

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

Институт математики, физики и информационных технологий
(наименование института полностью)

Кафедра Прикладная математика и информатика
(наименование института полностью)

02.03.03 Математическое обеспечение и администрирование информационных систем
(код и наименование направления подготовки, специальности)

Мобильные и сетевые технологии
(направленность (профиль) / специализация)

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА (БАКАЛАВРСКАЯ РАБОТА)

на тему Реализация метода регрессионного анализа для прогнозирования объема продаж
(на примере ООО «Парк Горького»)

Студент	<u>И.А. Дементьева</u> (И.О. Фамилия)	<u>_____</u> (личная подпись)
Руководитель	<u>канд. пед. наук, доцент, Т.А. Агошкова</u> (ученая степень, звание, И.О. Фамилия)	
Консультант	<u>М.В. Дайнеко</u> (ученая степень, звание, И.О. Фамилия)	

Тольятти 2021

Аннотация

Тема выпускной квалификационной работы - «Реализация метода регрессионного анализа для прогнозирования объема продаж на примере (ООО «Парк Горького»)».

В данной работе рассматривается алгоритм линейной регрессии как инструмент прогнозирования для принятия управленческих решений. Предвидеть результат работы важно для бизнеса: зная приближенные результаты продаж, принимается решение об объемах закупки товаров и приглашении на работу сотрудников со сдельной оплатой труда. Источником данных для анализа является ООО «Парк Горького» - парк аттракционов, расположенный в г.Сызрань. Деятельности парка аттракционов является сезонной, поэтому важно заработать максимальную прибыль в минимальные сроки.

Текущей проблемой для прогнозирования продаж в ручном режиме самой администрацией парка являются ограничительные меры в связи с эпидемией коронавирусной инфекции. В 2019 году были хорошие результаты продаж, но в 2020 почти весь сезон парк не работал. 2021 год является в какой-то степени синтезом 2019 и 2020 года: ограничительные меры есть, но они ослаблены по сравнению с 2020 годом, что позволяет приблизиться к уровню продаж 2019 года.

В процессе выполнения данной бакалаврской работы был разработан программный продукт, прогнозирующий объем продаж в разрезе внешних факторов таких, как погодные условия и день недели. Программный продукт включает в себя логическую часть, алгоритм которой основан на регрессионном анализе, и интерфейсную часть для вывода результатов прогноза на форму пользователя.

Abstract

The title of the graduation work is Implementing a regression analysis method for sales forecasting (by the example of OOO Park Gorky, a limited liability company under the laws of the Russian Federation). The aim of the work is to implement a sales forecasting algorithm for an amusement park.

The object of the graduation work is OOO Park Gorky, a limited liability company under the laws of the Russian Federation, that deals with providing amusement rides services. The subject of the investigation is identifying the dependence of sales on weather conditions and days of week. The graduation work may be divided into several logically connected parts which are: setting a task and choosing the best way to solve it; designing a programme for sales forecasting based on the regression analysis method; implementing a programme for sales forecasting based on the method of regression analysis.

We start with the statement of the problem and then logically pass over to its possible solutions. The amusement park operates only in warm seasons, so it is important to make a maximum profit in the short period of time. Sales forecasting is a relevant topic because forecasting financial results and the number of visitors is necessary for planning procurement and hiring employees with piece-rate wages. In the present graduation work, a problem solution method analysis is carried out. Regression analysis is chosen as the mathematical method of sales forecasting and Python is chosen as the programming language. To develop the user interface, QT Designer is used. The client has pointed out two factors affecting the profit (weather and days of week). If the number of factors increases in the long run, the multiple linear regression algorithm will allow us to take these factors into account to prepare a forecast.

Thus, a cross-platform tool to predict sales depending on the external factors (weather, day of week) is developed. At present, the programme is undergoing experimental-industrial operation at OOO Park Gorky, a limited liability company under the laws of the Russian Federation.

Содержание

Введение.....	5
1 Постановка задачи и выбор оптимального способа ее решения.....	8
1.1 Анализ деятельности заказчика и описание задачи работы.....	8
1.2 Анализ и выбор математического метода для решения задачи	11
1.3 Анализ и выбор технологии для решения задачи.....	13
2 Проектирование приложения для прогнозирования продаж на основе принципов регрессионного анализа	16
2.1 Описание этапов проектирования приложения	16
2.2 Основы регрессионного анализа	20
2.3 Построение регрессионной модели задачи	25
3 Реализация приложения для прогнозирования продаж на основе принципов регрессионного анализа.....	30
3.1 Реализация алгоритма регрессионного анализа	30
3.2 Разработка пользовательского интерфейса	38
Заключение	42
Список используемой литературы	44
Приложение А Функциональная модель главного модуля	47
Приложение Б Фрагмент кода	48

Введение

Одна из важных составляющих успеха бизнеса - планирование объема продаж. Выполненный правильно прогноз делает процесс ведения бизнеса эффективнее, и самое главное - позволяет контролировать и оптимизировать расходы.

Большинство организаций используют автоматизированные системы для ведения учета продаж, однако чаще всего эти системы являются примитивными с точки зрения прогнозирования: их функционал рассчитан на закрытие экономических вопросов, например, выдачу чеков клиентам и сдачу налоговой и бухгалтерской отчетности в контролирующие органы. Примером таких систем являются решения от компании 1С, в конфигурациях есть подсистема «Планирование», в которой реализованы отчеты планов продаж, однако анализируемый для прогнозирования объем данных таких отчетов достаточно плоский: подсистема учитывает только данные о продажах, прошедшие через базу 1С, без учета внешних факторов.

Заказчиком программного продукта для прогнозирования объема продаж выступило общество с ограниченной ответственностью «Парк Горького». Это достаточно молодая компания, которая работает на рынке развлечений с 2015 года. Территориально ООО «Парк Горького» находится в г.Сызрань, основным видом деятельности организации является оказание услуг по пользованию аттракционами и прокату развлекательного оборудования. Деятельность парка аттракционов носит сезонный характер, поэтому получить максимальную прибыль необходимо в сжатые сроки. Вдобавок по требованиям техники безопасности аттракционы не могут работать в непогоду, что также накладывает ограничения на период получения дохода организацией.

К вопросу анализа данных для прогнозирования: с каждым годом объем данных растет. По прогнозам компании International Data Corporation

объем данных всего мира к 2025 году увеличится в 10 раз и составит 163 зеттабайт. В силу того, что измерений для анализа может быть сколько угодно, применение типовых инструментов делает прогноз на основе анализа этих данных плоским и менее качественным.

В случае с ООО «Парк Горького» прогнозирование объема продаж может осуществляться в разрезе погодных условий. Причем, есть две составляющих анализа в разрезе погоды - температура и уровень ясности неба. В дождь запрещена как эксплуатация самих аттракционов, так и сами люди не захотят гулять под открытым небом. Температура теплого периода, как правило, не накладывает ограничений на эксплуатацию аттракционов, но имеет вполне логичное влияние на количество посетителей парка: в знойную погоду люди больше захотят провести время там, где прохладнее, а в холодную - там, где теплее.

Таким образом, разработка программного продукта для прогнозирования объема продаж является актуальной, т.к. прогнозирование финансовых результатов и, соответственно, количества посетителей парка аттракциона необходимо для планирования объема закупок и вызова сотрудников со сдельной оплатой труда. На объем продаж парка аттракционов влияет фактор погодных условий - в плохую погоду никто не захочет гулять в парке, а также фактор дня недели - в выходные дни посетителей больше, чем в будние.

Целью данной работы является реализация алгоритма прогнозирования продаж для парка развлечений.

Для достижения цели необходимо выполнить следующие задачи:

- проанализировать деятельность ООО «Парк Горького»;
- выгрузить из используемой системы данные о продажах;
- провести сравнение математических методов анализа данных;
- выбрать метод для анализа продаж;
- реализовать выбранный метод для выгруженных данных продаж;
- спроектировать пользовательский интерфейс программы;

- реализовать вывод результатов прогноза в разработанном интерфейсе пользователя;
- сделать выводы по проделанной работе.

Объектом исследования данной работы является парк аттракционов ООО «Парк Горького», деятельность которого связана с оказанием услуг по пользованию аттракционов и прокату оборудования.

Предмет исследования работы – выявление зависимости погодных условий и дня недели на объем продаж.

Работа состоит из трех разделов. В первом разделе подробно описывается поставленная задача, рассматриваются с теоретической точки зрения методы решения задачи, выполняется анализ существующих разработанных решений поставленной задачи, обосновываются математический метод решения задачи и технология для реализации этого метода.

Во втором разделе выбранный математический метод рассматривается детально, происходит построение модели задачи, выполняется проектирование структуры программного продукта и пользовательского интерфейса.

В третьем разделе описывается ход разработки программного продукта с разбивкой на две составляющие: реализация самого метода прогнозирования и реализация пользовательского интерфейса, согласно спроектированному в предыдущем разделе макету.

В процессе выполнения выпускной квалификационной работы продемонстрировано обладание такими компетенциями, как готовность анализировать проблемы и направления развития технологий программирования, готовность использовать навыки проектирования, реализации и анализа эффективности программного обеспечения для решения задач в различных предметных областях, готовность к разработке модулирующих алгоритмов и реализации их.

1 Постановка задачи и выбор оптимального способа ее решения

1.1 Анализ деятельности заказчика и описание задачи работы

ООО «Парк Горького» - крупный парк в Самарской области, расположенный в г.Сызрань. Парк получает прибыль с продажи билетов на развлекательные аттракционы. Основной массой посетителей, конечно, являются семьи с детьми, поэтому спросом пользуются так называемые семейные аттракционы — это различные карусели, прогулки на воде, прокат велосомобилей и электромобилей. Но и для посетителей постарше также есть развлечения: экстремальные аттракционы на проверку вестибулярного аппарата, занятия, требующие хорошей физической подготовки. Также есть и аттракционы для детей до 3 лет: это качалки, спокойные карусели, где они могут кататься в сопровождении взрослых, детские развивающие комнаты и т.д. Организационная диаграмма парка представлена на рисунке 1. ООО «Парк Горького» является коммерческой организацией, а целью любой такой организации является получение максимальных показателей продаж.

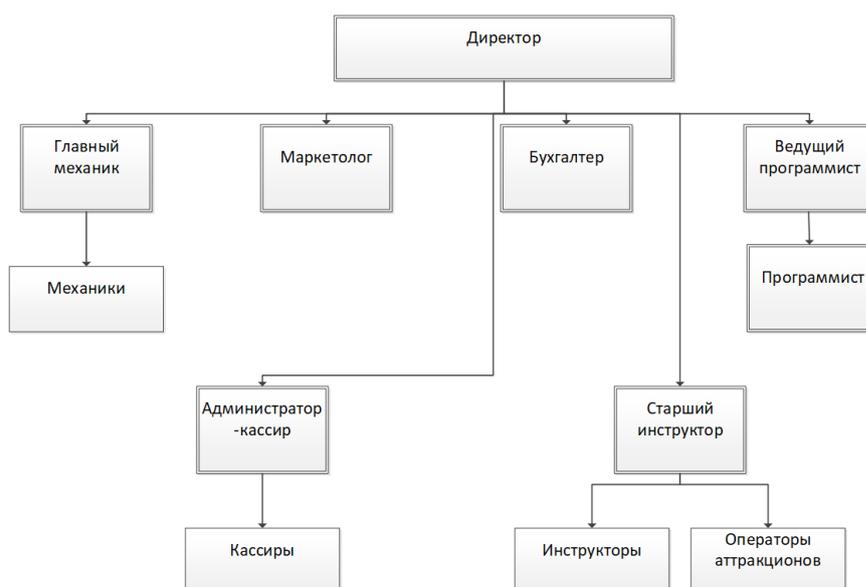


Рисунок 1 - Организационная диаграмма

Заказчиком были озвучены два основных фактора, влияющих на уровень продаж - день недели и погода. На самом деле факторов может быть больше, поэтому важно при выборе метода анализа данных выбрать такой метод, который позволит анализировать большое количество факторов одновременно.

Зависимость выручки от дня недели логична: в выходные и праздничные дни посетителей больше всего. Но бывают и исключения: причиной могут быть различные мероприятия, проводимые в парке для поднятия посещаемости, и случайное стечение обстоятельств.

Также важным фактором является погода - в плохие погодные условия не придут посетители и нельзя эксплуатировать аттракционы, т.к. они находятся под открытым небом.

Таким образом, два основных фактора для прогнозирования посещаемости парка — это день недели и погодные условия. Для того, чтобы предсказать объем будущих продаж, кроме этих двух составляющих, необходимо знать и объем продаж за предыдущие периоды. Для прогноза, какими будут прогнозы в новом сезоне можно взять данные о продажах в предыдущем сезоне. Но в случае с 2021 годом лучше будет взять для анализа данные за два года: 2019 год и 2020 год, в виду того, что посещаемость в 2019 году теоретически могла спокойно достигать пиковых значений, но в 2020 в силу ограничений, связанных с коронавирусной инфекцией, продажи резко упали и достигли минимальных порогов.

В настоящее время продажа билетов осуществляется в парке посредством 1С:Управление торговлей, версия 11.5.6. Конфигурация находится на поддержке, все механизмы являются типовыми. Продажи регистрируются объектом 1С Документ - «Чек ККМ». В начале рабочего дня кассир открывает смену в программе, пробивает чеки и в конце рабочего дня снимает Z-отчет. Самым простым способом выгрузки данных из 1С является экспорт табличным документом. Однако, в планах парка стоит задача перехода с 1С на другой программный продукт, который будет более

ориентирован на вид деятельности парков аттракционов. В данный момент владельцы не определились, на какой механизм они будут переходить, поэтому необходимо реализовать программу для прогнозирования объема продаж будущих периодов таким образом, чтобы файл выгрузки был универсальным для информационной системы. Список аттракционов приведен в таблице 1.

Таблица 1 - Аттракционы ООО «Парк Горького»

№ п/п	Название аттракциона
1	Аэрохоккей
2	Бамперные лодки
3	Батут «Африка»
4	Батут надувной «Крокодил»
5	Батут надувной «Тропикана»
6	Батут спортивный
7	Карусель «Вальс ракушек»
8	Веломобиль
9	Карусель «Колокольчик»
10	Карусель «Манеж»
11	Катальная гора «Астероид»
12	Качалка для детей
13	Качели «Ермак»
14	Колесо обозрения
15	Лабиринт «Жук»
16	Миникартинг
17	Цепочная карусель
18	Электромобиль

Выполненный результат прогноза будет полезен для сотрудников, выполняющих административные функции, это старший инструктор, администратор-кассир, главный механик и директор. Результат данного прогноза поможет определиться с тем, сколько закупок делать на тот или иной день, сколько приглашать на работу сотрудников со сдельным вариантом оплаты труда.

Таким образом, в данном параграфе был проведен анализ предметной области и с точки зрения бизнес-логики описана поставленная задача бакалаврской работы.

1.2 Анализ и выбор математического метода для решения задачи

Прогнозирование является неотъемлемой составляющей коммерческого вида деятельности. Прогнозы помогают людям рассматривать ситуацию с разных сторон и побуждают действовать сотрудников более рационально. Именно прогноз является обеспечивающим инструментом принятия рациональных решений в организации. Отсутствие прогноза заставляет руководителей предпринимать излишние меры предосторожности и тратить больше времени регулярно на ручной анализ.

Существуют математические методы прогнозирования, например, такие методы как:

- модель скользящего среднего;
- модель авторегрессии – скользящего среднего (ARMA);
- расширенная модель авторегрессии – скользящего среднего, или модель Бокса-Дженкинса (ARIMA);
- регрессионная модель.

Рассмотрим каждый из этих вариантов в отдельности. Первый вариант для получения прогноза продаж – построение модели скользящего среднего. Данная модель является одной из самых простых моделей временных рядов, которые используют для получения прогноза. Суть метода заключается в замене абсолютных значений ряда динамики на средние арифметические значения в определенных интервалах ряда [5]. Выбор интервалов происходит с помощью методов скользящего среднего: первые уровни постепенно убираются, последующие включаются. В результате получается сглаженный динамический ряд значений.

Данный метод легко реализуем даже с помощью типового табличного процессора и не требует особых навыков в области статистики. Недостатком метода является невозможность построения сложных зависимостей и низкая точность прогноза. Метод работает только тогда, когда для ряда четко прослеживается тенденция в динамике.

Модель авторегрессии – скользящего среднего (ARMA) базируется на предположении о том, что текущее значение исследуемого временного ряда зависит от линейной комбинации предыдущих значений временного ряда [9].

Модель ARMA обобщает две более простые модели временных рядов – модель авторегрессии AR и модель скользящего среднего MA. Для построения модели ARMA необходимо определить порядок модели и коэффициенты. Для определения порядка может быть использована автокорреляционная функция, а для определения коэффициентов можно использовать метод наименьших квадратов.

Модель ARIMA является расширенной моделью авторегрессии – скользящего среднего. Подход ARIMA заключается в оценке стационарности ряда. Различными тестами находят наличие единичных корней и порядок интегрирования временного ряда. Затем если необходимо, ряд может быть преобразован и для этой модели строится уже знакомая модель ARMA.

К преимуществам моделей ARMA и ARIMA можно отнести то, что эти модели имеют очень четкое математико-статистическое обоснование, что делает их одними из наиболее научно обоснованных моделей из всего множества моделей прогнозирования тенденций во временных рядах [9].

Один из явных недостатков моделей заключается в требовании к рядам данных: для построения адекватной модели ARIMA требуется не менее 40 наблюдений, а для ARMA – порядка 6–10 сезонов, что на практике не всегда возможно.

Модель регрессии позволяет использовать при прогнозировании причинные факторы, например, такие как цены, рекламные акции и другие экономические показатели [16].

Грамотно спроектированная модель регрессии дает значительное понимание взаимосвязей между переменными и учитывает сценарии «что, если». Например, модель регрессии включает цену в качестве параметра. Путем количественной оценки взаимосвязи между продажами и ценами на товары можно создавать прогнозы, отвечающие на такие вопросы, как «Что,

если мы поднимем цену на товар?» или «Что, если мы снизим цену на товар?» [9]. Создание таких альтернативных прогнозов может помочь в определении эффективной стратегии ценообразования.

Недостатком метода является то, что при построении такой модели могут быть использованы независимые параметры, и плохие прогнозы для таких параметров приведут к плохим прогнозам модели в целом.

В силу того, что спрос на услуги парка развлечений является зависимым уже сейчас от двух факторов - погодных условий и дня недели, а в дальнейшем факторов может быть больше, то в качестве математического метода прогнозирования спроса остановимся на модели регрессии. Использование множественной линейной регрессии позволит анализировать неограниченное количество факторов.

Таким образом, в данном параграфе были рассмотрены способы решения задачи: описаны математические методы прогнозирования спроса, выбран способ решения задачи бакалаврской работы.

1.3 Анализ и выбор технологии для решения задачи

С каждым годом объем данных растет. По прогнозам компании International Data Corporation объем данных всего мира к 2025 году увеличится в 10 раз и составит 163 зеттабайт.

По мере накопления данных организацией возникает логичное желание полезного использования этих данных. Для этого данные анализируют, и на основе результатов принимают те или иные управленческие решения.

Для того, чтобы на каком языке реализовать регрессионный анализ для прогнозирования объемов продаж необходимо понять, на каком из этих языков программа будет работать лучше. В качестве критерия оценивания будем рассматривать процессорное время работы уже реализованной задачи поиска элемента в неотсортированном векторе целых чисел.

Данная задача решается линейным поиском. Алгоритм заключается в последовательном прохождении всех элементов входного вектора до тех пор, пока не будет найдено искомое значение (успешный поиск) или будет достигнут конец вектора (неудачный поиск) [1].

Для оценки будем использовать набор данных, состоящий из 1000000 уникальных целых чисел в диапазоне от 1 до 2000000, выполним поиск 1000 раз со всеми целыми числами от 1 до 1000. Вероятность успешного поиска в таком случае составит около 50%, поэтому в первой половине случаев алгоритм пройдет через весь вектор, чтобы сделать вывод, что поиск был неудачным. Во второй половине алгоритм должен сделать $(n+1)/2$ оценок, чтобы найти элемент, где n - длина вектора.

Сравнение процессорного времени реализаций поиска элемента в неотсортированном векторе целых чисел на R, Python и Julia представлено на рисунке 2.

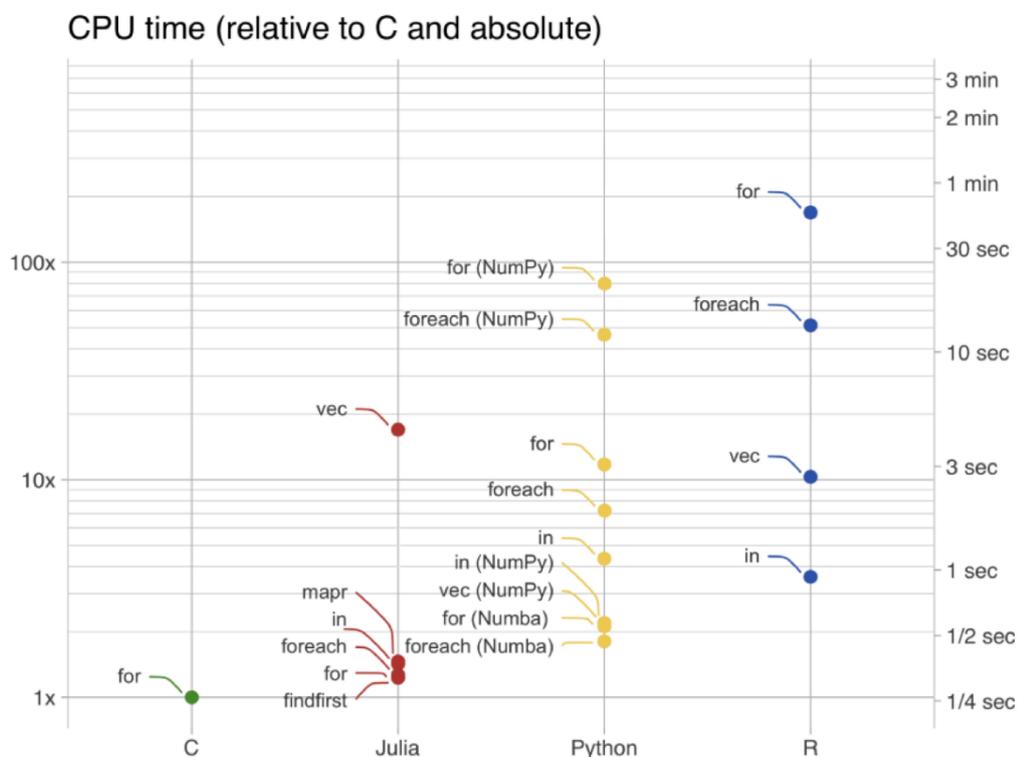


Рисунок 2 - Процессорное время реализаций поиска элемента в неотсортированном векторе целых чисел на R, Python и Julia

По полученным результатам видим, что самым медленным решением задачи стала реализация на R, а самым быстрым - Julia. В силу того, что Julia зачастую в настоящее использует библиотеки Python и имеет не такие активные человеческие сообщества поддержки, где люди делятся проблемами программ, написанных на этом языке, и находят решение этих проблем, в качестве языка программирования для реализации алгоритма регрессионного анализа выберем Python.

Таким образом, в данном параграфе были рассмотрены способы решения поставленной задачи на разных языках программирования, для реализации задачи бакалаврской работы был выбран Python.

Выводы по разделу 1:

1. в ходе анализа деятельности ООО «Парк Горького» была выявлена потребность в разработке инструмента, позволяющего выполнять прогнозирование объема продаж в разрезе определенных заказчиком факторов;

2. проведен анализ математических методов для решения задачи прогнозирования продаж, на основе выполненного анализа выбран метод линейной регрессии;

3. проведен анализ технологий, с помощью которых можно эффективно реализовать выбранный на предыдущем шаге метод линейной регрессии, языком программирования выбран Python.

2 Проектирование приложения для прогнозирования продаж на основе принципов регрессионного анализа

2.1 Описание этапов проектирования приложения

Проектирование программного продукта, решающего задачу прогнозирования продаж парка аттракционов, включает в себя:

- формирование структуры разрабатываемого продукта;
- выбор представления входных и выходных данных;
- создание макета пользовательского интерфейса;
- описание математического аппарата;
- определение готовых библиотек для решения задачи;
- реализация математического алгоритма;
- реализация подключения к дополнительным сервисам для построения прогноза продаж (API-подключение к сервису погоды);
- реализация пользовательского интерфейса и вывод результатов прогнозирования продаж на форму пользователя;
- тестирование и передача продукта в опытно-промышленную эксплуатацию.

Структура разрабатываемого продукта – набор функций, необходимых для решения задачи программным способом. Перед тем, как перейти к описанию работы программы, дадим определения входным данным и начальным условиям в контексте данной работы.

Входные данные – факторы, влияющие на уровень продаж, а также сами данные о продажах. Фактически это файлы, которые пользователь должен указать для того, чтобы получить прогноз продаж.

Начальные условия – условия, при которых пользователь хочет получить прогноз продаж. Это обязательно период получаемого прогноза и прогноз погоды для этого периода.

Идея получения прогноза продаж с использованием множественной линейной регрессии заключается в том, что входные данные проходят через некоторую функцию, которая построит на основе этих данных уравнение регрессии, и по начальным данным, указываемых пользователем, получает итоговые значения выручки для каждого дня периода, указанного пользователем.

Принцип работы ясен, перейдем к построению диаграммы компонентов. В предыдущем разделе для реализации был выбран язык программирования Python, для реализации пользовательского интерфейса будем использовать совместимую технологию QT.

Главный модуль, в котором будет прописана логика интерфейса и вызов функций приложения, назовем «main_ui». Функции, в которую как раз попадают входные параметры, дадим название «regression», из этой функции будет происходить вызов функции «get_weather». Функция «get_weather» необходима для подключения к сервису погоды и получения прогноза погоды для дат, заданных в начальных условиях. Также необходим элемент, в котором будет размещен, так называемый, дизайн. Это будет «main_des», который связан с главным модулем. Точно известно, что есть входные параметры, сейчас это «weather» и «sales», на этапе опытно-промышленной эксплуатации заказчиком параметры могут быть скорректированы. Диаграмма компонентов представлена на рисунке 3.

В настоящее время заказчик ведет учет продаж в системе 1С и планирует переход на другую систему. Также известна структура данных для анализа:

- дата выручки;
- сумма выручки.

Так как нам необходимо получение взаимосвязи и с днем недели тоже, то добавим в эту структуру атрибут дня недели. Получение этого атрибута автоматически из базы 1С не составит труда.

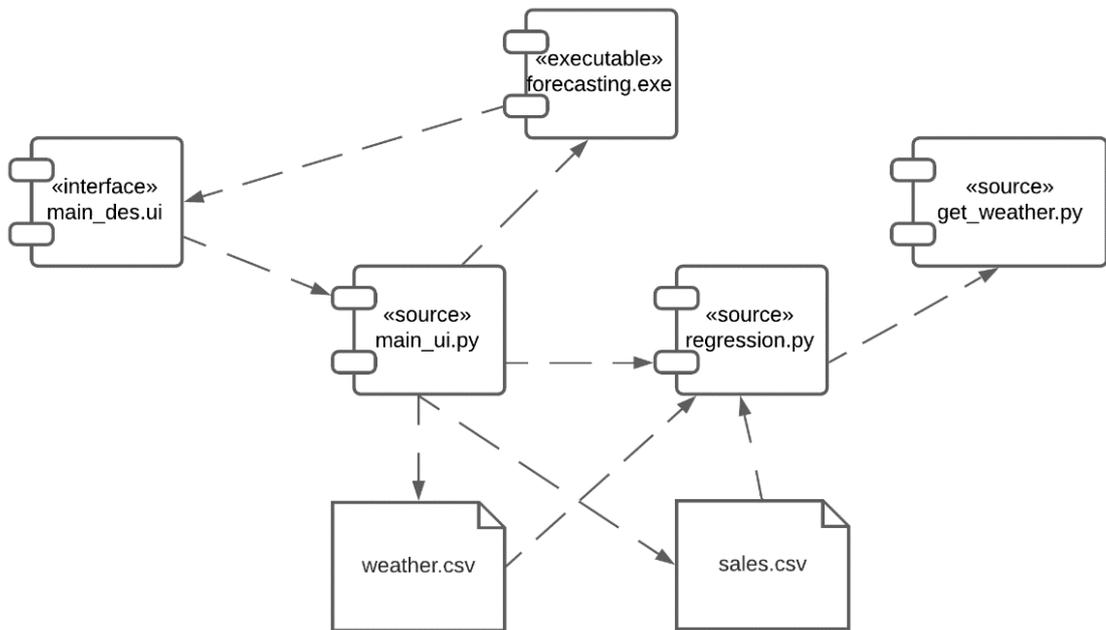


Рисунок 3 - Диаграмма компонентов

Функциональная модель программного продукта представлена на рисунке 4.



Рисунок 4 - Функциональная модель в IDEF0

В качестве формата входных данных предлагается использовать файл в формате csv – текстовый файл, предназначенный для представления табличных данных. Данный формат удобен при обработке на Python, данные из базы 1С также можно выгрузить в этом формате, используя функционал пользовательских обработок, размещенных в свободном доступе.

Выходные данные должны выводиться на экран пользователя с возможностью их копирования для дальнейшей обработки пользователем в табличном процессоре. Структура выходных данных:

- дата для планирования;
- планируемая выручка.

Затем данные проходят обработку, и пользователь видит результат прогноза. Макет интерфейса главного окна программы представлен на рисунке 5.

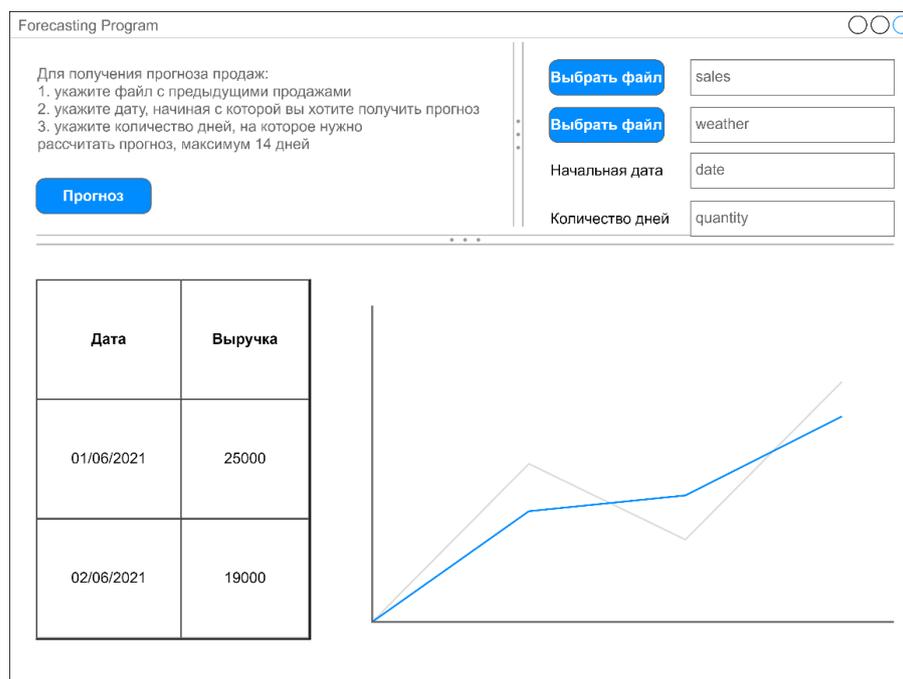


Рисунок 5 - Макет пользовательского интерфейса

Таким образом, в данном параграфе была описаны основные этапы проектирования разрабатываемого продукта.

2.2 Основы регрессионного анализа

В качестве метода прогнозирования был выбран метод линейной регрессии. Метод основан на установлении зависимости между двумя переменными. Модель линейной регрессии часто используется в эконометрике – науке, изучающей количественные и качественные экономические взаимосвязи. Рассмотрим подробнее этот метод.

Регрессия описывается в таком разделе математики, как статистика. Основными задачами математической статистики являются:

- указание способов сбора и группировки статистических сведений, полученных в результате наблюдения или в результате специально поставленных экспериментов;

- разработка методов анализа статистических данных в зависимости от целей исследования (оценка неизвестной вероятности события, оценка неизвестной функции распределения, оценка параметров распределения, вид которого известен, оценка зависимости случайных величины от одной или нескольких случайных величин, проверка статистических гипотез о виде неизвестного распределения или о величине параметров распределения, вид которого известен) [13].

Пусть поставлена задача изучить совокупность однородных объектов относительно некоторого качественного или количественного признака. В таком случае можно провести сплошное обследование - обследование каждого из объектов в совокупности относительно какого-либо признака. На практике сплошное обследование проводят редко, обычно случайным образом отбирают ограниченное число объектов и подвергают их изучению.

Выборочной совокупностью или выборкой называют совокупность случайно отобранных объектов. Генеральной совокупностью называют совокупность объектов, из которых производится выборка. Объемом совокупности называют число объектов этой совокупности [16].

Если значения выборки расположить в порядке возрастания, то полученный ряд называется вариационным. Разность между минимальным и максимальным элементами выборки называется размахом вариации.

Когда выборка содержит большое количество элементов или элементы выборки фактически не повторяются, то используют интервальный статистический ряд. Для его построения интервал, содержащий элементы выборки, разбивается на некоторое количество непересекающихся интервалов.

Пусть в результате измерений в процессе опытов получена некоторая зависимость одного признака от другого, требуется найти формулу, выражающую эту зависимость аналитически. Для этого задача ставится таким образом, чтобы обязательно учитывался характер исходной функции. Данная задача решается методом наименьших квадратов.

Статистические данные могут быть проанализированы методами корреляционного и регрессионного анализа.

Корреляционный и регрессионный анализы являются смежными разделами математической статистики и предназначены для изучения зависимости одного показателя от другого. Исследования взаимосвязи величин приводит к теориям корреляции, как к разделу теории вероятностей, так и к разделу математической статистики. Исследования зависимости случайной величины от ряда неслучайных и случайных величин приводят к модели регрессии и регрессионному анализу на базе выборочных данных.

Корреляционной зависимостью между двумя переменными величинами называется функциональная зависимость между значениями одной из них и условным математическим ожиданием другой [3]. Корреляционная зависимость двух случайных величин задается моделью, зависящей от этих двух величин.

Парная корреляция занимается изучением характеристик взаимосвязи двух случайных величин. Основой получения этих характеристик является совместное распределение и коэффициент корреляции. Наиболее

естественной оценкой коэффициента корреляции является выборочный коэффициент корреляции.

При изучении статистических связей форму связей характеризует функция регрессии. В случае парной регрессии имеется только детерминированный фактор и регрессионная модель.

Оценка параметров регрессии в условиях конкретной ситуации проводится по статистической совокупности, которая рассматривается в качестве выборки с помощью метода наименьших квадратов.

В качестве меры того, насколько хорошо регрессия описывает данную систему наблюдений, служит коэффициент детерминации. При этом за базу сравнения принято описание с помощью среднего арифметического.

В случае множеств корреляции основой статистического анализа служит матрица выборочных значений. В силу того, что парная зависимость осложнена влиянием случайных величин, наряду с матрицей выборочных значений также используются выборочные характеристики множественной корреляции.

Сравнивая выборочные обычные и частные коэффициенты парной корреляции, можно сделать вывод о том, насколько взаимозависимость между двумя величинами вызвана их собственной взаимозависимостью и зависимостью каждой из них от других случайных величин.

Уравнение множественной регрессии отражает корреляционную связь результативной (зависимой) переменной и нескольких объясняющих (независимых).

Термин «регрессия» был введён Фрэнсисом Гальтоном в конце 19-го века. Гальтон обнаружил, что дети родителей с высоким или низким ростом обычно не наследуют выдающийся рост и назвал этот феномен «регрессия к посредственности». Сначала этот термин использовался исключительно в биологическом смысле. После работ Карла Пирсона этот термин стали использовать и в статистике.

Примеры регрессионных моделей:

- линейные функции;
- алгебраические полиномы;
- ряды Чебышёва;
- нейронные сети без обратной связи, например, однослойный перцептрон Розенблатта, радиальные базисные функции и прочее.

Термину регрессионная модель, используемому в регрессионном анализе, можно сопоставить синонимы: «теория», «гипотеза». Эти термины пришли из статистики, в частности из раздела «проверка статистических гипотез». Регрессионная модель есть прежде всего гипотеза, которая должна быть подвергнута статистической проверке, после чего она принимается или отвергается.

Регрессионная модель имеет вид, представленный на формуле (1).

$$y = f(x, b) + \varepsilon, E(\varepsilon) = 0 \quad (1)$$

где b - параметры модели;

ε - случайная ошибка модели.

Функция y называется линейной регрессией, если функция $f(x, b)$ имеет следующий вид, представленной на формуле (2).

$$f(x, b) = b_0 + b_1x_1 + b_2x_2 + \dots + b_kx_k \quad (2)$$

где b_j — параметры (коэффициенты) регрессии;

x_j — факторы регрессии;

k - количество факторов модели.

Коэффициенты линейной регрессии показывают скорость изменения зависимой переменной по данному фактору, при фиксированных остальных факторах (в линейной модели эта скорость постоянна) [9].

Параметр b_0 при котором нет факторов, называют часто константой. Формально — это значение функции при нулевом значении всех факторов.

Для аналитических целей удобно считать, что константа — это параметр при «факторе», равном 1 (или другой произвольной постоянной, поэтому константой называют также и этот «фактор»). В таком случае, если перенумеровать факторы и параметры исходной модели с учетом этого (оставив обозначение общего количества факторов — k), то линейную функцию регрессии можно записать в следующем виде, представленном на формуле (3).

$$f(x, b) = b_1x_1 + b_2x_2 + \dots + b_kx_k = \sum_{i=1}^k b_i x_i = x^T b \quad (3)$$

где $x^T = (x_1, x_2, \dots, x_k)$ - вектор регрессоров;

$b = (b_1, b_2, \dots, b_k)^T$ - вектор-столбец параметров (коэффициентов).

Линейная модель может быть как с константой, так и без константы. Тогда в этом представлении первый фактор либо равен единице, либо является обычным фактором соответственно.

Регрессионная модель $f(w, x)$ — это параметрическое семейство функций, задающее отображение представлена на формуле (4).

$$f: W \times X \rightarrow Y \quad (4)$$

Для описания закономерностей задачи была выбрана регрессионная модель. В отличие от математической модели она более точная, хотя и не обладает такой гибкостью условий.

Регрессионная модель объединяет широкий класс универсальных функций, которые описывают некоторую закономерность. При этом для построения модели в основном используются измеряемые данные, а не знание свойств исследуемой закономерности. Такая модель часто не интерпретируема, но более точна. Это объясняется либо большим числом моделей-претендентов, которые используются для построения оптимальной

модели, либо большой сложностью модели. Нахождение параметров регрессионной модели называется обучением модели.

Таким образом, в данном параграфе были описаны основы регрессионного анализа.

2.3 Построение регрессионной модели задачи

Как говорилось ранее, выручка, полученная от продаж билетов на аттракционы, зависит на текущий момент от двух факторов. С учетом этих двух факторов составим модель задачи - описание с математической точки зрения алгоритма регрессионного анализа для задачи прогнозирования продаж. Прежде чем переходить к моделированию проанализируем входные параметры. В нашем случае это день недели и погодные условия.

Погодные условия зависят от уровня ясности неба и температуры. Ниже представлена классификация уровней ясности неба:

- ясно, идентификатор - 1;
- небольшая облачность, идентификатор - 2;
- облачно, идентификатор - 3;
- пасмурно, идентификатор - 4;
- небольшой дождь, идентификатор - 5;
- дождь, идентификатор - 6.

Если количество факторов, влияющих на выручку, будет увеличено или один из факторов будет составным, как в случае с фактором погодных условий, то для гибкости системы необходимо использовать множественную линейную регрессию.

Для получения уравнения множественной регрессии используется период для расчета 3 месяца, т.е. период, когда возможны разные вариации погодных условий. Также необходим объем продаж за этот период.

Исходные данные для получения уравнения регрессии погодных условий будут выглядеть следующим образом, представленном в Таблица 2.

Таблица 2 - Структура для построения зависимости выручки от погоды

Уровень ясности неба	Температура	Выручка
1	30	27000
3	20	25000
6	18	10000
7	19	0

Определив вектор оценок коэффициентов регрессии, по расчетным формулам будет получено уравнение множественной регрессии для погодных условий. Промежуточным результатом по этому анализу будет вывод о том, что больше влияет на выручку: температура или уровень ясности неба.

Затем получим все возможные комбинации уровня ясности неба и температуры, эти значения подставим в полученное уравнение регрессии и таким образом найдем коэффициенты погоды для основной регрессионной модели. Эти коэффициенты рассчитываются один раз и являются статичными.

Следующим шагом получим связь между выручкой и днем недели. Для этого используем линейную регрессию. Исходные данные для получения уравнения регрессии дня недели будет выглядеть следующим образом, представленном в Таблица 3.

Таблица 3 – Структура для построения зависимости выручки от дня недели

День недели	Выручка
Понедельник	13000
Вторник	12000
Среда	15000
Четверг	15500
Пятница	18000
Суббота	22000
Воскресенье	21000

В силу того, что влияние дня недели на выручку не статично и изменчиво, то для расчета этих коэффициентов будет браться последний месяц. Затем сформируем для каждого дня недели свой усредненный коэффициент по полученному уравнению регрессии дня недели.

Тогда исходные данные для итогового уравнения множественной регрессии будут выглядеть следующим образом, представленном в Таблица 4 ниже.

Таблица 4 – Структура для построения зависимости выручки от погоды и дня недели

День недели	Погодные условия	Выручка
Суббота	1,5	19000
Среда	1,9	13000
День недели	Погодные условия	Выручка
Понедельник	0,3	2500
Вторник	1,5	7800

Следовательно, основываясь на полученных коэффициентах погоды и дня недели, будет найдено уравнение множественной регрессии в зависимости выручки от дня недели и погодных условий. В итоге также будет получен вывод о том, что больше влияет на выручку: погода или день недели.

С учетом того, что входными параметрами для построения уравнения линейной регрессии сейчас являются два параметра: погодные условия и день недели, то финальное уравнение находится по формуле (3).

$$y = (a_1 + b_1x) \times (a_2 + b_2x) \quad (5)$$

где a_1 – свободный член линии оценки зависимости дня недели;

b_1 - градиент оценённой линии зависимости дня недели;

a_2 – свободный член линии оценки зависимости погоды;

b_2 - градиент оценённой линии зависимости погоды.

Как видим, мы уже используем уравнение множественной линейной регрессии, для гибкости системы в случае увеличения заказчиком количества анализируемых параметров финальное уравнение множественной линейной регрессии имеет вид, представленный на формуле (6).

$$y = a + b_1x_1 + \dots + b_nx_n + e \quad (6)$$

где a – свободный член;

b_i - регрессионные коэффициенты в момент времени i ;

e - член, содержащий ошибку.

Отличие между простой и множественной линейной регрессией заключается в том, что вместо линии регрессии в ней используется гиперплоскость.

Таким образом, в данном параграфе была описана регрессионная модель задачи. На основе построенной модели реализуется возможность получения прогноза продаж в будущем периоде.

Выводы по разделу 2:

- описаны этапы проектирования приложения для прогнозирования продаж с учетом внешних факторов;
- построены диаграммы, поясняющие структуру разрабатываемого приложения;
- разработан макет пользовательского интерфейса;
- описаны основы регрессионного анализа: задачи статистики, применение регрессии, формулы, поясняющие суть выбранного метода;
- построена регрессионная модель для поставленной задачи прогнозирования продаж;

– выполнение этапов, связанных с разработкой описано в следующем разделе.

3 Реализация приложения для прогнозирования продаж на основе принципов регрессионного анализа

3.1 Реализация алгоритма регрессионного анализа

Финальное регрессионное уравнение, которое и будет отвечать на вопрос бакалаврской работы прогнозирования объема продаж парка аттракционов, зависит от регрессионного уравнения зависимости погодных условий и выручки и от регрессионного уравнения дня зависимости дня недели и выручки.

Из базы 1С необходимо взять данные о продажах, стандартные отчеты, обычно выводящие данные в табличный документ использовать не будем, напишем свой запрос, которым получим дату, день недели и выручку. Текст запроса представлен на рисунке 6. Полученную выборку сохраним в файл csv.

```
ВЫБРАТЬ
    ВыручкаИСебестоимостьПродаж.Период КАК date_,
    ДЕНЬНЕДЕЛИ (ВыручкаИСебестоимостьПродаж.Период) КАК day_of_week,
    СУММА (ВыручкаИСебестоимостьПродаж.СуммаВыручки) КАК sales
ИЗ
    РегистрНакопления.ВыручкаИСебестоимостьПродаж
    КАК ВыручкаИСебестоимостьПродаж

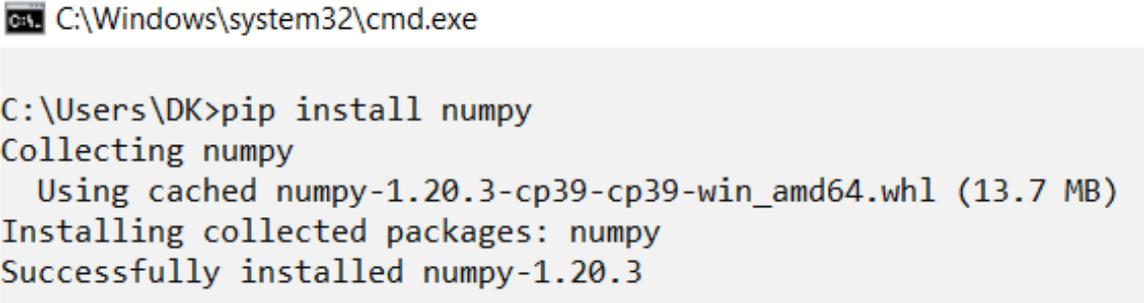
СТРУППИРОВАТЬ ПО
    ВыручкаИСебестоимостьПродаж.Период

УПОРЯДОЧИТЬ ПО
    date_
```

Рисунок 6 - Запрос для получения выручки из 1С:Управление Торговлей

Получив эти данные, можно составить уравнение зависимости выручки от дня недели. Программный код всего, что связано с регрессионным анализом, будем писать в отдельном файле regression.py.

Для работы с файлом csv необходимо подключить библиотеку csv, она является стандартной и отдельно ее устанавливать не нужно. Для работы с массивами будем использовать numpy [2], ее необходимо установить отдельно, установка для OS Windows представлена на рисунке 7.



```
C:\Windows\system32\cmd.exe

C:\Users\DK>pip install numpy
Collecting numpy
  Using cached numpy-1.20.3-cp39-cp39-win_amd64.whl (13.7 MB)
Installing collected packages: numpy
Successfully installed numpy-1.20.3
```

Рисунок 7 - Установка numpy

Библиотека для работы с датами calendar уже является встроенной, из этой библиотеки будем использовать класс timedelta для нахождения временного отрезка.

Функцию для нахождения регрессионных уравнений будем использовать одну, в зависимости от того, какую зависимость хотим получить, будем подавать разные входные данные. Входные данные функции - список, для удобства будем говорить матрица, в которой последний столбец это выручка, а впереди стоящие - факторы, оказывающее влияние на нее.

Пусть, что значения, оказывающие влияние на выручку, - новая матрица X , а сама выручка - Y . К матрице X добавим слева единичный столбец и транспонируем матрицу, получив X^T , затем умножаем матрицу X на X^T , находим обратную матрицу этого произведения $X \cdot X^T$. Затем умножаем отдельно транспонированную матрицу X^T на Y . Расчеты подготовлены, переходим к нахождению регрессионного уравнения: вектор оценок коэффициентов регрессии $y(x) = (X^T \cdot X)^{-1} \cdot (X^T \cdot Y)$. Реализация метода множественной регрессии представлена на рисунке 8.

```

# составим матрицы для решения линейного уравнения
matrix = np.array(
    [[len(table), last_line[1], last_line[2]],
     [last_line[1], last_line[3], last_line[7]],
     [last_line[2], last_line[7], last_line[4]]]
)
v = np.array([last_line[0], last_line[5], last_line[6]])
# y = k0 + k1x1 + k2x2.Массив с коэффициентами на выходе
k0 = np.linalg.solve(matrix, v)[0]
k1 = np.linalg.solve(matrix, v)[1]
k2 = np.linalg.solve(matrix, v)[2]
return k0, k1, k2

```

Рисунок 8 - Функция solve_regress

Сначала на вход подаем файл csv со следующим набором информации:

- date - дата;
- day_of_week - день недели;
- sales - выручка.

Открытие файла для записи [12] в переменную входных данных для первого вызова алгоритма регрессии представлено на рисунке 9.

```

def get_csv_first():
    file_name, _ = QFileDialog.getOpenFileName(window,
                                              '', 'Выручка',
                                              'CSV files (*.csv)')
    if len(file_name) > 0:
        form.lineEdit.setText(file_name)

```

Рисунок 9 - Открытие файла для чтения

Читаем таблицу с днями неделями и выручкой и передаем ее как параметр в solve_regress, реализация представлена на рисунке 10.

```

day_table = []
reader = csv.reader(f)
next(reader) # пропускаем заголовок
for line in reader:
    data = line[0].split(';')
    a = [int(data[1]), int(data[2])]
    day_table.append(a)

day_table = np.array(day_table)
x = day_table[:, 0]
y = day_table[:, 1]

```

Рисунок 10 - Обработка входной таблицы

Аналогичным образом поступим и с составлением уравнения зависимости выручки от погоды, используя при этом файл данными о выручке.

Для того, чтобы составить уравнение погодных условий нам необходима историческая информация о погоде. Для получения более корректной оценки необходимы данные о погоде за 2 года:

- 2019 год, когда все организации работали в штатном режиме без каких-либо ограничений на количество посетителей;
- 2020 год, когда из-за пандемии был введен ряд ограничительных мер.

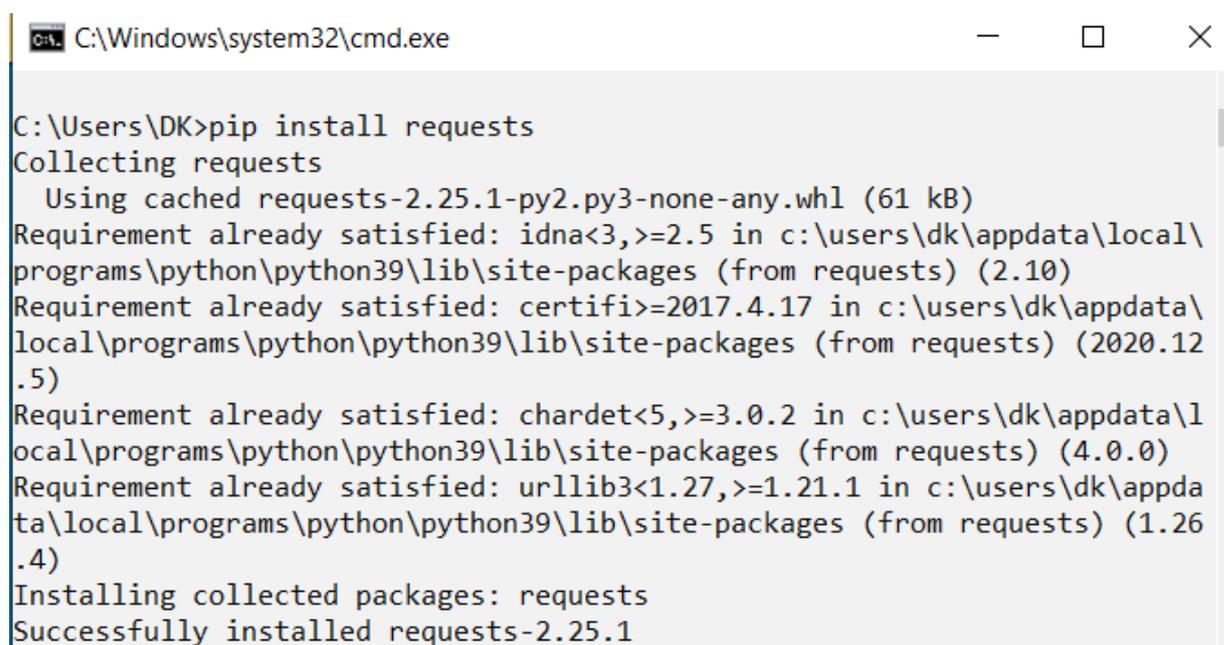
В 2021 году ограничения из-за коронавирусной инфекции также присутствуют, но эти ограничения были ослаблены по сравнению с 2020 годом.

Архивные значения погоды получены парсингом дневника погоды сервиса Gismeteo, алгоритм представлен в Приложении Б. Файл csv с архивными значениями погоды представляет собой следующий набор данных:

- date - дата;
- temperature - температура;
- sky - уровень ясности неба.

Чтобы получить прогноз продаж необходим прогноз погоды, для этого воспользуемся бесплатным подключением к API Яндекс.Погоды, API позволяет получать погодные данные в автоматизированном режиме. Технология Meteum оценивает фактическую погоду и строит прогноз для любых координат на территории России.

Для отправки HTTP-запросов необходимо установить библиотеку requests, установка на рисунке 11.



```
C:\Windows\system32\cmd.exe
C:\Users\DK>pip install requests
Collecting requests
  Using cached requests-2.25.1-py2.py3-none-any.whl (61 kB)
Requirement already satisfied: idna<3,>=2.5 in c:\users\dk\appdata\local\
programs\python\python39\lib\site-packages (from requests) (2.10)
Requirement already satisfied: certifi>=2017.4.17 in c:\users\dk\appdata\
local\programs\python\python39\lib\site-packages (from requests) (2020.12
.5)
Requirement already satisfied: chardet<5,>=3.0.2 in c:\users\dk\appdata\l
ocal\programs\python\python39\lib\site-packages (from requests) (4.0.0)
Requirement already satisfied: urllib3<1.27,>=1.21.1 in c:\users\dk\appda
ta\local\programs\python\python39\lib\site-packages (from requests) (1.26
.4)
Installing collected packages: requests
Successfully installed requests-2.25.1
```

Рисунок 11 - Установка requests

Форма запроса к API Яндекс.Погоды представлена на рисунке 12.

Параметры запроса:

- lat - широта в градусах, обязательное поле;
- lon - долота в градусах, обязательное поле;
- lang - сочетания языка и страны, для которых будут возвращены данные погодных формулировок;
- extra - расширенная информация об осадках.

```
GET https://api.weather.yandex.ru/v2/informers?  
  lat=<широта>  
  & lon=<долгота>  
  & [lang=<язык ответа>]  
  
X-Yandex-API-Key: <значение ключа>
```

Рисунок 12 - Форма запроса к API Яндекс.Погоды

Населенный пункт указывается при помощи значений широты (параметр `lat`) и долготы (параметр `lon`). Если данные параметры заданы некорректно, ответ будет содержать код ошибки 404.

Для построения уравнения нам необходима не только температура, но и уровень ясности неба, то для хранения значений этого фактора будем использовать словарь, работа со словарем уровня ясности неба представлена на рисунке 13.

```
dict_to_decode = {  
    'clear': 1,  
    'partly-cloudy': 2,  
    'cloudy': 3,  
    'overcast': 4,  
    'drizzle': 5,  
    'light-rain': 5,  
    'rain': 5,  
    'moderate-rain': 5,  
    'heavy-rain': 5,  
    'continuous-heavy-rain': 5,  
    'showers': 5,  
    'wet-snow': 5,  
    'light-snow': 5,  
    'snow': 5,  
    'snow-showers': 5,  
    'hail': 5,  
    'thunderstorm': 5,  
    'thunderstorm-with-rain': 5,  
    'thunderstorm-with-hail': 5  
}
```

Рисунок 13 - Словарь для работы с погодой

Результирующее уравнение получаем следующим образом, представленном на рисунке 14.

```
end_table = []
for i, line in enumerate(weather_table):
    day_predict = b0 + b1*x[i]
    weather_predict = c0 + c1*line[1] + c2*line[2]
    end_table.append([y[i], day_predict, weather_predict])

e0, e1, e2 = solve_regress(end_table)
```

Рисунок 14 - Финальное уравнение регрессии

Посмотрим, как связаны между собой заявленные заказчиком параметры. Сделать это можно, рассчитав коэффициенты корреляции для столбцов составленных уравнений. В Python с использованием библиотеки Numpy это делается вызовом функции corr. Рассчитаем коэффициент корреляции для исходных данных, расположенных в приложении. Вызов функции расчета коэффициента корреляции представлен на рисунке 15.

```
rho = np.corrcoef(sales, weather)
print (rho)
```

Рисунок 15 - Получение коэффициента корреляции

Значения на главной диагонали корреляционной матрицы (верхний левый и нижний правый) равны 1. Верхнее левое значение соответствует коэффициенту корреляции для x , а нижнее правое значение - коэффициент корреляции для y . Они всегда равны 1. Полученный коэффициент корреляции представлен на рисунке 16.

Получения прогноза уровня продаж происходит в функции get_future, реализация, которой представлена на рисунке 17.

```

===== RESTART: C:/Users/DK/Documents/regression.py =====
[[ 1.  0.7]
 [0.7  1. ]]
>>>

```

Рисунок 16 - Вывод коэффициента корреляции

```

def get_future(b0, b1, c0, c1, c2, e0, e1, e2, date_from, date_to):
    result_table = []
    from_api_table = Test_api.forecast()
    # заполним нужными датами массив
    req_date = []
    for i in range((date_to - date_from).days + 1):
        req_date.append(date_from + timedelta(days=i))
    req_date = [str(i) for i in req_date]

    for i in from_api_table:
        if i[0] in req_date:
            f_date = i[0]
            day = calendar.weekday(int(f_date[0:4]),
                                   int(f_date[5:7]),
                                   int(f_date[8:10])) + 1

            heaven = i[1]
            temp = i[2]

            fday_predict = b0 + b1*day
            fw_predict = c0 + c1*heaven + c2*temp

            future = e0 + e1*fday_predict + e2*fw_predict

            result_table.append([i[0], round(future)])
    return result_table

```

Рисунок 17 - Фрагмент функции get_future

Описание входных параметров функции get_future:

- b0, b1 - коэффициенты уравнения зависимости выручки от дня недели;
- c0, c1, c2, e1, e2 - коэффициенты, отвечающие за критерии температуры и уровня ясности неба, в уравнении зависимости выручки от погоды;
- date_from - начальная дата; date_to - конечная дата.

Таким образом, в данном параграфе было описано получение регрессионных уравнений зависимости выручки от дня недели и от погоды,

полученное уравнение используется для прогнозирования выручки следующих периодов методом регрессии.

3.2 Разработка пользовательского интерфейса

В параграфе проектирования программного продукта был разработан макет пользовательского интерфейса, в данном параграфе будет описана реализация этого макета.

В качестве технологии разработки пользовательского интерфейса был выбран QT. QT - программный продукт, выпущенный в 1993 году норвежской компанией Quasar Technologies, которая затем была переименована в Trolltech. QT доступна под двумя лицензиями: GPL и проприетарной, в рамках работы был выбран QT Designer с лицензией GPL. Для создания приложения необходимо установить и подключить библиотеку PyQt5. Библиотека подключается стандартной командой «`pip install PyQt5`». В QT Designer создадим Main Window Form и разместим на ней все элементы согласно макету. Полученный результат представлен на рисунке 18.

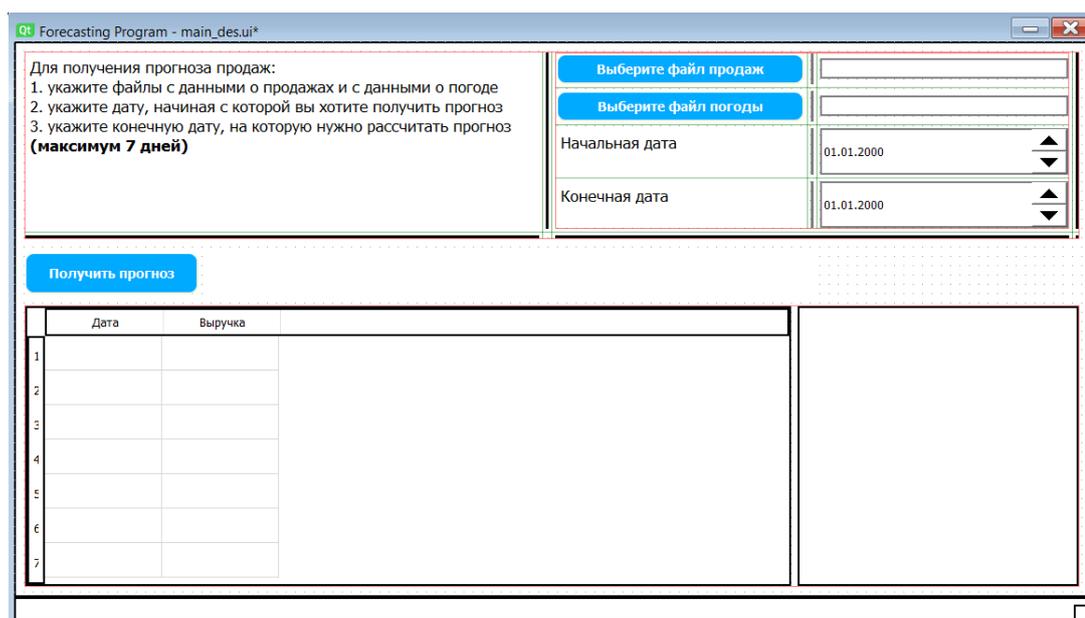


Рисунок 18 - Main Window Form

Чтобы при изменении размеров формы все элементы этой формы адаптировались, используем сетку и располагаем элементы внутри сетки. Окно Object Inspector представлено на рисунке 19.

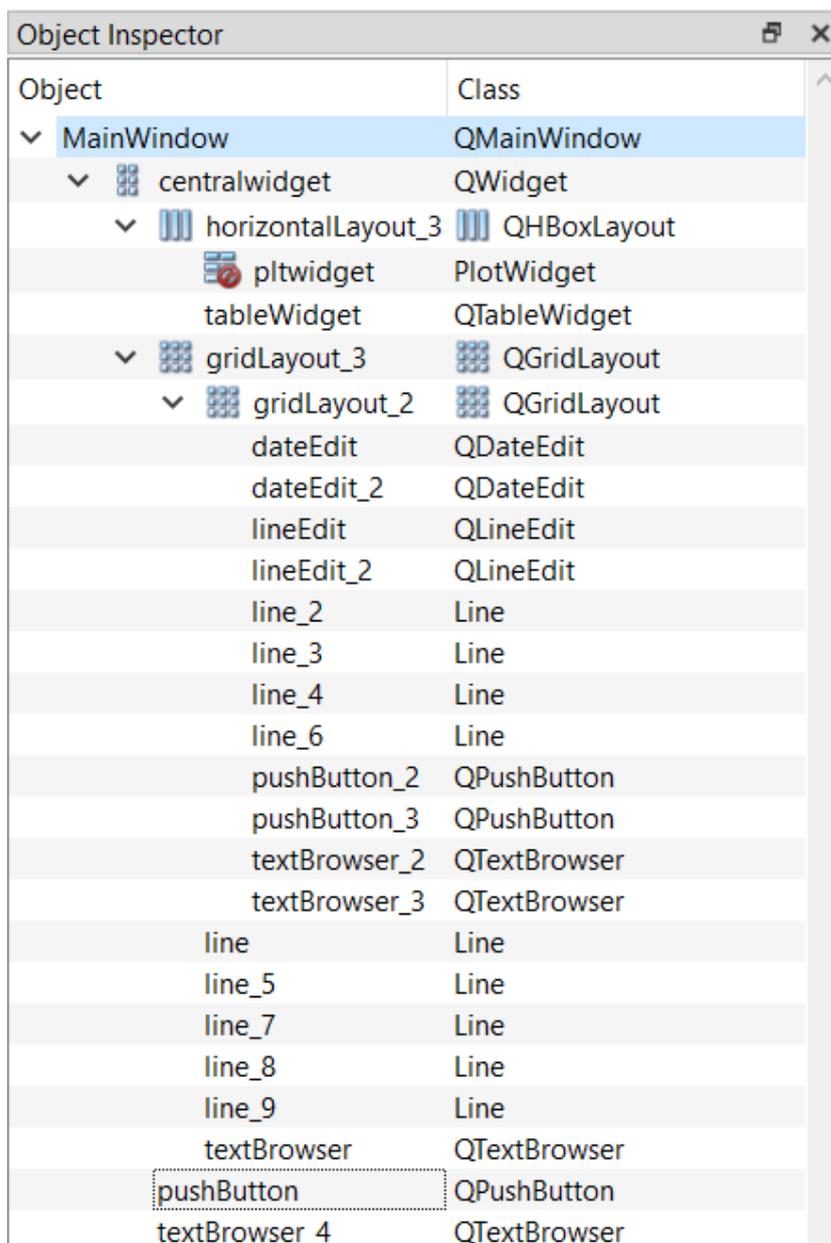


Рисунок 19 - Элементы Object Inspector

Затем необходимо связать форму с логикой самого приложения, для этого используем команду `ruic5` из командной строки. Вызов команды представлен на рисунке 20.

```
C:\Users\DK\Documents\forecasting>pyuic5 main_des.ui -o main_ui.py
```

Рисунок 20 - Использование команды pyuic5

В результате был создан файл main_ui.py, в котором уже можно работать над логикой приложения.

Функция открытия файлов csv уже была продемонстрирована в предыдущем параграфе, рассмотрим функцию, которая проверяет наличие этого файла перед началом анализа. Функция представлена на рисунке 21.

```
def correct_answers():
    window.statusBar().showMessage('')
    if form.lineEdit.displayText() == '' or form.lineEdit_2.displayText() == '':
        window.statusBar().showMessage(
            'Вы не указали файл с данными! Попробуйте снова')
    else:
        filename_first = os.path.abspath(form.lineEdit.displayText())
        filename_second = os.path.abspath(form.lineEdit_2.displayText())
        if os.path.exists(filename_first) and os.path.exists(filename_second):
            date_from = form.dateEdit.date().toPyDate()
            date_to = form.dateEdit_2.date().toPyDate()
            mat = regression.main_f(filename_first, filename_second, date_from, date_to)
            draw_table(matrix=mat)
        else:
            window.statusBar().showMessage(
                'Вы неправильно указали путь к файлу! Попробуйте снова')
```

Рисунок 21 - Обработка исключений

Результаты прогноза продаж выводятся пользователю в QTableWidgetItem, реализация функции вывода данных в таблицу представлена на рисунке 22.

```

def draw_table(matrix):
    for i in range(7):
        form.tableWidget.setItem(i, 0, QTableWidgetItem(''))
        form.tableWidget.setItem(i, 1, QTableWidgetItem(''))
    form.pltwidget.clear()
    x = []
    y = []
    for row in range(len(matrix)):
        date_for_view = '.'.join(matrix[row][0].split('-')[::-1])
        form.tableWidget.setItem(row, 0, QTableWidgetItem(date_for_view))
        form.tableWidget.setItem(row, 1, QTableWidgetItem(str(matrix[row][1])))

        x.append(row+1)
        y.append(matrix[row][1])

    form.pltwidget.plot(x, y, symbol='o', pen=0)

```

Рисунок 22 - Вывод данных прогноза продаж на форму

В этой же функции результаты попадают в PlotWidget- график прогноза продаж по дням недели.

Таким образом, в данном параграфе была описана реализация интерфейсной части приложения.

Выводы по разделу 3:

- реализован алгоритм множественной линейной регрессии, настроено API-подключение к сервису Яндекс.Погода для получения прогноза продаж;
- разработан пользовательский интерфейс с возможностью указания входных файлов для построения зависимости.

Заключение

Для любой коммерческой организации получение прибыли и минимизация издержек являются важными задачами. Для парка аттракционов ООО «Парк Горького» эта задача также является значимой, учитывая тот факт, что период деятельности парка ограничен теплым временем года. Погодные условия накладывают ограничения в целом на возможность работы, а день недели может как влиять на выручку положительно, если это выходной день, так и отрицательно, если это будний день. Именно эти факторы были предложены заказчиком в качестве входных параметров приложения прогнозирования объема продаж.

В работе был проведен анализ предметной области, выбраны и обоснованы математический метод прогноза и технология реализации, реализован алгоритм множественной линейной регрессии, позволяющий не только обрабатывать предложенные в настоящее время заказчиком факторы, но обрабатывать дополнительные параметры. Если количество параметров будет увеличено, то будет необходимо передать такие параметры в функцию построения уравнения линейной регрессии. Для понятного вывода информации об уровне продаж в будущем периоде был разработан пользовательский интерфейс.

Данные о погоде в ретроспективном периоде были получены парсингом дневника погоды, размещенном на сайте Gismeteo, источником погоды будущего периода является Яндекс.Погода, подключение к сервису выполнено с помощью API.

Также в работе были использованы нотация IDEF0 для построения функциональных моделей приложения и нотация UML для описания диаграммы компонентов.

Приложение написано на языке программирования Python, при выборе языка был проведен сравнительный анализ технологий, которые применяются для решения похожих задач. В результате этого анализа по

соотношению производительности к удобству использования был выбран Python. Пользовательский интерфейс был реализован с помощью технологии QT, которая имеет поддержку с Python.

Исполняемый файл приложения был скомпилирован для операционной системы Windows, но также может быть скомпилирован для Unix-подобных систем.

В результате было разработано кроссплатформенное приложение для прогноза продаж на примере ООО «Парк Горького». Таким образом, цель бакалаврской работы можно считать достигнутой.

Список используемой литературы

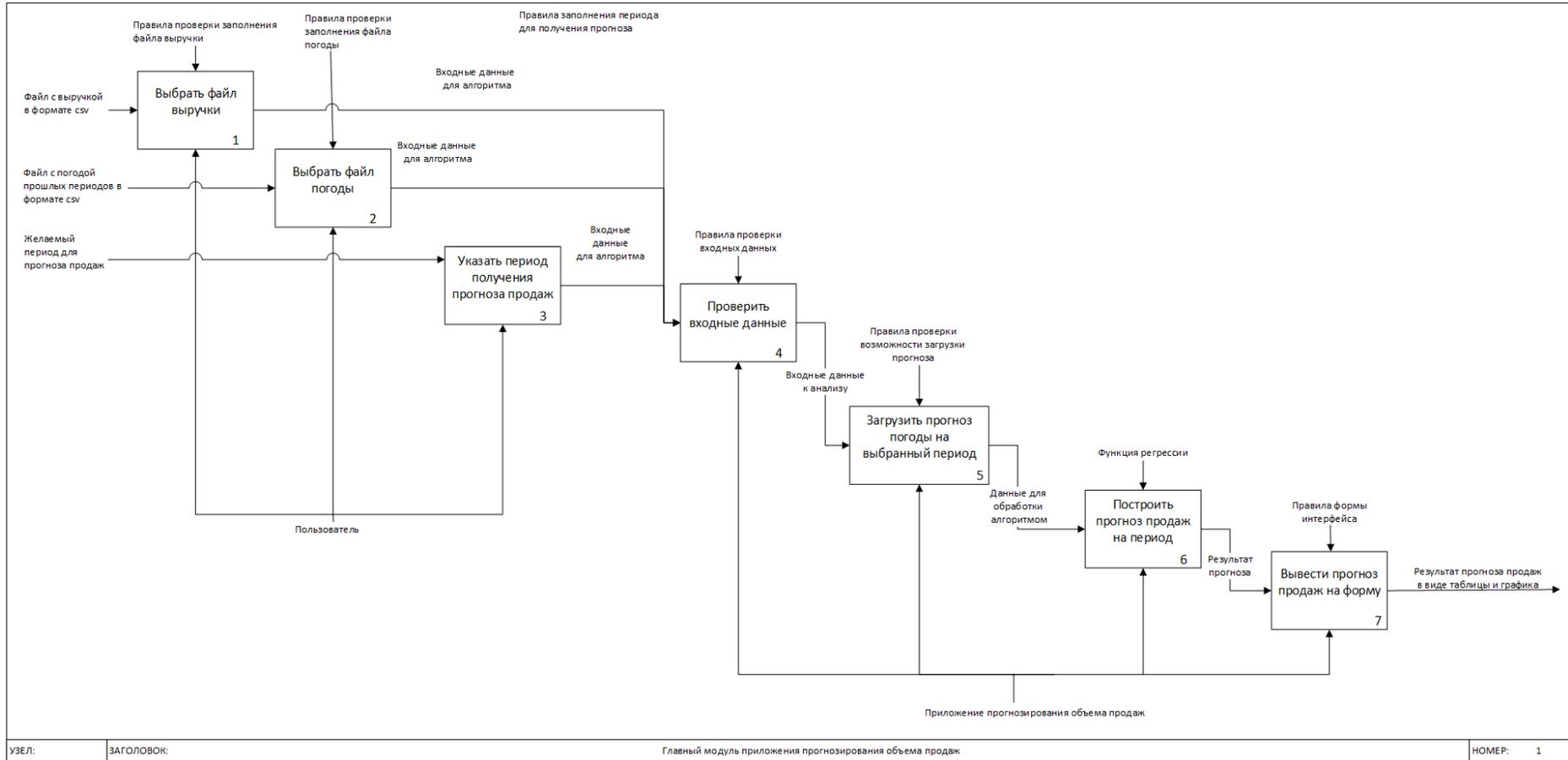
1. ГОСТ 34.601-90. Информационная технология. Комплекс стандартов на автоматизированные системы. Автоматизированные системы. Стадии создания. М. : Издательство стандартов, 2009. 6 с.
2. ГОСТ 19.404-79. Единая система программной документации. Пояснительная записка. Требования к содержанию и оформлению. М. : Издательство стандартов, 2010. 3 с.
3. Бейдер Д. Чистый Python. Тонкости программирования для профи. Санкт-Петербург : Изд. дом «Питер», 2018. 288 с.
4. Васильев А. Python на примерах. Практический курс. М : Наука и техника, 2018. 432 с.
5. Дерр В.Я. Теория вероятностей и математическая статистика : учеб. пособие. М: Астрель, 2021. 596 с.
6. Дрейпер Н., Смит Г. Прикладной регрессионный анализ. Множественная регрессия. Applied Regression Analysis. М. : «Диалектика», 2007. 912 с.
7. Иванов Б.Н. Теория вероятностей и математическая статистика : учеб. пособие. М : Издательство «Лань», 2019. 224 с.
8. Маккинни У. Python и анализ данных. М. : ДМК-Пресс, 2019. 540 с.
9. Максимов Ю.Д. Вероятностные разделы математики. Санкт-Петербург : «Иван Фёдоров», 2001. 592 с.
10. Мюллер А., Гвидо С. Введение в машинное обучение с помощью Python. Руководство для специалистов по работе с данными. Introduction to Machine Learning with Python: A Guide for Data Scientists. М. : Вильямс, 2017. 480 с.
11. Натан А. А., Горбачёв О. Г., Гуз С. А. Математическая статистика. : учеб. пособие. М.: МЗ Пресс МФТИ, 2004. 412 с.

12. Персиваль Г. Python. Разработка на основе тестирования. М : ДМК-Пресс, 2018. 622 с.
13. Плас Д.В. Python для сложных задач. Наука о данных и машинное обучение. Python Data Science Handbook: Essential Tools for Working with Data. Санкт-Петербург : Питер, 2017. 576 с.
14. Прохоренок Н., Дронов В. Python 3. Самое необходимое, 2-е изд. Санкт-Петербург : БХВ-Петербург, 2019. 608 с.
15. Радченко С. Г. Методология регрессионного анализа. К. : «Корнийчук», 2011. 376 с.
16. Рамальо Л. Python. К вершинам мастерства : [рус.] = Fluent Python. O'Reilly, 2015 : [пер. с англ.]. М : ДМК Пресс, 2016. 194 с.
17. Рейтц К., Шлюссер Т. Автостопом по Python. Санкт-Петербург : Издательский дом «Питер», 2017. 308 с.
18. Фёрстер Э., Рёнц Б. Методы корреляционного и регрессионного анализа. М.: Финансы и статистика, 1981. 302 с.
19. Шолле Ф. Глубокое обучение на Python. Санкт-Петербург : Питер, 2018. 400 с.
20. Adawadkar K. Python Programming - Applications and Future // International Journal of Advance Engineering and Research Development. 2017. P. 1-4.
21. Albalooshi F., Mahmood A. A Comparative Study on the Effect of Multiple Inheritance Mechanism in Java, C++, and Python on Complexity and Reusability of Code // International Journal of Advanced Computer Science and Applications (IJACSA). 2017. P. 33-37.
22. Beazley D.M. Python Essential Reference. Boston : Addison-Wesley Professional, 2009. 717 с.
23. Ekmekci B., McAnany C.E., Mura C. An Introduction to Programming for Bioscientists: A Python-Based Primer // PLOS Computational Biology. 2016. P. 110-114.

24. Fua P., Lis K. Comparing Python, Go, and C++ on the N-Queens Problem // Computer Vision Laboratory, EPFL. 2020. P. 18-19.
25. Muller E., Bednar J.A., Diesmann M., Gewaltig M., Hines M. Python in neuroscience // Frontiers in Neuroinformatics. 2015. P. 165-167.
26. Ozgur C., Colliau T., Rogers G., Hughes Z., Myer-Tyson E. MatLab vs. Python vs. R // Journal of Data Science. 2017. P. 355—372.
27. Peschke R., Nishimura K., Varner G. ARGG-HDL: A High Level Python Based Object-Oriented HDL Framework // IEEE Transactions on Nuclear Science. 2020. P. 28-30.
28. Rana Y. Python: Simple though an Important Programming language // International Research Journal of Engineering and Technology (IRJET). 2019. P. 1856—1858.

Приложение А

Функциональная модель главного модуля



Приложение Б

Фрагмент кода

```
import numpy as np
import csv
import calendar
import Test_api
from datetime import timedelta

def solve_regress(table):
    # [ [revenue, ясность, температура], ] - rev = y, ясн = x1, темп = x2
    # [ [revenue, day_predict, weather_predict], ] -
    rev = y, dayp = x1, weatp = x2
    for i in table:
        x1x1 = i[1] ** 2
        x2x2 = i[2] ** 2
        x1y = i[1] * i[0]
        x2y = i[2] * i[0]
        x1x2 = i[1] * i[2]
        yy = i[0] ** 2
        i.extend([x1x1, x2x2, x1y, x2y, x1x2, yy])
    table = np.array(table, dtype=np.float64)
    last_line = [sum(table[:, 0]), sum(table[:, 1]),
sum(table[:, 2]), sum(table[:, 3]),
sum(table[:, 4]), sum(table[:, 5]),
sum(table[:, 6]), sum(table[:, 7]),
sum(table[:, 8])]

    # составим матрицы для решения линейного уравнения
    matrix = np.array(
[[len(table), last_line[1], last_line[2]],
[last_line[1], last_line[3], last_line[7]]]
```

Продолжение приложения Б

```

[last_line[2], last_line[7], last_line[4]]
)
v = np.array([last_line[0], last_line[5], last_line[6]])
k0 = np.linalg.solve(matrix, v)[0]
k1 = np.linalg.solve(matrix, v)[1]
k2 = np.linalg.solve(matrix, v)[2]
return k0, k1, k2

def calc_equations(f, s, date_from, date_to):
    day_table = []
    reader = csv.reader(f)
    next(reader) # пропускаем заголовок
    for line in reader:
        data = line[0].split(';')
        a = [int(data[1]), int(data[2])]
        day_table.append(a)

    day_table = np.array(day_table)
    x = day_table[:, 0]
    y = day_table[:, 1]
    numerator = sum((x - np.average(x)) * (y - np.average(y)))
    denominator = sum((x - np.average(x)) ** 2)

    b1 = numerator / denominator
    b0 = np.average(y) - b1 * np.average(x)

    weather_table = []
    reader = csv.reader(s)
    next(reader) # пропускаем заголовок
    count = 0
    for line in reader:
        weather_data = line[0].split(';')
        b = [y[count], int(weather_data[2]), int(weather_data[1])]

```

Продолжение приложения Б

```

weather_table.append(b)
count += 1
c0, c1, c2 = solve_regress(weather_table) # c1 -
коэффициент, отвечающий за критерий чистоты неба
end_table = []
for i, line in enumerate(weather_table):
day_predict = b0 + b1*x[i]
weather_predict = c0 + c1*line[1] + c2*line[2]
end_table.append([y[i], day_predict, weather_predict])

e0, e1, e2 = solve_regress(end_table) # e1 -
отвечает за предсказание по дню, e2 - по погоде

return get_future(b0, b1, c0, c1, c2, e0, e1, e2, date_from, date_to)

def get_future(b0, b1, c0, c1, c2, e0, e1, e2, date_from, date_to):
result_table = []
from_api_table = Test_api.forecast()
# заполним нужными датами массив
req_date = []
for i in range((date_to - date_from).days + 1):
req_date.append(date_from + timedelta(days=i))
req_date = [str(i) for i in req_date]

for i in from_api_table:
if i[0] in req_date:
f_date = i[0]
day = calendar.weekday(int(f_date[0:4]),
int(f_date[5:7]),
int(f_date[8:10])) + 1
heaven = i[1]
temp = i[2]

```

Продолжение приложения Б

```
fday_predict = b0 + b1*day
fw_predict = c0 + c1*heaven + c2*temp

future = e0 + e1*fday_predict + e2*fw_predict

result_table.append([i[0], round(future)])
return result_table
def main_f(filename_first, filename_second, date_from, date_to):
    with open(filename_first, 'r') as f, open(filename_second, 'r') as s:
        return calc_equations(f, s, date_from, date_to)
```