

Министерство науки и высшего образования Российской Федерации ФГАОУ ВО «Южно-Уральский государственный университет (НИУ)»

Высшая школа электроники и компьютерных наук

Кафедра системного программирования

ДНЕВНИК

прохождения практики "Производственная практика,

(научно-исследовательская работа)"

студента: Разуева Георгия Анатольевича,   
группы КЭ-403 направление подготовки: 09.03.04 Программная инженерия Предприятие: ЮУрГУ (НИУ)

Дата начала практики: 03.02.2025

Назначен:

практикант

Дата окончания практики: 02.03.25

Руководитель практики от предприятия:

Доцент кафедры СП, к.ф.-м.н. Турлакова С.У

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

"расшифровка подписи"

М.П

Студент: Разуев Георгий Анатольевич, КЭ-403

ФИО (полностью), группа

Руководитель: Доцент кафедры СП, Долганина Н.Ю.

Должность, ученая степень Фамилия ИО

Тема: Разработка модулей обработки данных, кластеризации пользователей и анализа продуктовых метрик и воронок библиотеки для продуктовой аналитики.

**Календарный план  
выполнения производственной практики бакалавра**

**(научно-исследовательская работа)  
(*03.02.2025 – 02.03.2025*)**

|  |  |  |  |
| --- | --- | --- | --- |
| **№**  **п/п** | **Дата встречи** | **Выполненные работы** | **Подпись  руководителя** |
| 1 неделя практики | | | |
|  | 03.02.2025 | Формирование плана и структуры работы |  |
|  | 07.02.2025 | Изучение теоретической части проектирования моделей |  |
| 2 неделя практики | | | |
|  | 10.02.2025 | Разработка модуля обработки данных |  |
|  | 13.02.2025 | Разработка модуля кластеризации пользователей |  |
| 3 неделя практики | | | |
|  | 17.02.2025 | Разработка модуля анализа продуктовых метрик и воронок |  |
|  | 20.02.2025 | Проверка работоспособности |  |
| 4 неделя практики | | | |
|  | 24.02.2025 | Тестирование и исправление ошибок |  |

Студент /Разуев Г.А.

Фамилия ИО

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

**ОТЧЕТ**

**по производственной практике**

**(научно-исследовательская работа)**

|  |  |
| --- | --- |
|  | Выполнил, студент группы КЭ-403  Разуев Г.А.  Научный руководитель,  доц.каф.СП  Долганина Н.Ю.  Руководитель практики:  доц.каф.СП, к.ф.-м.н.\_\_  Турлакова С.У.\_\_\_\_\_\_\_  Дата: \_\_\_\_\_\_\_\_\_\_\_\_\_  Оценка: \_\_\_\_\_\_\_\_\_\_\_\_\_ |

Челябинск, 2025 г.

Министерство науки и высшего образования Российской Федерации

Южно-Уральский государственный университет

Кафедра системного программирования

УТВЕРЖДАЮ

Зав. кафедрой

системного программирования

\_\_\_\_\_\_\_\_\_\_\_\_\_ Л.Б. Соколинский

**ЗАДАНИЕ**

**на производственную практику**

**(научно-исследовательскую работу)**

1. **Тема работы**

Разработка модулей обработки данных, кластеризации пользователей и анализа продуктовых метрик и воронок библиотеки для продуктовой аналитики.

1. **Исходные данные к работе**
   1. Lean Analytics: Use Data to Build a Better Startup Faster [Электронный ресурс] URL: https://www.oreilly.com/library/view/lean-analytics/ (дата обращения: 03.02.2025 г.).
   2. Scikit Learn Documentation. [Электронный ресурс] URL: https://scikit-learn.org/stable/index.html (дата обращения: 03.02.2025 г.).
   3. Pandas Documentation. [Электронный ресурс] URL: https://pandas.pydata.org/docs/ (дата обращения: 03.02.2025 г.).
2. **Перечень подлежащих разработке вопросов**
   1. Разработать модуль обработки данных.
   2. Разработать модуль кластеризации пользователей.
   3. Разработать модуль анализа продуктовых метрик и воронок.
3. **Сроки**

Дата выдачи задания: 3 февраля 2025 г.

Срок сдачи законченной работы: 28 февраля 2025 г.

**Руководитель практики со стороны ЮУрГУ:**

Доцент кафедры СП,к.ф.-м.н. \_\_\_\_\_\_\_\_\_\_\_\_\_ Турлакова С.У.

подпись ФИО ответственного

**Научный руководитель практики:**

Доцент кафедры СП \_\_\_\_\_\_\_\_\_\_\_\_\_ Долганина Н.Ю.

подпись ФИО научного руководителя

**Задание принял к исполнению:**

\_\_\_\_\_\_\_\_\_\_\_\_\_ Разуев Г.А.

подпись ФИО студента

**ОГЛАВЛЕНИЕ**

[ВВЕДЕНИЕ 4](#_Toc191410550)

[1. АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ 6](#_Toc191410551)

[1.1. Предметная область проекта 6](#_Toc191410552)

[1.2. Обзор аналогов 7](#_Toc191410553)

[1.3. Анализ существующих решений 11](#_Toc191410554)

[1.4. Заключение 12](#_Toc191410555)

[2. ПРОЕКТИРОВАНИЕ 13](#_Toc191410556)

[2.1. Анализ требований к системе 13](#_Toc191410557)

[2.1.1. Функциональные требования 13](#_Toc191410558)

[2.1.2. Нефункциональные требования 14](#_Toc191410559)

[2.2. Диаграмма вариантов использования 14](#_Toc191410560)

[2.3. Архитектура библиотеки 15](#_Toc191410561)

[3. РЕАЛИЗАЦИЯ 18](#_Toc191410562)

[3.1. Реализация модулей хранения и обработки данных 18](#_Toc191410563)

[3.2. Реализация модуля кластеризации 20](#_Toc191410564)

[3.3. Реализация модуля метрик 23](#_Toc191410565)

[3.4. Реализация модуля дополнительных средств анализа 24](#_Toc191410566)

[3.4.1. Реализация инструментов для расчета продуктовых воронок 25](#_Toc191410567)

[3.4.2. Реализация инструментов для когортного анализа 26](#_Toc191410568)

[4. ТЕСТИРОВАНИЕ 27](#_Toc191410569)

[ЗАКЛЮЧЕНИЕ 29](#_Toc191410570)

[ЛИТЕРАТУРА 30](#_Toc191410571)

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

1. провести анализ литературы;
2. спроектировать прототип библиотеки;
3. разработать прототип библиотеки;
4. осуществить тестирование реализованных модулей.

**Структура и содержание работы**

Работа состоит из введения, четырех глав, заключения, списка литературы и двух приложений. Объем работы составляет 30 страниц, объем списка литературы – 11 источников.

# АНАЛИЗ ПРЕДМЕТНОЙ ОБЛАСТИ

## Предметная область проекта

В данной 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, они 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 некоторые задачи 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.

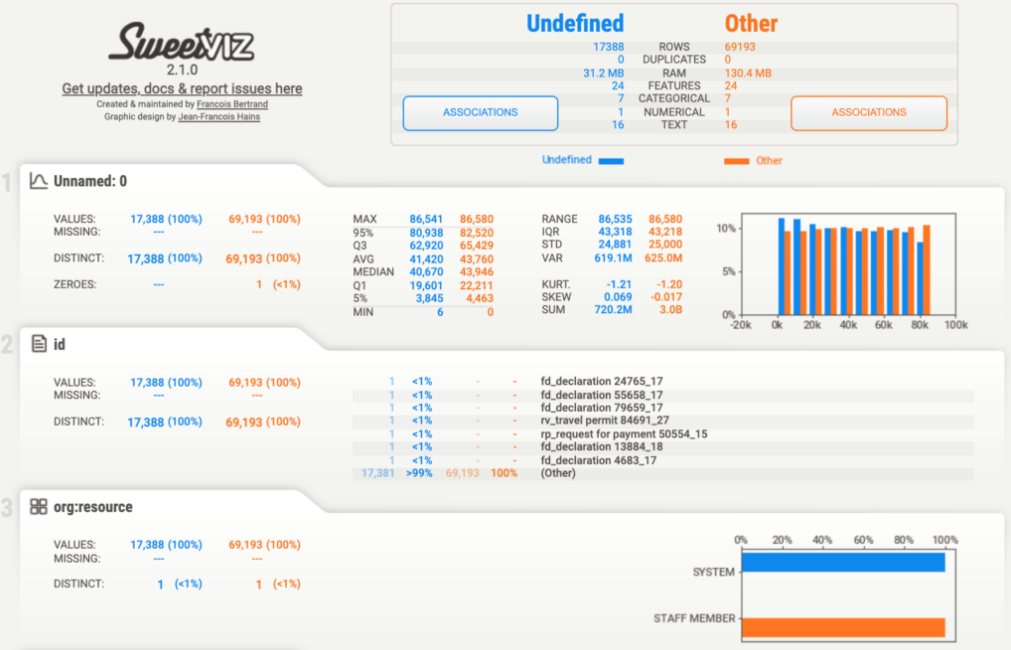


Рисунок – Скриншот отчета «Sweetviz»

Основные выявленные достоинства и недостатки библиотеки приведены в таблице 1.

Таблица – 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 широкого 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.

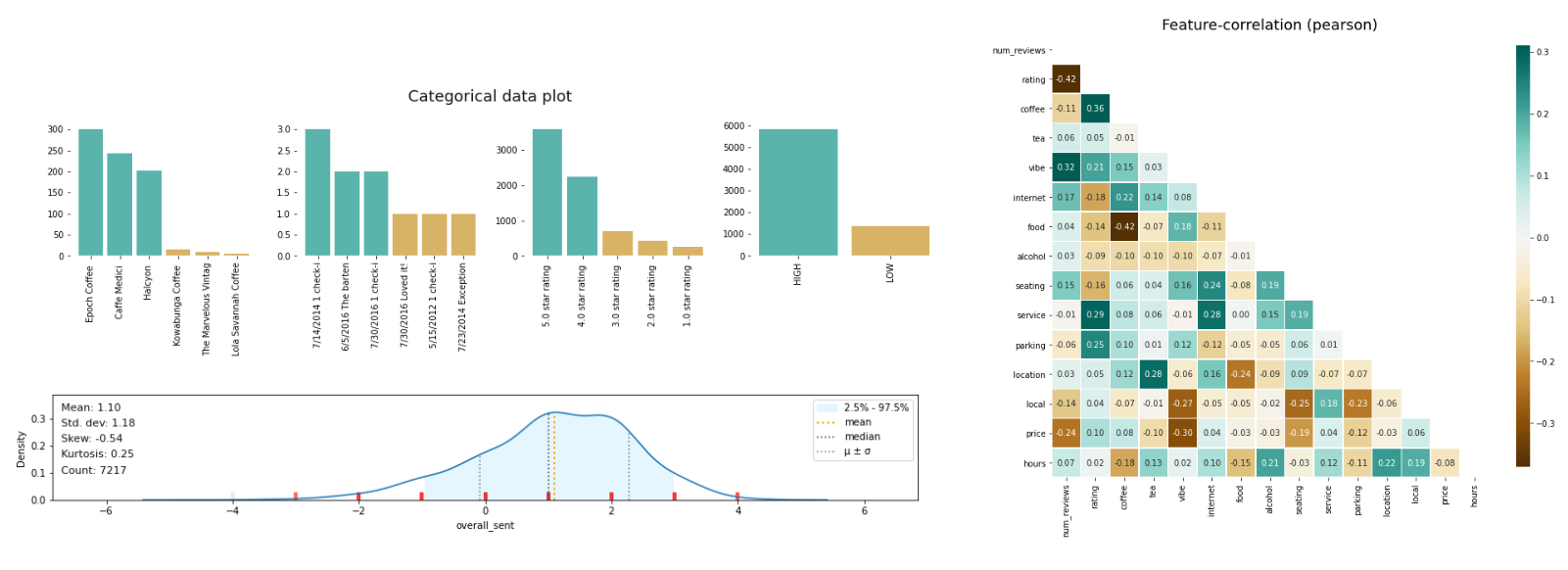


Рисунок – Скриншот отчета «KLib»

Основные выявленные достоинства и недостатки сервиса приведены в таблице 2.

Таблица – 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.

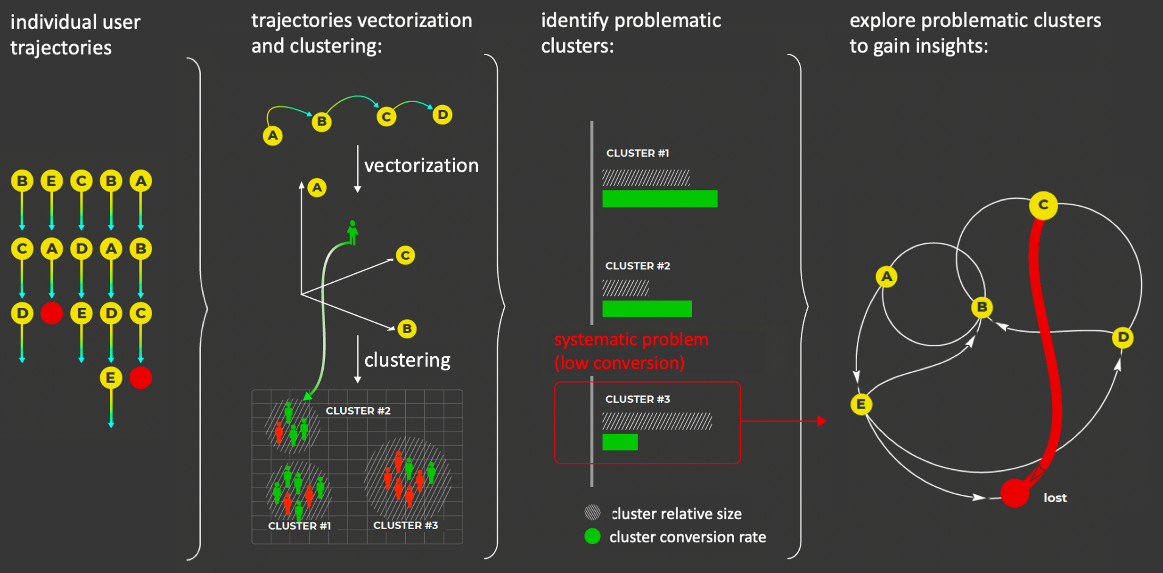


Рисунок – Скриншот описания библиотеки «Retentioneering»

Основные выявленные достоинства и недостатки библиотеки приведены в таблице 3.

Таблица – 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, выявление зависимостей и аномалий | 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 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, выявление зависимостей и аномалий, 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. «Pandas» [4] – одна из самых популярных библиотек для анализа и манипуляции табличными данными.
2. «NumPy»  [5] – 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.
3. «Matplotlib»  [6] – основная библиотека для построения двумерных графиков в Python. Она поддерживает различные типы графиков, включая линии, гистограммы, круговые диаграммы, тепловые карты и многое другое.
4. «Seaborn»  [7] – 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.
5. «Plotly»  [8] – мощная библиотека для интерактивной визуализации данных. Поддерживает создание как статичных, так и динамических графиков через веб-интерфейсы.
6. «StatsModels»  [9] – библиотека для статистической модели и оценок. Отличается поддержкой широкого спектра статистических моделей и методов.
7. «Scikit-Learn»  [10] – это библиотека для машинного обучения, включающая широкий спектр алгоритмов классификации, регрессии, кластеризации и снижения размерности. Также содержит инструменты для предобработки данных и оценки моделей.
8. «SciPy» [11] – Эта библиотека включает модули для оптимизации, интегрирования, интерполяции, статистики, обработки сигналов и изображений, а также многие другие специализированные задачи.

## 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, выявить когорты пользователей 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 [4] и NumPy [5];
2. инструменты для визуализации данных – библиотеки Matplotlib [6], seaborn [7], Plotly [8];
3. инструменты для расчета статистических показателей – библиотеки StatsModels [9] и SciPy [11];
4. Инструменты для машинного обучения – библиотека Scikit-learn [10]

## Диаграмма вариантов использования

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



Рисунок – Диаграмма вариантов использования

В разрабатываемой библиотеке основным актером, взаимодействующим с системой, является «Пользователь библиотеки» – 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.

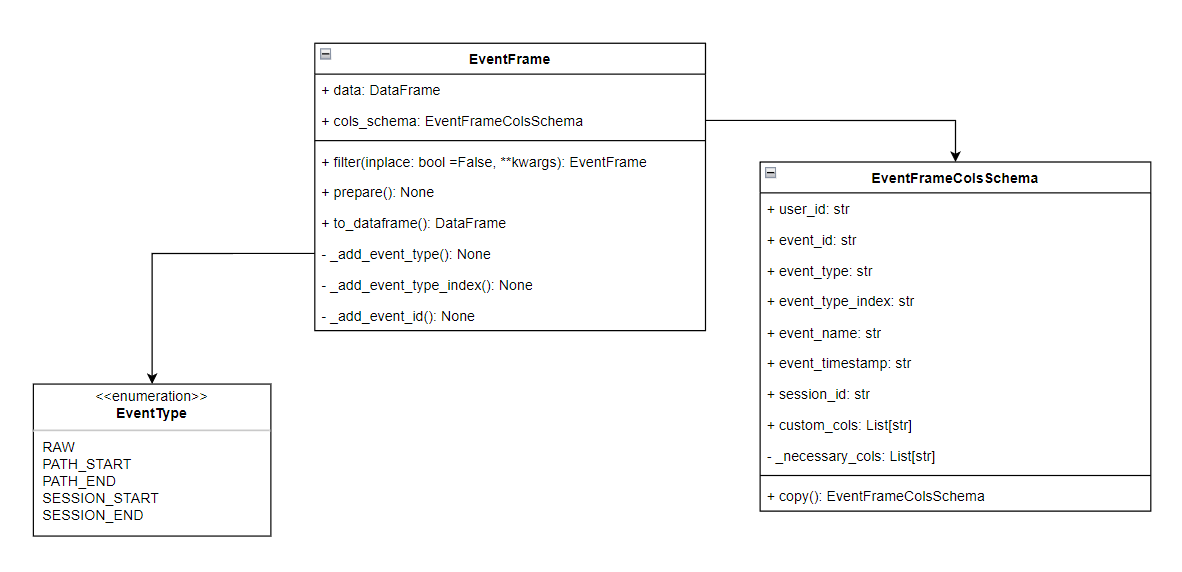


Рисунок – Диаграмма классов хранения данных

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.

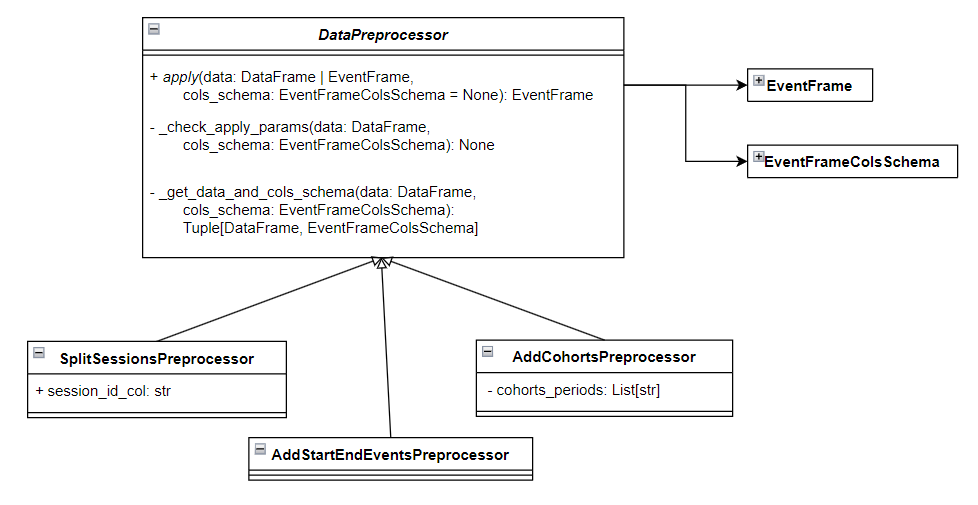


Рисунок – Диаграмма классов, отвечающих за предварительную обработку данных

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, обеспечивающих 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.

Важно предоставить пользователю возможность решать задачи кластеризации как встроенными методами библиотеки, так и сторонними алгоритмами, поэтому при проектировании важно разделить функционал на независящие программные элементы. На рисунке 7 представлена диаграмма классов, отвечающих за кластеризацию

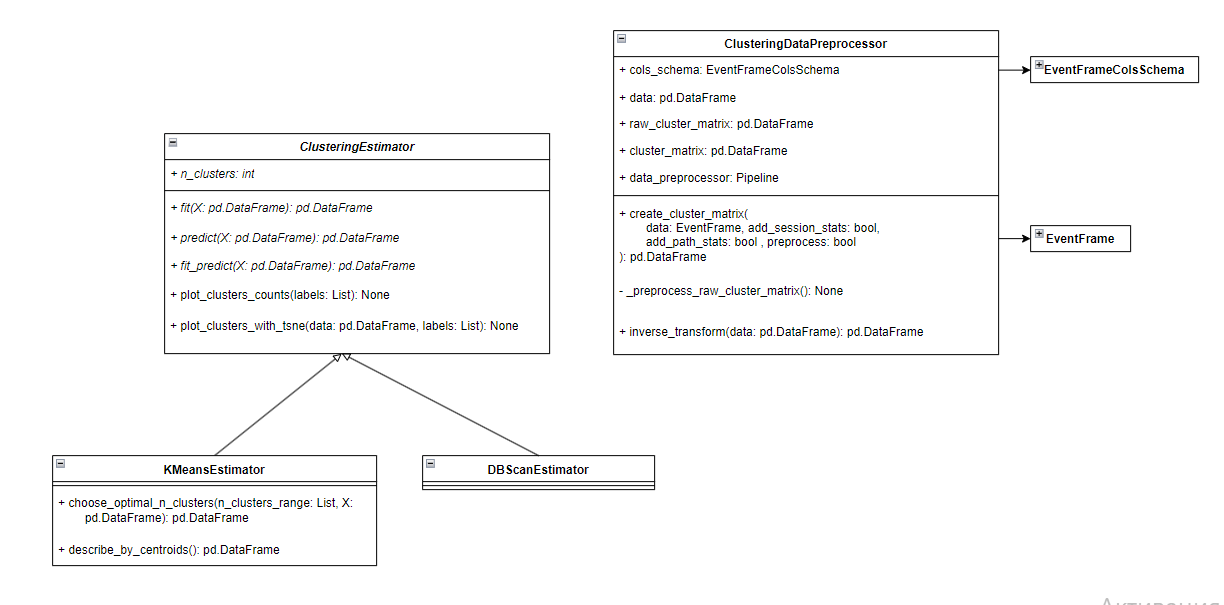


Рисунок – Представлена диаграмма классов, отвечающих за кластеризацию

Класс «ClusteringDataPreprocessor» используется исключительно для подготовки данных и извлечений из них статистик, а также приведения данных к исходному масштабу после проведения кластерного анализа. Этот класс никак не участвует в процессе обучения моделей для кластеризации, что позволяет пользователю выбирать средства кластеризации. Библиотека в свою очередь предлагает пользователю несколько таких средств – наследников абстрактного класса «ClusteringEstimator», каждый из которых реализует предоставляет набор методов для обучения моделей и предсказания меток кластеров.

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

# РЕАЛИЗАЦИЯ

## Реализация модулей хранения и обработки данных

В данном разделе описаны классы, отвечающие за хранение и предобработку данных в рамках разработанной библиотеки. Эти классы обеспечивают стандартизированный способ хранения данных и применяют различные алгоритмы обработки, что позволяет более эффективно выполнять анализ пользовательского опыта.

Модуль содержит две группы программных компонентов:

1. Классы для хранения данных в стандартизированном виде
2. Классы для предобработки данных

Среди первой группы самым главным является Класс «EventFrame», предназначенный для хранения данных о событиях пользователей в виде объекта «pandas.DataFrame» и схемы названий колонок данного объекта. Данный класс также добавляет в сырые данные служебную информацию, которая потребуется для корректной работы всех остальных программных компонентов системы. Этот класс является основой всей библиотеки и позволяет создавать программные компоненты, никак не зависящие от особенностей загруженных данных. Код конструктора класса представлен в листинге 1.

Листинг 1 – Конструктор класса «EventFrame»

def \_\_init\_\_(self, data: pd.DataFrame, cols\_schema: Union[Dict[str, str], 'EventFrameColsSchema'], prepare: bool = True):

self.data = data.copy()

self.cols\_schema = EventFrameColsSchema(cols\_schema)

if prepare:

self.prepare()

else:

if cols\_schema.event\_id is None

or cols\_schema.event\_type is None

or cols\_schema.event\_type\_index is None:

raise ValueError('Data must contain event\_id , event\_type and event\_type\_index columns')

Класс «EventFrameColsSchema» отвечает за определение схемы колонок, используемых в «EventFrame». Он обеспечивает гибкость и возможность работы с различными названиями колонок, что позволяет библиотеке адаптироваться под разнообразные данные.

Вторую группу программных инструментов для предобработки данных важно сделать стандартизированной, чтобы у пользователя была возможность создавать собственные конвейеры по обработке данных и не задумываться о деталях реализации каждого конкретного обработчика. Для этой цели был разработан абстрактный класс «DataPreprocessor», служит интерфейсом для всех процессоров данных. Он определяет основные методы, которые должны быть реализованы в подклассах, и обеспечивает базовую логику для валидации параметров и подготовки данных к обработке. Код основных методов абстрактного класса представлен в листинге 2.

Листинг 2 – Код основных методов абстрактного класса «DataPreprocessor»

class DataPreprocessor(ABC):

"""

Интерфейс для данных препроцессоров.

"""

@abstractmethod

def \_\_init\_\_(self, \*\*kwargs):

pass

@abstractmethod

def apply(self, data: EventFrame) -> 'EventFrame':

некоторое преобразование к объекту EventFrame.

"""

pass

На основе абстрактного класса были разработаны несколько препроцессоров, которые позволяют извлечь дополнительную информацию из данных и предоставить пользователю больше возможностей для дальнейшего анализа:

1. Класс «AddCohortsPreprocessor». Данный класс предназначен для определения когорты каждого пользователя и указания характеристик, важных для когортного анализа, таких как когортная группа, количество периодов, прошедших с момента первого появления в продукте и т.д. для каждого действия пользователя
2. Класс «SplitSessionsPreprocessor». Этот класс предназначен для разбиения последовательности действий пользователя на сессии на основе временных интервалов или определенных событий, что позволяет более детально изучать пользовательский опыт
3. Класс «AddStartEndEventsPreprocessor». Этот класс отвечает за добавление синтетических событий начало и конца пути каждого пользователя в продукте, что в свою очередь позволяет исследовать время жизни и жизненную ценность пользователя в продукте.

В данном разделе была реализована основа библиотеки, включающая ключевые классы для хранения и предобработки данных. Эти компоненты позволяют эффективно управлять пользовательскими данными, минимизируя влияние их особенностей на функциональность системы. Благодаря созданной архитектуре, дальнейшая разработка программных компонентов может производиться без учета каких-либо нетипичных аспектов пользовательских данных.

## Реализация модуля кластеризации

В данном разделе описаны классы, отвечающие за реализацию функционала кластеризации в разработанной библиотеке. Эти классы обеспечивают подготовку данных и выбор различных средств кластеризации, что позволяет пользователям эффективно проводить анализ групп пользователей.

Важно предоставить пользователю возможность решать задачи кластеризации как встроенными методами библиотеки, так и сторонними алгоритмами, поэтому важно разделить функционал по подготовке данных к кластеризации и по подготовке моделей машинного обучения для кластеризации на независимые программные элементы.

Класс «ClusteringDataPreprocessor» предназначен для подготовки входных данных и извлечения статистической информации. Данный класс строит матрицу пользовательских характеристик, среди которых: количество совершений каждого из действий, показатели регулярности посещений, статистики сессий и жизненного пути пользователя в продукте и т.д. Итоговый формат данных: каждая строка – конкретный пользователь, каждый столбец – статистика его пользовательского опыта. Помимо составления матрицы для дальнейшей кластеризации данный класс также масштабирует ее, чтобы повысить качество кластеризации и реализует метод обратной трансформации данных для последующего кластерного анализа. Код основного метода по подготовке матрицы статистик пользователей представлен в листинге 3.

Листинг 3 – Код метода составления матрицы «ClusteringDataPreprocessor»

def create\_cluster\_matrix(self, data: EventFrame,

meta\_info: Optional[pd.DataFrame] = None,

add\_session\_stats: bool = True, add\_path\_stats: bool = True,

preprocess: bool = True) -> pd.DataFrame:

self.\_check\_input\_params(data, meta\_info)

self.data = data.data.copy()

self.cols\_schema = data.cols\_schema

self.not\_preprocess\_cols.append(self.cols\_schema.user\_id)

self.\_prepare\_raw\_cluster\_matrix(

self.data, self.cols\_schema, meta\_info,

add\_session\_stats, add\_path\_stats

)

if preprocess:

self.\_create\_preprocessor\_pipeline()

self.\_preprocess\_raw\_cluster\_matrix()

return self.cluster\_matrix

return self.raw\_cluster\_matrix

Инструменты самой кластеризации также должны быть стандартизированы, чтобы подразумевать возможность удобного сравнения различных алгоритмов по эффективности. В связи с этим был разработан абстрактный класс «ClusteringEstimator», который определяет набор методов и атрибутов для всех инструментов кластеризаци. Код основных методов класса представлен в листинге 4. Помимо абстрактных методов класс также реализует ряд методов визуализации результатов кластериации, т.к. эти методы не зависят от выбора алгоритма, а строятся только на итоговых метках класса.

Листинг 4 – Код абстрактного класса «ClusteringEstimator »

class ClusteringEstimator(ABC):

@abstractmethod

def fit(self, X: pd.DataFrame, sample\_weight):

pass

@abstractmethod

def predict(self, X: pd.DataFrame):

pass

@abstractmethod

def fit\_predict(self, X: pd.DataFrame, sample\_weight):

pass

На основе абстрактного класса были разработаны несколько инструментов кластеризации, которые позволяют не только расставить метки кластеров, но и оптимизировать параметры алгоритмов и оценить качество предсказаний моделей:

1. Класс «KMeansEstimator» реализует алгоритм кластеризации K-means. Он наследует от ClusteringEstimator и предлагает удобный интерфейс для выбора оптимального количества кластеров разными способами и описания типичных представителей каждой группы через центроиды.
2. Класс «DBScanEstimator» реализует алгоритм кластеризации DBSCAN, который не требует настройки параметра количества кластеров и позволяет определять количество выбросов в данных.

В результате реализации данного модуля разрабатываемая библиотека предоставляет возможность собирать исчерпывающую информацию о пользователях и их взаимодействиях с продуктом. Удобный интерфейс для настройки параметров алгоритмов кластеризации, а также методы для оценки их эффективности, оптимизируют и повышают качество анализа сегментов.

## Реализация модуля метрик

Продуктовые метрики представляют собой количественные показатели, которые помогают оценивать эффективность продукта, они играют ключевую роль в принятии обоснованных бизнес-решений.

Анализ продуктовых метрик почти всегда сводится к оценке ключевых показателей эффективности (KPI) или к динамическому анализу метрик во временном ряду. KPI помогают определить, достигнуты ли поставленные цели, в то время как анализ метрик в динамике позволяет увидеть изменения и тренды в поведении пользователей за определённые периоды времени. Это критически важно для понимания изменений в пользовательском опыте и для адаптации бизнес-стратегий.

Класс «MetricKPI» и «MetricDinamic» в рамках модуля метрик предоставляют инструменты для работы с метриками. Каждая из метрик может быть рассчитана с разбиением по любому количеству признаков, что позволяет пользователям гибко настраивать анализ под свои потребности.

Класс «MetricKPI»: позволяет рассчитывать KPI с возможностью разбиения по нескольким признакам. Он включает методы для получения данных, вычисления значений и создания шаблона сводной таблицы для дальнейшего анализа. Он может быть использован для более сложных и глубоких аналитических исследований, таких как когортный анализ. В этом случае можно рассчитать кастомные метрики для каждой когорты, что улучшит понимание поведения пользователей.

Класс «MetricDinamic»: этот класс специализирован для анализа временных рядов и позволяет рассчитывать метрики по данным за различные временные интервалы. Для него также будут разработаны инструменты для анализа эконометрических показателей, таких как тренды и сезонность, а также для выявления аномалий в данных.

Таким образом, реализованные в модуле метрик классы обеспечивают возможность преобразования любой пользовательской функции обработки данных в стандартизированную метрику с расширенными характеристиками для детального анализа. Благодаря этой стандартизированности метрики могут быть эффективно использованы в других модулях библиотеки, становясь основой для более сложных анализов и концепций взаимодействия с пользователями.

## Реализация модуля дополнительных средств анализа

Модуль дополнительных средств анализа был разработан с целью расширения функциональности библиотеки, предоставляя мощные инструменты для углубленного исследования пользовательского опыта. Все реализованные компоненты этого модуля базируются на ранее разработанных фундаментальных элементах библиотеки, таких как хранилища данных, процессоры и метрики. Эти элементы обеспечивают надежную архитектуру для обработки и анализа больших объемов данных.

Данный модуль предназначен для оптимизации и автоматизации существующих методов продуктового анализа, таких как анализ продуктовых воронок и когортный анализ. Эти инструменты позволяют компаниям более эффективно отслеживать и понимать поведение пользователей, улучшая тем самым результаты бизнеса. Опираясь на уже имеющееся функциональное ядро, модуль способствует более глубокому анализу взаимодействия пользователей с продуктом и предоставляет новые возможности для принятия обоснованных решений.

### Реализация инструментов для расчета продуктовых воронок

Продуктовая воронка — это визуализация этапов, которые проходит пользователь от первоначального взаимодействия с продуктом до достижения ключевой цели, такой как покупка, подписка или другое целевое действие. Она помогает компаниям понять, на каких стадиях пользователи теряются и где возможны улучшения. Воронка может быть рассчитана с использованием различных метрик, таких как количество пользователей на каждом этапе воронки и процентное соотношение пользователей, которые переходят от одной стадии к другой. Основной метрикой является число уникальных пользователей на каждом этапе, а также процентное отношение между ними, что позволяет оценивать эффективность каждой стадии и выявлять узкие места.

В разработанной библиотеке класс «Funnel» служит основным инструментом для анализа продуктовых воронок. Он позволяет рассчитывать два вида воронок:

1. Открытая воронка (open): позволяет отслеживать пользователей на каждом этапе без необходимости достижения целевого действия.
2. Закрытая воронка (closed): подразумевает строгое следование пользователями через все этапы до достижения конечной цели.

Для каждого типа воронки пользователи могут быть разделены на сегменты, также в качестве этапов могут быть переданы любые комбинации пользовательских действий. Любую воронку также можно отобразить в виде интерактивного графика, на котором будут отображены все ключевые для анализа статистики.

### Реализация инструментов для когортного анализа

Когортный анализ — это метод, используемый для исследования и сравнения поведения пользователей, сгруппированных по общему признаку, времени первой активности. Когортный анализ позволяет компаниям глубже понять пользовательский опыт, отслеживая, как изменение продукта или маркетинга влияет на поведение определённых групп пользователей. Когортный анализ может быть выполнен через несколько метрик, таких как retention rate, churn rate и другие ключевые показатели эффективности (KPI). Основным шагом является создание когортной таблицы, где каждая когорта отображает поведение пользователей на протяжении времени.

В реализованной библиотеке класс «Cohorts» служит основным инструментом для выполнения когортного анализа. Реализация класса ориентирована на гибкость и возможность адаптации под различные виды когортного анализа. Он позволяет пользователям рассчитывать когортные таблицы с использованием различных пользовательских настроек и метрик (экземпляров класса «MetricKPI»). Основной функционал класса:

1. Функция «fit»: Основной метод, который собирает когортные данные и формирует стандартную когортную таблицу, которая отображает количество пользователей в разные периоды и позволяет оценивать retention rate и Charn rate.
2. Функция «fit\_by\_custom\_kpi»: Позволяет проводить анализ по пользовательским метрикам KPI, что делает класс универсальным для различных типов анализа.
3. Функция «plot»: обеспечивает графическое представление когортных данных, что делает анализ более наглядным и доступным.

# ТЕСТИРОВАНИЕ

Для проверки работоспособности библиотеки, необходимо провести тестирование. Чтобы проверить соответствие заявленному функционалу было выполнено функциональное тестирование, направленное на проверку соответствия работы библиотеки всем указанным требованиям.

Результаты тестирования представлены в таблице 4.

Таблица 4 – Результаты функционального тестирования

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| № теста | Название теста | Шаги | Ожидаемый результат | Тест пройден |
| 1 | Тест подготовки данных для кластеризации | Создать экземпляр класса «ClusteringDataPreprocessor» и вызывать метода create\_cluster\_matrix с различными комбинациями параметров. | Матрица кластеров формируется корректно для всех комбинаций параметров. | Да |
| 2 | Тест корректной работы препроцессоров по отдельности и в рамках конвейера | Создание экземпляров всех реализованных препроцессоров и применение их к набору данных по отдельности и в формате конвейера | В данных происходят корректные преобразования и генерируются синтетические поля и строки в обоих форматах применения | Да |
| 3 | Тест инициализации и подготовки EventFrame | 1. Создать экземпляр класса EventFrame с валидными данными. | Экземпляр класса создается без ошибок, все необходимые колонки правильно инициализируются. | Да |
| 4 | Тест работы алгоритма KMeans DBSCAN с разными параметрами | 1. Создать экземпляры классов наследников «ClusteringEstimator» с различными и обучить на нескольких наборах данных | Алгоритмы успешно обучается для всех заданных параметров без ошибок. | Да |

Окончание таблицы

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 5 | Тест подбора оптимального количества кластеров | Создать экземпляр класса «KMeansEstimator» и вызвать метод «choose\_optimal\_n\_clusters» с разными диапазонами кластеров. | Метод выбирает оптимальное количество кластеров корректно и отображает результаты. | Да |
| 6 | Тест визуализации результатов кластеризации | Создать экземпляр класса кластеризации и обучить его, затем вызвать методы plot\_clusters\_counts и plot\_clusters\_with\_tsne. | Dолжен отображаться график с количеством объектов в каждом кластере и точечная диаграмма меток кластера. | Да |
| 7 | Тест инверсии преобразования для кластерных данных | 1. Создать экземпляр класса «ClusteringDataPreprocessor» и вызвать метод inverse\_transform() с подготовленными данными. | Данные возвращаются в первоначальном формате и масштабе, включая не обработанные колонки и корректные значения. | Да |

**Вывод**

В данной главе было проведено функциональное. Все тесты показали, что система функционирует согласно требованиям.

ЗАКЛЮЧЕНИЕ

В соответствии с целями данной работы был проведен обзор научной литературы и существующих аналогов, который подтвердил актуальность поставленной задачи. Разработан прототип библиотеки для анализа пользователей в ИТ продукте. Для достижения поставленной цели решены следующие задачи:

1. провести анализ литературы;
2. спроектировать прототип библиотеки;
3. разработать прототип библиотеки;
4. осуществить тестирование реализованных модулей.

В процессе работы было создано ядро библиотеки, которое станет основой для дальнейшего расширения функционала и интеграции новых инструментов.

В планах на будущее – реализация инструментов для детального изучения тенденций пользовательского поведения, а также разработка средств для анализа эконометрических показателей динамических метрик и поиска аномалий в них. Эти дополнения позволят значительно повысить точность анализа и обогатят возможности пользователей в исследовании и оптимизации взаимодействия с продуктами.

.

ЛИТЕРАТУРА

1. «Sweetviz» documentation [Электронный ресурс] URL: https://pypi.org/project/sweetviz/1.0b6/ (дата обращения 05.02.2025 г.).
2. «Klib» documentation [Электронный ресурс] URL: https://klib.readthedocs.io/en/stable/ (дата обращения 05.02.2025 г.).
3. «Retentioneering» documentation [Электронный ресурс] URL: https://doc.retentioneering.com/stable/doc/ (дата обращения 05.02.2025 г.).
4. «Pandas» documentation [Электронный ресурс] URL:   
   https://pandas.pydata.org/pandas-docs/ (дата обращения 05.02.2025 г.).
5. «Numpy» documentation [Электронный ресурс] URL:   
   https://numpy.org/ (дата обращения 05.02.2025 г.).
6. «Matplotlib» documentation [Электронный ресурс] URL:   
   https://matplotlib.org/ (дата обращения 05.02.2025 г.).
7. «Seaborn» documentation [Электронный ресурс] URL: https://seaborn.pydata.org/ (дата обращения 05.02.2025 г.).
8. «Plotly» documentation [Электронный ресурс] URL:  [https://plotly.com/python](%20https://react.dev/) (дата обращения 05.02.2025 г.).
9. «Statmodels» documentation [Электронный ресурс] URL: https://www.statsmodels.org/stable/ (дата обращения 05.02.2025 г.).
10. «Scikit-Learn» [Электронный ресурс] URL:   
    [https://scikit-learn.org/](https://scikit-learn.org/%20) (дата обращения 05.02.2025 г.).
11. «SciPy» documentation [Электронный ресурс] URL: https://scipy.org/ (дата обращения 05.02.2025 г.).