

Министерство науки и высшего образования Российской Федерации  
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ  
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
«ОРЕНБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

Финансово-экономический факультет  
Кафедра математических методов и моделей в экономике

## **ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА**

Направление подготовки 01.03.04 Прикладная математика

**Математическое моделирование деятельности многофункционального  
центра (на примере ГАУ «МФЦ Иркутской области»)**

ОГУ 01.03.04. 1021. 898 ОО

Заведующий кафедрой  
канд. техн. наук, доцент

А.Г. Реннер

Руководитель  
канд. эконом. наук, доцент

О.С. Чудинова

Студент

О.Б. Чаганова

Оренбург 2021

Утверждаю  
заведующий кафедрой математических  
методов и моделей в экономике  
А.Г. Реннер  
«06» октября 2020 г.

## ЗАДАНИЕ

### на выполнение выпускной квалификационной работы

студентке Чагановой Ольге Борисовне

по направлению подготовки (специальности) 01.03.04 Прикладная математика

1 Тема ВКР Математическое моделирование деятельности многофункционального центра (на примере ГАУ «МФЦ Иркутской области»).

2 Срок сдачи студентом ВКР «22» июня 2021 г.

3 Цели и задачи ВКР: Целью ВКР является выработка рекомендаций по оптимизации работы МФЦ на основе методов прикладной математики.

Задачи ВКР:

1) анализ востребованности услуг МФЦ и выявление закономерностей в транзакциях посетителей центра;

2) моделирование деятельности МФЦ как системы массового обслуживания;

3) разработка рекомендательной системы для прогнозирования следующей услуги заявителя многофункционального центра.

4 Исходные данные к ВКР: информационная база МФЦ Иркутской области за период 06.01.2019 – 17.05.2020, включающая информацию о заявителях, истории подачи заявлений в МФЦ и талонах электронной очереди.

5 Перечень вопросов подлежащих разработке:

1) рассмотреть основные особенности организации деятельности многофункционального центра, которые необходимо учитывать при решении задач исследования;

2) провести обзор методов моделирования работы многофункционального центра, изложенных в научной литературе;

3) разработать алгоритм анализа и моделирования деятельности многофункционального центра;

4) провести анализ востребованности услуг МФЦ и выявить закономерности в транзакциях посетителей центра;

5) осуществить моделирование деятельности МФЦ как системы массового обслуживания;

6) разработать рекомендательной системы для прогнозирования следующей услуги заявителя.

6 Перечень графического (иллюстративного) материала рисунки, таблицы, графики

Дата выдачи и получения задания

Руководитель ВКР «06 » октября 2020г.

О.С. Чудинова

Студент

«06 » октября 2020г.

О.Б. Чаганова

## **Аннотация**

Выпускная квалификационная работа посвящена разработке математических моделей анализа и оптимизации деятельности многофункционального центра (на примере ГАУ «МФЦ Иркутской области»).

Работа состоит из введения, двух глав, заключения. Во введении обоснована актуальность, отражена степень проработанности темы, сформулирована цель, определены объект и предмет исследования, поставлены задачи, описана информационная база, перечислены инструментальные средства.

В первой главе представлены теоретические аспекты моделирования деятельности многофункционального центра. Рассмотрены особенности функционирования многофункционального центра, а также методы прикладной математики и машинного обучения, используемые при моделировании его деятельности. Предложен алгоритм анализа и моделирования деятельности многофункционального центра.

Во второй главе, посвященной реализации предлагаемого в работе алгоритма анализа и моделирования деятельности многофункционального центра, проведен анализ востребованности услуг МФЦ (на примере ГАУ «МФЦ Иркутской области») и выявление закономерностей в транзакциях заявителей при помощи методов поиска ассоциативных правил и последовательных шаблонов. Моделирование деятельности МФЦ как системы массового обслуживания позволило реализовать алгоритм нахождения оптимального распределения окон обслуживания по каждому типу услуг. Заключительной частью работы является разработка рекомендательной системы прогнозирования следующей услуги заявителя, в основе которой лежит модель рекуррентной нейронной сети долгой краткосрочной памяти.

В заключении содержатся рекомендации по оптимизации деятельности многофункционального центра №1 г. Иркутска, а также приведены основные результаты практической реализации предлагаемого алгоритма моделирования деятельности МФЦ.

Выпускная квалификационная работа изложена на 72 страницах, содержит 16 рисунков, 13 таблиц, 4 приложения. Список использованных источников включает 53 наименования работ отечественных и зарубежных авторов.

## **Abstract**

The final qualification work is devoted to the development of mathematical models for the analysis and optimization of the multifunctional center (on the example of the State Autonomous Institution «Multifunctional center of the Irkutsk Region»).

The work consists of an introduction, two chapters, and a conclusion. The relevance of the study is proved, the degree of elaboration of the topic is reflected, the goal is formulated, the object and subject of the study are defined, the tasks are set, the information base is described and the tools are listed in the introduction.

The first chapter presents the theoretical aspects of modeling the performance of a multifunctional center. The main features of the functioning of a multifunctional center, as well as the methods of applied mathematics and machine learning that can be used in modeling its performance are considered. An algorithm for analyzing and modeling the performance of a multifunctional center is proposed.

The second chapter is devoted to the implementation of the proposed algorithm for analyzing and modeling the performance of a multifunctional center. It contains an analysis of the demand for MfSC's services and results of applying association rules and sequential pattern mining algorithms to transaction database. Modeling the functioning of the MfSC as a queuing system made it possible to implement an algorithm for finding the optimal distribution of service windows for each type of service. The final part of the work is the development of a recommender system for predicting the next service of the applicant, which is based on a model of a recurrent neural network of long short-term memory.

The conclusion contains recommendations for optimizing the performance of the multifunctional center №1 in Irkutsk, as well as the main results of the implementation of the proposed algorithm.

The final qualification work is set out on 72 pages, contains 16 figures, 13 tables, 4 appendices. The list of sources used includes 53 titles of works by domestic and foreign authors.

## Содержание

Введение.....	6
1 Теоретические аспекты моделирования деятельности многофункционального центра.....	9
1.1 Назначение и особенности функционирования многофункционального центра.....	9
1.2 Обзор математических методов моделирования деятельности многофункционального центра .....	16
1.3 Концептуальная схема анализа и моделирования деятельности многофункционального центра .....	24
2 Разработка математических моделей выявления закономерностей в поступлении заявлений и организации процесса обслуживания заявок .....	37
2.1 Анализ востребованности услуг ГАУ «МФЦ Иркутской области» и выявление закономерностей в транзакциях заявителей .....	37
2.2 Моделирование деятельности многофункционального центра как системы массового обслуживания .....	46
2.3 Разработка рекомендательной системы для прогнозирования следующей услуги заявителя многофункционального центра.....	57
Заключение .....	65
Список использованных источников .....	68
Приложение А (обязательное) Исходные данные (фрагменты) .....	73
Приложение Б (обязательное) Программный код алгоритмов поиска ассоциативных правил и последовательных шаблонов на языке R .....	75
Приложение В (обязательное) Программный код для моделирования системы массового обслуживания на языке Python .....	81
Приложение Г (обязательное) Программный код рекомендательной системы на языке Python.....	90

## Введение

Развитие сферы оказания муниципальных и государственных услуг – крайне важный вопрос при реализации социально-экономической политики любого государства. С каждым годом предъявляются все более высокие требования к качеству, эффективности и доступности процесса предоставления услуг населению, индивидуальным предпринимателям и юридическим лицам. При этом принятие любых управленческих решений всегда требует оптимизации использования ресурсов по причине ограниченности корпоративных и государственных бюджетных средств, роста цен на энергоносители и другие виды сырья и материалов, кризисных явлений в экономике. В связи с этим перед учеными и исследователями ставится задача поиска новых организационных и технологических форм и методов предоставления услуг, позволяющих решить обозначенные проблемы и при этом повысить качество оказания услуг населению.

Непрерывное совершенствование информационных технологий, рост числа государственных и муниципальных услуг и усложнение процесса их получения послужило причиной создания специальных органов предоставления таких услуг. К ним относятся многофункциональные центры оказания государственных и муниципальных услуг (МФЦ). Многофункциональные центры функционируют в России уже более десяти лет, выполняя важнейшую функцию организации предоставления государственных и муниципальных услуг на принципах «одного окна».

Цифровизация государственных учреждений, в частности, многофункциональных центров оказания государственных и муниципальных услуг, – одна из важных задач развития экономической сферы России. Потребность в выполнении этого запроса не вызывает сомнений: цифровизация открывает широкие возможности для ускорения и упрощения информационного обмена между государственными органами и гражданами. Внедрение в МФЦ data-driven подхода, согласно которому накопленные компанией данные активно используются для оптимизации её работы, позволит улучшить сервис центров и разработать новые решения для повышения удовлетворенности заявителей. Следовательно, разработка и внедрение новых моделей, методов, механизмов управления, позволяющих автоматизировать рабочие процессы МФЦ и оказывающих поддержку при принятии решений – важнейшая задача для экономики современной России. Чтобы ускорить процесс коммуникации между гражданами и государством, повысить эффективность работы МФЦ и степень удовлетворенности граждан качеством получаемых услуг, необходимо всесторонне изучить деятельность и функции МФЦ и разработать математический инструмент оптимизации процессов предоставления услуг. Таким образом, выбранная тема исследования «Математическое моделирование деятельности многофункционального центра (на примере ГАУ «МФЦ Иркутской области»)» является актуальной.

Математический инструментарий, который может использоваться для описания рабочих процессов МФЦ, довольно обширен. Для моделирования деятельности МФЦ могут применяться методы имитационного моделирования: например, в работе А.Г. Жихарева и Д.П. Шарапова [10] разработана имитационная модель деятельности МФЦ по регистрации заявок и оказанию услуг, позволяющая моделировать различные варианты интенсивности поступающих заявок, вычислять среднее время обработки заявок и другие характеристики работы центра. Для расчета оптимального числа сотрудников, составления графика работы многофункционального центра могут применяться также методы и модели систем массового обслуживания. В работе Н.И. Сутягиной [28] построена модель МФЦ как многоканальная система массового обслуживания с ожиданием. А.В. Чуев, С.А. Юдицкий, В.З. Магергут [33] предлагают к рассмотрению один из способов построения графоаналитических моделей систем массового обслуживания – ООО-модель, которая представляет собой графовую модель с тремя уровнями (Обеспечение, Обслуживание, Отчетность). В некоторых исследованиях прибегают к объединению указанных подходов: так, А.Н. Кисляков в работе [13] представил имитационную модель системы массового обслуживания, которая описывает взаимодействие основных показателей деятельности многофункциональных центров с учетом использования механизмов электронного взаимодействия между всеми участниками.

Однако для рассмотренных работ характерен общий недостаток: при разработке модели МФЦ не учитывается динамичность процесса подачи заявлений, которая выражается в варьировании числа заявлений, подаваемых в МФЦ. Также в качестве модели в большинстве работ принимают простейшую многоканальную марковскую модель системы массового обслуживания, тогда как для реальной системы МФЦ характерно ограниченное время ожидания заявителя в очереди, а значит, более корректно использовать модель с «нетерпеливыми заявками».

Несмотря на развитость математического аппарата моделирования систем массового обслуживания, его использование в работе многофункциональных центров сопряжено с определенными трудностями. Такие трудности обусловлены, с одной стороны, огромным потоком клиентов, обращающихся за разными видами услуг, а с другой стороны, постоянным совершенствованием принципов и правовых основ организации деятельности центров.

Наиболее интересным и перспективным представляется направление, связанное с применением методов машинного обучения и анализа данных для моделирования деятельности МФЦ. Методы анализа данных позволяют находить закономерности в посещении гражданами центра, поступлении заявок, классифицировать заявителей. Методы машинного обучения могут использоваться для конструирования рекомендательной системы, что позволит гражданам заранее выяснить, какие услуги им могут понадобиться, и подать заявление на их получение за одно посещение центра. Это позволит снизить загруженность центра, повысить степень удовлетворенности работой МФЦ у населения и значительно сократить затраты. Однако научных работ, посвященных иссле-

дованию применения методов машинного обучения в работе МФЦ крайне мало, что свидетельствует о востребованности исследований в этом направлении.

На основе выше изложенного была сформулирована цель исследования – выработка рекомендаций по оптимизации работы МФЦ на основе методов прикладной математики.

Объектом исследования является Государственное автономное учреждение «Многофункциональный центр предоставления государственных и муниципальных услуг Иркутской области».

Предметом исследования являются математические модели выявления закономерностей в поступлении заявлений и организации процесса обслуживания заявок.

В соответствии с целью исследования поставлены следующие задачи:

1) анализ востребованности услуг МФЦ и выявление закономерностей в транзакциях посетителей центра;

2) моделирование деятельности МФЦ как системы массового обслуживания;

3) разработка рекомендательной системы для прогнозирования следующей услуги заявителя многофункционального центра.

Информационной базой исследования послужила информационная база МФЦ Иркутской области, полученная в результате прохождения конкурса IT-проектов «Цифровой прорыв». Данные представлены за период 06.01.2019 – 17.05.2020 и включают информацию о заявителях, истории подачи заявлений в МФЦ и талонах электронной очереди. Фрагменты таблиц, содержащих указанную информацию, приведены в приложении А.

В качестве инструментальных средств при решении задач исследования и визуализации полученных результатов были использованы языки программирования R и Python. Для оформления работы использовался текстовый редактор «Microsoft Word 2010».

В первой главе рассмотрены особенности функционирования многофункционального центра и приведены методы прикладной математики и машинного обучения, используемые при моделировании его деятельности. Предложен алгоритм моделирования деятельности многофункционального центра

Во второй главе представлена практическая реализация задач исследования. Проведен анализ востребованности услуг и выявление закономерностей в транзакциях заявителей. На основе моделирования деятельности МФЦ как системы массового обслуживания разработан алгоритм нахождения оптимального распределения окон. Разработана рекомендательная система прогнозирования следующей услуги заявителя, в основе которой лежит модель рекуррентной нейронной сети долгой краткосрочной памяти.

В заключении содержатся рекомендации по оптимизации деятельности многофункционального центра №1 г. Иркутска, а также основные результаты практической реализации предлагаемого алгоритма моделирования и оптимизации деятельности МФЦ.



# **1 Теоретические аспекты моделирования деятельности многофункционального центра**

## **1.1 Назначение и особенности функционирования многофункционального центра**

Выявление особенностей многофункциональных центров оказания услуг как объекта моделирования требует рассмотрения основных положений, регламентирующих его деятельность. Для этого обратимся к соответствующим источникам.

Ляхов В.П. [18] дает следующее определение многофункциональному центру предоставления государственных и муниципальных услуг – это «единое место приема, регистрации и выдачи необходимых документов гражданам и юридическим лицам при предоставлении государственных и муниципальных услуг. Деятельность МФЦ направлена на максимальное повышение комфортности граждан, обращающихся за получением государственных и муниципальных услуг». Основным назначением МФЦ является аккумулярование в рамках одного здания услуг органов государственной власти и местного самоуправления, а также ликвидация взаимодействия заявителя с чиновниками и ведомствами.

В МФЦ предоставляются услуги различных ведомств: федеральной миграционной службы (ФМС), управления налоговой службы (ИФНС), государственного кадастра недвижимости (ГКН), Росреестра, а также услуги в сфере жилищных отношений, земельно-имущественных отношений и градостроительства, социальной поддержки населения. Услуги предоставляются независимо от уровня власти, в компетенции которой находится предоставление услуги (федеральная, региональная, муниципальная) [34]. Широкий спектр оказываемых услуг различного типа приводит к неоднородности потока заявок от клиентов многофункциональных центров, что необходимо учитывать при моделировании его деятельности.

Помимо указанной ранее цели функционирования МФЦ, Н.В. Климовских [14] выделяет дополнительную задачу: содействие развитию субъектов малого и среднего предпринимательства и создание в МФЦ условий для привлечения граждан к предпринимательской деятельности в различных сферах. С этой целью происходит регулярное обновление перечня предоставляемых услуг для представителей малого бизнеса. По мнению автора, снижение административных барьеров, повышение качества обслуживания, а также просветительская деятельность МФЦ должны стать фактором, который увеличит количество субъектов малого предпринимательства, а, следовательно, приведет к росту налоговых поступлений от этого сектора экономики в областной бюджет и послужит развитию региона.

Поскольку предпринимательская деятельность обуславливает большое количество документальной работы, особенно важно рассмотреть специфику

создания и функционирования МФЦ именно для этой категории. Так, М.М. Филатова в работе [30] приводит формы функционирования МФЦ в сфере бизнеса:

- бизнес-окно — отдельное окно для приема предпринимателей в действующем центре, оказывающем услуги для всех категорий граждан;
- бизнес-зона — обособленная часть помещения, в которой организовано предоставление услуг предпринимателям;
- бизнес-офис — отдельное помещение или здание, в котором организован прием только представителей бизнеса.

Следовательно, при разработке модели функционирования МФЦ следует учитывать, что заявители, обращающиеся за услугами, относятся к различным типам. Особое положение занимают индивидуальные предприниматели, обслуживание которых носит обособленный характер и должно рассматриваться независимо от обслуживания других заявителей.

Однако для всех типов многофункциональных центров присущи общие черты. Берников М.Ю и Чижмина О.В. в работе [3] выделяют три ключевые характеристики МФЦ. Рассмотрим их:

1) ведение приема осуществляется не представителями органов власти и иных уполномоченных на предоставление услуг организаций, а специально выделенными операционистами, включенными в штат самостоятельной организации — МФЦ. Для оптимизации процедуры предоставления услуг в МФЦ выделяются отдельная категория операционистов, ответственных за согласование, сбор и прием документов, которые в традиционной форме собирались усилиями граждан;

2) организация ведения приема по так принципу «одного окна». Хотя оказание сравнительно простых услуг остается в компетенции отраслевых органов исполнительной власти, многофункциональный центр исполняет все функции, обеспечивающих взаимодействие с ними;

3) организация предоставления большинства услуг населению, проживающему в конкретном муниципальном образовании, на базе относящегося к нему МФЦ.

Работа МФЦ строится по принципу «одного окна», описание которого приводят, например, Филатова А.И. и Чумакова М.Н. в работе [29]. В соответствии с указанным принципом предоставление услуг федерального, регионального, муниципального уровня осуществляется в одном месте. Таким образом, МФЦ выступает в роли организатора процессов предоставления государственных и муниципальных услуг. В соответствии с Федеральным законом от 27 июля 2010 г. № 210-ФЗ «Об организации предоставления государственных и муниципальных услуг» принцип «одного окна» означает предоставление государственной или муниципальной услуги после однократного обращения заявителя с соответствующим запросом, а взаимодействие с органами, предоставляющими государственные или муниципальные услуги, осуществляется многофункциональным центром без участия заявителя в соответствии с норматив-

ными правовыми актами и соглашением о взаимодействии.

Основными целями реализации принципа «одного окна» являются:

- упрощение процедур получения гражданами и юридическими лицами государственных услуг, ведущее к сокращению сроков их предоставления;
- повышение удобства получения гражданами и юридическими лицами государственных услуг;
- сокращение количества документов, требуемых для предоставления заявителями при подаче заявления;
- унификация и автоматизация административных процедур предоставления государственных услуг.

Таким образом, принцип «одного окна» предусматривает исключение участия заявителей в процессах сбора и предоставления в разные инстанции различных документов и справок, подтверждающих сведения о личности, правах и льготах, необходимые для получения заявителем той или иной муниципальной услуги. Кроме того, он подразумевает автоматизацию обработки обращений заявителей и требует непрерывного повышения качества обслуживания заявителей. Для того чтобы функционирование МФЦ было устойчивым и в то же время быстро адаптировалось к меняющимся условиям (изменения в законодательстве; внешние факторы, вызывающие изменение потока заявителей), необходимо разработать гибкую информационную систему поддержки принятия решений, позволяющую быстро реагировать на изменения и модифицировать работу центров в соответствии с ними.

Ниже приведены основные правила организации работы МФЦ, рассмотренные в работе Фирсовой Е.Г. [31].

1. Многофункциональный центр организует предоставление государственных и муниципальных услуг по принципу «одного окна» в соответствии с соглашениями о взаимодействии с федеральными органами исполнительной власти, органами государственных внебюджетных фондов, органами исполнительной власти субъектов Российской Федерации, органами местного самоуправления.

2. Многофункциональный центр действует в соответствии с законодательством Российской Федерации и своим уставом. Многофункциональный центр осуществляет свою деятельность в соответствии с требованиями комфортности и доступности для получателей государственных и муниципальных услуг.

3. В многофункциональном центре обеспечиваются:

- а) функционирование автоматизированной информационной системы многофункционального центра;
- б) возможность оплаты государственных и муниципальных услуг.

4. В многофункциональном центре может быть также организовано предоставление:

- а) услуг, которые являются необходимыми и обязательными для предоставления государственных и муниципальных услуг;

б) услуг, предоставляемых государственными и муниципальными учреждениями и другими организациями, в которых размещается государственное задание (заказ) или муниципальное задание (заказ);

в) дополнительных (сопутствующих) услуг (нотариальные услуги, услуги банка, копировально-множительные услуги, услуги местной, внутризоновой сети связи общего пользования, а также безвозмездные услуги доступа к справочным правовым системам).

5. Для организации взаимодействия с заявителями помещение многофункционального центра делится на следующие функциональные секторы (зоны):

а) сектор информирования и ожидания;

б) сектор приема заявителей.

6. В многофункциональном центре организуется отдельная телефонная линия, предназначенная для ответов на вопросы заинтересованных лиц.

7. При предоставлении государственных и муниципальных услуг в многофункциональных центрах обеспечиваются следующие условия обслуживания заявителей:

а) обращение заявителей в многофункциональный центр осуществляется, в том числе по предварительной записи;

б) время ожидания в очереди для подачи документов и получения результата услуги не превышает 15 минут;

в) прием заявителей в многофункциональном центре осуществляется не менее 5 дней в неделю. График (режим) работы многофункционального центра предусматривает возможность обращения за получением государственных и муниципальных услуг в вечернее время, до 20.00, и не менее чем в один из выходных дней (суббота, воскресенье).

8. Многофункциональный центр использует автоматизированную информационную систему, обеспечивающую:

а) взаимодействие с единой системой межведомственного электронного взаимодействия, региональной системой межведомственного электронного взаимодействия;

б) доступ к электронным сервисам органов, предоставляющих государственные услуги, и органов, предоставляющих муниципальные услуги, в том числе через единую систему межведомственного электронного взаимодействия;

в) интеграцию с региональным порталом государственных и муниципальных услуг;

г) интеграцию с электронной очередью;

д) экспертную поддержку заявителей, работников многофункционального центра и работников центра телефонного обслуживания по вопросам порядка и условий предоставления государственных и муниципальных услуг;

е) поддержку деятельности работников многофункционального центра по приему, выдаче, обработке документов, поэтапную фиксацию хода предоставления государственных и муниципальных услуг с возможностью контроля сроков предоставления государственной или муниципальной услуги и проведе-

ния отдельных административных процедур;

ж) формирование электронных комплектов документов, содержащих заявления (запросы) о предоставлении государственной или муниципальной услуги в форме электронного документа, электронные образы документов, необходимых для оказания государственной или муниципальной услуги;

з) поддержку принятия решений о возможности, составе и порядке формирования межведомственного запроса в иные органы и организации;

и) поддержку формирования комплекта документов для представления в орган, предоставляющий государственную услугу, или в орган, предоставляющий муниципальную услугу, в соответствии с требованиями нормативных правовых актов и соглашений о взаимодействии;

к) хранение сведений об истории обращений заявителей в соответствии с требованиями законодательства Российской Федерации к программно-аппаратному комплексу информационных систем персональных данных;

л) автоматическое распределение нагрузки между работниками многофункционального центра.

10. Обслуживание заявителей в привлекаемой организации осуществляется в соответствии со следующими требованиями:

а) прием заявителей осуществляется не менее 3 дней в неделю и не менее 6 часов в день;

б) максимальный срок ожидания в очереди – 15 минут.

Таким образом, деятельность многофункциональных центров хорошо регламентирована, что позволяет использовать официальные приказы и законодательные акты при моделировании работы МФЦ. Так, например, согласно приведенным выше пунктам распределение нагрузки на работников центров происходит автоматизированно при помощи системы электронной очереди, что позволяет моделировать работу центра как систему массового обслуживания. Характеристики системы, в свою очередь, зависят от графика работы центра, который также утвержден и детерминирован. Наличие базы данных, которая хранит информацию об истории обращений заявителей центра, позволяет разработать информационную систему консультирования заявителей о порядке подачи заявлений.

К деятельности многофункциональных центров относятся следующие виды работ [22]:

1) прием запросов заявителей о предоставлении государственных или муниципальных услуг;

2) информирование заявителей о порядке предоставления государственных и муниципальных услуг в МФЦ, а также консультирование заявителей о порядке предоставления услуг;

3) взаимодействие с государственными органами и органами местного самоуправления по вопросам предоставления государственных и муниципальных услуг, в том числе посредством направления межведомственного запроса с использованием информационно-технологической инфраструктуры;

4) выдача заявителям документов органов, предоставляющих государ-

ственные и муниципальные услуги по результатам предоставления услуг

5) иные функции, установленные нормативными правовыми актами и соглашениями о взаимодействии.

Таким образом, в деятельности МФЦ по оказанию услуг населению и юридическим лицам можно выделить три основные типа услуг: прием документов, выдача результатов и консультирование заявителей. Каждый из них характеризуется своим регламентом и порядком проведения, а потому его следует рассматривать в отдельности от остальных типов услуг.

Согласно работе Малько В.А. [19], порядок предоставления услуг заявителям в МФЦ выглядит следующим образом:

1) заявитель подает обращение в МФЦ, прикладывая необходимые для предоставления услуги документы;

2) операционист принимает документы, формирует полный пакет документов, необходимых для оказания услуги, а также выполняет работы, связанные с оформлением и выдачей заявителю готового итогового документа;

3) в процессе выполнения деятельности по оказанию услуг МФЦ обособленно от заявителя взаимодействует с органами власти, ответственных за оказание услуги;

4) органы власти проверяют и обрабатывают документы, принимают итоговое решение, после чего документы снова направляются в МФЦ. При повторном обращении заявитель получает итоговый документ или обоснованный отказ.

Следовательно, для заявителя большая часть времени при обслуживании в МФЦ проходит при ожидании выполнения услуги. Оптимизация этой части оказания услуг находится в зоне ответственности государственных органов, оказывающих эту услуги. Однако не менее важным с точки зрения удовлетворенности сервисом МФЦ является время, проводимое заявителем непосредственно в офисе центра, которое складывается из времени ожидания в очереди и времени обслуживания операционистом. Поэтому разработка модели работы офиса МФЦ, с помощью которой будет осуществлена оптимизация первичного обслуживания заявителя непосредственно в центре, – это задача, имеющая важное прикладное значение.

Большая часть научных работ посвящена исследованию правовых вопросов при организации работы многофункционального центра. Например, в работе Фурина А.Г. [32] развитие многофункциональных центров рассматривается как процесс формирования нового института, вследствие чего автор определяет основы его создания и функционирования. Целью работы является доказательство гипотезы о том, что МФЦ – институт государственного сектора, оптимизирующий деятельность субъектов экономики, посредством эффективного предоставления государственных и муниципальных услуг. Автор приходит к выводу, что МФЦ является институтом, обеспечивающим положительный эффект в процессе предоставления государственных и муниципальных услуг всем заинтересованным агентам.

Исупова И.Н. [12] приводит обзор различных моделей организации МФЦ.

Модель первого шага действует как справочная служба и не предполагает большого количества предоставляемых услуг. Модель магазина повседневного спроса объединяет простые электронные услуги отдельных ведомств, не предполагая сложной интеграции (в основном вертикальная интеграция и почти полное отсутствие горизонтальной). Истинная модель принципа «одного окна» – полностью интегрированная модель: объединение простых и комплексных услуг, значительный реинжиниринг внутренних процессов, полная интеграция (как вертикальная, так и горизонтальная). Автор приходит к выводу, что принятие решения о создании МФЦ в том или ином регионе России, а также о выборе модели организации, должно основываться на детальном анализе всех сторон данного проекта, причем должно быть сравнение между модернизацией существующих институтов и созданием МФЦ в каждом конкретном регионе отдельно. Для построения модели МФЦ, адекватно отражающей реальные рабочие процессы оказания государственных и муниципальных услуг, следует ориентироваться на более сложную модель «одного окна».

Заполева Е.О., Захарченко А.А., Пить В.В., Мудрик Е.С. в работе [11] рассматривают вопросы повышения качества оказания и доступности государственных и муниципальных услуг. Активное развитие телекоммуникационных технологий позволяют значительно расширить систему предоставления государственных услуг населению и сократить время, которое тратится на их получение. В статье приводится пример мониторинга оценки удовлетворенности населением существующими формами взаимодействия с государственными органами. В исследовании оценивались такие показатели работы службы, как график работы МФЦ; территориальная доступность МФЦ; размер, оснащенность, интерьер помещения МФЦ; организация очереди; уровень обслуживания со стороны специалистов и др.

Мироненко Н.В., Тихенькая Н.С. и Кондакова К.С. [20] рассматривают проблемы функционирования многофункциональных центров услуг и анализируют итоги создания таких центров как фактора формирования конкурентной среды и повышения качества предоставления государственных услуг. По мнению авторов, одним из направлений развития МФЦ может являться внедрение модели участия органов власти в деятельности МФЦ: МФЦ как представитель интересов заявителя. Такая модель предполагает представление интересов заявителей в отношениях с органами государственной власти на всех уровнях, услуги которых осуществляются в МФЦ. Такие отношения носят характер представительства, где правовыми основаниями служат закон, договор поручения, доверенность. Для реализации этой модели на федеральном, региональном и муниципальном уровнях должны быть приняты соответствующие правовые акты. Вместе с тем использование принципа представительства требует одновременного принятия соответствующих правовых актов на всех уровнях власти.

Вопрос перспектив развития системы многофункциональных центров в Российской Федерации рассматривают Берников М.Ю. и Чижмина О.В. [3]. Авторы утверждают, что создание современных и удобных для граждан форматов предоставления услуг – важнейшая задача при построении сервисного государ-

ства. К приоритетным задачам можно отнести и разработку ряда нормативно-правовых актов для оптимизации правовой и структурной базы работы МФЦ, утверждение единого перечня государственных услуг.

Подводя итог, можно сказать, что дальнейшее развитие системы многофункциональных центров в субъектах РФ позволит создать благоприятную основу для эффективного взаимодействия государства и граждан. Однако для оптимизации всей системы необходимо применять комплексный подход, включающий в себя не только правовое регулирование, учет технических и технологических аспектов работы МФЦ, но и современный математический инструментарий функционирования системы обслуживания.

## **1.2 Обзор математических методов моделирования деятельности многофункционального центра**

С точки зрения поставленных задач исследования наибольший интерес представляет анализ работ, посвященных применению математического инструментария для моделирования процессов, протекающих в МФЦ. Многофункциональный центр предоставления государственных и муниципальных услуг – это сложная система многочисленных и взаимосвязанных между собой ведомств, подразделений и отделов, которая должна обеспечивать слаженную и ритмичную работу всех своих подчастей. Широкий спектр видов деятельности и функций МФЦ обуславливает разнообразие решаемых задач и используемых для этого математических методов и моделей.

Многие работы посвящены моделированию деятельности МФЦ в виде бизнес-процессов при помощи различных нотаций. Такой подход позволяет деконструировать функции МФЦ, найти «узкие места» и причины неэффективности в бизнес-процессе, провести имитационное моделирование и функционально-стоимостный анализ бизнес-процесса с помощью программных продуктов бизнес-моделирования. Ковалев С.С. в статье [15] предлагает описание регламента предоставления услуг МФЦ с точки зрения процессного подхода. Проведенное автором моделирование процесса предоставления услуги гражданам позволило выявить возможные способы его совершенствования, в частности, на стадии подачи документов, а также возможные риски. Очевидным преимуществом процессного подхода является возможность формулирования процедуры получения услуги в виде последовательности действий, понятной для пользователей МФЦ.

Такого же подхода к изучению деятельности МФЦ придерживались Э.А. Михайлова и С.Н. Кошачова [21]. В работе были рассмотрены такие бизнес-процессы, как общая организация функционирования МФЦ, организация процесса предоставления услуг, оценка качества и эффективности работы МФЦ. Авторы предложили следующий алгоритм исследования: на первом этапе для выделенных бизнес-процессов были построены композиционные диа-



граммы «Как есть»; на втором этапе, проанализировав существующие диаграммы, авторы предложили свои коррективы, которые отразили в диаграммах «Как будет». Например, по мнению исследователей, процесс оценки качества предоставления услуг и эффективности работы МФЦ требует изменений, поэтому было предложено создание специальной комиссии, уполномоченной контролировать соблюдение требований к организации предоставления услуг и взаимодействию МФЦ как с заявителями, так и с исполнителями. Таким образом, процессный подход к описанию деятельности многофункциональных центров применяется достаточно часто, однако он требует участия специалистов, глубоко понимающих специфику предметной области, а потому не является универсальным.

Оптимизация работы многофункционального центра и расчет основных характеристик его работы может достигаться за счет применения методов теории массового обслуживания. Применение аппарата теории массового обслуживания позволяет найти оптимальное соотношение между числом обслуживающих единиц и качеством обслуживания: увеличение числа обслуживающих единиц положительно сказывается на качестве обслуживания, но иметь излишние обслуживающие единицы экономически невыгодно.

Анализ работы СМО существенно упрощается, если она может быть описана как марковский процесс. Случайный процесс является марковским, если для любого момента времени вероятностные характеристики процесса в текущий момент времени  $t$  зависят только от его состояния в предшествующий момент  $t-1$  и не зависят от более ранних состояний. Общие правила и принципы построения модели СМО, описываемой марковским процессом, приводятся, например, в работе Будылиной Е.А. и Гарькиной И.А. [4]. Авторы приводят основные выводы из теории систем массового обслуживания: для моделирования работы устройства, состоящего из двух узлов, каждый из которых может выйти из строя в случайный момент времени, приведен вывод системы дифференциальных уравнений Колмогорова для вероятностей состояний системы. Также описана система уравнений, решение которой задает предельные вероятности состояния системы. Отметим, что в статье не приводятся обобщения модели на случай  $n$ -канальной системы обслуживания, а рассмотрен только частный случай двухканальной СМО.

Приложение теории системы массового обслуживания к описанию функционирования многофункционального центра на примере МФЦ Нижегородской области продемонстрировано в статье Н.И. Сутягиной [28]. Автор использует простейшую  $n$ -канальную модель теории массового обслуживания, согласно которой на обслуживание поступают идентичные заявки, очередь неограниченна, поступление заявок и время обслуживания определяются распределением вероятности, а вероятность появления двух или более заявок за малый интервал времени столь мала, что ею можно пренебречь. При сделанных предположениях число входящих заявок распределено по закону Пуассона, а время ожидания в очереди начала обслуживания считают распределенным экспоненциально. Таким образом, моделирование деятельности служащих центра в виде

системы массового обслуживания позволило решить задачу определения оптимального интервала назначений на прием. Помимо этого автору статьи удалось найти оптимальное решение проблемы выбора числа специалистов центра с использованием экспертного метода попарных сравнений альтернатив. Однако стоит отметить существенный недостаток проведенного исследования: указанные предположения о законах распределения входного потока и времени обслуживания не проверяются при помощи методов статистического анализа, а значит, полученные автором результаты могут не обладать высокой степенью достоверности.

Многие авторы для моделирования СМО применяют метод имитационного моделирования, поскольку он позволяет моделировать системы различной сложности. Сущность такого подхода состоит в том, что на основании описания принципов функционирования системы и численных методов разрабатывается моделирующий алгоритм, имитирующий поведение системы и ее элементов, внешние воздействия на систему, взаимодействия элементов и последовательное изменение состояний всей системы во времени [9].

А.Г. Жихарев и Д.П. Шарапов в статье [10] на примере ГАУ «МФЦ Белгородской области» продемонстрировали применение программного инструментария имитационного моделирования процессов и систем многофункциональных центров. Сбор данных осуществлялся один месяц. Проведенные эксперименты с применением разработанной авторами модели показали, что модель является адекватной и позволяет моделировать различные варианты интенсивности поступающих заявок, вычислять среднее время обработки заявок и среднее число заявок, поступающих в различные дни недели.

А.Н. Кисляков в работе [13] также осуществил построение имитационной модели деятельности МФЦ с использованием теории систем массового обслуживания. Проанализировав лучшие практики оценки качества и доступности предоставления государственных услуг, автор сформировал ряд показателей, которые могут использоваться при оценке работы многофункционального центра. К ним были отнесены общая эффективность работы МФЦ, экономическая эффективность работы МФЦ, эффективность использования механизмов электронного взаимодействия, доход от предоставления платных услуг, среднее время ожидания заявителя в очереди, количество жалоб и др. Определив основные показатели, автор проводит построение графа взаимодействия между основными показателями работы МФЦ. На основании построенной модели и индикаторной сети можно построить временной ряд, позволяющий отследить динамику работы МФЦ и качество оказания услуг. Предложенный А.Н. Кисляковым подход позволяет оценить деятельность МФЦ не только с позиции экономической эффективности и социальной значимости, но и с позиции использования современных технологий электронного правительства, а также оценить эффективность информационного взаимодействия между участниками МФЦ: единым порталом государственных и муниципальных услуг, системой межведомственного электронного взаимодействия, органами местного самоуправления.

Группа авторов А.В. Чуев, С.А. Юдицкий, В.З. Магергут в работе [33] рассмотрели работу многофункционального центра предоставления государственных и муниципальных услуг как систему массового обслуживания. В качестве способа построения графоаналитических моделей систем массового обслуживания выбрана ООО-модель МФЦ, состоящая из трех уровней – обслуживание, обеспечение, отчетность. Каждый уровень представляет собой индикаторную сеть, состоящую из одного или нескольких двудольных графов. На уровне обеспечения была сформулирована обобщенная модель работы специалиста с заявителем авторами: в частности, было введено множество циклических графов работы специалистов, поскольку государственные и муниципальные услуги могут предоставлять разными организациями, работающими параллельно в разных окнах приема. Уровень обслуживания представляет собой множество циклических графов взаимодействия с посетителями на этапе выбора услуг, которое задает порядок действий заявителя при записи на прием и обслуживании с помощью системы электронной очереди. На уровне отчетности модели рассчитываются основные показатели работы системы в отчетном периоде. На основе полученной модели строится лента поведения, по которой можно отследить динамику работы МФЦ и качества оказания услуг, а также могут проводиться имитационные эксперименты. Результатом имитационного моделирования также являются временные графики ключевых показателей, которые учитываются при принятии решения.

Е.Г. Гаврилова, В.С. Панченко, М.П. Маркова в работе [8] разработали две имитационные модели многофункционального центра как системы массового обслуживания: одна из моделей построена на экспериментальных данных, полученных при исследовании работы действующей системы МФЦ, другая – на основе моделируемых параметров обслуживания клиентов МФЦ. С целью устранения причин возникновения очередей и уменьшения времени ожидания клиентов МФЦ была проведена структурная оптимизация изучаемой СМО, по итогам которой было выявлено, что требуется увеличить число окон для работы с клиентами и четко определить временной регламент обслуживания клиентов. Для решения задачи авторы использовали пешеходную библиотеку AnyLogic. Недостатком исследования является отсутствие подробного исследования характеристик построенной системы массового обслуживания; кроме того, авторы не учитывают, что на оказание различных услуг тратится различное время и задействовано различное окон обслуживания, вследствие чего их модель подлежит дальнейшему уточнению.

Для рассмотренных выше работ, посвященных моделированию деятельности МФЦ как системы массового обслуживания, характерен общий недостаток: не учитывается динамичность процесса подачи заявлений. Авторы берут данные за определенный промежуток времени и на их основе рассчитывают характеристики центра, не делая предположений о том, что число клиентов в МФЦ, может существенно варьироваться день ото дня. Помимо этого наиболее часто в качестве модели СМО принимают простейшую марковскую модель с многими каналами, не принимая во внимание тот факт, что для реальной си-

системы МФЦ характерно ограниченное время ожидания заявителя в очереди, а значит, разумнее использовать СМО с «нетерпеливыми заявками».

История обращений заявителя в МФЦ представляет собой набор транзакций. Поэтому для поиска в них полезных закономерностей можно использовать такие методы машинного обучения, как поиск ассоциативных правил или последовательных шаблонов. В научных работах решение этой задачи именно в рамках системы многофункционального центра не описано, однако можно воспользоваться результатами, полученными в других областях.

Наиболее известными алгоритмами поиска ассоциативных правил являются Apriori и FP-Growth (Frequent Pattern Growth). Алгоритм Apriori – базовый алгоритм для определения частотных правил, предложенный в работе R. Agrawal и R. Srikant [35]. Поиск ассоциативных правил выполняется в два этапа:

- 1) в пуле имеющихся признаков находят наиболее часто встречающиеся комбинации элементов;

- 2) из найденных на первом этапе комбинаций наиболее часто встречающихся наборов формируют ассоциативные правила.

FP-Growth алгоритм является модификацией алгоритма Apriori. Его подробное описание приводят Пальмов С.В. и Французова Е.Н. в работе [25]. Главное отличие FP-Growth алгоритма – предварительная обработка исходного набора данных, результатом которой является достаточно компактная древовидная структура – Frequent Pattern Tree (дерево популярных предметных наборов, FP-дерево).

Процесс работы алгоритма FP-Growth имеет следующий вид:

- 1) построение FP-дерева, представляющего информацию о часто встречающихся предметных наборах;

- 2) извлечение из FP-дерева часто встречающихся наборов элементов.

Сравнительные исследования классического алгоритма Apriori и FP-Growth показали, что с увеличением размеров исходного набора данных, временные затраты на поиск часто встречающихся наборов элементов растут для FP-Growth намного медленнее, чем для Apriori.

В работе В. Ayvaz [37] описывается применение разновидностей алгоритма поиска ассоциативных правил Apriori и FP-Growth для определения групп товаров, которые зачастую продаются вместе. Информационной базой исследования выступает база данных компании, занимающейся продажей аппаратного обеспечения. В результате сравнения работы двух алгоритмов, алгоритм FP-Growth показал более высокую скорость работы, затратив в 71 раз меньше времени, чем Apriori-алгоритм. Однако Apriori-алгоритм сумел выявить на 14 ассоциативных правил больше, причем их уровень достоверности был достаточно высок. Ценность полученных результатов заключается в том, что ранее компания не знала о существовании таких закономерностей, а значит, недополучала прибыль. На практике полученные результаты могут учитываться при проектировании сайта интернет-магазина и проведении маркетинговых акций.

Применение метода ассоциативных правил для анализа транзакций имеет существенные недостатки: метод учитывает лишь факт совместного появления товаров, игнорируя временной аспект, и не является клиентоориентированным, поэтому не позволяет разрабатывать персональные рекомендации. Преодолеть эти ограничения позволяет метод последовательных шаблонов, который по своей природе является расширением теории ассоциативных правил. Подробное описание существующих алгоритмов поиска последовательных шаблонов приводится в работе С. Моонеу [46]. Исследователями были разработаны целые семейства алгоритмов, которые применяются для конкретных наборов данных. Однако наиболее популярным алгоритмом поиска последовательных шаблонов для транзакционных баз данных является алгоритм SPADE. Разработчиком алгоритма является М. Закі [51]. SPADE имеет преимущество перед другими алгоритмами в виде уменьшения времени работы и требуемой памяти за счет оптимизации работы поиска шаблонов. SPADE использует комбинаторные свойства для декомпозиции исходной задачи на более мелкие подзадачи, которые могут быть решены независимо с использованием эффективных методов поиска решений на сетке и простых операций соединения. При этом все последовательные шаблоны могут быть обнаружены только при трех сканированиях базы данных

Приложение метода последовательных шаблонов для анализа банковских транзакций описывается в работе Н.Б. Паклина и С.В. Уланова [24]. Для базы данных транзакций задача поиска последовательных шаблонов заключается в обнаружении максимальных последовательностей среди всех последовательностей, имеющих поддержку выше заданного порога. Каждая такая максимальная последовательность и представляет собой последовательный шаблон. Такие шаблоны обладают более высокой прогнозной силой, чем ассоциативные правила, и потому внедрение подобного функционала позволило бы повысить эффективность работы банка за счет повышения лояльности клиентов. Так как база данных МФЦ имеет структуру, аналогичную структуре базы данных банковских транзакций, метод последовательных шаблонов может быть применим и для клиентов многофункционального центра.

Разработка рекомендательной системы, составляющей прогноз о следующей услуге, за которой обратится заявитель, позволила бы ускорить процесс взаимодействия между клиентом и многофункциональным центром и повысить удовлетворенность пользователей системы МФЦ. Поскольку история заказов услуг клиентов МФЦ представляет собой последовательность транзакций, упорядоченных во времени, для составления прогноза можно воспользоваться методами прогнозирования последовательностей (sequence prediction). Стоит отметить, что последовательность транзакций не является числовой, а значит, классические методы прогнозирования временных рядов не могут использоваться для решения задачи прогнозирования услуги, за которой обратится клиент, на основе его прошлых обращений.

В качестве подходов для решения задачи прогнозирования следующей услуги можно использовать нейросетевые модели. Например, в работе

С. Shibata и J. Heinz [50] для прогнозирования элементов последовательности предлагается использовать модель рекуррентной нейронной сети с долгой краткосрочной памятью. Такая архитектура рекуррентных нейронных сетей позволяет выявлять зависимости между элементами последовательности, разделенными большими промежутками времени. В качестве базовой модели авторы предлагают двухслойную модель с нелинейной функцией активации softmax на выходном слое. Между выходным слоем и модулями долгой краткосрочной памяти располагается полностью связанный слой с функцией активации ReLU. Выходом нейронной сети является предсказанный элемент последовательности. Базовая модель показала хорошие результаты в предсказании конечного элемента последовательности на пятнадцати различных датасетах.

Y. Zhao и др. [53] также используют модели искусственных нейронных сетей для решения описанной выше задачи. При этом они определяют задачу прогнозирования элементов последовательности как задачу классификации, в которой начальные элементы последовательности – это признаки для классификации, а ее конечный элемент – метка класса. Авторы сравнили между собой работу трех моделей: многослойный персептрон, сверточная нейронная сеть, рекуррентная нейронная сеть с долгой краткосрочной памятью.

Многослойный персептрон представляет собой нейронную сеть прямого распространения, состоящую из нескольких слоев, нейроны которых связаны с каждым нейроном на предыдущем и следующем слое. На скрытых слоях используется нелинейная функция активации, на выходном слое используется функция softmax. Функция ошибок представляет собой кросс-энтропийную функцию. Сверточная нейронная сеть также является сетью прямого распространения с нелинейной функцией активации на слоях свертки. Она также может включать субдискретизирующие слои, которые располагаются после слоев свертки. Рекуррентная нейронная сеть с долгой краткосрочной памятью была разработана для решения проблемы исчезающего градиента, что подробно описывается в работе Пустынного Я.Н. [26]. Такой тип архитектуры нейронной сети показывает хорошие результаты при моделировании долгосрочных зависимостей.

Обучив описанные модели нейронных сетей, авторы сравнили их работу при прогнозировании последовательностей на нескольких тестовых выборках. Наилучший результат показала модель многослойного персептрона: авторы объяснили это тем, что многослойный персептрон более точно учитывает порядок элементов последовательности во входных данных.

I. Sato и др. в работе [20] провели сравнительный анализ шести моделей, которые могут использоваться для прогнозирования элементов последовательности: языковая модель n-gram; комбинация модели n-gram со спектральной языковой моделью; рекуррентная нейронная сеть; алгоритм XGBoost; комбинация рекуррентной нейронной сети и XGBoost; гибридная модель нейронной сети и модели n-gram. Языковая модель n-gram предсказывает элемент последовательности на основе вычисления относительной частоты появления подпоследовательности с прогнозным элементом длины  $n$  в тренировочной выборке.

Алгоритм XGBoost – это метод градиентного бустинга, в котором базовым классификатором является дерево принятия решений. Ряд вычислительных экспериментов показал, что наиболее высокую точность для решения задачи прогнозирования элемента последовательности обеспечила комбинация модели n-gram со спектральной языковой моделью.

R. Begleiter и др. [38] для прогнозирования элементов дискретных последовательностей с конечным набором элементов использовали подход, в основе которого лежат марковские модели переменного порядка. Для сравнения между собой авторы выбрали такие модели, как модель взвешивания контекстного дерева, предсказание по частичному совпадению и вероятностные суффиксные деревья. Марковские модели переменного порядка обладают важным преимуществом по сравнению с моделями марковских цепей фиксированного порядка, в которых каждая переменная в последовательности зависит от фиксированного числа предыдущих отсчетов, за счет большей гибкости и адаптивности для решения конкретной задачи.

Хорошие результаты при прогнозировании элементов последовательности показывает алгоритм компактного дерева предсказаний (CPT, Compact Prediction Tree). В методе используются три структуры данных: дерево предсказания, обратный индекс и массив указателей. Пустое дерево предсказаний состоит из одного узла – корня. В процессе обучения модели дерево заполняется по следующему алгоритму: для очередной последовательности проверяется наличие ее первого элемента среди потомков корня; если таковой потомок имеется, то переходят к поиску следующего элемента последовательности в потомке; иначе, вставляют оставшиеся элементы последовательности ветвью, начиная с последнего узла, для которого нашлись совпадения или корня, если совпадений не было вообще. Обратный индекс является двумерной двоичной матрицей, строки которой соответствуют уникальным элементам последовательностей, содержащихся в дереве предсказания. Массив указателей содержит указатели на последние элементы последовательностей в РТ доступные по номерам последовательностей [2]. Применение метода CPT приводится в работе T. Gueniche и P. Fournier-Viger [41].

Подводя итог, можем сказать, что для решения задачи оптимизации работы МФЦ могут применяться различные математические методы и модели, которые включают в себя имитационные модели [8, 9, 10, 33], методы систем массового обслуживания [4, 13, 28], моделирование бизнес-процессов [15, 21], методы машинного обучения. Среди методов машинного обучения выделяются ассоциативные правила и последовательные шаблоны, которые позволяют находить закономерности в транзакциях клиентов [24, 25, 35, 37], а также методы прогнозирования элементов последовательностей [38, 41, 53], которые можно применить для разработки рекомендательной системы, позволяющей сделать прогноз о следующей услуге заявителя. На настоящий момент очень малое число работ посвящено вопросу внедрения методов машинного обучения в работу МФЦ, поэтому это направление исследования является актуальным и перспективным.

### 1.3 Концептуальная схема анализа и моделирования деятельности многофункционального центра

На основании изложенных в параграфах 1.1 и 1.2 теоретических сведений разработана концептуальная схема анализа и моделирования деятельности многофункционального центра, состоящая из трех этапов.

1. Анализ востребованности услуг МФЦ и выявление закономерностей в обращениях заявителей.

На данном этапе осуществляется предварительный анализ исходных данных, включающий в себя решение следующих задач:

- определение наиболее востребованных услуг многофункционального центра для каждого типа заявителя. Этот этап реализуется при помощи написания запросов к базе данных многофункционального центра;

- изучение динамики общего числа поданных заявлений за рассматриваемый период времени и проверка гипотезы о наличии влияния дня недели на число поданных заявлений. Проверка гипотезы осуществляется при помощи параметрического или непараметрического однофакторного дисперсионного анализа, в зависимости от результатов проверки гипотезы о нормальном законе распределения результативного признака. Полученные результаты могут быть использованы при составлении графика работы сотрудников многофункционального центра;

- выявление закономерностей в обращениях за услугами клиентов МФЦ. Этот этап включает в себя реализацию алгоритма поиска ассоциативных правил Apriori [35] и реализацию алгоритма поиска последовательных шаблонов SPADE [51].

Алгоритм Apriori относится к алгоритмам обучения без учителя. На множестве объектов  $X$  задано  $k$  бинарных признаков  $\mathcal{F} = \{f_1, \dots, f_k\}$ ,  $f_j: X \rightarrow [0,1]$  и имеется выборка объектов  $X^n = \{x_1, \dots, x_n\} \subset X$ . Каждому набору признаков  $\varphi \subseteq \mathcal{F}$  в соответствие ставится предикат  $\varphi(x)$ , равный конъюнкции всех признаков, включенных в  $\varphi$ :

$$\varphi(x) = \bigwedge_{f \in \varphi} f(x), x \in X. \quad (1.3.1)$$

Если  $\varphi(x) = 1$ , то это означает, что признаки набора  $\varphi$  совместно встречаются у объекта  $x$ . Поддержкой (support) набора  $\varphi$  в выборке  $X^n$  называется функция:

$$\text{supp}(\varphi) = \frac{1}{n} \sum_{i=1}^n \varphi(x_i). \quad (1.3.2)$$

Поддержка показывает, у какой доли объектов из обучающей выборки



встречается набор  $\varphi$ . Набор  $\varphi$  называется частым, если его поддержка превышает некоторый заданный пользователем порог:  $\text{supp}(\varphi) \geq \text{MinSupp}$ .

Используя введенные обозначения, можем дать следующее определение: ассоциативным правилом вида  $\varphi \rightarrow y$  называется пара непересекающихся наборов  $\varphi, y \subseteq \mathcal{F}$ , если выполнены два условия:

1. Совместная пара наборов  $\varphi, y$  является частой:  $\text{supp}(\varphi \cup y) \geq \text{MinSupp}$ ;
2. Если в выборке встречается набор  $\varphi$ , то вместе с ним часто встречается и набор  $y$ :

$$\text{conf}(\varphi, y) = \text{supp}(y|\varphi) = \frac{\text{supp}(\varphi \cup y)}{\text{supp}(\varphi)} \geq \text{MinConf}. \quad (1.3.3)$$

Показатель  $\text{conf}(\varphi, y)$  называется достоверностью (confidence) правила; он представляет собой условную вероятность того, что если в наборе встречается набор  $\varphi$ , то в этом же наборе будет присутствовать набор  $y$ .

Используются и другие показатели для оценки обнаруженных правил. Улучшение (lift) – показатель, позволяющий оценить, полезнее ли найденное правило случайного угадывания:

$$\text{lift}(\varphi, y) = \frac{\text{supp}(\varphi \cup y)}{\text{supp}(\varphi) \cdot \text{supp}(y)}. \quad (1.3.4)$$

Если значение улучшения больше единицы, то с помощью правила предсказать наличие элемента  $Y$  в наборе, содержащем  $X$ , надежнее, чем случайное угадывание.

Алгоритм Apriori позволяет значительно сократить время поиска ассоциативных правил по сравнению с полным перебором, поскольку в его основе лежит свойство антимонотонности: поддержка любого (множества) набора элемента не может превышать минимальной поддержки любого из его подмножеств [1]. Формально это свойство записывается следующим образом:

$$\forall \varphi, \phi \in \mathcal{F}: \varphi \subset \phi, \text{supp}(\varphi) \leq \text{supp}(\phi). \quad (1.3.5)$$

Свойство позволяет сделать следующий вывод: для того чтобы набор  $\phi$  являлся частым необходимо, чтобы любое его подмножество  $\varphi \subset \phi$  было частым.

Алгоритм Apriori основан на поиске в ширину и имеет следующий вид.

1 этап: выделение всех часто встречающихся наборов:

1.1. выделение всех частых исходных признаков (одноэлементных наборов):

$$G_1 = \{f \in \mathcal{F}: \text{supp}(f) \geq \text{MinSupp}\}. \quad (1.3.6)$$

1.2. выделение частых наборов мощности  $j$ : для всех  $j = \overline{2, k}$

$$G_j = \{\varphi \cup \{f\} \mid \varphi \in G_{j-1}, f \in G_1, \text{supp}(\varphi \cup \{f\}) \geq \text{MinSupp}\}. \quad (1.3.7)$$

Если  $G_j = \emptyset$  (новые частые наборы не генерируются), то выход из цикла по  $j$ .

2 этап: выделение ассоциативных правил:

2.1.  $R = \emptyset$  (исходное множество ассоциативных правил пусто).

2.2. для  $\forall \varphi \in G_j, j = \overline{2, k}$ : выполнение процедуры  $\text{AssocRules}(\varphi, \emptyset)$ .

Опишем процедуру  $\text{AssocRules}(\varphi, y)$ :

для  $\forall f \in \varphi$ :

$\varphi' = \varphi \setminus \{f\}, y' = y \cup \{f\}$ ;

если  $\text{conf}(\varphi', y') \geq \text{MinConf}$ , то  $(\varphi', y') \in R$ ;

если  $|\varphi'| > 1$ , то  $\text{AssocRules}(\varphi', y')$ .

В качестве ориентира при практической реализации поиска ассоциативных правил можно воспользоваться работой В. Ауваз [37], в которой постановку задачи можно изменить в соответствии с целью определения закономерностей в обращениях заявителей МФЦ.

Алгоритмы поиска последовательных шаблонов позволяют учитывать порядок появления элементов в наборе, то есть учитывать временной фактор.

Введем необходимые обозначения. Пусть  $\Sigma$  – конечное множество элементов, называемое алфавитом, а  $|\Sigma|$  – мощность алфавита (число символов, входящих в него). Последовательностью  $s$  называют упорядоченный набор элементов:  $s = (s_1, \dots, s_k)$ , при этом префиксом последовательности называется подстрока  $s[1:j] = (s_1, s_2, \dots, s_j)$ , а суффиксом – подстрока  $s[j:k] = (s_j, s_{j+1}, \dots, s_k)$ .

Последовательность  $r = (r_1, \dots, r_m)$  называется подпоследовательностью последовательности  $s = (s_1, \dots, s_k)$ , если существует взаимно однозначное отображение  $\phi: [1, m] \rightarrow [1, k]$  такое, что  $r[i] = s[\phi(i)]$  и  $\forall i, j$  (индексы элементов в последовательности  $r$ )  $i < j \Rightarrow \phi(i) < \phi(j)$ . Таким образом, каждый индекс элемента в  $r$  соответствует определенному индексу элемента в  $s$ , причем порядок элементов сохраняется.

Аналогично определению из области поиска ассоциативных правил вводится понятие поддержки последовательности  $r$ . Пусть имеется исходное множество последовательностей  $D = \{s_1, s_2, \dots, s_N\}$ , тогда поддержкой последовательности  $r$  является величина

$$\text{supp}(r) = \frac{|\{s_i \in D \mid r \subseteq s_i\}|}{N}. \quad (1.3.8)$$

Так же как и ранее, последовательность считается частой, если ее поддержка превышает заранее заданный порог  $\text{MinSupp}$ .

Алгоритм SPADE использует вертикальный формат базы данных. Его идея заключается в том, чтобы для каждого элемента записать идентификатор последовательности, в которую он входит, и его позицию (индекс) в этой по-

следовательности. Таким образом, для каждого элемента алфавита  $s \in \Sigma$  сохраняется множество наборов вида  $(i, \text{pos}(s))$ , обозначаемое  $\mathcal{L}(s)$ , где  $\text{pos}(s)$  – множество позиций, на которых элемент  $s$  располагается в последовательности  $s_i \in D$ .

Алгоритм SPADE приведен в [52] и имеет следующий вид.

1. Задать  $\mathcal{F} = \emptyset$ ,  $k = 0$ ,  $P = \{(s, \mathcal{L}(s)) \mid s \in \Sigma, \text{supp}(s) \geq \text{MinSupp}\}$ .

2. Выполнить процедуру SPADE( $P, \text{MinSupp}, \mathcal{F}, k$ ):

для  $\forall r_a \in P$ :

$$\mathcal{F} = \mathcal{F} \cup \{(r_a, \text{supp}(r_a))\},$$

$$P_a = \emptyset,$$

для  $\forall r_b \in P$ :

$$r_{ab} = r_a + r_b[k],$$

$$\mathcal{L}(r_{ab}) = \mathcal{L}(r_a) \cap \mathcal{L}(r_b),$$

если  $\text{supp}(r_{ab}) \geq \text{MinSupp}$ , то

$$P_a = P_a \cup \{r_{ab}, \mathcal{L}(r_{ab})\}.$$

если  $P_a \neq \emptyset$ , то SPADE( $P_a, \text{MinSupp}, \mathcal{F}, k + 1$ ).

При практической реализации алгоритма SPADE можно ориентироваться на работу Н.Б. Паклина и С.В. Уланова [24], в которой демонстрируется применение поиска последовательных шаблонов для анализа транзакций банковских клиентов.

2. Моделирование деятельности МФЦ как системы массового обслуживания.

На первом этапе необходимо определить основные параметры системы массового обслуживания (число окон обслуживания, тип очереди, наличие ограничений на время ожидания заявки), а также тип системы массового обслуживания. Если время между поступлением последовательных заявок и время обслуживания заявки распределено по экспоненциальному закону распределения, то СМО является марковской. На этапе проверки закона распределения рассчитываются также оценки интенсивности входного потока и интенсивности обслуживания.

Поскольку для многофункциональных центров характерна ситуация, когда клиент может уйти до начала обслуживания из-за длительного времени ожидания в очереди, то при моделировании системы массового обслуживания следует накладывать ограничение на время ожидания. Такая система массового обслуживания носит название СМО с «нетерпеливыми» заявками. Особенности такой системы массового обслуживания и подходы к ее моделированию описываются, например, в работах Осипова Г.С. [23] и Вентцеля Е.С. [6].

Пусть имеется  $n$ -канальная СМО с ожиданием. Очередь будем считать ограниченной по длине  $m$ , поскольку физическая вместимость офиса МФЦ ограничена. Время пребывания в очереди ограничено некоторым случайным сроком  $T_{оч}$  с математическим ожиданием  $t_{оч}$ . На каждую заявку, ожидающую в очереди, действует «поток уходов» с интенсивностью  $\nu = (t_{оч})^{-1}$ . Если этот поток пуассоновский, то процесс, протекающий в СМО, будет марковским про-

цессом гибели-размножения. Его граф состояний будет иметь вид, представленный на рисунке 1, где  $\lambda$  – интенсивность поступления заявок,  $\mu$  – интенсивность обслуживания.

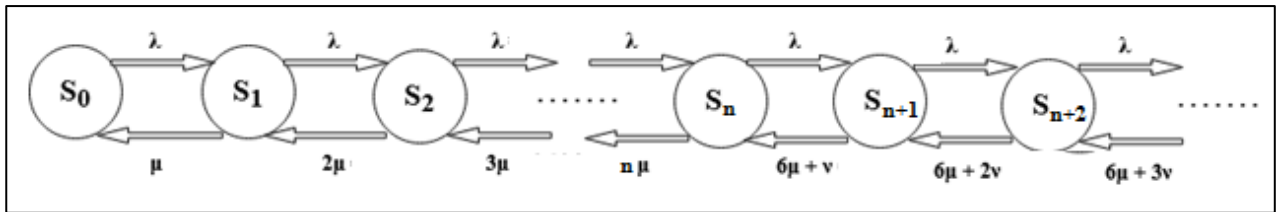


Рисунок 1 – Граф состояний системы массового обслуживания с «нетерпеливыми заявками»

Для такой СМО известны формулы расчета вероятностей состояний системы в стационарном режиме:

$$P_0 = \sum_{k=0}^n \frac{\rho^k}{k!} + \frac{\rho^n}{n!} \sum_{t=1}^m \frac{\rho^t}{(n+\beta) \cdot \dots \cdot (n+t \cdot \beta)}^{-1}. \quad (1.3.9)$$

где  $\rho = \lambda / \mu$ ;  
 $\beta = \nu / \mu$ .

$$P_i = \frac{\rho^i}{i!} P_0, \quad i = \overline{1, n}. \quad (1.3.10)$$

$$P_{n+j} = \frac{\rho^n}{n!} \frac{\rho^j}{(n+\beta) \cdot \dots \cdot (n+j \cdot \beta)} P_0, \quad j = 1, m. \quad (1.3.11)$$

Известны формулы для расчета характеристик системы в стационарном режиме (таблица 1).

Таблица 1 – Формулы для расчета характеристик системы массового обслуживания с «нетерпеливыми» заявками в стационарном режиме

Характеристика	Формула
1	2
Среднее число окон, занятых обслуживанием	$N_{зан} = \sum_{k=1}^{n-1} k P_k + n \cdot 1 - \sum_{k=0}^{n-1} P_k$
Среднее число заявок, ожидающих в очереди	$L_{оч} = \frac{\rho}{\beta} - \frac{N_{зан}}{\beta}$
Абсолютная пропускная способность – среднее число заявок, обслуженных за единицу времени	$A = \lambda - \nu L_{оч}$

1	2
Относительная пропускная способность – доля обслуженных заявок от общего числа	$q = \frac{A}{\lambda}$
Среднее время ожидания заявки в очереди	$t_{ож} = \frac{L_{оч}}{\lambda}$
Среднее время обслуживания заявки	$t_{обсл} = \frac{q}{\mu}$
Среднее время пребывания заявки в СМО	$t_{СМО} = t_{ож} + t_{обсл}$

Приложение системы массового обслуживания с «нетерпеливыми» заявками для описания рабочих процессов МФЦ в научной литературе найдено не было, поэтому это является предметом для дальнейшей разработки.

В рассмотренных ранее работах, посвященных моделированию деятельности МФЦ как системы массового обслуживания, не учитывается динамичность процесса подачи заявлений: авторы берут данные за определенный промежуток времени, не делая предположений о том, что число клиентов в МФЦ, может существенно варьироваться день ото дня. Чтобы адаптировать модель системы массового обслуживания к текущей ситуации, мы предлагаем воспользоваться методами прогнозирования временных рядов. Это позволит получить прогноз числа заявлений в МФЦ на каждый день на основании информации о загруженности МФЦ за предшествующий временной период.

Поскольку в МФЦ оказывается три вида услуг (прием документов, выдача результатов, консультирование), а каждый вид услуги характеризуется отдельными показателями обслуживания, и под него отводится определенное число окон, мы будем прогнозировать число заявлений по каждому виду услуг в отдельности. Получив прогноз о числе заявлений на конкретный день, можно определить оптимальное число операторов МФЦ и распределить окна обслуживания под разные виды услуг на основе расчета характеристик центра как системы массового обслуживания.

Для прогнозирования временных рядов можно воспользоваться моделями искусственных нейронных сетей, поскольку они позволяют не выдвигать предположения о характере тренда ряда и наличии сезонности. Среди них для прогнозирования одномерных временных рядов все чаще используют модели рекуррентных нейронных сетей (RNN) и их модификацию – рекуррентные нейронные сети с долгой краткосрочной памятью (RNN LSTM). За счет наличия ячеек особого типа RNN LSTM позволяют эффективно обрабатывать долгосрочные зависимости. Кроме того, рекуррентные нейронные сети с долгой краткосрочной памятью не подвержены проблеме «исчезающего градиента» [26]. Особенность архитектуры этой сети заключается в том, что в слоях этой сети находятся не нейроны, а «блоки памяти», которые содержат элементы, регулирующие состояние и отклик элемента. Наличие этих элементов позволяет LSTM-сети решать задачу распознавания краткосрочных и долгосрочных тенденций, присутствующих в данных [16]. Архитектура рекуррентной нейронной сети долгой краткосрочной памяти представлена на рисунке 2.

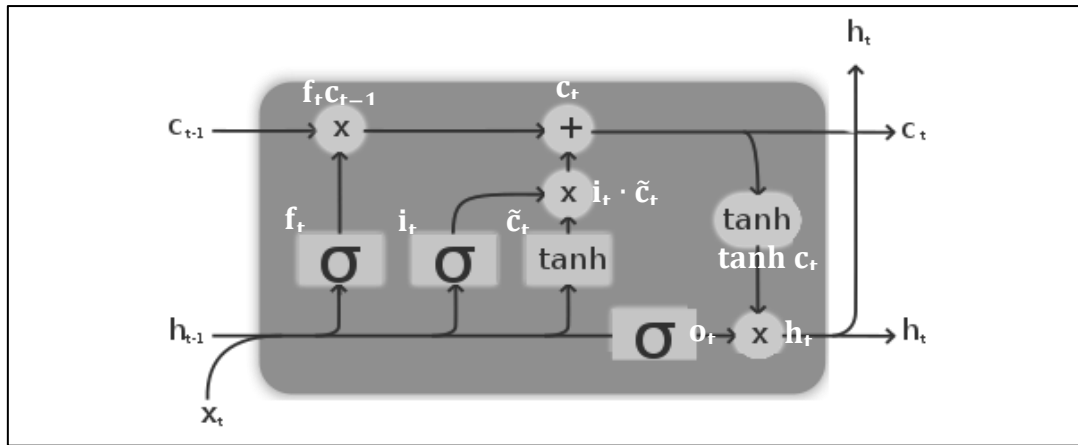


Рисунок 2 – Архитектура ячейки рекуррентной нейронной сети долгой краткосрочной памяти (LSTM cell)

Ключевыми компонентами LSTM-модуля являются состояние ячейки и три фильтра. Основной составляющей компонентой LSTM-ячейки является ее состояние  $C_t$ , которое передается во времени. Фильтры определяют прохождение сигнала: входной фильтр, фильтр забывания и выходной фильтр. Ячейка способна добавлять или удалять информацию из состояния, используя фильтры. Модель LSTM-ячейки впервые была описана в работе S. Hochreiter и J. Schmidhuber [43].

Изначально слой фильтра забывания (forget gate layer) определяет, какую информацию можно «забыть». Фильтр забывания моделируется следующей функцией:

$$f_t = \sigma(W_f[h_{t-1}, x_t] + b_f), \quad (1.3.12)$$

где  $W_f$  и  $b_f$  – веса слоя фильтра забывания;

$h_{t-1}$  – значения выхода из предыдущей ячейки;

$x_t$  – значения текущего входа;

$[h_{t-1}, x_t]$  – операция конкатенации тензоров  $h_{t-1}, x_t$ ;

$\sigma(x) = 1 / (1 + e^{-x})$  – сигмоидальная функция активации.

На следующем этапе решается, какая новая информация будет храниться в состоянии ячейки. Этот этап состоит из двух частей:

1) слой входного фильтра (input gate layer) определяет, какие значения необходимо обновить:

$$i_t = \sigma(W_i[h_{t-1}, x_t] + b_i), \quad (1.3.13)$$

где  $W_i$  и  $b_i$  – веса слоя входного фильтра.

2) следующий слой, в котором функцией активации служит гиперболический тангенс, выводит вектор значений  $c_t$ , которые можно добавить в состояние

ячейки:

$$c_t = \tanh(W_c[h_{t-1}, x_t] + b_c), \quad (1.3.14)$$

где  $W_c$  и  $b_c$  – веса слоя входного фильтра.

После этого происходит обновление состояния ячейки  $c_t$ :

$$c_t = f_t \cdot c_{t-1} + i_t \cdot c_t. \quad (1.3.15)$$

На последнем этапе определяется выход ячейки. Выходной фильтр (output gate layer) определяет, какая информация из состояния ячейки будет выведена:

$$o_t = \sigma(W_o[h_{t-1}, x_t] + b_o), \quad (1.3.16)$$

где  $W_o$  и  $b_o$  – веса слоя выходного фильтра.

Выход ячейки зависит от состояния ячейки и значений, полученных от выходного фильтра:

$$h_t = o_t \cdot \tanh c_t. \quad (1.3.17)$$

В качестве алгоритма оптимизации для обновления весов нейронной сети, наиболее распространенным является алгоритм стохастической оптимизации Adaptive Moment Estimation (adam), математическая модель которого имеет вид [45]:

$$g_t = \nabla_{\theta} L(\theta_{t-1}); \quad (1.3.18)$$

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t, \quad \hat{m}_t = \frac{m_t}{1 - \beta_1^t}; \quad (1.3.19)$$

$$\nu_t = \beta_2 \nu_{t-1} + (1 - \beta_2) g_t^2, \quad \nu_t = \frac{\nu_t}{1 - \beta_2^t}; \quad (1.3.20)$$

$$\theta_t = \theta_{t-1} - \frac{\alpha}{\nu_t + \varepsilon} \hat{m}_t, \quad (1.3.21)$$

где  $\theta$  – вектор-параметров модели;

$L(\theta)$  – функция потерь;

$g$  – вектор-градиент функции потерь;

$m$  – вектор накопленных значений градиента (импульс);

$\nu$  – вектор накопленных значений квадрата градиента;

$\alpha$  – коэффициент обучения, общий для всех параметров;

$\beta_1$  – параметр сглаживания для  $m$ ;

$\beta_2$  – параметр сглаживания для  $\nu$ .

Алгоритм adam объединяет идеи двух алгоритмов оптимизации: Мomen-

tum SGD (стохастический градиентный спуск с накоплением градиента) и Root Mean Square Propagation (RMSProp, метод адаптивного скользящего среднего). Накопление градиента (1.3.19) позволяет избегать застревания в точках локального минимума, а адаптивное изменение коэффициента обучения (1.3.20) позволяет подбирать оптимальную скорость обучения для каждого параметра модели.

Практическая значимость моделирования работы многофункционального центра как системы массового обслуживания заключается в возможности априорного анализа эффективности и оптимизации функционирования МФЦ за счет определения наиболее эффективного распределения окон обслуживания под оказание услуг определенного типа.

3. Разработка рекомендательной системы для прогнозирования следующей услуги заявителя.

Для прогнозирования следующей услуги, за которой обратится заявитель, на основе информации об его прошлых обращениях, можно воспользоваться методами и моделями прогнозирования последовательностей и многоклассовой классификации. Научных работ, в которых проводилось исследование приложения указанных моделей для прогнозирования транзакций клиентов МФЦ, обнаружено не было, поэтому адаптация и внедрение моделей прогнозирования последовательностей в систему МФЦ также является предметом для дальнейшей разработки.

Наиболее популярным решением для прогнозирования элементов последовательности является модель рекуррентной нейронной сети с долгой краткосрочной памятью (LSTM-сеть), поскольку ее архитектура позволяет выявлять зависимости между элементами последовательности, разделенными большими промежутками времени. Подробное описание применения модели рекуррентной нейронной сети с долгой краткосрочной памятью для прогнозирования последовательных событий приводится в работах [50, 53].

Если задачу прогнозирования следующей услуги заявителя МФЦ представить как задачу многоклассовой классификации, где один класс соответствует отдельной услуге, то можно воспользоваться методами классификации. Среди них выделяется алгоритм градиентного бустинга деревьев принятий решений. Бустинг — это подход к построению ансамбля алгоритмов, в рамках которого базовые алгоритмы строятся последовательно, а каждый следующий алгоритм строится таким образом, чтобы исправлять ошибки уже построенной композиции. Направленное построение композиций в бустинге позволяет использовать простые базовые алгоритмы, например, неглубокие деревья, что позволяет сократить вычислительную сложность задачи и время на ее решение. Математическая модель алгоритма имеет вид [42]:

На тренировочной выборке  $X = \{X_i\}, i = 1, N$  обучается ансамбль из  $K$  деревьев классификации:

$$F_K(x) = \sum_{k=1}^K f(x; \gamma_k), \quad f \in \mathfrak{F}, \quad (1.3.22)$$



где  $\mathcal{F}$  – множество деревьев классификации;  
 $\gamma_k$  – параметры дерева, добавляемого на  $k$ -м шаге.

Предсказанное значение класса  $y_i$  для  $i$ -го объекта вычисляется, как

$$y_i = F_K(x_i) = \sum_{k=1}^K f(x_i; \gamma_k). \quad (1.3.23)$$

Работа алгоритма осуществляется по шагам:

1. Задать  $f_0(x) = 0$ ;

2. Для  $k = 1, K$ :

2.1. Вычислить коэффициент при  $k$ -м дереве в ансамбле и его параметры:

$$(\beta_k, \gamma_k) = \arg \min_{\beta, \gamma} \sum_{i=1}^N L(y_i, F_{k-1}(x_i) + \beta \cdot f(x_i; \gamma)). \quad (1.3.24)$$

2.2. Добавить новое дерево к ансамблю:

$$F_k(x) = F_{k-1}(x) + \beta_k \cdot f(x; \gamma_k), \quad (1.3.25)$$

где  $L(y_i, y_i)$  – функция потерь;

$\beta$  – величина шага;

$\gamma$  – параметры дерева (переменные для разбиения, число объектов во внутренних узлах и прогнозные значения в листьях).

Целевая функция, которая минимизируется в алгоритме, – это функция потерь  $L(y_i, y_i)$ . В случае многоклассовой классификации в качестве функции потерь может выступать категориальная кросс-энтропия:

$$CE(y_{ij}, y_{ij}) = -\frac{1}{N} \sum_{i=1}^N \sum_{j=1}^C y_{ij} \cdot \log(y_{ij}), \quad (1.3.26)$$

где  $N$  – число объектов в выборке (батче);

$C$  – число классов;

$y_{ij}$  – бинарная переменная ( $i$ -тый объект принадлежит  $j$ -му классу);

$y_{ij}$  – оценка вероятности ( $i$ -тый объект принадлежит  $j$ -му классу).

Чтобы добавить очередное  $k$ -тое дерево к ансамблю:

$$F_k(x) = F_{k-1}(x) + f(x; \gamma_k), \quad (1.3.27)$$

необходимо минимизировать функцию потерь:

$$\sum_{i=1}^N L(y_i, F_{k-1}(x_i) + f(x_i; \gamma_k)) \rightarrow \min. \quad (1.3.28)$$

Минимум функции (22) ищется при помощи методов градиентного спуска. Поскольку направлением наискорейшего убывания функции многих переменных является ее вектор-антиградиент, то новое -тое дерево строится в соответствии со следующим соотношением:

$$f(x_i; \gamma_k) = -L'_f(y_i, F_{k-1}(x_i)). \quad (1.3.29)$$

Соотношение (1.3.28) означает, что добавляемое на  $k$ -ом шаге дерево следует обучать на тренировочной выборке вида:

$$X^k = \{x_i, -L'(y_i, F_{k-1}(x_i))\}, i = 1, N. \quad (1.3.30)$$

В настоящее время разработано большое количество модификаций алгоритма, среди которых наиболее часто применяемыми являются алгоритмы XGBoost [40], LightGBM [44] и CatBoost [48].

Для прогнозирования элементов последовательности можно также использовать языковые модели, к которым относится модель компактного дерева предсказания (Compact Prediction Tree, CPT), предложенная в работе Т. Gueniche, Ph. Fournier-Viger и V.S. Tseng [41]. Модель CPT включает в себя три структуры данных:

1) дерево предсказаний (Prediction Tree, PT): предназначено для хранения обучающих последовательностей.

2) обратный индекс (Inverted Index, II): двумерная двоичная матрица, строки которой соответствуют уникальным элементам последовательностей, содержащихся в PT.

3) массив указателей (Lookup Table, LT): указатели на последние элементы последовательностей в PT, доступные по номерам последовательностей.

Пусть для последовательности  $S = (s_1, s_2, \dots, s_n)$  требуется определить элемент  $s_{n+1}$ . Обученная модель CPT позволяет сделать это следующим образом: после задания некоторого числа  $m = 2, n - 1$  из дерева предсказаний выбираются все последовательности, обозначаемые  $\mathbb{S}$ , которые содержат последние  $m$  элементов последовательности  $S$ :  $(s_{n-m}, s_{n-m+1}, \dots, s_n)$ , причем необязательно в том же порядке. Для каждой последовательности  $S_i \in \mathbb{S}$  выделяется общий префикс  $S_i$  и  $S$ , обозначаемый  $p^i = (p_1, p_2, \dots, p_{n_i})$ . Множество всех таких префиксов обозначим  $P$ . Для каждого префикса из  $P$  производится следующая процедура: каждому элементу  $p_j, j = 1, n_i$  из префикса  $p^i \in P$  ставится в соответствие число  $c = k/r$ , где  $k$  – число вхождений элемента  $p_j$  в последовательностях из  $\mathbb{S}$ ,  $r$  – число последовательностей в PT, содержащих этот элемент.

Элементы с минимальным значением  $k$  – кандидаты. Искомым элементом принимают кандидата с наименьшим значением  $s$ .

Визуализация предложенной схемы исследования представлена на рисунке 3.

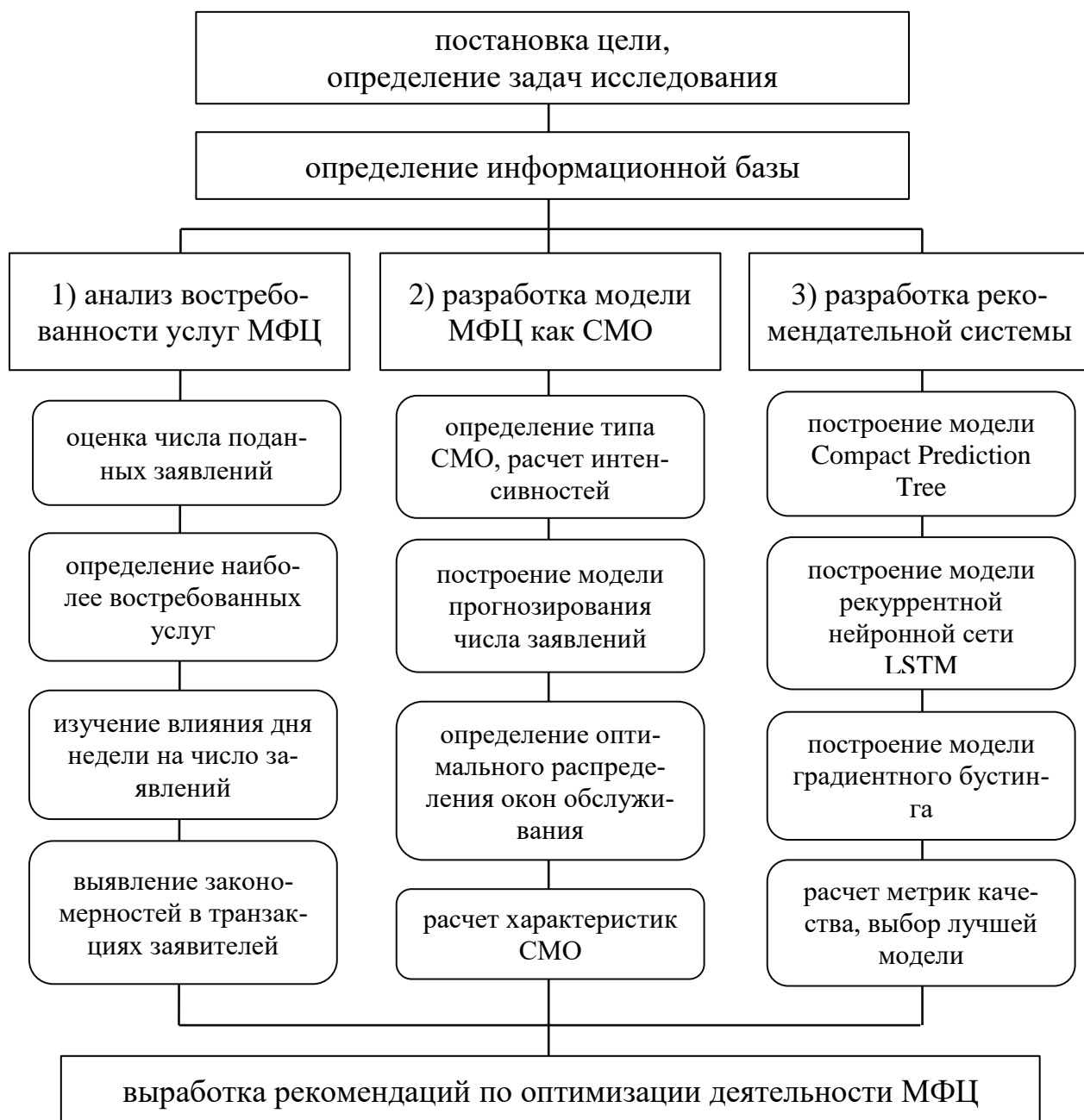


Рисунок 3 – Схема алгоритма проведения исследования

Обобщая сказанное в первой главе, можно сделать следующие выводы:

— многофункциональный центр предоставления государственных и муниципальных услуг является новой перспективной формой организации взаимодействия между населением и государством, позволяющей значительно упростить и ускорить процесс оказания услуг. Этот факт обуславливает актуальность решения задач по оптимизации работы МФЦ, которая может быть до-

стигнута при применении математических методов моделирования и прогнозирования, а также методов машинного обучения. При этом необходимо учитывать специфику организации работы и функционирования многофункциональных центров, которая хорошо изучена и изложена в работах [18, 19, 20, 21, 22, 31, 32];

— для математического моделирования деятельности МФЦ могут применяться разнообразные математические методы и модели: методы имитационного моделирования, аппарат теории систем массового обслуживания, методы моделирования бизнес-процессов. Приложения указанных методов в разрезе работы многофункциональных центров достаточно хорошо разработаны и описаны в работах российских авторов [4,8,9,10,21,28,33];

— для решения более специфических задач – например, прогнозирования следующей услуги, за которой обратится заявитель, можно применять такие методы, как поиск ассоциативных правил, анализ последовательных шаблонов, нейронные сети как модели прогнозирования последовательностей. В настоящее время научных работ, посвященных внедрению указанных методов в систему МФЦ, нет, что говорит о перспективности изучения этого направления. Однако методы поиска ассоциативных правил и последовательных шаблонов широко применяются в смежных областях – от торговли до банковской сферы [24, 37, 39]. Если задачу прогнозирования следующей услуги представить как задачу прогнозирования последовательностей, то можно воспользоваться подходами, описанными в работах [38,50,53];

— в рамках работы предложен алгоритм математического моделирования деятельности многофункциональных центров, схематичное представление которого приведено на рисунке 3.

## 2 Разработка математических моделей выявления закономерностей в поступлении заявлений и организации процесса обслуживания заявок

### 2.1 Анализ востребованности услуг ГАУ «МФЦ Иркутской области» и выявление закономерностей в транзакциях заявителей

Исходные данные о заявителях МФЦ, истории их обращений за услугами и талонах электронной очереди содержат наблюдения за период 06.01.2019 – 18.05.2020. За указанный период времени заявления на оказание услуг или получение консультаций подали 888862 заявителя, что составило 50,92% от всех пользователей, зарегистрированных в системе МФЦ региона. Результаты расчета точечных оценок основных числовых характеристик для случайной величины «Число заявлений, поданных пользователем системы МФЦ за период 06.01.2019 – 18.05.2020» представлены в таблице 2.

Таблица 2 – Точечные оценки основных числовых характеристик показателя «Число заявлений, поданных пользователем системы МФЦ за период 06.01.2019 – 18.05.2020»

Число наблюдений, шт.	Среднее значение, шт.	Медиа-на, шт.	Среднеквадратическое отклонение, шт.	Минимальное значение, шт.	Максимальное значение, шт.	Коэффициент асимметрии	Коэффициент эксцесса
888862	3,05	2	50,53	1	20749	220,39	64411,80

Ввиду того, что наблюдается сильный разброс в числе поданных заявлений (размах составил 20749), было выдвинуто предположение, что при анализе этого показателя нужно учитывать тип заявителя и рассчитывать оценки числовых характеристик отдельно внутри каждой группы. Полученные результаты расчета оценок числовых характеристик приведены в таблице 3.

Таблица 3 – Точечные оценки основных числовых характеристик показателя «Число заявлений, поданных пользователем системы МФЦ за период 06.01.2019 – 18.05.2020» с учетом типа заявителя

Число наблюдений, шт.	Среднее значение, шт.	Медиа-на, шт.	Среднеквадратическое отклонение, шт.	Минимальное значение, шт.	Максимальное значение, шт.	Коэффициент асимметрии	Коэффициент эксцесса
Для физических лиц							
873248	2,89	2	49,79	1	20748	252,29	82760,77
Для юридических лиц							
7828	21,01	2	211,81	1	12058	36,11	1694,70
Для индивидуальных предпринимателей							
7786	2,72	1	23,27	1	1557	60,18	3760,35

Как можно заметить, размах числа поданных заявлений все еще довольно высок в каждой группе. Это может быть связано с тем, как давно заявитель пользуется сервисом МФЦ. Если разбить исходную выборку на группы по времени их регистрации в системе МФЦ, то, возможно, в данных наблюдалось бы большее однообразие, однако формат исходных данных не предполагает хранение информации о первом обращении заявителя в МФЦ.

Самыми востребованными услугами МФЦ являются:

а) для физических лиц:

1) государственный кадастровый учет недвижимого имущества и (или) государственная регистрация прав на недвижимое имущество (395010 заявлений);

2) регистрационный учёт граждан Российской Федерации по месту пребывания и по месту жительства в пределах Российской Федерации (в части приема и выдачи документов о регистрации и снятии граждан Российской Федерации с регистрационного учета по месту пребывания и по месту жительства в пределах Российской Федерации) (181890 заявлений);

3) государственный кадастровый учет недвижимого имущества и (или) государственная регистрация прав на недвижимое имущество (170109 заявлений);

4) предоставление сведений, содержащихся в Едином государственном реестре недвижимости (108355 заявлений);

5) выдача, замена паспортов гражданина Российской Федерации, удостоверяющих личность гражданина Российской Федерации на территории Российской Федерации (96922 заявления);

6) выдача справок о наличии (отсутствии) судимости и (или) факта уголовного преследования либо о прекращении уголовного преследования" (89096 заявлений);

7) регистрационный учёт граждан Российской Федерации по месту пребывания и по месту жительства в пределах Российской Федерации (в части приема и выдачи документов о регистрации и снятии граждан Российской Федерации с регистрационного учета по месту пребывания и по месту жительства в пределах Российской Федерации) (86195 заявлений);

8) регистрация пользователя ЕСИА (70121 заявление);

9) назначение и выплата пособия на ребенка в Иркутской области (68653 заявления);

10) предоставление в Иркутской области отдельных мер социальной поддержки семьям, имеющим детей (62862 заявления);

б) для юридических лиц:

1) государственный кадастровый учет недвижимого имущества и (или) государственная регистрация прав на недвижимое имущество (63417 заявлений);

2) предоставление сведений, содержащихся в Едином государственном реестре недвижимости (40458 заявлений);

3) государственный кадастровый учет недвижимого имущества и (или) государственная регистрация прав на недвижимое имущество (27547 заявлений);

4) предоставление сведений, содержащихся в Едином государственном реестре недвижимости (8297 заявлений);

5) осуществление миграционного учета иностранных граждан и лиц без гражданства в Российской Федерации (в части приема уведомления о прибытии иностранного гражданина или лица без гражданства в место пребывания и предоставления отметки о приеме уведомления) (8292 заявлений);

6) осуществление миграционного учета иностранных граждан и лиц без гражданства в Российской Федерации (в части приема уведомления о прибытии иностранного гражданина или лица без гражданства в место пребывания и предоставления отметки о приеме уведомления) (3176 заявлений);

7) государственная регистрация юридических лиц, физических лиц в качестве индивидуальных предпринимателей и крестьянских (фермерских) хозяйств ЭЛЕКТРОННАЯ (2605 заявлений);

8) государственная регистрация юридических лиц, физических лиц в качестве индивидуальных предпринимателей и крестьянских (фермерских) хозяйств (1318 заявлений);

9) предоставление сведений, содержащихся в Едином государственном реестре недвижимости (Филиал) (1289 заявлений);

10) государственная услуга по выдаче лицензий на розничную продажу алкогольной продукции на территории Иркутской области (835 заявлений);

в) для индивидуальных предпринимателей:

1) государственная регистрация юридических лиц, физических лиц в качестве индивидуальных предпринимателей и крестьянских (фермерских) хозяйств ЭЛЕКТРОННАЯ (2922 заявлений);

2) регистрация на Портале Бизнес-навигатора МСП (2869 заявлений);

3) услуга по предоставлению по заданным параметрам информации о формах и условиях финансовой поддержки субъектов малого и среднего предпринимательства (2092 заявлений);

4) государственный кадастровый учет недвижимого имущества и (или) государственная регистрация прав на недвижимое имущество (1560 заявлений);

5) услуга по подбору по заданным параметрам информации об имуществе, включенном в перечни государственного и муниципального имущества, предусмотренные частью 4 статьи 18 Федерального закона от 24 июля 2007 г № 209-ФЗ «О развитии малого и среднего предпринимательства в Российской Федерации», и свободном от прав третьих лиц (1318 заявлений);

6) государственная регистрация юридических лиц, физических лиц в качестве индивидуальных предпринимателей и крестьянских (фермерских) хозяйств (1225 заявлений);

7) осуществление миграционного учета иностранных граждан и лиц без гражданства в Российской Федерации (в части приема уведомления о прибытии

иностранный гражданин или лица без гражданства в место пребывания и проставления отметки о приеме уведомления) (1006 заявлений);

8) государственный кадастровый учет недвижимого имущества и (или) государственная регистрация прав на недвижимое имущество (776 заявлений);

9) услуга по предоставлению по заданным параметрам информации об объемах и номенклатуре закупок конкретных и отдельных заказчиков, определенных в соответствии с Федеральным законом от 18 июля 2011 г. № 223-ФЗ «О закупках товаров, работ, услуг отдельными видами юридических лиц», у субъектов малого и среднего предпринимательства в текущем году (729 заявлений);

10) осуществление миграционного учета иностранных граждан и лиц без гражданства в Российской Федерации (в части приема уведомления о прибытии иностранного гражданина или лица без гражданства в место пребывания и проставления отметки о приеме уведомления) (595 заявлений).

В 2020 г. по сравнению с 2019 г. в начале года наблюдалась тенденция к увеличению общего числа заявлений на предоставление услуг (рисунок 4). Однако в апреле произошел резкий спад числа обращений в МФЦ. Это можно объяснить введением режима нерабочих дней с 30 марта по 5 апреля, во время которых многие услуги, предоставляемые МФЦ, оказались невостребованными. Несмотря на это можно высказать предположение, что в дальнейшем услуги МФЦ останутся востребованными у населения.

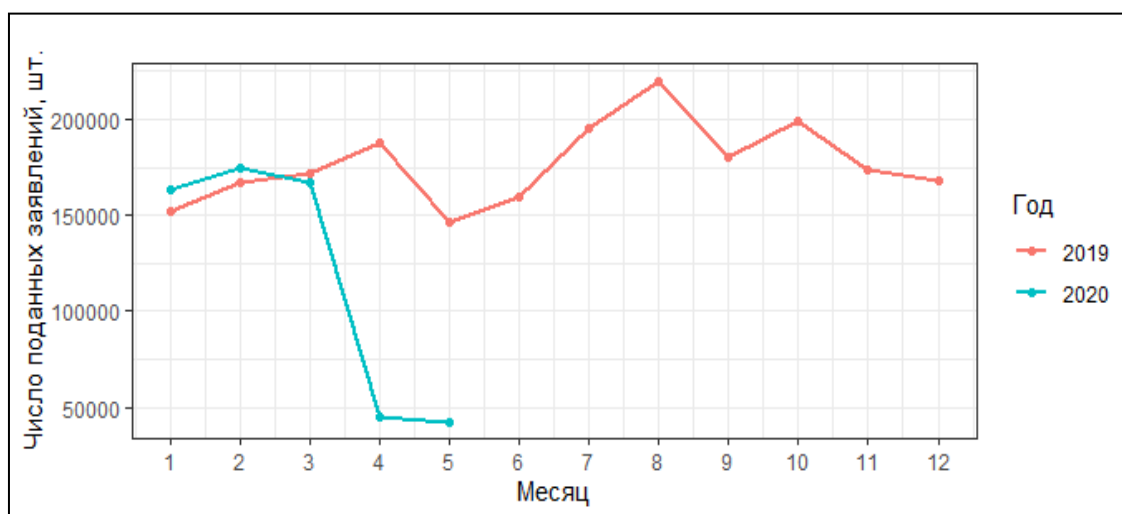


Рисунок 4 – Динамика числа поданных заявлений на оказание услуги или получение консультации в МФЦ

Проанализируем, как зависит число поданных заявлений от месяца и дня недели. На рисунке 5 приведена диаграмма распределения числа заявлений по месяцам и по дням недели. Поскольку данные за 2020 г. представлены только за пять месяцев, число заявлений по месяцам рассчитывалось только по данным за 2019 г.



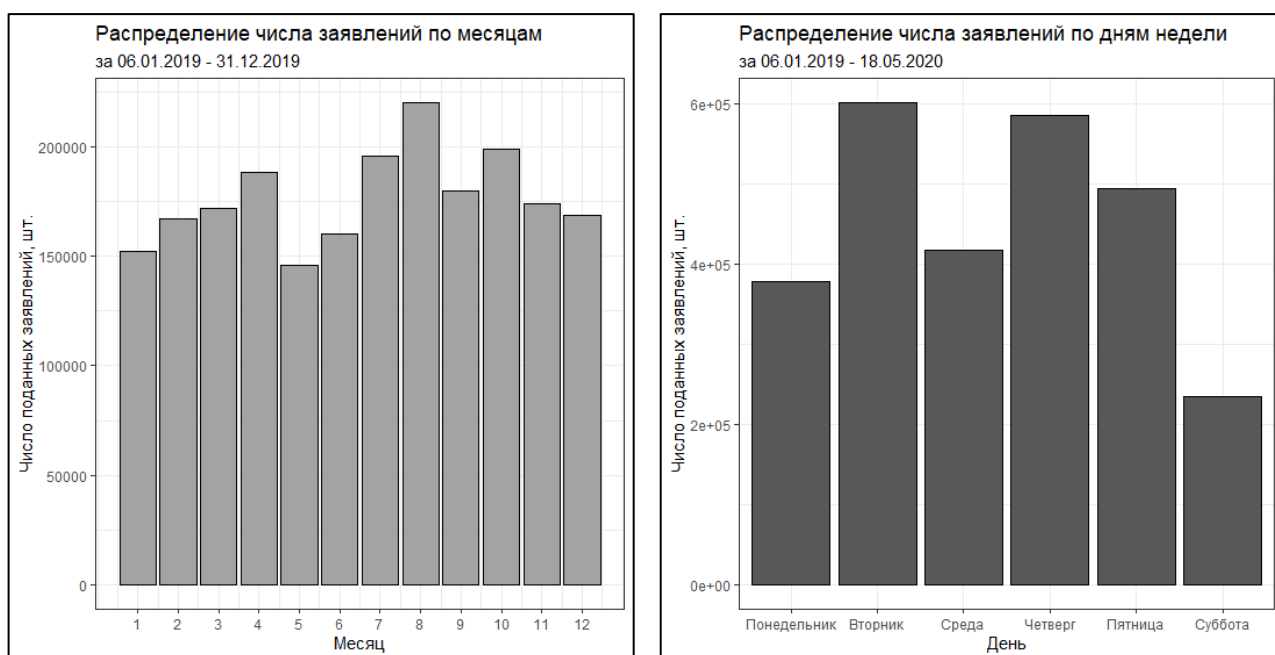


Рисунок 5 – Распределение числа заявлений по месяцам и по дням недели

Согласно рисунку 5, наибольшее число заявлений приходится на август, октябрь и апрель, наименьшее – на январь и май. Пик числа заявлений приходится на вторник и четверг, а меньше всего офисы МФЦ загружены в понедельник, среду и субботу. Таким образом, можно предположить, что число заявлений зависит как от месяца, так и от дня недели. Чтобы проверить эту гипотезу, воспользуемся методами однофакторного дисперсионного анализа. Однако для проверки гипотезы о влиянии месяца на число заявлений требуется данные за больший промежуток времени, поэтому далее исследуем влияние только дня недели.

Априорная линейная модель однофакторного дисперсионного анализа имеет вид:

$$\xi_{ij} = m_0 + \alpha_j + \varepsilon_{ij}, \quad j = \overline{1, k}, i = \overline{1, n_j}, \quad (2.1.1)$$

где  $m_0$  – математическое ожидание случайной величины  $\xi$ ;

$\alpha_j$  – вклад в величину  $\xi_{ij}$ , обусловленный влиянием фактора А (влияние или эффект j-го уровня фактора А);

$\varepsilon_{ij}$  – вклад в величину  $\xi_{ij}$ , обусловленный влиянием неучтенных факторов, случайных ошибок эксперимента.

Гипотеза об отсутствии влияния фактора А на  $\xi$  формируется следующим образом:

$$H_0: \alpha_j = 0 \quad \forall j = \overline{1, k};$$

$$H_1: \exists j : \alpha_j \neq 0.$$

Результативным признаком  $\xi$  является число поданных заявлений на ока-

зание услуг, а фактором, оказывающим влияние на результативный признак, – день недели.

Предварительно исключим из выборки наблюдения за праздничные дни и воскресенья, поскольку они будут вносить искажения в результаты статистических критериев, так как число заявлений в такие дни значительно ниже обычного. Реализация теста Колмогорова-Смирнова с поправкой Лиллиефорса показала, что распределение результативного признака отлично от нормального ( $D = 0.13053$ ,  $p\text{-value} < 2.2e-16$ ), поэтому необходимо использовать методы непараметрического однофакторного дисперсионного анализа – например, критерий Краскала-Уоллиса.

Согласно результатам, полученным при реализации теста Краскала-Уоллиса, наблюдаемое значение статистики  $\chi^2_{\text{набл}} = 201,47$ , наблюдаемый уровень значимости  $p\text{-value} < 2.2e-16$  – нулевая гипотеза об отсутствии влияния дня недели на число поданных заявлений отвергается. Руководство центра может учесть эту информацию для составления графика работы МФЦ или графика отпусков сотрудников центра.

Для выявления закономерностей в транзакциях клиентов МФЦ можно воспользоваться методами поиска ассоциативных правил [24, 39] и последовательных шаблонов [51].

Поскольку для каждого типа заявителей специфичны определенные услуги, выборка была предварительно разделена на три части: для физических лиц, юридических лиц и индивидуальных предпринимателей. На следующем этапе были исключены из рассмотрения заявители, у которых за исследуемый период времени было зафиксировано только одно обращение. Также не были включены заявители с числом заявлений больше ста.

После выполнения указанных преобразований на языке программирования R был реализован алгоритм поиска ассоциативных правил Apriori. Результаты его работы для физических лиц приведены в таблице 4. Минимальное значение поддержки установлено на уровне 1,5%; минимальное значение достоверности – 50%.

Таблица 4 – Результат работы алгоритма поиска ассоциативных правил для физических лиц

№	«Если» (ID услуги)	«То» (ID услуги)	Поддерж- ка, %	Достовер- ность, %	Лифт	Число транзакций
1	604	651	4,77	82,67	10,59	21384
2	651	604	4,77	61,05	10,59	21384
3	901	1169	4,28	96,06	8,21	19205
4	491	155	1,80	56,27	14,99	8097
5	692	805	1,80	54,52	9,56	8074
6	107	651	1,73	91,33	11,70	7776
7	569	604	1,54	94,48	16,39	6892

Результаты работы алгоритма Apriori для юридических лиц приведены в таблице 5. Минимальное значение поддержки установлено на уровне 2%; минимальное значение достоверности – 50%.

Таблица 5 – Результат работы алгоритма поиска ассоциативных правил для юридических лиц

№	«Если» (ID услуги)	«То» (ID услуги)	Поддерж- ка, %	Достовер- ность, %	Лифт	Число транзакций
1	98	134	17,22	55,85	1,25	831
2	4	134	7,54	62,65	1,40	364
3	4,98	134	3,96	82,68	1,85	191
4	134,4	98	3,96	52,47	1,70	191
5	1020	134	3,31	53,51	1,20	160
6	1020	98	3,27	52,84	1,71	158
7	598	1306	2,65	90,14	16,42	128
8	1020,98	134	2,47	75,32	1,68	119
9	1020,134	98	2,47	74,38	2,41	119
10	320,98	134	2,09	75,94	1,70	101
11	826	949	2,03	70,50	13,29	98

Результаты работы алгоритма Apriori для индивидуальных предпринимателей приведены в таблице 6. Минимальное значение поддержки установлено на уровне 7%; минимальное значение достоверности – 50%.

Таблица 6 – Результат работы алгоритма поиска ассоциативных правил для индивидуальных предпринимателей

№	«Если» (ID услуги)	«То» (ID услуги)	Поддерж- ка, %	Достовер- ность, %	Лифт	Число транзакций
1	598	1306	13,68	87,77	2,23	531
2	60	1306	12,36	58,90	1,50	480
3	700	1306	10,64	81,46	2,07	413
4	43	1306	9,43	89,71	2,28	366
5	598	60	8,89	57,02	2,72	345
6	43	598	8,22	78,19	5,02	319
7	598	43	8,22	52,73	5,02	319
8	598,6	1306	8,09	91,01	2,31	314
9	1306,598	60	8,09	59,13	2,82	314
10	1306,6	598	8,09	65,42	4,20	314
11	43,598	1306	7,78	94,67	2,40	302
12	1306,43	598	7,78	82,51	5,29	302
13	1306,598	43	7,78	56,87	5,41	302
14	700	598	7,62	58,38	3,75	296

Опишем полученные результаты. Поддержка означает долю транзакций, для которых было характерно появление набора услуг. Отсортировав правила по убыванию значения поддержки, приведем наиболее часто встречающиеся правила:

а) для физических лиц:

1) услуга №604 (Предоставление в Иркутской области отдельных мер социальной поддержки семьям, имеющим детей) => услуга №651 (Назначение и выплата пособия на ребенка в Иркутской области) с вероятностью 0,83. Обратное правило имеет достоверность 0,61;

2) услуга №901 (Выдача справок о том, является или не является лицо подвергнутым административному наказанию за потребление наркотических средств или психотропных веществ без назначения врача либо новых потенциально опасных психоактивных веществ) => услуга №1169 (Выдача справок о наличии (отсутствии) судимости и (или) факта уголовного преследования либо о прекращении уголовного преследования) с вероятностью 0,96;

3) услуга №491 (Поиск учетной записи ЕСИА) => услуга №155 (Восстановление доступа к подтвержденной учетной записи ЕСИА) с вероятностью 0,56;

б) для юридических лиц:

1) услуга №98 (Государственный кадастровый учет недвижимого имущества и (или) государственная регистрация прав на недвижимое имущество) => услуга №134 (Государственный кадастровый учет недвижимого имущества и (или) государственная регистрация прав на недвижимое имущество – повторное) с вероятностью 0,56;

2) услуга №4 (Предоставление сведений, содержащихся в Едином государственном реестре недвижимости – повторное) => услуга №134 (Государственный кадастровый учет недвижимого имущества и (или) государственная регистрация прав на недвижимое имущество – повторное) с вероятностью 0,62;

3) услуга №4 и услуга №98 => услуга №134 с вероятностью 0,82;

в) для индивидуальных предпринимателей:

1) услуга №598 (Услуга по предоставлению по заданным параметрам информации об объемах и номенклатуре закупок конкретных и отдельных заказчиков, определенных в соответствии с Федеральным законом от 18 июля 2011 г. № 223-ФЗ «О закупках товаров, работ, услуг отдельными видами юридических лиц», у субъектов малого и среднего предпринимательства в текущем году) => услуга №1306 (Услуга по предоставлению по заданным параметрам информации о формах и условиях финансовой поддержки субъектов малого и среднего предпринимательства) с вероятностью 0,88;

2) услуга №60 (Услуга по подбору по заданным параметрам информации об имуществе, включенном в перечни государственного и муниципального имущества, предусмотренные частью 4 статьи 18 ФЗ от 24 июля 2007 г № 209-ФЗ «О развитии малого и среднего предпринимательства в Российской Федерации», и свободном от прав третьих лиц) => услуга №1306 с вероятностью 0,59;

3) услуга №700 (Услуга по предоставлению информации об органах

государственной власти РФ, органах местного самоуправления, организациях, образующих инфраструктуру поддержки субъектов малого и среднего предпринимательства, о мерах и условиях поддержки, предоставляемой на федеральном, региональном и муниципальном уровнях субъектам малого и среднего предпринимательства) => услуга №1306 с вероятностью 0,81.

Поскольку транзакции клиентов упорядочены во времени, алгоритм поиска последовательных шаблонов SPADE также может находить полезные закономерности в последовательностях обращений клиентов МФЦ. Для его реализации необходимо предварительное преобразование исходной базы данных: данные группируются по каждому заявителю, а затем – по времени, прошедшему с первого дня базы данных, измеренному в месяцах. Таким образом, для каждого клиента составляется последовательность услуг, за которыми он обращался в МФЦ в течение одного конкретного месяца. В целях выявления более общих закономерностей и оптимизации времени работы алгоритма из базы данных были удалены пользователи, для которых число заявлений за все историю обращений в МФЦ превысило сто обращений. После чего к преобразованной базе данных применяется алгоритм SPADE.

При реализации алгоритма SPADE для физических и юридических лиц последовательных шаблонов не было обнаружено. Для индивидуальных предпринимателей результат работы алгоритма приведен в таблице 7. Минимальная поддержка установлена на уровне 20%, минимальная достоверность – 50%. Отметим достаточно большие значения поддержки и вместе с тем малую вариативность фигурирующих в шаблонах услуг.

Таблица 7 – Результат работы алгоритма поиска последовательных шаблонов для индивидуальных предпринимателей

№	«Если» (ID услуги)	«То» (ID услуги)	Поддержка, %	Достоверность, %	Лифт
1	134	134	48,70	74,39	1,14
2	134	98	38,06	58,13	1,21
3	134,134	134	34,79	71,43	1,09
4	134,134	98	32,77	67,30	1,41
5	98	98	28,33	59,19	1,24
6	134,134,134	98	26,82	77,11	1,61
7	134,134,134	134	26,49	76,14	1,16
8	134,98	98	24,31	63,88	1,33
9	134,134,98	98	22,80	69,57	1,45
10	134,134,134,134	98	22,46	84,81	1,77
11	134,134,134,134	134	20,96	79,11	1,21
12	134,134,134,98	98	20,29	75,63	1,58

Полученные последовательные шаблоны не являются достаточно информативными ввиду присутствия в них только двух видов услуг. Однако они поз-

воляют сделать вывод, что для большого числа заявителей характерно обращение за услугами №134 и №98, представляющими собой первичное или повторное обращение за услугой «Государственный кадастровый учет недвижимого имущества и (или) государственная регистрация прав на недвижимое имущество». Такая закономерность может указывать на то, что заявитель является индивидуальным предпринимателем, оказывающим риэлтерские услуги.

## **2.2 Моделирование деятельности многофункционального центра как системы массового обслуживания**

Для построения математической модели работы многофункционального центра воспользуемся аппаратом системы массового обслуживания. Это позволит провести априорный анализ эффективности, на основе которого можно предложить варианты оптимизации функционирования МФЦ: в частности, на основе расчета характеристик системы массового обслуживания оценить загруженность системы, выявить «узкие» места, найти оптимальное распределение окон, отведенных под оказание услуг определенного типа [13, 28].

Для построения модели используются исходные данные, представленные двумя информационными базами:

1) по талонам электронной очереди (рисунок А.3): идентификатор талона, начало обслуживания, окончание обслуживания

2) по оказанию услуг в рамках талона (рисунок А.4): идентификатор талона ЭО, тип услуги, идентификатор пользователя.

Для рассмотрения выберем только один офис: Отдел по обслуживанию заявителей в г. Иркутске № 1. За период 06.01.2019 – 17.05.2020 в нем было выдано 240 тысяч талонов электронной очереди. Согласно информации, указанной на официальном сайте ГАУ «МФЦ Иркутской области», номинальное количество окон обслуживания в выбранном офисе равно 30.

Поскольку в МФЦ оказывается три вида услуг (прием документов, выдача результатов, консультирование), а каждый вид услуги характеризуется отдельными показателями обслуживания, и под него отводится определенное число окон, мы будем моделировать работу центра по каждому виду услуг в отдельности. Динамика числа талонов электронной очереди по дням приведена на рисунке 6.

Отметим снижение числа заявлений, которое можно объяснить введением режима нерабочих дней из-за пандемии COVID-19 в период с 30 марта по 30 апреля 2020 г. Поскольку резкое снижение числа заявлений было вызвано внешними непрогнозируемыми обстоятельствами, в дальнейшем этот период не будет учитываться при построении модели; период наблюдения составит 348 дней (с 9 января 2019 г. по 28 апреля 2020 г.).

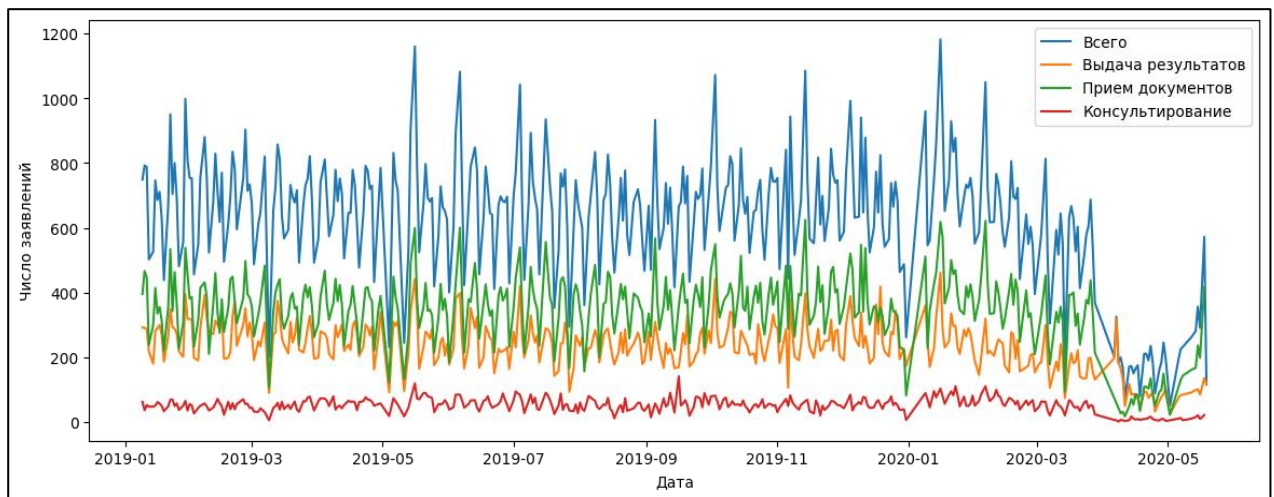


Рисунок 6 – Динамика числа талонов электронной очереди, зарегистрированных за один день, за период 09.01.2019 – 19.05.2020

Первым этапом исследования деятельности центра как системы массового обслуживания является исследование закона распределения входного потока (времени между выдачей последовательных талонов) и времени обслуживания одной заявки (времени обслуживания заявителя по одному талону).

Проведем исследование закона распределения входного потока по каждому виду услуг. Временной период примем равным одному дню, который выбирается случайно. Предварительный анализ гистограммы распределения времени между поступлением последовательных заявок на обслуживание (рисунок 7) позволил выдвинуть предположение, что для каждого типа услуги оно распределено экспоненциально.

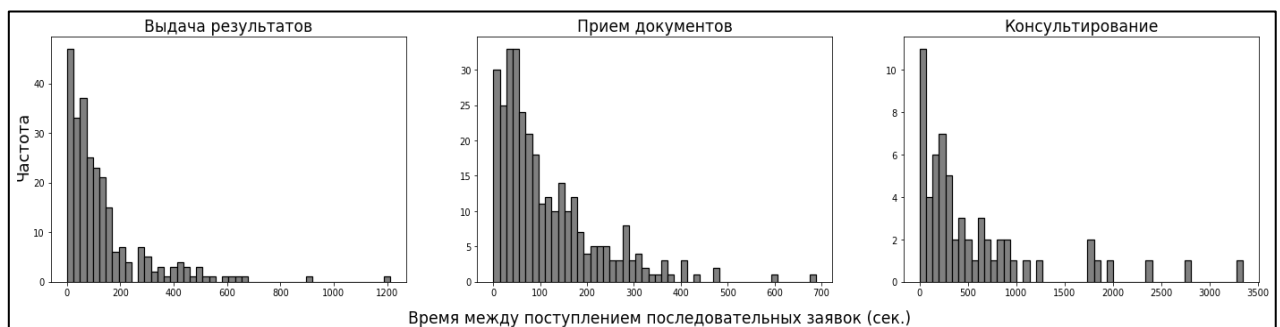


Рисунок 7 – Гистограммы распределения времени между регистрацией последовательных талонов для каждого типа услуг

Сформулируем гипотезы для проверки выдвинутого предположения:

$$H_0: P_{\xi}(x) = \lambda e^{-\lambda x}, x > 0, \lambda > 0;$$

$$H_1: P_{\xi}(x) \neq \lambda e^{-\lambda x}, x > 0, \lambda > 0.$$

Воспользуемся непараметрическим критерием согласия Андерсона-Дарлинга, апостериорное значение статистики которого имеет вид [36]:

$$S_{\Omega}(x_{1,n}) = -n - 2 \sum_{i=1}^n \frac{2i-1}{2n} \ln(F(x_i, \theta)) + 1 - \frac{2i-1}{2n} \ln(1 - F(x_i, \theta)) \quad , \quad (2.2.1)$$

где  $n$  – объем выборки;

$x_i$  – упорядоченные по возрастанию элементы выборки;

$F(x, \theta)$  – теоретический закон распределения с вектором параметров  $\theta$ .

В пределе при справедливости нулевой гипотезы статистика  $S_{\Omega}(\xi_{1,n})$  подчиняется закону с функцией распределения  $a_2(s)$  [17].

Результаты проверки гипотезы об экспоненциальном законе распределения времени между регистрацией последовательных талонов сведены в таблицу 8. На уровне значимости 0,05 все гипотезы принимаются. В качестве параметра распределения  $\lambda$  принимается его оценка  $\lambda = 1 / \bar{x}$ , где  $\bar{x}$  – выборочное среднее времени между регистрацией последовательных талонов (мин.).

Таблица 8 – Результаты проверки гипотез об экспоненциальном законе распределения времени между регистрацией последовательных талонов

Тип услуги	Число наблюдений	$S_{\text{набл}}$	Критическая точка	Вывод
1) Выдача результатов	301	1,221	1,338	$\xi_1 \sim \text{Exp}(0,512)$
2) Прием документов	349	0,295	1,339	$\xi_2 \sim \text{Exp}(0,611)$
3) Консультирование	41	0,694	1,326	$\xi_3 \sim \text{Exp}(0,073)$

В случае простейшего входного потока, распределенного по закону Пуассона с параметром  $\lambda$ , длительность временного интервала между заявками распределена по экспоненциальному закону с параметром  $\lambda$  [7]. Таким образом, число заявок в единицу времени распределено по закону Пуассона.

Далее проведем исследование закона распределения времени обслуживания одной заявки. Предварительный анализ гистограммы распределения времени обслуживания по одному талону за один случайным образом выбранный день (рисунок 8) также позволяет выдвинуть предположение, что для каждого типа услуги оно распределено экспоненциально.

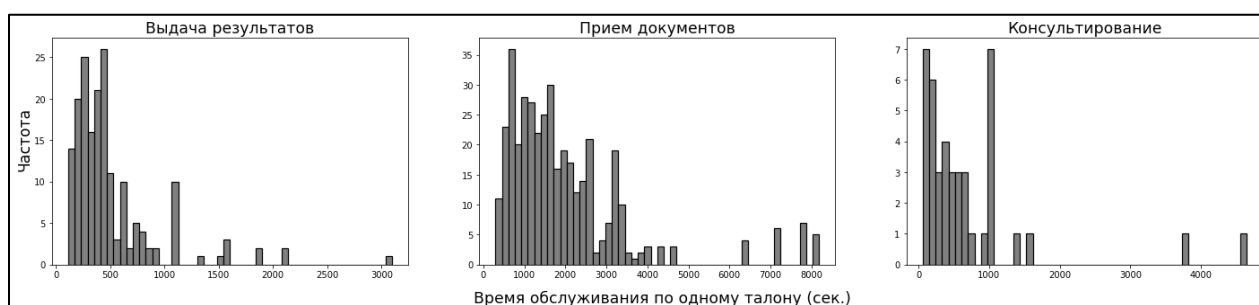


Рисунок 8 – Гистограммы распределения времени обслуживания по одному талону для каждого типа услуг



Сформулируем гипотезы для проверки выдвинутого предположения:

$$H_0: P_{\eta}(x) = \mu e^{-\mu x}, x > 0, \mu > 0;$$

$$H_1: P_{\eta}(x) \neq \mu e^{-\mu x}, x > 0, \mu > 0.$$

Результаты проверки гипотез при помощи критерия согласия Андерсона-Дарлинга сведены в таблицу 9. На уровне значимости 0,05 все гипотезы принимаются. В качестве параметра распределения  $\mu$  принимается его оценка  $\mu = \frac{1}{\bar{x}}$ , где  $\bar{x}$  – выборочное среднее времени обслуживания (мин.).

Таблица 9 – Результаты проверки гипотез об экспоненциальном законе распределения времени обслуживания одной заявки

Тип услуги	Число наблюдений	$S_{\text{набл}}$	Критическая точка	Вывод
1) Выдача результатов	301	1,307	1,338	$\eta_1 \sim \text{Exp}(0.085)$
2) Прием документов	349	1,283	1,339	$\eta_2 \sim \text{Exp}(0.035)$
3) Консультирование	41	0,827	1,326	$\eta_3 \sim \text{Exp}(0,085)$

Таким образом, проверка законов распределения входного потока и времени обслуживания показала, что для моделирования деятельности офиса №1 ГАУ «МФЦ Иркутской области» можно использовать модель марковской СМО. Дальнейшее моделирование работы МФЦ как системы массового обслуживания состоит из двух этапов: построение модели прогнозирования числа заявлений по каждому виду услуги и определение оптимального распределения окон обслуживания. Для прогнозирования числа заявлений по каждому виду услуги применяется рекуррентная нейронная сеть с двумя LSTM-слоями, архитектура которой приведена на рисунке 9.

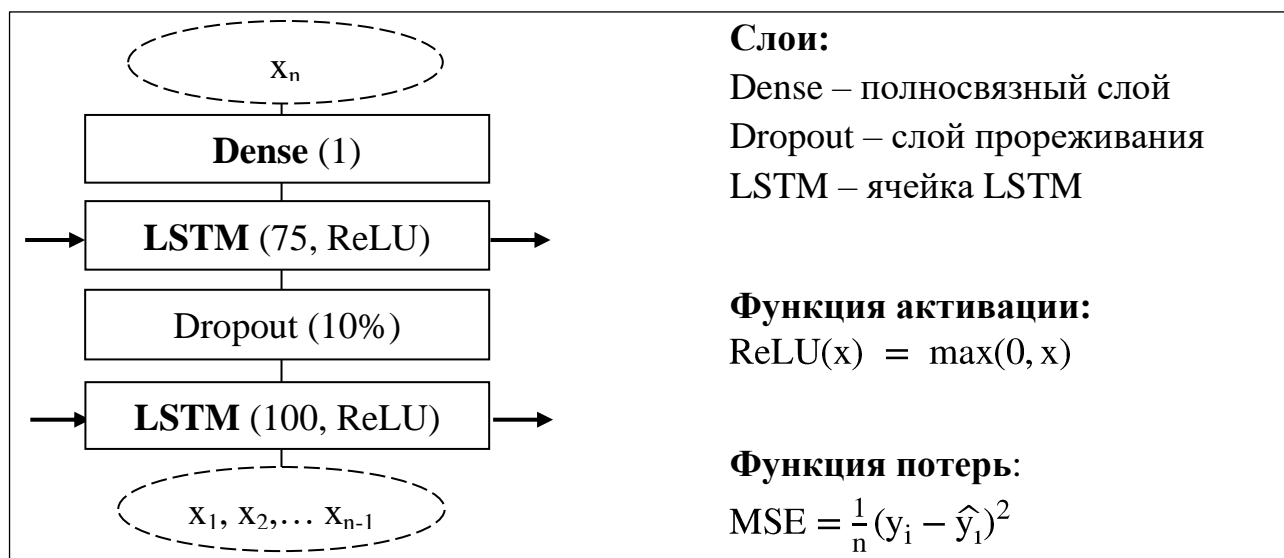


Рисунок 9 – Архитектура рекуррентной нейронной сети долгой краткосрочной памяти, используемой для прогнозирования числа заявлений

Слой Dropout используется как инструмент борьбы с переобучением модели. В качестве алгоритма оптимизации параметров нейронной сети выбран алгоритм Adaptive Moment Estimation (Adam).

Для тестирования качества моделей выборка была разбита на две части: тренировочная (327 дней) и тестовая (21 день). Результаты прогнозирования представлены на рисунке 10. На тестовой выборке были рассчитаны метрики качества прогнозирования, сведенные в таблицу 10. Также был получен прогноз на следующие 10 дней. Отметим, что для каждого временного ряда наилучшее качество обеспечивают различные размеры скользящего окна (период, на основе которого строится прогноз): 6 дней для консультирования, 9 дней для приема документов и для выдачи результатов.

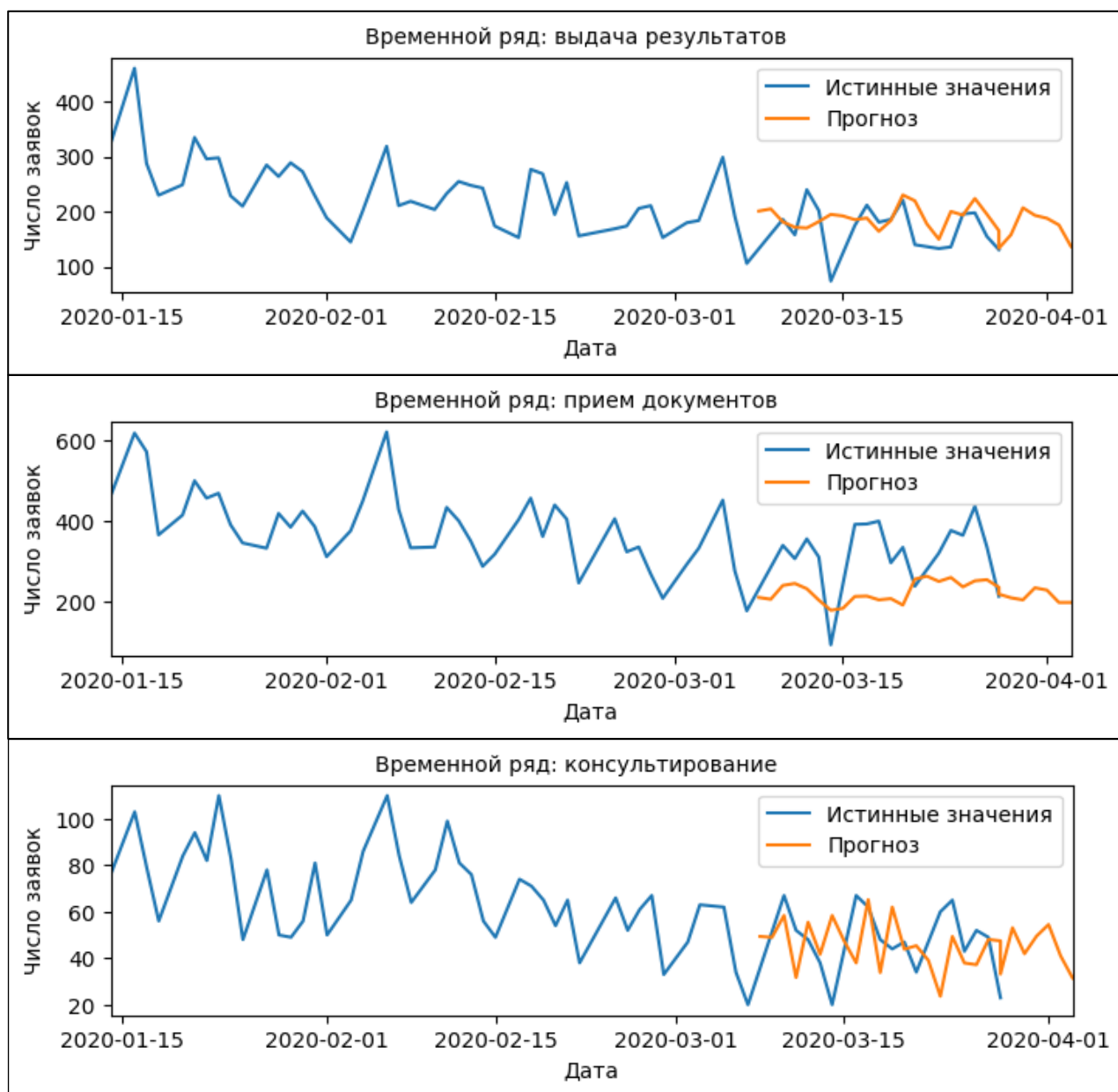


Рисунок 10 – Результаты прогнозирования числа заявлений

Таблица 10 – Метрики качества прогнозирования числа посетителей (шт.)

Временной ряд	Метрики				
	Максимальная ошибка	Средняя абсолютная ошибка	Средняя отрицательная ошибка	Средняя положительная ошибка	Корень среднеквадратической ошибки
1) Выдача результатов	63	23	23	24	28
2) Прием документов	77	24	23	24	30
3) Консультирование	18	9	11	9	11

Анализируя полученные результаты, можно сказать, что построенные модели показывают приемлемое качество. Поскольку для МФЦ важно качество обслуживания клиентов, ситуация, когда для обслуживания заявителей не хватает окон, недопустима. Чтобы предотвратить такую ситуацию, в дальнейшем к спрогнозированному числу заявлений будем добавлять значение средней абсолютной ошибки.

Вторая задача – непосредственное моделирование работы МФЦ как системы массового обслуживания. Ранее мы доказали, что исследуемая СМО является марковской. Однако стоит учесть характерную особенность МФЦ: время ожидания одной заявки (посетителя) в очереди ограничено некоторым случайным сроком  $T_{оч}$  со средним значением  $t_{оч}$ . Несмотря на то что в формальной постановке задачи ограничение на длину очереди не накладывается, очередь будем считать ограниченной 30 посетителями ( $m = 30$ ), что позволит найти распределение вероятностей состояния системы и основные характеристики в стационарном режиме [6, 23].

Согласно историческим данным, среднее предельное время ожидания в очереди  $t_{оч}$  составляет 45 минут. Таким образом, на каждую заявку, ожидающую в очереди, действует некий «поток уходов» с интенсивностью  $\nu = 1 / t_{оч}$ . Ее оценка составила  $\nu = 0,022 \text{ мин}^{-1}$ . Система электронной очереди не предусматривает учет талонов, по которым обслуживание не было произведено вследствие ухода посетителя, поэтому будем предполагать, что этот поток, как и поток входных заявок, пуассоновский.

Для расчета характеристик СМО необходимо определить интенсивности поступления заявок  $\lambda$  и интенсивности обслуживания  $\mu$ . Оценки параметров  $\lambda$  и  $\mu$  рассчитываются отдельно по каждому виду услуг.

Поступление заявок в систему МФЦ может значительно колебаться день ото дня. Поэтому для расчета интенсивности поступления заявок мы не будем использовать статистические данные за прошлый период работы МФЦ, так как они могут быть уже не актуальны, а рассчитаем их, как отношение прогнозного значения числа заявлений на один день вперед к длительности рабочего дня

(минут). Прогнозное значение числа заявлений найдем при помощи построенных ранее моделей прогнозирования.

Напротив, процедуры обслуживания заявителей значительных изменений не претерпевают, поэтому среднее время обслуживания по каждому виду услуги – относительно постоянная величина. Оценку интенсивности обслуживания найдем, как  $\mu = 1 / t_{\text{обсл}}$ , где  $t_{\text{обсл}}$  – среднее время обслуживания заявки (минут), рассчитанное по данным талонов электронной очереди за 2020 г (три месяца). Результаты расчета интенсивности обслуживания по каждому виду услуг приведены в таблице 11.

Таблица 11 – Результаты расчета оценок интенсивности обслуживания по каждому виду работы

Тип услуги	Интенсивность поступления заявок (мин <sup>-1</sup> )	Интенсивность обслуживания (мин <sup>-1</sup> )
1) Выдача результатов	$\lambda_{\text{del}} = 0,4262$	$\mu_{\text{del}} = 0,0832$
2) Прием документов	$\lambda_{\text{ser}} = 0,7774$	$\mu_{\text{ser}} = 0,0305$
3) Консультация	$\lambda_{\text{con}} = 0,1221$	$\mu_{\text{con}} = 0,0990$

Определив значения интенсивностей, мы можем перейти к непосредственному моделированию работы МФЦ как системы массового обслуживания. Перед нами стоит задача определения оптимального распределения числа окон обслуживания под прием документов, выдачу результатов и консультирование. Для этого необходимо решить, в каком режиме следует моделировать систему: стационарном или динамическом.

Для определения характеристик системы массового обслуживания в динамическом режиме нужно решить систему уравнений Колмогорова вида:

$$\begin{cases} \frac{dP(t)}{dt} = \Lambda^T P(t) \\ P(t = t_0) = P_0 \end{cases}, \quad (2.2.2)$$

где  $\Lambda = \{\lambda_{ij}\}_{i,j=0}^{n+m}$  – матрица интенсивностей перехода из состояния  $i$  в состояние  $j$ ;

$P(t)$  – вектор вероятностей состояния системы в момент времени  $t$ ;

$P_0(t_0) = (1 \ 0 \ 0 \ \dots \ 0)^T$  – вектор вероятностей состояния системы в начальный момент времени  $t_0$ .

Общим решением системы является решение вида:

$$P(t) = \sum_{i=0}^n c_i e^{\alpha_i t} \gamma_i, \quad (2.2.3)$$

где  $\alpha_i$  – собственное число матрицы  $\Lambda^T$ ;

$\gamma_i$  – соответствующее ему собственный вектор.

Для выполнения условия устойчивости по Ляпунову необходимо, чтобы все собственные числа матрицы  $\Lambda^T$  были отрицательны, кроме одного, равного нулю.

В исследуемом офисе МФЦ распределение окон имеет следующий вид: 9 окон – для выдачи результатов, 18 – для приема документов, 3 – для консультирования. Зная число окон, а также интенсивности поступления заявок и интенсивности обслуживания, определим распределение вероятностей состояний системы в динамическом режиме для каждого вида услуг. Их графики приведены на рисунке 11.

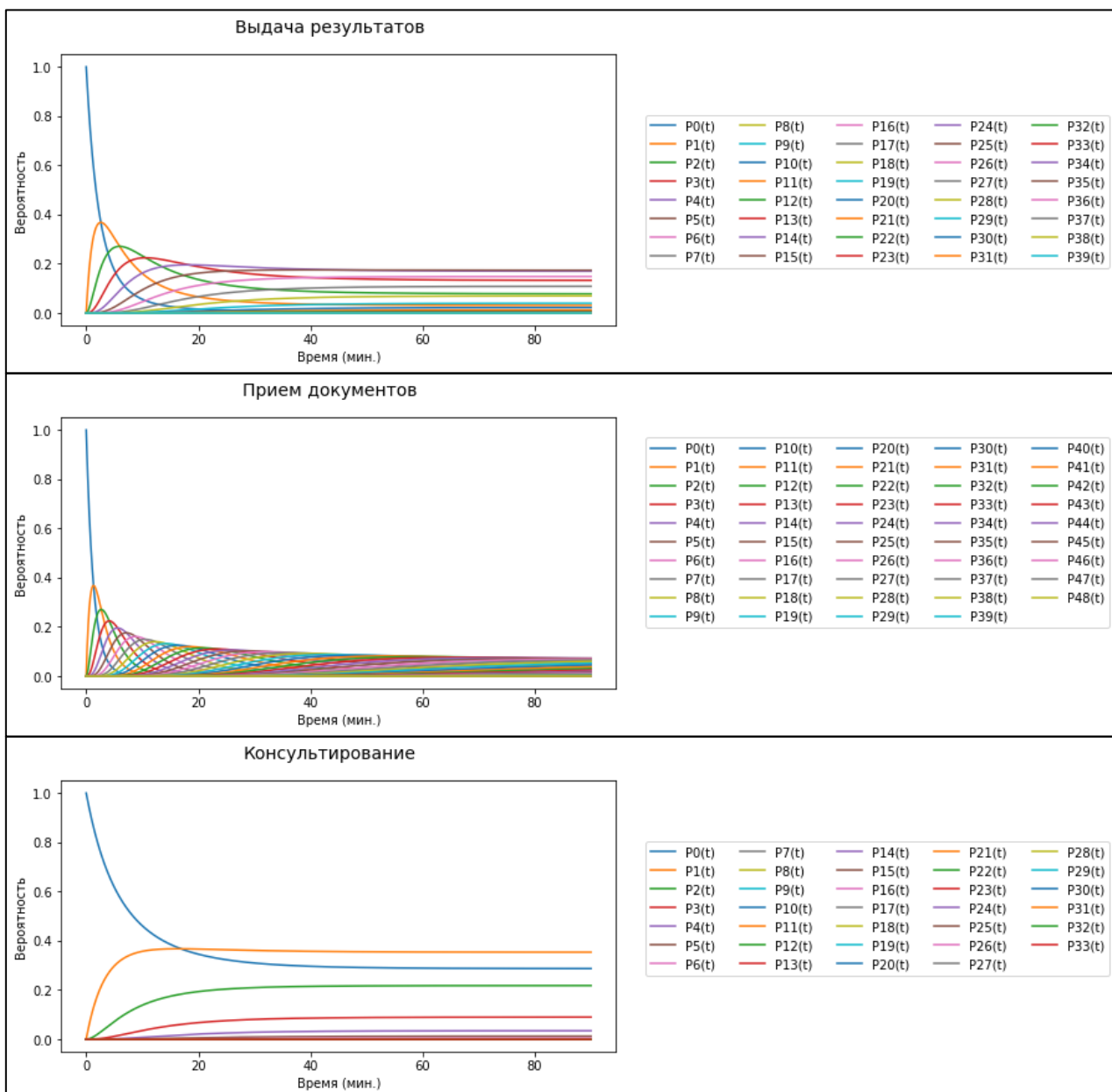


Рисунок 11 – Графики распределения вероятностей состояния системы в динамическом режиме

Анализируя графики, можно заметить, что система переходит в стацио-

нарный режим работы за достаточно короткий срок – 30-50 минут, тогда как рабочий день центра составляет 7-8 часов. Поэтому при дальнейшем моделировании работы центра будем рассматривать только стационарный режим и расчет характеристик работы МФЦ будем приводить только для этого случая.

Для определения оптимального распределения окон обслуживания под разные виды в качестве критерия оптимальности выберем среднее время пребывания посетителя в МФЦ, которое складывается из времени пребывания в очереди и времени обслуживания. Кроме того, наложим ограничения на относительную пропускную способность центра (доля обслуженных заявителей от общего числа прибывших заявителей в единицу времени). Формализуем поставленные требования в виде системы следующего вида:

$$T = T_{\text{ser}}(n_{\text{ser}}) + T_{\text{del}}(n_{\text{del}}) + T_{\text{con}}(n_{\text{con}}) \rightarrow \min, \quad \begin{cases} n_{\text{ser}} + n_{\text{del}} + n_{\text{con}} = n, \\ q_{\text{ser}} \geq q_{\text{ser}}^*, \\ q_{\text{del}} \geq q_{\text{del}}^*, \\ q_{\text{con}} \geq q_{\text{con}}^*, \\ n_{\text{ser}}, n_{\text{del}}, n_{\text{con}} > 0. \end{cases} \quad (2.2.4)$$

где  $T_i = t_{\text{ож}i} + t_{\text{обсл}i}$  – общее время, проводимое в МФЦ заявителем для получения услуги  $i$ -го типа;

$t_{\text{ож}i}, t_{\text{обсл}i}$  – время ожидания в очереди и время обслуживания заявителя при получении услуги  $i$ -го типа соответственно;

$n_i$  – число окон обслуживания, выделяемых под  $i$ -тый тип услуг;

$n$  – общее число окон обслуживания центра;

$q_i$  – относительная пропускная способность окон обслуживания по  $i$ -му типу услуг;

$q_i^*$  – нижняя граница относительной пропускной способности для услуги  $i$ -го типа;

$i \in [\text{ser}, \text{del}, \text{con}]$ ,  $\text{ser}$  – прием документов,  $\text{del}$  – выдача результатов,  $\text{con}$  – консультация.

Поскольку общее число окон обслуживания фиксировано, решить поставленную задачу можно при помощи перебора различных комбинаций  $n_{\text{del}}$ ,  $n_{\text{ser}}$  и  $n_{\text{con}}$ , удовлетворяющих требованию  $n_{\text{ser}} + n_{\text{del}} + n_{\text{con}} = n$ . Приведем псевдокод алгоритма нахождения оптимального распределения окон обслуживания.

1. Ввод:  $n$  (общее число окон обслуживания);  $m$  (число мест в очереди);  $\lambda_{\text{del}}, \lambda_{\text{ser}}, \lambda_{\text{con}}$  (интенсивности поступления заявлений по типу услуги);  $\mu_{\text{del}}, \mu_{\text{ser}}, \mu_{\text{con}}$  (интенсивности обслуживания по типу услуги);  $\nu$  (интенсивность ухода заявителей из очереди);  $q_{\text{del}}^*, q_{\text{ser}}^*, q_{\text{con}}^*$  (нижние границы относительной пропускной способности окон обслуживания).

2. Присвоить  $n_{\text{del}}^0 \leftarrow 2, n_{\text{ser}}^0 \leftarrow 5, n_{\text{con}}^0 \leftarrow 2, T^{\text{MAX}} \leftarrow 1000$ .

3. Для  $n_{\text{ser}}$  от  $n_{\text{ser}}^0$  до  $(n - n_{\text{con}}^0 - n_{\text{del}}^0 + 1)$ :

Для  $n_{con}$  от  $n_{con}^0$  до  $(n - n_{ser} - n_{del}^0 + 1)$ :

$n_{del} \leftarrow n - n_{ser} - n_{con}$ .

Создание объекта класса СМО:

$syst_{del} \leftarrow \text{QueuingSystem}(n_{del}, m, \lambda_{del}, \mu_{del}, \nu);$

$syst_{ser} \leftarrow \text{QueuingSystem}(n_{ser}, m, \lambda_{ser}, \mu_{ser}, \nu);$

$syst_{con} \leftarrow \text{QueuingSystem}(n_{del}, m, \lambda_{con}, \mu_{con}, \nu).$

Вызов метода «Найти характеристики СМО»:

$syst_{del}.\text{FindCharacteristics}();$

$syst_{ser}.\text{FindCharacteristics}();$

$syst_{con}.\text{FindCharacteristics}().$

Присвоить  $T \leftarrow syst_{del}.T + syst_{ser}.T + syst_{con}.T$ .

Если  $(T^{MAX} < T)$  и  $(syst_{del}.q < q_{del}^*)$  и  $(syst_{ser}.q < q_{ser}^*)$  и  $(syst_{con}.q < q_{con}^*)$ :

Присвоить  $T^{MAX} \leftarrow T$ ;

$n_{del}^* \leftarrow n_{del}$ ;

$n_{ser}^* \leftarrow n_{ser}$ ;

$n_{con}^* \leftarrow n_{con}$ .

4. Если  $T^{MAX} == 1000$ :

Напечатать «Оптимального распределения окон, удовлетворяющего заданным условиям, обнаружить не удалось».

Иначе:

Вывод  $n_{del}^*, n_{ser}^*, n_{con}^*$ .

Воспользуемся разработанной моделью, чтобы определить оптимальное распределение окон по конкретному виду услуг и рассчитать основные характеристики работы выбранного офиса МФЦ.

Общее число окон обслуживания в офисе  $n = 30$ , очередь будем считать ограниченной 30 посетителями ( $m = 30$ ). Система для определения оптимального распределения окон обслуживания имеет вид:

$$T = T_{ser}(n_{ser}) + T_{del}(n_{del}) + T_{con}(n_{con}) \rightarrow \min, \quad \begin{cases} n_{ser} + n_{del} + n_{con} = 30, \\ q_{ser} \geq 0,85, \\ q_{del} \geq 0,80, \\ q_{con} \geq 0,85, \\ n_{ser}, n_{del}, n_{con} > 0. \end{cases} \quad (2.2.5)$$

Решив эту систему, мы нашли оптимальное распределение окон по каждому виду услуг, после чего, воспользовавшись формулами для расчета характеристик марковской системы массового обслуживания с ограниченным временем ожидания, рассчитали характеристики системы (таблица 12). В скобках приведено сравнение результатов по сравнению с изначальным распределением окон обслуживания в многофункциональном центре (7 окон – для выдачи результатов, 21 – для приема документов, 2 – для консультирования).

Таблица 12 – Характеристики работы МФЦ как системы массового обслуживания с оптимальным распределением окон

Характеристика	Тип услуги		
	Выдача результатов	Прием документов	Консультирование
Прогнозное число заявлений (шт.)	182	349	50
Средняя абсолютная ошибка (шт.)	23	24	9
Спрогнозированное число заявлений с поправкой на ошибку (шт.)	205	373	59
Вероятность того, что клиент останется необслуженным	$1,516 \cdot 10^{-11}$	$1,379 \cdot 10^{-5}$	$6,322 \cdot 10^{-20}$
Оптимальное число окон	7 (-2)	21 (+3)	2 (-1)
Коэффициент занятости окон	0,715 (+0,149)	0,983 (+0,013)	0,573 (+0,168)
Среднее число заявителей, ожидающих в очереди (чел.)	0,533 (+0,440)	6,611 (-3,727)	0,381 (+0,302)
Абсолютная пропускная способность (чел. / час)	24,863 (-0,585)	37,829 (+4,971)	6,815 (-0,403)
Относительная пропускная способность	0,972 (-0,023)	0,811 (-0,107)	0,931 (-0,054)
Среднее время ожидания заявителя в очереди (мин.)	1,250 (+1,031)	8,503 (-4,796)	3,124 (+2,477)
Среднее время обслуживания заявителя (мин.)	11,686 (-0,274)	26,561 (-0,489)	9,398 (-0,585)
Среднее общее время пребывания заявителя в офисе МФЦ (мин.)	12,936 (+0,756)	35,065 (-5,285)	12,522 (+1,921)

Таким образом, оптимальным является следующее распределение: 5 окон обслуживания – для выдачи результатов, 23 окна обслуживания – для приема документов, 2 окна обслуживания – для консультирования. При таком распределении работа МФЦ характеризуется следующими показателями:

- наибольшая нагрузка характерна для окон обслуживания, отведенных под прием документов, наименьшая – под консультирование. Однако стоит отметить, что в целом система справляется с обслуживанием заявителей: простаивающих окон нет, а вероятность того, что клиент уйдет необслуженным, стремится к нулю;

- около 8,5 минут в среднем ожидают в очереди заявители, обратившиеся за подачей документов, 3 минуты ожидают в очереди на консультирование. Менее всего ожидают в очереди посетители, обратившиеся за получением результатов – 1,3 минуты;

- в среднем больше всего времени проводят в МФЦ заявители, обратившиеся за подачей документов (около 35 минут), в то время как получение ре-



зультатов или консультирование в среднем занимает около 12-13 минут соответственно.

Перераспределение окон обслуживания (увеличение числа окон для приема документа за счет снижения числа окон для выдачи результатов и консультирования) позволило более равномерно распределить нагрузку. Коэффициент занятости для окон выдачи результатов и консультирования вырос, что говорит об уменьшении времени их простаивания. Также перераспределение позволяет снизить время ожидания в очереди при подаче документов, при этом время ожидания очереди по другим услугам выросло незначительно.

Таким образом, в параграфе 2.2 был реализован алгоритм моделирования деятельности многофункционального центра как системы массового обслуживания. Разработанный алгоритм позволил построить прогноз числа заявлений по каждому виду услуг и найти оптимальное распределение окон обслуживания. Алгоритм является гибким: он адаптируется к изменяющимся условиям и позволяет находить оптимальное распределение окон обслуживания на основе текущей ситуации, что позволяет использовать его даже в нетипичных режимах функционирования центров.

### **2.3 Разработка рекомендательной системы для прогнозирования следующей услуги заявителя многофункционального центра**

Для МФЦ оптимизация времени, затрачиваемого на предоставление услуг заявителю, может быть достигнута за счет параллельного оказания нескольких услуг. Однако часто возникают ситуации, когда клиент не обладает сразу всей информацией о том, какие услуги ему могут понадобиться в дальнейшем. Поэтому заявителю приходится подавать несколько заявлений в МФЦ в разные дни, в то время как он мог бы подать их за одно обращение. Это приводит к повышению нагрузки на систему МФЦ: происходит увеличение численности очереди и продолжительности времени ожидания, что приводит к повышению недовольства среди клиентов центра. Чтобы избежать таких ситуаций и помочь заявителю эффективнее воспользоваться услугами МФЦ, предлагается разработать рекомендательную систему, которая на основе истории прошлых обращений заявителя будет составлять прогноз, какая услуга в дальнейшем может понадобиться заявителю.

Для построения моделей, лежащих в основе рекомендательной системы, используется информация о заявителях и их истории подачи заявлений (дата обращения, услуга) в 51 офисе МФЦ. Всего оказывается 1309 услуг. Данные представлены за 9.01.2019 – 17.05.2020 гг.

За услугами в МФЦ обращаются заявители трех типов: физические лица, юридические лица, индивидуальные предприниматели. Поскольку для каждого типа заявителей специфичны определенные услуги, на основе одного алгоритма будут обучены три модели – для каждого типа клиентов в отдельности. Для

проверки качества построенных моделей исходное множество заявителей каждого типа было разбито на тренировочное (80% заявителей) и тестовое (20% заявителей) множества.

История обращений за услугами клиентов МФЦ представляет собой последовательность транзакций, упорядоченных во времени, поэтому для составления прогноза можно воспользоваться методами прогнозирования последовательностей (sequence prediction).

Формализуем постановку задачи. Для этого дадим следующие определения [41]:

1) Конечный алфавит  $I$  – это некоторое множество уникальных объектов  $(i_1, i_2, \dots, i_N)$ , где  $N < \infty$ .

2) Последовательность  $\overline{S} = (s_1, s_2, \dots, s_n)$  – это упорядоченный список элементов алфавита  $I$ :  $\forall k = 1, n \ s_k \in I$ .

Задача прогнозирования следующего элемента последовательности заключается в предсказании элемента  $s_{n+1}$  последовательности  $\overline{S}$  на основании информации о предшествующих ему элементах  $s_1, s_2, \dots, s_n$ .

Последовательность транзакций заявлений клиентов МФЦ не является числовой: элементами последовательности являются услуги. Следовательно, классические методы прогнозирования временных рядов не могут использоваться для решения задачи прогнозирования следующей услуги. В рамках настоящей работы мы рассмотрим применение следующих методов и моделей (математические модели алгоритмов приведены в параграфе 1.3).

1. Модель компактного дерева предсказания (Compact Prediction Tree, CPT), предложенная в работе Т. Gueniche, Ph. Fournier-Viger и V.S. Tseng [41]. Несмотря на то, что к модели CPT выдвигают ряд критических замечаний: отсутствие механизмов борьбы с зашумленностью данных, отсутствие теоретического обоснования выбора максимальной длины префикса, она является одним из традиционных подходов к решению задачи прогнозирования последовательностей, показывающим хорошую точность при работе с данными различного формата. Для данной модели настраиваемым параметром является длина входной последовательности.

2. Модель рекуррентной нейронной сети долгой краткосрочной памяти (recurrent neural network with long short-term memory, RNN LSTM). Нейронные сети такой архитектуры способны хранить информацию об удаленных на большой промежуток времени зависимостях и не подвержены проблеме «исчезающего градиента» [26]. Особенность их архитектуры заключается в том, что в слоях этой сети находятся не нейроны, а так называемые «блоки памяти», которые содержат элементы, регулирующие состояние и отклик элемента [43].

Модель рекуррентной нейронной сети долгой краткосрочной памяти. Архитектура сети приведена на рисунке 12. В скобках указаны число нейронов в слое и функция активации. Поскольку задача прогнозирования следующей услуги сводится к задаче многоклассовой классификации, в качестве функции потерь принимается категориальная кросс-энтропия (1.3.26).

В качестве алгоритма оптимизации для обучения нейронной сети, выбран

алгоритм стохастической оптимизации Adaptive Moment Estimation (adam).

Перед обучением модели тренировочные и тестовые последовательности были подвергнуты процедуре One-Hot Encoding: пусть  $a = (a_1, a_2, \dots, a_m)$  – последовательность элементов  $a_i$ ,  $i = 1, m$ , тогда  $a_{i_s} = 1$ , если  $i$ -тый элемент кодирует услугу с номером  $s$  и  $a_{i_s} = 0$  в противном случае. Таким образом, каждый элемент последовательности кодируется вектором длины 1309 (по числу услуг, оказываемых в МФЦ).

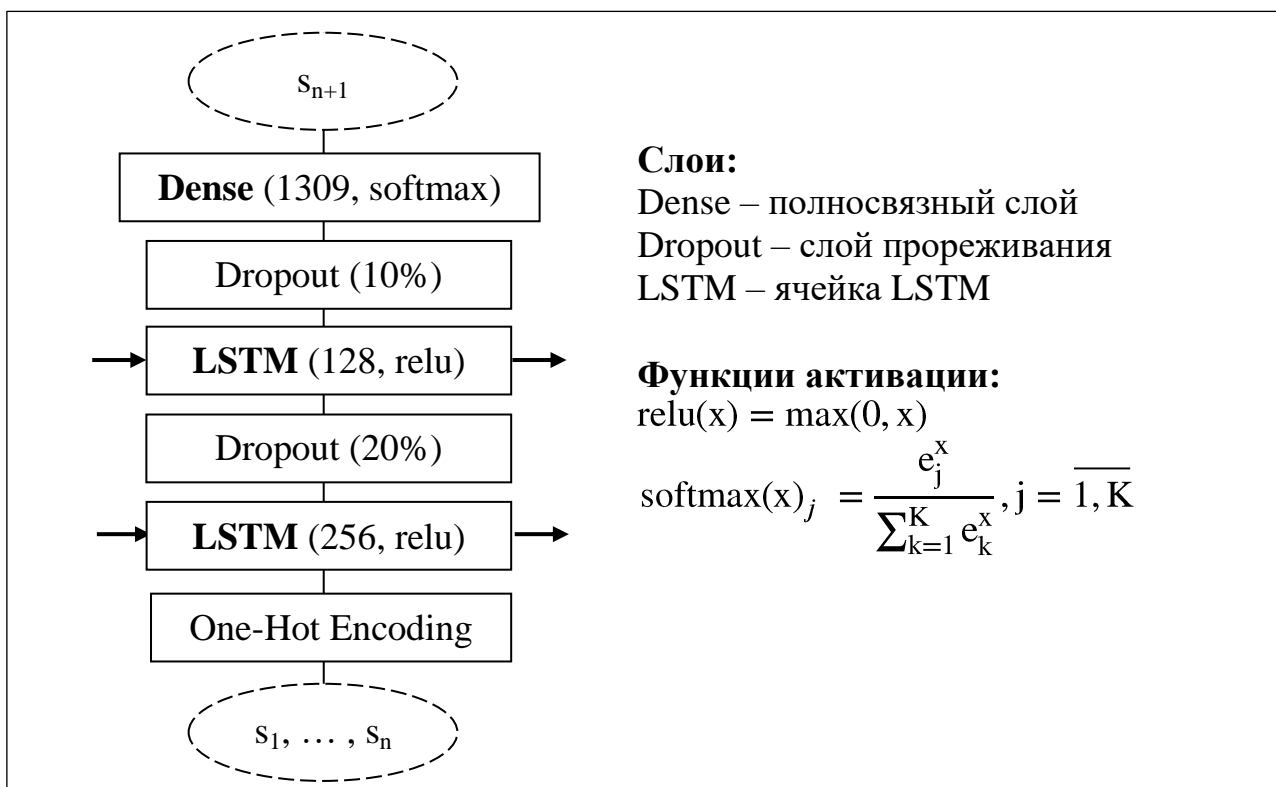


Рисунок 12 – Архитектура нейронной сети, использованной для прогнозирования следующей услуги заявителя

3. Алгоритм градиентного бустинга на деревьях принятия решений CatBoost, являющийся разработкой компании «Яндекс» и впервые представленный в работе [48] в 2018 г. Отличительная особенность алгоритма CatBoost от других видов алгоритм градиентного бустинга – способность работы с категориальными признаками без их предварительной кодировки, поэтому алгоритм CatBoost наилучшим образом подходит для решения задачи прогнозирования услуг заявителей.

Бустинг деревьев принятия решений применяют для решения задач регрессии или классификации, поэтому для решения задачи прогнозирования элементов последовательности конечного алфавита ее необходимо свести ее к задаче многоклассовой классификации.

Модель для каждого типа заявителей имеет следующие параметры:

а) Для физических лиц (ФЛ): число деревьев – 150, коэффициент скоро-

сти обучения – 0,05, глубина деревьев – 4.

б) Для юридических лиц (ЮЛ): число деревьев – 300, коэффициент скорости обучения – 0,08, глубина деревьев – 4.

в) Для индивидуальных предпринимателей (ИП): число деревьев – 150, коэффициент скорости обучения – 0,1, глубина деревьев – 4.

Для обучения моделей отбирались заявители, число заявлений которых не меньше числа  $n_{input}$  (длина входной последовательности – префикс), определявшегося в ходе вычислительных экспериментов. Если число заявлений у заявителя из тестовой выборки превышает величину  $n_{input}$ , то для прогнозирования используется последовательность из  $n_{input}$  последних услуг. Для заявителей из тренировочной выборки с числом заявлений больше  $n_{input}$  применялась процедура скользящего окна, что позволило увеличить размер обучающей выборки и в ходе обучения учесть больше закономерностей в транзакциях клиентов. Таким образом, длина префикса показывает, сколько последних обращений заявителя используются для составления прогноза о его следующей услуге.

Объемы тренировочных выборок составили:

- 148781 примеров для физических лиц;
- 116197 для юридических лиц;
- 6647 для индивидуальных предпринимателей.

Для измерения качества работы полученных моделей на отложенной тестовой выборке рассчитываются следующие метрики:

1) Categorical Accuracy (точность) – показывает долю объектов, для которых предсказан истинный класс:

$$\text{Categorical Accuracy} = \frac{TP}{\text{test size}}, \quad (2.3.1)$$

где  $TP$  – число объектов, для которых предсказанный класс совпадает с истинным ( $\hat{y}_{ij} = y_{ij}$ );

$\text{test size}$  – размер тестовой выборки.

2) Top 5 Categorical Accuracy – показывает долю объектов, для которых  $K$  предсказанных классов с наибольшей вероятностью присутствует истинный класс:

$$\text{Top } K \text{ Categorical Accuracy } (K) = \frac{TP(K)}{\text{test size}}, \quad (2.3.2)$$

где  $TP(K)$  – число объектов, для которых истинный класс входит в  $K$  наиболее вероятных предсказанных классов.

Сравнение результатов работы указанных алгоритмов приведено в таблице 13. Реализация алгоритмов проводилась на языке Python в облачной среде Google Colab с использованием аппаратного ускорителя GPU, что позволило значительно повысить скорость вычислений.

Таблица 13 – Сравнение результатов прогнозирования услуги заявителей при помощи алгоритмов CPT, RNN LSTM и CatBoost

Характеристика	Алгоритм								
	CPT			RNN LSTM			CatBoost		
	ФЛ	ЮЛ	ИП	ФЛ	ЮЛ	ИП	ФЛ	ЮЛ	ИП
Categorical Accuracy, %	17,4	39,4	23,6	35,3	70,8	35,5	31,8	71,2	32,9
Время обучения, с.	18,3	0,7	0,6	734,2	96,9	30,9	1946,2	79,2	115,3
Длина префикса	1	1	5	5	5	3	5	5	3
Общая точность, %	16,6			36,4			33,1		

Согласно полученным результатам наименьшую общую точность показал алгоритм CPT, хотя он и обеспечивает самые быстрые вычисления. Рекуррентная нейронная сеть долгой краткосрочной памяти и градиентный бустинг показали почти идентичные результаты при прогнозировании заявлений юридических лиц и индивидуальных предпринимателей, однако при прогнозировании заявлений физических лиц RNN LSTM показала точность на 3,3% выше, чем алгоритм CatBoost. Наибольшая общая точность также достигнута при применении модели нейронной сети.

Стоит отметить, что все три алгоритма справились с задачей прогнозирования следующей услуги юридических лиц лучше, чем для других типов заявителей. Это может объясняться тем, что юридические процедуры чаще осуществляются по одному, уже определенному шаблону последовательных обращений, тогда как для физических лиц и индивидуальных предпринимателей характерен более нечеткий порядок подачи заявлений.

Таким образом, наилучшие результаты показала модель рекуррентной нейронной сети долгой краткосрочной памяти. На рисунках 13-15 представлены графики динамики значений функций потерь и метрик по эпохам для модели рекуррентной нейронной сети долгой краткосрочной памяти. Оптимальное число эпох при фиксированных гиперпараметрах сети определялось по графику динамики значений функции потерь таким образом, чтобы избежать переобучения модели под тренировочную выборку. Разница в числе эпох объясняется также разницей в объемах обучающих выборок.

Модель для прогнозирования услуг физических лиц обучалась 8 эпох (рисунок 13). Итоговое значение метрики Top 5 Categorical Accuracy составило 67,5%. Таким образом, физическим лицам целесообразнее рекомендовать набор из нескольких услуг.

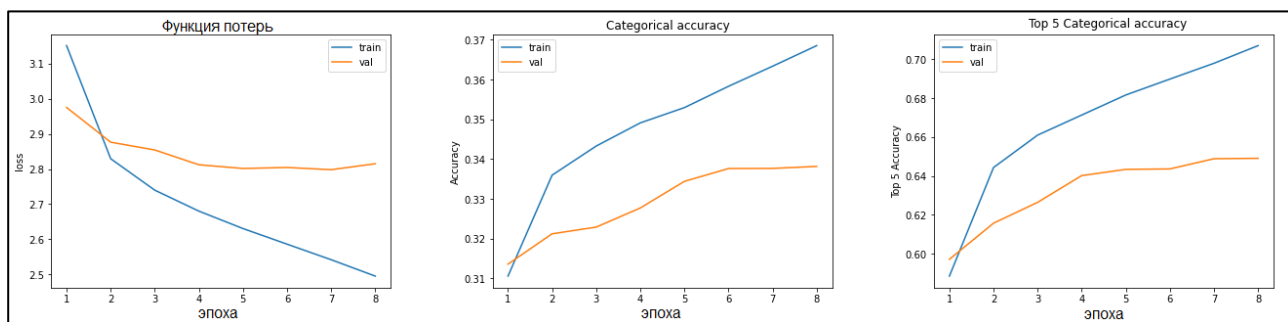


Рисунок 13 – Процесс обучения рекуррентной нейронной сети долгой краткосрочной памяти для прогнозирования услуг физических лиц

Модель для прогнозирования услуг юридических лиц обучалась 7 эпох (рисунок 14). Итоговое значение метрики Top 5 Categorical Accuracy составило 90,9%.

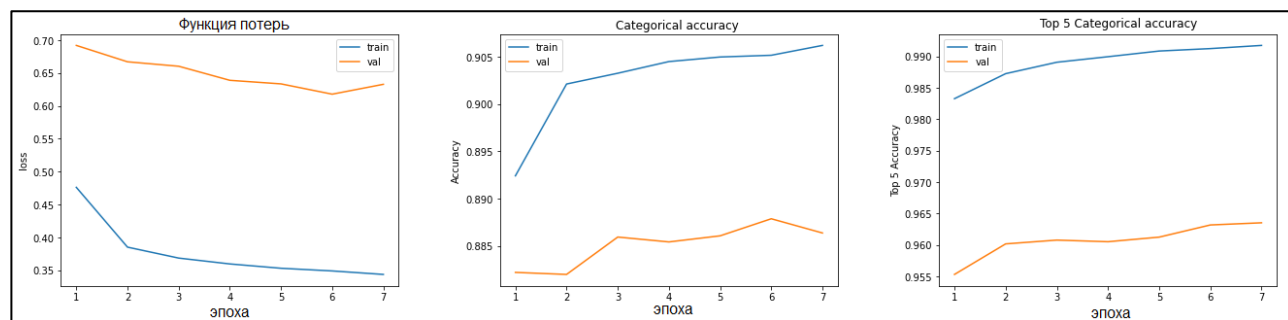


Рисунок 14 – Процесс обучения рекуррентной нейронной сети долгой краткосрочной памяти для прогнозирования услуг юридических лиц

Модель для прогнозирования услуг индивидуальных предпринимателей обучалась 20 эпох (рисунок 15). Итоговое значение метрики Top 5 Categorical Accuracy составило 67,5%.

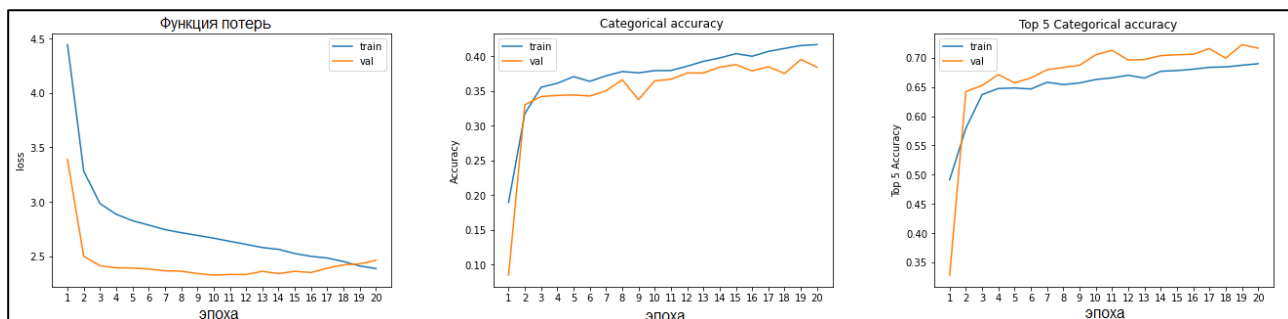


Рисунок 15 – Процесс обучения рекуррентной нейронной сети долгой краткосрочной памяти для прогнозирования услуг индивидуальных предпринимателей

Продemonстрируем работу алгоритма прогнозирования на заявителях каждого типа, для которых был получен правильный прогноз (рисунок 16).

return_true_predicts(X_test_ctb1, y_test_ctb1, y_pred_ctb1)							
	ser0	ser1	ser2	ser3	ser4	pred_ser	is_true_pred
13790	1129	278	404	1220	637	1220	True
13794	98	98	98	98	98	98	True
13795	98	98	98	783	98	98	True
13796	98	462	98	98	98	98	True
13800	1020	1020	1020	1020	1020	1020	True

a)

return_true_predicts(X_test_ctb2, y_test_ctb2, y_pred_ctb2)							
	ser0	ser1	ser2	ser3	ser4	pred_ser	is_true_pred
0	134	134	134	98	98	98	True
81	134	134	134	578	1206	134	True
82	4	4	4	4	4	4	True
83	134	134	134	134	134	134	True
84	98	98	98	98	98	98	True

б)

[ ] return_true_predicts(X_test_ctb3, y_test_ctb3, y_pred_ctb3)					
	ser0	ser1	ser2	pred_ser	is_true_pred
1	60	43	700	1306	True
2	98	134	98	98	True
11	320	320	320	1306	True
12	1304	134	134	134	True
13	43	598	1306	60	True

в)

Рисунок 16 – Демонстрация работы рекомендательной системы для: а) физических лиц; б) юридических лиц; в) индивидуальных предпринимателей

Приведем интерпретацию работы алгоритмов на примере одного заявителя из каждой группы:

а) для физического лица:

- 1) услуга №1129: Назначение и выплата единовременного пособия при рождении ребенка отдельным категориям граждан;
- 2) услуга №278: Заявление об обмене страхового свидетельства;
- 3) услуга №404: Назначение и выплата ежемесячного пособия по уходу за ребенком отдельным категориям граждан;
- 4) услуга №1220: Назначение и выплата пособия на ребенка в Иркутской области;
- 5) услуга №637: Назначение и осуществление ежемесячной выплаты в связи с рождением (усыновлением) первого ребенка;

Прогноз: услуга №1220: Назначение и выплата пособия на ребенка в Иркутской области;

б) для юридического лица:

- 1) услуга №134: Государственный кадастровый учет недвижимого имущества и (или) государственная регистрация прав на недвижимое имущество;

2) услуга №134;  
3) услуга №134;  
4) услуга № 578: Выдача градостроительного плана земельного участка, расположенного на территории города Иркутска;

5) услуга № 1206: Присвоение и аннулирование адресов на территории города Иркутска;

Прогноз: услуга №134: Государственный кадастровый учет недвижимого имущества и (или) государственная регистрация прав на недвижимое имущество;

в) для индивидуального предпринимателя:

1) услуга №60: Информирование об имуществе, включенном в перечни государственного и муниципального имущества ФЗ «О развитии малого и среднего предпринимательства в РФ»;

2) услуга №43: Информирование об участии субъектов малого и среднего предпринимательства в закупках товаров, работ;

3) услуга №700: Информирование о мерах и условиях поддержки предоставляемой субъектам малого и среднего предпринимательства;

Прогноз: услуга №1306: Предоставление информации о формах и условиях финансовой поддержки субъектов малого и среднего предпринимательства.

Таким образом, по результатам написания второй главы можно сформулировать следующие выводы:

— в результате анализа востребованности услуг МФЦ были выявлены наиболее востребованные услуги у каждого типа заявителя, определены месяцы, на которые приходится наибольшее число обращений граждан (август, октябрь и апрель) и наименьшее (январь и май), доказана неравномерность распределения числа заявлений по дням недели. В результате применения методов поиска ассоциативных правил и последовательных были выявлены закономерности в обращениях заявителей, в частности, выделена группа индивидуальных предпринимателей, оказывающих риэлтерские услуги;

— при помощи разработанной модели МФЦ как системы массового обслуживания было определено оптимальное распределение окон обслуживания под прием документов, выдачу результатов и консультирование на основе предварительного прогнозирования числа заявлений по каждому виду услуг, а также рассчитаны основные характеристики работы центра;

— реализованы три метода, позволяющие составлять прогноз о следующей услуге заявителя МФЦ. Среди них наиболее высокую точность для всей совокупности заявителей из тестовой выборки обеспечила модель рекуррентной нейронной сети долгой краткосрочной памяти. Наилучшие результаты прогноза, оцениваемые по метрикам Categorical Accuracy и Top 5 Categorical Accuracy, модель продемонстрировала для юридических лиц.



## Заключение

Результатом выполнения выпускной квалификационной работы является исследование деятельности многофункциональных центров на основе методов прикладной математики, включающее в себя несколько этапов, результатом которого является выработка рекомендаций по оптимизации деятельности многофункционального центра.

В результате предварительного анализа имеющихся данных были рассчитаны оценки основных числовых характеристик случайной величины «Число заявлений, поданных пользователем системы МФЦ за период 06.01.2019 – 18.05.2020». С помощью механизма написания запросов к базе данных были определены наиболее востребованные услуги у каждого типа заявителя. Исследование востребованности в услугах МФЦ показало, что наиболее высокая нагрузка на офисы МФЦ приходится на август, октябрь и апрель, в то время как в январе и мае наблюдается спад. Также при помощи проведения однофакторного дисперсионного анализа доказано влияние дня недели на число подаваемых услуг: пик приходится на вторник и четверг, а меньше всего офисы МФЦ загружены в понедельник, среду и субботу. Руководство центра может учесть эту информацию для составления расписания работы МФЦ и графика отпусков сотрудников центра.

Для выявления закономерностей между услугами, за которыми обращаются заявители, были применены методы поиска ассоциативных правил и последовательных шаблонов. Для каждого типа заявителей (физические лица, юридические лица и индивидуальные предприниматели) были определены собственные правила. Применение метода ассоциативных и последовательных шаблонов позволило выявить особую группу заявителей, оказывающих риэлтерские услуги в качестве индивидуальных предпринимателей. Для них специалисты центра могут разработать отдельную программу обслуживания, в которой может быть определена отдельная процедура подачи документов на оказание услуги, что позволит оптимизировать временные затраты таких заявителей и повысить удобство пользования сервисом.

Для определения оптимального режима функционирования МФЦ на примере офиса № 1 было проведено моделирование деятельности МФЦ как системы массового обслуживания. Предварительное исследование типа СМО доказало, что для моделирования работы МФЦ можно использовать СМО марковского типа с уходами из очереди. При разработке модели для учета динамичности процесса подачи заявлений была построена модель прогнозирования числа заявлений по каждому виду услуг. Построенная модель функционирования МФЦ в форме марковской системы массового обслуживания позволила определить оптимальное распределение окон обслуживания и рассчитать показатели работы офиса МФЦ. Разработанный алгоритм адаптируется к изменяющимся условиям и позволяет находить оптимальное распределение окон обслуживания на основе текущих данных о загруженности офиса МФЦ.

Моделирование деятельности МФЦ как системы массового обслуживания позволило выявить узкие места в работе центра. Так, например, при типичном распределении окон обслуживания на оказание услуг различного типа (9 окон – для выдачи результатов, 18 – для приема документов, 3 – для проведения консультаций) было выявлено, что:

- операторы, занятые приемом документов, испытывают повышенную нагрузку: коэффициент занятости близок к единице, тогда как для операторов, занимающихся выдачей результатов и консультированием, он составляет 0,566 и 0,404;

- окна обслуживания, предназначенные для выдачи документов и консультирования, имеют высокие показатели простоя: в среднем 4 и 2 окна соответственно не заняты обслуживанием клиентов;

- вследствие нехватки окон обслуживания клиенты, обратившиеся за подачей документов, ожидают в среднем 7-8 минут, в то время как для других услуг это время составляет меньше минуты.

Полученные результаты указывают на неравномерное распределение нагрузки на операторов, вследствие чего возникает необходимость оптимизации работы центра. Для проведения оптимизации был применен разработанный алгоритм динамического определения оптимального распределения окон обслуживания. Использование динамической модели позволит прогнозировать число заявлений на каждый день, опираясь на информацию о загруженности центра в предыдущий период, и в режиме реального времени определять оптимальное распределение окон обслуживания под определенные типы услуг. Так, например, при числе заявлений на выдачу результатов, прием документов и получение консультации, равном 181, 350 и 50 соответственно, под эти услуги нужно выделить 7, 21 и 2 окна. Такое распределение позволит обеспечить необходимый уровень относительной пропускной способности окон и минимизировать время пребывания заявителя в центре, а также более равномерно распределить нагрузку. Отмечается снижение времени ожидания в очереди при подаче документов, при этом время ожидания очереди по другим услугам выросло незначительно

Заключительной частью работы являлась разработка рекомендательной системы, позволяющей прогнозировать следующую услугу заявителя. В основе рекомендательной системы лежат модели компактного дерева предсказаний, рекуррентной нейронной сети долгой краткосрочной памяти и алгоритм градиентного бустинга решающих деревьев. В ходе вычислительных экспериментов было выявлено, что модель рекуррентной нейронной сети обеспечивает наиболее высокую общую точность, верно предсказывая следующую услугу для 36,4% заявителей из тестовой выборки. При этом для юридических лиц отдельно модель обеспечивает точность 70,8%.

Внедрение рекомендательной системы прогнозирования следующей услуги в информационную среду МФЦ позволит повысить качество консультирования специалистами МФЦ заявителей о порядке подачи заявлений. Также ее применение позволит заранее уведомлять заявителей о возможности подачи

документов на следующую услугу, что позволит им сократить время получения услуг и при наличии необходимых документов подать несколько заявлений за одно обращение. Такой функционал наиболее полезен для юридических лиц, поскольку для них система демонстрирует наиболее высокую точность. Использование рекомендательной системы положительно отразится на степени удовлетворенности заявителей качеством работы центра, поскольку позволит им экономить время при подготовке и подаче документов, а также параллельно подавать заявления на несколько услуг.

Стоит отметить, что разработанные модели полностью автоматизированы на языке программирования Python, что позволяет встроить их в информационную систему многофункциональных центров. При этом они не требуют сбора дополнительной информации: для их функционирования требуются только данные из внутренней базы данных МФЦ. Для определения оптимально числа окон обслуживания необходим ежедневный учет информации о загруженности центра; для корректной работы рекомендательной системы можно предложить заново обучать модель рекуррентной нейронной сети через равные интервалы времени, например, ежемесячно. Добавление новых данных позволит модели учитывать больше информации при обучении и находить более сложные и актуальные закономерности.

## Список использованных источников

1. Барсегян, А.А. Анализ данных и процессов: учеб. пособие / А.А. Барсегян, М.С. Куприянов, И.И. Холод, М.Д. Тесс, С.И. Елизаров. – СПб : БВХ-Петербург, 2009. – 512 с.
2. Бекташев, Р.А. Применение методов искусственного интеллекта для планирования персональной стратегии обучения. / Р.А. Бекташев. // Интеллектуальные системы. Теория и приложения. – 2020. – т. 24, №2. – С. 7-22.
3. Берников, М.Ю. Развитие системы многофункциональных центров предоставления государственных и муниципальных услуг / М.Ю. Берников, О.В. Чижмина // Вестник государственного и муниципального управления. – 2015. – № 3(18). – С. 57-63.
4. Будылина, Е.А. Системы массового обслуживания: марковские процессы с дискретными состояниями / Е.А. Будылина, И.А. Гарькина, Я.И. Сухов // Молодой ученый. – 2014. – № 6 (65). – С. 145-148.
5. Ваулина, К.В. Влияние неопределенности исходных данных на эффективность статистического имитационного моделирования нерелефторной системы. Часть 1. Тестовая сим-модель / К.В. Ваулина, О.Н. Маслов // Инфокоммуникационные технологии. – 2016. – т.14, №4. – С. 390-405.
6. Вентцель, Е.С. Исследование операций. / Е.С. Вентцель. // М. : «Советское радио», 1972. – 552 с.
7. Волков, И.К. Случайные процессы: учеб. для вузов / И.К. Волков, С.М. Зуев. – М. : Изд-во МГТУ им. Н.Э. Баумана, 1999. – 448 с.
8. Гаврилова, Е.Г. Изучение принципов построения имитационной модели системы массового обслуживания на примере работы многофункционального центра / Е.Г. Гаврилова, В.С. Панченко, М.П. Маркова // Информационные технологии и автоматизация управления : материалы IX Всерос. науч.-практ. конф. студентов, аспирантов, работников образования и пром-сти, 17-19 мая 2018 года. – Омск : Изд-во ОмГТУ, 2018. – С. 20–28.
9. Есенбекова, А.Э. Имитационное моделирование как главный инструмент конструирования сложных процессов и систем / А.Э. Есенбекова, Л.К. Джумахметова, С.М. Дусталиева // Технические науки в России и за рубежом : материалы VII Междунар. науч. конф. (г. Москва, ноябрь 2017 г.). – М. : Буки-Веди, 2017. – С. 165-167.
10. Жихарев, А.Г. Системно-объектное имитационное моделирование систем массового обслуживания на примере «Многофункционального центра предоставления государственных и муниципальных услуг» / А.Г. Жихарев, Д.П. Шарапов. // Научный результат. Информационные технологии. – 2017. – т.2, №1. – С. 48-54.
11. Заполева, Е.О. Развитие системы многофункциональных центров предоставления государственных и муниципальных услуг населению (на примере Приморского края) / Е.О. Заполева, А.А. Захарченко, В.В. Пить, Е.С. Мудрик // Вестник Российского университета дружбы народов. – 2016. –

№3. – С. 24-30.

12. Исупова, И.Н. Многофункциональные центры как основной механизм повышения эффективности предоставления государственных услуг населению в России / И.Н. Исупова // Общество: политика, экономика, право. – 2009. – №4. – С. 12-18.

13. Кисляков, А.Н. Анализ показателей эффективности деятельности многофункциональных центров как системы массового обслуживания / А.Н. Кисляков // Ученые записки. – 2016. – №3(19). – С. 48-52.

14. Климовских, Н.В. Особенности организации предоставления государственных и муниципальных услуг в многофункциональных центрах / Н.В. Климовских, А.А. Ратова, Е.С. Агаркова // Экономика и бизнес: теория и практика. – №11. – 2018. – С. 137-140.

15. Ковалев, С.С. Построение моделей процесса предоставления государственных и муниципальных услуг / С.С. Ковалев // Сборник докладов всероссийской научно-практической конференции молодых ученых с международным участием – Пермь : Изд-во Пермского государственного национального исследовательского университета, 2016. – №5. – С. 219-224.

16. Лабусов, М.В. Нейронные сети долгой краткосрочной памяти и их использование для моделирования финансовых временных рядов. / М.В. Лабусов // Инновации и инвестиции. – 2020. – № 3. – С. 167-171.

17. Лемешко, Б.Ю. Статистический анализ данных, моделирование и исследование вероятностных закономерностей. Компьютерный подход : монография / Б.Ю. Лемешко, С.Б. Лемешко, С.Н. Постовалов, Е.В. Чимитова. – Новосибирск : Изд-во НГТУ, 2011. – 888 с.

18. Ляхов, В.П. Региональный опыт реализации МФЦ в России в контексте оптимизации функционирования местного самоуправления и модернизации российского общества / В.П. Ляхов // Власть. – 2015. – №1. – С. 44-49.

19. Малько, В.А. Многофункциональные центры как технология повышения качества и доступности предоставления государственных и муниципальных услуг. / В.А. Малько, Т.Г. Голубева. // Научное сообщество студентов XXI столетия. Общественные науки. – 2014. – № 4(19). – С. 65-74.

20. Мироненко, Н.В. Многофункциональные центры предоставления государственных и муниципальных услуг в регионах России: проблемы и перспективы / Н.В. Мироненко, Н.С. Тихенькая, К.С. Кондакова // Государственная политика. – 2014. – № 2/1(12). – С. 105-109.

21. Михайлова, Э.А. Анализ бизнес-процессов предоставления государственных и муниципальных услуг в электронном виде (на примере работы МФЦ) / Э.А. Михайлова, С.Н. Кошачова // Вестник Рыбинской государственной авиационной технологической академии им. П.А. Соловьева. – 2015. – №1(32). – С. 229-237.

22. Об утверждении Правил организации деятельности многофункциональных центров предоставления государственных и муниципальных услуг: постановление Правительства Российской Федерации [Электронный ресурс].: постановление Правительства Рос. Федерации от 22.12.2012 №1376 // Гарант :

справочная правовая система, 2021. – Режим доступа : <http://www.garant.ru/>. – 10.10.2020.

23. Осипов, Г.С. Системы массового обслуживания с ограниченной длительностью ожидания / Г.С. Осипов // Бюллетень науки и практики. – 2016. – №12. – С. 28-36.

24. Паклин, Н.Б. Последовательные шаблоны в банковском директ-маркетинге / Н.Б. Паклин, С.В. Уланов // TERRA ECONOMICUS. – 2009. – т.7, № 2(3). – С. 99-103.

25. Пальмов, С.В. Алгоритм поиска ассоциативных правил FP-Growth / С.В. Пальмов, Е.Н. Французова // Национальная ассоциация ученых (НАУ). – 2016. – № 10 (26). – С. 27-28.

26. Пустынный, Я.Н. Решение проблемы исчезающего градиента с помощью нейронных сетей долгой краткосрочной памяти / Я.Н. Пустынный // Инновации и инвестиции. – 2020. – № 2. – С. 130-132.

27. Савинов, Ю.Г. Оптимизация в СМО с нетерпеливыми заявками. / Ю.Г. Савинов, Е.Д. Табакова, И.Д. Сафиуллов. // Ученые записки УлГУ. Сер. Математика и информационные технологии. – 2019. – №1. – С. 92-98.

28. Сутягина, Н.И. Моделирование деятельности многофункционального центра как системы массового обслуживания / Н.И. Сутягина // Карельский научный журнал. – 2015. – №1(10) – С. 199-202.

29. Филатова, А.И. Принцип одного окна как основа предоставления государственных и муниципальных услуг / А.И. Филатова, М.Н. Чумакова // Научные труды Московского гуманитарного университета. – 2018. – №4. – С. 133-138.

30. Филатова, М.М. Система оказания государственных муниципальных услуг в многофункциональном центре / М.М. Филатова // Научные труды Московского гуманитарного университета. – 2019. – №2. – С. 51-56.

31. Фирсова, Е.Г. Многофункциональные центры государственных и муниципальных услуг, их роль и правила организации деятельности / Е.Г. Фирсова // Проблемы и перспективы экономики и управления : материалы III Междунар. науч. конф. (г. Санкт-Петербург, декабрь 2014 г.). – Санкт-Петербург : Заневская площадь, 2014. – С. 46-48.

32. Фурин, А.Г. Государственные услуги в «одном окне»: многофункциональный центр как институт предоставления государственных услуг / А.Г. Фурин // Российское предпринимательство. – 2009. – № 6(2). – С. 43-50.

33. Чуев, А.В. Расширение концепции ООО-модели для систем массового обслуживания на примере многофункционального центра предоставления государственных и муниципальных услуг / А.В. Чуев, С.А. Юдицкий, В.З. Магергут // Научные ведомости. Серия История. Политология. Экономика. Информатика. – 2015. – №1 (198). – С. 85-93.

34. Яковлева, Н.А. Многофункциональный центр: новая модель взаимодействия государства и граждан при предоставлении государственных и муниципальных услуг / Н.А. Яковлева // Финансовая газета. Региональный выпуск. – 2015. – №2. – С. 136-139.

35. Agrawal, R. Fast Algorithms for Mining Association Rules in Large Databases / R. Agrawal, R. Srikant // Proceedings of the 20th International Conference on Very Large Data Bases, Santiago de Chile, 12-15 September. – 1994. – Pp. 487-499.
36. Anderson, T.W. Asymptotic theory of certain «goodness of fit» criteria based on stochastic processes / T.W. Anderson, D.A. Darling // Ann. Math. Statist. – 1952. – Vol. 23. – Pp. 193-212.
37. Ayvaz, B. Determination of Association Rules with Market Basket Analysis: Application in the Retail Sector / B. Ayvaz // Southeast Europe Journal of Soft Computing. – 2018. – Vol.7.. – Pp. 10-19.
38. Begleiter, R. On Prediction Using Variable Order Markov Models / R. Begleiter, E. Ran, Y. Golan // Journal Artificial Intelligence Research. – 2004. – Vol. 22. – Pp. 385-421.
39. Berka, P. Machine Learning and Association Rules / P. Berka, J. Rauch // Prague: University of Economics, 2010. – 30 p.
40. Chen T. XGBoost: A scalable tree boosting system / T. Chen, C. Guestrin // In Proceeding of the 22<sup>nd</sup> ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. – 2016. – Pp. 785-794.
41. Gueniche, T. Compact Prediction Tree: A Lossless Model for Accurate Sequence Prediction / T. Gueniche, P. Viger // 9th International Conference on Advanced Data Mining and Applications. – 2013. – Pp.122-134.
42. Hastie, T. The Elements of Statistical Learning: Data Mining, Inference, and Prediction / T. Hastie, R. Tibshirani, J. Friedman // Springer Series in Statistics, 2017. – 764 p.
43. Hochreiter, S. Long Short-term Memory / S. Hochreiter, J. Schmidhuber // Neural Computation. – 1997. – Vol. 9(8). – Pp.1735-1780.
44. Ke, G. LightGBM: A Highly Efficient Gradient Boosting Decision Tree / G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, T. Liu // In Proceeding of the 31st Conference on Neural Information Processing Systems, Long Beach, CA, USA, 2017. – 9 p.
45. Kingma, D.P. Adam: a method for stochastic optimization / D.P. Kingma, J.L. Ba // ICLR, 2015. – 15 p.
46. Mooney, C.H. Sequential Pattern Mining: Approaches and Algorithms / C.H. Mooney // ACM Computing Surveys. – 2013. – Vol. 45(2). – Pp. 146-192.
47. Natekin, A. Gradient Boosting Machines, A Tutorial / A. Natekin, A. Knoll // Frontiers in neurorobotics. – 2013. – Vol. 7. – Pp. 221-242.
48. Prokhorenkova, L. CatBoost: unbiased boosting with categorical features / L. Prokhorenkova, G. Gusev, A. Vorobev, A.-V. Dorogush, A. Gulin // 32nd Conference on Neural Information Processing Systems. Montréal, Canada. – 2018. – Pp. 1093-1124.
49. Sato, I. Evaluation of Machine Learning Methods on SPiCe / I. Sato, K. Chubachi, Diptarama // Proceedings of The 13th International Conference on Grammatical Inference, PMLR 57. – 2017. – Pp. 149-153.
50. Shibata, Ch. Predicting Sequential Data with LSTMs Augmented with Strictly 2-Piecewise Input Vectors / Ch. Shibata, J. Heinz // Proceedings of The 13th

International Conference on Grammatical Inference. – 2017. – Pp. 137-142.

51. Zaki, M.J. SPADE: An Efficient Algorithm for Mining Frequent Sequences / M.J. Zaki // Machine Learning. – 2001. – Vol. 42. – Pp. 31–60.

52. Zaki, M.J. Data Mining and Machine Learning: Fundamental Concepts and Algorithms / M.J. Zaki, W. Meira Jr. // Cambridge University Press, 2020. – 776 p.

53. Zhao, Y. Sequence Prediction Using Neural Network Classifiers / Y. Zhao, S. Chu, Y. Zhou, K. Tu // Proceedings of The 13th International Conference on Grammatical Inference. – 2017. – Pp. 164-169.



## Приложение А (обязательное)

### Исходные данные (фрагменты)

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
2019-01-06 15:55:33.527000000	7184765	43	193686	134	6559747	5639097	6559747	10	1853152	6559747	SERVICE	316134	0	48333290	06.01.2019	FEDERAL	06.01.2019	-06.16:41:09.367000000		
2019-01-06 16:42:30.203000000	7184765	46880280	193686	491	6559747	5639098	6559747	10	5	6559747	SERVICE	316160	0		06.01.2019	FEDERAL	06.01.2019	-06.16:49:07.493000000		
2019-01-06 17:40:47.163000000	4579720	43	193686	815	6559747	5639099	6559747	10	5	6559747	SERVICE	49801718	0	329298	06.01.2019	FEDERAL	06.01.2019	-06.17:42:28.270000000		
2019-01-08 14:24:48.943000000	7184765	43	193686	134	6559747	5639100	6559747	10	1853152	6559747	SERVICE	316134	0	48333290	08.01.2019	FEDERAL	08.01.2019	-08.15:17:09.787000000		
2019-01-08 15:19:51.693000000	7184765	46880280	193686	491	6559747	5639101	6559747	10	5	6559747	SERVICE	316160	0		08.01.2019	FEDERAL	08.01.2019	-08.15:20:31.417000000		
2019-01-08 15:28:17.503000000	4579720	43	193686	815	6559747	5639102	6559747	10	5	6559747	SERVICE	49801718	0	329298	08.01.2019	FEDERAL	08.01.2019	-08.15:29:53.653000000		
2019-01-09 08:42:55.627000000	184297601	43	16732367	234	15224350	5639103	15224350	10	1853152	15224350	SERVICE	49801718	0	331384	16.01.2019	FEDERAL	16.01.2019	-16.17:09:04.337000000		
2019-01-09 08:51:48.840000000	184297930	43	16732367	234	15224350	5639104	15224350	10	1853152	15224350	SERVICE	49801718	0	331384	17.01.2019	FEDERAL	17.01.2019	-17.16:59:09.297000000		
2019-01-09 08:57:43.967000000	16760150	2266851	149837815	524	1788235	5639105	1788235	10	5	1788235	SERVICE	316160	0		09.01.2019	FEDERAL	09.01.2019	-09.08:57:58.920000000		
2019-01-09 09:02:26.490000000	50274725	43	73486914	234	19136693	5639106	19136693	10	5	19136693	SERVICE	49801718	0	331384	10.01.2019	FEDERAL	2019-01-10:17:12.773000000			
2019-01-09 09:02:32.167000000	126456305	43	93996850	134	14857798	5639107	14857798	10	5	14857798	CONSULTATION	316134	0	48333290		FEDERAL	2019-01-09:09:02:32.453000000			
2019-01-09 09:02:56.897000000	106828248	43	45792762	134	8433794	5639108	8433794	10	5	8433794	CONSULTATION	316134	0	48333290		FEDERAL	2019-01-09:09:02:57.183000000			
2019-01-09 09:03:35.287000000	76537188	43	66838572	4	1956656	5639109	1956656	10	5	1956656	CONSULTATION	9315713	0	48333336		FEDERAL	2019-01-09:09:03:35.573000000			
2019-01-09 09:03:37.870000000	83029018	43	22835182	222	332483	5639110	332483	10	5	332483	CONSULTATION	316164	0	75331341		REGIONAL	2019-01-09:09:03:38.157000000			
2019-01-09 09:03:42.997000000	44663476	2266851	3165876	524	1015667	5639111	1015667	10	5	1015667	SERVICE	316160	0		09.01.2019	FEDERAL	09.01.2019	-09.09:04:18.593000000		
2019-01-09 09:03:52.870000000	184299299	2266851	15499767	524	10179007	5639112	10179007	10	5	10179007	SERVICE	316160	0		09.01.2019	FEDERAL	09.01.2019	-09.09:04:20.680000000		
2019-01-09 09:04:03.257000000	76537188	43	33788239	603	8433794	5639113	8433794	10	5	8433794	CONSULTATION	49801718	0	40572355		FEDERAL	2019-01-09:09:04:03.540000000			
2019-01-09 09:04:10.307000000	23596674	43	16365697	412	1023229	5639114	1023229	10	5	1023229	CONSULTATION	49801718	0	50285368		FEDERAL	2019-01-09:09:05:18.910000000			
2019-01-09 09:04:21.387000000	137180947	43	40482419	603	10259743	5639115	10259743	10	5	10259743	CONSULTATION	49801718	0	40572355		FEDERAL	2019-01-09:09:04:21.670000000			
2019-01-09 09:04:21.433000000	168424181	2266851	16268503	524	10258016	5639116	10258016	10	5	10258016	SERVICE	316160	0		09.01.2019	FEDERAL	09.01.2019	-09.09:04:35.703000000		
2019-01-09 09:04:45.123000000	6111034	2266851	4574137	524	4396466	5639117	4396466	10	5	4396466	SERVICE	316160	0		09.01.2019	FEDERAL	09.01.2019	-09.09:06:57.053000000		
2019-01-09 09:04:55.597000000	44663476	2266851	3165876	907	1015667	5639118	1015667	10	5	1015667	SERVICE	316160	0		09.01.2019	FEDERAL	09.01.2019	-09.09:07:59.830000000		
2019-01-09 09:04:57.843000000	19523101	65129676	19639454	1169	1788235	5639119	1788235	10	1853152	1788235	SERVICE	49801718	0		19.03.2019	FEDERAL	19.03.2019	-19.10:08:25.490000000		
2019-01-09 09:04:58.550000000	168424181	2266851	16268503	1258	10258816	5639120	10258816	10	5	10258816	SERVICE	316160	0		09.01.2019	FEDERAL	09.01.2019	-09.09:06:41.407000000		

Рисунок А.1 – Фрагмент таблицы с данными о заявлениях

A	B	C	D	E	F
requester_type	id	gender	edit_timestamp	deleted	birth_year
1	511	2	2017-07-19 17:16:40.1330000000	0	1956
1	512	1	2018-01-01 04:17:18.6170000000	0	1990
3	626	1	2015-05-23 18:08:26.8470000000	0	1985
2	813		2015-11-16 15:47:04.0300000000	0	
1	1023	1	2015-04-13 11:39:28.2070000000	0	1987
1	1037	2	2018-12-26 10:51:01.8630000000	0	1980
1	1261	1	2014-01-29 16:00:15.7000000000	0	1980
1	1651	2	2019-05-17 15:28:08.2500000000	0	1990
1	2981	1	2015-10-05 11:15:27.3770000000	0	1986
1	5432	2	2019-09-13 12:36:23.3730000000	0	1988
1	5433	2	2013-10-08 03:06:56.1670000000	0	
1	5434	2	2020-02-15 14:24:03.1100000000	0	1979
1	5435	2	2013-10-08 03:06:56.1670000000	0	
1	5436	2	2017-09-27 09:28:43.9700000000	0	1982
1	5437	2	2013-10-08 03:06:56.1670000000	0	
1	5438	2	2015-10-05 13:52:47.8670000000	0	1980
1	5439	1	2019-02-21 13:26:53.3370000000	0	1960
1	5440	2	2018-01-01 05:19:06.9330000000	0	1943
1	5441	2	2014-12-24 09:40:17.9100000000	0	2012
1	5442	1	2013-10-08 03:06:56.1670000000	0	
1	5443	1	2013-10-08 03:06:56.1670000000	0	
1	5444	2	2013-10-08 03:06:56.1670000000	0	
1	5445	2	2017-07-25 11:52:41.5270000000	0	1935

Рисунок А.2 – Фрагмент таблицы с данными о заявителях

A	B	C	D	E
id	ticket_number	serving_started	serving_finished	service_name
0000232c-39e0-4f2e-92c5-03082127eedc	D74	2019-10-03 04:20:16.070000000	2019-10-03 04:48:15.293000000	Регистрация права собственности (купля-продажа, дарение, аренда)
00004031-0c1d-493d-af50-70019ba843fd	31	2020-04-11 01:01:26.423000000	2020-04-11 01:33:04.580000000	Приём документов
0000616f-87b4-45cc-938f-d531a4c1d92d	3105	2020-04-14 10:40:43.183000000	2020-04-14 11:56:55.740000000	Материнский (семейный) капитал
00006c7c-e94b-415e-b803-e1a9b9d3a4dc	D86	2020-01-29 05:29:33.120000000	2020-01-29 05:53:46.467000000	Регистрация права собственности (купля-продажа, дарение, аренда)
0000b58e-d90c-4ac2-beb1-9b9fa273b3d	O20	2020-02-26 08:33:42.563000000	2020-02-26 08:38:07.657000000	Консультация
0000be8d-01ac-4bf3-862e-381ff0a79487	B128	2020-05-15 10:00:27.863000000	2020-05-15 10:07:33.363000000	Выдача прочих документов
0000cb86-81d6-4cee-9b52-052ba7c57a2e	P11	2020-03-19 03:29:50.470000000	2020-03-19 03:37:13.873000000	Приём документов
0000f11b-9ba1-4330-8470-1d92e0e444f1	D33	2019-11-19 03:18:33.887000000	2019-11-19 03:30:42.030000000	Прочие услуги, 2 и более услуги
0000f75f-4535-4090-8537-b30165bfe768	31	2020-03-16 01:53:21.933000000	2020-03-16 01:58:49.300000000	Дополнительные платные услуги
00013b88-7219-43c5-953b-b031fd409d35	K2	2020-04-16 03:53:39.243000000	2020-04-16 04:02:22.373000000	Выдача документов
00015378-040f-4ddd-927c-b342e9790644	B86	2019-11-26 06:27:02.573000000	2019-11-26 06:38:31.263000000	Выдача
0001826e-cbc2-4d75-a3a4-59fd8c43a96c	D214	2020-01-21 08:43:40.327000000	2020-01-21 09:00:15.217000000	Регистрационный учет (прописки) Миграционный учет
0001cb54-fe53-40d6-a70c-d117894c1018	B3	2020-02-08 01:07:57.640000000	2020-02-08 01:12:57.940000000	Выдача прочих документов
0001d4e9-2ac9-41f1-adc3-3899ff7bba7e	D18	2020-02-07 02:00:30.810000000	2020-02-07 02:24:38.027000000	Регистрация права собственности (купля-продажа, дарение, аренда)
0001d96e-7346-4ebc-b703-cbc9fb697783	K109	2020-03-10 08:33:46.887000000	2020-03-10 08:41:13.227000000	Выдача документов
00023070-1616-45b9-8489-abfed094532e	P2	2020-04-27 01:42:50.880000000	2020-04-27 01:48:42.723000000	Приём документов
00025ffb-1393-4ad6-9e14-a38f1edf0be3	D121	2020-03-13 08:20:17.100000000	2020-03-13 08:53:11.880000000	Прочие услуги, 2 и более услуги
0002609f-6a4c-4484-b1c5-3e35e98ebf6a	O20	2020-01-14 08:21:10.257000000	2020-01-14 08:24:23.937000000	Консультация
000274d2-1d81-4334-9ceb-514d402125e2	K64	2020-04-07 04:47:51.157000000	2020-04-07 04:55:27.120000000	Выдача документов
00027acb-f82d-4cbb-bfdd-f5de27cbfa8b	D6	2020-02-15 01:03:31.217000000	2020-02-15 01:38:47.487000000	Регистрация права собственности (купля-продажа, дарение, аренда)
0002978d-ed57-4597-9c62-6caca8902f76	C18	2020-02-18 09:39:38.787000000	2020-02-18 09:43:50.613000000	Паспорт гражданина РФ
00029df6-5320-4b89-aa26-9f15a052cd97	P10	2020-04-30 08:07:49.530000000	2020-04-30 09:13:14.987000000	Приём документов на прочие услуги
000294b-9d48-4b5d-93ce-75d4e0300592	T13	2020-02-06 03:51:51.590000000	2020-02-06 03:57:03.007000000	Выдача паспортов и водительских удостоверений

Рисунок А.3 – Фрагмент таблицы с данными о талонах электронной очереди

A	B	C	D	E	F	G
id	equeue_ticket	cpгу_order	branch_id	type	cpгу_user	change_timestamp
2407671	1546995662524~kut	5639106	52	SERVICE	73486914	2019-01-09 09:02:26.770000000
2407672	1546995683850~irk7	5639107	45	CONSULTATION	93996850	2019-01-09 09:02:32.453000000
2407673	1546995713514~usl	5639108	30	CONSULTATION	45792762	2019-01-09 09:02:57.183000000
2407674	1546995630039~irk4	5534076	4	DELIVERY	24492403	2019-01-09 09:03:30.113000000
2407675	1546995717703~ang2	5639109	8	CONSULTATION	66838572	2019-01-09 09:03:35.573000000
2407676	1546995616767~bkl	5639110	13	CONSULTATION	22835182	2019-01-09 09:03:38.157000000
2407677	1546995681023~brt1	5639111	5	SERVICE	3165876	2019-01-09 09:03:43.287000000
2407678	1546995674953~bdb	5639112	35	SERVICE	15499767	2019-01-09 09:03:53.163000000
2407679	1546995703118~usl	5639113	30	CONSULTATION	33788239	2019-01-09 09:04:03.540000000
2407680	1546995613204~snk	5639114	10	CONSULTATION	16365697	2019-01-09 09:04:10.593000000
2407681	1546995625409~brt2	5589796	6	DELIVERY	1205419	2019-01-09 09:04:17.297000000
2407682	1546995681023~brt1	5639111	5	DELIVERY	3165876	2019-01-09 09:04:18.593000000
2407683	1546995674953~bdb	5639112	35	DELIVERY	15499767	2019-01-09 09:04:20.680000000
2407684	1546995802430~zim	5639116	40	SERVICE	16268503	2019-01-09 09:04:21.727000000
2407685	1546995656904~tai	5639115	48	CONSULTATION	40482419	2019-01-09 09:04:21.670000000
2407686	1546995802430~zim	5639116	40	DELIVERY	16268503	2019-01-09 09:04:35.703000000
2407687	1546995748934~chr	5639117	25	SERVICE	4574137	2019-01-09 09:04:45.413000000
2407688	1546995681023~brt1	5639118	5	SERVICE	3165876	2019-01-09 09:04:55.887000000
2407689	1546995630135~uil	5639119	9	SERVICE	19839454	2019-01-09 09:04:57.933000000
2407690	1546995802430~zim	5639120	40	SERVICE	16268503	2019-01-09 09:04:59.843000000
2407691	1546995647369~tul	5606980	36	DELIVERY	45782587	2019-01-09 09:05:02.023000000

Рисунок А.4 – Фрагмент таблицы с данными об оказании услуг в рамках талона

## Приложение Б (обязательное)

### Программный код алгоритмов поиска ассоциативных правил и последовательных шаблонов на языке R

```
library(arules)
library(arulesSequences)
library(Matrix)
library(data.table)
library(stringr)
library(plyr)
library(dplyr)
library(psych)
library(tidyverse)

setwd("C:/Users/Оля/Desktop/Цифровой прорыв МФЦ")

# Функция: сколько месяцев прошло между двумя днями
elapsed_months <- function(end_date, start_date) {
  ed <- as.POSIXlt(end_date)
  sd <- as.POSIXlt(start_date)
  12 * (ed$year - sd$year) + (ed$mon - sd$mon)
}

#-----Обучающая выборка-----
train <- fread("train.csv")
colnames(train) <- c("order_date", "requester", "service", "cpgu_user", "service_title", "receipt_mfc", "order_number", "mfc", "internal_status", "external_status", "sub_department", "creation_mfc", "order_type", "department_id", "deleted", "deleter_fk", "custom_service_id", "close_date", "service_level", "issue_date", "change_timestamp")
train <- train[1:(nrow(train)-1)]

#-----Информация о заявителях-----
requester <- fread("requester.csv")
clean_requester <- select(requester, id, requester_type)
colnames(clean_requester) <- c("requester", "requester_type")
clean_requester1 <- clean_requester[requester_type == 1]
clean_requester2 <- clean_requester[requester_type == 2]
clean_requester3 <- clean_requester[requester_type == 3]

#-----АССОЦИАТИВНЫЕ ПРАВИЛА-----
#-----Удаление ненужных полей-----
clean_train <- select(train, requester, order_date, service_title)

#-----Преобразование полей-----
clean_train$order_date <- str_trunc(clean_train$order_date, width = 10, ellipsis = "")
clean_train$order_date <- as.Date(clean_train$order_date, format = "%Y-%m-%d")
clean_train$requester <- as.integer(clean_train$requester)

#-----Группировка по заявителям-----
gr_clean_train <- clean_train[order(order_date), .(order_date, service_title, .N), by = requester][N > 1] #заявители, у которых больше одной заявки
gr_clean_train <- gr_clean_train[N < 100]
gr_clean_train$N <- NULL
```

```

#-----Разбиение на подвыборки по типу заявителей-----
train1 <- merge(x = gr_clean_train, y = clean_requester1, by = "requester")
train1$requester_type <- NULL

train2 <- merge(x = gr_clean_train, y = clean_requester2, by = "requester")
train2$requester_type <- NULL

train3 <- merge(x = gr_clean_train, y = clean_requester3, by = "requester")
train3$requester_type <- NULL

#Все транзакции в 1 день
train1$order_date <- as.Date("2019-02-14",format = "%Y-%m-%d")
train2$order_date <- as.Date("2019-02-14",format = "%Y-%m-%d")
train3$order_date <- as.Date("2019-02-14",format = "%Y-%m-%d")

#-----Преобразование данных к классу транзакции-----
itemList <- ddply(gr_clean_train,c("requester","order_date"),
                  function(df1)paste(df1$service_title,
                                     collapse = ","))

itemList1 <- ddply(train1,c("requester","order_date"),
                   function(df1)paste(df1$service_title,
                                     collapse = ","))

itemList2 <- ddply(train2,c("requester","order_date"),
                   function(df1)paste(df1$service_title,
                                     collapse = ","))

itemList3 <- ddply(train3,c("requester","order_date"),
                   function(df1)paste(df1$service_title,
                                     collapse = ","))

#-----Запись новых данных в файл-----
items1 <- itemList1
items1$order_date <- NULL
colnames(items1) <- c("TID", "Items")
items1$TID <- NULL
#write.csv(items1, "items1.csv", row.names = F)

items2 <- itemList2
items2$order_date <- NULL
colnames(items2) <- c("TID", "Items")
items2$TID <- NULL
#write.csv(items2, "items2.csv", row.names = F)

items3 <- itemList3
items3$order_date <- NULL
colnames(items3) <- c("TID", "Items")
items3$TID <- NULL
#write.csv(items3, "items3.csv", row.names = F)

#в созданных файлах необходимо удалить название столбца

#-----Чтение транзакций-----
transactions1 <- arules::read.transactions(
  file = "items1.csv",
  rm.duplicates = 1,
  format = "basket",
  sep = ",",
  cols = NULL)

```

```

transactions2 <- arules::read.transactions(
  file = "items2.csv",
  rm.duplicates = 1,
  format = "basket",
  sep = ",",
  cols = NULL)

transactions3 <- arules::read.transactions(
  file = "items3.csv",
  rm.duplicates = 1,
  format = "basket",
  sep = ",",
  cols = NULL)

#-----Ассоциативные правила-----
assrules1 <- apriori(transactions1, parameter = list(support = 0.015, confidence
= 0.5, minlen = 2))
inspect(assrules1)
summary(assrules1)
inspect(sort(assrules1, by="support", decreasing=TRUE))

assrules2 <- apriori(transactions2, parameter = list(support = 0.02, confidence =
0.5, minlen = 2))
inspect(assrules2)
summary(assrules2)
inspect(sort(assrules2, by="support", decreasing=TRUE))

assrules3 <- apriori(transactions3, parameter = list(support = 0.07, confidence =
0.5, minlen = 2))
inspect(assrules3)
summary(assrules3)
inspect(sort(assrules3, by="support", decreasing=TRUE))

#-----График востребованности услуг-----
frequentItems1 <- eclat(transactions1, parameter = list(supp = 0.075, maxlen =
15)) # calculates support for frequent items
inspect(frequentItems1)

itemFrequencyPlot(transactions1, topN=10, type="relative", main="Частота появле-
ния услуги в транзакциях\n(для физических лиц)", sub = "для физических лиц", xlab =
"Номер услуги", ylab = "Поддержка") # plot frequent items

frequentItems2 <- eclat(transactions2, parameter = list(supp = 0.05, maxlen =
15)) # calculates support for frequent items
inspect(frequentItems2)

itemFrequencyPlot(transactions2, topN=10, type="relative", main="Частота появле-
ния услуги в транзакциях\n(для юридических лиц)", sub = "для физических лиц", xlab =
"Номер услуги", ylab = "Поддержка") # plot frequent items

frequentItems3 <- eclat(transactions3, parameter = list(supp = 0.12, maxlen =
15)) # calculates support for frequent items
inspect(frequentItems3)

itemFrequencyPlot(transactions3, topN=10, type="relative", main="Частота появле-
ния услуги в транзакциях\n(для индивидуальных предпринимателей)", sub = "для физических
лиц", xlab = "Номер услуги", ylab = "Поддержка") # plot frequent items
#-----ПОСЛЕДОВАТЕЛЬНЫЕ ШАБЛОНЫ-----
#-----Преобразование исходных данных к нужному виду-----

```

```

train <- select(train, requester, order_date, service_title)
train$order_date <- str_trunc(train$order_date, width = 10, ellipsis = "")
train$requester <- as.integer(train$requester)

#-----Разбиение на подвыборки для каждого типа заявителя-----
train1 <- merge(x = train, y = clean_requester1, by = "requester")
train1$requester_type <- NULL

train2 <- merge(x = train, y = clean_requester2, by = "requester")
train2$requester_type <- NULL

train3 <- merge(x = train, y = clean_requester3, by = "requester")
train3$requester_type <- NULL

# Дата, с которой начинается отсчет
start_month <- "2019-01-06"

#-----Группировка по заявителям-----
gr_train1 <- train1[order(order_date), .(order_date, service_title, .N), by = requester][N > 1] #заявители, у которых больше одной заявки
gr_train2 <- train2[order(order_date), .(order_date, service_title, .N), by = requester][N > 1] #заявители, у которых больше одной заявки
gr_train3 <- train3[order(order_date), .(order_date, service_title, .N), by = requester][N > 1] #заявители, у которых больше одной заявки

#-----Создание eventID-----
gr_train1$eventID <- elapsed_months(gr_train1$order_date, start_month)
gr_train2$eventID <- elapsed_months(gr_train2$order_date, start_month)
gr_train3$eventID <- elapsed_months(gr_train3$order_date, start_month)

gr_train1$order_date <- NULL
gr_train2$order_date <- NULL
gr_train3$order_date <- NULL

#-----Удаление дублирующихся записей-----
gr_train1 <- gr_train1[!duplicated(gr_train1)] # клиент за 1 месяц об-ратился за
1 услугой несколько раз
gr_train2 <- gr_train2[!duplicated(gr_train2)]
gr_train3 <- gr_train3[!duplicated(gr_train3)]

#-----Пересчет N-----
gr_train1 <- select(gr_train1, requester, service_title, eventID)
gr_train1 <- gr_train1[order(eventID), .( service_title, eventID, .N), by = requester]

gr_train2 <- select(gr_train2, requester, service_title, eventID)
gr_train2 <- gr_train2[order(eventID), .( service_title, eventID, .N), by = requester]

gr_train3 <- select(gr_train3, requester, service_title, eventID)
gr_train3 <- gr_train3[order(eventID), .( service_title, eventID, .N), by = requester]

#---Удаление клиентов с более чем 100 заявлениями за всю историю обращений---
transactions1_sp <- gr_train1[N < 100]
transactions1_sp$N <- NULL

transactions2_sp <- gr_train2[N < 100]
transactions2_sp$N <- NULL

transactions3_sp <- gr_train3[N < 100]

```

```

transactions3_sp$N <- NULL

#-----Создание последовательностей транзакций-----
trans_sequence1 <- transactions1_sp %>%
  group_by(requester, eventID) %>%
  dplyr::summarize(
    SIZE = n(),
    ServiceLevel = paste(as.character(service_title), collapse = ';')
  )

trans_sequence2 <- transactions2_sp %>%
  group_by(requester, eventID) %>%
  dplyr::summarize(
    SIZE = n(),
    ServiceLevel = paste(as.character(service_title), collapse = ';')
  )

trans_sequence3 <- transactions3_sp %>%
  group_by(requester, eventID) %>%
  dplyr::summarize(
    SIZE = n(),
    ServiceLevel = paste(as.character(service_title), collapse = ';')
  )

names(trans_sequence1) = c("sequenceID", "eventID", "SIZE", "items")
names(trans_sequence2) = c("sequenceID", "eventID", "SIZE", "items")
names(trans_sequence3) = c("sequenceID", "eventID", "SIZE", "items")

#-----Перевод в факторы и сортировка по sequenceID и eventID-----
trans_sequence1 <- data.frame(lapply(trans_sequence1, as.factor))
trans_sequence1 <- trans_sequence1[order(trans_sequence1$sequenceID,
trans_sequence1$eventID),]

trans_sequence2 <- data.frame(lapply(trans_sequence2, as.factor))
trans_sequence2 <- trans_sequence2[order(trans_sequence2$sequenceID,
trans_sequence2$eventID),]

trans_sequence3 <- data.frame(lapply(trans_sequence3, as.factor))
trans_sequence3 <- trans_sequence2[order(trans_sequence3$sequenceID,
trans_sequence3$eventID),]

#-----Запись в файлы и чтение в формате транзакций-----
write.table(trans_sequence1, "trans_sequence1 2 0.txt", sep=";", row.names =
FALSE, col.names = FALSE, quote = FALSE)
trans_matrix1 <- read_baskets("trans_sequence1 2 0.txt", sep = ";", info =
c("sequenceID", "eventID", "SIZE"))

write.table(trans_sequence2, "trans_sequence2 2 0.txt", sep=";", row.names =
FALSE, col.names = FALSE, quote = FALSE)
trans_matrix2 <- read_baskets("trans_sequence2 2 0.txt", sep = ";", info =
c("sequenceID", "eventID", "SIZE"))

write.table(trans_sequence3, "trans_sequence3 2 0.txt", sep=";", row.names =
FALSE, col.names = FALSE, quote = FALSE)
trans_matrix3 <- read_baskets("trans_sequence3 2 0.txt", sep = ";", info =
c("sequenceID", "eventID", "SIZE"))

#-----Преобразование eventID (д.6. положительным)-----
trans_matrix1_new <- trans_matrix1

```

```

trans_matrix1_new@itemsetInfo$eventID<-trans_matrix1_new@itemsetInfo$eventID+1

trans_matrix2_new <- trans_matrix2
trans_matrix2_new@itemsetInfo$eventID<-trans_matrix2_new@itemsetInfo$eventID+1

trans_matrix3_new <- trans_matrix3
trans_matrix3_new@itemsetInfo$eventID<-trans_matrix3_new@itemsetInfo$eventID+1

#-----Последовательные шаблоны-----
s1 <- cspade(trans_matrix1_new, parameter = list(support = 0.05), control =
list(verbose = TRUE))
summary(s1)
as(s1, "data.frame") # показать последовательности
as(ruleInduction(s1, confidence = 0.2, control = list(verbose = TRUE)), "da-
ta.frame") # показать правила

s2 <- cspade(trans_matrix2_new, parameter = list(support = 0.3), control =
list(verbose = TRUE))
summary(s2)
ruleInduction(s2, confidence = 0.2, control = list(verbose = TRUE))
as(ruleInduction(s2, confidence = 0.15, control = list(verbose = TRUE)), "da-
ta.frame")

s3 <- cspade(trans_matrix3_new, parameter = list(support = 0.1), control =
list(verbose = TRUE))
summary(s3)
as(s3, "data.frame")
as(ruleInduction(s3, confidence = 0.5, control = list(verbose = TRUE)), "da-
ta.frame")

```



## Приложение В (обязательное)

### Программный код для моделирования системы массового обслуживания на языке Python

```
# 1 ПОДКЛЮЧЕНИЕ БИБЛИОТЕК
import numpy as np
import pandas as pd
import math
import datetime
import random
import math
import matplotlib.pyplot as plt
from matplotlib.pyplot import figure
import seaborn as sns
from datetime import datetime, timedelta
from keras.models import Sequential
from keras.layers import Dense, LSTM, Dropout, RepeatVector, TimeDistributed
from keras.utils import np_utils
from sklearn.preprocessing import MinMaxScaler
from sklearn.metrics import mean_squared_error, mean_absolute_error, max_err
or
import scipy
from scipy import stats
from numpy import linalg as LA
from google.colab import drive
drive.mount('/content/drive')

# 2 ПРОВЕРКА ГИПОТЕЗ
# Функция для подготовки данных для входного потока
def prepare_data(DATA, type):
    df = DATA[DATA['type'] == type]
    df = df.astype({"type": str, "change_timestamp": str})
    df['date'] = df['change_timestamp'].str[0:10]
    df['time'] = df['change_timestamp'].str[11:19]
    df['date'] = pd.to_datetime(df['date'])
    df['time'] = pd.to_datetime(df['time'], format= '%H:%M:%S' )
    df.drop(columns=['type', 'change_timestamp'], inplace=True)
    return df

# Функция для определения числа заявлений за каждый день
def count_num_of_appl_per_day(df, crop_date=""):
    num_of_appl = df.loc[:, ['date', 'time']].groupby('date').agg('count')
    num_of_appl.rename(columns={'time' : 'N'}, inplace=True)
    if crop_date:
        num_of_appl = num_of_appl.loc[:crop_date]
    return num_of_appl

# Функция для определения времени обслуживания
def get_time_between_appls(df):
    df = df.sort_values(by=['date', 'time'])
    df['diff'] = (df['time'] -
df['time'].shift()).fillna(pd.Timedelta('0 days')).dt.total_seconds()
    df = df[df['diff'] >= 0]
    df.drop(columns=['time'], inplace=True)
```

```

return df

# Функция для подготовки данных для обслуживания
def prepare_data_timeser(data, type):
    DATA = data[data['branch_id'] == 1]
    DATA = DATA[DATA['type'] == type]
    DA-
    TA.drop(columns=['id', 'equeue_ticket', 'branch_id', 'type'], inplace=True)
    DATA['date'] = DATA['serving_started'].str[0:10]
    DATA['serving_started'] = pd.to_datetime(DATA['serving_started'])
    DATA['serving_finished'] = pd.to_datetime(DATA['serving_finished'])
    DATA['time'] = (DATA['serving_finished'] -
    DATA['serving_started']).dt.total_seconds()
    DATA['date'] = pd.to_datetime(DATA['date'])
    DATA.drop(columns=['serving_started', 'serving_finished'], inplace=True)
    DATA.sort_values(by='date', inplace=True)
    return DATA

# 2.1 ВХОДНОЙ ПОТОК
# Подготовка данных
e_queue = pd.read_csv('drive/My Drive/Данные/equeue_ticket_order_act.csv', s
ep=';', usecols=[1,3,4,6], index_col=0)
e_queue = e_queue[e_queue['branch_id'] == 1] # оставляем только первый офис
e_queue.drop(columns=['branch_id'], inplace=True)

num_of_appl = e_queue.copy()
num_of_appl['date'] = num_of_appl['change_timestamp'].str[0:10]
num_of_appl['date'] = pd.to_datetime(num_of_appl['date'])
num_of_appl.sort_values(by=['date'], inplace=True)
num_of_appl = num_of_appl.loc[:, ['date', 'change_timestamp']].groupby('date
').agg('count')
num_of_appl.rename(columns={'change_timestamp':'count'}, inplace=True)

delivery = prepare_data(e_queue, 'DELIVERY')
service = prepare_data(e_queue, 'SERVICE')
consultation = prepare_data(e_queue, 'CONSULTATION')

count_delivery = count_num_of_appl_per_day(delivery, '2020-03-28')
count_service = count_num_of_appl_per_day(service, '2020-03-28')
count_consultation = count_num_of_appl_per_day(consultation, '2020-03-28')

# График динамики временных рядов
figure(figsize=(14, 5), dpi=100)
plt.plot(num_of_appl['count'], label='Всего')
plt.plot(count_delivery['N'], label='Выдача результатов')
plt.plot(count_service['N'], label='Прием документов')
plt.plot(count_consultation['N'], label='Консультирование')
plt.ylabel('Число заявлений')
plt.xlabel('Дата')
plt.legend()

time_diff_delivery = get_time_between_appls(delivery)
time_diff_service = get_time_between_appls(service)
time_diff_consultation = get_time_between_appls(consultation)

example_del = time_diff_delivery[time_diff_delivery['date'] == pd.to_dateti
ple_del = time_diff_delivery[time_diff_delivery['date'] == pd.to_datetime("
2019-04-12")]

```

```

example_ser = time_diff_service[time_diff_service['date'] == pd.to_datetime(
    '2019-05-08')]
example_con = time_diff_consultation[time_diff_consultation['date'] == pd.to_datetime(
    "2019-04-12")]

# Гистограммы времени между последовательными заявками
fig, axs = plt.subplots(1, 3, figsize=(10,4))
plt.subplots_adjust(wspace=0.2, hspace=0.1, bottom = 0.1, top = 1, right=2)
axs[0].hist(example_del['diff'], bins=50, color='gray', edgecolor='black', linewidth=1.2)
axs[0].set_title('Выдача результатов', fontsize='xx-large')
axs[1].hist(example_ser['diff'], bins=50, color='gray', edgecolor='black', linewidth=1.2)
axs[1].set_title('Прием документов', fontsize='xx-large')
axs[2].hist(example_con['diff'], bins=50, color='gray', edgecolor='black', linewidth=1.2)
axs[2].set_title('Консультирование', fontsize='xx-large')
fig.text(1.07, -0.05, 'Время между поступлением последовательных заявок (сек.)', ha = 'center', fontsize = 'xx-large')
fig.text(0.07, 0.6, 'Частота', va='center', rotation='vertical', fontsize='xx-large')

# Проверка гипотез об экспоненциальном ЗР
ex_del = example_del['diff'][example_del['diff'] < 300]
stats.anderson(ex_del, 'expon')
stats.anderson(example_ser['diff'], 'expon')
stats.anderson(example_con['diff'], 'expon')

# 2.2 ВРЕМЯ ОБСЛУЖИВАНИЯ
tickets = pd.read_csv('drive/My Drive/Данные/equeue_ticket.csv', sep = ';', index_col = False, encoding = "cp1251", usecols = [0,2,3], error_bad_lines =False)
e_queue_tickets = pd.read_csv('drive/My Drive/Данные/equeue_ticket_order_act.csv', usecols=[0,1,3,4], sep=';', index_col=0)

tickets_full = tickets.merge(right= e_queue_tickets, left_on='id', right_on='equeue_ticket')
del tickets, e_queue_tickets

timeser_del = prepare_data_timeser(tickets_full, 'DELIVERY')
timeser_ser = prepare_data_timeser(tickets_full, 'SERVICE')
timeser_con = prepare_data_timeser(tickets_full, 'CONSULTATION')

# Гистограммы времени между последовательными заявками
fig, axs = plt.subplots(1, 3, figsize=(10,4))
plt.subplots_adjust(wspace=0.2, hspace=0.1, bottom = 0.1, top = 1, right=2)
axs[0].hist(example_time_del['time'], bins=50, color='gray', edgecolor='black', linewidth=1.2)
axs[0].set_title('Выдача результатов', fontsize='xx-large')
axs[1].hist(example_time_ser['time'], bins=50, color='gray', edgecolor='black', linewidth=1.2)
axs[1].set_title('Прием документов', fontsize='xx-large')
axs[2].hist(example_time_con['time'], bins=50, color='gray', edgecolor='black', linewidth=1.2)
axs[2].set_title('Консультирование', fontsize='xx-large')
fig.text(1.07, -0.07, 'Время обслуживания по одному талону (сек.)', ha = 'center', fontsize='xx-large')

```

```

fig.text(0.07, 0.6, 'Частота', va='center', rotation='vertical', fontsize='xx-large')

# Проверка гипотез об экспоненциальном ЗР
stats.anderson(example_time_del['time'], 'expon')
stats.anderson(example_time_ser['time'], 'expon')
stats.anderson(example_time_con['time'], 'expon')

# 2 ПРОГНОЗИРОВАНИЕ ЧИСЛА ЗАЯВОК
# функция для разбиения последовательности на обучающие наборы данных
def split_sequence(sequence, n_steps):
    X, y = list(), list()
    for i in range(len(sequence)):
        end_ix = i + n_steps
        if end_ix > len(sequence)-1:
            break
        seq_x, seq_y = sequence[i:end_ix], sequence[end_ix]
        X.append(seq_x)
        y.append(seq_y)
    return np.array(X), np.array(y)

# функция для прогнозирования
def predict(model, data, num_epochs, size_of_window, test_data_size, fut_pred):
    n_features = 1
    X, y = split_sequence(data, size_of_window)
    X = X.reshape((X.shape[0], X.shape[1], n_features))

    X_train, y_train = X[:-test_data_size], y[:-test_data_size]
    X_test, y_test = X[-test_data_size:], y[-test_data_size:]
    y_test = y_test.reshape(len(X_test), 1)
    del X, y
    model.fit(X_train, y_train, verbose=0, epochs = num_epochs, validation_split = 0.1)
    y_hat = model.predict(X_test)
    dif = y_test - y_hat
    RMSE = mean_squared_error(y_test, y_hat)**0.5
    MAE = mean_absolute_error(y_test, y_hat)
    ME = max_error(y_test, y_hat)
    MPE = sum(dif[dif>0])/len(dif[dif>0])
    MNE = sum(dif[dif<0])/len(dif[dif<0])
    Metrics = dict({'Root Mean Square Error':RMSE,
                    'Mean Absolute Error':MAE,
                    'Max Error':ME, 'Mean Positive Error': MPE,
                    'Mean Negative Error': MNE})
    y_hat = [item for sublist in y_hat for item in sublist]
    # прогнозирование на будущий период (вне выборки)
    y_fut_hat = []
    for i in range(fut_pred):
        X = np.array(data[-size_of_window:]).reshape(1, size_of_window, 1)
        y_pred = model.predict(X)[0][0]
        y_fut_hat.append(y_pred)
        data.append(y_pred)
    return y_hat, y_fut_hat, Metrics

# 2.1 ПРОГНОЗИРОВАНИЕ
count_delivery = count_num_of_appl_per_day(delivery, '2020-03-28')
count_service = count_num_of_appl_per_day(service, '2020-03-28')
count_consultation = count_num_of_appl_per_day(consultation, '2020-03-28')

```

```

list_delivery = list(count_delivery['N']) #348дней
list_service = list(count_service['N'])
list_consultation = list(count_consultation['N'])

test_data_size = 21
fut_pred = 7

time = np.arange(datetime(2020,3,29)-timedelta(days=test_data_size),
                  datetime(2020,3,29), timedelta(days=1)).astype(datetime)
time2 = np.arange(time[-1], time[-1] + timedelta(days=fut_pred), timedelta(days=1)).astype(datetime)

# ВЫДАЧА РЕЗУЛЬТАТОВ
train_window = 9

# Модель
lstm_keras = Sequential()
lstm_keras.add(LSTM(100, activation='relu', return_sequences=True, input_shape=(train_window, 1)))
lstm_keras.add(Dropout(0.1))
lstm_keras.add(LSTM(75, activation='relu'))
lstm_keras.add(Dense(1))
lstm_keras.compile(optimizer='adam', loss='mse')

preds_del, preds_fut_del, metrics_del = predict(lstm_keras, list_delivery,
epochs, train_window, test_data_size, fut_pred)

# График
figure(figsize=(8,2), dpi=100)
plt.xlabel("Дата")
plt.ylabel('Число заявок')
plt.suptitle('Временной ряд: выдача результатов', ha='center', fontsize=10)
plt.autoscale(axis='x', tight=True)
plt.plot(count_delivery['N'][-60:],label='Истинные значения')
plt.plot(list(time)+list(time2),preds_del+preds_fut_del, label='Прогноз')
plt.legend()
plt.show()

# ПРИЕМ ДОКУМЕНТОВ
train_window = 10

# Модель
lstm_keras = Sequential()
lstm_keras.add(LSTM(100, activation='relu', return_sequences=True, input_shape=(train_window, 1)))
lstm_keras.add(Dropout(0.1))
lstm_keras.add(LSTM(75, activation='relu'))
lstm_keras.add(Dense(1))
lstm_keras.compile(optimizer='adam', loss='mse')

preds_ser, preds_fut_ser, metrics_ser = predict(lstm_keras, list_service,
epochs, train_window, test_data_size, fut_pred)

# График
figure(figsize=(8,2), dpi=100)
plt.xlabel("Дата")
plt.ylabel('Число заявок')
plt.suptitle('Временной ряд: прием документов', ha='center', fontsize=10)
plt.autoscale(axis='x', tight=True)

```

```

plt.plot(count_service['N'][-60:],label='Истинные значения')
plt.plot(list(time)+list(time2),preds_ser+preds_fut_ser, label='Прогноз')
plt.legend()
plt.show()

# КОНСУЛЬТИРОВАНИЕ
train_window = 6

# Модель
lstm_keras = Sequential()
lstm_keras.add(LSTM(100, activation='relu', return_sequences=True, input_shape=(train_window, 1)))
lstm_keras.add(Dropout(0.1))
lstm_keras.add(LSTM(75, activation='relu'))
lstm_keras.add(Dense(1))
lstm_keras.compile(optimizer='adam', loss='mse')

preds_con, preds_fut_con, metrics_con = predict(lstm_keras, list_consultation, epochs, train_window, test_data_size, fut_pred)

# График
figure(figsize=(8,2), dpi=100)
plt.xlabel("Дата")
plt.ylabel('Число заявок')
plt.suptitle('Временной ряд: консультирование', ha='center', fontsize=10)
plt.autoscale(axis='x', tight=True)
plt.plot(count_consultation['N'][-60:],label='Истинные значения')
plt.plot(list(time)+list(time2),preds_con+preds_fut_con, label='Прогноз')
plt.legend()
plt.show()

3 МОДЕЛИРОВАНИЕ СМО
# РАСЧЕТ ИНТЕНСИВНОСТЕЙ
mu_del = (timeser_del[(timeser_del['date'] > '2020-01-01') &
(timeser_del['date'] <= '2020-03-28')]['time'].mean() / 60)**-1
mu_ser = (timeser_ser[(timeser_ser['date'] > '2020-01-01') &
(timeser_ser['date'] <= '2020-03-28')]['time'].mean() / 60)**-1
mu_con = (timeser_con[(timeser_con['date'] > '2020-01-01') &
(timeser_con['date'] <= '2020-03-28')]['time'].mean() / 60)**-1

lam_del = (8*60/(preds_fut_del[0] + 23)) ** -1
lam_ser = (8*60/(preds_fut_ser[0] + 24)) ** -1
lam_con = (8*60/(preds_fut_con[0] + 9)) ** -1

# КЛАСС СМО
class QueuingSystem(object):
    def __init__(self, N, M, LAM, MU, NU):
        self.n = N # Число окон обслуживания
        self.m = M # Число мест в очереди
        self.lam = LAM # Интенсивность входного потока
        self.nu = NU # Интенсивность обслуживания
        self.mu = MU # Интенсивность ухода заявок из очереди
        self.rho = self.lam / self.mu
        self.beta = self.nu / self.mu

#Матрица плотностей перехода из состояния в состояние
def FindL(self, n, m, lam, mu, nu):
    L = np.zeros((n+m+1, n+m+1))
    for i in range(n+1):

```

```

        L[i][i+1] = lam
        L[i][i-1] = i*mu
    for i in range(m):
        L[n+i][n+i+1] = lam
        L[n+i][n+i-1] = n*mu + i*nu
    for i in range(1, m+n):
        L[i][i] = -(L[i][i+1] + L[i][i-1])
    L[0][0] = -lam
    L[n+m][n+m-1] = n*mu + m*nu
    L[n+m][n+m] = -L[n+m][n+m-1]
    return L

#Стационарное распределение вероятности
def FindProbabilities(self, n, m, lam, mu, nu):
    P = np.zeros((n+m+1, 1))
    for i in range(n+1):
        P[0] += pow(lam/mu, i)/math.factorial(i)
    mul = 0
    mul_k = 1
    for j in range(1, m+1):
        for k in range(1, j+1):
            mul_k *= (lam/(n*mu + k*nu))
        mul += mul_k
        mul_k = 1
    P[0] += pow(lam/mu, n)/math.factorial(n)*mul
    P[0] = pow(P[0], -1)
    for i in range(1, n+1):
        P[i] = P[0]*pow(lam/mu, i)/math.factorial(i)
    mul_k = 1
    k = 0
    for i in range(n+1, n+m+1):
        k = k+1
        for k in range(1, k+1):
            mul_k *= (lam/(n*mu + k*nu))
        P[i] = P[n]*mul_k
        mul_k = 1
    return(P)

# Среднее число занятых каналов (z)
def AverageNumberOfBusyChannels(self, n, P):
    avernum = 0
    for i in range(1, n):
        avernum += i*P[i]
    avernum += n*(1-sum(P[0:n]))
    return float(avernum)

# Среднее число звонков в очереди (r)
def AverageNumberOfApplicationsInQueue(self, n, P, rho, beta):
    return float(rho/beta - self.AverageNumberOfBusyChannels(n, P)/beta)

# Абсолютная пропускная способность
def AbsoluteThroughput(self, lam, nu, n, P, rho, beta):
    return lam -
nu*self.AverageNumberOfApplicationsInQueue(n, P, rho, beta)

# Относительная пропускная способность
def RelativeThroughput(self, lam, nu, n, P, rho, beta):
    return self.AbsoluteThroughput(lam, nu, n, P, rho, beta)/lam

# Среднее время пребывания в очереди

```

```

def AverageTimeSpendingInQueue(self, n, P, rho, beta, lam):
    return self.AverageNumberOfApplicationsInQueue(n, P, rho, beta)/lam

# Среднее время обслуживания
def AverageServiceTime(self, lam, mu, nu, n, P, rho, beta):
    return self.RelativeThroughput(lam, nu, n, P, rho, beta)/mu

# Расчет характеристик системы
def FindCharacteristics(self):
    self.L = self.FindL(self.n, self.m, self.lam, self.mu, self.nu)
    self.P = self.FindProbabilities(self.n, self.m, self.lam, self.mu, self.nu)

    self.P0 = float(self.P[0])
    self.P_otk = float(self.P[-1])
    self.N_busy = self.AverageNumberOfBusyChannels(self.n, self.P)
    self.N_pr = self.n - self.N_busy
    self.k_busy = self.N_busy / self.n
    self.k_pr = 1 - self.k_busy
    self.q = self.RelativeThroughput(self.lam, self.nu, self.n, self.P, self.rho, self.beta)
    self.A = 60*self.AbsoluteThroughput(self.lam, self.nu, self.n, self.P, self.rho, self.beta)
    self.L_queue = self.AverageNumberOfApplicationsInQueue(self.n, self.P, self.rho, self.beta)
    self.L_total = self.L_queue + self.N_busy
    self.t_queue = self.AverageTimeSpendingInQueue(self.n, self.P, self.rho, self.beta, self.lam)
    self.t_serv = self.AverageServiceTime(self.lam, self.mu, self.nu, self.n, self.P, self.rho, self.beta)
    self.t_total = self.t_queue + self.t_serv

# СТАЦИОНАРНЫЙ РЕЖИМ
#Функция для расчета вероятностей в динамическом режиме
def StationaryPhase(system, type, legend_ncol=7, bbox_to_anchor=(1.9,0.8)):
    w, v = LA.eig(system.L.T) # собственные числа и собственные векторы
    b = np.zeros_like(v[0])
    b[0] = 1
    C = LA.solve(v, b)
    t = np.linspace(0, 90, 1000) # временной промежуток
    # динамическое создание переменных
    probas = []
    for i in range(system.n+system.m+1):
        probas.append("p" + str(i))
    values = [0 for i in range(system.n+system.m+1)]
    i = 0
    for x,y in zip(probas, values):
        globals()[x] = y
        for j in range(system.n+system.m+1):
            globals()[x] += C[j]*v[:,j][i]*np.exp(w[j]*t)
        i += 1
    i = 0
    fig, ax = plt.subplots(figsize=(8,4))
    fig.suptitle(type, fontsize=14)
    for x,y in zip(probas, values):
        ax.plot(t, globals()[x], label="P" + str(i) + "(t)")
        i += 1
    ax.set_xlabel("Время (мин.)")
    ax.set_ylabel("Вероятность")
    ax.legend(ncol=legend_ncol, bbox_to_anchor=bbox_to_anchor)

```



```

plt.show()

#константы
m = 30 # ограниченность очереди
nu = 1/45 #мин^-1
n = 30 #всего окон обслуживания

#определение характеристик
system_delivery = QueuingSystem(7, m, lam_del, mu_del, nu)
system_delivery.FindCharacteristics()
system_service = QueuingSystem(22, m, lam_ser, mu_ser, nu)
system_service.FindCharacteristics()
system_consultation = QueuingSystem(1, m, lam_con, mu_con, nu)
system_consultation.FindCharacteristics()

#графики
StationaryPhase(system_delivery, 'Выдача результатов', 5)
StationaryPhase(system_service, 'Прием документов', 5, (1.9, 0.95))
StationaryPhase(system_consultation, 'Консультирование', 5)

4 ОПРЕДЕЛЕНИЕ ОПТИМАЛЬНОГО РАСПРЕДЕЛЕНИЯ ОКОН
def FindOptimalNumberOfServiceWindows(n, m, lam_d, lam_s, lam_c, mu_d, mu_s,
mu_c, nu, q_opt_d, q_opt_s, q_opt_c):
    n_c_init = 2
    n_d_init = 2
    L_TOTAL_MAX = 1000
    for n_s in range(5, (n - n_c_init - n_d_init + 1)):
        for n_c in range(n_c_init, n - n_d_init - n_s + 1):
            n_d = n - n_s - n_c
            system_delivery = QueuingSystem(n_d, m, lam_d, mu_d, nu)
            system_delivery.FindCharacteristics()
            system_service = QueuingSystem(n_s, m, lam_s, mu_s, nu)
            system_service.FindCharacteristics()
            system_consultation = QueuingSystem(n_c, m, lam_c, mu_c, nu)
            system_consultation.FindCharacteristics()
            L_TOTAL = system_delivery.L_total + system_service.L_total + system_co
nsultation.L_total
            if ((L_TOTAL < L_TOTAL_MAX) and (system_delivery.q > q_opt_d) and (sys
tem_service.q > q_opt_s) and (system_consultation.q > q_opt_c)):
                L_TOTAL_MAX = L_TOTAL
                n_d_opt = n_d
                n_s_opt = n_s
                n_c_opt = n_c
            if (L_TOTAL_MAX == 1000):
                print("Оптимального распределения окон обслуживания, удовлетворяющим зад
анным условиям, обнаружить не удалось. Попробуйте изменить условия.")
                return 0,0,0
            else:
                return n_d_opt, n_s_opt, n_c_opt
n_del, n_ser, n_con = FindOptimalNumberOfServiceWindows(n, m, lam_del, lam_s
er, lam_con, mu_del, mu_ser, mu_con, nu, 0.85, 0.8, 0.85)

#Расчет характеристик
system_delivery = QueuingSystem(n_del, m, lam_del, mu_del, nu)
system_delivery.FindCharacteristics()
system_service = QueuingSystem(n_ser, m, lam_ser, mu_ser, nu)
system_service.FindCharacteristics()
system_consultation = QueuingSystem(n_con, m, lam_con, mu_con, nu)
system_consultation.FindCharacteristics()

```

## Приложение Г (обязательное)

### Программный код рекомендательной системы на языке Python

```
# 1 Подключение библиотек
!pip install cpt
pip install catboost
pip install keras-tqdm
import pandas as pd
import numpy as np
import tensorflow as tf
import catboost
from cpt.cpt import Cpt
import json
from sklearn.model_selection import train_test_split, cross_val_score
from sklearn.metrics import mean_squared_error, accuracy_score
from sklearn.utils import shuffle
from sklearn.preprocessing import LabelEncoder, MinMaxScaler
import math
import matplotlib.pyplot as plt
from keras.models import Sequential
from keras.layers import Dense, LSTM, Dropout
from keras.utils import np_utils
from keras.callbacks import History #история обучения
from keras_tqdm import TQDMNotebookCallback
from google.colab import drive
drive.mount('/content/drive')

# 2 Загрузка данных
raw_data = pd.read_csv('drive/My Drive/BKP/1 ДАННЫЕ/train.csv', usecols = [1,
4], nrows=2713723, names=['ID', 'service_title'], dtype=np.int32)
re-
quester = pd.read_csv('drive/My Drive/BKP/1 ДАННЫЕ/requester.csv', sep=';',
usecols = [0,1])
requester.columns = ['type', 'ID']
data_with_types = raw_data.merge(requester, on='ID')

#разбивка данных по каждому виду заявителей
data1 = data_with_types[data_with_types['type'] == 1].copy()
data2 = data_with_types[data_with_types['type'] == 2].copy()
data3 = data_with_types[data_with_types['type'] == 3].copy()
del data_with_types

# функция для деления выборки заявителей на тренировочную и тестовую
def split_requesters(data_about_requesters, test_size = 0.2):
    unique_ids = list(set(data_about_requesters['ID'])) #множество уникальных
значений ID
    indexes = list(range(0, len(unique_ids)))
    indexes = shuffle(indexes, random_state=8)
    test_indexes = unique_ids[:int(len(unique_ids)*test_size)]
    test_requesters = data_about_requesters[data_about_requesters['ID'].isin(t
est_indexes)]
    train_indexes = unique_ids[int(len(unique_ids)*test_size):]
```

```

train_requesters = data_about_requesters[data_about_requesters['ID'].isin(
train_indexes)]
return train_requesters, test_requesters

#Train и test выборки по типам заявителей
train_req1, test_req1 = split_requesters(data1, 0.17)
train_req2, test_req2 = split_requesters(data2, 0.245)
train_req3, test_req3 = split_requesters(data3, 0.1832)
TRAIN = pd.concat([train_req1, train_req2, train_req3])
TEST = pd.concat([test_req1, test_req2, test_req3])
del train_req1['type'], test_req1['type'], train_req2['type'], test_req2['ty
pe'], train_req3['type'], test_req3['type'], data1, data2, data3

# 3 COMPACT PREDICTION TREE

# Преобразование данных к нужному формату для CPT
def prepare_data_for_cpt(DATA, min_n, max_n, type):
    data_cpt = DATA.copy()
    data_cpt['N'] = data_cpt.groupby("ID")['ID'].transform('count')
    data_cpt.sort_values(by='ID', inplace=True)
    data_cpt = data_cpt[data_cpt['N'] >= min_n]
    data_cpt = data_cpt.groupby(['ID']).aggregate(lambda x: list(x))
    data_cpt['N'] = data_cpt['N'].str[0]
    data_with_N_more_than_nmax = data_cpt[data_cpt['N'] >= max_n].copy()
    da-
ta_with_N_more_than_nmax['service_title'] = data_with_N_more_than_nmax['serv
ice_title'].str[-max_n:]
    data_with_N_more_than_nmax['N'] = max_n
    data_cpt = data_cpt[data_cpt['N'] <= max_n]
    data_cpt = pd.concat([data_cpt, data_with_N_more_than_nmax], axis=0)
    data_cpt.reset_index(inplace=True)
    data_cpt.drop(columns=['ID', 'N'], inplace=True)
    data_cpt.columns = ['Service Sequence by ID']
    if type == 'train':
        return(data_cpt['Service Sequence by ID'])
    elif type == 'test':
        return data_cpt['Service Sequence by ID'].str[0:-
1], data_cpt['Service Sequence by ID'].str[-1]

# Определение оптимальных параметров
def FindOptimalParametres(train, X_test, y_test, tuned_params):
    max_score = 0
    y_true = y_test
    for mbr in tuned_params['MBR']:
        for length in tuned_params['split_length']:
            model = Cpt(MBR = mbr, split_length = length)
            model.fit(train)

            y_pred = model.predict(list(X_test))
            score = accuracy_score(y_true, y_pred)
            if score > max_score:
                max_score = score
                optimal_model = model
                optimal_length = length
                optimal_MBR = mbr
    print("Optimal parametres:\nsplit_length = ", optimal_length, "\nMBR = ",
optimal_MBR, "\nAccuracy = ", max_score)

# 3.1 Физические лица

```

```

# ДАННЫЕ
min_n = 3 #минимальное число заявлений, которое должен подать заявитель
max_n = 10 #максимальное число заявлений, которое может подать заявитель
train_cpt1 = prepare_data_for_cpt(train_req1, min_n, max_n, 'train')
X_test_cpt1, y_test_cpt1 = prepare_data_for_cpt(test_req1, min_n, max_n, 'test')
# МОДЕЛЬ
model_cpt1 = Cpt()
model_cpt1.fit(train_cpt1)
y_pred = model_cpt1.predict(list(X_test_cpt1))
accuracy_score(y_test_cpt1, y_pred) #0.107159
# ТЮНИНГ БАЗОВОЙ МОДЕЛИ
tuned_params1 = {'MBR': [0], 'split_length': range(0,10,1)}
FindOptimalParametres(train_cpt1, X_test_cpt1, y_test_cpt1, tuned_params1)

# 3.2 Юридические лица
# ДАННЫЕ
min_n = 4 #минимальное число заявлений, которое должен подать заявитель
max_n = 10 #максимально число заявлений, которое может подать заявитель
train_cpt2 = prepare_data_for_cpt(train_req2, min_n, max_n, 'train')
X_test_cpt2, y_test_cpt2 = prepare_data_for_cpt(test_req2, min_n, max_n, 'test')
# МОДЕЛЬ
model_cpt2 = Cpt()
model_cpt2.fit(train_cpt2)
y_pred = model_cpt2.predict(list(X_test_cpt2))
accuracy_score(y_test_cpt2, y_pred)
# ТЮНИНГ БАЗОВОЙ МОДЕЛИ
tuned_params2 = {'MBR': [0], 'split_length': range(0,10,1)}
FindOptimalParametres(train_cpt2, X_test_cpt2, y_test_cpt2, tuned_params2)

# 3.3 Индивидуальные предприниматели
# ДАННЫЕ
min_n = 3 #минимальное число заявлений, которое должен подать заявитель
max_n = 7 #максимально число заявлений, которое может подать заявитель
train_cpt3 = prepare_data_for_cpt(train_req3, min_n, max_n, 'train')
X_test_cpt3, y_test_cpt3 = prepare_data_for_cpt(test_req3, min_n, max_n, 'test')
# МОДЕЛЬ
model_cpt3 = Cpt()
model_cpt3.fit(train_cpt3)
y_pred = model_cpt3.predict(list(X_test_cpt3))
accuracy_score(y_test_cpt3, y_pred) # 0.209
# ТЮНИНГ БАЗОВОЙ МОДЕЛИ
tuned_params3 = {'MBR': [0], 'split_length': range(0,10,1)}
FindOptimalParametres(train_cpt3, X_test_cpt3, y_test_cpt3, tuned_params3)

# 4 RNN LSTM
# Функция для подготовки тестовых данных
def prepare_test_data_for_rnn_lstm(DATA, min_n):
    data_rnn = DATA.copy()
    data_rnn['service_title'] = data_rnn['service_title'].astype('str')
    data_rnn['N'] = data_rnn.groupby("ID")['ID'].transform('count')
    data_rnn.sort_values(by='ID', inplace=True)
    data_rnn = data_rnn[data_rnn['N'] >= min_n]
    data_rnn = data_rnn.groupby(['ID']).aggregate(lambda x: list(x))
    data_rnn['N'] = data_rnn['N'].str[0]
    data_with_N_more_than_min_n = data_rnn[data_rnn['N'] > min_n].copy()

```

```

da-
ta_with_N_more_than_minnn['service_title'] = data_with_N_more_than_minnn['serv
ice_title'].str[-min_n:]
data_with_N_more_than_minnn['N'] = min_n
data_rnn = data_rnn[data_rnn['N'] <= min_n]
data_rnn = pd.concat([data_rnn, data_with_N_more_than_minnn], axis=0)
data_rnn.reset_index(inplace=True)
data_rnn.drop(columns=['ID', 'N'], inplace=True)
data_rnn.columns = ['Service Sequence by ID']
data_rnn['START_SERVICES'] = data_rnn['Service Sequence by ID'].str[0:-1]
data_rnn['END_SERVICE'] = data_rnn['Service Sequence by ID'].str[-1]
data_rnn.drop(columns=['Service Sequence by ID'], inplace=True)
X_rnn = data_rnn['START_SERVICES']
y_rnn = data_rnn['END_SERVICE']
X_rnn = np.vstack(X_rnn.to_numpy()).reshape(X_rnn.shape[0], min_n - 1, 1)
y_rnn = np.array(y_rnn).reshape(y_rnn.shape[0], 1)
X_rnn = np_utils.to_categorical(X_rnn, num_classes=1310)[:,:,:1:]
y_rnn = np_utils.to_categorical(y_rnn, num_classes=1310)[:,:1:]
return X_rnn, y_rnn

# Функция для подготовки тренировочных данных
def extend_train_data_for_rnn_lstm(DATA, min_n, max_last_n):
    data_rnn = DATA.copy()
    data_rnn['service_title'] = data_rnn['service_title'].astype('str')
    data_rnn['N'] = data_rnn.groupby("ID")['ID'].transform('count')
    data_rnn.sort_values(by='ID', inplace=True)

    data_rnn = data_rnn[data_rnn['N'] >= min_n]
    data_rnn = data_rnn.groupby(['ID']).aggregate(lambda x: list(x))
    data_rnn['N'] = data_rnn['N'].str[0]
    data_with_N_equal_minnn = data_rnn[data_rnn['N'] == min_n].copy()
    data_with_N_more_than_minnn = data_rnn[data_rnn['N'] > min_n].copy()
    da-
    ta_with_N_more_than_minnn['service_title'] = data_with_N_more_than_minnn['serv
ice_title'].str[-max_last_n:]
    data_with_N_equal_minnn.reset_index(inplace=True)
    data_with_N_equal_minnn.drop('ID', axis=1, inplace=True)
    new_llist = list()
    data_list = list(data_with_N_more_than_minnn['service_title'])
    for i in range(len(data_with_N_more_than_minnn)):
        lllist = data_list[i]
        for j in range(len(lllist) - min_n + 1):
            new_llist.append(lllist[j:j+min_n])
    extended_data = pd.DataFrame({'service_title' : new_llist, 'N' : min_n})
    data_rnn = pd.concat([data_with_N_equal_minnn, extended_data], axis=0)
    data_rnn.drop(columns=['N'], inplace=True)
    data_rnn['START_SERVICES'] = data_rnn['service_title'].str[0:-1]
    data_rnn['END_SERVICE'] = data_rnn['service_title'].str[-1]
    data_rnn.drop(columns=['service_title'], inplace=True)
    X_rnn = data_rnn['START_SERVICES']
    y_rnn = data_rnn['END_SERVICE']
    X_rnn = np.vstack(X_rnn.to_numpy()).reshape(X_rnn.shape[0], min_n - 1, 1)
    y_rnn = np.array(y_rnn).reshape(y_rnn.shape[0], 1)
    X_rnn = np_utils.to_categorical(X_rnn, num_classes=1310)[:,:,:1:]
    y_rnn = np_utils.to_categorical(y_rnn, num_classes=1310)[:,:1:]
    return X_rnn, y_rnn

# Функция для визуализации процесса обучения
def build_graphs(history, num_epochs, K):
    train_loss = history.history['loss']

```

```

val_loss = history.history['val_loss']
train_acc = history.history['categorical_accuracy']
val_acc = history.history['val_categorical_accuracy']
train_top_k_acc = history.history['top_k_categorical_accuracy']
val_top_k_acc = history.history['val_top_k_categorical_accuracy']
xc = range(1, num_epochs+1, 1)
fig, (ax1, ax2, ax3) = plt.subplots(1, 3)
plt.subplots_adjust(left=0.1, right=3, wspace=0.3, hspace=0.3, bottom=0.1,
top=1)
ax1.set_title('Loss on train and validation subsamples')
ax1.set_xlabel('epoch')
ax1.set_ylabel('loss')
ax1.plot(xc, train_loss, label="train")
ax1.plot(xc, val_loss, label="val")
ax1.set_xticks(np.arange(min(xc), max(xc)+1, 1))
ax1.legend()
ax2.set_title('Categorical accuracy on train and validation subsamples')
ax2.set_xlabel('epoch')
ax2.set_ylabel('Accuracy')
ax2.plot(xc, train_acc, label="train")
ax2.plot(xc, val_acc, label="val")
ax2.set_xticks(np.arange(min(xc), max(xc)+1, 1))
ax2.legend()
ax3.set_title('Top ' + str(K) + ' Categorical accuracy on train and valida
tion subsamples')
ax3.set_xlabel('epoch')
ax3.set_ylabel('Top ' + str(K) + ' Accuracy')
ax3.plot(xc, train_top_k_acc, label="train")
ax3.plot(xc, val_top_k_acc, label="val")
ax3.set_xticks(np.arange(min(xc), max(xc)+1, 1))
ax3.legend()
None

```

```

# 4.1 Физические лица

```

```

# ДАННЫЕ

```

```

min_n = 6

```

```

max_last_n = 8

```

```

X_train_rnn1, y_train_rnn1 = extend_train_data_for_rnn_lstm(train_req1, min_
n, max_last_n)

```

```

X_test_rnn1, y_test_rnn1 = prepare_test_data_for_rnn_lstm(test_req1, min_n)

```

```

# МОДЕЛЬ

```

```

num_epochs1 = 8

```

```

K1 = 5

```

```

model_rnn1 = Sequential()

```

```

mod-

```

```

el_rnn1.add(LSTM(512, input_shape=(X_train_rnn1.shape[1], X_train_rnn1.shape
[2]), return_sequences=True, activation='relu'))

```

```

model_rnn1.add(Dropout(0.2))

```

```

model_rnn1.add(LSTM(256, activation='relu'))

```

```

model_rnn1.add(Dropout(0.1))

```

```

model_rnn1.add(Dense(y_train_rnn1.shape[1], activation='softmax'))

```

```

model_rnn1.compile(loss='categorical_crossentropy', optimizer='adam',
                    metrics=[tf.keras.metrics.CategoricalAccuracy(),
                             tf.keras.metrics.TopKCategoricalAccuracy(k=K1)])

```

```

history1 = model_rnn1.fit(X_train_rnn1, y_train_rnn1, epochs=num_epochs1,
validation_split=0.2, batch_size=64, verbose=1)

```

```

# ТОЧНОСТЬ

```

```

y_pred_rnn1 = model_rnn1.predict(X_test_rnn1, verbose=0)

```

```

accuracy_score_rnn(y_test_rnn1, y_pred_rnn1)

```

```

m = tf.keras.metrics.TopKCategoricalAccuracy(k=5)

```

```

m.update_state(y_test_rnn1, y_pred_rnn1)
m.result().numpy()
build_graphs(history1, num_epochs1, K1)

# 4.2 Юридические лица
# ДАННЫЕ
min_n = 7
max_last_n = 10000
X_train_rnn2, y_train_rnn2 = extend_train_data_for_rnn_lstm(train_req2,
min_n, max_last_n)
X_test_rnn2, y_test_rnn2 = prepare_test_data_for_rnn_lstm(test_req2, min_n)

#6, 30, 6, epochs=10, Acc 0.71036
# МОДЕЛЬ
num_epochs2 = 7
K = 5 #top number
model_rnn2 = Sequential()
model_rnn2.add(LSTM(512, input_shape=(X_train_rnn2.shape[1],
X_train_rnn2.shape[2]), return_sequences=True, activation='relu'))
model_rnn2.add(Dropout(0.2))
model_rnn2.add(LSTM(256, activation='relu'))
model_rnn2.add(Dropout(0.1))
model_rnn2.add(Dense(y_train_rnn2.shape[1], activation='softmax'))
model_rnn2.compile(loss='categorical_crossentropy', optimizer='adam',
metrics=[tf.keras.metrics.CategoricalAccuracy(),
tf.keras.metrics.TopKCategoricalAccuracy(k=K)])
history2 = model_rnn2.fit(X_train_rnn2, y_train_rnn2, epochs=num_epochs2,
validation_split=0.2, shuffle=True, verbose=1)
# ТОЧНОСТЬ
y_pred_rnn2 = model_rnn2.predict(X_test_rnn2, verbose=0)
accuracy_score_rnn(y_test_rnn2, y_pred_rnn2)
build_graphs(history2, num_epochs2, 5)
m = tf.keras.metrics.TopKCategoricalAccuracy(k=5)
m.update_state(y_test_rnn2, y_pred_rnn2)
m.result().numpy()

# 4.3 Индивидуальные предприниматели
# ДАННЫЕ
min_n = 4
max_last_n = 10000
X_train_rnn3, y_train_rnn3 = extend_train_data_for_rnn_lstm(train_req3, min_n,
max_last_n)
X_test_rnn3, y_test_rnn3 = prepare_test_data_for_rnn_lstm(test_req3, min_n)
# МОДЕЛЬ
num_epochs3 = 20
K = 5
model_rnn3 = Sequential()
model_rnn3.add(LSTM(256, input_shape = (X_train_rnn3.shape[1],
X_train_rnn3.shape[2]), return_sequences=True, activation='relu'))
model_rnn3.add(Dropout(0.1))
model_rnn3.add(LSTM(128, activation='relu'))
model_rnn3.add(Dropout(0.1))
model_rnn3.add(Dense(y_train_rnn3.shape[1], activation='softmax'))
model_rnn3.compile(loss='categorical_crossentropy', optimizer='adam',
metrics=[tf.keras.metrics.CategoricalAccuracy(),
tf.keras.metrics.TopKCategoricalAccuracy(k=K)])
history3 = model_rnn3.fit(X_train_rnn3, y_train_rnn3, epochs=num_epochs3,
validation_split=0.2, shuffle=True, verbose=1)
# ТОЧНОСТЬ
y_pred_rnn3 = model_rnn3.predict(X_test_rnn3, verbose=0)

```



```

accuracy_score_rnn(y_test_rnn3, y_pred_rnn3)
build_graphs(history3, num_epochs3, 5)
m = tf.keras.metrics.TopKCategoricalAccuracy(k=5)
m.update_state(y_test_rnn3, y_pred_rnn3)
m.result().numpy()

# 5 CatBoost
# Функция для подготовки тренировочных данных
def extend_train_data_for_catboost(DATA, min_n, max_last_n):
    data_xgb = DATA.copy()
    data_xgb['service_title'] = data_xgb['service_title'].astype('str')
    data_xgb['N'] = data_xgb.groupby("ID")['ID'].transform('count')
    data_xgb.sort_values(by='ID', inplace=True)
    data_xgb = data_xgb[data_xgb['N'] >= min_n]
    data_xgb = data_xgb.groupby(['ID']).aggregate(lambda x: list(x))
    data_xgb['N'] = data_xgb['N'].str[0]
    data_with_N_equal_min_n = data_xgb[data_xgb['N'] == min_n].copy()
    data_with_N_more_than_min_n = data_xgb[data_xgb['N'] > min_n].copy()
    data_with_N_more_than_min_n['service_title'] = data_with_N_more_than_min_n['service_title'].str[-max_last_n:]
    data_with_N_equal_min_n.reset_index(inplace=True)
    data_with_N_equal_min_n.drop('ID', axis=1, inplace=True)
    new_llist = list()
    data_list = list(data_with_N_more_than_min_n['service_title'])
    for i in range(len(data_with_N_more_than_min_n)):
        llist = data_list[i]
        for j in range(len(llist) - min_n + 1):
            new_llist.append(llist[j:j+min_n])
    extended_data = pd.DataFrame({'service_title' : new_llist, 'N' : min_n})
    data_xgb = pd.concat([data_with_N_equal_min_n, extended_data], axis=0)
    data_xgb.drop(columns=['N'], inplace=True)
    data_xgb['START_SERVICES'] = data_xgb['service_title'].str[0:-1]
    data_xgb['END_SERVICE'] = data_xgb['service_title'].str[-1]
    data_xgb.drop(columns=['service_title'], inplace=True)
    X_xgb = data_xgb[['START_SERVICES']]
    y_xgb = data_xgb[['END_SERVICE']]
    col_names = list()
    for i in range(min_n-1):
        col_names.append('ser' + str(i))
    X_xgb[col_names] = X_xgb.START_SERVICES.to_list()
    X_xgb = X_xgb.drop(columns=['START_SERVICES'])
    return X_xgb, y_xgb

# Функция для подготовки тестовых данных
def prepare_test_data_for_catboost(DATA, min_n):
    data_xgb = DATA.copy()
    data_xgb['service_title'] = data_xgb['service_title'].astype('str')
    data_xgb['N'] = data_xgb.groupby("ID")['ID'].transform('count')
    data_xgb.sort_values(by='ID', inplace=True)
    data_xgb = data_xgb[data_xgb['N'] >= min_n]
    data_xgb = data_xgb.groupby(['ID']).aggregate(lambda x: list(x))

    data_xgb['N'] = data_xgb['N'].str[0]
    data_with_N_more_than_min_n = data_xgb[data_xgb['N'] > min_n].copy()
    data_with_N_more_than_min_n['service_title'] = data_with_N_more_than_min_n['service_title'].str[-min_n:]
    data_with_N_more_than_min_n['N'] = min_n

```



```

data_xgb = data_xgb[data_xgb['N'] <= min_n]
data_xgb = pd.concat([data_xgb, data_with_N_more_than_min_n], axis=0)
data_xgb.reset_index(inplace=True)
data_xgb.drop(columns=['ID', 'N'], inplace=True)
data_xgb['START_SERVICES'] = data_xgb['service_title'].str[0:-1]
data_xgb['END_SERVICE'] = data_xgb['service_title'].str[-1]
data_xgb.drop(columns=['service_title'], inplace=True)
X_xgb = data_xgb[['START_SERVICES']]
y_xgb = data_xgb[['END_SERVICE']]
col_names = list()
for i in range(min_n-1):
    col_names.append('ser' + str(i))
X_xgb[col_names] = X_xgb.START_SERVICES.to_list()
X_xgb = X_xgb.drop(columns=['START_SERVICES'])
return X_xgb, y_xgb

# Функция для оценки точности
def accuracy_ctb(y_true, y_pred):
    res = y_true == y_pred
    return (len(res[res['END_SERVICE'] == True]) / len(res))

# Функция для составления таблицы с правильными прогнозами
def return_true_predicts(X_test, y_test, y_pred):
    predicts_df = pd.DataFrame(data = X_test)
    predicts_df['pred_ser'] = y_pred
    predicts_df['is_true_pred'] = y_test == y_pred
    return predicts_df[predicts_df['is_true_pred'] == True]

# 5.1 Физические лица
# ДАННЫЕ
min_n = 6
max_last_n = 7
X_train_ctb1, y_train_ctb1 = extend_train_data_for_catboost(train_req1, min_n, max_last_n)
X_test_ctb1, y_test_ctb1 = prepare_test_data_for_catboost(test_req1, min_n)
cat_features1 = list(range(0, X_train_ctb1.shape[1]))
#МОДЕЛЬ
model_ctb1 = catboost.CatBoostClassifier(iterations=150, learning_rate=0.05, depth=4, loss_function = 'MultiClass', random_state = 8, task_type = 'GPU')
model_ctb1.fit(X_train_ctb1, y_train_ctb1, cat_features=cat_features1, verbose=10)
#ТОЧНОСТЬ
y_pred_ctb1 = model_ctb1.predict(data=X_test_ctb1)
print("Accuracy is ", accuracy_ctb(y_test_ctb1, y_pred_ctb1))
return_true_predicts(X_test_ctb1, y_test_ctb1, y_pred_ctb1)

# 5.2 Юридические лица
# ДАННЫЕ
min_n = 6
max_last_n = 150
X_train_ctb2, y_train_ctb2 = extend_train_data_for_catboost(train_req2, min_n, max_last_n)
X_test_ctb2, y_test_ctb2 = prepare_test_data_for_catboost(test_req2, min_n)
cat_features2 = list(range(0, X_train_ctb2.shape[1]))
#МОДЕЛЬ
model_ctb2 = catboost.CatBoostClassifier(iterations = 300, learning_rate=0.08, depth=4, loss_function='MultiClass', random_state=8, task_type='GPU')

```

```

model_ctb2.fit(X_train_ctb2, y_train_ctb2, cat_features
cat_features2, verbose = 10)
#ТОЧНОСТЬ
y_pred_ctb2 = model_ctb2.predict(data=X_test_ctb2)
print("Accuracy is ", accuracy_ctb(y_test_ctb2, y_pred_ctb2))
return_true_predicts(X_test_ctb2, y_test_ctb2, y_pred_ctb2)

# 5.3 Индивидуальные предприниматели
# ДАННЫЕ
min_n = 4
max_last_n = 1000
X_train_ctb3, y_train_ctb3 = extend_train_data_for_catboost(train_req3, min_n, max_last_n)
X_test_ctb3, y_test_ctb3 = prepare_test_data_for_catboost(test_req3, min_n)
cat_features3 = list(range(0, X_train_ctb3.shape[1]))
#МОДЕЛЬ
model_ctb3 = catboost.CatBoostClassifier(iterations=150, learning_rate =
0.1, depth=4, loss_function='MultiClass', random_state=8, task_type='GPU')
model-
el_ctb3.fit(X_train_ctb3, y_train_ctb3, cat_features=cat_features3, verbose=
10)
#ТОЧНОСТЬ
y_pred_ctb3 = model_ctb3.predict(data=X_test_ctb3)
print("Accuracy is ", accuracy_ctb(y_test_ctb3, y_pred_ctb3))
return_true_predicts(X_test_ctb3, y_test_ctb3, y_pred_ctb3)

```