

Создание интерактивного веб-приложения для анализа данных с использованием Streamlit

Андрюенко Иван Сергеевич

Приамурский государственный университет имени Шолом-Алейхема

Студент

Аннотация

В данной статье описан процесс создания веб-приложения для анализа данных из CSV-файлов с использованием библиотеки Streamlit на языке программирования Python. Рассмотрены ключевые аспекты создания интерактивных элементов, таких как загрузка файлов, визуализация данных и построение графиков. В качестве примера приведена реализация приложения для работы с табличными данными, которое позволяет загружать CSV-файлы, отображать содержимое, выполнять базовый анализ и строить графики на основе выбранных столбцов.

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

Creating an interactive web application for data analysis using Streamlet

Andrienko Ivan Sergeevich

Sholom-Aleichem Priamursky State University

Student

Abstract

This article describes the process of creating a web application for analyzing data from CSV files using the Streamlit library in the Python programming language. The key aspects of creating interactive elements, such as file downloads, data visualization and plotting, are considered. As an example, the implementation of an application for working with tabular data is shown, which allows you to upload CSV files, display content, perform basic analysis and build graphs based on selected columns.

Keywords: web application, Streamlet, Python, data analysis, data visualization.

1 Введение

1.1 Актуальность

В современном мире быстрый анализ данных является одной из ключевых задач в различных областях науки и бизнеса. Для упрощения процесса анализа данных были разработаны многочисленные инструменты, одним из которых является библиотека Streamlit для Python. Streamlit позволяет легко создавать интерактивные веб-приложения для работы с данными, минимизируя время разработки. В этой статье рассматривается

разработка приложения для анализа данных, основанного на загрузке CSV-файлов и визуализации данных.

1.2 Обзор исследований

В исследовании Р.Р. Абдулвелеева и Е.М. Иванина рассматриваются особенности быстрого развертывания 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. Хамимдова 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].

1.3 Цель исследования

Цель исследования – разработать интерактивное приложение для анализа данных из CSV-файлов с использованием библиотеки Streamlit. Приложение должно позволять пользователям загружать данные, просматривать их, выполнять базовый статистический анализ и строить графики.

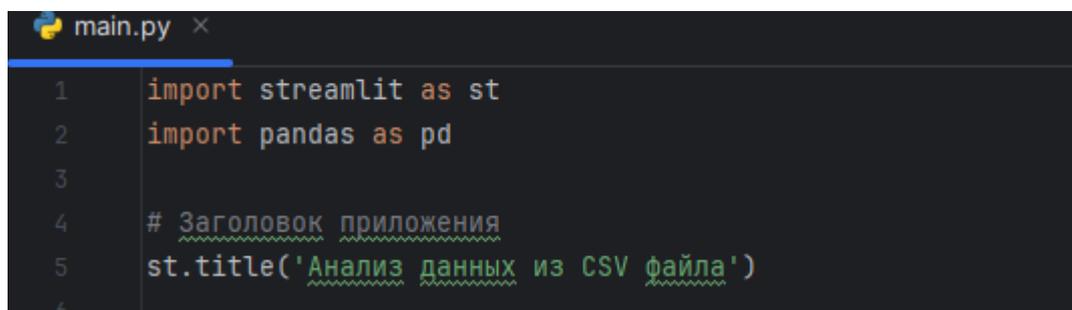
2 Материалы и методы

Для реализации программы использовались Python и библиотека Streamlit. Приложение включает ключевые элементы как загрузка файла, отображение данных, статистический анализ и визуализация.

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')
```

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 квартальные 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())
```

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.

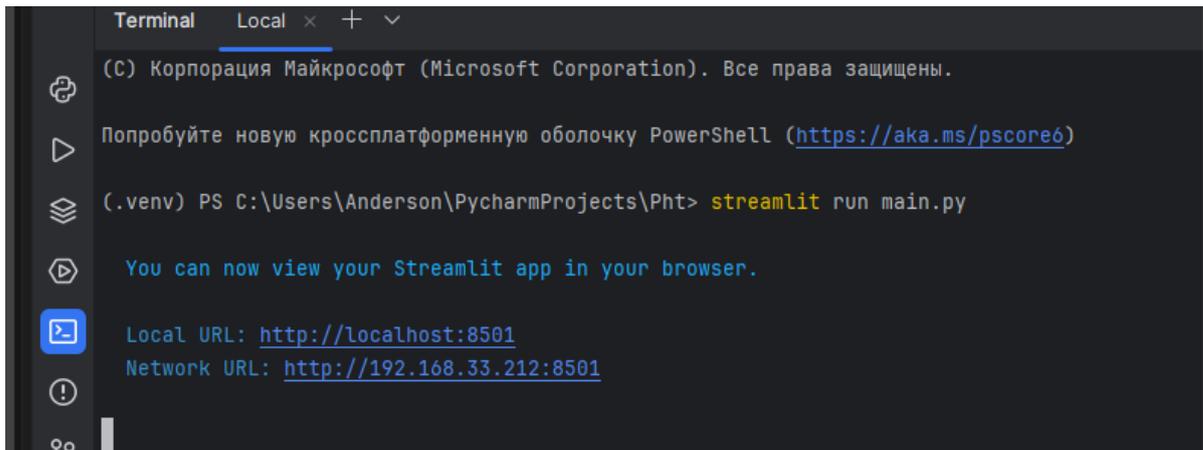
Кроме того, программа выводит тип данных выбранного столбца с помощью конструкции `data[column].dtype`, которая указывает, какие типы данных (например, `int64`, `float64` или `object`) присутствуют в выбранном столбце. Это может быть полезным для понимания структуры данных и их дальнейшей обработки. 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

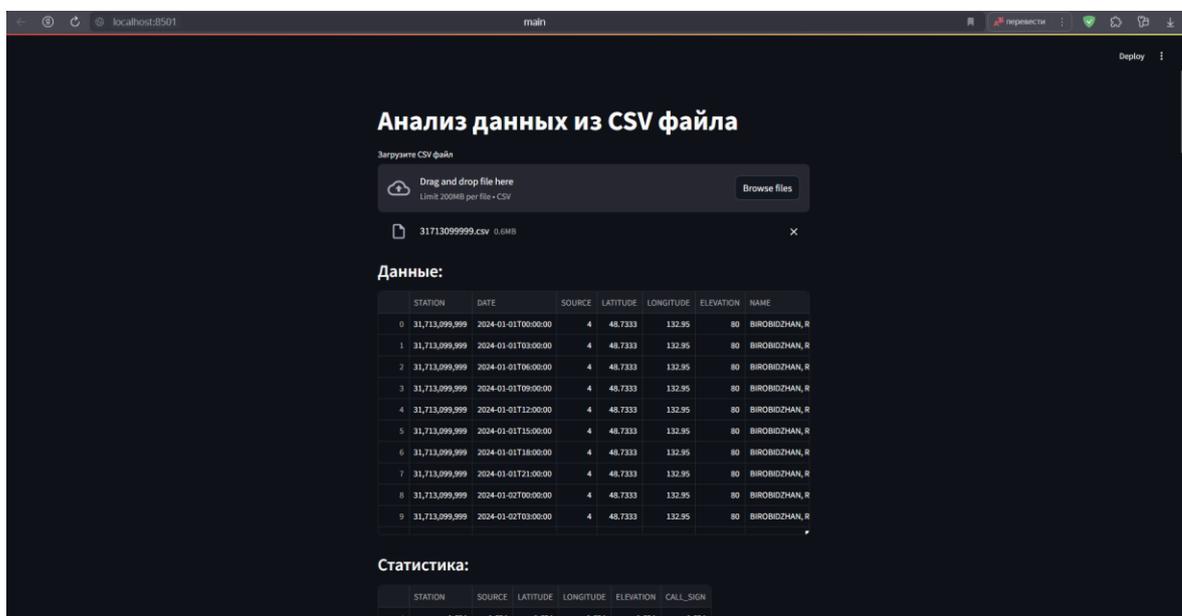
на адрес сервера, по которой можно открыть веб-интерфейс программы в браузере (рис. 4).



```
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).



Анализ данных из CSV файла

Загрузите CSV файл

Drag and drop file here
Limit 200MB per file • CSV

Browse files

31713099999.csv 0.6MB

Данные:

	STATION	DATE	SOURCE	LATITUDE	LONGITUDE	ELEVATION	NAME
0	31,713,099,999	2024-01-01T00:00:00	4	48.7333	132.95	80	БИРОВАДЖАН, R
1	31,713,099,999	2024-01-01T03:00:00	4	48.7333	132.95	80	БИРОВАДЖАН, R
2	31,713,099,999	2024-01-01T06:00:00	4	48.7333	132.95	80	БИРОВАДЖАН, R
3	31,713,099,999	2024-01-01T09:00:00	4	48.7333	132.95	80	БИРОВАДЖАН, R
4	31,713,099,999	2024-01-01T12:00:00	4	48.7333	132.95	80	БИРОВАДЖАН, R
5	31,713,099,999	2024-01-01T15:00:00	4	48.7333	132.95	80	БИРОВАДЖАН, R
6	31,713,099,999	2024-01-01T18:00:00	4	48.7333	132.95	80	БИРОВАДЖАН, R
7	31,713,099,999	2024-01-01T21:00:00	4	48.7333	132.95	80	БИРОВАДЖАН, R
8	31,713,099,999	2024-01-02T00:00:00	4	48.7333	132.95	80	БИРОВАДЖАН, R
9	31,713,099,999	2024-01-02T03:00:00	4	48.7333	132.95	80	БИРОВАДЖАН, R

Статистика:

	STATION	SOURCE	LATITUDE	LONGITUDE	ELEVATION	CALL_SIGN
count	1,624	1,624	1,624	1,624	1,624	1,624

Рис. 6. Результат программы

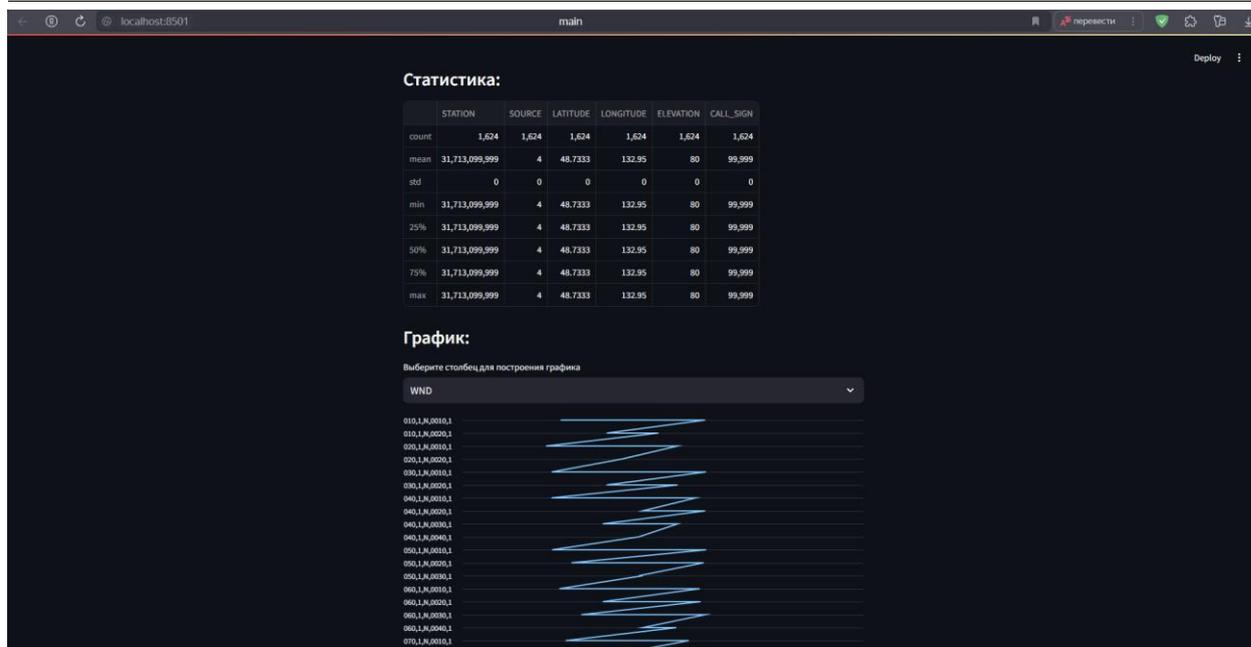


Рис. 6. Результат программы

Выводы

В данной статье описан процесс создания веб-приложения для анализа данных с использованием библиотеки Streamlit. Рассмотрены возможности использования интерактивных элементов для загрузки и визуализации данных, а также базового статистического анализа. Streamlit является удобным и гибким инструментом для создания приложений без необходимости глубоких знаний в области веб-разработки. Созданное приложение может быть использовано в образовательных и научных целях для быстрой визуализации данных.

Библиографический список

1. Абдулвелеева Р.Р., Иванин Е.М. Использование библиотеки быстрой сборки и развертывания web-приложения Streamlit в качестве визуализации данных и составления дашбордов. В кн.: 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. Стольпинский 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.