

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное бюджетное образовательное учреждение высшего образования
«Тольяттинский государственный университет»

Кафедра _____ «Прикладная математика и информатика» _____
(наименование)

01.03.02 Прикладная математика и информатика
(код и наименование направления подготовки / специальности)

Компьютерные технологии и математическое моделирование
(направленность (профиль) / специализация)

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА (БАКАЛАВРСКАЯ РАБОТА)

на тему «Применение компьютерного моделирования в разработке и оптимизации функциональности корпоративного сайта»

Обучающийся _____ А. А. Марков _____
(Инициалы Фамилия) (личная подпись)

Руководитель _____ канд.пед.наук, доцент, О.М. Гущина _____
(ученая степень (при наличии), ученое звание (при наличии), Инициалы Фамилия)

Консультант _____ канд.пед.наук, доцент, А. В. Егорова _____
(ученая степень (при наличии), ученое звание (при наличии), Инициалы Фамилия)

Тольятти 2024

Аннотация

Тема бакалаврской работы: «Применение компьютерного моделирования в разработке и оптимизации функциональности корпоративного сайта».

Бакалаврская работа посвящена процессам разработки и оптимизации функциональности корпоративного сайта, а также применения в этих процессах компьютерного моделирования.

В ходе выполнения исследований по бакалаврской работе был описан процесс создания корпоративных сайтов, а также произведён обзор и выбор компьютерной модели и рассмотрено практическое применение компьютерной модели в разработке и оптимизации функциональности корпоративного сайта. Был выбран способ оценки оптимизации сайта и описана его математическая модель.

Во введении прописывается актуальность темы, написаны цель и задачи.

В первом разделе рассматривается предметная область исследования и проводится анализ компьютерных моделей.

Во втором разделе произведен обзор существующих способов оценки оптимизации корпоративного сайта и выбор метода линейной регрессии, также мы описали и создали архитектуру системы и спроектировали систему.

В третьем разделе была описана реализация ПО, а также произведено тестирование и анализ полученных результатов.

В заключении представлены результаты выполнения бакалаврской работы.

Бакалаврская работа состоит из введения, трёх разделов, заключения и списка использованной литературы.

Бакалаврская работа состоит из 43 страниц, 15 рисунков, 3 таблиц, 28 источников и 2 листингов.

Abstract

Topic of the bachelor's thesis: "Application of computer modeling in the development and optimization of the functionality of a corporate website."

The bachelor's thesis is devoted to the processes of developing and optimizing the functionality of a corporate website, as well as the use of computer modeling in these processes.

During the bachelor's work, the process of creating corporate websites was described, as well as a review and selection of a computer model was made and the practical application of a computer model in the development and optimization of the functionality of a corporate website was considered. A method for assessing site optimization was chosen and its mathematical model was described.

The introduction states the relevance of the topic, the purpose and objectives are written.

The first section examines the subject area of the study and analyzes computer models.

In the second section, we reviewed existing methods for assessing the optimization of a corporate website and selected the linear regression method; we also described and created the system architecture and designed the system.

The third section described the implementation of the software, as well as testing and analysis of the results obtained.

In conclusion, the results of the bachelor's work are presented.

The bachelor's thesis consists of an introduction, three sections, a conclusion and a list of references.

The bachelor's thesis consists of 43 pages, 15 figures, 3 tables, 28 sources and 2 listings.

Содержание

Введение.....	5
1 Описание компьютерной модели в разработке и оптимизации функциональности корпоративного сайта.....	8
1.1 Постановка задачи и описание процесса разработки корпоративных сайтов.....	8
1.2 Обзор и выбор компьютерной модели.....	9
1.3 Применение компьютерной модели в разработке корпоративного сайта	11
1.4 Применение компьютерной модели в оптимизации функциональности корпоративного сайта	13
2 Применение компьютерной модели в разработке и алгоритм оптимизации функциональности корпоративного сайта.....	18
2.1 Описание методов анализа данных для оптимизации корпоративных сайтов.....	18
2.2 Математическая модель метода линейной регрессии.....	19
2.3 Применение диаграммы Ганта в разработке сайта.....	23
2.4 Разработка системы для дальнейшего написания программного кода .	24
3 Реализация и тестирование программного обеспечения	28
3.1 Реализация программного обеспечения	28
3.2 Тестирование программного обеспечения	33
Заключение	37
Список используемой литературы	38
Приложение А Листинг (реализация графика продаж)	40
Приложение Б Листинг (реализация графика линейной регрессии)	42

Введение

С развитием информационных технологий и ростом конкуренции в сфере онлайн-бизнеса актуальной задачей становится разработка и оптимизация функциональности корпоративного сайта с помощью компьютерного моделирования. Для успешного решения этой задачи необходимо провести исследования, направленные на разработку инновационных методов моделирования и оптимизации, которые учтут особенности бизнеса и потребности пользователей [1]. Актуальность данной темы обусловлена возможностью создания новых подходов к анализу и улучшению пользовательского опыта на корпоративных сайтах.

Анализируя текущее состояние исследований в данной области, мы можем отметить, что она находится в активном развитии и демонстрирует значительный прогресс. Изучение различных источников по данной проблеме позволяет отметить, что аналитики и разработчики корпоративных сайтов активно работают над разработкой новых технологий, методов и подходов для оптимизации и улучшения пользовательского опыта на корпоративных сайтах [2].

В настоящее время применение компьютерного моделирования становится все более популярным инструментом для анализа и проектирования функциональности корпоративных сайтов. Это позволяет разработчикам предварительно оценить эффективность и эргономичность сайта, а также выявить и исправить возможные проблемы до его запуска [3].

Однако, несмотря на достигнутые результаты, можно сделать вывод, что данная тема еще не полностью раскрыта и нуждается в дальнейшей разработке. Существующие исследования изучают различные аспекты применения компьютерного моделирования в разработке корпоративных сайтов, но остаются некоторые нерешенные проблемы.

Прежде всего, существует необходимость в разработке более точных и реалистичных моделей, которые бы учитывали не только функциональность,

но и визуальный аспект, удобство использования и эффективность корпоративных сайтов. Такие модели должны принимать во внимание разнообразные аспекты, включая поведение пользователей, их предпочтения, контекст использования и т.д. [4].

Кроме того, необходимо разработать стандарты и рекомендации для использования компьютерного моделирования в разработке корпоративных сайтов, чтобы обеспечить единый подход и согласованность при проектировании и оптимизации функциональности сайтов. Это позволит повысить эффективность использования компьютерного моделирования в практической работе и достичь более качественных результатов.

Таким образом, следует отметить, что применение компьютерного моделирования в разработке и оптимизации функциональности корпоративного сайта является активной и перспективной областью исследований, но требует дальнейшего развития и углубленного изучения для достижения более полного понимания и эффективного применения в практике [5].

Объектом исследования бакалаврской работы является разработка компьютерной модели для оптимизации функциональности корпоративного сайта.

Предметом исследования – компьютерная модель, алгоритмы, библиотеки, которые будут использованы для разработки корпоративного сайта.

Цель работы - разработка компьютерной модели для оптимизации функциональности корпоративного сайта, а также для развития деятельности компании и оценка её эффективности в предоставлении пользователям точной и актуальной информации сайта.

Задачи бакалаврской работы включают в себя:

- Анализ и обзор существующих методов компьютерного моделирования и их применимости в разработке и оптимизации функциональности корпоративных сайтов.

- Изучение требований и целей компании, связанной с корпоративным сайтом, для определения функциональности, которая должна быть реализована.
- Разработка компьютерной модели для корпоративного сайта, включающей в себя структуру сайта, основные функции и взаимодействия.
- Создание интерактивной симуляции на основе компьютерной модели, позволяющей визуализировать и тестировать функциональность сайта в различных сценариях использования.
- Проектирование и оптимизация функциональности сайта с использованием компьютерного моделирования, включая поиска узких мест, анализ пользовательского опыта и предложение улучшений.
- Практическое тестирование разработанной функциональности сайта с использованием реальных данных и пользователей, с целью проверки эффективности и работоспособности.
- Сравнение результатов компьютерного моделирования с существующими аналогами.
- Анализ эффективности применения компьютерного моделирования в разработке и оптимизации функциональности корпоративных сайтов.

В первом разделе рассматривается постановка задачи на работу, а также процесс создания корпоративных сайтов. Далее происходит выбор компьютерной модели и её применение в разработке и оптимизации функциональности корпоративных сайтов. Во втором разделе, описана математическая модель метода линейной регрессии. Также была спроектирована система для разработки и архитектура системы. В третьем разделе, описан процесс реализации программного обеспечения, и тестирования ПО.

1 Описание компьютерной модели в разработке и оптимизации функциональности корпоративного сайта

1.1 Постановка задачи и описание процесса разработки корпоративных сайтов

Основной задачей данной работы является оптимизация корпоративного сайта с применением компьютерного моделирования. Компьютерная модель будет рассчитывать необходимые данные на основе предпочтения пользователей и визуализировать полученный запрос.

Цель бакалаврской работы – расширить функционал сайта, посредством добавления компьютерной модели, использующей сложные математические преобразования на основе введённых данных от пользователей.

Корпоративный сайт представляет собой веб-пространство, где компания может представить свою деятельность, продукты, услуги и контактную информацию. Корпоративный сайт может содержать различные разделы, такие как новости, пресс-релизы, блог, вакансии и т.д. В зависимости от целей компании, на сайте могут быть представлены различные разделы и страницы. Корпоративный сайт помогает компании установить контакт с потенциальными клиентами, а также предоставляет информацию для существующих клиентов [6].

Процесс разработки корпоративных сайтов - это последовательность шагов и этапов, направленных на создание веб-присутствия для компании или организации с целью представления ее деятельности, продуктов или услуг в онлайн-среде. Это комплексный процесс, который подразумевает делегирование в процессе разработки на команду специалистов, каждый из которых занимается определённым делом. Рассмотрим процесс создания сайта (таблица 1).

Таблица 1 – План создания и разработки корпоративного сайта

Название	Исполнитель	Описание действия
Планирование	Все участники	Определение целей и задач, исследование ЦА (кто будет посещать сайт)
Дизайн	Веб-Дизайнер	Создание дизайна сайта, выбор цвета, шрифтов, расположение блоков с информацией, и т.д.
Вёрстка	Специалист по вёрстке	Описание дизайна на языке HTML/CSS для дальнейшего добавления функционала.
Разработка	Отдел программистов	Программирование сайта, добавление функциональности и контента.
Тестирование	Отдел тестировщиков	Проверка работоспособности сайта, на всех устройствах, платформах, браузерах.
Запуск	Отдел программистов	Размещение сайта на хостинге, а также регистрация доменного имени, провести SEO-оптимизацию для улучшения позиций в поисковых запросах.
Поддержка	Отдел программистов	Обновление информации, исправление ошибок, обеспечение безопасности сайта, и дальнейшая оптимизация производительности.

На основе таблицы 1 можно сделать вывод, что разработка корпоративного сайта - это комплексный процесс. Для того, чтобы увеличить КПД и ускорить разработку корпоративного сайта необходимо использовать компьютерную модель.

1.2 Обзор и выбор компьютерной модели

В бакалаврской работе я буду использовать несколько компьютерных моделей для того, чтобы ускорить и оптимизировать процесс создания корпоративного сайта, а также для оптимизации функциональности корпоративного сайта. Однако необходимо выбрать подходящие для этих целей компьютерные модели. Для этого проведем сравнительный анализ компьютерных моделей (таблица 2).

Рассмотрим лишь некоторые виды компьютерных моделей, и каждый из них имеет свои достоинства и применения в различных областях знаний и исследований.

Таблица 2 – Сравнительный анализ компьютерных моделей

Компьютерная модель	Описание модели	Недостатки модели	Преимущества модели
Физические модели	Математическое описание реальной системы путем использования физических законов и уравнений.	Дорогостоящие; Сложные в создании и обслуживании.	Точное представление Объекта или процесса; возможность проведения физических экспериментов.
Статистические модели	Основаны на статистическом анализе данных и вероятностных методах и моделируют случайные процессы и систем с неопределенностью	Требуют большого объема данных для анализа; Могут быть неточными.	Учитывают вероятностные закономерности; Могут использоваться для прогнозирования
Аналитические модели	Основаны на формальном анализе математических уравнений и моделей, и получают аналитические решения и выводы о системе, исходя из известных уравнений и параметров	Требуют точной мат. формализации; Не всегда могут учесть все аспекты реального объекта.	Математический анализ, быстрые результаты; Возможность проведения расчетов различной сложности.
Имитационные модели	Создают компьютерную модель, которая имитирует поведение какой-либо системы или процесса	Требуют больших вычислительных ресурсов; Могут быть долгими в вычислениях.	Позволяют моделировать сложные системы и процессы; Помогают проводить различные сценарии.

На основе сравнительного анализа можно сделать вывод, что наиболее подходящими являются аналитические и статистические модели, которые я буду использовать в своей работе. Аналитические модели будут применяться в разработке сайта, а статистические компьютерные модели будут использованы мной для оптимизации функциональности корпоративного сайта.

1.3 Применение компьютерной модели в разработке корпоративного сайта

При разработке корпоративных сайтов возникает ряд проблем, с которыми сталкивается команда разработчиков и с некоторыми из них поможет справляться компьютерная модель [7]. Рассмотрим самые основные проблемы, с которыми сталкивается команда разработчиков:

- Проектирование удобного и интуитивно понятного пользовательского интерфейса.
- Адаптивный дизайн для разных платформ.
- Безопасность сайта.
- Интеграция с системами управления контентом.
- Поддержка различных языков и локализация.
- Продвижение сайта в поисковых системах.
- Поддержка и обновление.

Однако главной проблемой в разработке сайта, с которым сталкиваются специалисты – это эффективное управление проектом. Разработка корпоративного сайта является комплексным процессом, требующим хорошего управления проектом. Разработчики должны тщательно планировать, определять цели и требования проекта, назначать роли и ответственности, следить за сроками и ресурсами, а также обеспечивать своевременную коммуникацию с заказчиком и другими заинтересованными сторонами [8].

Мною будет создана компьютерная модель для того, чтобы минимизировать и оптимизировать время, потраченное на все стадии разработки сайта, а также для ведения отчётности, указания дедлайнов и распределение задач в разработке корпоративного сайта.

Эта компьютерная модель является аналитической и представляет собой линейную гистограмму, в которой будут отображаться даты, каждый этап разработки с указанием сроков выполнения, и времени, потраченного на

каждый этап. Она поможет ускорить и оптимизировать процесс разработки корпоративного сайта. Самый распространенный вариант подобной оптимизации рабочего процесса является диаграмма Ганта. Пример такой диаграммы представлен ниже (рисунок 1).



Рисунок 1 – Пример плана-реализации проекта

Из рисунка 1 видно, что такая диаграмма отлично подойдет для команды разработчиков, в которой каждый работает над поставленной задачей и некоторые процессы происходят параллельно друг другу.

1.4 Применение компьютерной модели в оптимизации функциональности корпоративного сайта

Компьютерные модели могут быть очень полезными инструментами для оптимизации функциональности корпоративного сайта. Рассмотрим основные способы применения компьютерных моделей в оптимизации функциональности корпоративного сайта:

- Анализ поведения пользователей.
- Проверка изменений на сайте до и после внедрения изменений.
- Анализ ключевых слов и структуры ссылок.
- Планирование ресурсов и масштабирование сайта.
- Создание персонализированного контента для улучшения пользовательского опыта [9].

В моей работе я буду применять статистические компьютерные модели, которые будут представлены в виде графиков. Корпоративный сайт организации, в которой я прохожу преддипломную практику, специализируется на продаже строительных блоков. В организации передо мной была поставлена задача составить график продаж каждого блока по отдельности. В виду этого первая модель будет являться графиком продаж по месяцам.

В первой компьютерной модели будут представлены продажи ассортимента. Вертикальная ось будет количеством проданных строительных блоков, а горизонтальная ось - временной промежуток продаж по месяцам. Она будет включать в себя семь графиков разных цветов для указания каждого вида строительных блоков по отдельности. Это необходимо для выявления пика продаж, сезонности, а также для отслеживания динамики выручки и чистой прибыли по всему ассортименту.

Основные виды блоков: стеновые, фундаментные и вентиляционные. Их графики продаж будут представлены в виде компьютерной модели. Перед тем, как внедрять что-то в сайт необходимо разработать дизайн того, как это

выглядит. Для того, чтобы разработать дизайн будущего графика продаж мною было использовано бесплатное приложение для графического дизайна Figma. Результат дизайна приведён ниже (рисунок 2).

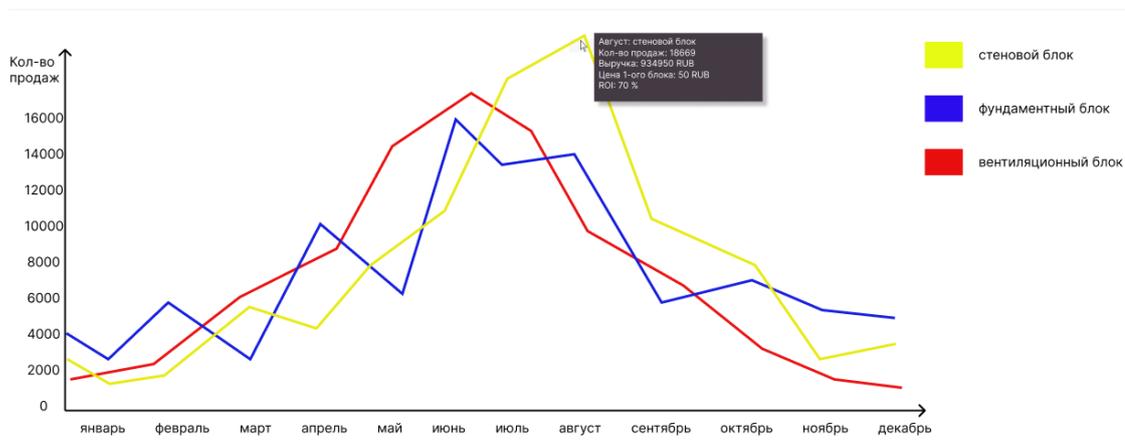


Рисунок 2 – Прототип модели графика продаж

На основе этого графика продаж сотрудники компании смогут визуально определять наиболее выгодные для продажи товары и планировать дальнейшее производства исходя из рентабельности вложений.

Еще одной задачей, которая передо мной поставлена является оптимизация производительности корпоративного сайта, потому что оптимизированный веб-сайт значительно улучшает пользовательский опыт, что приводит к увеличению посещаемости и удержанию посетителей сайта. Кроме того, быстрый веб-сайт помогает улучшить рейтинг в результатах поиска, что приводит к увеличению органического трафика и повышению конверсии.

Чтобы измерить производительность вашего веб-сайта, первым делом нужно получить базовый показатель. Ключевыми показателями оптимизации производительности веб-сайта являются время загрузки страницы, время до первого байта (TTFB) и время взаимодействия (TTI) [10].

Время загрузки страницы — это широко используемый и простой для понимания показатель. Он измеряет время, необходимое веб-странице для

полной загрузки и отображения пользователю в браузере. Время загрузки страницы включает в себя время, необходимое для загрузки с сервера всех необходимых ресурсов, таких как HTML, CSS, JavaScript, изображений и других элементов, и их отображения на устройстве [11].

Время до первого байта (TTFB) — это время, необходимое браузеру для получения первого байта данных с сервера после отправки запроса. Чем ниже TTFB, тем быстрее отвечает сервер, что, в свою очередь, гарантирует лучшее взаимодействие с пользователем. TTFB рассчитывается по формуле (формула 1) [12]:

$$\text{TTFB} = \text{TTS} - \text{TTCP} \quad (1)$$

где

TTS (Time to Start) – время начала передачи данных с сервера.

TTCP (Time to TCP) - время установки TCP-соединения.

Время до взаимодействия (TTI) измеряет, сколько времени требуется веб-странице, чтобы стать полностью интерактивной и реагировать на действия пользователя. Это тот момент, когда пользователи могут эффективно взаимодействовать с сайтом и получать необходимую им информацию. TTI рассчитывается по формуле (формула 2) [13]:

$$\text{TTI} = \text{FCP} + \text{DCL} + \text{LCP} \quad (2)$$

где

FCP (First Contentful Paint) – время, за которое браузер отобразил первый контент на странице.

DCL (Dom Content Loaded) – время, за которое браузер завершил загрузку DOM.

LCP (Largest Contentful Paint) – время, за которое браузер отобразил самый большой контент на странице.

Для наглядного примера, я нарисовал принцип расчёта ТТІ, основываясь на открытых источниках литературы (рисунок 3).

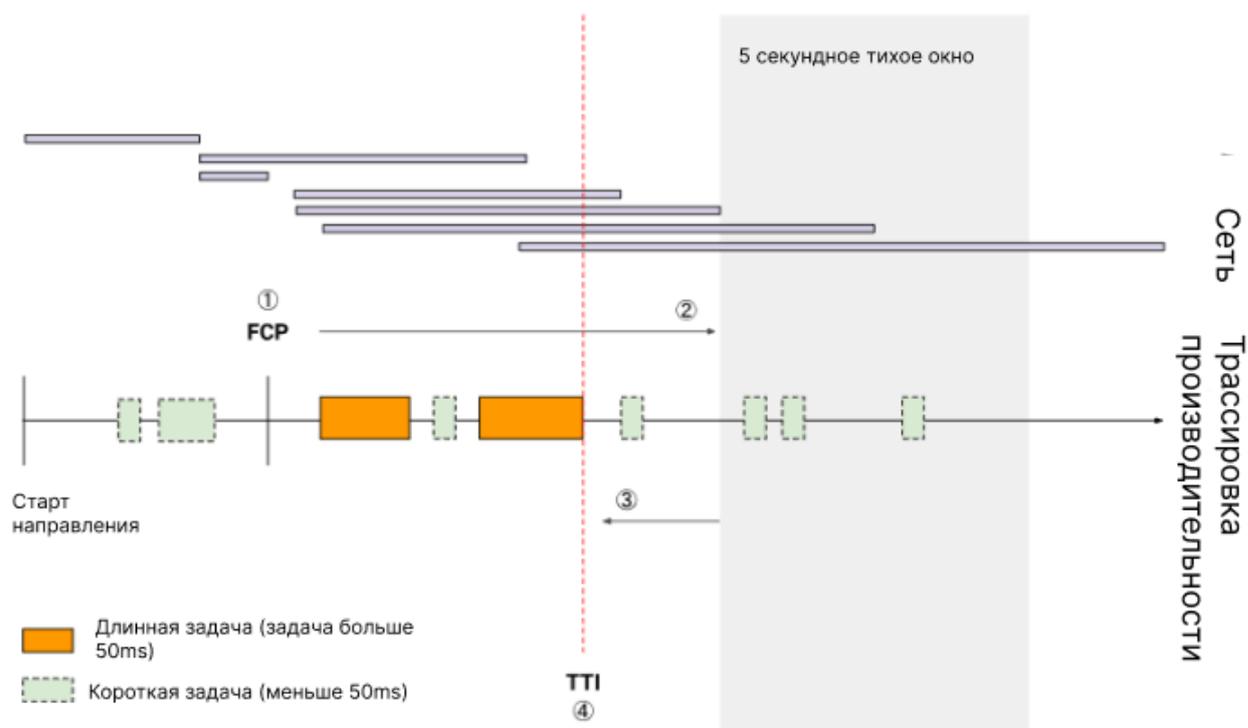


Рисунок 3 – Принцип расчёта ТТІ

Для описания диаграммы я выделю 4 основных шага:

- Начало первой контентной отрисовки (FCP).
- Нахождение тихого окна (окно больше 5 секунд, определяется отсутствием задач за этот промежуток времени).
- Поиск длинной задачи в обратном направлении (если она отсутствует, останавливаемся на FCP).
- ТТІ (время окончания длинной задачи перед тихим окном(или тоже значение что и FCP, если длинные задачи не были найдены).

Вторая модель является анализом функциональности и оптимизации сайта. Вертикальная ось – это количество пользователей на сайте, а горизонтальная ось – это время до первого байта в мс. На этой модели будет отображаться изменение производительности сайта для отслеживания

влияния наших действий на функциональность и оптимизацию сайта.

Для нахождения этих двух показателей я буду использовать языки программирования такие как: PHP и Python. В Python будет использована библиотека Selenium WebDriver для тестирования производительности веб-страниц, а также для измерения TTI. В PHP будут использованы библиотеки Guzzle для измерения TTFB при обращении к серверу.

Выводы по разделу 1:

Изучив все справочные материалы, мной была поставлена задача, применить компьютерные модели в разработке и оптимизации функциональности корпоративного сайта.

Будут созданы три компьютерные модели. Первая будет эффективно распределять время, затраченное на каждый этап разработки. Вторая будет расширять функциональность сайта. Третья будет отслеживать производительность и влияние наших действий для оптимизации корпоративного сайта.

2 Применение компьютерной модели в разработке и алгоритм оптимизации функциональности корпоративного сайта

2.1 Описание методов анализа данных для оптимизации корпоративных сайтов

Алгоритм оптимизации играет ключевую роль в успешной работе корпоративного сайта. Стоит учитывать, то, что корпоративный сайт, как правило является визитной карточкой компании, и его эффективность напрямую влияет на бизнес-результаты организации [14].

Вот несколько причин, почему так важен алгоритм оптимизации для корпоративного сайта:

- Улучшение видимости в поисковых системах.
- Повышение конверсии.
- Улучшение пользовательского опыта.
- Конкурентное преимущество.

Для того, чтобы эффективно отследить и визуализировать скорость загрузки сайта в зависимости от нагрузки на домен сайта необходимо применить один из алгоритмов, который анализирует эти данные и предоставляет наглядную статистику [15].

Существует множество методов анализа данных, которые можно использовать для оптимизации корпоративных сайтов. Рассмотрим актуальные методы, которые можно использовать и выявим наиболее эффективный и доступный метод в нашем случае на основе сравнительного анализа различных методов анализа данных корпоративных сайтов (таблица 3).

Таблица 3 – Сравнительный анализ методов анализа данных для оптимизации корпоративного сайта

Метод анализа данных	Описание метода	Недостатки метода	Преимущества метода
А/В тестирование	Сравнивает две или более версии веб-страницы (или ее элементов) с целью определения, какая из них лучше работает и приводит к более высокой конверсии [16].	Требует времени и ресурсов для проведения тестов; Сложно интерпретировать результаты при наличии множественных переменных.	Позволяет сравнивать эффективность различных вариантов веб-страницы; Помогает определить, какие изменения приводят к улучшению конверсии.
Метод линейной регрессии	Моделирует зависимости между переменными и предсказания значений целевой переменной на основе значений других переменных [17].	Предполагает линейную зависимость между переменными; Чувствителен к выбросам в данных.	Прост в интерпретации результатов. Позволяет моделировать зависимости между переменными.
Кластерный анализ	Группирует пользователей сайта на основе их поведения, что помогает создать стратегии улучшения пользовательского опыта [18].	Требует определения количества кластеров; Результаты могут быть интерпретированы по-разному.	Помогает выявить сегменты пользователей с похожим поведением; Позволяет создавать персонализированные стратегии.
Прогнозирование временных рядов	Прогнозирует будущие значения метрик сайта на основе исторических данных, что помогает в планировании маркетинговых кампаний и стратегий [19].	Сложно учесть не стационарность временного ряда. Требует аккуратного выбора модели и параметров.	Позволяет предсказывать будущие значения; Помогает в планировании бизнес-стратегий.

При исследовании основных методов анализа данных я остановился на выборе метода линейной регрессии, так как считаю, что он лучше всего подходит для реализации моей задачи. На основе метода линейной регрессии.

2.2 Математическая модель метода линейной регрессии

Принцип действия линейной регрессии заключается в поиске линейной

зависимости между одной или несколькими независимыми переменными (факторами) и зависимой переменной. Целью линейной регрессии является построение модели, которая наилучшим образом описывает эту зависимость и позволяет предсказывать значения зависимой переменной на основе значений независимых переменных [20].

Путем использования модели линейной регрессии мы можем выявить и оценить взаимосвязь между зависимой переменной и одной или более независимыми переменными. Целью модели линейной регрессии является построение аппроксимации связи между независимой переменной x и зависимой переменной y линейной функцией [21]. Математически данную связь можно выразить уравнением (формула 3):

$$y_i = \beta_0 + \beta_1 * x_i + \varepsilon_i \quad (3)$$

где

y_i – наблюдаемое значение зависимой переменной для i -го наблюдения.

x_i – значение независимой переменной для i -го наблюдения.

β_0 - коэффициент сдвига (интерсепт), который представляет собой значение зависимой переменной y при $x=0$.

β_1 – коэффициент наклона, который представляет собой изменение в y , вызванное изменением в x .

ε_i - случайная ошибка (остаток) для i -го наблюдения, которая представляет расхождение между наблюдаемым и предсказанным значением y [22].

Цель состоит в том, чтобы подобрать такие значения коэффициентов наклона и сдвига, которые минимизируют сумму квадратов остатков (расхождений между наблюдаемыми и предсказанными значениями) [23].

В своей ВКР я буду применять модель линейной регрессии для построения графика загруженности сайта. Все данные будут извлекаться из csv файла, в котором указаны столбцы количества пользователей на сайте и

времени загрузки сайта. В таком случае, исходя из формулы 3, наблюдаемое значение зависимой переменной для i -го наблюдения y_i будет количество пользователей на сайте, а значение независимой переменной для i -го наблюдения x_i будет время загрузки страницы корпоративного сайта.

Коэффициент β_0 называется свободным членом или константой регрессии. Он отображает точку пересечения с осью координат Y [24]. Это значение, которое принимает зависимая переменная, когда независимая равна нулю (формула 4):

$$y = \beta_0 + \beta_1 * 0 = \beta_0 \quad (4)$$

где

y – наблюдаемое значение зависимой переменной.

β_0 - коэффициент сдвига.

β_1 – коэффициент наклона.

Параметр β_1 является коэффициентом наклона и показывает во сколько раз значение будет увеличиваться по оси y при смещении на единицу по оси x . Если коэффициент положительный, то по мере увеличения независимой переменной будет увеличиваться также зависимая переменная y . В случае, если коэффициент отрицательный, это означает, что увеличение независимой переменной x приводит к уменьшению зависимой переменной y [25].

Метод наименьших квадратов заключается в том, чтобы сумма квадратов отклонений отдельных значений зависимой переменной от регрессионной линии была минимальной [26].

Разность между наблюдаемыми и предсказанными значениями может быть как положительной (если предсказанное значение меньше наблюдаемого), так и отрицательной (если предсказанное значение больше наблюдаемого). Однако, если сложить все разности, положительные и отрицательные значения взаимно компенсируются. Чтобы этого избежать, разница возводится в квадрат. Цель этого метода состоит в том, чтобы

минимизировать сумму квадратов остатков RSS. Таким образом, сумма остаточных квадратов вычисляется по формуле (формула 5):

$$RSS = \sum_{i=1}^n \epsilon_i^2 = \sum_{i=1}^n (y_i - \beta_0 - \beta_1 * x_i)^2 \quad (5)$$

где

y_i – наблюдаемое значение зависимой переменной для i -го наблюдения,

x_i – значение независимой переменной для i -го наблюдения,

β_0 - коэффициент сдвига (интерсепт), который представляет собой значение зависимой переменной y при $x=0$,

β_1 – коэффициент наклона, который представляет собой изменение в y , вызванное изменением в x ,

ϵ_i - случайная ошибка (остаток) для i -го наблюдения, которая представляет расхождение между наблюдаемым и предсказанным значением y .

Чтобы найти оптимальные значения для β_0 и β_1 , мы дифференцируем RSS по β_0 и β_1 , и приравниваем производные к нулю. Решение этих уравнений дает оценки параметров β_0 и β_1 , которые минимизируют сумму квадратов остатков [27].

Полученные формулы для оценок коэффициентов МНК для простой линейной регрессии выглядят следующим образом (формула 6 и формула 7):

$$\widehat{\beta}_1 = \frac{\sum_{i=1}^n (x_i - \bar{x}) * (y_i - \bar{y})}{\sum_{i=1}^n (x_i - \bar{x})^2} \quad (6)$$

$$\widehat{\beta}_0 = \bar{y} - \widehat{\beta}_1 * \bar{x} \quad (7)$$

где

\bar{y} - среднее значение y ,

\bar{x} - среднее значение x ,

$\widehat{\beta}_1$ - коэффициент сдвига МНК,

$\widehat{\beta}_0$ – коэффициент наклона МНК.

После оценки коэффициентов мы можем использовать модель для прогнозирования значений зависимой переменной y для новых значений независимой переменной x [28]. В результате описания математической модели метода линейной регрессии было выяснено, что зависимой переменной в моем случае будет являться количество пользователей на сайте, а значение независимой переменной для i -го наблюдения x_i будет время загрузки страницы корпоративного сайта.

2.3 Применение диаграммы Ганта в разработке сайта

На практике мне поручили создать диаграмму Ганта для более эффективной работы над корпоративным сайтом. Сейчас я попробую исследовать эту диаграмму и разобраться в ней. Диаграмма Ганта – это, инструмент визуализации проекта, который представляет собой столбчатую диаграмму, где по горизонтальной оси отображается время, а по вертикальной оси - задачи или этапы проекта. Каждая задача представлена отрезком столбца, длина которого соответствует времени, необходимому для выполнения задачи. Диаграмма Ганта широко применяется в различных областях и сферах деятельности, где требуется планирование, управление и мониторинг проектов. Рассмотрим преимущества диаграммы Ганта:

- Визуализация временных рамок.
- Простота восприятия.
- Идентификация зависимостей.
- Управление ресурсами.
- Мониторинг прогресса.

На основе изученной информации мною была спроектирована диаграмма Ганта для рабочего проекта создания корпоративного сайта. Результат диаграммы представлен ниже (рисунок 4).

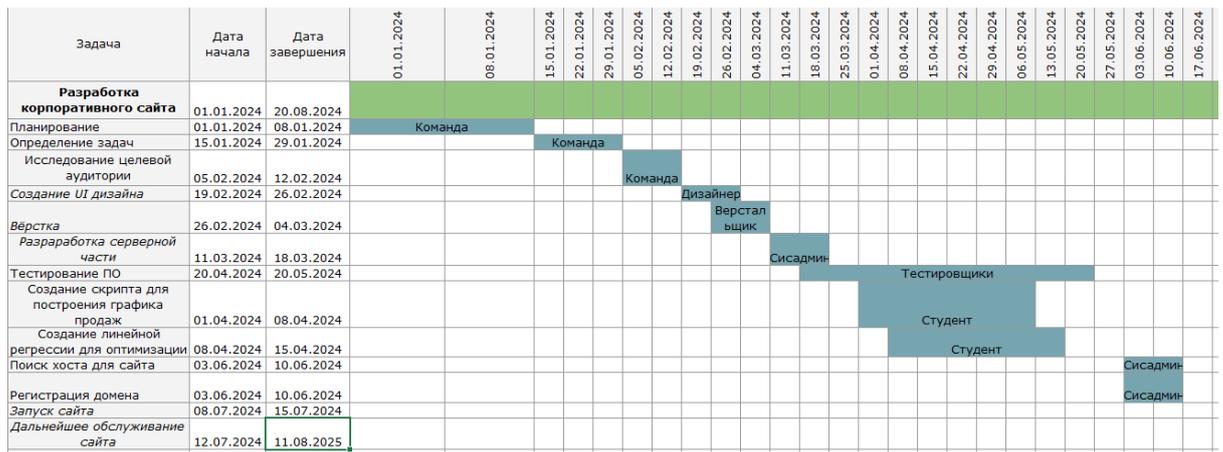


Рисунок 4. Диаграмма Ганта

Данная диаграмма улучшает эффективность работы, помогает грамотно распределить задачи, и расставить приоритет над ними. На ней вы можете видеть всех участников ответственных за те или иные задачи, а также их сроки выполнения. После создания этой диаграммы, моим дальнейшим заданием было реализовать метод, который я выбрал для анализа данных, и оптимизации корпоративного сайта.

2.4 Разработка системы для дальнейшего написания программного кода

Перед тем как приступить к реализации программного обеспечения необходимо описать проект по модулям, которые включают в себя модуль сбора и предварительной обработки данных, модуль обучения и прогнозирования и модуль визуализации результатов.

Модуль сбора и предварительной обработки данных отвечает за загрузку данных из источника, CSV-файла, и их предварительную обработку с использованием библиотеки pandas.

Модуль обучения и прогнозирования производит обучение модели линейной регрессии с использованием библиотеки scikit-learn и прогнозирование времени загрузки страницы на основе полученной модели.

Модуль визуализации результатов отвечает за построение графиков и визуализацию результатов прогнозирования с использованием библиотеки Plotly.

Программный продукт будет состоять из двух частей. В первой части будет реализация метода линейной регрессии для визуализации результатов теста устойчивости сайта к нагрузкам. Вторая часть программного продукта будет состоять из визуализации графика продаж. Для дальнейшего написания программного кода и реализации программного обеспечения нам необходимо описать блок-схемы, благодаря которым мы будем описывать реализацию ПО. Для начала опишем блок-схему программного кода линейной регрессии. Работа алгоритма представлена ниже (рисунок 5).

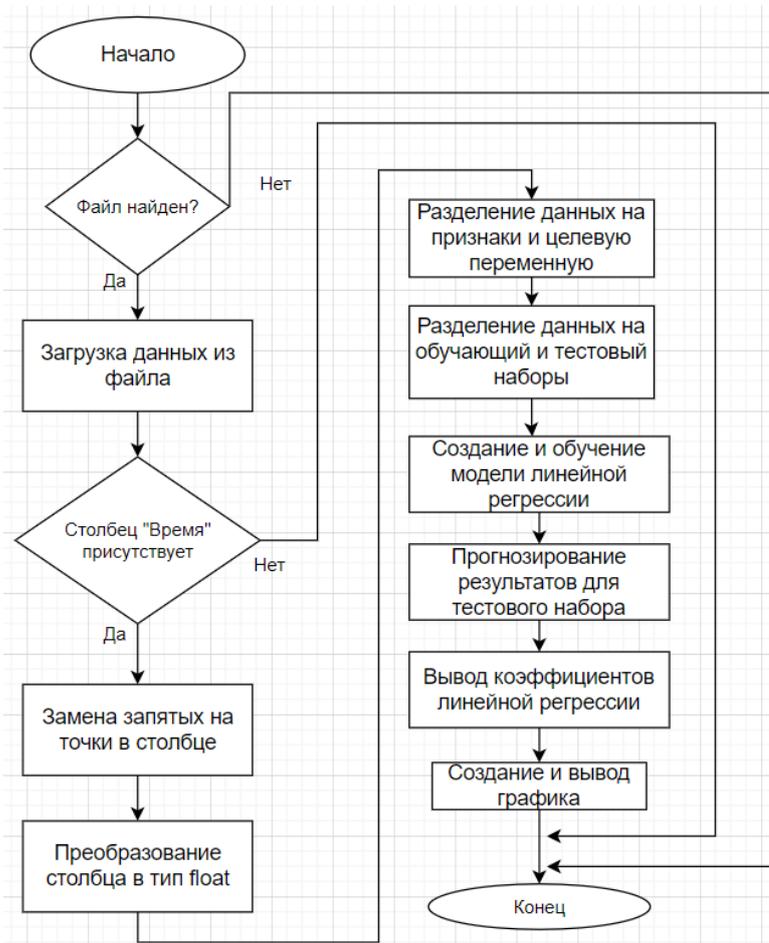


Рисунок 5 – Блок-схема алгоритма реализации линейной регрессии

Из диаграммы, представленной на рисунке 5, мы видим, что изначально происходит проверка на наличие файла в формате csv. Затем происходит загрузка данных из этого файла и проходит проверка на наличие столбца «Время». Для дальнейшей корректной работы необходимо заменить запятые на точки в столбцах и преобразовать значения столбца «Время» в тип float, чтобы проводить с ним дальнейшие вычисления.

Далее мы видим, что происходит разделение данных на признаки и целевую переменную, а также на обучающую и тестовую выборку. Затем создаётся и обучается модель линейной регрессии, после чего прогнозируются результаты для тестового набора данных.

В конце реализации алгоритма линейной регрессии происходит вывод коэффициентов линейной регрессии и визуализация результатов. Рассмотрим диаграмму второй части программного обеспечения, которая является визуализацией графика продаж продукции кампании (рисунок 6).



Рисунок 6 – Алгоритм реализации графика продаж

Из диаграммы, представленной на рисунке 6, мы видим, что построение графика это более простой процесс, чем реализация линейной регрессии и он состоит из ввода данных о продажах, после чего происходит создание и визуализация графика продаж, а также его сохранение в формате html и вывода на локальном хосте.

На основе проделанной работы по созданию системы для дальнейшей разработки, можно начинать реализацию программного обеспечения, которое будет соответствовать созданной системе. ПО будет строить график продаж для сотрудников компании для определения сезонности и текущей маржинальности за выбранный период по каждому виду продукции, а также ПО будет визуализировать алгоритм линейной регрессии для своевременной оптимизации нагрузки на домен.

Выводы по разделу 2:

Во втором разделе был произведен сравнительный анализ методов анализа данных для оптимизации корпоративного сайта и выбор пал на метод линейной регрессии. Затем была описана математическая модель этого метода.

Следующей решенной задачей стала реализация диаграммы Ганта на основе реального проекта и описание системы по модулям для дальнейшей разработки ПО.

3 Реализация и тестирование программного обеспечения

3.1 Реализация программного обеспечения

Программное обеспечение было разработана на языке программирования Python, который отвечает всем современным стандартам и имеет широкий встроенный функционал. В качестве среды для разработки была выбрана программа PyCharm. Реализация в данной работе будет состоять из двух частей. В первой части будет описана визуализация графика о продажах кампании, а во второй будет представлен график линейной регрессии, который помогает своевременно обнаруживать и устранять неполадки, связанные с оптимизацией корпоративного сайта. Сначала, приступим к реализации первой части программного обеспечения.

На сайте компании помимо основного функционала необходимо визуализировать данные о продажах за определенный период для трех видов блоков, чтобы наглядно показать динамику продаж и выявить тенденции. Визуализация данных является важным инструментом для анализа и принятия обоснованных управленческих решений. В данном случае, для представления данных о продажах используется язык Python, библиотека Plotly и функция `go.Scatter`, которые позволяют создать наглядный и интерактивный график рассеяния (`scatter plot`).

Plotly - это библиотека для визуализации данных, которая поддерживает множество типов графиков и позволяет легко настраивать их внешний вид. Она также предоставляет возможность сохранять графики в различных форматах, включая HTML, что удобно для последующего анализа и корректировки.

Для представления данных о продажах в виде графика была выбрана функция `go.Scatter`, которая создает график рассеяния (`scatter plot`). Он представляет собой двумерный график, на котором точки данных отображаются в виде точек на декартовой плоскости. Этот тип графика

хорошо подходит для демонстрации динамики изменения данных во времени и позволяет наглядно увидеть тенденции и закономерности. В нашем случае, scatter plot позволяет наглядно показать изменение продаж каждого вида блоков в течение года, что может помочь выявить сезонные колебания, пики и спады спроса. Кроме того, scatter plot позволяет легко сравнить продажи разных видов блоков, что может быть полезно для принятия управленческих решений. Объявление всех необходимых для работы библиотек представлено ниже (рисунок 7).

```
import plotly.graph_objs as go
from plotly.offline import plot
```

Рисунок 7 – Объявление всех необходимых библиотек

Хотя существуют и другие типы графиков, такие как гистограммы, линейчатые диаграммы и точечные диаграммы, scatter plot является наиболее подходящим для нашего случая, так как он позволяет наглядно показать динамику изменения данных во времени и сравнительный анализ продаж разных видов блоков. Этот тип графика хорошо подходит для демонстрации динамики изменения данных во времени и позволяет наглядно увидеть тенденции и закономерности. Далее происходит заполнение графика тестовыми данными.

Объект Figure используется для хранения и управления графиком. Он позволяет добавлять, удалять и модифицировать данные, а также настраивать внешний вид графика с помощью методов `update_layout` и `update_traces`.

После CSS настройки графика в программе создается объект Figure с тремя кривыми (для каждого вида блоков), настраивается внешний вид графика (заголовок, названия осей, цвета и шрифт) и сохраняется в виде HTML-файла. Реализация данного фрагмента описана ниже (рисунок 8).

```

# Создаем объект Figure с тремя кривыми
fig = go.Figure(data=[
    go.Scatter(x=sales_data['2023 год'], y=sales_data['Стеновой блок'], name='Стеновой блок'),
    go.Scatter(x=sales_data['2023 год'], y=sales_data['Фундаментный блок'], name='Фундаментный блок'),
    go.Scatter(x=sales_data['2023 год'], y=sales_data['Вентиляционный блок'], name='Вентиляционный блок')
])

# Настраиваем внешний вид графика
fig.update_layout(
    title='Продажи за год',
    xaxis_title='2023 год',
    yaxis_title='Продажи',
    plot_bgcolor='white', # Цвет фона графика
    paper_bgcolor='white', # Цвет фона области рисования
    font=dict(
        size=18, # Размер шрифта
        color='black' # Цвет шрифта
    )
)

# Сохраняем график в виде HTML-файла и открываем его в браузере
plot(fig, filename='sales_chart.html')

```

Рисунок 8 – Программная реализация объекта Figure, настройка и сохранение графика продаж

Приступим к реализации второй части программного обеспечения, которая заключается в том, чтобы с помощью языка Python провести анализ определения зависимости времени загрузки страницы сайта от количества посетителей. Этот анализ имеет важное значение для оптимизации работы сайта и обеспечения комфортного пользовательского опыта. Время загрузки страницы является одним из ключевых факторов, влияющих на удовлетворенность пользователей и их возможность использовать сайт эффективно.

Построение зависимости времени загрузки страницы от количества посетителей позволяет оценить, как изменение числа пользователей влияет на производительность сайта. Это, в свою очередь, может помочь в принятии решений о необходимости масштабирования серверных ресурсов, оптимизации кода сайта или внедрении других мер, направленных на

повышение скорости загрузки страниц. Рассмотрим какие библиотеки используются в данном коде.

Для проведения анализа и построения модели линейной регрессии используется набор данных, содержащий информацию о времени загрузки страницы и количестве посетителей в различные моменты времени. Данные предварительно обрабатываются с помощью библиотеки pandas, а затем разделяются на обучающую и тестовую выборки с использованием функции `train_test_split` из библиотеки `scikit-learn`. Реализация представлена ниже (рисунок 9).

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn import metrics
import plotly.graph_objs as go
from plotly.offline import plot

# Загрузка данных из CSV-файла
data = pd.read_csv(r'C:\Users\им\Desktop\ТГУ\Антон\Лин.csv', encoding='cp1251', sep=';')
print(data.head())

data[['Время']] = data[['Время']].apply(lambda x: x.str.replace(',', '.'))

# Преобразование столбца 'Время' в тип float
X = data[['Время']].astype(float)

# Разделение данных на признаки (X) и целевую переменную (y)
y = data['Количество'].values

# Разделение данных на обучающий и тестовый наборы
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=0)
```

Рисунок 9 – Работа с данными и объявление всех библиотек

Модель линейной регрессии обучается на обучающей выборке с помощью класса `LinearRegression`, также из `scikit-learn`. После обучения модели она используется для прогнозирования времени загрузки страницы на тестовой выборке. Результаты прогноза сравниваются с фактическими

значениями времени загрузки, и вычисляется ошибка среднего квадратичного с использованием функции `metrics.mean_squared_error` из `scikit-learn`.

Для визуализации результатов используется библиотека `Plotly`, которая позволяет создавать интерактивные графики. В данном коде строится график, на котором отображаются исходные данные и прогноз, что позволяет наглядно оценить точность построенной модели линейной регрессии. Реализация представлена ниже (рисунок 10).

```
# Создание и обучение модели линейной регрессии
regressor = LinearRegression()
regressor.fit(X_train, y_train)

# Прогнозирование результатов для тестового набора
y_pred = regressor.predict(X_test)

# Вывод коэффициентов линейной регрессии
print('Коэффициент наклона (a):', regressor.coef_[0])
print('Пересечение с осью Y (b):', regressor.intercept_)

# Вычисление ошибки
print('Ошибка среднего квадратичного:', metrics.mean_squared_error(y_test, y_pred))

# Вывод графика с помощью plotly
trace1 = go.Scatter(x=X_test.squeeze(), y=y_test, mode='markers', name='Исходные данные')
trace2 = go.Scatter(x=X_test.squeeze(), y=y_pred, mode='lines', name='Прогноз')
layout = go.Layout(title='Линейная регрессия', xaxis=dict(title='Время'), yaxis=dict(title='Количество'))
fig = go.Figure(data=[trace1, trace2], layout=layout)
plot(fig, filename='lin.html')
```

Рисунок 10 – Обучение модели и построение графика линейной регрессии

В ходе реализации программного обеспечения был использован язык программирования `Python`, что позволило значительно сократить код, используя множество встроенных функций и библиотек. Программный код отвечает всем требованиям и стандартам компании, в которой я прохожу преддипломную практику.

3.2 Тестирование программного обеспечения

Для тестирования программного обеспечения был выбран метод функционального тестирования, который сосредотачивается на проверке функциональных требований системы. Этот метод концентрируется на оценке того, насколько программа соответствует ожиданиям пользователя. Происходит это посредством выполнения проверки функционала и возможностей, доступных в приложении. Необходимо удостовериться, что программное обеспечение ведет себя так, как ожидается со стороны пользователей. Оно фокусируется на том, как приложение взаимодействует с пользователем и какие функции оно предоставляет. Основное внимание уделяется интерфейсу пользователя и функциональным возможностям.

Произведём тестирование первой программы, которая визуализирует годовой график продаж для сотрудников компании. Это необходимо для того, чтобы определить среднюю маржинальность, сезонность и спрос каждого вида продукции по отдельности. График продаж представлен ниже (рисунок 11).

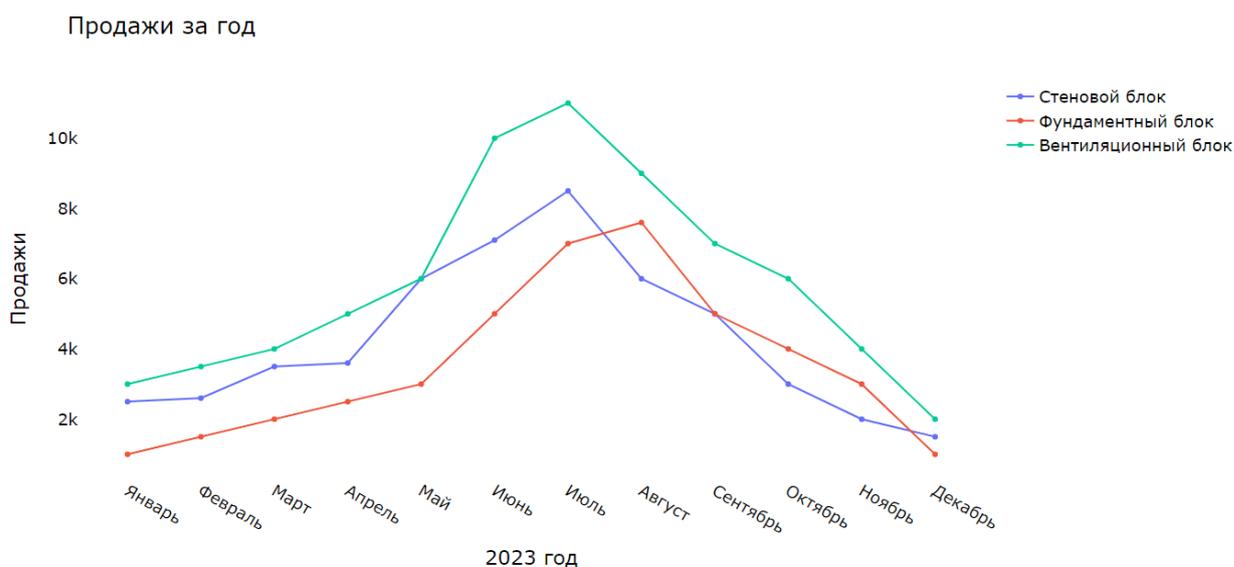


Рисунок 11 - Общий вид графика продаж

На данном графике наглядно показано, как изменялась выручка за каждый вид продукции кампании по отдельности в течение года. На нём можно отследить всплески и падения прибыли, а также отследить сезонность. Передо мной также была поставлена задача сделать график интерактивным. В виду этого, я добавил возможность увеличивать масштаб графика, посредством выделения области на экране. Также была добавлена возможность увидеть название графика, выбранный месяц и его выручку за этот месяц при наведении на точку на графике (рисунок 12).

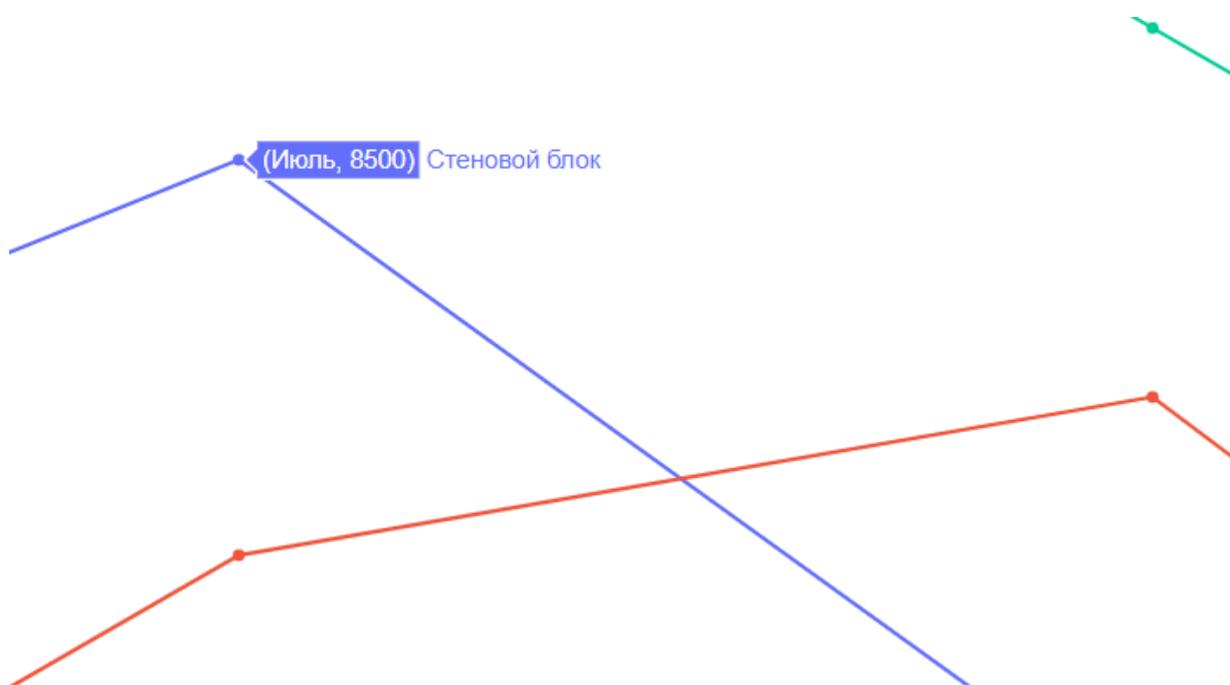


Рисунок 12 - Приближение графика и визуализация за выбранный период

На рисунке 10 видно, что за рассматриваемый период в июле было продано 8500 стеновых блоков. Такой вид графика удобно анализировать. Помимо этого, я добавил возможность скрыть график при нажатии на него на легенде графика. Сделано это для того, чтобы при большом объёме данных избавляться от лишнего шума (рисунок 13).

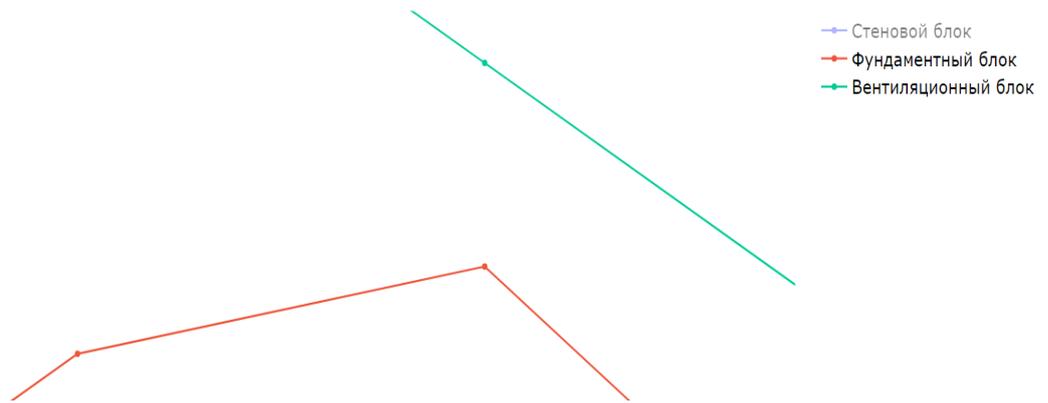


Рисунок 13 - Скрытие графика

Вторая программа, которую я реализовал, представляет собой визуализацию графика алгоритма линейной регрессии с течением времени для того, чтобы отследить как сильно нагружается система при увеличении количества пользователей и своевременно принимать меры по оптимизации корпоративного сайта. Результат предоставлен ниже (рисунок 14).

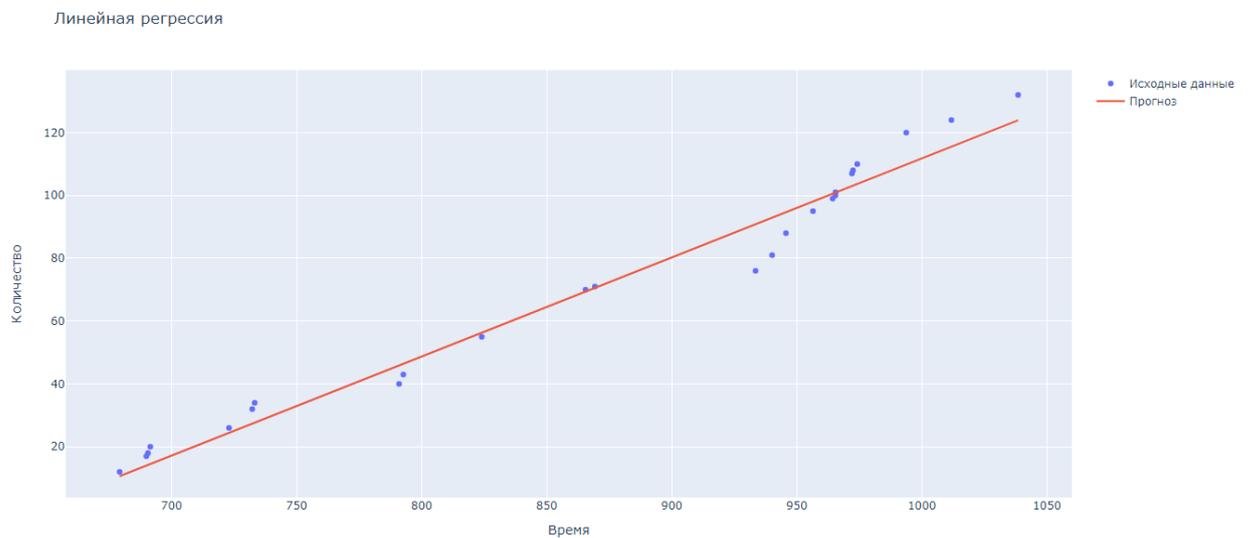


Рисунок 14 - График линейной регрессии

На рисунке 11 представлен график линейной регрессии. На нем можно отследить, что с увеличением количества пользователей на сайтекратно увеличивается время загрузки страницы в миллисекундах. Также можно

сделать вывод, что никаких проблем не выявлено, потому что нет резких всплесков и провалов на графике. Также я реализовал возможность приближения графика посредством выделения области на экране и визуализация результатов при приближении (рисунок 15).

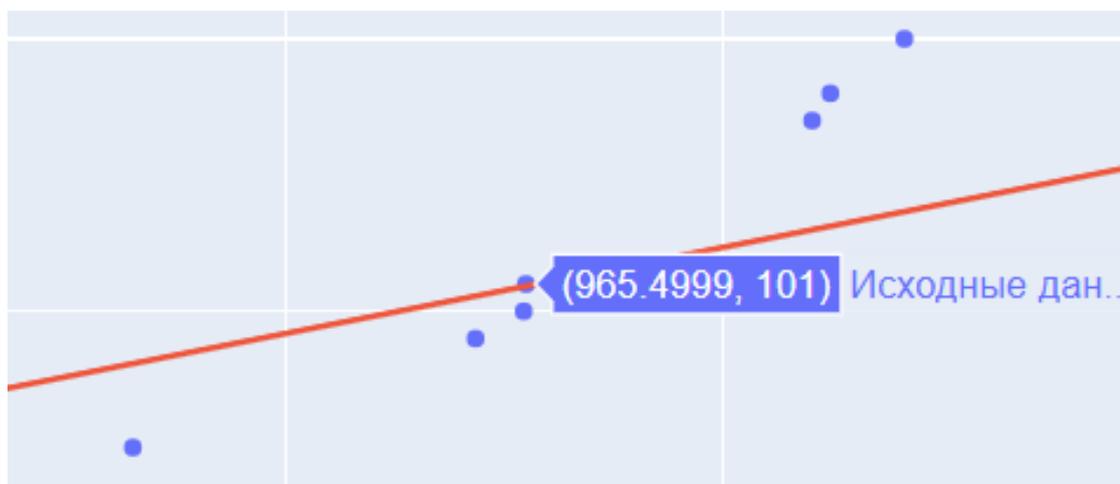


Рисунок 15 - Функционал график линейной регрессии

В результате тестирования никаких проблем не выявлено. Программа работает стабильно и готова к внедрению на корпоративном сайте. График продаж наглядно показывает выручку по каждому виду производимой продукции. Также была реализация интерактивных элементов программного обеспечения. Описанный выше функционал полностью отвечает всем требованиям кампании, в которой я прохожу преддипломную практику.

Выводы по разделу 3:

В третьем разделе ВКР был описан написанный код, а также было протестировано программное обеспечение на тестовом наборе данных для дальнейшего внедрения на корпоративный сайт. Также была произведена проверка того, как система реагирует на некорректный ввод данных со стороны пользователя и проведен анализ полученных результатов.

Заключение

Бакалаврская работа посвящена применению компьютерной модели в разработке и оптимизации функциональности корпоративного сайта.

В ходе выполнения ВКР были поставлены задачи на исследования.

Была поставлена задача на исследование, описан процесс создания корпоративных сайтов, а также был произведен обзор и выбор компьютерной модели для дальнейшего использования. После этого было рассмотрено практическое применение компьютерных моделей для разработки и оптимизации функциональности корпоративного сайта.

Далее был произведен сравнительный анализ способов оценки данных для оптимизации корпоративного сайта. Наиболее предпочтительным оказался метод линейной регрессии. Для этого метода была описана математическая модель.

Также программное обеспечение было описано по составным частям посредством описания компонентов сбора, анализа и визуализации данных и описаны блок-схемы программного кода.

В заключительном разделе ВКР был показан процесс реализации и функционального тестирования программного обеспечения и произведён анализ полученных результатов. Тестирование было произведено на тестовых данных с возможностью в дальнейшем использовать программное обеспечение для более сложных вычислений

Цель работы была выполнена: разработано программное обеспечение, которое помогает оптимизировать процесс разработки корпоративного сайта, а также расширяет его и функционал и отслеживает нагрузку на домен с целью своевременной оптимизации корпоративного сайта.

Задачи, определённые для достижения цели работы, были выполнены в полном объёме.

Список используемой литературы

1. Авербух А. И. Компьютерное моделирование: система ModSim. — М.: Интернет-Университет Информационных Технологий (ИУИТ), 2005. — 352 с.
2. Айвазян С. А., Бухштабер В. М., Енюков И. С. Прикладная статистика: Классификация и снижение размерности. — М.: Финансы и статистика, 1989. — 607 с.
3. Герасимов Ю. С., Петров И. Б. Методы и средства компьютерного моделирования: Учебное пособие. — СПб.: СПбГУИТМО, 2012. — 320 с.
4. Дроздов Д. В. Компьютерное моделирование и информационные технологии в управлении: Учебное пособие. — М.: Аспект Пресс, 2008. — 256 с.
5. Жиглов А. Н. Компьютерное моделирование в принятии управленческих решений. — М.: КомКнига, 2008. — 208 с.
6. Завгородний В. Г., Булавская Т. В., Чжан Вэйдун. Методы компьютерного моделирования. — М.: Издательство МГУ, 2002. — 288 с.
7. Иванов А. М. Моделирование и оптимизация бизнес-процессов: Учебное пособие. — М.: ИНФРА-М, 2013. — 368 с.
8. Колесов В. В., Солодовников В. В., Шитов В. А. Методы оптимизации в инженерном моделировании. — М.: Физматлит, 2006. — 368 с.
9. Корнилов А. А., Косухин В. В. Проектирование информационных систем. — СПб.: Питер, 2008. — 512 с.
10. Лебедев В. М., Хорева Н. Н. Компьютерное моделирование: Учебник для вузов. — М.: Финансы и статистика, 2006. — 544 с.
11. Лотоцкий В. А., Костринский А. Б. Методы оптимизации и компьютерные технологии. — М.: МИСИС, 2004. — 272 с.
12. Мельников Ю. Б., Нестеров С. С. Программное обеспечение компьютерных сетей и информационных систем. — М.: Финансы и статистика, 2011. — 288 с.

13. Непомнящих Г. О., Шашков А. С. Программирование на Python 3: Подробное руководство. — СПб.: БХВ-Петербург, 2010. — 704 с.
14. Олейников О. А. Методы оптимизации: Конспект лекций. — М.: МФТИ, 2005. — 268 с.
15. Петров И. Б., Герасимов Ю. С. Программное обеспечение компьютерного моделирования. — СПб.: Питер, 2007. — 240 с.
16. Booch, G., Rumbaugh, J., & Jacobson, I. (2005). The Unified Modeling Language User Guide. Addison-Wesley.
17. Craig, L. (2018). HTML and CSS: Visual QuickStart Guide. Peachpit Press.
18. Fowler, M. (2002). UML Distilled: A Brief Guide to the Standard Object Modeling Language. Addison-Wesley.
19. Freeman, E., Robson, E., & Bates, B. (2007). Head First HTML and CSS: A Learner's Guide to Creating Standards-Based Web Pages. O'Reilly Media.
20. Goyal, P. (2013). Flask Web Development: Developing Web Applications with Python. O'Reilly Media.
21. Hohpe, G., & Woolf, B. (2003). Enterprise Integration Patterns: Designing, Building, and Deploying Messaging Solutions. Addison-Wesley Professional.
22. Johnson, E. H., & Tittel, E. (2001). XML How to Program. Prentice Hall.
23. Molluzzo, J., & Nguyen, T. (2006). JavaServer Faces: The Complete Reference. McGraw-Hill Osborne Media.
24. Pilgrim, M. (2009). Dive into Python 3. Apress.
25. Powell, T. (2018). Learning Python: Powerful Object-Oriented Programming. O'Reilly Media.
26. Sebesta, R. W. (2018). Programming the World Wide Web. Pearson.
27. Tidwell, D. (2005). Designing Interfaces: Patterns for Effective Interaction Design. O'Reilly Media.
28. van Rossum, G., & Drake, F. L. (2009). Python 3 Reference Manual. CreateSpace Independent Publishing Platform.

Приложение А

Листинг (реализация графика продаж)

Листинг 1 – реализация графика продаж.

```
import plotly.graph_objs as go
from plotly.offline import plot

# Предположим, у нас есть данные о продажах за определенный период
sales_data = {
    '2023 год': ['Январь', 'Февраль', 'Март', 'Апрель', 'Май', 'Июнь', 'Июль',
'Август', 'Сентябрь', 'Октябрь', 'Ноябрь', 'Декабрь'],
    'Стеновой блок': [2500, 2600, 3500, 3600, 6000, 7100, 8500, 6000, 5000,
3000, 2000, 1500],
    'Фундаментный блок': [1000, 1500, 2000, 2500, 3000, 5000, 7000, 7600,
5000, 4000, 3000, 1000],
    'Вентиляционный блок': [3000, 3500, 4000, 5000, 6000, 10000, 11000,
9000, 7000, 6000, 4000, 2000]
}

# Создаем объект Figure с тремя кривыми
fig = go.Figure(data=[
    go.Scatter(x=sales_data['2023 год'], y=sales_data['Стеновой блок'],
name='Стеновой блок'),
    go.Scatter(x=sales_data['2023 год'], y=sales_data['Фундаментный блок'],
name='Фундаментный блок'),
    go.Scatter(x=sales_data['2023 год'], y=sales_data['Вентиляционный
блок'], name='Вентиляционный блок')
])
```

Продолжение Приложения А

```
# Настраиваем внешний вид графика
fig.update_layout(

    title='Продажи за год',
    xaxis_title='2023 год',
    yaxis_title='Продажи',
    plot_bgcolor='white', # Цвет фона графика
    paper_bgcolor='white', # Цвет фона области рисования
    font=dict(
        size=18, # Размер шрифта
        color='black' # Цвет шрифта
    )
)

# Сохраняем график в виде HTML-файла и открываем его в браузере
plot(fig, filename='sales_chart.html')
```

Приложение Б

Листинг (реализация графика линейной регрессии)

Листинг 2 – реализация линейной регрессии.

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn import metrics
import plotly.graph_objs as go
from plotly.offline import plot

# Загрузка данных из CSV-файла
data = pd.read_csv(r'C:\Users\мм\Desktop\ТГУ\Антон\Лин.csv',
encoding='cp1251', sep=';')
print(data.head())

# Предположим, что в CSV-файле есть два столбца: 'feature' и 'target'
# 'feature' - это признак, 'target' - это целевая переменная

data[['Время']] = data[['Время']].apply(lambda x: x.str.replace(',', '.'))

# Преобразование столбца 'Время' в тип float
X = data[['Время']].astype(float)

# Разделение данных на признаки (X) и целевую переменную (y)
y = data['Количество'].values

# Разделение данных на обучающий и тестовый наборы
```

Продолжение Приложения Б

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=0)

# Создание и обучение модели линейной регрессии
regressor = LinearRegression()
regressor.fit(X_train, y_train)

# Прогнозирование результатов для тестового набора
y_pred = regressor.predict(X_test)

# Вывод коэффициентов линейной регрессии
print('Коэффициент наклона (a):', regressor.coef_[0])
print('Пересечение с осью Y (b):', regressor.intercept_)

# Вычисление ошибки
print('Ошибка среднего квадратичного:',
metrics.mean_squared_error(y_test, y_pred))

# Вывод графика с помощью plotly
trace1 = go.Scatter(x=X_test.squeeze(), y=y_test, mode='markers',
name='Исходные данные')
trace2 = go.Scatter(x=X_test.squeeze(), y=y_pred, mode='lines',
name='Прогноз')
layout = go.Layout(title='Линейная регрессия', xaxis=dict(title='Время'),
yaxis=dict(title='Количество'))
fig = go.Figure(data=[trace1, trace2], layout=layout)
plot(fig, filename='lin.html')
```