

Федеральное государственное автономное  
образовательное учреждение  
высшего образования  
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт космических и информационных технологий  
институт

«Информатика»  
кафедра

УТВЕРЖДАЮ  
Заведующий кафедрой

\_\_\_\_\_ А. С. Кузнецов  
подпись                      инициалы, фамилия

« \_\_\_\_\_ » \_\_\_\_\_ 2019 г.

**БАКАЛАВРСКАЯ РАБОТА**

09.03.04 Программная инженерия  
код и наименование специальности

Разработка системы поддержки принятия решения при диагностировании  
степени тяжести острого панкреатита

тема

Руководитель	_____	старший преподаватель	А. С. Михалев
	подпись, дата	должность, ученая степень	инициалы, фамилия
Выпускник	_____		Д. А. Мелех
	подпись, дата		инициалы, фамилия
Консультант	_____	ген. дир., канд. техн. наук	Н. А. Сергеева
	подпись, дата	должность, ученая степень	инициалы, фамилия
Нормоконтролер	_____	доцент, канд. техн. наук	О. А. Антамошкин
	подпись, дата	должность, ученая степень	инициалы, фамилия

Красноярск 2019

## РЕФЕРАТ

Выпускная квалификационная работа по теме «Разработка системы поддержки принятия решения при диагностировании степени тяжести острого панкреатита» содержит 55 страниц текстового документа, 33 использованных источника, 19 иллюстраций, 2 приложения.

**ОСТРЫЙ ПАНКРЕАТИТ, МЕТОД ОПОРНЫХ ВЕКТОРОВ, СИСТЕМА ПОДДЕРЖКИ ПРИНЯТИЯ РЕШЕНИЯ, ВЕБ-ПРИЛОЖЕНИЕ.**

Целью работы является разработка системы поддержки принятия решения при диагностировании степени тяжести острого панкреатита.

Для достижения поставленной цели решались следующие задачи:

- проанализирована предметная область;
- разработана модель прогнозирования степени тяжести острого панкреатита;
- выбран набор инструментов для разработки веб-приложения;
- определены требования к разрабатываемой системе;
- спроектировано веб-приложение;
- разработана система.

В ходе выполнения выпускной квалификационной работы были проведены операции по обработке данных и разработана модель прогнозирования на основе метода опорных векторов, спроектировано и разработано клиент-серверное приложение с внедрением полученного математического аппарата.

## СОДЕРЖАНИЕ

Введение.....	5
1. Анализ предметной области .....	8
1.1 Актуальность .....	9
1.2 Описание системы.....	10
1.3 Анализ аналогов .....	10
2 Данные и алгоритмы .....	13
2.1 Выбор инструмента для анализа данных.....	13
2.2 Данные.....	15
2.2.1 Формат исходных данных .....	15
2.2.2 Формат набора данных .....	16
2.2.3 Описание набора данных .....	16
2.3 Выбор основного алгоритма .....	17
2.3.1 Классификация или регрессия .....	18
2.4 Отбор информативных признаков.....	20
2.5 Подбор параметров модели.....	23
3 Выбор средств разработки .....	29
3.1 Серверная часть.....	29
3.2 Клиентская часть .....	31
3.3 База данных.....	34
4 Требования.....	36
4.1 Цели создания системы .....	36
4.2 Требования к разрабатываемой системе.....	36
5 Проектирование.....	38
5.1 База данных.....	38
5.2 Архитектура.....	40
5.3 Применение моделей .....	41
5.4 Прикладной программный интерфейс .....	43
6 Разработка .....	44
6.1 Создание БД и заполнение .....	44
6.2 Разработка серверной части .....	46

6.3 Разработка клиентской части.....	50
Заключение .....	55
Список сокращений .....	56
Список использованных источников .....	57
Приложение А Справка о сдаче в тестовое использование.....	60
Приложение Б Свидетельство о государственной регистрации .....	63

## ВВЕДЕНИЕ

В последнее время медицинская область переживает технологический подъем. Возникают, формируются и реализуются задачи по максимальной цифровизации различных рутинных операций, начиная от систем хранения данных и вплоть до частичной автоматизации (как системы поддержки) процессов по выявлению болезней и сопровождению лечения. Данная тенденция автоматизации медицинских процессов привела к появлению множества информационных систем по работе с данными – медицинских информационных систем (МИС). Например, к ним относятся: «БАРС.Здравоохранение – МИС», МИС qMS, К-МИС и т.д.

Наличие подобных функционирующих МИС, в свою очередь, сблизило такие области, как медицину, когнитивный анализ данных и технологии машинного обучения. По мере накопления данных в МИС стали появляться различные программы с применением искусственного интеллекта (ИИ), направленные на решение той или иной задачи в медицине. Например, уже сейчас существуют интеллектуальные системы, которые помогают врачам устанавливать диагноз, отслеживать состояние пациента или назначать лечение: BotkinAI, Watson Health, DeepMind Health и т.д.

Разработка интеллектуальных систем в медицинской сфере, как правило, ведется для: улучшения качества диагностики заболеваний, поиска новых закономерностей с целью совершенствования процесса распознавания или создания лекарственных препаратов, ускорения и упрощения процесса прогнозирования состояния пациента или исхода лечения и т.д.

Одним из таких заболеваний, потенциально имеющих запрос на решение с применением анализа данных и технологий машинного обучения, является острый панкреатит. Данное заболевание имеет высокий приоритет по причине быстротечности заболевания и высокой смертности.

Необходимость принятия каких-либо мер в отношении данного заболевания видна в собранной статистике. Например, в период с 1995 по 2010

гг. заболеваемость острым панкреатитом в г. Красноярске увеличилась больше чем в 2 раза, а в Красноярском крае в период с 2002 по 2010 гг. – в 1,5 раза. С 2002 по 2010 гг. треть летальных исходов в Красноярском крае была от острого панкреатита [1]. Исследования проблемы острого панкреатита в Красноярском крае на период времени с 2009 по 2017 гг. по-прежнему подтверждают высокую стабильность смертности пациентов с соответствующим заболеванием [2]. На мировом уровне данное заболевание также имеет высокие показатели смертности и требует привлечения внимания [3].

В целом, проблема острого панкреатита актуальна больше 40 лет [1, 2, 3]. Данному факту способствует множество причин, следствием которых является высокая стабильность летальных исходов от данного заболевания и, соответственно, его актуальность. В частности, среди причин выделяются: наличие субъективных показателей со стороны пациента, требования высокой скорости оказания необходимого лечения в соответствии с поставленной степенью тяжести, отсутствие необходимой аппаратуры для надежного диагностирования степени тяжести.

Хоть в настоящее время для диагностирования степени тяжести данного заболевания применяют различные шкалы (Apache II, Sofa, Ranson, Glasgow–Imrie, SAPS, MODS), но, зачастую, их невозможно использовать в стационарах из-за указанных выше причин, в особенности – нехватки аппаратуры [4]. При этом сами шкалы имеют определенный уровень ошибки постановки верной степени тяжести, которую на практике видно только по истечению 3-х суток (самое раннее) от начала лекарственной терапии.

В связи с особенностью причин и высокой приоритетностью проблемы было принято решение по реализации системы поддержки принятия решения по острому панкреатиту с применением анализа данных и машинного обучения для построения моделей распознавания степеней тяжести соответствующего заболевания. Данная система носит рекомендательный характер и не является заменой врача.

В качестве Заказчика в рамках проекта выступает компания ООО «Ар Ди Сайнс».

## **1. Анализ предметной области**

В мировом масштабе в настоящее время медицинская область активно развивается в направлении разработки систем поддержки принятия врачебных решений.

Все больше появляется различных решений по распознаванию одной или нескольких из определенной категории заболеваний, проверке на наличие конкретного диагноза, а также назначения лечения. Данная тенденция наблюдается и в России благодаря принятой стратегии развития персонафицированной медицины при поддержке здравоохранения Российской Федерации.

Медицинская область потенциально предоставляет объем данных, достаточный для постановки задачи и ее решения с применением анализа данных и машинного обучения.

С технологической стороны обработка «сырых» данных из историй болезни, содержащихся в МИС, по-прежнему имеет высокую сложность реализации из-за специфики внесения результатов обследований пациентов, к которым можно отнести: отсутствие жестких требований по заполнению полей, что приводит к частичному использованию структурированности данных, отличие терминологии в различных однотипных поликлиниках/больницах внутри одной нозологии (заболевания) или отсутствие полной унификации, что порождает новые задачи: парсинг слабоструктурированных текстов, восполнение пропусков, составление словарей синонимичных медицинских терминов и т.д.

Подобные особенности сбора и хранения данных приводят к тому, что процесс внедрения интеллектуальных систем затруднителен и имеет высокую сложность глобального обобщения решения. По этой причине существует необходимость плавного внедрения технологий с применением ИИ.



## 1.1 Актуальность

На уровне Красноярского края проблема острого панкреатита выделяется врачами как наиболее приоритетная по ряду причин. К ним относят:

- стабильно высокую смертность;
- подтверждение степени тяжести не ранее, чем через 3-е суток;
- быстротечность данного заболевания;
- сложность применения существующих методик (шкал) для постановки степени тяжести;
- наличие субъективных показателей в некоторых шкалах;
- вероятность неправильного лечения в ходе ошибочной постановки степени тяжести;
- позднее хирургическое вмешательство в ряде случаев.

Высокая приоритетность проблемы острого подтверждается экспертами и посредством статистических показателей (рисунок 1).

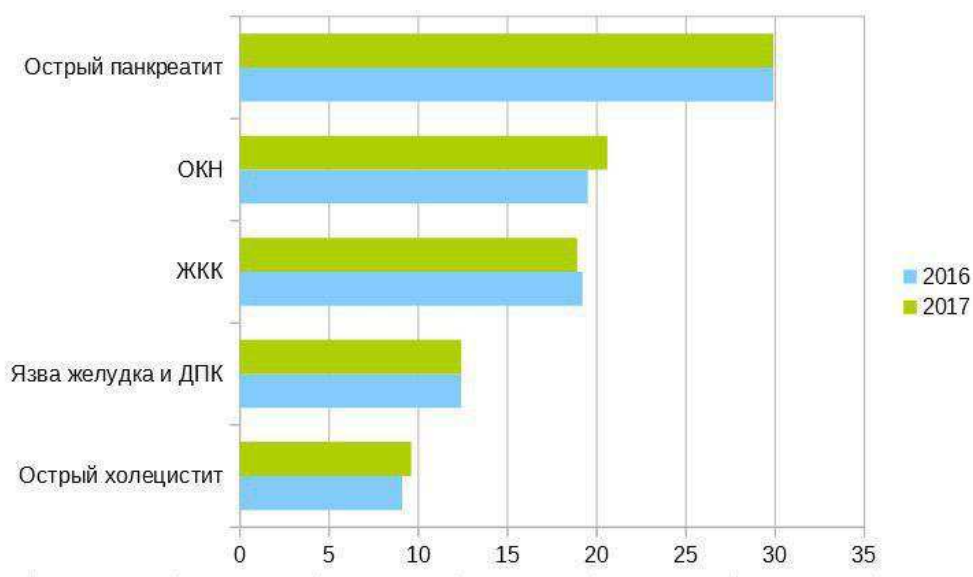


Рисунок 1 – Распределение умерших больных по нозологиям

## **1.2 Описание системы**

Разрабатываемая система поддержки принятия решения по острому панкреатиту или Sapia функционирует в формате клиент-серверного приложения (веб-приложения).

Данная система ориентирована на выдачу прогнозов о степени тяжести острого панкреатита посредством использования обученных моделей классификации.

Вся логика по работе с моделями располагается на стороне сервера, а взаимодействие с ними по заранее предусмотренным сценариям предоставляется посредством прикладного программного интерфейса (API). В целях безопасности и упрощения логики работы клиентской части передача обученных моделей по запросу для их непосредственного использования не предусмотрена.

Система предоставляет возможность ведения последовательности записей по одной истории болезни пациента, которые в совокупности характеризуют динамику его состояния в рамках конкретной истории болезни.

Веб-приложение Sapia максимально обезличена по отношению к пациентам. Данная система не содержит конфиденциальной информации. Она хранит уникальный идентификатор пациента, дату полученных значений показателей пациента, характеризующих его состояние, сами значения показателей, экспертную оценку и результаты прогнозов. Этот фактор позволяет в дальнейшем интегрировать Sapia в медицинские системы, стыковать посредством API с МИС.

## **1.3 Анализ аналогов**

В медицинской области существует множество задач, которые решены или потенциально могут быть решены с применением технологий машинного обучения. К числу таких проблем, как правило, относят задачи по преждевременному определению онкологических, сердечно-сосудистых

заболеваний, распознаванию риска возникновения болезни Альцгеймера, Паркинсона и т.д. Подобные заболевания имеют высокий приоритет из-за своей специфичности, высокой сложности диагностирования и рисках. Для их решения, как правило, исследуются новые методы и применяются последние разработки. Данными задачами занимаются множество компаний, включая такие IT-гиганты, как: компания IBM, корпорация Microsoft, компания Google, корпорация Intel и т.д.

Для медицинской сферы компанией IBM была разработана платформа Watson Health, которая нашла свое применение в некоторых клиниках из США, Индии и Таиланда. Данная платформа способна определить у пациентов наличие раковой опухоли. К онкологическим заболеваниям, на которых испытывалась платформа Watson Health, можно отнести: рак молочной железы, рак легких, рак прямой кишки, рак шейки матки, рак яичников, рак желудка и рак простаты. Также отмечается, что она ориентирована на работу с сердечно-сосудистыми заболеваниями [5, 6].

Помимо вышеприведенного продукта от компании IBM существуют и разрабатываются аналогичные системы в направлении онкологических заболеваний. Например, корпорация Microsoft занимается методами лечения от рака, анализом медицинских снимков и идентификацией злокачественных новообразований [7].

Наряду с онкологическими заболеваниями ведутся разработки и в других направлениях. Например, В 2018 году в Англии была запущена система с ИИ ScanNav, которая параллельно с врачом проводит УЗИ-диагностику для обнаружения патологий у беременных женщин. В области же микробиологии исследователи из VIDMC, используя микроскоп с ИИ Boston, диагностируют смертельные инфекции крови [8].

Также следует отметить технологию DeepMind Health, которая используется врачами Национальной службы здравоохранения Великобритании и направлена на сканирование головы и шейного отдела позвоночника для определения здоровых и пораженных тканей [9].

Из российских программ с уклоном в медицину выделяется система поддержки принятия врачебных решений с использованием технологий ИИ «Botkin.AI» для диагностирования онкологических заболеваний.

В настоящее время программ с применением ИИ для решения медицинских задач относительно немного и в основном они направлены на решение различных онкологических заболеваний.

Существующие аналоги, включая вышеприведенные, являются смежными с разрабатываемой системой поддержки принятия решения острого панкреатита на уровне предметной области. Данные технологии акцентируют внимание на решении задач по постановке других диагнозов, но потенциально в будущем могут стать конкурентами с помощью постепенного расширения своего функционала. Однако данный фактор имеет низкую вероятность, а также уменьшается по степени влияния за счет принципа работы разрабатываемой системы: по окончании разработки веб-приложения Sapia можно будет использовать как напрямую, так и посредством интеграции в другие медицинские системы или стыковки с ними через API.

## 2 Данные и алгоритмы

Ядром в системах с применением ИИ выступает сложный математический базис, в частности, представляющий собой обученные модели, способы их использования, обучения, дообучения и т.д.

Для реализации подобного базиса необходимо пройти ряд этапов, начиная от сбора данных и заканчивая моделированием.

### 2.1 Выбор инструмента для анализа данных

Перед тем, как приступить к непосредственной работе с данными, необходимо определиться с основным инструментом, который будет использоваться на протяжении всех этапов работы с данными.

В настоящее время выделяют несколько наиболее популярных языков, которые можно использовать для анализа данных. К ним относят: Julia, R, Python и др.

Julia – высокоуровневый высокопроизводительный язык программирования с акцентом на параллельные и распределенные вычисления. Предназначен для проведения численного анализа, научных исследований. Данный инструмент с математическим уклоном также может использоваться для общего назначения – написания программ.

К особенностям и преимуществам языка программирования Julia относятся:

- динамическая типизация;
- высокая производительность в сравнение с типизированными языками;
- возможности метапрограммирования;
- вызов функций на Python с помощью PyCall;
- вызов функций на C напрямую;
- ориентация на параллельные и распределенные вычисления;

– возможность определять дополнительные типы, которые не уступают в скорости работы встроенным.

Недостатками этого языка являются: небольшая база реализованных алгоритмов для анализа данных, периодическая нестабильность при работе с пакетами Julia, небольшое сообщество разработчиков. Данные слабые стороны обусловлены тем, что язык программирования Julia является молодым относительно Python, R и в настоящее время только набирает свою популярность [10].

По части базы реализованных методов для различной обработки данных Julia, особенно, уступает языку программирования R.

R служит удобным инструментом для статистических вычислений, по работе с графическим отображением результатов. Данный мультипарадигменный язык является удобным инструментом для анализа данных.

Он включает в себя большое количество различных статистических и графических методов, в частности: линейное и нелинейное моделирование, классические статистические тесты, анализ временных рядов, классификацию, кластеризацию и др.

Однако данному языку программирования характерна низкая производительность и специфичность. R предоставляет широкий функционал для проведения различных исследований, однако он не предполагает своего применения для разработки программ в целом [11].

По общему использованию R уступает Julia и в особенности – Python. Python – высокоуровневый язык программирования общего назначения, ориентированный на повышение производительности разработчика и читаемости кода.

К преимуществам данного языка программирования относят:

- большое сообщество разработчиков;
- низкий порог вхождения;

– наличие крупных и качественно проработанных пакетов (Numpy, Pandas, Scipy, Scikit-learn, Tensorflow, Keras) для работы в области машинного обучения.

К явным недостаткам Python относят типобезопасность по отношению к тому же языку R, а также наличие двух параллельно развивающихся версий [12, 13].

Другие языки программирования имеют меньший уклон в область анализа данных и в настоящее время набирают необходимый для этого базовый инструментарий.

Из вышеизложенных инструментов для моделирования было принято решение использовать язык программирования Python по причине наличия большого числа различных пакетов для анализа данных, машинного обучения, низкого порога вхождения, а также потенциального использования при разработки серверной части веб-приложения.

## **2.2 Данные**

При разработке систем с применением технологии машинного обучения необходимо тщательно подходить к выбору данных и их обработке, т.к. сформированная выборка напрямую влияет на качество моделей распознавания.

### **2.2.1 Формат исходных данных**

Исходные истории болезни были предоставлены 20-й больницей г. Красноярска и краевой клинической больницей.

Возможности МИС qMS предполагают создание выходных xml-файлов, в которых в структурированном виде содержатся значения необходимых для обработки данных полей. Однако, ввиду низкой дисциплины заполнения историй, значительная часть данных вносилась как текст.

На момент извлечения данных в 20-ой больнице г. Красноярска отсутствовала синхронизация базой данных (БД) с лабораторными анализаторами и данные анализов также вносились вручную.

Ввиду вышеуказанных причин единственной возможностью учесть полный набор всей внесенной информации стал экспорт всех полей в единый текстовый документ, что несло в себе избыточную информацию.

### **2.2.2 Формат набора данных**

При согласовании с врачом производился отбор признаков, которые необходимо было извлечь из историй болезни, на основе чего был сформирован набор данных больных острым панкреатитом. Для дальнейших коммуникаций с экспертом выборка использовалась в формате \*.xlsx, для последующей работы с различными алгоритмами – \*.csv.

### **2.2.3 Описание набора данных**

Сформированная выборка в составе 130 объектов (историй болезни) была представлена в табличном формате и состояла из уникального идентификатора пациента, признаков, характеризующих состояние пациента, и экспертной оценки степени тяжести пациента.

Количество рассматриваемых признаков, с учетом экспертной оценки, составляет 27. Они представлены в формате двух шкал: численные (количественные) и ранговые.

К численным признакам относятся: температура, размер головки, размер тела, размер хвоста, гемоглобин, тромбоциты, эритроциты, лейкоциты, амилаза, аспартатаминотрансфераза, билирубин общий, глюкоза, калий, лимфоциты, моноциты, мочевины, натрий, скорость оседания эритроцитов, эозинофилы.

К ранговым признакам, включая экспертную оценку, относятся: контур, структура, эхогенность, дополнительные образования, перистальтика, вздутие



живота, жидкость в брюшной полости, факт перевода из другого ЛПУ (как отсутствие положительной динамики от назначенной терапии), степень тяжести.

Также выборка имеет 28-й признак – возраст, который не рассматривался при моделировании по причине наличия высокой доли пропусков (больше 50% от всего набора данных). По решению Заказчика данный показатель было принято оставить для дальнейшего импорта в БД и накопления новыми данными.

В качестве метки класса для каждого объекта установлена степень тяжести, которая имеет 3 категории: 1, 2, 3 – легкая, промежуточная (средняя) и тяжелая соответственно. Мощности классов в грубом соотношении 2:1:1 для классов 1, 2, 3 соответственно.

### **2.3 Выбор основного алгоритма**

В рамках работы с алгоритмами со стороны Заказчика параллельно велась работа с моделями классификации и регрессии, которые затем должны были быть включены в алгоритмическое ядро сервиса. В рамках выпускной квалификационной работы необходимо было выбрать один из популярных и потенциально подходящих алгоритмов для построения прогнозирующей модели.

Для решения задачи был взят метод опорных векторов [14]. Данный метод имеет большую популярность при решении различных прикладных задач, в частности, медицинских [15, 16].

Для более точного подбора прогнозирующей модели возникает необходимость в рассмотрении как линейных разделений классов, так и нелинейных. Метод опорных векторов удовлетворяет данному условию из-за своей специфики в подходе к решению задачи. Данный метод предлагает различные варианты нелинейных разделений, включая возможность подстройки и подбора своего варианта (функции).

### 2.3.1 Классификация или регрессия

На языке программирования Python в качестве основной использовалась библиотека Scikit-learn [17].

На основе метода опорных векторов в данной библиотеке реализованы как классификатор (SVC) [18, 19], так и регрессия (SVR) [20]. Ключевым гиперпараметром (параметром) у классификатора и регрессии является вид ядра. Оно позволяет строить нелинейные разделители в исходном признаковом пространстве и определяет основные последующие параметры для настройки и их количество.

Вид ядра напрямую влияет на процедуру повышения размерности признакового пространства, что в свою очередь оказывает влияние на результат работы классификатора и регрессии.

От выбранного ядра зависит количество предлагаемых параметров для дальнейшей настройки SVC, SVR. Реализация от Scikit-learn предлагает следующие ядра [21]:

- линейное;
- полиномиальное;
- радиально-базисную функцию;
- гиперболический тангенс.

Для предварительного выбора алгоритма было произведено обучение моделей SVC и SVR при одинаковых стандартных настройках ядер и рассмотрены результаты их прогнозирования с помощью точности классификации новых объектов.

Для решения задачи классификации при наличии трех классов на протяжении всего этапа моделирования рассматривались два стандартных подхода: «один против одного» (OvO) и «один против всех» (OvR).

Метод OvO предполагает обучение набора моделей, по количеству соответствующих числу классов, каждая из которых настроена на разделение своей уникальной, относительно других моделей, пары классов.

Метод OvR совпадает с OvO по количеству обучаемых моделей, но отличается по логике разделения классов – он предполагает разделение одного класса от всех остальных.

В качестве критерия качества для всех моделей использовалась оценка площади под кривой (AUC).

Для проведения расчета точности набор данных был разделен на обучающую и тестовую выборки в количестве 100 и 30 объектов соответственно. Количество рассматриваемых различных разбиений выборки в данном соотношении – 1000 [22].

В соответствии с вышеизложенными методиками при запуске SVC (таблица 1) и SVR (таблица 2) по методу OvR с одинаковыми стандартными настройками ядер точность сильно различалась между классификацией и регрессией в пользу последнего алгоритма.

Таблица 1 – Точность классификации при стандартных настройках ядер, все признаки для классификации

Ядро	1vR		2vR		3vR	
	Обучающая выборка	Тестовая выборка	Обучающая выборка	Тестовая выборка	Обучающая выборка	Тестовая выборка
Линейное	0,89299	0,83671	0,63887	0,50893	0,86599	0,81719
Радиально-базисная функция	0,831	0,80627	0,5	0,5	0,82239	0,81413
Полиномиальное	0,51423	0,51282	0,5	0,5	0,5	0,5
Гиперболический тангенс	0,82569	0,81015	0,5	0,5	0,65726	0,61718

Таблица 2 – Точность регрессии при стандартных настройках ядер, все признаки для регрессии

Ядро	1vR		2vR		3vR	
	Обучающая выборка	Тестовая выборка	Обучающая выборка	Тестовая выборка	Обучающая выборка	Тестовая выборка
Линейное	0,94018	0,87254	0,82091	0,57952	0,96145	0,89535
Радиально-базисная функция	0,93033	0,89208	0,87276	0,61167	0,96172	0,93034
Полиномиальное	0,91722	0,89341	0,89598	0,61688	0,95511	0,92874
Гиперболический тангенс	0,91784	0,89121	0,80678	0,58091	0,95328	0,93361

Более высокая точность метрики AUC в случае регрессий при обоих методах OvO и OvR объясняется принципом работы данной метрики в совокупности с работой регрессии и набором выдаваемых ей значений: регрессия выдает вещественное число, которое может стремиться к одной из двух заданных меток (например, 0 и 1), а метод расчета AUC позволяет выбирать наиболее оптимальную границу для интерпретации оценок регрессии в метки.

В обоих случаях дальнейший подбор параметров у классификатора, включая набор признаков, не приводят к улучшению точности выше 0,9, в отличие от регрессии.

Как результат, было решено применить методике мета-классификации. Мета-классификатор (далее классификатор) представляет собой двухступенчатую систему из регрессии, результаты которых выступают в роли метаданных, и стандартного классификатора.

## 2.4 Отбор информативных признаков

Количество объектов в выборке, а также количество признаков, позволяют проводить более детальный подбор признаков. Вместо использования случайного леса с популярным алгоритмом BorutaPy, анализа весов признаков на таких алгоритмах, как случайный лес, CatBoost [23] для отбора

информативных признаков применялся другой подход. Выборка объемом в 130 пациентов и 27-мерное признаковое пространство (без выходного признака) позволяет достаточно быстро провести отбор значимых признаков для SVR с тем или иным ядром с помощью оптимального прореживания [24]. Поэтапно из выборки выбрасывается тот признак, удаление которого оказывает наибольшее положительное влияние на ошибку (AUC увеличивается) или незначительное изменение ошибки в негативную сторону (AUC уменьшается), после чего выбирается тот набор значимых признаков, при котором AUC имеет максимальное значение.

Последовательность количества удаляемых признаков по принципу прореживания и динамика точности на тестовых выборках представлены на рисунке 2 на примере регрессий 1vR и 3vR. На рисунке 2 вертикальная ось отвечает за значение точности метрики AUC, горизонтальная ось демонстрирует связь уровня AUC с количеством исключенных признаков от полного их объема до одного. Для обеих регрессий 1vR и 3vR синим цветом отображен AUC после удаления признака, а красным – до удаления.

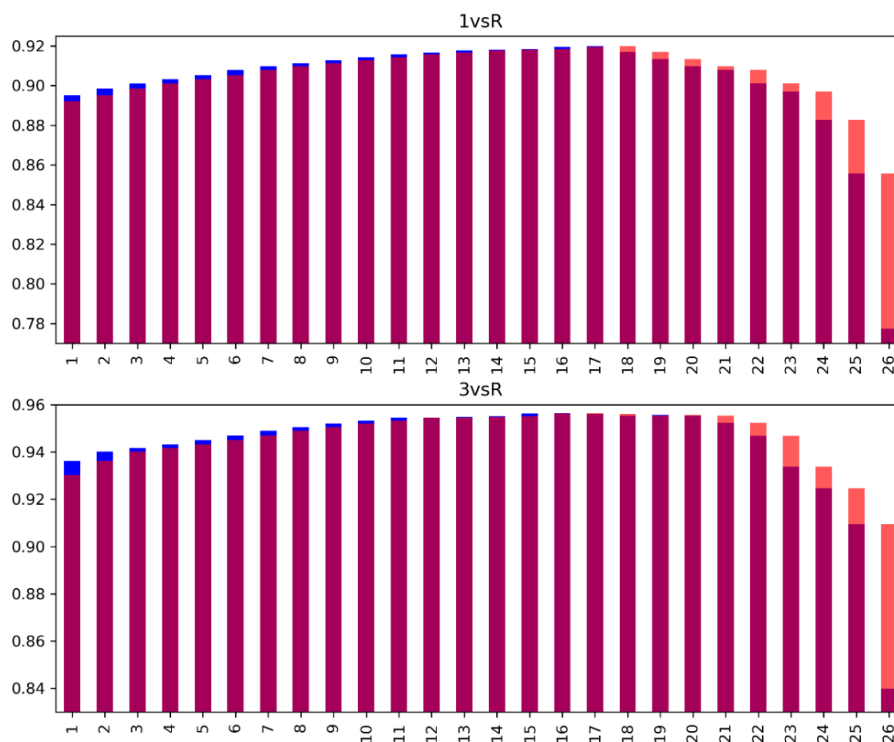


Рисунок 2 – Результат применения метода прореживания для регрессий

Все упомянутые виды ядер (таблица 1) отличаются по принципу преобразования размерности признакового пространства, из-за чего каждое ядро рассматривается отдельно.

Аналогичным образом, приведенным на рисунке 2, был произведен подбор значимых признаков для всех регрессий с различными подходами. В ходе подбора для шести регрессий (3 для OvR и 3 для OvO) результирующие наборы признаков отличаются как в оптимальном признаковом пространстве (количественно), так и в уникальности наборов признаков относительно друг друга (качественно). Метод OvO преимущественно имеет меньший набор признаков, чем метод OvR (лидирующей регрессией в данном плане является регрессия 1v3).

Для каждого ядра (таблица 1) и каждой регрессии OvR, OvO было проведено оптимальное прореживание (таблица 3).

Таблица 3 – Точность при стандартных настройках ядер, после отбора признаков

OvR						
Ядро	1vR		2vR		3vR	
	Обучающая выборка	Тестовая выборка	Обучающая выборка	Тестовая выборка	Обучающая выборка	Тестовая выборка
Линейное	0,93166	0,91174	0,78116	0,73043	0,96051	0,94701
Радиально-базисная функция	0,93565	0,92001	0,84338	0,77136	0,96992	0,95647
Полиномиальное	0,92712	0,92158	0,74711	0,73304	0,95666	0,95528
Гиперболический тангенс	0,92443	0,91732	0,82687	0,75253	0,96326	0,95688
OvO						
Ядро	1v2		1v3		2v3	
	Обучающая выборка	Тестовая выборка	Обучающая выборка	Тестовая выборка	Обучающая выборка	Тестовая выборка
Линейное	0,89931	0,86393	0,9935	0,98957	0,94783	0,91973
Радиально-базисная функция	0,89388	0,8622	0,99223	0,98812	0,93706	0,92092
Полиномиальное	0,87624	0,85944	0,99147	0,98927	0,95435	0,91763
Гиперболический тангенс	0,88449	0,86187	0,99159	0,98783	0,93183	0,92273

Разница результатов таблицы 1 и таблицы 3 демонстрируют важность и эффективность процесса отбора признаков пространства.

В ходе проведения оптимального прореживания для отбора значимых признаков результаты метрики AUC для всех регрессий значительно улучшились (таблица 3). Отсюда также следует, что по результатам отбора информативных признаков лидирует радиально-базисная функция для подхода OvR, тогда как для подхода OvO большая часть SVR имеет линейное ядро в качестве оптимального. Таким образом, выбор подхода рассмотрения решения задачи классификации (регрессии) может повлиять на выбор типа ядра.

## 2.5 Подбор параметров модели

По аналогии с этапом отбора информативных признаков подбор оптимальных параметров осуществлялся для всех четырех видов ядра параллельно.

Для подбора оптимальных параметров использовался многомерная сетка из значений параметров. Количество параметров у ядер напрямую определяло затрачиваемое время на подбор оптимальной комбинации значений параметров для конкретного ядра. По этой причине, например, линейное ядро имело меньший шаг между рассматриваемыми значениями признака в отличие от полиномиального.

Довольно быстро позволяют произвести подбор два вида ядра: линейное и радиально-базисная функция. В случае линейного ядра – одномерная сетка, радиально-базисная функция – двумерная сетка. Гиперболический тангенс требует больше времени и вычислительных мощностей, тогда как полиномиальное еще на порядок.

Для подстройки параметров с полиномиальным ядром использовался параллельный подбор для различных степенных показателей ядра: 2, 3, 4, 5, что облегчило процесс отладки в процессе перебора значений и ускорило сам процесс.

После подбора параметров были проведены расчеты показателя AUC для обучающей и тестовой выборок для каждого ядра (таблица 4).

Таблица 4 – Точность при оптимальных настройках ядер, после отбора признаков

OvR						
Ядро	1vR		2vR		3vR	
	Обучающая выборка	Тестовая выборка	Обучающая выборка	Тестовая выборка	Обучающая выборка	Тестовая выборка
Линейное	0,92801	0,91829	0,78144	0,73218	0,95691	0,95492
Радиально-базисная функция	0,94427	0,92273	0,86202	0,78053	0,97548	0,96291
Полиномиальное	0,92831	0,92496	0,85153	0,77859	0,95623	0,95694
Гиперболический тангенс	0,92130	0,91859	0,74421	0,73705	0,96326	0,95688
OvO						
Ядро	1v2		1v3		2v3	
	Обучающая выборка	Тестовая выборка	Обучающая выборка	Тестовая выборка	Обучающая выборка	Тестовая выборка
Линейное	0,89903	0,86407	0,99351	0,99089	0,94966	0,92353
Радиально-базисная функция	0,89432	0,8624	1	0,99377	0,94024	0,92357
Полиномиальное	0,87623	0,86476	0,99166	0,9903	0,95009	0,92473
Гиперболический тангенс	0,8993	0,86887	0,99036	0,98876	0,93172	0,92547

После подбора параметров приоритет ядер изменился: для метода OvR типы ядер остались прежними, а в случае метода OvO типы ядер для регрессий 1v2 и 1v3 сместились в сторону более сложной структуры (таблицы 4). В условиях небольшой выборки объемом в 130 объектов для определения точности проводился скользящий экзамен. Из таблицы 4 видно, что AUC для регрессий 2vR и 1v2 на порядок ниже. По этой причине далее за основу бралось двумерное пространство (две регрессии), где каждый объект, независимо от метки, подается на вход регрессиям. Данная методика возможна по причине небольшого



количества уникальных рангов выходного признака, а именно из-за наличия 3 классов.

В условиях небольшой выборки объемом в 130 объектов для определения точности проводился скользящий экзамен. В качестве оценки точности использовалась метрика AUC.

Скользящий экзамен основан на проведении  $N$  обучений, где  $N$  – количество объектов в выборке. За  $N$  итераций каждый раз поочередно выбрасывается один объект  $x$ , регрессии настраиваются на оставшихся  $N - 1$  объектах, а затем следует прогноз (оценки регрессии) для соответствующего объекта  $x$ .

Скользящий экзамен проводился для каждой из регрессий независимо. Для регрессии 1vR точность – 0,91851, для регрессии 3vR – 0,9628. Без учета 2 степени тяжести для регрессии 1vR точность – 0,97864, для регрессии 3vR – 0,99205. В случае подхода OvO для регрессии 1v3 точность – 0,99453, а для регрессии 2v3 – 0,98212. Результаты для последнего подхода выше из-за принципа расчета точности. Для регрессии OvO метрика AUC рассчитывалась только по тем объектам, классы которых соответствовали данной регрессии, тогда как для OvR в любом случае используется вся выборка.

С учетом полученных показателей точности (таблица 4), особенностях методов OvO и OvR при согласовании с Заказчиком было принято решение об использовании оценок регрессий 1vR и 3vR с радиально-базисной функцией в качестве ядра с параметрами, соответствующими таблице 4, в виде метаданных. Второй комбинацией регрессий рассматривалась пара 1v3 и 1v2, но регрессии 1vR и 3vR выигрывают за счет логики выделения косвенным путем 2 степени тяжести.

В качестве дополнительной меры для дальнейшего улучшения алгоритма, при согласовании с Заказчиком и экспертом предметной области было принято решение по идентификации 1 и 3 степени тяжести по причине низкой точности определения 2 степени тяжести на пробный период. Данное решение было предпринято в целях накопления обучающей выборки новыми данными и

проведения более тщательной оценки степени тяжести острого панкреатита за счет привлечения группы экспертов.

Результаты классификации для OvO и OvR при вышеуказанных условиях приведены на рисунках 3 и 4 соответственно. Разделяющая линия на рисунках 3 и 4 является разделяющей гиперплоскостью линейного классификатора. Классы разнесены по цветам: 1 класс – зеленый цвет, 2 класс – желтый цвет, 3 класс – красный цвет.

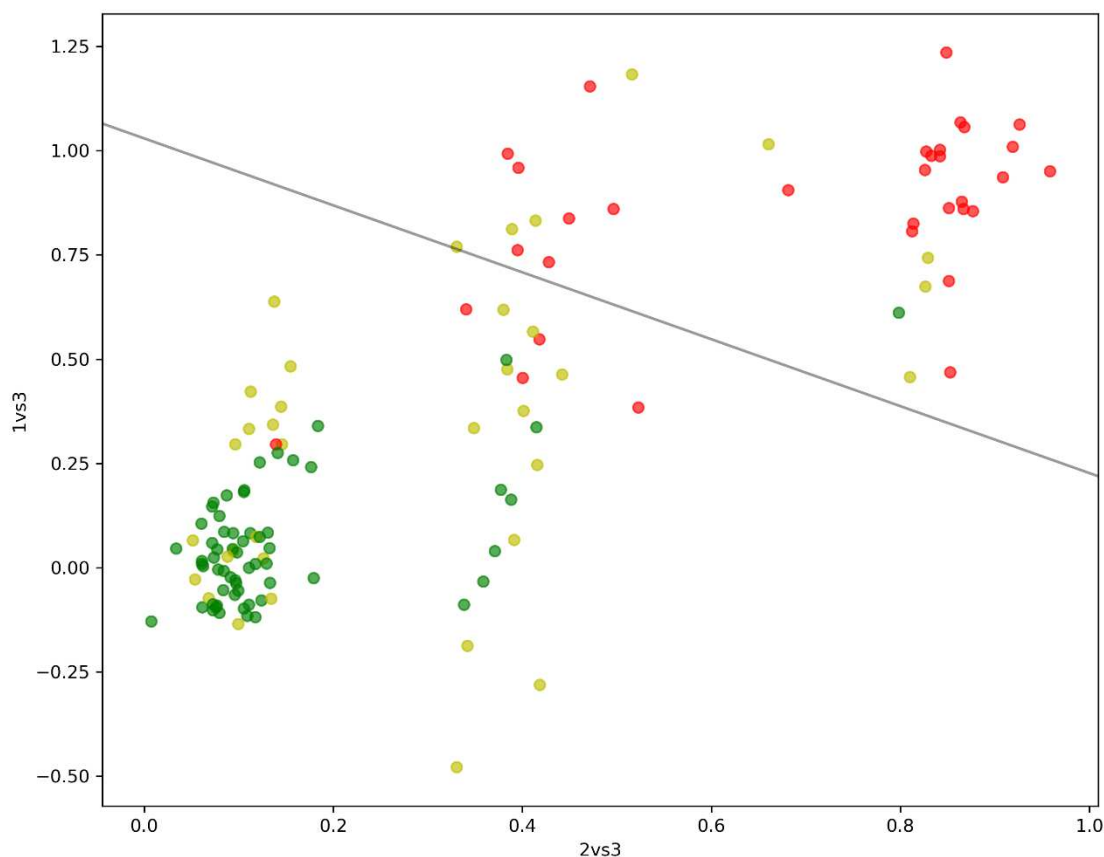


Рисунок 3 – Классификация на основе метаданных, метод «один против одного»

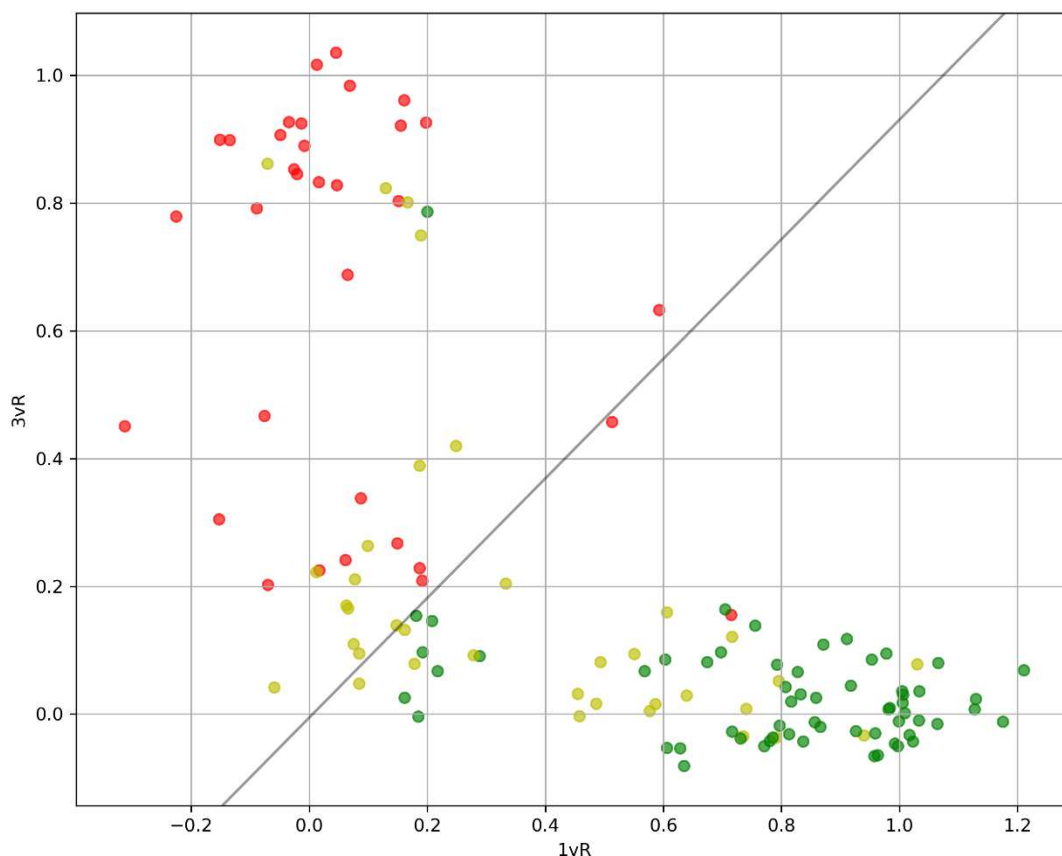


Рисунок 4 – Классификация на основе метаданных, метод «один против всех»

В дальнейшем в качестве значимых признаков для регрессии  $1vR$  будут использоваться: дополнительные образования, гемоглобин, лейкоциты, амилаза, аспаратаминотрансфераза, билирубин общий, калий, натрий, перистальтика, жидкость в брюшной полости. Для регрессии  $3vR$ : температура, размер головки, размер хвоста, структура, эритроциты, лейкоциты, амилаза, моноциты, скорость оседания эритроцитов, жидкость в брюшной полости, перевод из другого ЛПУ.

На момент построения моделей экспертом был предоставлен еще один признак – креатинин. По отношению к нему по аналогии была произведена попытка добавления данного атрибута, но по методу оптимального прореживания он показал отрицательное воздействие и не был включен в конечный набор признаков, используемых регрессиями.

Вместе с полученными моделями по требованию Заказчика был сформирован регрессор по всем трем степеням тяжести для более гибкого отображения процесса перехода состояния пациента из одной степени тяжести в

другую. Также данный регрессор был введен в целях дальнейшего перехода на формирование решения, основанного на нечеткой логике.

### **3 Выбор средств разработки**

Перед любой разработкой программного продукта необходимо определиться со стеком используемых технологий. Данный этап является в важным в жизненном цикле продукта, т.к. определяет некоторые его особенности: скорость реализации проекта, сложность расширения, например, за счет внедрения нового персонала, скорость работы системы в целом, уровень безопасности и т.д.

Специфика системы Sapia заключается в выборе инструментов с расчетом на периодическое проведение исследований в области моделирования данных, разработки новых моделей прогнозирования и их внедрения.

#### **3.1 Серверная часть**

Существует множество инструментов для разработки серверной части: Node.js, .NET, Ruby on Rails, Flask/Django и др.

Node.js – событийно-ориентированный язык программирования. Он позволяет разработчикам использовать язык программирования JavaScript для написания инструментов командной строки и сценариев на стороне сервера.

Node.js часто используется для разработки простых одностраничных приложений (SPA), к основным преимуществам которого часто относят асинхронность, высокую скорость работы. Также отличительной особенностью выделяется возможность быстрого переключения и сочетания разработки клиентской и серверной частей из-за специфики использования языка программирования JavaScript в обоих случаях. Но в условиях разрабатываемого веб-приложения возникает большая необходимость в объемных библиотеках машинного обучения с включенными специфичными инструментами по обработке данных [25].

.NET Core, который представляет собой более новую платформу на базе C# относительно .NET Framework – это универсальная платформа разработки,

которая поддерживается корпорацией Microsoft и сообществом .NET на сайте GitHub. Она является кроссплатформенной, поддерживает Windows, macOS и Linux и может использоваться на устройствах, в виртуальной среде с удаленным доступом, во внедренных системах и в сценариях Интернета вещей.

Перечисленные ниже особенности наиболее полно определяют платформу .NET Core:

- гибкая разработка;
- кроссплатформенность;
- программы командной строки;
- совместимость платформы .NET Core с .NET Framework, Xamarin и Mono благодаря .NET Standard;
- открытый исходный код;
- платформа .NET Core поддерживается корпорацией Microsoft.

Не смотря на свои преимущества, в основном технологию .NET Core в основном применяют для реализации крупных проектов, где требуется высокая устойчивость к большой нагрузке на сервер. По этой причине NET Core не является приоритетным в выборе инструментов разработки серверной части [26].

Потенциальным конкурентом Node.js является Python в связке с веб-фреймворками, среди которых наиболее популярными считаются Flask и Django.

Фреймворк для разработки веб-сайтов Flask относится к категории микрофреймворков. В случае написания небольших проектов Flask крайне удобен и прост в использовании, но при дальнейшем расширении могут возникнуть трудности по реализации из-за, например, плохо спроектированной архитектуры системы. Фреймворк Flask предоставляет самые базовые возможности в сравнении с Django.

Django – это веб-среда Python высокого уровня, которая способствует быстрой разработке и чистому, прагматичному дизайну. Он предлагает более жесткую архитектуру. При использовании Django сайт строится из одного или нескольких приложений, которые рекомендуется делать отчуждаемыми и

подключаемыми. Это одно из существенных архитектурных отличий данного фреймворка от Ruby on Rails и других.

Данный веб-фреймворк имеет хорошо написанную документацию, автоматически генерируемую административную панель, поддержку паттерна проектирования «Модель-Шаблон-Представление» (MTV) и т.д. Этот инструмент веб-разработки использовался в таких проектах, как: YouTube, Instagram, Google Search, DropBox, Reddit, Spotify и т.д.

Django предоставляет большой набор инструментов для веб-разработчика. Например, при работе над проектом Django предоставляет защиту от таких ошибок, как:

- инъекции SQL;
- кросс-сайт подлоги;
- clickjacking;
- кросс-сайтовый скриптинг [27].

Из всех вышеперечисленных инструментов для разработки серверной части приоритетными выделяются Node.js и Django. Но с учетом того выбора языка программирования Python в качестве инструмента для работы с моделями, в частности, классификаторами и регрессиями, было принято решение о включении в стек технологий для разработки веб-приложения «Sapia» веб-фреймворк Django.

### **3.2 Клиентская часть**

Существует множество инструментов для разработки клиентской части веб-приложения. Как правило, выделяют следующие: фреймворк AngularJS, библиотеку ReactJS и библиотеку Vue.

AngularJS – это JavaScript-фреймворк, который ориентирован на разработку SPA. Он нацелен на расширения приложений на основе шаблона

«Модель-Представление-Контроллер», упрощение процесса разработки и тестирования.

AngularJS основан на стандартном JavaScript и языке гипертекстовой разметки (HTML), поэтому он автоматически решает задачу по манипуляциям с объектной моделью документа (DOM) и с технологией обращения к серверу без перезагрузки страницы. Инструмент предоставляет модульные строительные блоки кода JS, которые можно совмещать.

К преимуществам AngularJS относят:

- позволяет привязывать данные к HTML с помощью выражений, а директивы AngularJS позволяют разработчикам расширять функционал HTML и создавать новые конструкции;

- архитектура AngularJS универсальна и подходит под реализацию широкого спектра веб-приложений;

- позволяет управлять клиентской и серверной частями проекта с помощью JavaScript;

- модульность AngularJS позволяет облегчить процесс распределения работ по нескольким участникам проекта.

Наряду с преимуществами AngularJS имеет ряд недостатков:

- низкая гибкость;

- не приспособлен обрабатывать манипуляции с DOM с большим количеством данных;

- не поддерживает высоконагруженные галереи фотографий и другие проблемы с производительностью [28].

Данный фреймворк AngularJS по производительности и наличию большого количества потенциально лишнего функционала уступает ReactJS и Vue.

JavaScript-библиотека ReactJS предназначена для разработки пользовательских интерфейсов. Он разрабатывается и поддерживается Facebook, Instagram и сообществом отдельных разработчиков и корпораций. Данная



библиотека характеризуется высокой скоростью, простотой и масштабируемостью.

К преимуществам ReactJS относят:

- простой синтаксис;
- высокий уровень гибкости и максимальная отзывчивость;
- связывание данных от больших к меньшим, при котором дочерние элементы не могут влиять на родительские данные;
- простая миграция между версиями.

ReactJS считается хорошим инструментом для разработки клиентской части веб-приложения и пользуется большой популярностью, однако и у него есть ряд недостатков, к которым, например, относят:

- неполнота документации;
- в определенных случаях разработчики имеют большой выбор, что в дальнейшем может привести к путанице в понимании процесса работы программы;
- высокий порог вхождения [29].

Не смотря большую популярность JavaScript-библиотека ReactJS, он не является единственным лидером в выборе для разработки веб-приложения по прогнозированию степени тяжести острого панкреатита. Ее конкурентом выступает более молодой аналог – фреймворк Vue.

JavaScript-фреймворк Vue предназначен для создания пользовательских интерфейсов. Особенностью является легкая интегрируемость в проекты с использованием других JavaScript-библиотек. Может функционировать как веб-фреймворк для разработки одностраничных приложений в реактивном стиле. JavaScript-фреймворк Vue относительно схожа JavaScript-фреймворком Angular по характеристикам, но при этом по масштабам является более компактной.

Среди преимуществ Vue выделяют следующие:

- усиленный HTML;
- масштабирование;

– хорошо написанная и подробная документация, уменьшающая время обучения разработчиков;

– адаптивность Vue позволяет совершить быстрый перевод проекта с одного фреймворка на другой по причине архитектурного и дизайнерского сходства с AngularJS и ReactJS;

– удобная интеграция, позволяющая Vue использовать как для создания одностраничных приложений, так и для более сложных веб-приложений;

– низкие требования к объему памяти наряду с высокой скоростью работы.

В качестве недостатков Vue выделяют риск чрезмерной гибкости, что может возникнуть в ходе разработки больших проектов [30].

По причине отсутствия потенциально опасных рисков со стороны Vue к разрабатываемому продукту, низкого порога вхождения и высокой адаптивности было принято решение о разработке клиентской части с применением JavaScript-фреймворк Vue.

### **3.3 База данных**

В настоящее время существует множество различных систем управления базами данных (СУБД) [31]. Но несмотря на то, что все СУБД выполняют одну и ту же задачу, подход к ее выполнения сильно варьируется в широких пределах. Так, например, была исключена СУБД MySQL [32] по причине того, что она требует большие временные затраты на реализацию несложных задач. В случае MySQL – создание инкрементных резервных копий.

СУБД от компании Oracle не подходит по причине высокой стоимости, а также есть риск необходимости модернизировать оборудование для внедрения СУБД. По ресурсоемкости проигрывает и СУБД от Microsoft.

В итоге, в качестве СУБД была выбрана PostgreSQL. PostgreSQL – это объектно-реляционная система управления базами данных, основанная на

POSTGRES. В POSTGRES появилось множество новшеств, которые были реализованы в некоторых коммерческих СУБД гораздо позднее.

В случае с PostgreSQL минусом можно выделить документацию, однако к плюсам относятся: простой процесс импорта данных, масштабируемость, поддержка формата JavaScript Object Notation (JSON) и наличие свободной лицензии.

PostgreSQL поддерживает большую часть стандарта SQL и предлагает множество современных функций:

- сложные запросы;
- внешние ключи;
- триггеры;
- изменяемые представления;
- транзакционная целостность;
- многоверсионность.

Также пользователи могут расширять возможности PostgreSQL, например, создавая свои: типы данных, функции, операторы, агрегатные функции, методы индексирования и процедурные языки.

Благодаря свободной лицензии, PostgreSQL разрешается бесплатно использовать, изменять и распространять всем и для любых целей – личных, коммерческих или учебных [33].

## **4 Требования**

В процессе обсуждений с Заказчиком были выявлены и структурированы требования к функционалу и процессу работы разрабатываемого продукта.

### **4.1 Цели создания системы**

Целью данного проекта является разработка системы, которая должна привести к следующим изменениям:

- повысить объективность постановки степени тяжести острого панкреатита на момент поступления больного;
- упростить процесс прогнозирования степени тяжести за счет учета возможности подгрузки данных из МИС;
- предоставить возможность объективно оценить степень тяжести в первые сутки поступления больного в отличие от традиционных третьих;
- предоставить доступ к прогнозу районными больницами, где отсутствуют более совершенные средства диагностики;
- повысить качество накапливаемой выборки;
- предоставить возможность накопления данных о состоянии пациента в динамике;
- предоставить эксперту возможность поставить итоговую степень тяжести.

### **4.2 Требования к разрабатываемой системе**

Разрабатываемая система должна соответствовать ниже приведенным требованиям:

- данный продукт должен быть реализован в формате клиент-серверного приложения (веб-приложения);

- необходимо разработать систему распознавания степени тяжести острого панкреатита с помощью внесенных значений показателей, характеризующих состояние пациента;
- разрабатываемая программа должна иметь форму ввода значений показателей пациента;
- конечный продукт должен содержать обученные модели и использовать для прогнозирования состояния пациента;
- в условиях уменьшения признакового пространства необходимо добиться максимальной точности прогнозирования;
- разрабатываемая программа должна предоставлять максимально компактный формат результатов прогнозирования;
- данная программа должна предоставлять возможность ведения динамики состояния пациента с минимальной дискретностью – 1 сутки;
- по мере использования веб-приложение должно накапливать данные;
- разрабатываемая система не должна хранить конфиденциальной информации о пациентах, либо данные подобного характера должны быть одобрены Заказчиком;
- конечный продукт по отношению к пациентам должен хранить уникальный идентификатор с датой регистрации показателей, значения показателей, степень тяжести эксперта, результаты работы модели или моделей;
- при разработке программы для экспертов данной области должны проводиться презентации результатов для постепенного внедрения в медицинскую сферу.

## 5 Проектирование

После проведения анализа предметной области, обработав данные и обучив модели, выбрав инструменты разработки, необходимо спроектировать разрабатываемую систему, с учетом указанных требований со стороны Заказчика.

### 5.1 База данных

На рисунке 5 представлена логическая схема БД.

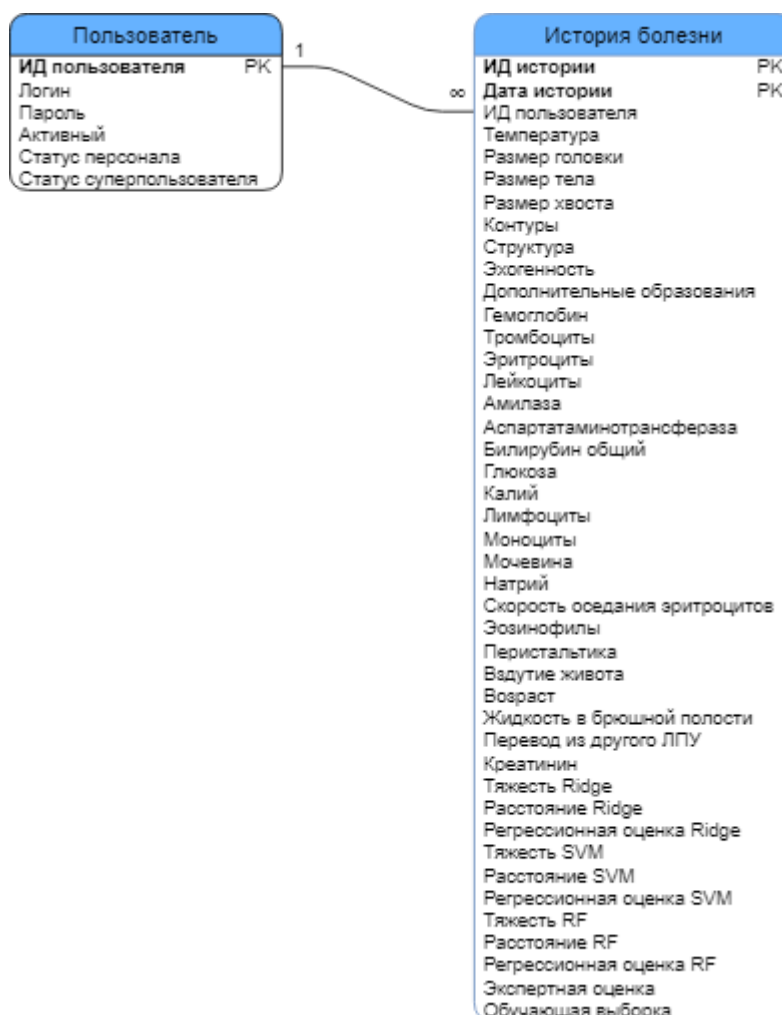


Рисунок 5 – Логическая схема базы данных

БД разрабатываемого веб-приложения предназначена для решения следующих задач:

- накопления выборки для дальнейшей настройки моделей;
- предоставления возможности ведения состояния пациента в динамике;
- ограничения работы на уровне зарегистрированного пользователя;
- связки добавленной истории с конкретным врачом посредством идентификатора пользователя.

Таблица пользователей необходима для ограничения доступа к системе в целом. Только авторизовавшийся пользователь получает доступ к взаимодействию с прогнозирующими моделями посредством взаимодействия с графическим интерфейсом. Также идентификатор пользователя используется для привязки к истории конкретного пользователя.

Основной сущностью является таблица по историям болезни, представляющая собой расширенную выборку. Данная таблица хранит идентификатор истории (ИД истории), дату снятия показателей пациента, идентификатор пользователя, который добавил конкретную запись в БД (ИД пользователя), значения показателей пациента, результаты прогнозов, степень тяжести врача (экспертная оценка) и логическое поле принадлежности к обучающей выборке (обучающая выборка).

К значениям показателей пациента относятся: температура, размер головки, размер тела, размер хвоста, контуры, структура, экзогенность, дополнительные образования, гемоглобин, тромбоциты, эритроциты, лейкоциты, амилаза, аспаратаминотрансфераза, билирубин общий, глюкоза, калий, лимфоциты, моноциты, мочевины, натрий, скорость оседания эритроцитов, эозинофилы, перистальтика, вздутие живота, возраст, жидкость в брюшной полости, перевод из другого ЛПУ, креатинин.

К результатам прогнозов в рамках выпускной квалификационной работы относятся: степень тяжести SVR, расстояние SVR, регрессионная оценка SVR.

По требованию Заказчика в систему также включены аналогичные приведенным результаты по гребневой регрессии и случайному лесу. Работа по данным алгоритмам проходила со стороны Заказчика.

## 5.2 Архитектура

Архитектура веб-приложения следует архитектуре веб-фреймворка Django (рисунок 6), представляющая собой паттерн MTV.

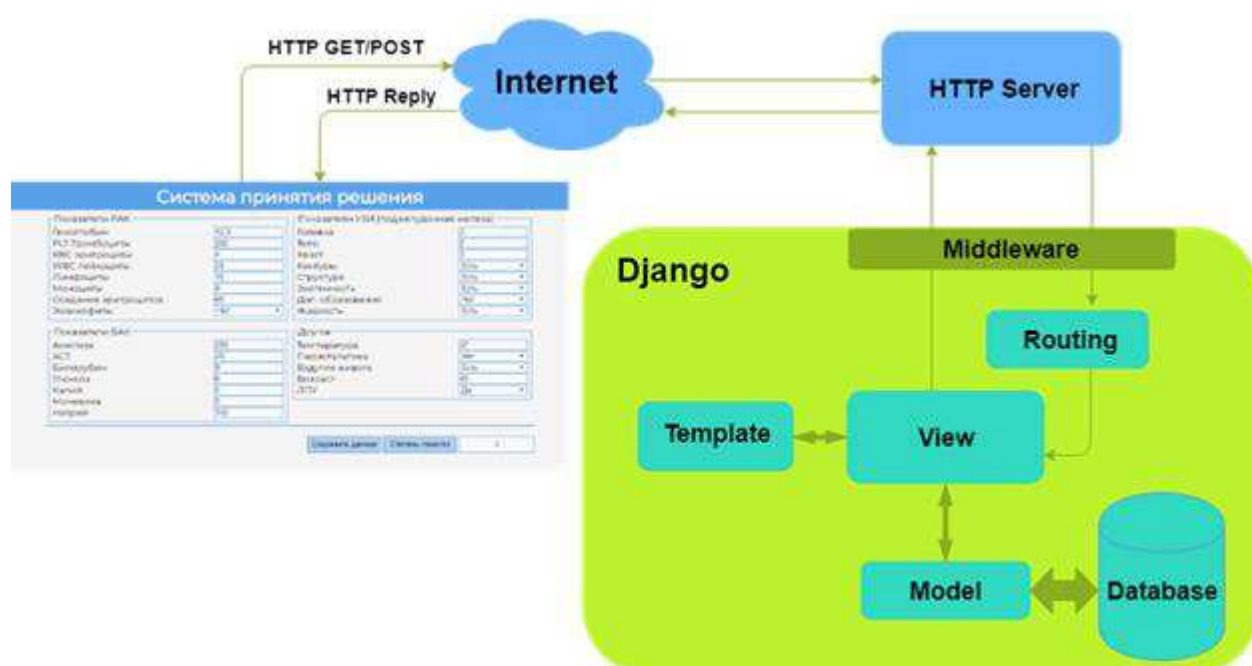


Рисунок 6 – Архитектура веб-приложения

В Django модель представляет собой слой уровня доступа к данным при помощи собственного объектно-реляционного отображения, которая позволяет работать с записями таблицы на объектном уровне. Модель соответствует логической схеме базы данных (рисунок 5).

Шаблоны являются слоем представления данных. Этот слой принимает решения относительно представления данных: как и что должно отображаться на странице или в другом типе документа.



Представление – слой бизнес-логики, описывающий, как получать доступ к моделям и применять соответствующий шаблон.

На верхнем уровне веб-фреймворк Django представляет собой разработку сайта с декомпозицией на приложения.

По причине малых масштабов разрабатываемого продукта и нелогичности разведения таблиц БД разрабатываемый веб-сайт представляет собой одно приложение «Панкреатит», которое соответствует рисунку 6.

### 5.3 Применение моделей

Обученные модели требуют правильного использования для получения верных прогнозов соответствующих их логике. По этой причине, все процессы с их использованием происходят на стороне сервера, а готовый результат выдается клиентской части. В свою очередь клиентская часть формирует набор данных и отправляет запрос серверу для получения прогноза.

Схема выполнения прогноза приведена на рисунке 7.

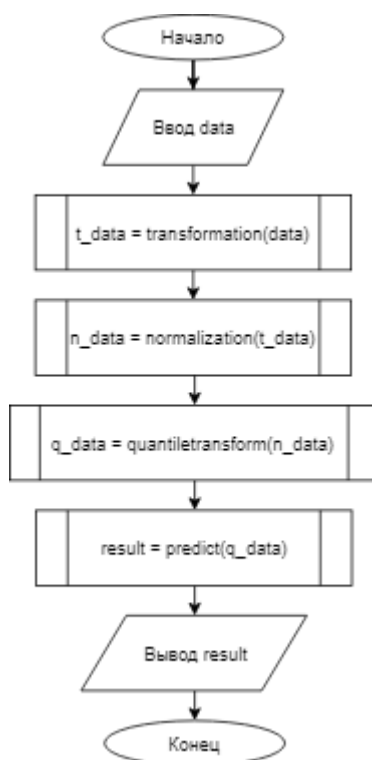


Рисунок 7 – Прогнозирование степени тяжести

На рисунке 7 схема, направленная на определение степени тяжести острого панкреатита, поверхностно отображает последовательность вызова ряда функций: проверки и преобразования значений в числовой вид (transformation), нормализации данных (normalization), применения квантильного преобразования к некоторым признакам (quantiletransform) и, соответственно, прогноз с помощью обученных моделей (predict). Элементы ввода и вывода имитируют поступление запроса с новыми данными и отправку ответа с результатами моделей.

Функция проверки и преобразования значений предназначена для стыковки и выступает в роли дополнительной защиты, если на сервер пришли данные в строковом формате.

Функция нормализации отвечает за процедуру приведения значений поступившей истории болезни в пределы  $[0; 1]$  относительно заданных пределов – минимумов и максимумов для каждого численного признака.

Квантильное преобразование применяется для преобразования распределения ряда численных признаков в равномерное.

Блок прогнозирования включает в себя получение обученных моделей и передачу им сформированных данных для получения класса (степени тяжести) и/или регрессионной оценки.

Ключевым требованием при использовании моделей является строгое соблюдение выполнения всех процессов, изложенных на рисунке 7. Допустимо объединение нескольких блоков в одну функцию на уровне реализации, но с указанной последовательностью.

Блок прогнозирования является расширяемым. По мере развития проекта, данный блок может включать в себя новые обученные модели и иные алгоритмы распознавания, настроенные под конкретный формат данных. Также, блок прогнозирования предусматривает возможность распараллеливания выполнения процессов по прогнозированию на различных моделях с целью уменьшения времени отклика.

## 5.4 Прикладной программный интерфейс

Для работы с данными, включая модели, необходимо спроектировать API – составить список возможных запросов к серверу.

Для взаимодействия с моделями необходимо пройти аутентификацию и авторизацию, а также должна быть предоставлена возможность выхода из системы. Для этого в API входят методы входа и выхода из системы соответственно.

Непосредственно для работы с основным функционалом веб-приложения в API также должны входить:

- запрос на получение прогноза новой истории болезни;
- запрос на получение прогноза существующей истории болезни;
- запрос на сохранение значений показателей с прогнозом;
- запрос для поиска идентификатора истории по префиксу;
- запрос на получение списка дат для конкретного идентификатора;
- запрос на получение последних значений показателей для соответствующего идентификатора истории болезни;
- запрос на получение значений показателей для соответствующего идентификатора истории болезни и даты;
- запрос на обновление экспертной оценки для соответствующего идентификатора истории болезни и даты.

## **6 Разработка**

Одним из ключевых этапов проекта являлась разработка системы, сопутствующая всему жизненному циклу проекта. В данный этап входили:

- создание БД;
- формирование классов в слое доступа к данным;
- заполнение БД;
- разработка серверной части;
- разработка клиентской части.

Все основные процессы программы поддержки принятия решения для диагностирования степени тяжести острого панкреатита базируются на использовании данных по пациентам. По этой причине перед разработкой основного функционала системы необходимо было создать БД и привести ее в рабочее состояние.

### **6.1 Создание БД и заполнение**

Для работы с данными были выполнены следующие задачи: создана БД и настроен доступ к ней, а затем импортирована выборка.

После установки PostgreSQL был создан Django-проект, добавлен один привилегированный пользователь, изменены настройки подключения к БД, прописана модель по работе с данными по острому панкреатиту (историями болезни) и выполнены сгенерированные миграции.

В результате при запуске проекта и переходе на стандартную страницу администратора от Django, открывается возможность выполнения CRUD-операций – операций по созданию, считыванию, обновлению и удалению объектов. На рисунке 8 приведен интерфейс для взаимодействия с таблицей истории болезни.

Пользователь:	admin ▼ ✎ +
Идентификатор:	<input type="text"/>
Температура:	<input type="text"/>
Головка:	<input type="text"/>
Тело:	<input type="text"/>
Хвост:	<input type="text"/>
Контуры:	<input type="text"/>
Структура:	<input type="text"/>
Эхогенность:	<input type="text"/>
Доп.образования:	<input type="text"/>
Гемоглобин:	<input type="text"/>
Тромбоциты:	<input type="text"/>
Эритроциты:	<input type="text"/>
Лейкоциты:	<input type="text"/>
Амилаза:	<input type="text"/>

Рисунок 8 – Администрирование Django

С помощью скрипта на языке Python и настроек для подключения к БД со стороны Django-проекта была выполнена процедура импорта выборки острого панкреатита. В качестве исходной информации выступал файл с расширением \*.csv, который использовался во время работы с данными, в частности, для

обучения моделей (рисунок 9). В процессе работы скрипта была импортирована выборка в БД (рисунок 10).

	505	510	511	513	514	515	516	518	523	524	526	527	534	535	539	540	541	542	545	548	557	567	heavy	age
0	36.6	3.000000	2.000000	1.0	1.0	1.0	0.0	103.0	386.0	3.92	24.83	362.00	4.88	3.080	10.8	9.0	5.10	140.0	65.00	1.3	1.0	0.0	2.0	35.0
1	37.5	2.300000	1.500000	1.0	1.0	1.0	0.0	65.0	195.0	2.12	7.68	27.00	5.63	3.130	15.1	9.0	5.90	142.0	20.00	0.1	0.0	0.0	3.0	54.0
2	37.6	6.100000	2.850000	1.0	1.0	1.0	1.0	111.0	317.0	3.48	33.68	242.00	6.14	3.110	5.4	6.9	3.90	147.0	38.00	0.0	0.0	1.0	3.0	48.0
3	36.7	2.300000	1.600000	1.0	0.0	1.0	0.0	137.0	344.0	5.09	9.88	48.00	5.89	4.790	23.5	10.8	5.10	139.0	15.00	0.7	1.0	0.0	1.0	58.0
4	37.0	2.758830	2.863175	1.0	1.0	1.0	0.0	172.0	262.0	5.38	13.53	68.00	5.25	4.350	21.1	8.3	3.30	140.0	46.00	1.0	1.0	0.0	1.0	28.0
5	36.6	3.000000	1.600000	1.0	1.0	0.0	0.0	152.0	266.0	4.85	15.60	172.00	13.15	4.390	16.5	4.0	5.50	130.0	41.00	1.0	1.0	0.0	2.0	48.0
6	37.4	3.300000	3.600000	1.0	1.0	0.0	0.0	89.0	139.0	2.90	11.13	134.00	7.05	4.590	16.7	17.0	14.80	135.0	56.00	0.1	0.0	0.5	3.0	40.0
7	37.5	4.300000	2.900000	1.0	1.0	1.0	1.0	124.0	487.0	4.40	8.06	36.00	10.43	4.570	16.1	8.3	4.50	130.0	45.00	1.2	1.0	0.5	3.0	51.0
8	37.3	2.900000	1.600000	1.0	1.0	1.0	0.0	145.0	215.0	5.16	12.69	47.00	5.59	4.510	21.9	11.5	5.10	138.0	18.00	2.4	1.0	0.0	3.0	56.0
9	36.6	3.500000	2.400000	1.0	1.0	1.0	0.0	163.0	211.0	5.20	10.13	28.00	13.65	3.990	15.8	7.5	3.30	131.0	13.00	0.9	1.0	0.5	1.0	39.0
10	37.2	2.000000	1.800000	1.0	1.0	0.0	0.0	138.0	272.0	4.58	8.26	103.00	6.63	4.560	17.4	20.5	3.60	132.0	31.00	2.3	1.0	0.0	3.0	43.0
11	37.3	2.800000	2.600000	1.0	1.0	0.0	0.0	101.0	319.0	5.37	41.64	91.00	12.90	3.190	2.5	6.8	8.60	137.0	37.00	0.5	1.0	1.0	3.0	65.0
12	37.8	3.500000	1.500000	1.0	1.0	1.0	1.0	116.0	331.0	4.37	13.63	33.00	15.37	2.530	9.5	9.8	3.40	136.0	55.00	1.0	0.0	0.5	3.0	70.0
13	36.7	3.000000	2.000000	1.0	1.0	0.0	0.0	132.0	294.0	3.84	11.04	189.00	12.50	4.150	11.5	15.5	2.00	141.0	43.00	0.8	1.0	0.0	1.0	36.0
14	37.3	4.100000	2.056058	1.0	1.0	1.0	1.0	171.0	321.0	6.23	27.03	32.00	6.28	3.190	2.1	3.3	7.30	135.0	3.00	0.0	0.0	0.5	3.0	68.0
15	37.8	2.500000	2.000000	1.0	1.0	0.0	0.0	146.0	198.0	5.02	5.07	312.00	6.37	4.130	6.5	10.1	9.00	136.0	53.00	0.2	0.0	0.5	3.0	55.0
16	37.6	2.500000	1.400000	1.0	1.0	1.0	0.0	127.0	392.0	4.47	10.06	134.00	14.35	2.270	7.8	9.1	13.60	129.0	25.00	0.6	0.0	0.5	3.0	70.0
17	37.1	3.700000	2.700000	1.0	1.0	0.0	0.0	82.0	109.0	2.72	8.72	26.00	4.82	3.660	8.7	5.8	31.40	136.0	31.00	0.6	0.0	0.0	3.0	29.0
18	37.6	3.200000	1.700000	1.0	1.0	1.0	0.0	117.0	324.0	4.16	20.75	2960.00	7.83	2.720	3.4	7.3	6.30	134.0	58.00	0.2	1.0	0.5	2.0	47.0
19	36.9	4.000000	2.900000	1.0	1.0	1.0	0.0	96.0	191.0	3.10	8.79	137.00	9.48	3.300	5.6	6.9	10.70	132.0	72.00	0.3	1.0	1.0	3.0	59.0
20	37.4	3.000000	1.800000	1.0	1.0	1.0	0.0	108.0	235.0	3.10	15.00	54.00	9.68	3.870	6.3	11.1	6.00	130.0	66.00	0.4	1.0	0.0	3.0	45.0
21	36.7	3.100000	1.900000	1.0	1.0	0.0	1.0	125.0	513.0	4.27	7.71	59.00	6.59	4.090	27.9	10.8	2.20	138.0	44.00	2.2	1.0	0.0	2.0	68.0
22	36.7	3.000000	1.600000	1.0	1.0	1.0	0.0	129.0	252.0	4.52	8.41	114.00	7.29	3.380	33.2	7.1	4.80	131.0	46.00	2.4	1.0	0.0	3.0	68.0
23	37.0	3.200000	2.200000	1.0	1.0	0.0	0.0	61.0	337.0	1.86	20.17	566.00	5.74	2.930	4.1	6.8	14.20	137.0	78.00	0.1	0.0	0.5	3.0	62.0
24	38.3	3.200000	2.500000	1.0	1.0	0.0	0.0	121.0	120.0	3.66	11.40	1034.00	6.93	3.700	9.0	3.0	5.70	135.0	50.00	1.0	0.0	0.0	3.0	31.0
25	36.7	4.800000	3.600000	1.0	1.0	1.0	1.0	104.0	147.0	3.13	13.21	1177.00	8.13	5.140	6.0	4.5	10.40	135.0	52.00	1.1	0.0	1.0	3.0	45.0
26	36.8	2.100000	0.800000	1.0	0.0	1.0	0.0	124.0	178.0	4.52	3.04	53.00	5.35	4.720	48.4	7.9	5.60	146.0	13.00	1.6	1.0	0.0	2.0	81.0
27	37.1	1.600000	1.200000	1.0	1.0	1.0	0.0	132.0	226.0	4.64	4.23	85.00	5.70	3.840	37.1	6.4	5.00	137.0	18.00	0.7	1.0	1.0	1.0	61.0
28	36.8	2.452244	2.000000	1.0	1.0	0.0	1.0	129.0	503.0	4.45	14.86	304.00	4.90	5.100	15.8	6.7	5.70	144.0	43.00	15.7	1.0	0.0	2.0	40.0
29	36.8	3.500000	2.000000	1.0	1.0	1.0	1.0	137.0	209.0	5.01	9.26	76.00	5.31	4.930	40.0	7.9	4.70	137.0	24.00	1.9	1.0	0.0	2.0	52.0
30	37.3	3.600000	2.700000	1.0	1.0	1.0	0.0	88.0	465.0	4.67	13.45	73.00	6.37	3.380	4.7	4.4	8.30	142.0	40.00	0.0	1.0	0.0	3.0	43.0

Рисунок 9 – Часть данных из исходной выборки

temperature	head	body	tail	rbc	wbc	amylase	ast	bilirubin	glucose	potassium	lymphocytes	monocytes	urea	abdominaldist
double precis	doubl	doub	dout	doub	doubl	double p	dou	double p	double p	double pre	double precis	double prec	doub	double precisi
37.6	6.1	2.85	3	3.48	33.68	242	32.2	20.5	6.14	3.11	5.4	6.9	3.9	1
37.4	3.3	3.6	3.2	2.9	11.13	134	57.3	32.7	7.05	4.59	16.7	17	14.8	0.5
37.5	4.3	2.9	3728	4.4	8.06	36	17.5	7.6	10.43	4.57	16.1	8.3	4.5	0.5
37.3	2.9	1.6	3.9	5.16	12.69	47	38.8	7.6	5.59	4.51	21.9	11.5	5.1	0
36.6	3.5	2.4	1.96	5.2	10.13	28	24.6	14.7	13.65	3.99	15.8	7.5	3.3	0.5
37.2	2	1.8	0836	4.58	8.26	103	16.9	9.6	6.63	4.56	17.4	20.5	3.6	0
37.3	2.8	2.6	9912	5.37	41.64	91	30	22	12.9	3.19	2.5	6.8	8.6	1
37.8	3.5	1.5	2.7	4.37	13.63	33	18.5	10.6	15.37	2.53	9.5	9.8	3.4	0.5
36.7	3	2	6307	3.84	11.04	189	31	22.1	12.5	4.15	11.5	15.5	2	0
37.3	4.1	36058	4	6.23	27.03	32	25.2	19.6	6.28	3.19	2.1	3.3	7.3	0.5
37.8	2.5	2	2.5	5.02	5.07	312	56.8	52.6	6.37	4.13	6.5	10.1	9	0.5
37.6	2.5	1.4	3.5	4.47	10.06	134	17.7	10.7	14.35	2.27	7.8	9.1	13.6	0.5
37.1	3.7	2.7	3.2	2.72	8.72	26	15	5.6	4.82	3.66	8.7	5.8	31.4	0

Рисунок 10 – Пример записей истории болезни из локальной БД

## 6.2 Разработка серверной части

Разработка серверной части, в основном, предполагает работу с представлением (views.py). Однако она оперирует с данными из БД посредством

запросов, прописанных в `api.py`, и с помощью вспомогательных функций, представляющих собой разработанный математический базис или ядро.

Для различных дополнительных вычислений была выделена директория «`preprocessing`», а для работы с двухступенчатым классификатором – директория «`classifiers`». Данные разделения не несут смысловой нагрузки, они выполнены в целях исключения дублирования программного кода и соответствуют схеме Django-приложения (рисунок 6), т.к. в совокупности представляют собой слой представления.

В первой директории определены два файла с расширением `*.py`: файл всех стандартных списков атрибутов, используемых программой и, непосредственно, файл с дополнительными вычислениями.

Списки атрибутов представляют собой определенного характера комбинацию признаков, описанных ранее, результаты работы обученных моделей и дополнительную идентифицирующую информацию.

К стандартным спискам атрибутов относятся: список всех признаков и прогнозов, соответствующих полям таблицы истории болезни (рисунок 5), список численных признаков, список ранговых признаков, а также списки признаков, соответствующих каждой используемой регрессии.

К дополнительным вычислениям относятся: проверка формата данных и, при необходимости, приведение к соответствующему виду, нормализация данных по списку численных признаков и другие дополнительные преобразования.

В корне слоя представления `views.py` содержится функционал по обработке запросов. Он содержит реализованные функции по:

- прогнозированию степени тяжести;
- авторизации и аутентификации;
- получению необходимых страниц для взаимодействия с системой;
- сохранению всего набора данных с прогнозом;
- поиску идентификатора по полученному префиксу;

- получению списка существующих дат для конкретного идентификатора;
- получению последних значений показателей для соответствующего идентификатора истории болезни;
- получению значений показателей для соответствующего идентификатора истории болезни и даты;
- обновлению экспертной оценки для соответствующего идентификатора истории болезни и даты.

Первостепенными запросами являются запросы на получение страницы, предполагающие работу слоев представления и шаблона, а также на авторизацию с аутентификацией, необходимую для получения доступа к основному функционалу системы.

Основными или ключевыми элементами функционала выступают возможности по прогнозированию степени тяжести острого панкреатита и сохранению результатов.

Запрос на получение прогноза требует передачу в json-формате 29 значений показателей пациента, соответствующей им даты и идентификатора истории болезни (рисунок 11).

```
{
  abdominaldist: "Есть", addformations: "Нет",
  age: "61", amylase: "85", ast: "22.1",
  bilirubin: "11", body: "1.2", contours: "Есть"
  creatinine: "65", datetime: "04.05.2019",
  echogenicity: "Есть", eosinophils: "0.7",
  esr: "18", glucose: "5.7", head: "1.6",
  hgb: "132", liquid: "Есть", lpu: "Нет",
  lymphocytes: "37.1", monocytes: "6.4",
  patientID: "3123", peristalsis: "Есть",
  plt: "226", potassium: "3.84", rbc: "4.64",
  sodium: "137", structure: "Есть", tail: "1.7",
  temperature: "37.1", urea: "5", wbc: "4.23"
}
```

Рисунок 11 – Пример данных в запросе для прогноза

Ответ на запрос с рисунка 11 представлен на рисунке 12.



```
{ridge_heavy: 1, ridge_dist: 0.301, svr_heavy: 1, svr_dist: 0.462, rfr_heavy: 1, ...}
  rfr_dist: 0.56
  rfr_heavy: 1
  rfr_rgr: 1.63
  ridge_dist: 0.301
  ridge_heavy: 1
  ridge_rgr: 1.56
  svr_dist: 0.462
  svr_heavy: 1
  svr_rgr: 1.58
```

Рисунок 12 – Пример ответа от сервера на запрос по прогнозу

Запрос на сохранение включает все передаваемые значения из запроса на получение прогноза, но при этом также должны быть переданы результаты самого прогноза (рисунок 13).

```
{
  abdominaldist: "Есть", addformations: "Нет",
  age: "61", amylase: "85", ast: "22.1",
  bilirubin: "11", body: "1.2", contours: "Есть",
  creatinine: "65", datetime: "01.05.2019",
  echogenicity: "Есть", eosinophils: "0.7",
  esr: "18", glucose: "5.7", head: "1.6",
  hgb: "132", liquid: "Есть", lpu: "Нет",
  lymphocytes: "37.1", monocytes: "6.4",
  patientID: "TEST", peristalsis: "Есть",
  plt: "226", potassium: "3.84", rbc: "4.64",
  rfr_dist: "0.56", rfr_heavy: "1", rfr_rgr: 1.63,
  ridge_dist: "0.301", ridge_heavy: "1", ridge_rgr: 1.56,
  sodium: "137", structure: "Есть", svr_dist: "0.462",
  svr_heavy: "1", svr_rgr: 1.58, tail: "1.7",
  temperature: "37.1", urea: "5", wbc: "4.23"
}
```

Рисунок 13 – Пример набора значений для сохранения

Запросы на получение списка идентификаторов, соответствующих отправленному на сервер префиксу, и списка дат, соответствующих отправленному идентификатору, хранят префикс идентификатора и идентификатор соответственно. В качестве ответа при поиске конкретного идентификатора сервер возвращает список значений из БД, удовлетворяющих условию префикса. Для запроса по получению списка дат сервер возвращает все существующие в БД даты для полученного идентификатора.

Для обновления или сохранения экспертной оценки необходимо отправить соответствующий запрос, передав идентификатор, дату показаний и степень тяжести (по мнению эксперта).

### 6.3 Разработка клиентской части

Разработка клиентской части включала в себя написание разметки, ее стилизацию и внесение интерактивности при взаимодействии с серверной частью.

В целом, клиентская часть представляет собой следующие страницы: авторизация, работа с новой историей болезни или с обновлением существующих, определение экспертной оценки степени тяжести.

Перед началом работы в системе пользователю необходимо авторизоваться через соответствующее окно (рисунок 14).

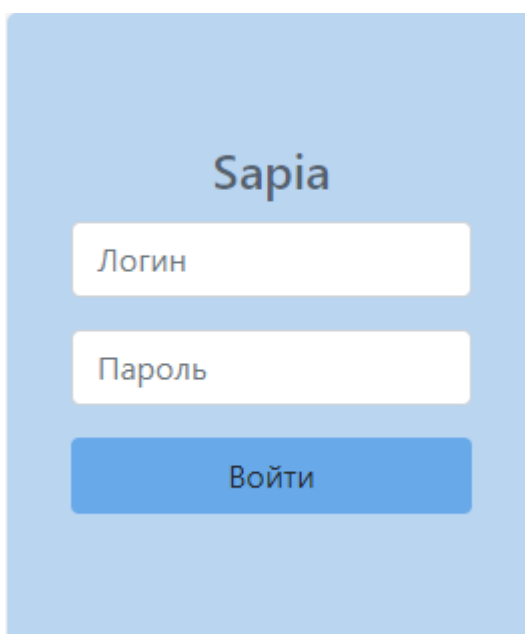


Рисунок 14 – Окно авторизации

После авторизации пользователю открывается страница по работе с пациентом на вкладке добавления новой истории болезни (рисунок 15).

**Система поддержки принятия решения**

Saripа Пациент Экспертная оценка Выход

Новая история Обновить историю

Показатели РАК

Гемоглобин		Лимфоциты	
Тромбоциты		Моноциты	
Эритроциты		СОЭ	
Лейкоциты		Эозинофилы	

Показатели Б/Х крови

Амилаза		Калий	
АСТ		Креатинин	
Билирубин общий		Мочевина	
Глюкоза		Натрий	

Показатели УЗИ

Размер головки		см	Структура	Нет
Размер тела		см	Эхогенность	Нет
Размер хвоста		см	Доп. образ.	Нет
Контур	Нет		Жидкость в б/п	Нет

Дополнительно

Дата		Вздутие живота	Нет
Возраст		Перистальтика	Нет
Температура		Перевод из ЛПУ	Нет

Очистить	Сброс	
Медиана(ы)	Прогноз	
Сохранить		
Модель	Степень	Расстояние
Ridge		
SVM		
RF		

Модель	Оценка	Перрепсор
Ridge		
SVM		
RF		

Рисунок 15 – Страница по работе с пациентом, вкладка добавления новой истории болезни

Для получения прогноза на странице новой истории пользователю необходимо заполнить все представленные поля (рисунок 15) и нажать на кнопку «Прогноз» для получения результатов работы моделей (рисунок 16).

**Система поддержки принятия решения**

Saripа Пациент Экспертная оценка Выход

Новая история Обновить историю

ВЭ911

Показатели РАК

Гемоглобин	132	Лимфоциты	37,1
Тромбоциты	226	Моноциты	6,4
Эритроциты	4,64	СОЭ	18
Лейкоциты	4,23	Эозинофилы	0,7

Показатели Б/Х крови

Амилаза	85	Калий	3,84
АСТ	22,1	Креатинин	65
Билирубин общий	11	Мочевина	5
Глюкоза	5,7	Натрий	137

Показатели УЗИ

Размер головки	1,6	см	Структура	Есть
Размер тела	1,2	см	Эхогенность	Есть
Размер хвоста	1,7	см	Доп. образ.	Нет
Контур	Есть		Жидкость в б/п	Есть

Дополнительно

Дата	01.04.2019	Вздутие живота	Есть
Возраст	61	Перистальтика	Есть
Температура	37,1	Перевод из ЛПУ	Нет

Очистить	Сброс	
Медиана(ы)	Прогноз	
Сохранить		
Модель	Степень	Расстояние
Ridge	1	0.302
SVM	1	0.453
RF	1	0.56

Модель	Оценка	Перрепсор
Ridge	1.55	
SVM	1.59	
RF	1.65	

Рисунок 16 – Прогноз нового пациента

Для обновления состояния пациента необходимо перейти на вкладку обновления истории и, с помощью поиска, выбрать один из существующих

идентификаторов, что в свою очередь выполнит дополнительный запрос на получение последних данных по пациенту, а затем отобразит их (рисунок 17).

**Система поддержки принятия решения**

Саріа Пациент Экспертная оценка Выход

Новая история Обновить историю

ВЭ2063/16

Показатели РАК			
Гемоглобин	137	Лимфоциты	22
Тромбоциты	234	Моноциты	8
Эритроциты	6	СОЭ	11
Лейкоциты	3,7	Эозинофилы	1

Показатели УЗИ			
Размер головки	3	Структура	Есть
Размер тела	2,1	Эхогенность	Есть
Размер хвоста	3,1	Доп. образ.	Нет
Контур	Есть	Жидкость в бл	Нет

Показатели Б/Х крови			
Амилаза	687	Калий	4,9
АСТ	110	Креатинин	80
Билирубин общий	21	Мочевина	7,9
Глюкоза	5,2	Натрий	148

Дополнительно			
Дата	11.05.2016	Вздутие живота	Нет
Возраст	75	Перистальтика	Есть
Температура	36,6	Перевод из ЛПУ	Нет

Очистить	Сброс	
Медиана(ы)	Прогноз	
Сохранить		
Модель	Степень	Расстояние
Ridge	1	1.064
SVM	1	0.89
RF	1	1

Модель	Оценка	Перспектив
Ridge	1.06	
SVM	1.04	
RF	1.33	

Рисунок 17 – Обновление истории болезни

После получения последних данных из истории болезни пользователь должен отредактировать поля под новую дату, нажать на кнопку «Прогноз», а затем, после получения прогноза, нажать на кнопку «Сохранить» для сохранения данных в БД.

Кроме основных кнопок «Прогноз» и «Сохранить» пользователю представлен функционал по очистке всех полей и сброса результатов прогноза с помощью кнопок «Очистить» и «Сброс» соответственно, а также кнопка «Медиана(ы)», предоставляющая возможность восполнения пропущенных полей за счет медиан по признакам.

Кнопка «Медиана(ы)» была внесена в систему по причине наличия случаев, когда эксперт не имеет всех показателей. Например, к таким могут относиться размеры поджелудочной железы (размер головки, тела или хвоста). Данная возможность является временным решением и при дальнейшем развитии проекта может быть удалена или заменена на другое решение.

При наличии в БД историй болезни пользователь может определить экспертную оценку, перейдя на страницу экспертной оценки (рисунок 18).

**Система поддержки принятия решения**

Sarìa      Пациент   Экспертная оценка   Выход

Идентификатор

Дата

1 степень тяжести

**Обновить**

**Показатели РАК**

Гемоглобин	<input type="text"/>	Лимфоциты	<input type="text"/>
Тромбоциты	<input type="text"/>	Моноциты	<input type="text"/>
Эритроциты	<input type="text"/>	СОЭ	<input type="text"/>
Лейкоциты	<input type="text"/>	Эозинофилы	<input type="text"/>

**Показатели Б/Х крови**

Амилаза	<input type="text"/>	Калий	<input type="text"/>
АСТ	<input type="text"/>	Креатинин	<input type="text"/>
Билирубин общий	<input type="text"/>	Мочевина	<input type="text"/>
Глюкоза	<input type="text"/>	Натрий	<input type="text"/>

**Показатели УЗИ**

Размер головки	<input type="text"/>	см	Структура	<input type="text"/>
Размер тела	<input type="text"/>	см	Эхогенность	<input type="text"/>
Размер хвоста	<input type="text"/>	см	Доп. образ.	<input type="text"/>
Контур	<input type="text"/>	Нет	Жидкость в б/п	<input type="text"/>

**Дополнительно**

Дата	<input type="text"/>	Вздутие живота	<input type="text"/>
Возраст	<input type="text"/>	Перистальтика	<input type="text"/>
Температура	<input type="text"/>	Перевод из ЛПУ	<input type="text"/>

Рисунок 18 – Окно определения экспертной оценки

Перед определением степени тяжести по мнению эксперта пользователю необходимо, посредством поиска, выбрать один из существующих идентификаторов, после чего выбрать одну из полученных дат. При наличии значений полей идентификатора и даты система автоматически отправляет запрос на получения соответствующих значений показателей и затем отображает их (рисунок 19).

**Система поддержки принятия решения**

Sarìa      Пациент   Экспертная оценка   Выход

ВЭ2012/16

07.07.2017

Текущая: 1 степень тяжести

2 степень тяжести

**Обновить**

**Показатели РАК**

Гемоглобин	132	Лимфоциты	20
Тромбоциты	211	Моноциты	4
Эритроциты	4,65	СОЭ	32
Лейкоциты	14,3	Эозинофилы	3

**Показатели Б/Х крови**

Амилаза	62	Калий	5,4
АСТ	29	Креатинин	89
Билирубин общий	6,8	Мочевина	4,4
Глюкоза	5,8	Натрий	138

**Показатели УЗИ**

Размер головки	3,9	см	Структура	Есть
Размер тела	2,5	см	Эхогенность	Есть
Размер хвоста	3,6	см	Доп. образ.	Нет
Контур	Есть		Жидкость в б/п	Нет

**Дополнительно**

Дата	07.07.2017	Вздутие живота	Нет
Возраст	61	Перистальтика	Есть
Температура	36,6	Перевод из ЛПУ	Нет

Рисунок 19 – Экспертная оценка

После выбора идентификатора истории болезни и даты эксперту отображается сохраненная на последний момент степень тяжести и возможность ее изменить с помощью другой степени тяжести и нажатием кнопки «Обновить» (рисунок 19).

## ЗАКЛЮЧЕНИЕ

В ходе работы над проектом было разработано математическое ядро, а затем внедрено в разработанное веб-приложение для диагностирования степени тяжести острого панкреатита.

Математический базис представляет собой ряд отдельно функционирующих алгоритмов классификации и регрессии, а также надстройку над ними для использования в шаблонном формате. В частности, в состав математического базиса входит метод опорных векторов.

Разработанная система поддержки принятия решения предоставляет оценку состояния пациента с острым панкреатитом в формате классовой принадлежности с дополнительным расчетом меры близости к выбранному классу, а также в формате регрессионной оценки.

В настоящий момент времени веб-приложение функционирует в тестовом режиме на удаленном сервере от компании ООО «Ар Ди Сайнс». Зарегистрировано несколько аккаунтов для врачей и параллельно с развитием проекта эксперты вносят данные по пациентам за 2019 год.

Впоследствии накопленные данные планируется включить в обучающую выборку с дальнейшим дообучением моделей в целях повышения точности прогноза. Реализованный функционал, предоставляющий возможность регистрации данных по одному и тому же пациенту во временной шкале, предполагается использовать для развития системы в направлении отслеживания динамики состояния пациента.

## СПИСОК СОКРАЩЕНИЙ

В данной работе используются следующие сокращения:

БД – база данных;

ИИ – искусственный интеллект;

МИС – медицинская информационная система;

СУБД – система управления базами данных;

API – прикладной программный интерфейс;

AUC – площадь под кривой;

DOM – объектная модель документа;

HTML – язык гипертекстовой разметки;

JSON – JavaScript Object Notation;

MTV – паттерн «Модель-Шаблон-Представление»;

OvO – «один против одного»;

OvR – «один против всех»;

SPA – одностраничное приложение;

SVC – классификатор на основе метода опорных векторов;

SVR – регрессия на основе метода опорных векторов.



## СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Черданцев Д. В., Арутюнян А. В., Арутюнян А. В. Состояние проблемы острого панкреатита в г. Красноярске и Красноярском крае //Acta Biomedica Scientifica. – 2011. – №. 4-2.
2. Черданцев Д. В. и др. ПРОБЛЕМА ОСТРОГО ПАНКРЕАТИТА В КРАСНОЯРСКОМ КРАЕ //Современные проблемы науки и образования. – 2019. – №. 2. – С. 151-151.
3. Roberts S. E. et al. The incidence and aetiology of acute pancreatitis across Europe //Pancreatology. – 2017. – Т. 17. – №. 2. – С. 155-165.
4. Ризаев К.С., Хаджибаев Ф.А., Баймурадов Ш.Э. Определение степени тяжести течения и лечения острого панкреатита // Материалы пленума правления ассоциации гепатопанкреатобилиарных хирургов стран СНГ - 2015 г. стр.123-125.
5. Hoyt R. E. et al. IBM Watson analytics: automating visualization, descriptive, and predictive statistics //JMIR public health and surveillance. – 2016. – Т. 2. – №. 2. – С. e157.
6. Ross C., Swetlitz I. IBM pitched its Watson supercomputer as a revolution in cancer care. It's nowhere close //STAT News. – 2017.
7. Bass D. Microsoft Develops AI to Help Cancer Doctors Find the Right Treatments //Bloomberg. Archived from the original on. – 2017. – Т. 11.
8. Bloch B. N., Lenkinski R. E., Rofsky N. M. The role of magnetic resonance imaging (MRI) in prostate cancer imaging and staging at 1.5 and 3 Tesla: the Beth Israel Deaconess Medical Center (BIDMC) approach //Cancer Biomarkers. – 2008. – Т. 4. – №. 4-5. – С. 251-262.
9. Chu C. et al. Applying machine learning to automated segmentation of head and neck tumour volumes and organs at risk on radiotherapy planning CT and MRI scans //F1000Research. – 2016. – Т. 5.
10. Bezanson J. et al. Julia: A fast dynamic language for technical computing //arXiv preprint arXiv:1209.5145. – 2012.

11. Ihaka R., Gentleman R. R: a language for data analysis and graphics //Journal of computational and graphical statistics. – 1996. – Т. 5. – №. 3. – С. 299-314.
12. Sanner M. F. et al. Python: a programming language for software integration and development //J Mol Graph Model. – 1999. – Т. 17. – №. 1. – С. 57-61.
13. Pedregosa F. et al. Scikit-learn: Machine learning in Python //Journal of machine learning research. – 2011. – Т. 12. – №. Oct. – С. 2825-2830.
14. Вапник В. Н., Червоненкис А. Я. Теория распознавания образов: статистические проблемы обучения. – Наука. Гл. ред. физ.-мат. лит., 1974.
15. Chen Y. J. et al. Chinese traditional medicine recognition by support vector machine (SVM) terahertz spectrum //Guang pu xue yu guang pu fen xi= Guang pu. – 2009. – Т. 29. – №. 9. – С. 2346-2350.
16. Zhang J. et al. SVM-based ultrasonic medicine image diagnosis //Medical Image Acquisition and Processing. – International Society for Optics and Photonics, 2001. – Т. 4549. – С. 92-96.
17. Hackeling G. Mastering Machine Learning with scikit-learn. – Packt Publishing Ltd, 2017.
18. Guyon I., Boser B., Vapnik V. Automatic capacity tuning of very large VC-dimension classifiers //Advances in neural information processing systems. – 1993. – С. 147-155.
19. Cortes C., Vapnik V. Support-vector networks //Machine learning. – 1995. – Т. 20. – №. 3. – С. 273-297.
20. Smola A. J., Schölkopf B. A tutorial on support vector regression //Statistics and computing. – 2004. – Т. 14. – №. 3. – С. 199-222.
21. Fan R. E. et al. LIBLINEAR: A library for large linear classification //Journal of machine learning research. – 2008. – Т. 9. – №. Aug. – С. 1871-1874.
22. Воронцов К. В. Математические методы обучения по прецедентам (теория обучения машин) //Москва. – 2011. – С. 119-121.
23. Dorogush A. V., Ershov V., Gulin A. CatBoost: gradient boosting with categorical features support //arXiv preprint arXiv:1810.11363. – 2018.

24. Загоруйко Н. Г. Когнитивный анализ данных //Новосибирск: Академическое издательство ГЕО. – 2013.
25. Cantelon M. et al. Node. js in Action. – Greenwich : Manning, 2014. – С. 17-20.
26. Freeman A. Pro Asp. net Core Mvc. – Apress, 2016.
27. Forcier J., Bissex P., Chun W. J. Python web development with Django. – Addison-Wesley Professional, 2008.
28. Darwin P. B., Kozlowski P. AngularJS web application development. – Packt Publ., 2013.
29. Vipul A. M., Sonpatki P. ReactJS by Example-Building Modern Web Applications with React. – Packt Publishing Ltd, 2016.
30. Filipova O. Learning Vue. js 2. – Packt Publishing Ltd, 2016.
31. Исаченко А. Н., Бондаренко С. П. Модели данных и СУБД. – 2007.
32. DuBois P., Foreword By-Widenius M. MySQL. – New riders publishing, 1999.
33. PostgreSQL D. PostgreSQL //Web resource: [http://www. PostgreSQL. org/about](http://www.PostgreSQL.org/about). – 2010.

## **ПРИЛОЖЕНИЕ А**

### **Справка о сдаче в тестовое использование**





## **ПРИЛОЖЕНИЕ Б**

### **Свидетельство о государственной регистрации**

Федеральное государственное автономное  
образовательное учреждение  
высшего образования  
«СИБИРСКИЙ ФЕДЕРАЛЬНЫЙ УНИВЕРСИТЕТ»

Институт космических и информационных технологий  
институт

«Информатика»  
кафедра

УТВЕРЖДАЮ

Заведующий кафедрой

А. С. Кузнецов

подпись

инициалы, фамилия

« *аск* »

*07*

2019 г.

**БАКАЛАВРСКАЯ РАБОТА**

09.03.04 Программная инженерия  
код и наименование специальности

Разработка системы поддержки принятия решения при диагностировании  
степени тяжести острого панкреатита  
тема

Руководитель

*А.С. Михалев* 05.07.19

подпись, дата

старший преподаватель

должность, ученая степень

А. С. Михалев

инициалы, фамилия

Выпускник

*Мелех* 05.07.19

подпись, дата

Д. А. Мелех

инициалы, фамилия

Консультант

*Сергеева* 04.08.18

подпись, дата

ген. дир., канд. техн. наук

должность, ученая степень

Н. А. Сергеева

инициалы, фамилия

Нормоконтролер

*Антамошкин* 05.07.19

подпись, дата

доцент, канд. техн. наук

должность, ученая степень

О. А. Антамошкин

инициалы, фамилия

Красноярск 2019