МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение  
высшего образования

**«Южно-Уральский государственный университет**

**(национальный исследовательский университет)»**

**Высшая школа электроники и компьютерных наук**

**Кафедра системного программирования**

|  |
| --- |
| ДОПУСТИТЬ К ЗАЩИТЕ  Заведующий кафедрой, д.ф.-м.н., профессор  \_\_\_\_\_\_\_\_\_\_ Л.Б. Соколинский  «\_\_\_»\_\_\_\_\_\_\_\_\_\_\_ 2025 г. |

**Разработка библиотеки для анализа действий пользователя в IT продукте**

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

ЮУрГУ – 09.03.04 .2025. 308-063.ВКР

|  |  |
| --- | --- |
|  | Научный руководитель,  доцент кафедры СП,  \_\_\_\_\_\_\_\_\_\_ Н.Ю. Долганина  Автор работы, студент группы КЭ-403  \_\_\_\_\_\_\_\_\_\_ Г.А. Разуев  Ученый секретарь  (нормоконтролер)  \_\_\_\_\_\_\_\_\_\_\_\_\_ И.Д. Володченко  «\_\_\_»\_\_\_\_\_\_\_\_\_\_\_ 2025 г. |

Челябинск, 2025 г.

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение  
высшего образования

**«Южно-Уральский государственный университет**

**(национальный исследовательский университет)»**

**Высшая школа электроники и компьютерных наук**

**Кафедра системного программирования**

УТВЕРЖДАЮ

Зав. кафедрой СП

\_\_\_\_\_\_\_\_\_\_ Л.Б. Соколинский

29.01.2025 г.

**ЗАДАНИЕ**

**на выполнение выпускной квалификационной работы бакалавра**

студенту группы КЭ-403

Разуеву Георгию Анатольевичу,

обучающемуся по направлению

09.03.04 «Программная инженерия»

1. **Тема работы** (утверждена приказом ректора от \_\_.\_\_.2025 г. № \_\_\_)

Разработка библиотеки для анализа действий пользователя в IT продукте.

1. **Срок сдачи студентом законченной работы:** 03.06.2025 г.
2. **Исходные данные к работе2**
3. Библиографические ссылки на научные статьи.
4. Библиографические ссылки на книги.
5. Библиографические ссылки на онлайн-ресурсы.
6. **Перечень подлежащих разработке вопросов**
7. Разработка библиотеки для анализа действий пользователя.
8. Написание документации и публикация библиотеки.
9. **Дата выдачи задания:** 29.01.2025 г.

**Научный руководитель,**

доцент кафедры СП А.Т. Латипова

**Задание принял к исполнению** Г.А. https://cdn.discordapp.com/attachments/1248322359912300546/1248324057242271754/image.png?ex=66633fdf&is=6661ee5f&hm=95944c179077338df54661477f8ba6b9ad81997915ffd2f336f79e68d562e414&=Разуев

**ОГЛАВЛЕНИЕ**

[ВВЕДЕНИЕ 4](#_Toc190276209)

[1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ 6](#_Toc190276210)

[1.1. Предметная область проекта 6](#_Toc190276211)

[1.2. Обзор аналогов 6](#_Toc190276212)

[1.3. Анализ существующих решений 10](#_Toc190276213)

[1.4. Заключение 11](#_Toc190276214)

[2. ПРОЕКТИРОВАНИЕ 12](#_Toc190276215)

[2.1. Анализ требований к системе 12](#_Toc190276216)

[2.1.1. Функциональные требования 12](#_Toc190276217)

[2.1.2. Нефункциональные требования 12](#_Toc190276218)

[2.2. Диаграмма вариантов использования 13](#_Toc190276219)

[2.3. Архитектура библиотеки 14](#_Toc190276220)

[ЛИТЕРАТУРА 16](#_Toc190276221)

ВВЕДЕНИЕ

**Актуальность**

Современный рынок IT-продуктов и сервисов характеризуется высокой конкуренцией между компаниями за каждого пользователя. В такой ситуации привлечение новых пользователей и удержание существующих становятся критически важными задачами для бизнеса. Одним из ключевых инструментов, помогающих достичь этих целей, является анализ пользовательского опыта.

Разработка библиотеки для анализа действий пользователей в IT-продукте предоставляет аналитикам, и, как следствие, бизнесу мощный инструмент для глубокого понимания потребностей своих клиентов. Этот инструмент позволяет исследовать поведение пользователей, выявлять проблемные места в продукте и оптимизировать пользовательский опыт. В результате, продукт становится более удобным, функциональным и привлекательным, что в свою очередь способствует увеличению удовлетворенности и лояльности пользователей.

Одним из ключевых преимуществ использования такой библиотеки является ее способность значительно сократить время работы аналитиков над исследованиями. Вместо того чтобы вручную проделывать огромное количество манипуляций с большими наборами данных, аналитики могут использовать автоматизированные инструменты для быстрого и точного анализа. Это позволяет им сосредоточиться на более сложных задачах, таких как разработка стратегий улучшения продукта и принятие решений на основе данных.

Таким образом разрабатываемый продукт способен значительно сократить затраты времени аналитиков, позволив им сосредоточиться на более глобальных задачах, при этом предоставить исчерпывающие данные для принятия бизнес-решений на их основе. В итоге использование разрабатываемого инструмента позволяют компании улучшать свои продукты и услуги, более точно прогнозировать потребности клиентов, оптимизировать маркетинговые кампании и предлагать персонализированные предложения. Это позволит им не только удерживать существующих клиентов, но и привлекать новых, что способствует росту целевых показателей бизнеса и развитию компании.

**Постановка задачи**

Целью выпускной квалификационной работы является разработка библиотеки для исследования пользовательского опыта. Для достижения поставленной цели необходимо решить следующие задачи:

1. провести анализ литературы;
2. спроектировать библиотеку;
3. разработать библиотеку;
4. осуществить тестирование библиотеки;
5. написать документацию библиотеки;
6. опубликовать библиотеку на GitHub.

**Структура и содержание работы**

Работа состоит из введения, четырех глав, заключения, списка литературы и двух приложений. Объем работы составляет 50 страниц, объем списка литературы – 15 источников.

В первой главе описывается …

Вторая глава посвящена …

В третьей главе …

# АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ

## Предметная область проекта

В данной выпускной квалификационной работе предметная область охватывает концепции и методологии, связанные с анализом пользовательского опыта в контексте информационных технологий. С учетом высокой конкуренции на рынке IT-продуктов и сервисов, исследование фактического поведения пользователей становится критически важным для принятия обоснованных бизнес-решений.

Пользовательский опыт (UX) — это восприятие и реакция пользователя на взаимодействие с продуктом или сервисом. Это включает в себя не только функциональные аспекты, такие как удобство и доступность, но и эмоциональные, связанные с ощущениями и удовлетворенностью пользователя.

Основные методики, применяемые в продуктовых исследованиях:

1. Продуктовые метрики: Метрики позволяют количественно оценить эффективность продукта и понять, насколько пользователи довольны его функционалом.Эти метрики помогают отслеживать и анализировать изменения в пользовательском опыте с течением времени.
2. Продуктовые гипотезы: Формулирование гипотез — это основа научного подхода к продуктовому исследованию. Гипотезы помогают определить ключевые вопросы, на которые необходимо ответить в ходе исследования.
3. Сегментация пользователей: Для более точного анализа пользовательского опыта необходимо делить пользователей на группы с похожими характеристиками и потребностями. Сегментация может осуществляться на основе демографических данных, поведения, уровня вовлеченности и других факторов. Это позволяет разрабатывать более целевые стратегии, направленные на удовлетворение потребностей конкретных групп пользователей.
4. Контролируемые эксперименты: Эти эксперименты, такие как A/B-тестирование, позволяют сравнивать два или более варианта продукта, чтобы определить, какой из них обеспечивает лучший пользовательский опыт.
5. Когортный анализ: Этот метод позволяет анализировать поведение пользователей по группам (когортам), которые объединены по определенному критерию, например, времени регистрации.

Разрабатываемая библиотека будет направлена на оптимизацию манипуляций с данными, что существенно упростит применение всех упомянутых методик, что даст возможность специалистам сосредоточиться на более стратегических аспектах исследований, что, в конечном счете, приведет к улучшению качества продуктов и повышению удовлетворенности пользователей.

## Обзор аналогов

Существует достаточно много инструментов для частичной автоматизации анализа данных, это обусловлено тем, что любой бизнес имеет потребность в проведении исследований, которые послужат основанием для принятия глобальных решений, а любой аналитик заинтересован в оптимизации собственной деятельности. Однако большая часть этих инструментов предоставляют достаточно узкий функционал, автоматизирую только небольшой спектр задач. Далее будут рассмотрены 3 аналога, каждый из которых является популярной библиотекой, которая оптимизирует определенный круг задач по анализу данных.

Первым рассматриваемым аналогом является библиотека «Sweetviz» [1]. это библиотека Python с открытым исходным кодом, которая упрощает процесс разведочного анализа данных, создавая красивые визуализации и сводные отчёты всего несколькими строками кода. Она позволяет быстро получить представление о наборе данных, сравнить наборы данных и проанализировать целевые значения. Скриншот примера отчета, составленного библиотекой представлен на рисунке 1.

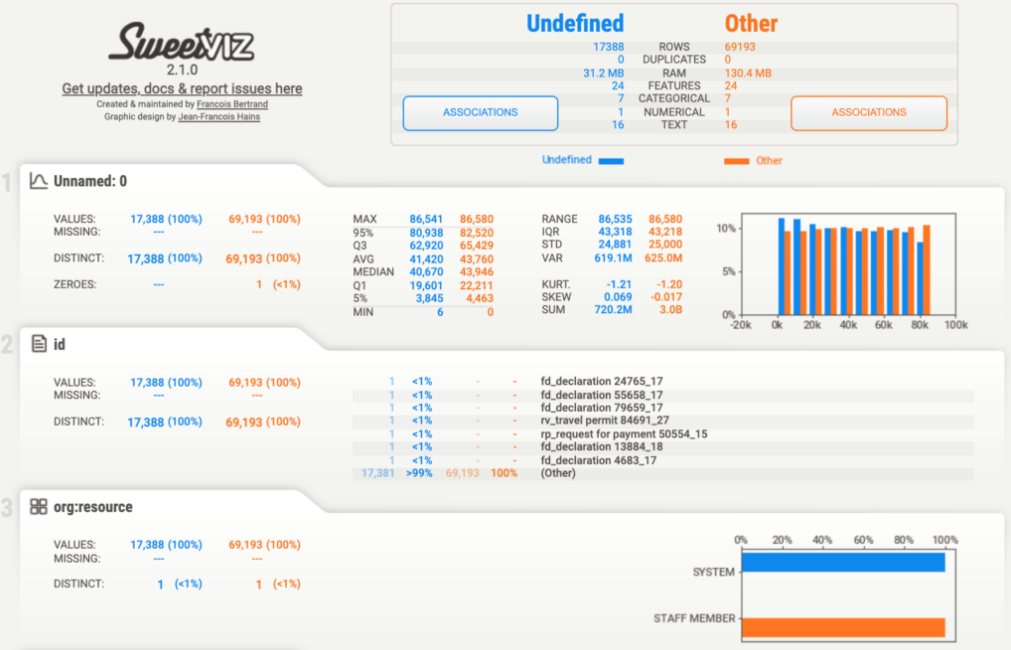


Рисунок 1 – Скриншот отчета «Sweetviz»

Основные выявленные достоинства и недостатки библиотеки приведены в таблице 1.

Таблица 1 – Преимущества и недостатки «Sweetviz»

|  |  |
| --- | --- |
| Преимущества | Недостатки |
| Возможность интерактивного исследования отчета при помощи интуитивно понятного интерфейса | Отсутствие возможности детальной настройки отчета |
| Возможность проводить первичное исследование, выявление примитивных зависимостей и аномалий | Отсутствие возможности предобрабатывать данные |
| Возможность получения функционала с использованием очень небольшого объема кода | Отсутствие инструментов конкретно для анализа действий пользователя |
| Возможность делиться интерактивным отчетом с заказчиками анализа |  |

Следующим рассматриваемым аналогом является «KLib» [2]. это библиотека Python, которая предоставляет возможности автоматического разведочного анализа данных (EDA) и профилирования данных. Она предлагает различные функции и визуализации для быстрого изучения и анализа наборов данных. Скриншот отчета, составленного данной библиотекой представлен на рисунке 2.

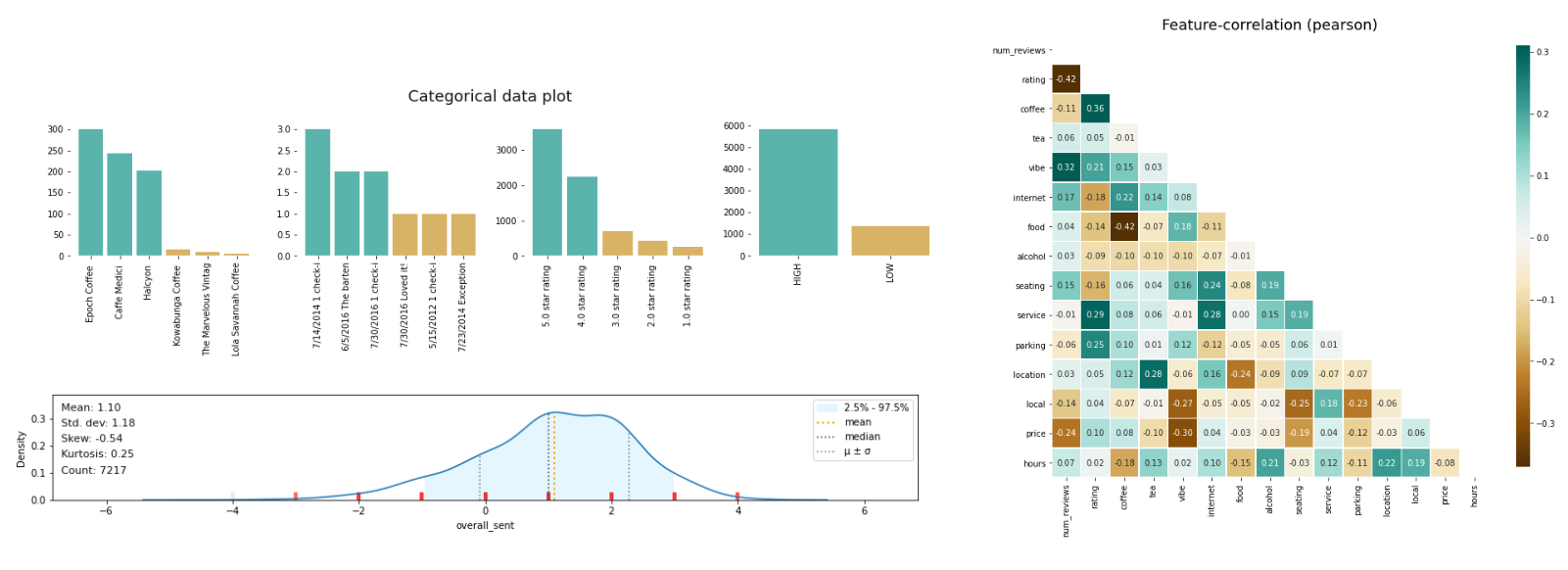


Рисунок 2 – Скриншот отчета «KLib»

Основные выявленные достоинства и недостатки сервиса приведены в таблице 2.

Таблица 2 – Достоинства и недостатки «KLib»

|  |  |
| --- | --- |
| **Достоинства** | **Недостатки** |
| Возможность составления отчета при помощи минимального объема кода | Отсутствие возможности детальной настройки отчета |
| Возможность проводить первичное исследование, выявление примитивных зависимостей и аномалий | Отсутствие возможности предобрабатывать данные |
|  | Отсутствие инструментов конкретно для анализа действий пользователя |

Последним и наиболее релевантным рассматриваемым аналогом является «Retentioneering» [3]. Это библиотека Python, которая значительно упрощает анализ потоков действий пользователей, пользовательских путей и журналов событий. Может использоваться для изучения поведения пользователей, сегментации пользователей и формирования гипотез о том, что побуждает пользователей к желаемым действиям или к отказу от продукта. Скриншот описания данной библиотеки представлен на рисунке 3.

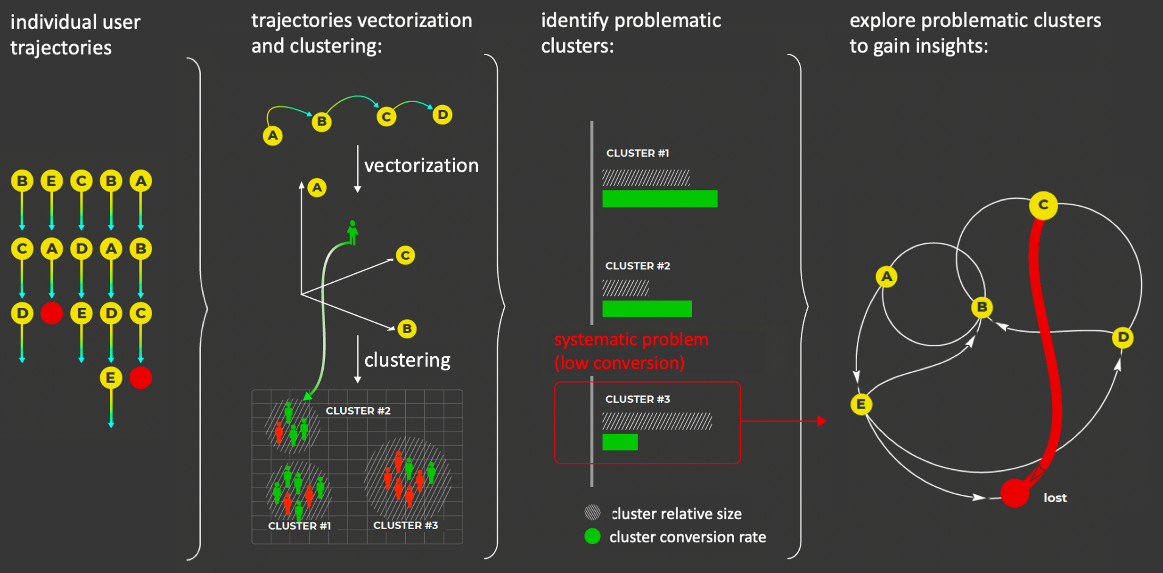


Рисунок 3 – Скриншот описания библиотеки «Retentioneering»

Основные выявленные достоинства и недостатки библиотеки приведены в таблице 3.

Таблица 3 – Достоинства и недостатки «Retentioneering»

|  |  |
| --- | --- |
| **Достоинства** | **Недостатки** |
| Возможность исследовать последовательности действий пользователей | Отсутствие возможности исследования отдельных сессий пользователя, только всего их пути в приложении |
| Возможность проводить первичное исследование данных | Отсутствие возможности получить легко интерпретируемую информацию о популярности пользовательских сценариев |
| Возможность получения широкого функционала с использованием небольшого объема кода | Отсутствие инструментов для выявления аномалий в метриках |
| Возможность предобрабатывать данные | Отсутствие инструментов для расчета и анализа популярных продуктовых метрик |
| Возможность выявлять сегменты пользователей |  |

Обзор аналогов показал, что большая часть существующих инструментов оптимизируют только первичное исследование данных и выявление поверхностных закономерностей, не предоставляя функционал для исследования конкретно пользовательского опыта. А единственная библиотека, предназначенная именно для анализа действий пользователя в продукте, имеет ряд ограничений. Однако анализ также выявил ряд присущих всем преимуществ, которые должны быть и в разрабатываемом продукте, среди них: возможность проводить первичное исследование данных, возможность предобработки данных, возможность получения широкого функционала с использованием небольшого объема кода. Среди недостатков, которые стоит предусмотреть: отсутствие возможности детальной настройки исследования, отсутствие инструментов для расчета популярных продуктовых метрик.

Таким образом для того, чтобы разрабатываемая библиотека решала проблему оптимизации широкого спектра задач по исследованию пользовательского опыта необходимо учесть все достоинства и недостатки аналогов.

## Анализ существующих решений

В качестве основы для разрабатываемого продукта будут использованы python библиотеки для работы с данными, являющиеся наиболее часто используемыми инструментами в области науки о данных:

1. «NumPy» – это библиотека для научных вычислений. Она используется для выполнения математических операций над большими объемами числовых данных, таких как линейная алгебра, матричные операции и работа с векторами.
2. «SciPy» – Эта библиотека включает модули для оптимизации, интегрирования, интерполяции, статистики, обработки сигналов и изображений, а также многие другие специализированные задачи..
3. «Pandas» – одна из самых популярных библиотек для анализа и манипуляции табличными данными.
4. «StatsModels» – библиотека для статистической модели и оценок. Отличается поддержкой широкого спектра статистических моделей и методов.
5. «Matplotlib» – основная библиотека для построения двумерных графиков в Python. Она поддерживает различные типы графиков, включая линии, гистограммы, круговые диаграммы, тепловые карты и многое другое.
6. «Seaborn» – построенная поверх Matplotlib библиотека для создания статичных и интерактивных графиков. Она ориентирована на более эстетичное представление данных и имеет встроенную поддержку различных типов статистических графиков.
7. «Plotly» – мощная библиотека для интерактивной визуализации данных. Поддерживает создание как статичных, так и динамических графиков через веб-интерфейсы.
8. «Scikit-Learn» – это библиотека для машинного обучения, включающая широкий спектр алгоритмов классификации, регрессии, кластеризации и снижения размерности. Также содержит инструменты для предобработки данных и оценки моделей

Для разработки будет использована среда программирования PyCharm 2019.3.3 Community Edition.

## Заключение

В ходе анализа предметной области был проведен обзор аналогов, в ходе которого были выявлены недостатки, которые нужно учесть и достоинства, которые необходимо воссоздать в разрабатываемом продукте.

Также был проведен анализ существующих решений, в результате которого, для реализации библиотеки было принято решение использовать библиотеки python для работы с данными, расчета статистических показателей и машинного обучения, а среду программирования PyCharm для разработки.

# ПРОЕКТИРОВАНИЕ

## Анализ требований к системе

После рассмотрения предметной области и обзора доступных средств разработки были выявлены функциональные и нефункциональные требования, которые должны быть учтены при разработке библиотеки.

### Функциональные требования

Функциональные требования необходимы для определения процессов, действий и операций, которые способна выполнить система. Функциональные требования используются для выражения поведения системы и определения ее функциональности. Исходя из назначения и анализа аналогов, разрабатываемая система должна соответствовать требованиям, которые представлены ниже.

1. Пользователь должен иметь возможность подготовить данные для исследования, т.е. при необходимости отсортировать события в логическом порядке, добавить разбиение на сессии, разметить существующие и добавить новые события и т.д.;
2. Библиотека должна предоставлять широкий спектр инструментов для анализа действий пользователей, среди которых визуализации последовательностей действий, расчет популярности пользовательских сценариев и т.д.;
3. Библиотека должна предоставлять инструменты для анализа продуктовых метрик, как наиболее популярных, так и пользовательских;
4. Библиотека должна предоставлять инструменты для анализа продуктовых воронок;
5. Библиотека должна предоставлять инструменты для сегментации пользователей на основе их действий и дополнительной информации.

### Нефункциональные требования

Нефункциональные требования служат для описания условий и качества разрабатываемой среды с целью создания системы с приемлемым качеством. В ходе рассмотрения инструментов для реализации системы, представленных в первой главе, а также из задач, которые решает разрабатываемая система, к ней были сформированы следующие нефункциональные требования.

1. Инструменты для манипуляций с данными – библиотеки Pandas и NumPy;
2. инструменты для визуализации данных – библиотеки Matplotlib, seaborn, Plotly;
3. инструменты для расчета статистических показателей – библиотеки StatsModels и SciPy;
4. Инструменты для машинного обучения – библиотека Scikit-learn

## Диаграмма вариантов использования

Для составления диаграммы вариантов использования был использован язык графического описания UML. На рисунке 4 представлена составленная диаграмма.

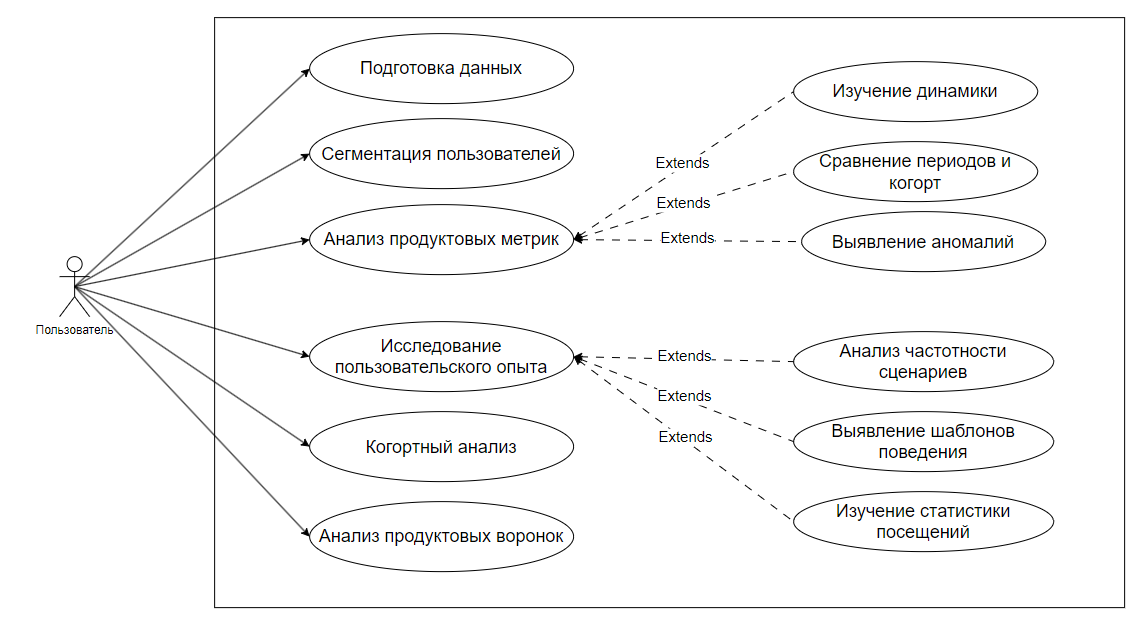


Рисунок 4 – Диаграмма вариантов использования

В разрабатываемой библиотеке основным актером, взаимодействующим с системой, является «Пользователь библиотеки» – аналитик или разработчик, использующий библиотеку для анализа поведения пользователей в IT продукте.

## Архитектура библиотеки

Первый и один из самых ответственных этапов проектирования – создание архитектуры классов хранения данных, т.к. их будут использовать все остальные классы библиотеки. На рисунке 5 представлена диаграмма классов хранения данных.

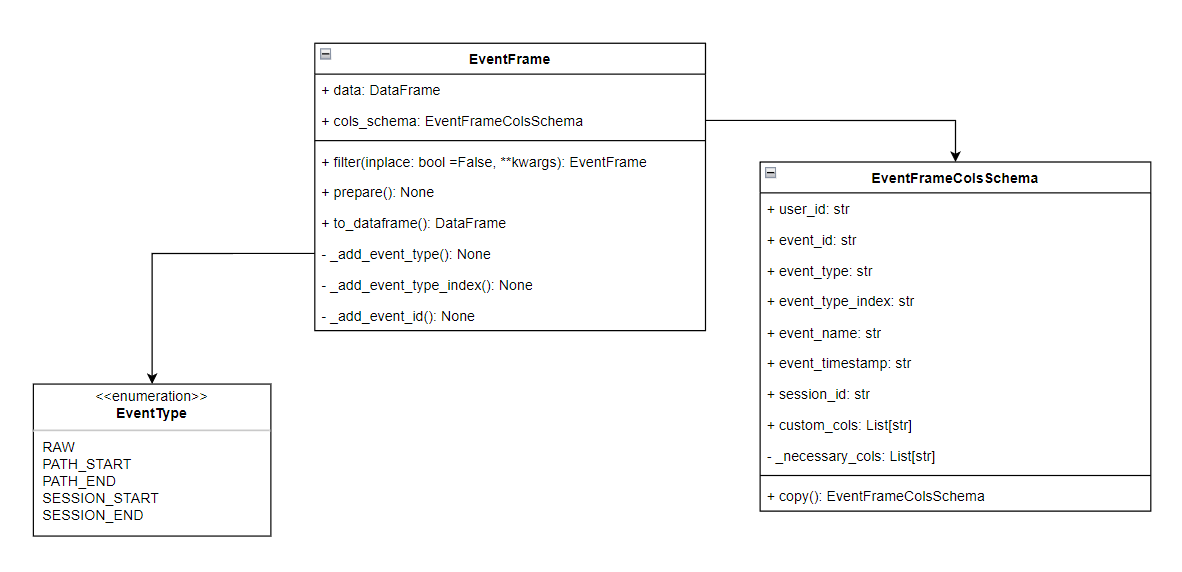


Рисунок 5 – диаграмма классов хранения данных

Основной класс хранения данных – «EventFrame», он хранит сами данные в виде pandas.DataFrame, схему названий колонок в виде экземпляра класса «EventFrameColsSchema» и определяет несколько методов по преобразованию данных для дальнейшего их использования инструментами библиотеки. Класс «EventFrameColsSchema» предназначен для того, чтобы использовать данные с любыми названиями колонок, т.к. он хранит схему соответствия оригинальному названию колонки и ее смысловому содержанию.

Второй этап проектирования сосредотачивается на классах для обработки данных, который является не менее важным компонентом библиотеки. На рисунке 6 представлена диаграмма классов, отвечающих за предварительную обработку данных.

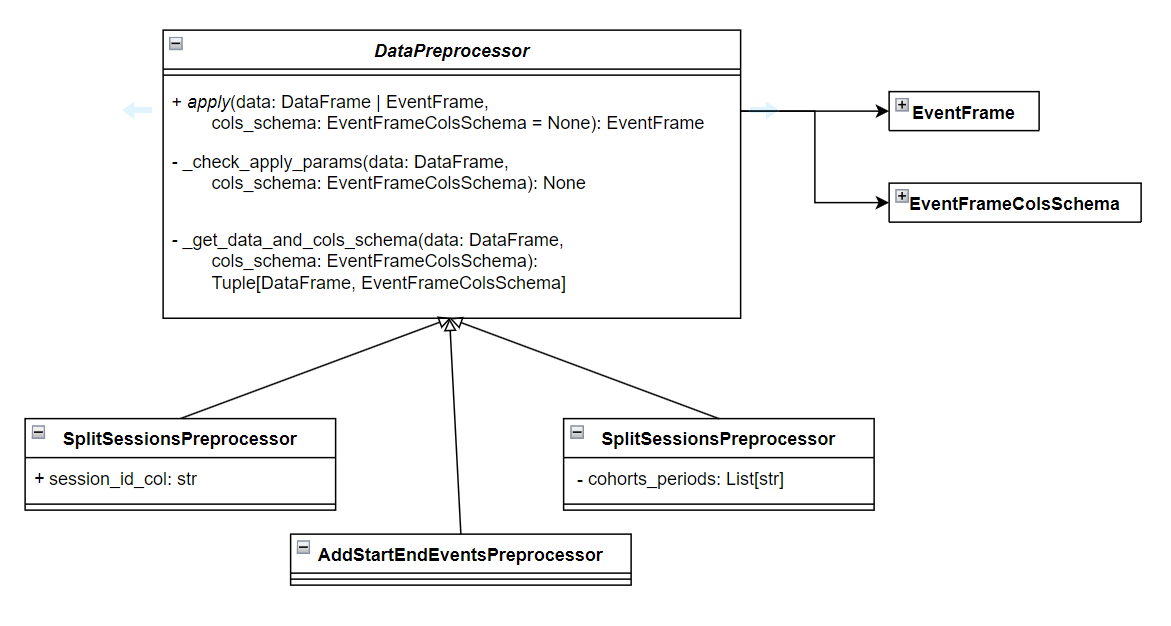


Рисунок 6 - диаграмма классов, отвечающих за предварительную обработку данных

Абстрактным классом для обработки данных является «DataPreprocessor», который содержит абстрактный метод «apply» для применения различных предобработок к данным, хранящимся в классе EventFrame. Класс «DataPreprocessor» также включает в себя несколько ключевых приватных методов методов, обеспечивающие проверку параметров и получение данных в структурированном виде. Также на диаграмме представлены несколько конкретных реализаций абстрактного класса, каждый из которых предназначен для решения конкретной задачи предобработки.

Данная архитектура классов для обработки данных гарантирует гибкость и модульность, позволяя легко добавлять новые методы обработки и адаптировать библиотеку под изменяющиеся требования пользователя.

ЛИТЕРАТУРА

1. «Seeking alpha» [Электронный ресурс] URL: [https://seekingalpha.com/](https://seekingalpha.com/%20) (дата обращения 04.03.2024).
2. «Investing.com» [Электронный ресурс] URL: <https://www.investing.com/> (дата обращения 04.03.2024).
3. «MorningStar» [Электронный ресурс] URL: [https://www.morningstar.com/](https://www.morningstar.com/%20) (дата обращения 04.03.2024).
4. API «YahooFinance» [Электронный ресурс] URL:   
   https://python-yahoofinance.readthedocs.io/ (дата обращения 04.03.2024).
5. API «Polygon.io» [Электронный ресурс] URL:   
   [https://polygon.io/](https://polygon.io/%20%20)  (дата обращения 04.03.2024).
6. «Scikit-Learn» [Электронный ресурс] URL:   
   [https://scikit-learn.org/](https://scikit-learn.org/%20) (дата обращения 04.03.2024).
7. Django Documentation [Электронный ресурс] URL: [https://docs.djangoproject.com/](https://docs.djangoproject.com/%20) (дата обращения: 14.02.2024 г.).
8. Django Documentation [Электронный ресурс] URL:  [https://react.dev/](%20https://react.dev/) (дата обращения: 14.02.2024 г.).
9. Figma [Электронный ресурс] URL: https://www.figma.com/ (дата обращения: 14.02.2024 г.).
10. John Lintner. “The valuation of risk assets and the selection of risky investments in stock portfolios and capital budgets”. In: The Review of Economics and Statistics 47.1 (1965).
11. Django Documentation [Электронный ресурс] URL: https://www.sqlite.org/docs (дата обращения: 14.02.2024 г.).
12. Дронов В.А. Django 3.0. Практика создания веб-сайтов на Python. – СПб.: БХВ-Петербург, 2021. – 704 с.: ил. – (Профессиональное программирование).