МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
«АЛТАЙСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»
Институт математики и информационных технологий
Кафедра информатики

МЕДИЦИНСКАЯ ИНФОРМАЦИОННАЯ СИСТЕМА ДЛЯ КОНТРОЛЯ И ПРОГНОЗИРОВАНИЯ ГИПОПИТУИТАРИЗМА У ДЕТЕЙ И ПОДРОСТКОВ

выпускная квалификационная работа

Выполнил:
студент группы 474б,
Русаков Михаил Олегович
_____
(подпись)

Научный руководитель:
к.т.н., доцент
Хворова Любовь Анатольевна
_____
(подпись)

Допустить к защите:
зав. кафедрой, к.ф.-м.н., доцент
Козлов Денис Юрьевич
_____
(подпись)
«___»_____ 2021 г.

Работа защищена:
«___»_____ 2021 г.

Оценка: _____

Председатель ГЭК:
_____
(подпись)
_____

Барнаул 2021

РЕФЕРАТ

Тема выпускной квалификационной работы: «Медицинская информационная система для контроля и прогнозирования гипопитуитаризма у детей и подростков».

Цель исследования – разработка медицинской информационной системы, позволяющей осуществлять контроль и прогнозирование течения гипопитуитаризма в детском возрасте.

Объект исследования – данные медицинского обследования детей и подростков Алтайского края, страдающих гипопитуитаризмом.

Предмет исследования – технологии проектирования и разработки информационных систем и поддержки принятия врачебных решений.

В результате работы решены следующие задачи: извлечены числовые характеристики из таблиц; найдены в тексте упоминания медицинских концептов; извлечены фрагменты текстовых данных; извлечены числовые характеристики из текста; создана база данных на основе извлеченной информации; составлена структура и реализован программный интерфейс медицинской информационной системы; проведен корреляционный анализ зависимости атрибутов; построены и обучены модели прогнозирования; проведена оценка качества полученных моделей.

Для извлечения и обработки данных, а также для построения моделей машинного обучения выбран высокоуровневый язык программирования Python. Для разработки функционала информационной системы и её графического интерфейса выбраны объектно-ориентированный язык программирования C# и технология создания событийно-ориентированных приложений Windows Form.

Ключевые слова: медицинская информационная система, искусственный интеллект, машинное обучение, интеллектуальные системы поддержки принятия решений, гипопитуитаризм.

Дипломная работа состоит из введения, двух глав, заключения, списка использованной литературы и приложений. Работа изложена на 32 страницах

компьютерного текста, включает 6 рисунков, 2 таблицы, 1 приложение и 23 источника литературы.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

В настоящее время информационные технологии, технологии искусственного интеллекта и анализа больших данных активно используются в различных сферах человеческой деятельности. Наиболее высокий экономический и социальный эффект от применения современных информационных технологий достигается в областях, где при принятии решений анализируется большое количество данных, а модели принятия решений сложны для понимания одним человеком. Область здравоохранения обладает этими свойствами. В настоящее время здравоохранение переживает целый ряд перемен, например, внедрение электронных медицинских карт (ЭМК). Значительный рост цифровых медицинских данных предоставляет большие возможности для применения методов анализа данных и разработки инновационных ИТ-решений. Оснащенные искусственным интеллектом инструменты способны выявлять значимые закономерности в данных и могут применяться во всех областях медицины, включая принятие врачебных решений, разработку лекарственных препаратов, уход за пациентами и другие [6,15].

Гипопитуитаризм – эндокринное заболевание, связанное со сниженной или отсутствующей секрецией гормонов гипофиза. Наиболее часто у детей и подростков диагностируют дефицит гормона роста (ДГР), который характеризуется низкой скоростью роста [1].

Целью проводимого исследования является разработка медицинской информационной системы, позволяющей осуществлять контроль и прогнозирование течения гипопитуитаризма в детском возрасте.

Объект исследования – данные медицинского обследования детей и подростков Алтайского края, страдающих гипопитуитаризмом.

Предмет исследования – технологии проектирования и разработки информационных систем и поддержки принятия врачебных решений.

Разрабатываемая медицинская информационная система (МИС) объединяет систему для хранения и обработки медицинской информации в цифровой форме и интеллектуальную систему поддержки принятия врачебных решений.

Актуальность и практическая значимость проводимого исследования в области применения информационных технологий в прогнозировании течения гипопитуитаризма у детей и подростков обусловлены:

- необходимостью достижения ускоренных темпов роста в первые годы лечения гипопитуитаризма и их нормализации в последующем [5];
- необходимостью снижения рисков развития осложнений;
- потребностью в применении информационных технологий для автоматизации процессов хранения и обработки медицинских данных и разработки систем принятия решений в сфере здравоохранения на основе технологий искусственного интеллекта [2].

Для формирования базы данных и построения моделей прогнозирования использовались текстовые медицинские выписки, содержащие деперсонализированную информацию о пациентах КГБУЗ «Алтайский краевой клинический центр охраны материнства и детства».

Для достижения поставленной цели необходимо решить следующие задачи:

1) изучение предметной области;
2) извлечение информации из текстовых медицинских выписок детей и подростков Алтайского края, страдающих гипопитуитаризмом;
3) проектирование базы данных;
4) обработка данных и отбор признаков;
5) выбор, построение и обучение моделей прогнозирования роста детей и подростков после курса лечения;
6) сравнение и оценка качества построенных моделей;
7) разработка модульной медицинской информационной системы.

Выпускная работа состоит из введения, двух глав, заключения, библиографического списка и приложений. Во введении сформулированы цель и задачи, определены объект и предмет исследования, раскрыты актуальность и практическая значимость выпускной работы.

В первой главе подробно раскрыты основные понятия, связанные с эндокринным заболеванием гипопитуитаризм. Рассмотрены основные понятия и

задачи интеллектуального анализа данных. Приведен обзор инструментов, использованных в данном исследовании.

Вторая глава посвящена разработке медицинской информационной системы. В ней подробно описан процесс извлечения данных детей, разработки графического интерфейса. Также был описан процесс интеллектуального анализа данных, направленный на достижение поставленной цели: построение моделей прогнозирования роста после прохождения курса лечения. Дана оценка качеству построенных моделей.

В ходе выполнения исследования обработан большой объем текстовой медицинской информации, создана база данных, содержащая обезличенную информацию медицинского обследования детей и подростков Алтайского края, страдающих гипопитуитаризмом, построены и обучены модели прогнозирования роста детей и подростков, создана медицинская информационная система.

1. ТЕОРЕТИЧЕСКИЕ АСПЕКТЫ ИССЛЕДОВАНИЯ

1.1 Гипопитуитаризм

Заболевания эндокринной системы занимают одну из лидирующих позиций среди всего спектра заболеваний. Особое внимание уделяется развитию эндокринных заболеваний у детей и подростков, так как данная патология отличается длительностью течения, развитием осложнений, ухудшением качества жизни и нередко приводит к инвалидности [12,16].

Гипопитуитаризм (ГП) – эндокринное заболевание, которое характеризуется снижением или отсутствием одного, или более гормонов гипофиза. Нарушение секреции, ограниченное гормоном роста, является изолированной формой дефицита гормона роста (ДГР) и встречается наиболее часто. Также ДГР может встречаться с дефицитом других гормонов гипофиза. Кроме того, нередки случаи развития множественного дефицита гормонов гипофиза в течение нескольких лет после манифестации изолированной формы дефицита гормона роста [1].

Различают врождённый гипопитуитаризм и приобретённый гипопитуитаризм, различающиеся по причинам появления. Врождённый ГП может развиваться в результате дородовой и родовой травмы, либо мутаций в генах, которые контролируют продукцию соматотропного гормона (СТГ), стимулирующего рост костей, хрящей и мягких тканей. Причинами приобретённого ГП могут служить опухоли гипоталамуса и гипофиза, опухоли других отделов мозга, травмы (черепно-мозговые, хирургические повреждения) и инфекции (вирусный, бактериальный энцефалит и менингит) [5].

В связи с тем, что проявление гипопитуитаризма связано со сниженной секрецией одного или нескольких гормонов, его осложнения зависят от недостаточности соответственного гормона и могут вызывать низкорослость или карликовость, сердечную недостаточность, ожирение, нарушение половой функции, бесплодие.

1.2 Интеллектуальный анализ данных

Интеллектуальный анализ данных (ИАД) – это совокупность методов нахождения в данных сведений, пригодных для использования в различных сферах

деятельности человека. Для обнаружения закономерностей и тенденций используются алгоритмы, основанные на математическом анализе и статистике. Чаще всего такие зависимости сложно определить при просмотре данных, поскольку связи сложны для восприятия человека, либо информации слишком много.

В ходе интеллектуального анализа данных рассматриваются наблюдения (объекты), которые состоят из совокупности значений признаков (факторов, характеристик). Задача ИАД заключается в нахождении зависимости целевого признака (зависимой переменной) от остальных признаков (независимых переменных).

Алгоритмы интеллектуального анализа данных можно разделить на два типа: обучение с учителем и обучение без учителя [4]. Обучение с учителем предполагает наличие обучающей выборки, состоящей из пар объект/ответ. Алгоритм учит модель соответствовать этим данным как можно сильнее. После обучения модель может использоваться для предсказания с некоторой вероятностью целевой переменной, подавая на вход новые объекты. К задачам обучения с учителем относятся задачи регрессии и классификации.

Обучение без учителя предполагает наличие данных без заранее известных ответов. Целью обучения без учителя является нахождение связей в уже имеющемся наборе данных. Примером такой задачи служит кластеризация объектов, то есть группировка на основе схожести признаков.

Данные могут быть как уже хорошо подготовленные, так и иметь много неточностей, ошибок, пропусков, что в свою очередь может сильно повлиять на результаты. Для эффективного применения алгоритмов данные необходимо обработать.

1.3 Обзор использующихся в исследовании инструментов

Python – интерпретируемый высокоуровневый язык программирования с динамической строгой типизацией, позволяет решать задачу по обработке данных и применению к ним методов машинного обучения. К одной из причин популярности среди it-специалистов можно отнести ориентирование языка

программирования на повышение читаемости кода и обеспечение переносимости программ, написанных на нём. Python имеет множество модулей, библиотек и фреймворков, позволяющих подстроиться под самые разные задачи. В данном исследовании используются следующие библиотеки:

1) NumPy – библиотека, предоставляющая многомерные массивы и набор процедур для операций с ними. Массивы NumPy схожи с встроенным типом данных языка Python list, но позволяет более эффективно хранить и быстрее выполнять операции с массивами данных больших размеров [18].
2) Pandas – библиотека, содержащая высокопроизводительные структуры данных DataFrame и инструменты анализа. DataFrame представляет собой таблицу, схожу с электронной таблицей Microsoft Excel. Библиотека pandas предлагает большой спектр методов по работе с DataFrame, в частности, она позволяет выполнять SQL-подобные запросы, проводить очистку данных, обрабатывать пропущенные значения. Pandas не требует, чтобы все записи в массиве были одного и того типа, каждый столбец может иметь свой собственный тип [20].
3) Scikit-learn – один из широко используемых пакетов, предоставляющий различные алгоритмы машинного обучения и содержащий подробную документацию по каждому из них. С помощью него можно реализовать различные алгоритмы классификации, регрессии и кластеризации. Вместе с тем библиотека Scikit-learn предлагает большое количество функций для предобработки данных, тонкой настройки и оценки моделей машинного обучения [22].

C# — современный объектно-ориентированный и типобезопасный язык программирования. C# позволяет разработчикам создавать множество типов безопасных и надежных приложений, работающих в экосистеме .NET [17]. Инструментарий C# позволяет решать широкий круг задач. На нём разрабатывают:

- Web-приложения;
- различные игровые программы;

- приложения для платформ Android или iOS;
- программы для Windows;

Windows Forms – технология разработки событийно-ориентированных интеллектуальных клиентов – приложений с полнофункциональным графическим интерфейсом, простых в развертывании и обновлении, способных работать при наличии или отсутствии подключения к Интернету и использующих более безопасный доступ к ресурсам на локальном компьютере по сравнению с традиционными приложениями Windows. Взаимодействие с приложением происходит через элементы управления – визуальные классы, которые получают введённые пользователем данные и могут инициировать различные события. Событие — это действие, требующее реагирования или обработки в коде. События могут генерироваться действиями пользователя (например, нажатием кнопки мыши или клавиши на клавиатуре), программным кодом или системой [23].

ONNX – открытая библиотека программного обеспечения, предназначенная для построения моделей искусственного интеллекта. ONNX поддерживает взаимодействие между разными средами, то есть модель можно обучить в одной из многих популярных сред для машинного обучения, например Python, преобразовать в формат ONNX и использовать её в другой [19].

2. РАЗРАБОТКА МЕДИЦИНСКОЙ ИНФОРМАЦИОННОЙ СИСТЕМЫ

Основой разрабатываемой информационной системы является база данных, содержащая информацию о пациентах, результатах медицинского обследования и лечения.

2.1 Обработка текстовых медицинских данных и создание базы данных

Исходные данные для проведения исследования представлены в виде текстовых медицинских выписок, содержащих деперсонализированную информацию о пациентах и их пребывании в стационаре. Часть данных в выписках оформлена в табличной форме, другая часть – в форме записей врача на естественном языке. Каждая медицинская выписка заполняется врачом вручную и, в случае пациентов с гипопитуитаризмом, имеет следующие разделы: сведения о пациенте, сведения о диагнозе и осложнениях, анамнезы заболевания и жизни, результаты обследований, проведенное лечение, рекомендации врача.

Для внесения данных в информационную систему и разработки системы поддержки принятия решений медицинские выписки необходимо обработать, извлечь из них информацию и внести ее в базу данных [14].

Задача извлечения данных из текстовых медицинских выписок сводится к решению следующих задач:

1) извлечение числовых характеристик из таблиц;
2) извлечение числовых характеристик из текста;
3) нахождение в тексте упоминаний медицинских концептов (диагноз, сопутствующие заболевания, жалобы и симптомы, лекарственные препараты, медицинские процедуры), извлечение фрагментов текстовых данных.

Поскольку шаблоны выписок с годами менялись, во избежание потери информации при решении поставленных задач, необходимо учитывать следующие факты:

- выписки могут содержать различные сокращения и аббревиатуры, например, в таблице общий анализ крови показатель «Эритроциты» может иметь следующие наименования: «Эритроциты», «Эр», «Эритр.»;

- могут использоваться слова синонимы, например, находился/прибывал в стационаре и др.;
- таблицы, содержащие одинаковые показатели, могут иметь разную структуру;
- расположение показателей в таблице не имеет определенного порядка.

Эффективно работать с текстовыми данными позволяет библиотека python-docx [10] языка программирования Python, предназначенная для работы с текстовыми файлами формата «.docx». Библиотека предоставляет множество инструментов для работы с текстовыми файлами, в том числе, позволяет извлекать и обрабатывать таблицы.

В таблицах хранится информация о биохимическом анализе крови, общем анализе крови, анализе мочи, гормональных тестах. Класс Document, позволяющий читать выписки, имеет свойство Document.tables, которое предоставляет список таблиц в порядке следования в документе.

Таблицы в выписках могут иметь две структуры:

1) наименования показателей располагаются в первой строке таблицы слева направо, их значения в следующей строке;
2) наименования показателей располагаются в первом столбце таблицы сверху вниз, их значения и дата в следующих столбцах.

Многие характеристики, такие как дата рождения, дата поступления и дата выписки из больницы, основной и сопутствующий диагноз, параметры физического развития ребёнка (рост, вес, SDS, оценка полового развития), результаты обследований, проведённое и рекомендованное лечение, находятся вне таблиц. У некоторых из них есть определённая последовательность, например, сопутствующий диагноз, если он есть, всегда идёт после основного. Также необходимо учитывать, что при описании препаратов, используемых при госпитализации, дозировка может быть неизменной, а может меняться на протяжении лечения пациента. В этом случае либо препарат пишется с новой строки и нужной дозировкой, либо в одну строку и между дозировками ставится

разделитель. Например, «Растан по 1,4мл/сут п/к бедра х 1 р в день в 21.00 с 01.08ю11 по 03.08.11→затем по 1,5мл/сут (2,0мг/сут или 0,038мг/кг/сут) п/к бедра х 1 р в день в 21.00».

Из-за того, что такие данные представлены на естественном языке, их структура может отличаться между выписками. Примером таких характеристик служат проведенное и рекомендованное лечение препаратом «Растан». В одной выписке описание лечения имеет вид «Растан 0,90 мг в п/к бедра в 21ч 17.06-22.06.15», в другой «Растан 0.25мг/сут (0,030 мг/кг/сут) в п/к бедра ежедневно в 21ч (05.10.18-08.10.18)».

Оценка физического развития пациента производится с помощью перцентильных таблиц и кривых, которые зависят от пола. Поэтому для контроля течения гипопитуитаризма необходимо знать – пациент мальчик или девочка. В деперсонализированных выписках не указан пол, поэтому для определения данного параметра используются формулы полового развития: для мальчиков формула имеет формат Ax-степень, Pb-степень, Ma-степень, G-степень, для девочек – Ax-степень, Pb-степень, Ma-степень, Me-степень.

Для извлечения текстовых и числовых данных из текста используются регулярные выражения – формальный язык поиска и осуществления манипуляций с подстроками в тексте, основанный на использовании метасимволов, реализованный в стандартном модуле re языка программирования Python. Методы re.search() и re.findall() позволяют применять регулярные выражения к поиску заданных шаблонов. Метод re.search() возвращает первое совпадение с шаблоном, а re.findall() – список всех найденных совпадений.

База данных – набор данных, которые хранятся в соответствии со схемой данных таким образом, чтобы нужная информация могла быть найдена и обработана с помощью электронного устройства. Схема базы данных есть описание содержания, структуры и ограничений целостности. Создать базу данных и предоставить пользователям возможность взаимодействовать (или отправлять

запросы системе на выполнение операций) с ней позволяет комплекс программ, называемых системой управления базами данных (СУБД).

Извлечённые данные сохраняются в формате .xlsx, после чего могут импортироваться в нужную систему управления базами данных. В качестве СУБД выбрана СУБД Microsoft Office Access.

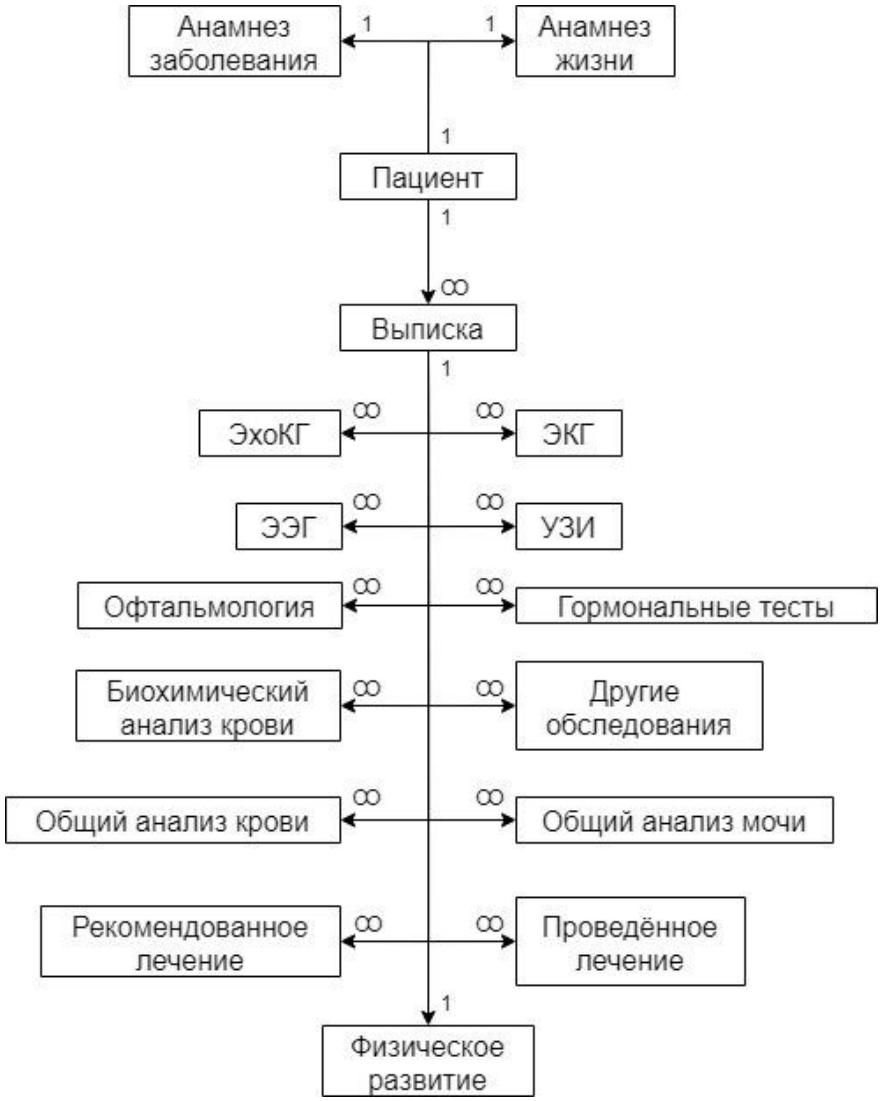

Рисунок 2.1 – ER-диаграмма базы данных

На рисунке 2.1 представлена ER-диаграмма базы данных, отражающая связи между сущностями – таблицами. База данных содержит 17 таблиц. В общей сложности в базе данных представлены более 150 наименований атрибутов. Таблицы, имеющиеся в базе, можно разделить на два типа: таблицы, содержащие данные о пациенте, которые не зависят от периода пребывания в стационаре (например, пол, возраст пациента и др.), и таблицы, включающие данные,

относящиеся к периоду пребывания на стационарном лечении (результаты анализов и обследований, проводимое медикаментозное лечение и т.д.). Для связи таблиц первого типа используется идентификатор пациента, а для таблиц второго типа – идентификатор выписки.

К преимуществам СУБД в сравнении с хранением выписок в виде отдельных документов можно отнести [3]:

1) Быстродействие. Компьютер может находить, выбирать, добавлять, изменять и удалять данные быстрее человека.
2) Компактность. Нет необходимости в создании и ведении большого количества отдельных файлов.
3) Низкие трудозатраты. Механическую и рутинную работу берёт на себя компьютер.
4) Защита. Данные легче защитить от случайных потерь, изменений и несанкционированного доступа.

2.2 Функциональные характеристики информационной системы

Целью информационной системы является обеспечение врачей-эндокринологов необходимым инструментарием для эффективного контроля за течением заболевания:

1) МИС позволит осуществить быстрый поиск пациента и доступ к его медицинским выпискам – формализованной истории болезни, фиксировать показатели физического развития пациента на перцентильных кривых;
2) МИС даст возможность прогнозировать рост пациента с помощью интеллектуальной системы поддержки принятия решений, базирующейся на моделях машинного обучения. Благодаря интеллектуальной системе врач сможет подбирать оптимальную стратегию лечения на основе прогнозируемого роста пациента.

17 июня 2019 года Минюстом России зарегистрирован приказ Минздрава России № 911н «Об утверждении Требований к государственным информационным системам в сфере здравоохранения субъектов Российской

Федерации, медицинским информационным системам медицинских организаций и информационным системам фармацевтических организаций». Приказ определяет обязательный состав функций, которые должны обеспечивать информационные системы в медицинских организациях, в том числе: ведение электронной медицинской карты, централизованных систем (подсистем) хранения и обработки результатов диагностических и лабораторных исследований, реализация возможности телемедицинских консультаций и другое. В соответствии с приказом, разработанная в рамках исследования МИС ориентирована на коллективное использование специалистами-эндокринологами одного медицинского учреждения и может быть интегрирована в информационную систему медицинского учреждения как отдельный модуль, предназначенный для узких специалистов.

Информационная система имеет модульную структуру. Каждый модуль разработан с целью решать определенный круг задач. МИС включает в себя следующие компоненты:

1) Хранилище данных – компонент, предназначенный для хранения структурированных медицинских данных.
2) Модуль «История болезни» предназначен для просмотра, редактирования, добавления и удаления информации о пациенте и его пребывании в стационаре.
3) Модуль «Контроль показателей физического развития пациента» позволяет врачам наблюдать отметки показателей физического развития пациента на перцентильных кривых, отслеживать изменения в динамике.
4) Модуль «Прогнозирование заболевания» – система поддержки принятия решений, предназначенная для прогнозирования роста пациента с помощью моделей машинного обучения, подбора оптимального лечения для пациента.
5) Подсистема информационной безопасности предназначена для обеспечения информационной безопасности через разграничение прав доступа к данным информационной системы.

Хранилище данных представлено созданной базой данных, сформированной из извлечённой и обработанной информации из текстовых медицинских выписок детей и подростков.

Для реализации программного интерфейса выбран объектно-ориентированный язык программирования C# и API Windows Forms.

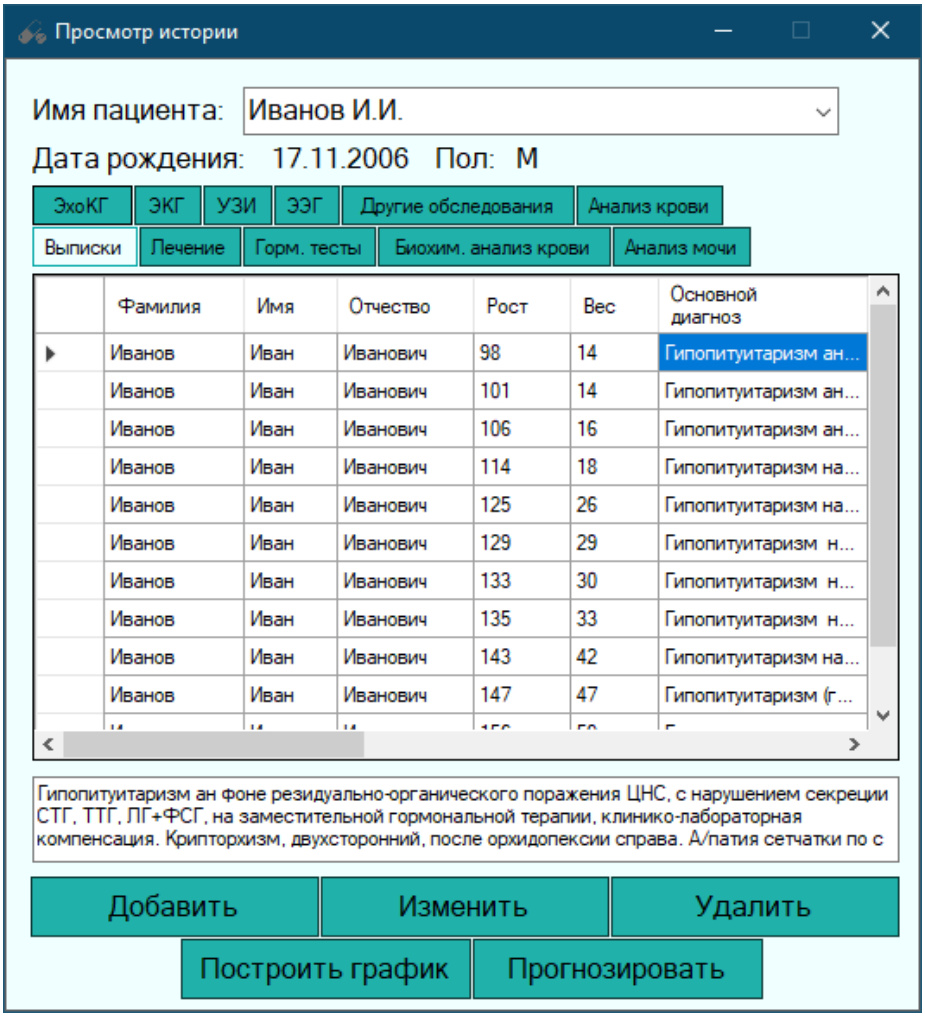

Рисунок 2.2 – Интерфейс модуля «История болезни»

Модуль «История болезни» представлен совокупностью окон: «Просмотр истории», «Изменение записи» и «Добавление записи». На рисунке 2.2 представлен интерфейс окна «Просмотр истории». Просмотр истории болезни конкретного пациента осуществляется через элементы управления класса DataGridView, предоставляющие настраиваемые таблицы для отображения данных с базовым источником или без него. Выбор пациента осуществляется с помощью элемента управления класса ComboBox, образующий выпадающий список элементов, в данном случае пациентов из базы данных.

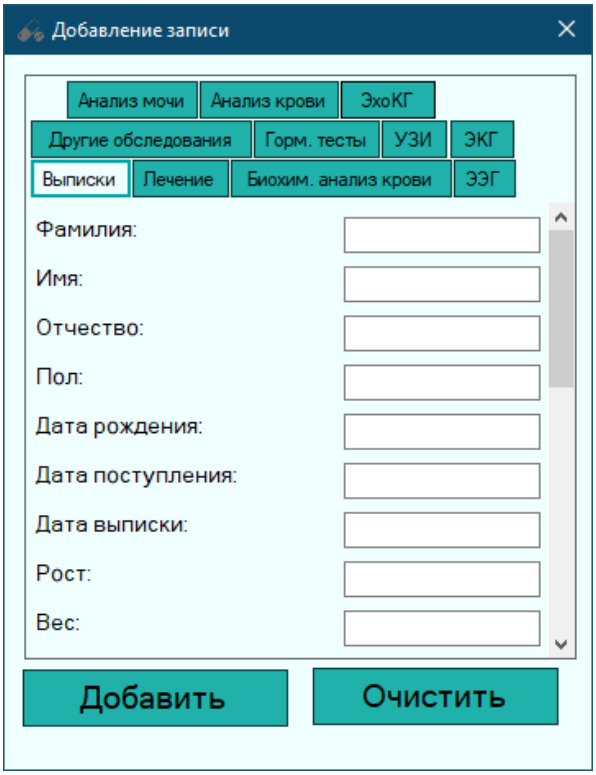

Рисунок 2.3 – Окно добавления записи пациента

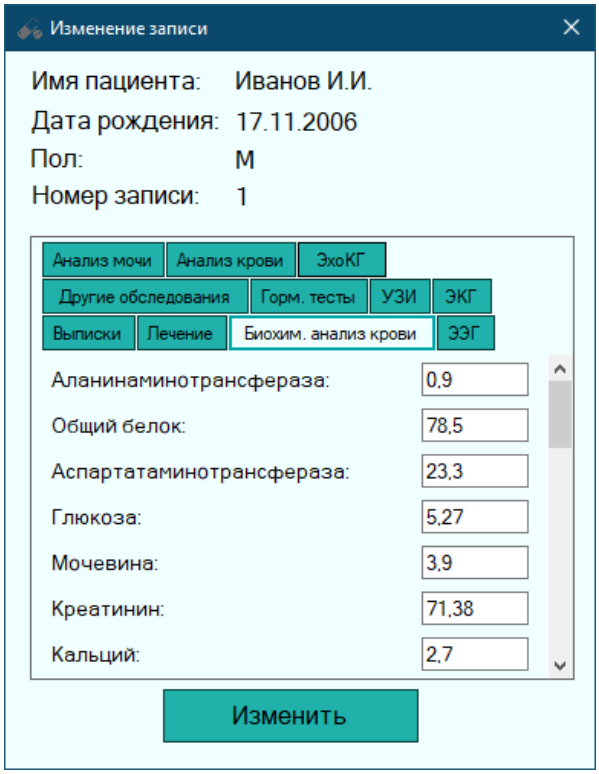

Рисунок 2.4 – Окно изменения записи пациента

На рисунках 2.3 и 2.4 представлены интерфейсы для добавления новой записи и редактирования записи, выделенной в окне просмотра истории пациента в таблице с данными. Текущие значения параметров обследований, тестов, анализов и прочего выставляются при загрузке формы, после чего их возможно удалить или редактировать. Для ввода или вывода текста используются элементы управления TexBox, предназначенные для обмена информацией между пользователем и программой. Информация предоставляется в виде однострочного или многострочного текста, в зависимости от значения свойства Multiline.

Модуль «Контроль показателей физического развития пациента» (рис. 2.5) состоит их графиков и отметок на них. На оси абсцисс находится возраст пациента, на левой оси ординат – рост пациента, на правой оси ординат – вес. Левый график соответствует возрасту ребёнка до 3 лет, возраст измеряется в месяцах, единичный отрезок по оси абсцисс равен 1 месяцу. Правый график соответствует возрасту ребёнка от 2 до 17 лет, возраст измеряется в годах, единичный отрезок по оси абсцисс равен 1 году. На обоих графиках рост и вес измеряются в сантиметрах и

килограммах соответственно. Фиолетовые точки соответствуют значению роста в конкретном возрасте, зелёные – весу.

Для отображения показателей физического развития пациента на перцентильных кривых, представленных в экземпляре класса Bitmap, создаётся элемент управления класса Graphics пространства имён System.Drawing, который затем рисует точки на изображении, соответствующие росту или весу в определенном возрасте.

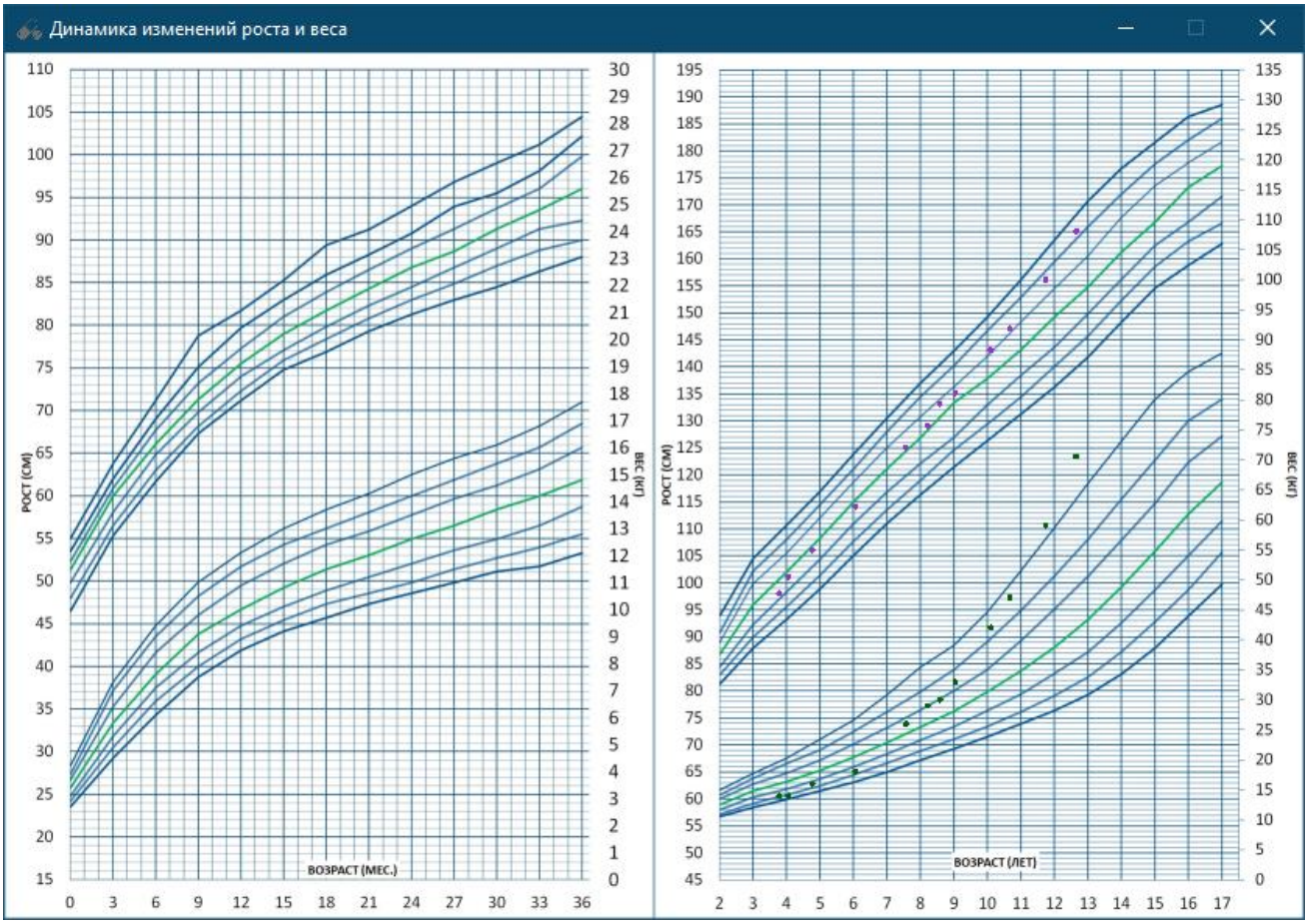

Рисунок 2.5 – Окно для отображения показателей физического развития пациента на перцентильных кривых

На рисунке 2.6 представлен интерфейс модуля «Прогнозирование роста». При загрузке окна выносятся результаты обследований и анализов текущей госпитализации. После чего врач может наблюдать примерное значение роста пациента, подбирая и выбирая оптимальную дозировку препарата при следующей госпитализации после прохождения курса лечения. Модель, с помощью которой

производится прогноз, имеет формат «.onnx». Для взаимодействия с моделью в C# используется пакет Microsoft.ML.OnnxRinTime.

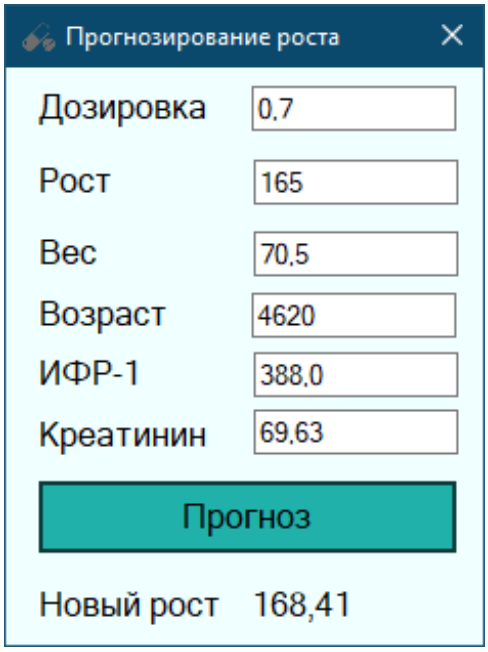

Рисунок 2.6 – Окно для прогнозирования роста при следующей госпитализации пациента

Целостность и конфиденциальность данных о пациентах является важной составляющей МИС. Подсистема информационной безопасности предназначена для обеспечения информационной безопасности через разграничение прав доступа к данным информационной системы. Системный администратор не имеет возможности просматривать, редактировать, удалять или добавлять информацию о пациентах, но может управлять данными медицинских сотрудников, работающих в программе. В Приложении 1 представлены интерфейсы входа в программу и управления мед. персоналом, доступных администратору. В свою очередь, врач имеет право на просмотр выписок и манипулирование ими.

Для защиты пароля используется его хеширование. Хеширование – процесс преобразования массива входных данных произвольной длины в битовую строку фиксированной длины, выполняемый определённым алгоритмом. Простого хеширования пароля недостаточно для обеспечение надёжной защиты данных. Поэтому при регистрации пользователя хеш-функция применяется к комбинации «соль + пароль», где соль генерируется с помощью функции формирования ключа (KDF) и сохраняется вместе с полученной строкой. KDF – функция, формирующая

секретный ключ на основе полученного пароля с помощью псевдослучайной функции. При проверке пароля происходит хеширование «соль + проверяемый пароль» и сравнивается со значением, полученным раннее. Для хеширования пароля в C# используется класс Rfc2898DeriveBytes, реализующий функцию формирования ключа на основе пароля по стандарту PBKDF2 посредством генератора псевдослучайных чисел HMACSHA1.

Программа доступна по ссылке:
https://drive.google.com/file/d/1OOXyPD3fQlUWm_Ur6AK1ygXel0-5ep4_/view?usp=sharing

3.2 Интеллектуальная система поддержки принятия врачебных решений

Интеллектуальная система поддержки принятия решений, интегрируемая в МИС, предназначена для прогнозирования роста пациента с целью выбора оптимальной стратегии лечения. Основой интеллектуальной системы поддержки принятия решений является модель, полученная в результате обучения машинного алгоритма на данных из МИС. Применение методов машинного обучения в медицине – это способ реализации персонализированного подхода к диагностике и прогнозированию заболеваний [13].

Задача прогнозирования роста детей и подростков, страдающих гипопитуитаризмом, является задачей восстановления регрессии. Регрессия – зависимость математического ожидания одной случайной величины от одной или нескольких других случайных величин, то есть $E(y|x) = f(x)$. Регрессия может быть представлена в виде суммы:

$$y = f(x) + \varepsilon,$$

где y – целевая переменная, $x = (x_1, \dots, x_n) \in X \subset R^n$ – признаковое описание объектов, n – количество независимых переменных, $f(x)$ – функция регрессионной зависимости, ε – случайная ошибка. Тогда решением задачи регрессии является нахождение такой функции f, которая наилучшим образом интерполирует элементы обучающей выборки $\tilde{X} = (\tilde{\boldsymbol{x}}_i, \tilde{y}_i)_{i=1}^{m}, \tilde{\boldsymbol{x}}_i \in X, \tilde{y} \in R, m$ – количество наблюдений. Чтобы найти такую функцию, необходимо определить метрику, по которой можно судить о том, насколько близка данная функция к обучающей

выборке. Такая метрика называется функционалом ошибки $Q(y(\boldsymbol{x}), X)$, который достигает минимума при наилучшем решении. В таком случае, задача регрессии эквивалентна задаче минимизации функционала ошибки, а процесс минимизации называется обучением модели:

$$y(\boldsymbol{x}) = \underset{y(\boldsymbol{x}) \in Y}{\operatorname{argmin}} Q(y(\boldsymbol{x}), X).$$

Зависимой переменной в рассматриваемой задаче выступает рост пациента при следующем поступлении в стационар после курса лечения (5-7 месяцев) препаратом «Растан®». Набор независимых (входных) переменных определен в ходе исследования с помощью корреляционного анализа.

Корреляционный анализ тесно связан с регрессионным анализом, с его помощью можно определить необходимость включения тех или иных признаков в регрессионную модель. Для корреляционного анализа используется линейный коэффициент корреляции:

$$r_{XY} = \frac{cov_{XY}}{\sigma_X \sigma_Y},$$

где cov_{XY} – ковариация между случайными переменными X и Y, σ_X и σ_Y – среднеквадратические отклонения X и Y. Чем больше по модулю коэффициент корреляции между признаком и целевой переменной, тем более информативным является данный признак. Независимым величинам соответствует нулевой коэффициент корреляции, значения +1 и -1 коэффициента корреляции соответствуют наличию линейной зависимости, которая является самой сильной из всех возможных форм зависимости между переменными.

Таблица 2.1

Значения коэффициентов корреляции отобранных независимых переменных с целевой переменной.

Признак	**Коэффициент корреляции**
Текущий рост пациента	0.916
Вес пациента	0.829
Возраст пациента	0.677
ИФР-1	0.696

Дозировка препарата «Растан®» (мг)	0.805
Уровень креатинина в крови	0.562

На основе корреляционного анализа отобраны наиболее информативные признаки: текущий рост пациента, вес, возраст, уровень инсулиноподобного фактора роста 1 (ИФР-1), дозировка препарата и уровень креатинина в крови. В таблице 2.1 представлены коэффициенты корреляции между зависимой переменной и отобранными независимыми переменными.

Для разработки интеллектуальной системы поддержки принятия решений, предназначенной для прогнозирования роста пациента, принято решение обучить и оценить качество нескольких моделей: многомерная линейная регрессия, метод опорных векторов, деревья решений. Для построения моделей выбран язык программирования Python и библиотека Scikit-learn.

Многомерная линейная регрессия. Задачей одномерной линейной регрессии является нахождение прямой, характеризующей зависимость целевой переменной от независимой. Модель имеет вид:

$$y = ax + b,$$

где x – фактор, a – параметр, b – свободный член. a и b находятся с помощью метода наименьших квадратов (МНК), задачей которого является минимизация среднеквадратической ошибки межу спрогнозированным значением зависимой переменной и реальным значением:

$$\sum_{i=1}^{n}(y_i - f(x_i))^2 \to min,$$

где n – количество наблюдений, i – номер наблюдения, y_i – реальное значение целевой переменной в i-ом наблюдении, x_i – значение параметра в i-ом наблюдении, $f(x_i)$ – значение, спрогнозированное моделью при x_i.

Многомерной называют линейную регрессию, в модели которой объекты и признаки являются k-мерными векторами, то есть модель имеет вид:

$$y = \sum_{i=1}^{k} a_i x_i + b,$$

где k – количество факторов, x_i – i-й фактор, a_i – i-й параметр, b – свободный член

Отличие многомерной линейной регрессии от одномерной, заключается в том, что вместо линии регрессии в ней используется гиперплоскость. Для оценки

параметров линии регрессии, как и в одномерном случае, применяется метод наименьших квадратов. Построение модели линейной регрессии в Python осуществляется с помощью класса LinearRegression() модуля linear_model библиотеки Scikit-learn.

SVM регрессия. Метод опорных векторов (SVM) – универсальный набор алгоритмов машинного обучения, которые используются для решения линейных и нелинейных задач классификации и регрессии.

Метод опорных векторов для решения задачи линейной классификации заключается в построении границы решения таким образом, чтобы объекты обучающей выборки находились на наибольшем расстоянии от нее [19]. Для этого по обеим сторонам границы решения строятся две параллельных гиперплоскости, которые «опираются» на ближайшие объекты каждого из классов. Алгоритм работает в предположении, что чем больше расстояние между гиперплоскостями, тем меньше будет средняя ошибка классификатора.

В задаче регрессии алгоритм старается поместить как можно больше объектов между гиперплоскостями. Разделяющая гиперплоскость задаётся уравнением $(w, x) + b = 0$, где $x \in X \subset R^n$ – признаковое описание, $w \in R^n$ – весовые коэффициенты, b – свободный член, $(\cdot, \cdot)$ – операция скалярного произведения. Тогда границы принятия решения можно задать уравнениями:

$(w, x) + b = \varepsilon$,

$(w, x) + b = -\varepsilon$,

где ε – допустимое отклонение результата от фактического значения при обучении. Задача построения оптимальной разделяющей гиперплоскости сводится к задаче:

$\frac{1}{2}\|w\|^2 \to min$, при условии $\begin{cases} y_i - (w, x) - b \le \varepsilon \\ (w, x) + b - y_i \le \varepsilon \end{cases}$, которая решается квадратичным программированием.

Также алгоритмы можно кернелизировать, то есть модифицировать с использованием ядра, для нелинейной регрессии. Ключевая идея в основе ядерных методов для решения задач с такими линейно неразделимыми данными состоит в том, чтобы создать нелинейные комбинации исходных признаков и функцией

отображения $\varphi(\cdot)$ спроецировать их на пространство более высокой размерности, где они становятся линейно разделимыми. При этом необходимо выполнить ядерный трюк (kernel trick), то есть заменить скалярное произведение между двумя точками ядерной функцией:

$$k\left(x^{(i)}, x^{(j)}\right) = \varphi(x^{(i)})^T \varphi\left(x^{(j)}\right).$$

В библиотеке Scikit-learn языка Python содержится несколько классов, с помощью которых можно построить регрессию опорных векторов: LinearSVR(), SVR() и NuSVR(). Для решения задачи прогнозирования роста выбрана модель LinearSVR().

Деревья решений. Деревья принятия решений позволяют решать задачи классификации и регрессии с помощью построения логических схем. Основываясь на признаках в обучающем наборе данных, модель дерева решений обучается на иерархически организованной системе вопросов. При этом задаваемый вопрос на каждом последующем иерархическом уровне зависит от ответа, полученного на предыдущем уровне. Деревья решений применяются для решения многих практических задач, так как имеют целый ряд преимуществ: работают с данными любого типа, не требуют предварительной обработки данных, легко интерпретируются и позволяют автоматически отбирать наиболее релевантные признаки для модели.

Задача оптимизации состоит в оптимизации целевой функции, на основе которой вычисляется прирост информации при каждом расщеплении. Функция имеет вид:

$$IG\left(D_p, f\right) = I\left(D_p\right) - \sum_{j=1}^{m} \frac{N_j}{N_p} I\left(D_j\right),$$

где f - это признак, по которому выполняется расщепление, D_p и D_j – набор данных родительского и j-го дочернего узла, I – критерий расщепления, N_p – общее число образцов в родительском узле и N_j – число образцов в j-ом дочернем узле. Как можно убедиться, прирост информации – это просто разница между неоднородностью родительского узла и суммой неоднородностей дочерних узлов: чем ниже неоднородность дочерних узлов, тем больше прирост информации.

В качестве критерия расщепления используется среднеквадратическая ошибка (MSE):

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2,$$

где n – количество наблюдений, y_i – истинное значение в i -ом наблюдении, $\hat{y}_i$ – предсказанное значение в i -ом наблюдении.

Деревья решений для задач классификации и регрессии реализованы в классе DecisionTreeRegressor() модуля tree библиотеки Scikit-learn.

Оптимизация гиперпараметров построенных моделей проводилась с помощью поиска по сетке, реализованного в классе GridSearchCV() модуля sklearn.model_selection.

Для оценки качества обученных моделей выбран коэффициент детерминации (R2). При оценке регрессионных моделей значение коэффициента детерминации интерпретируется как соответствие модели данным. Коэффициент детерминации может принимать значения от 0 до 1. Для приемлемых моделей считается, что коэффициент детерминации должен быть не меньше 0.5 (50%). Модели с коэффициентом детерминации выше 0.8 (80%) можно признать достаточно хорошими. Значения коэффициента детерминации для каждой модели приведены в таблице 2.2.

Таблица 2.2

Значения коэффициента R2 для построенных моделей.

Модель	**R2**
Linear Regression	0.86
Linear regression SVM	0.93
Regression decision trees	0.85

Высокие значения коэффициента детерминации говорят о хорошем соответствии построенных моделей данным. Наибольшее значение коэффициента детерминации имеет модель линейной регрессии SVM (93%). Полученная модель, сохранённая в формате «.onnx», станет основой интеллектуальной системы поддержки принятия решений, которая интегрирована в МИС.

ЗАКЛЮЧЕНИЕ

Информационные системы, внедряемые в медицинские учреждения различного профиля, являются ключевым звеном в процессе информатизации здравоохранения. Рост объема электронной медицинской информации дает возможность анализировать течение различных заболеваний, применять современные методы и подходы анализа данных и искусственного интеллекта для исследования процессов принятия врачебных решений [11]. Применение технологий искусственного интеллекта и внедрение систем поддержки принятия врачебных решений в медицинские учреждения позволит улучшить и ускорить процесс диагностики заболеваний в трудных диагностических случаях, подбирать оптимальные стратегии лечения пациентов и прогнозировать результаты терапии.

В процессе выполнения исследования достигнута цель и решены все поставленные задачи:

1) изучена предметная область;
2) извлечена информация из текстовых медицинских выписок детей и подростков Алтайского края, страдающих гипопитуитаризмом;
3) спроектирована база данных;
4) произведена обработка данных и отбор признаков;
5) выбраны, построены и обучены модели прогнозирования роста детей и подростков после курса лечения;
6) проведено сравнение и дана оценка качества построенных моделей;
7) разработана модульная медицинская система.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1) Воронцова М.В. Гипопитуитаризм у детей и подростков // Медицинский совет. 2019. № 2. С. 250-258.
2) Гусев А.В., Зарубина Т.В. Поддержка принятий врачебных решений в медицинских информационных системах медицинской организации // Врач и информационные технологии. 2017. № 2. С. 60-72.
3) Дейт К. Дж. Введение в системы баз данных, 8-е издание.: Пер. с англ. — М.: Издательский дом "Вильяме", 2005. — 1328 С.
4) Мюллер А., Гвидо С. Введение в машинное обучение с помощью Python. Руководство для специалистов по работе с данными. – М.: «Вильямс», 2017. – 393 С.
5) Нагаева Е.В. Федеральные клинические рекомендации по диагностике и лечению гипопитуитаризма у детей и подростков // Проблемы эндокринологии. 2013. № 59 (6). С. 27-43.
6) Панова Т.В. Информационные технологии в российской медицине: перспективы и возможности // Экономические науки. 2017. № 5 (150). С. 53-56.
7) Рашка С. Python и машинное обучение. – М.: ДМК Пресс, 2017. – 418 с.
8) Скит Д. С42 C# для профессионалов: тонкости программирования, 3-е изд. : Пер. с англ. — М. : ООО «И.Д. Вильямс», 2014. – 608 С.
9) Фролов А.В., Фролов Г.В. Визуальное проектирование приложений C#. – М.: КУДИЦ-ОБРАЗ, 2003. – 512 С.
10) Battineni G., Chintalapudi N., Amenta F. Machine learning in medicine: Performance calculation of dementia prediction by support vector machines (SVM) // Informatics in Medicine Unlocked. 2019. Vol 16. 100200.
11) Hong N., Park H., Rhee Y. Machine Learning Applications in Endocrinology and Metabolism Research: An Overview // Endocrinology and Metabolism. March 2020. Vol. 35 (1). P. 71–84.

12) Ibáñez L., Barouti K., Markantes G.K., Armeni A.K., Georgopoulos N.A. Pediatric endocrinology: an overview of the last decade // Hormones (Athens). December 2018. Vol. 17 (4). P. 439–449.

13) Johnson K.B., Wei W.Q., Weeraratne D., Frisse M.E., Misulis K., Rhee K., Zhao J., Snowdon J. L. Precision Medicine, AI, and the Future of Personalized Health Care //Clinical and Translational Science. 2021. Vol 14 (1). P. 86-93.

14) Moskalev I.V., Krotova O.S., Khvorova L.A., Bobkova D.G. 2020 Journal of Physics: Conference Series 1615 012031

15) Subramanian M., Wojtusciszyn A., Favre L., Boughorbel S., Shan J., Letaief K.B., Pitteloud N., Chouchane L. Precision medicine in the era of artificial intelligence: implications in chronic disease management // Journal of Translational Medicine. December 2020. Vol. 18. Issue 1. DOI: 10.1186/s12967-020-02658-5.

16) Yeliosof O., Gangat M. Diagnosis and management of hypopituitarism // Current Opinion in Pediatrics. August 2019. Vol 31. P. 531-536.

17) C# [Электронный ресурс] – Режим доступа: https://python-docx.readthedocs.io/en/latest/ – Загл. с экрана (23.04.2021).

18) NumPy [Электронный ресурс] – Режим доступа: https://numpy.org/doc/stable/ – Загл. с экрана (15.03.2021).

19) ONNX [Электронный ресурс] – Режим доступа: https://onnx.ai/ – Загл. с экрана ().

20) Pandas [Электронный ресурс] – Режим доступа: https://pandas.pydata.org/docs/ – Загл. с экрана (25.03.2021).

21) Python-docx [Электронный ресурс] – Режим доступа: https://python-docx.readthedocs.io/en/latest/ – Загл. с экрана (10.02.2021).

22) Scikit-learn [Электронный ресурс] – Режим доступа: https://scikit-learn.org/0.21/documentation.html – Загл. с экрана (20.04.2021).

23) Windows Forms [Электронный ресурс] – Режим доступа: https://docs.microsoft.com/ru-

ru/dotnet/desktop/winforms/?view=netframeworkdesktop-4.8 – Загл. с экрана (27.04.2021).

ПРИЛОЖЕНИЕ 1

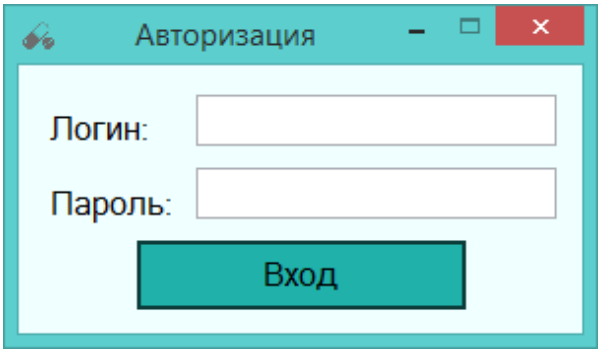

Рисунок 1. Окно входа

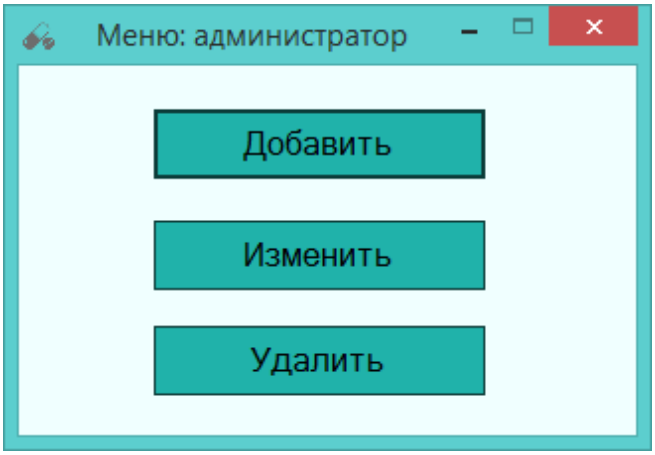

Рисунок 2. Окно меню администратора

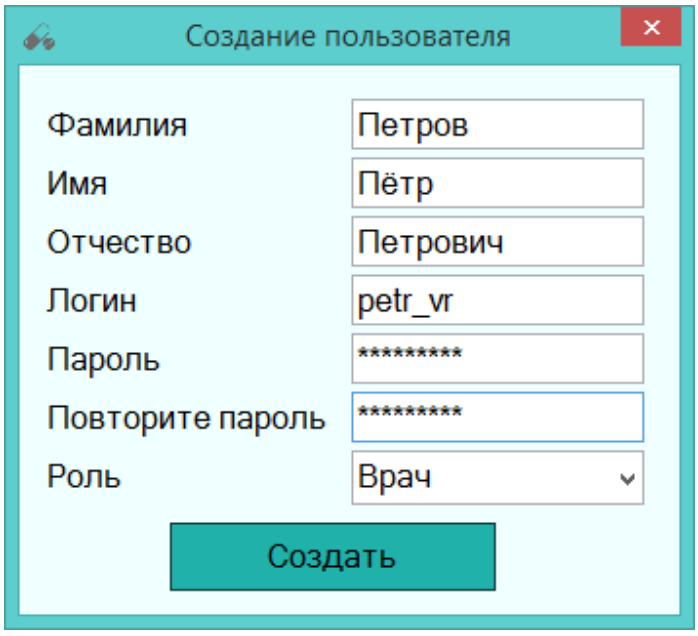

Рисунок 3. Окно создания нового пользователя