**ДИПЛОМНАЯ РАБОТА**

**Сравнение различных библиотек для визуализации данных: Matplotlib, Seaborn и Plotly**

Выполнил студент потока:

Агабеков Рафик Шекербегович

2024

**СОДЕРЖАНИЕ**

[Введение………………………………………………………………………………... 3](#_30j0zll)

[1. Основные понятия и определения………………………………………………. 5](#_1fob9te)

[2. Методы и подходы к разработке………………………………………………… 8](#_3znysh7)

[2.1 Исследование методов визуализации данных……………………………….. 8](#_2et92p0)

[2.2 Подходы к сравнительному анализу библиотек……………………………. 10](#_tyjcwt)

[2.3 Выбор готового набора данных для анализа……………………………….. 12](#_3dy6vkm)

[3. Проектирование приложения……………………………………………………... 15](#_1t3h5sf)

[3.1 Планирование и анализ требований………………………………………….. 15](#_4d34og8)

[3.2 Основные требования…………………………………………………………. 16](#_2s8eyo1)

[3.3 Технические требования………………………………………………………. 17](#_17dp8vu)

[4. Разработка в соответствии с созданной документацией………………………... 18](#_3rdcrjn)

[5. Анализ и интерпретация результатов……………………………………………. 23](#_26in1rg)

[Заключение…………………………………………………………………………… 25](#_lnxbz9)

# Введение

Визуализация данных играет важнейшую роль в аналитике и принятии решений. С ростом объемов данных и сложностью их анализа эффективные инструменты визуализации становятся неотъемлемой частью работы аналитиков, разработчиков и исследователей. Сегодня существует множество библиотек для визуализации, каждая из которых имеет свои преимущества и ограничения. В рамках данной работы сравниваются Matplotlib, Seaborn и Plotly — три популярные библиотеки, широко применяемые для анализа данных.

В эпоху, когда принципы data-driven решений становятся основой бизнеса, специалисты, владеющие инструментами визуализации, высоко востребованы. Умение правильно выбрать и применять подходящие инструменты для представления данных может существенно повысить их интерпретируемость и ценность для конечного пользователя. Исследование и сравнение библиотек, таких как Matplotlib, Seaborn и Plotly, позволяет лучше понимать их сильные и слабые стороны, а также области применения.

Разработка приложений для визуализации данных и создание графиков являются важной частью современных исследовательских и аналитических проектов. Сравнение функциональности и удобства использования различных библиотек в реальных сценариях поможет определить их оптимальное применение. Это, в свою очередь, предоставит полезные рекомендации для аналитиков, инженеров данных и программистов.

Работа с инструментами визуализации данных позволяет развить технические и аналитические навыки, которые являются важными для построения карьеры в области анализа данных и разработки программного обеспечения. Выбор данной темы обусловлен желанием освоить современные инструменты анализа данных, получить опыт создания приложений с графическим интерфейсом и лучше понять ключевые аспекты работы с Python-библиотеками.

Таким образом, тема "Сравнение различных библиотек для визуализации данных: Matplotlib, Seaborn и Plotly" представляет актуальность как с точки зрения теоретических исследований, так и практического применения. Она отвечает требованиям рынка, имеет значительный прикладной потенциал и открывает перспективы для профессионального развития.

Целью данного исследования является сравнение функциональных возможностей, удобства использования и области применения библиотек Matplotlib, Seaborn и Plotly на примере создания набора визуализаций, разработанных в едином приложении.

Задачи исследования:

1. Провести обзор современных библиотек Python для визуализации данных, включая Matplotlib, Seaborn и Plotly.
2. Изучить особенности, преимущества и ограничения каждой библиотеки.
3. Реализовать набор визуализаций с использованием всех трех библиотек, ориентируясь на данные Titanic.
4. Разработать графический интерфейс приложения для демонстрации результатов сравнения.
5. Провести анализ удобства использования, функциональных возможностей и производительности библиотек.
6. Сформулировать выводы и рекомендации по выбору инструментов визуализации данных в зависимости от задач и контекста.

Цели и задачи исследования направлены на формирование практического понимания работы с инструментами визуализации, их сравнительную оценку и разработку полезного программного продукта.

# Основные понятия и определения

Обзор основных понятий:

1. Библиотека визуализации данных: Набор инструментов и функций для создания графиков, диаграмм и других видов визуализаций данных. В данном контексте анализируются библиотеки Matplotlib, Seaborn и Plotly, каждая из которых обладает уникальными возможностями для отображения информации.
2. График (Plot): Визуальное представление данных, позволяющее анализировать зависимости, распределение или тренды. Графики могут быть статичными или интерактивными, в зависимости от используемой библиотеки и инструментов.
3. Матplotlib: Популярная библиотека для создания статичных графиков в Python. Она предоставляет широкие возможности для кастомизации и оформления графиков, включая поддержку различных типов графиков, таких как линейные, гистограммы, диаграммы рассеяния и т. д.
4. Seaborn: Библиотека для статистической визуализации, построенная на основе Matplotlib. Seaborn упрощает создание сложных статистических графиков, таких как тепловые карты, графики распределения и парные графики. Seaborn также предоставляет улучшенную эстетическую стилизацию по сравнению с Matplotlib.
5. Plotly: Библиотека для создания интерактивных графиков и дашбордов. Plotly позволяет пользователю взаимодействовать с графиками, что делает ее удобной для создания веб-приложений и представления данных в реальном времени.
6. Интерактивность: Способность пользователя взаимодействовать с графиками, например, изменять масштабы, фильтровать данные или наводить курсор для получения подробной информации. Интерактивные графики полезны для более глубокого анализа данных, особенно в больших и сложных наборах.
7. Тепловая карта (Heatmap): Тип графика, который используется для отображения матриц данных, где значения представлены цветами. Тепловые карты часто применяются для визуализации корреляций или других числовых взаимосвязей между переменными.
8. Гистограмма: Вид графика, который используется для отображения распределения числовых данных. Она делит данные на интервалы (или «корзины») и отображает количество наблюдений, попавших в каждый интервал.
9. Диаграмма рассеяния (Scatter plot): График, который используется для отображения точек данных по двум числовым переменным. Он помогает визуализировать взаимосвязь между переменными.
10. Корреляция (Correlation): Статистическая мера, которая указывает на степень и направление взаимосвязи между двумя переменными. Визуализация корреляции с помощью тепловой карты или диаграммы рассеяния помогает выявить зависимости между переменными.
11. Частотное распределение: Статистический метод, который показывает, как часто различные значения переменной встречаются в данных. Гистограммы и другие типы графиков могут использоваться для представления частотных распределений.
12. Визуализация данных: Процесс создания графиков, диаграмм и других видов представлений данных, который помогает исследователям и аналитикам лучше понять информацию и выявить тренды, паттерны и аномалии в данных.
13. Дашборд (Dashboard): Визуальное представление ключевых показателей и данных в виде нескольких связанных графиков и диаграмм, обычно используемое для мониторинга и анализа. Дашборды позволяют пользователю анализировать данные в реальном времени и принимать обоснованные решения.
14. Анализ данных: Процесс обработки, организации и визуализации данных для выявления значимых трендов, закономерностей и взаимосвязей. Визуализация данных является неотъемлемой частью этого процесса, позволяя пользователю наглядно интерпретировать результаты анализа.

Эти основные понятия и определения являются важными для понимания технологий и инструментов, используемых в процессе разработки и сравнения библиотек визуализации данных, а также для правильного применения их в различных типах проектов и исследований.

# Методы и подходы к разработке

## Исследование методов визуализации данных

Визуализация данных является важным инструментом анализа информации, позволяя преобразовать сложные числовые и текстовые данные в понятные и наглядные графические формы. Этот метод широко используется в различных областях, включая науку, бизнес-аналитику, образование и технологии.

Основная цель визуализации данных — выявление закономерностей, трендов и аномалий, которые могут быть упущены при анализе данных в табличной или текстовой форме. Графики, диаграммы и другие виды визуальных представлений помогают более эффективно воспринимать данные, способствуют коммуникации результатов и повышают точность принятия решений.

Существует множество методов визуализации, которые могут быть условно разделены на следующие категории:

1. Гистограммы и диаграммы распределения

* Используются для анализа распределения данных. Например, гистограммы позволяют визуализировать частоту значений в заданных интервалах.

1. Диаграммы рассеяния (Scatter Plots)

Применяются для изучения взаимосвязей между двумя переменными, предоставляя возможность оценить корреляцию или выявить зависимости.

1. Тепловые карты (Heatmaps)

* Позволяют визуализировать корреляционные матрицы или другие матрицы данных, где цветовое кодирование используется для отображения величины значений.

1. Интерактивные визуализации

* Эти методы позволяют пользователю взаимодействовать с данными в реальном времени, например, наводить курсор для получения дополнительных деталей или изменять параметры отображения.

Современные инструменты для построения графиков предоставляют разнообразные возможности для создания наглядных визуализаций. Рассмотрим три популярных библиотеки Python:

1. Matplotlib

* Эта библиотека является стандартом де-факто для построения статических графиков. Она предоставляет гибкий интерфейс для создания разнообразных типов графиков, от простых линейных диаграмм до сложных многослойных композиций. Однако, Matplotlib требует больше кода для настройки графиков, что делает её менее удобной в использовании для новичков.

1. Seaborn

* Построенная на основе Matplotlib, эта библиотека обеспечивает более высокоуровневый интерфейс для построения графиков. Она включает готовые стили и функции для работы с категорическими данными, построения тепловых карт и создания эстетически привлекательных графиков.

1. Plotly

* Инструмент, ориентированный на интерактивные визуализации. Он предоставляет возможности для создания динамических графиков, которые можно просматривать в браузере, масштабировать, фильтровать и анализировать в реальном времени. Plotly широко используется в веб-приложениях и для демонстрации данных в презентациях.

В рамках данного проекта визуализация данных используется для анализа набора данных Titanic, который содержит информацию о пассажирах корабля. Применение различных методов визуализации данных, таких как гистограммы возрастов, тепловые карты корреляций и интерактивные графики, позволяет исследовать зависимости и закономерности в данных, а также сравнить функциональные возможности библиотек Matplotlib, Seaborn и Plotly.

Таким образом, исследование методов визуализации данных подтверждает их значимость для анализа информации и служит теоретической основой для проведения экспериментов и сравнительного анализа инструментов в рамках проекта.

## Подходы к сравнительному анализу библиотек

Сравнительный анализ библиотек для визуализации данных — это процесс систематического изучения их возможностей, ограничений и эффективности для различных задач. В рамках проекта рассматриваются библиотеки Matplotlib, Seaborn и Plotly. Анализ проводится с использованием набора критериев, которые позволяют оценить функциональность, удобство использования и применимость каждой из библиотек.

Для сравнения библиотек применяются следующие ключевые критерии:

1. Функциональность:

* Оценка доступных типов графиков (гистограммы, линейные графики, тепловые карты, интерактивные диаграммы и т.д.), поддерживаемых функций и возможностей настройки.

1. Эстетичность и качество графиков:

* Насколько графики, созданные с использованием библиотеки, выглядят профессионально и привлекательно без значительных дополнительных настроек.

1. Интерактивность:

* Возможность взаимодействовать с графиками в реальном времени (увеличение, фильтрация, добавление аннотаций).

1. Простота и удобство использования:

* Оценивается сложность освоения библиотеки, объем кода, необходимый для построения базовых и сложных графиков, а также наличие удобного интерфейса для пользователя.

1. Производительность:

* Время, необходимое для построения графиков, особенно при работе с большими объемами данных.

1. Сообщество и документация:

* Наличие активного сообщества разработчиков, подробной документации, примеров и шаблонов.

Для проведения сравнительного анализа реализуются одинаковые визуализации, используя каждую из библиотек. Это позволяет наглядно продемонстрировать их возможности и подходы к решению одинаковых задач.

1. Сравнительные визуализации:

* Построение гистограммы распределения возрастов пассажиров Titanic по классам.
* Построение тепловой карты корреляций между числовыми признаками.
* Построение диаграммы рассеяния с осями "Возраст" и "Стоимость билета".

1. Оценка сложности кода:

* Для каждой библиотеки анализируется объем и структура кода, необходимого для создания указанных визуализаций. Простота и интуитивность синтаксиса учитываются при выставлении оценки.

1. Визуальное сравнение графиков:

* Графики сравниваются по качеству оформления, наличию стандартных стилей и уровню детализации, предоставляемому каждой библиотекой.

1. Анализ интерактивности:

* Для Plotly проверяется возможность изменения масштаба графиков, добавления динамических аннотаций и использования фильтров.
* Для Matplotlib и Seaborn анализируется поддержка интерактивных возможностей через сторонние инструменты или дополнительные модули.

Обоснование выбора подходов:

* Тестирование на идентичном наборе данных: Использование одинакового набора данных (например, Titanic) позволяет исключить влияние данных на результаты анализа и сосредоточиться на возможностях библиотек.
* Учет современных требований: В анализ включаются такие критерии, как интерактивность и производительность, которые особенно важны в современных приложениях, работающих с большими объемами данных.
* Сравнение типовых задач: Выбор задач визуализации основан на их популярности в аналитике данных, что делает анализ релевантным для реальных сценариев.

Сравнительный анализ библиотек предоставляет полезные выводы, которые помогут выбрать наиболее подходящий инструмент для конкретной задачи:

* Matplotlib оптимален для базовых визуализаций, где требуется полный контроль над графиками.
* Seaborn хорошо подходит для создания сложных графиков с минимальными усилиями благодаря встроенным функциям и эстетичному стилю.
* Plotly является лидером в интерактивности и подходит для веб-приложений и презентаций.

Применение подходов к сравнительному анализу в рамках проекта позволит объективно оценить сильные и слабые стороны каждой библиотеки, сформулировать рекомендации для их использования и продемонстрировать полученные результаты.

## Выбор готового набора данных для анализа

Набор данных Titanic, основанный на информации о пассажирах легендарного лайнера, является одним из самых популярных для анализа и обучения. Его структура и содержимое позволяют изучать широкий спектр методов визуализации и аналитики. Этот набор включает данные о возрасте, стоимости билетов, классе обслуживания, поле и других характеристиках пассажиров. Такие данные не только удобны для обработки, но и дают возможность выявлять интересные закономерности, например, исследовать влияние социального статуса или возраста на выживаемость.

Одной из главных причин выбора набора Titanic для данного проекта является его доступность и историческая значимость. Это данные, которые тесно связаны с реальными событиями, что делает работу более увлекательной и понятной даже для тех, кто только начинает осваивать аналитические инструменты. Titanic уже предварительно подготовлен для анализа, однако требует выполнения важных шагов, таких как обработка пропущенных значений и преобразование категориальных данных, что полезно для развития практических навыков.

Набор данных Titanic состоит из 891 записи, каждая из которых представляет одного пассажира. Основные столбцы включают:

* Survived — бинарный признак, указывающий, выжил ли пассажир (0 — нет, 1 — да).
* Pclass — класс пассажира (1-й, 2-й или 3-й).
* Name — имя пассажира.
* Sex — пол пассажира (мужской или женский).
* Age — возраст пассажира.
* SibSp — количество братьев/сестёр или супругов на борту.
* Parch — количество родителей или детей на борту.
* Fare — стоимость билета.
* Embarked — порт посадки (C — Шербур, Q — Квинстаун, S — Саутгемптон).

Эта информация предоставляет множество возможностей для анализа, от изучения корреляций между признаками до построения сложных визуализаций.

Ещё одним преимуществом является возможность применения различных библиотек для визуализации. Данные Titanic можно использовать для построения гистограмм, тепловых карт, точечных графиков и других видов визуализации, что позволяет продемонстрировать возможности Matplotlib, Seaborn и Plotly.

Этот набор идеально подходит для сравнительного анализа методов визуализации. Его простота сочетается с разнообразием, что делает Titanic отличным выбором для демонстрации теоретических и практических подходов в рамках данного проекта.

# 3. Проектирование приложения

## 3.1 Планирование и анализ требований

На этапе планирования было проведено исследование доступных библиотек для визуализации данных. Основное внимание уделено Matplotlib, Seaborn и Plotly, поскольку эти инструменты наиболее популярны, функциональны и обладают широкими возможностями для графического представления информации.

Приложение будет представлять собой настольную программу с графическим интерфейсом пользователя (GUI), реализованным с использованием библиотеки Tkinter. Структура включает:

* Модуль для обработки и предварительного анализа данных (набор Titanic).
* Модуль для создания визуализаций с использованием выбранных библиотек.
* Интерактивный интерфейс для отображения графиков и сравнения их характеристик.

Создается минимально работоспособная версия приложения, включающая:

1. Загрузка данных Titanic и базовую обработку пропущенных значений.
2. Построение графиков для одной библиотеки.
3. Простейший пользовательский интерфейс для взаимодействия.

Реализация ключевого функционала:

1. Создание визуализаций: Построение различных типов графиков (гистограммы, точечные диаграммы, тепловые карты) для Matplotlib, Seaborn и Plotly.
2. Сравнение результатов: Включение элементов интерфейса для переключения между графиками и их сопоставления.
3. Увеличение интерактивности: Интеграция интерактивного графика Plotly в общий интерфейс.

Разработка удобного интерфейса для управления визуализацией данных, включая кнопки, текстовые метки и контейнеры для графиков. Интерфейс обеспечивает:

* Возможность переключения между визуализациями.
* Открытие интерактивных графиков Plotly в браузере.

Проводится тестирование приложения для оценки корректности работы визуализаций и интерфейса. Результаты тестирования учитываются для устранения недочетов и улучшения функционала.

Таким образом, структура и функциональность приложения определяются исходя из целей и задач исследования, что позволяет провести полноценное сравнение возможностей библиотек визуализации данных.

## 3.2 Основные требования

1. Загрузка данных: Приложение должно автоматически загружать и обрабатывать готовый набор данных Titanic, удаляя пропущенные значения в ключевых столбцах, таких как возраст и стоимость билета.
2. Построение графиков: Пользователь должен иметь возможность увидеть визуализацию данных в трех различных библиотеках: Matplotlib, Seaborn и Plotly.
3. Сравнение визуализаций: Интерфейс должен обеспечивать одновременное отображение всех графиков для удобного сравнения.
4. Интерактивные элементы: Приложение должно позволять пользователю открыть интерактивный график Plotly в браузере для более глубокого анализа.
5. Удобный интерфейс: Приложение должно иметь простой и интуитивно понятный графический интерфейс, доступный для пользователей без опыта программирования.

## 3.3 Технические требования

1. Язык программирования: Python.
2. Библиотеки визуализации:
3. Matplotlib: для создания базовых статичных графиков.
4. Seaborn: для построения сложных визуализаций с аналитическими элементами, например, тепловых карт.
5. Plotly: для построения интерактивных графиков.
6. Графический интерфейс: Использование библиотеки Tkinter для создания настольного приложения.
7. Обработка данных: Библиотека Pandas для работы с таблицами данных.
8. Файловая система: Приложение должно сохранять созданные графики в формате PNG в корневую папку проекта при первом запуске.
9. Кросс-платформенность: Приложение должно работать на операционных системах Windows, MacOS и Linux при наличии установленного Python и необходимых библиотек.
10. Использование ресурсов: Оптимизация приложения для работы на компьютерах с ограниченными ресурсами, с минимальными задержками при построении графиков.

# 4. Разработка в соответствии с созданной документацией

Разработка приложения для визуализации данных с использованием Matplotlib, Seaborn и Plotly была структурирована на несколько этапов: проектирование пользовательского интерфейса, реализация обработки данных, создание визуализаций и интеграция интерактивных функций. В качестве инструмента для планирования и отслеживания прогресса использовался Notion, что позволило структурировать задачи и эффективно управлять временем.

Интерфейс приложения был создан с использованием библиотеки Tkinter, что обеспечило простоту реализации и кросс-платформенность.

1. Созданы основные элементы графического интерфейса (рисунок 1):

* Заголовок приложения и кнопки управления.
* Контейнеры для размещения визуализаций и дополнительных функций.

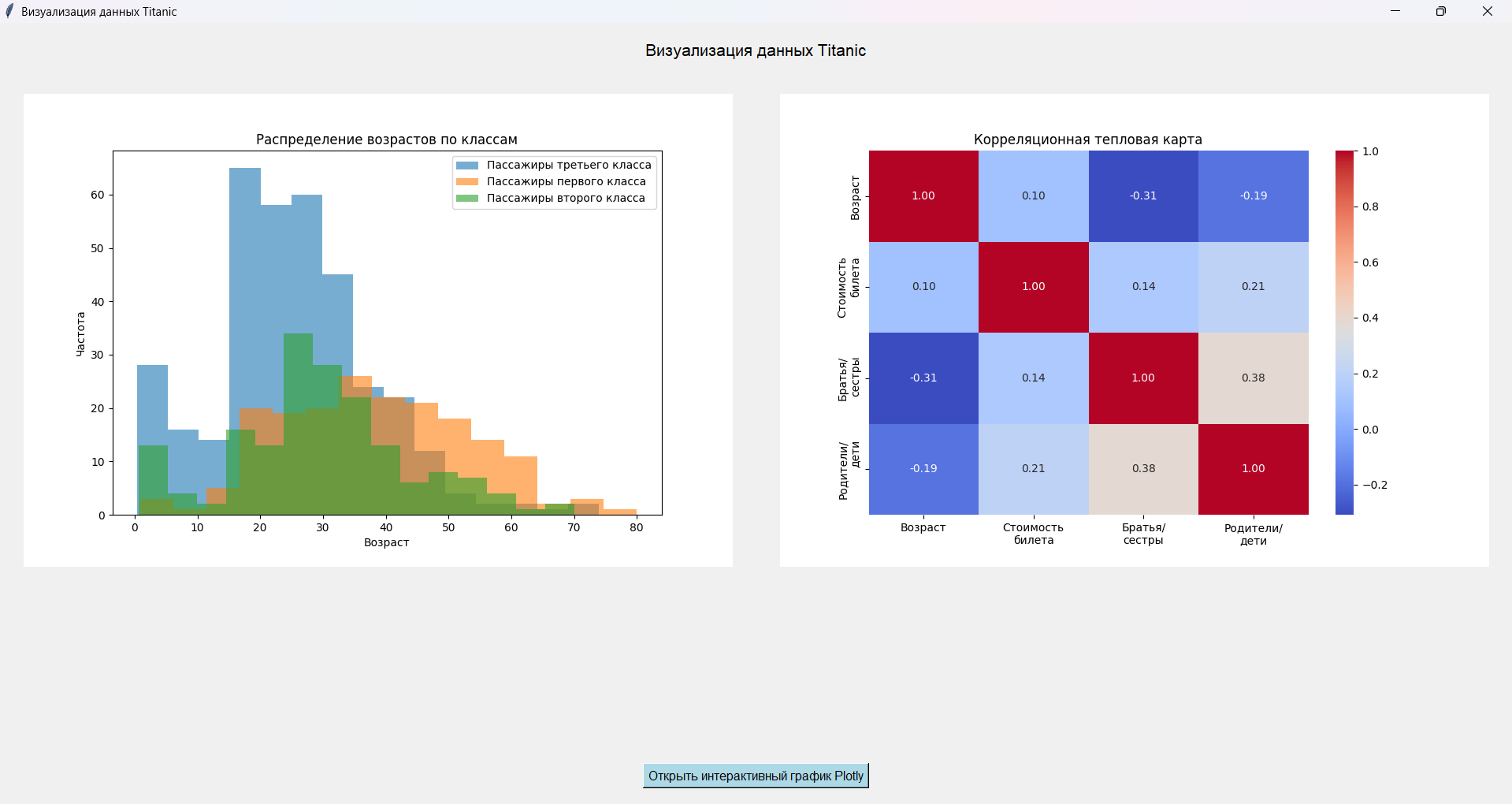


Рисунок 1 – Главная страница системы

1. Реализована кнопка для открытия интерактивного графика Plotly в веб-браузере (рисунок 2).

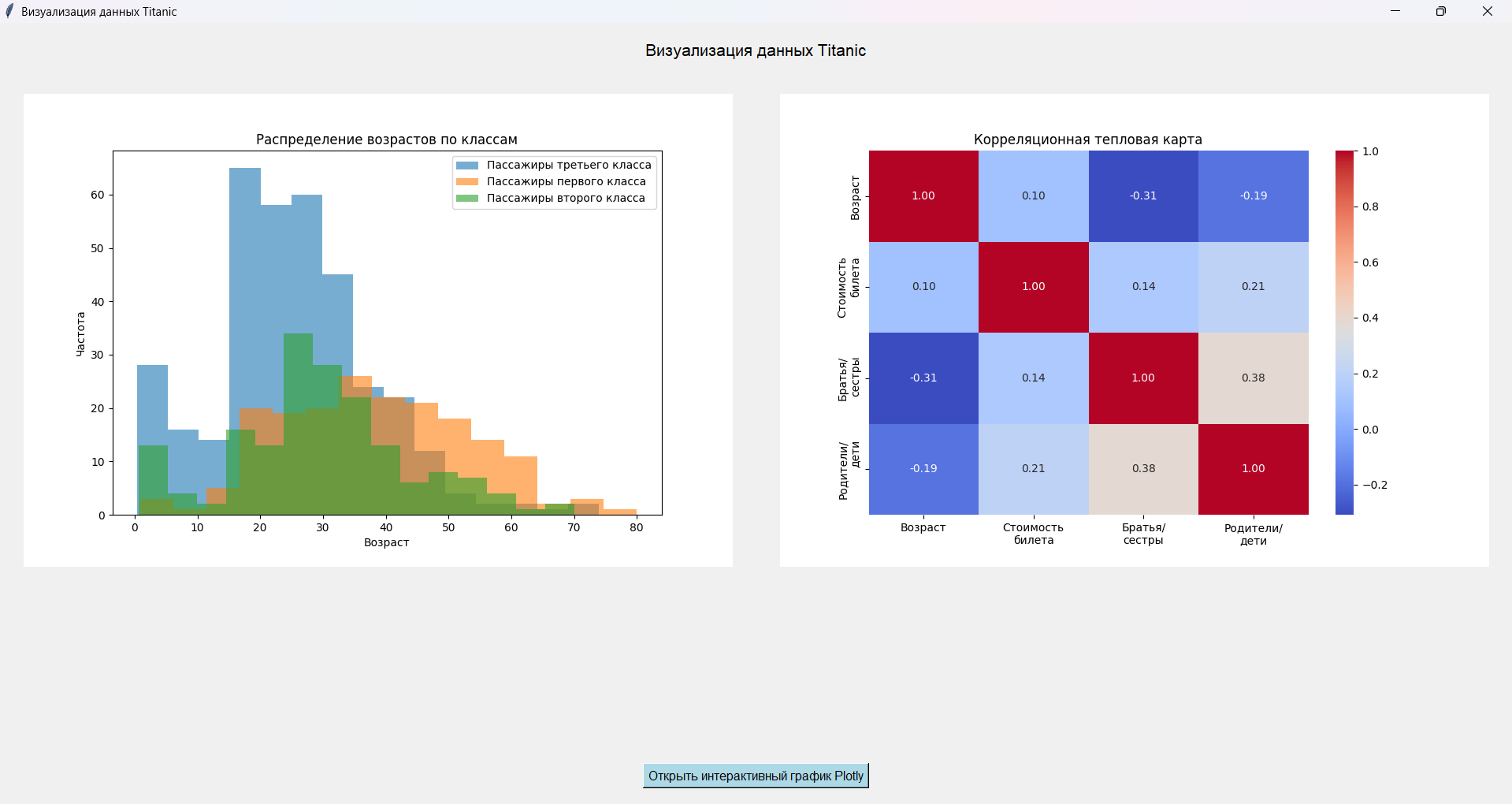


Рисунок 2 – Кнопка для открытия графика

1. Добавлены элементы управления, обеспечивающие доступность визуализаций для пользователей без опыта программирования (рисунок 3).

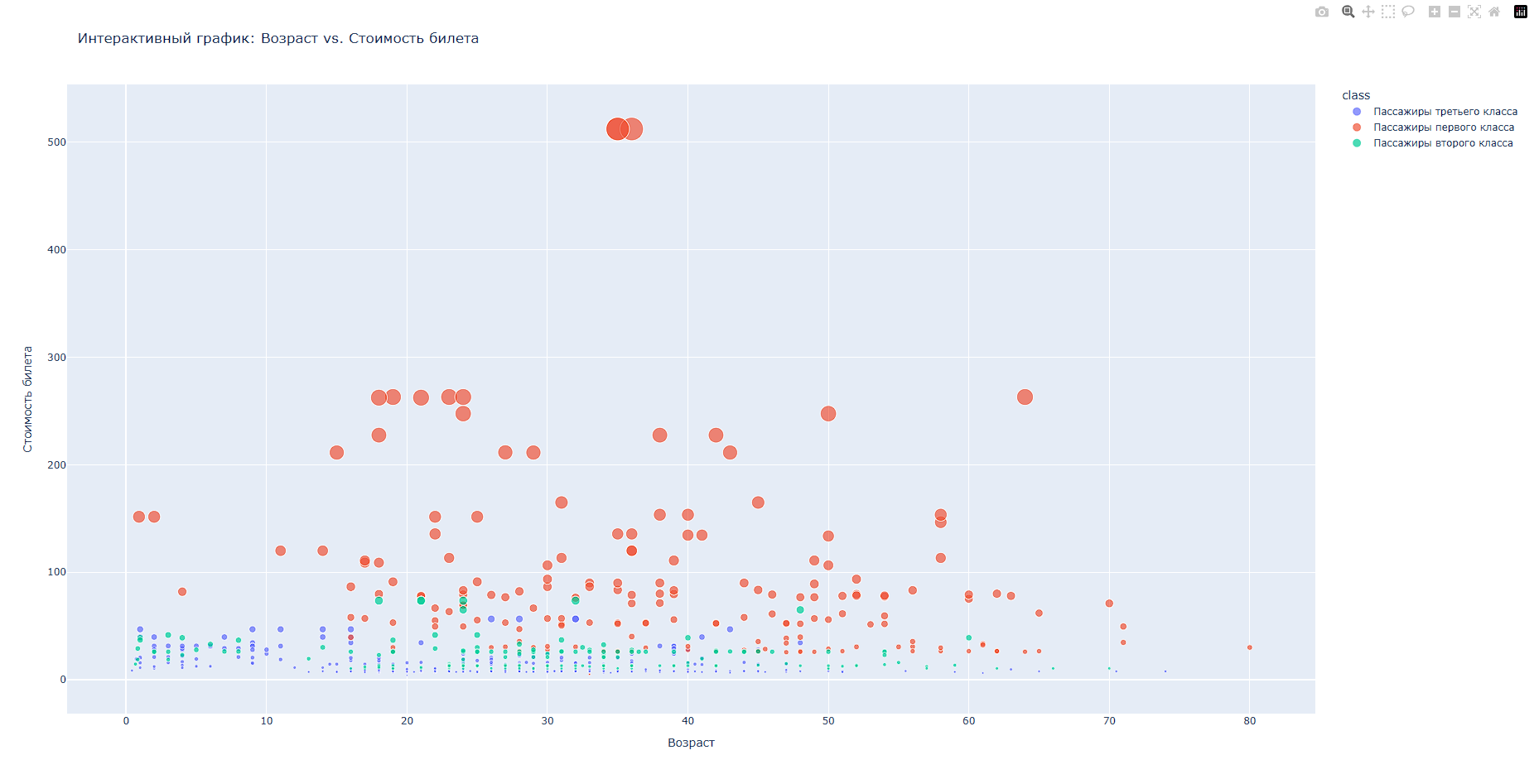


Рисунок 3 – Элементы управления

Для подготовки набора данных Titanic была использована библиотека Pandas:

* Выполнена загрузка данных из готового набора из библиотеки Seaborn.
* Удалены строки с отсутствующими значениями в ключевых столбцах (возраст и стоимость билета), чтобы исключить ошибки при построении графиков.
* Данные классифицированы по классам пассажиров с применением словаря для перевода на понятный язык.

Визуализация данных:

* Matplotlib: Создана гистограмма распределения возрастов по классам пассажиров. Для улучшения восприятия добавлены метки классов и легенда (рисунок 4).

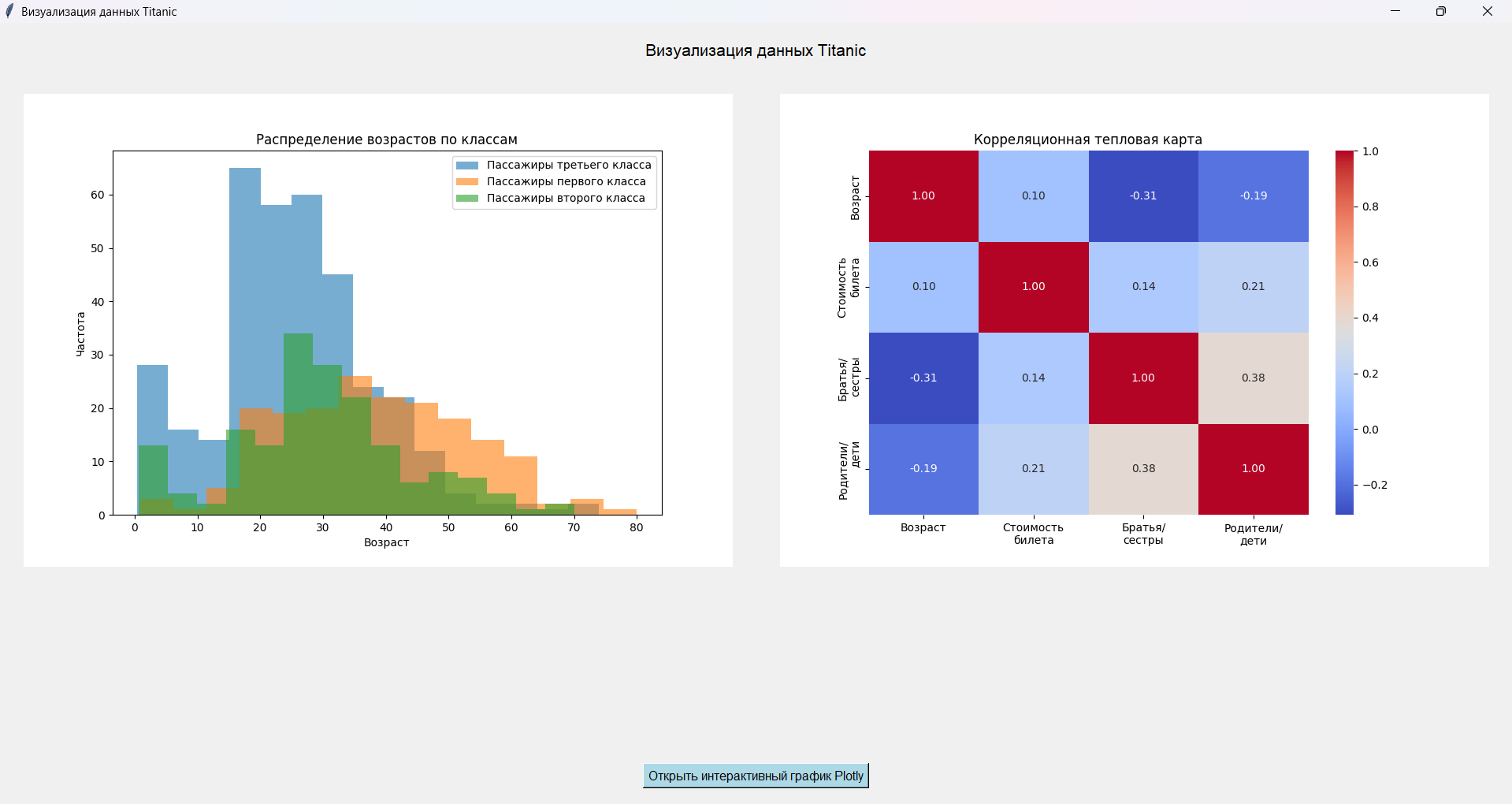


Рисунок 4 – Гистограмма распределения

* Seaborn: Реализована тепловая карта корреляции между ключевыми параметрами набора данных (например, возраст, стоимость билета). Подписаны оси с использованием локализации (рисунок 5).

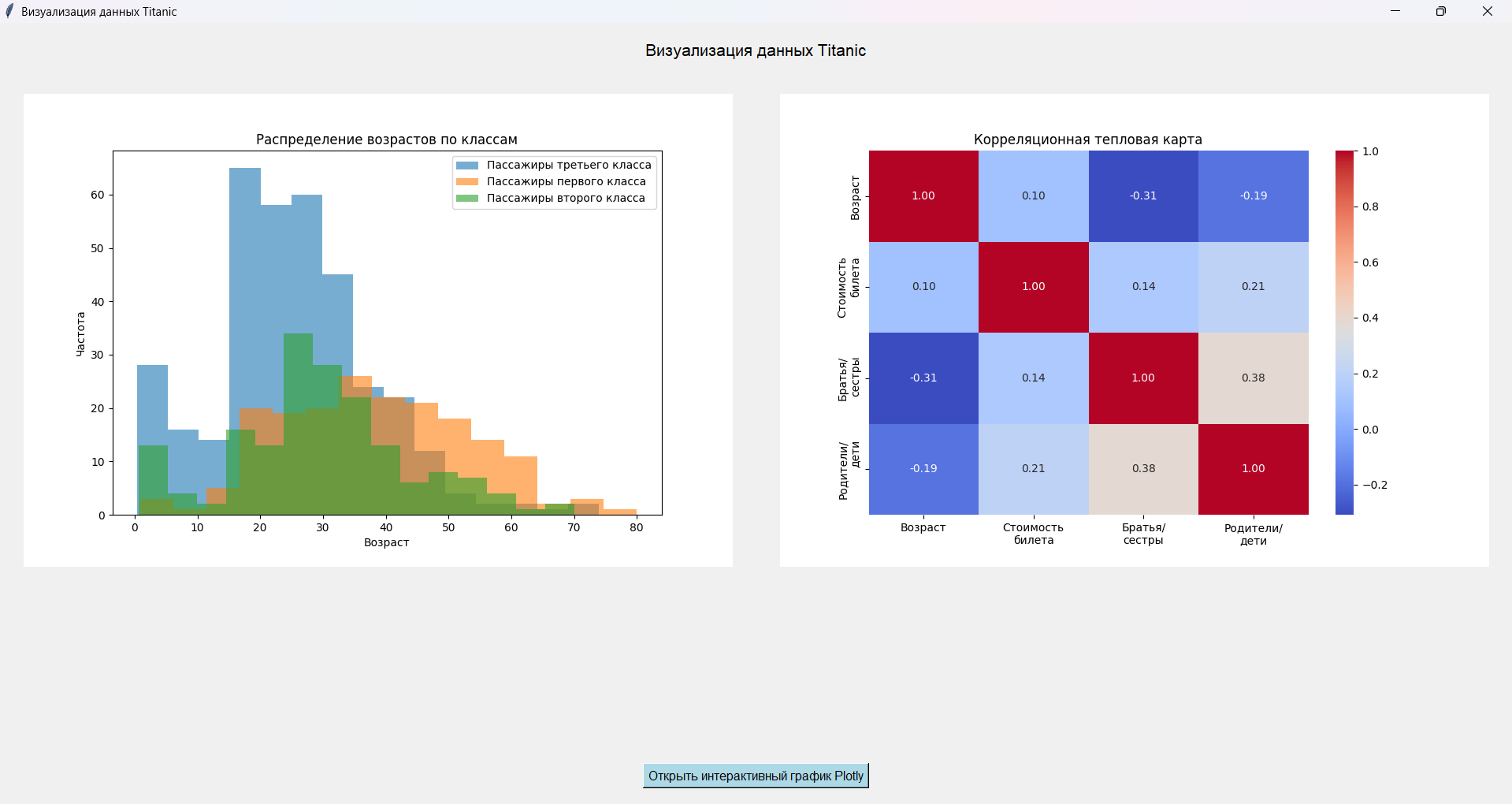


Рисунок 5 – Тепловая карта

* Plotly: Построен интерактивный график для анализа связи между возрастом и стоимостью билета, с дополнительной информацией по полу и месту посадки пассажиров (рисунок 6).

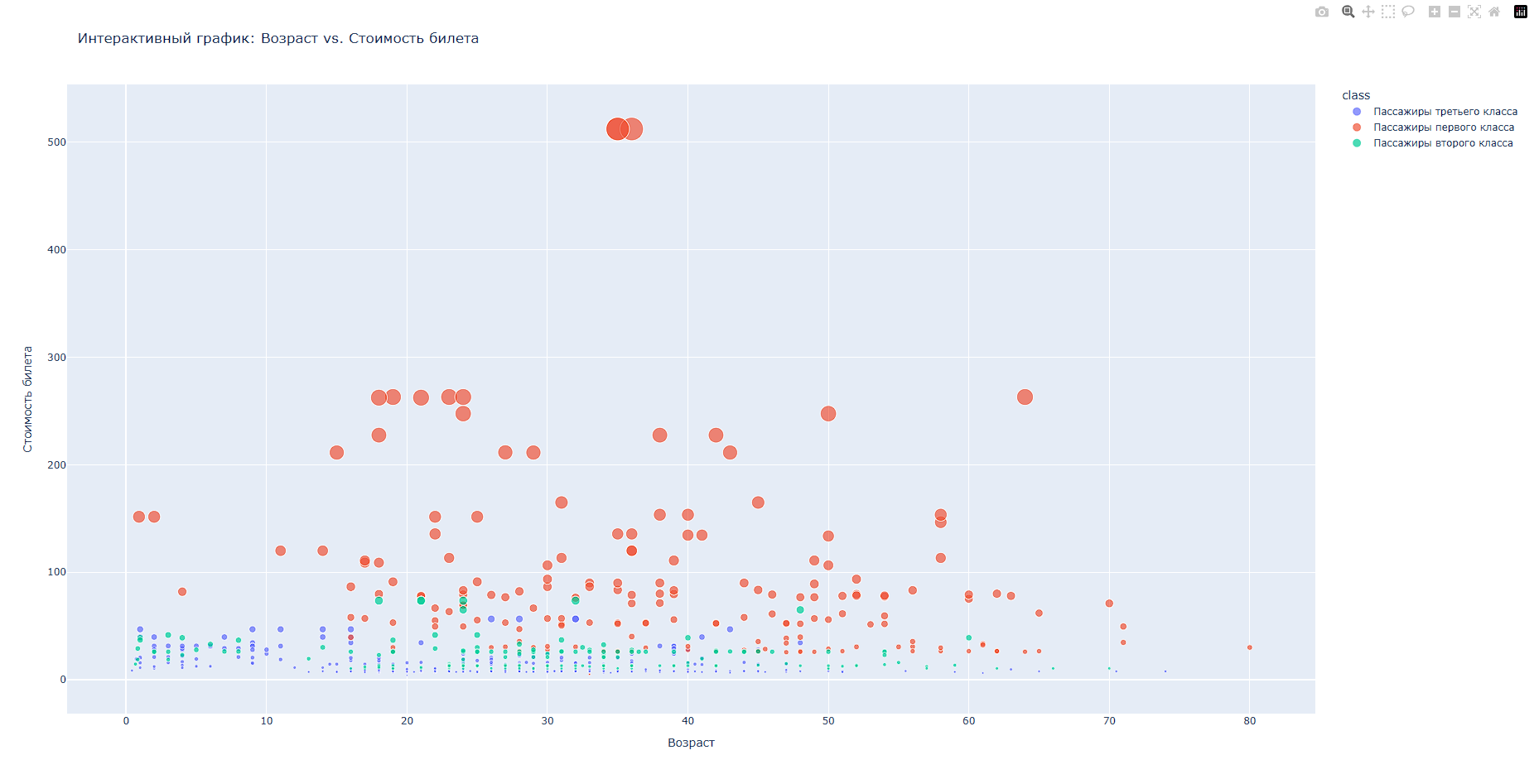


Рисунок 6 – Интерактивный график

Интеграция интерактивных функций:

* Реализована возможность сохранения статичных графиков в формате PNG в корневую папку проекта при первом запуске.
* Добавлена функция автоматического открытия интерактивного графика Plotly в браузере, что упрощает взаимодействие с приложением.

Создано приложение с удобным интерфейсом, позволяющее пользователям исследовать данные Titanic с помощью визуализаций в трех разных библиотеках. Интуитивное размещение элементов и лаконичная структура интерфейса делают программу доступной для широкого круга пользователей.

Файловая структура проекта (рисунок 7):

* Основные файлы Python для запуска приложения.
* Подпапка для хранения сохраненных графиков.
* Временная директория для хранения HTML-файлов интерактивных графиков.

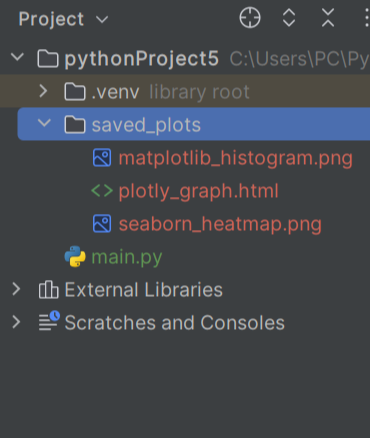


Рисунок 7 – Структура проекта

# 5. Анализ и интерпретация результатов

В ходе эксперимента были протестированы три библиотеки для визуализации данных: Matplotlib, Seaborn и Plotly. Для анализа использовался набор данных Titanic, содержащий информацию о возрасте, стоимости билета, классе пассажиров и других параметрах.

Основными критериями для оценки эффективности библиотек стали:

* Качество визуализаций: насколько понятны и информативны графики, созданные с помощью каждой библиотеки.
* Гибкость: возможности по настройке графиков и добавлению интерактивных элементов.
* Производительность: время, необходимое для построения графиков и обработки данных.

Matplotlib продемонстрировал высокую гибкость и множество вариантов настройки визуализаций. Он идеально подошел для статичных графиков, таких как гистограммы и линейные графики. Время отклика на небольших объемах данных было минимальным. Однако на больших объемах данных производительность может снизиться, а добавление интерактивных элементов потребует дополнительных усилий.

Seaborn использует Matplotlib как основу и предоставляет более высокоуровневые функции для создания статистических визуализаций. Он оказался удобным для построения корреляционных карт и более сложных графиков с использованием различных цветов, и стилей. В отличие от Matplotlib, Seaborn был удобнее для работы с корреляционными тепловыми картами, но не предоставляет нативной поддержки интерактивных графиков.

Plotly продемонстрировал самые лучшие результаты по части интерактивности. Все графики, созданные с его помощью, могут быть автоматически изменены пользователем (масштабирование, фильтрация данных, вывод дополнительной информации при наведении курсора). Время отклика при построении графиков было немного выше, чем у Matplotlib и Seaborn, но интерактивность и доступность графиков через браузер компенсировали это. Однако для простых визуализаций, требующих статичных изображений, использование Plotly может быть избыточным.

Рекомендации по выбору библиотеки в зависимости от задачи:

* Для задач, где важна высокая гибкость, создание статичных и сложных графиков с полной кастомизацией, рекомендуется использовать Matplotlib. Эта библиотека идеально подходит для подготовки визуализаций, которые будут использованы в отчётах или презентациях.
* Если требуется создать красивые и информативные графики без глубоких настроек, например, для анализа корреляции между переменными, Seaborn станет отличным выбором. Он обеспечит простоту в использовании и привлекательный визуальный стиль по умолчанию.
* Для задач, где ключевым требованием является интерактивность, например, для онлайн-приложений или веб-дашбордов, Plotly будет оптимальным вариантом. Интерактивные графики обеспечат пользователю максимальное удобство при анализе данных и взаимодействии с ними.

# Заключение

В ходе работы над проектом, посвященным сравнительному анализу библиотек для визуализации данных, были успешно исследованы и протестированы три популярные библиотеки: Matplotlib, Seaborn и Plotly. Каждая из них продемонстрировала свои сильные стороны и ограничения в зависимости от поставленных задач.

Реализованные графики и визуализации соответствуют заявленным требованиям, показывая, как различные библиотеки справляются с задачами создания статичных и интерактивных графиков, а также визуализацией статистических данных и их взаимосвязей. В результате анализа было выяснено, что Matplotlib и Seaborn подходят для статичных, но детально настроенных графиков, в то время как Plotly является идеальным выбором для создания интерактивных дашбордов и веб-приложений.

В будущем планируется расширить анализ, добавив дополнительные метрики производительности для каждой библиотеки, что поможет более точно выбрать инструмент в зависимости от специфики проекта. Также будет полезно интегрировать созданные визуализации в полноценные дашборды для наглядного представления данных в реальном времени. Кроме того, можно будет доработать функционал, улучшив пользовательский интерфейс и добавив дополнительные интерактивные элементы.

Таким образом, выполненная работа стала основой для дальнейшего изучения и применения инструментов визуализации в различных проектах, что откроет новые возможности для более эффективного анализа данных и принятия решений на основе визуальных представлений информации.