новое приложение. Далее выбирается репозиторий, содержащий файл `main.py`. На этом этапе указывалась ветка репозитория и путь к скрипту приложения (рис.8).

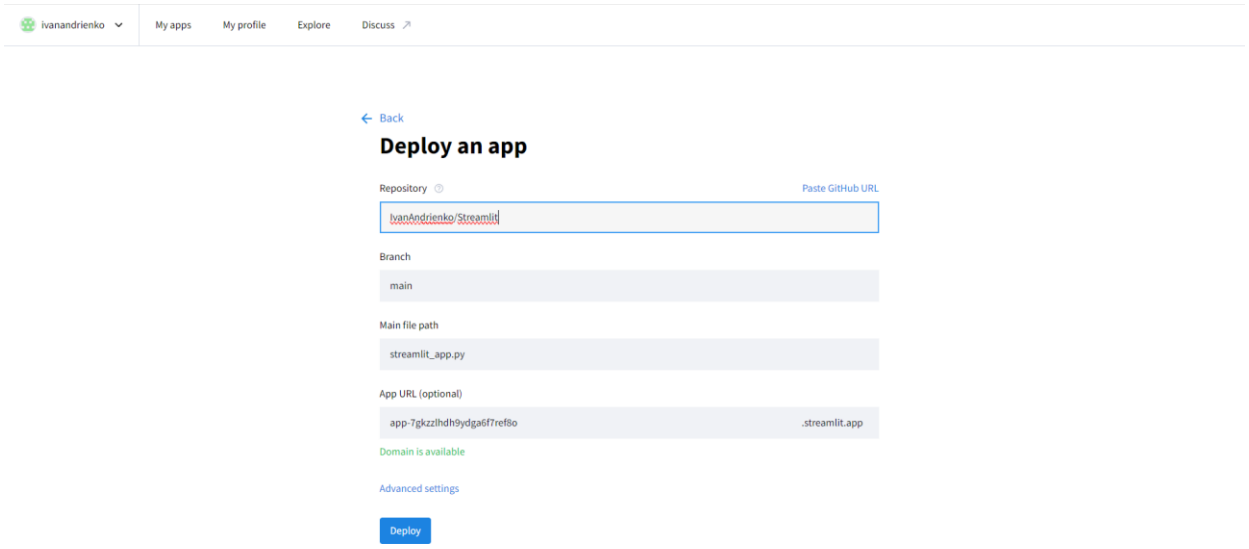


Рис. 8. Выбор репозитория

После завершения настройки происходит автоматическое развёртывание приложения. Streamlit Community Cloud самостоятельно обрабатывал запросы, создавал серверное окружение и предоставлял уникальную ссылку для доступа к приложению (рис. 9).

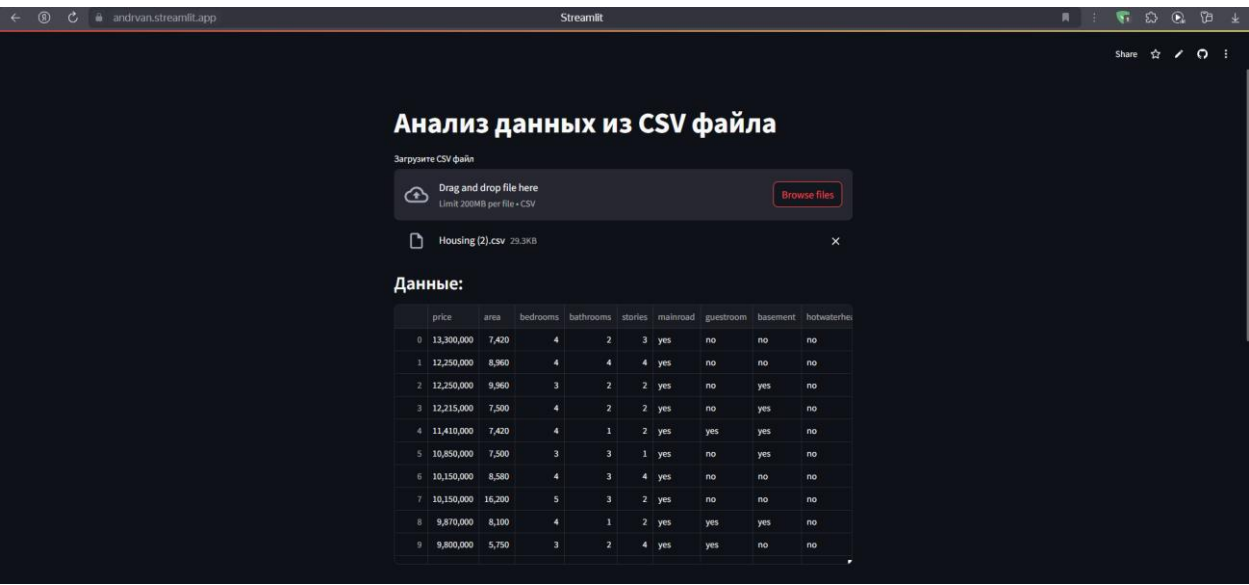


Рис. 9. Размещенное приложение

Приложение успешно функционирует на платформе, предоставляя пользователям возможность загружать CSV-файлы, анализировать данные и визуализировать результаты. Помимо этого, Streamlit Community Cloud позволяет разработчикам отслеживать статистику посещений приложения, что предоставляет важную информацию о количестве пользователей и частоте их взаимодействия с приложением. Внесение изменений в код выполняется через обновление репозитория на GitHub, после чего платформа автоматически обновляла развернутое приложение, что облегчало процесс

поддержки и дальнейшей доработки. Приложение доступно по ссылке <https://andrvan.streamlit.app/> [8].

4 Выводы

В данной работе представлен процесс создания и развёртывания интерактивного веб-приложения для анализа данных с использованием библиотеки Streamlit и платформы Streamlit Community Cloud. Рассмотрены основные этапы разработки, включая реализацию пользовательского интерфейса, визуализацию данных и интеграцию с облачным хостингом. Приложение успешно функционирует, предоставляя пользователям возможность загружать данные, анализировать их и визуализировать результаты. Streamlit Community Cloud продемонстрировала простоту использования и удобство развёртывания, позволяя избежать сложностей, связанных с настройкой серверной инфраструктуры. Кроме того, возможность отслеживания статистики посещений предоставляет разработчикам ценные данные о взаимодействии пользователей с приложением.

Библиографический список

1. Абдулвелеева Р.Р., Иванин Е.М. Использование библиотеки быстрой сборки и развёртывания web-приложения Streamlit в качестве визуализации данных и составления дашбордов // Актуальные проблемы современной науки, техники и образования. Тезисы 80-й международной научно-технической конференции. 2022. С. 340.
2. Конова П.С. Библиотека Streamlit как инструмент обработки и визуализации больших данных // Столыпинский вестник. 2022. Т. 4. № 3.
3. Демичев В.В., Алехина Е.А., Дашиева Б.Ш. Статистический анализ урожайности кормовых культур средствами автоматизированных модулей и библиотеки Streamlit // Российский экономический интернет-журнал. 2022. №4.
4. Кузнецова О.Ю., Кузнецов Р.Н., Кузьмин А.В. Особенности реализации модели машинного обучения для прогнозирования послеоперационных осложнений с использованием библиотеки Streamlit // Модели, системы, сети в экономике, технике, природе и обществе. 2023. № 3 (47). С. 167-176.
5. Хамимдова Д.Н. Развёртывание модели машинного обучения с помощью Streamlit // Евразийский Союз Ученых. Серия: технические и физико-математические науки. 2023. № 6 (109). С. 15-20.
6. Keerthi Ms.P., Reddy G.S., Raghava V.S., Reddy K.B. Streamlit Interface for Multiple Disease Diagnosis // International Journal for Research in Applied Science and Engineering Technology. 2023. Т. 11. № 2. С. 1159-1164.
7. Akkem Ya., Kumar B.S., Varanasi A. Streamlit Application for Advanced Ensemble Learning Methods in Crop Recommendation Systems – A Review and Implementation // Indian Journal of Science and Technology. 2023. Т. 16. № 48. С. 4688-4702.
8. URL: <https://andrvan.streamlit.app/> (дата обращения 17.12.2024).