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

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

Кафедра «Прикладная математика и информатика»
(наименование кафедры)
01.03.02 Прикладная математика и информатика
(код и наименование направления подготовки, специальности)
Системное программирование и компьютерные технологии
(направленность (профиль)/специализация

БАКАЛАВРСКАЯ РАБОТА

на тему **Применение метода случайного леса для прогнозирования риска сердечно-сосудистых заболеваний**

Студент К.Ю. Калугин
(И.О. Фамилия) (личная подпись)
Руководитель Э.В. Егорова
(И.О. Фамилия) (личная подпись)
Консультанты Н.В. Андрюхина
(И.О. Фамилия) (личная подпись)

Допустить к защите

Заведующий кафедрой к.т.н., доцент, А.В. Очеповский _____
(ученая степень, звание, И.О. Фамилия) (личная подпись)
«_____»_____20_____г.

Тольятти 2019

АННОТАЦИЯ

51 с., 17 рисунков, 6 таблиц, 30 библиографических источников, 7 приложений.

МАШИННОЕ ОБУЧЕНИЕ, ЗАДАЧА ПРОГНОЗИРОВАНИЯ, КЛАССИФИКАЦИЯ, ИНТЕЛЛЕКТУАЛЬНЫЙ АНАЛИЗ ДАННЫХ, БАЗА ДАННЫХ

Сердечно-сосудистые заболевания являются одной из основных причин смертности во всем мире. Ранняя профилактика имеет большое значение для снижения риска их возникновения. Для повышения эффективности ранней диагностики данных заболеваний было решено разработать интеллектуальную систему, использующую метод случайного леса для прогнозирования риска сердечно-сосудистых заболеваний.

Объектом исследования работы стал процесс прогнозирования риска сердечно-сосудистых заболеваний. Предметом исследования является интеллектуальная система, использующая метод случайного лес для прогнозирования риска сердечно-сосудистых заболеваний.

В данной бакалаврской работе предложен метод решения поставленной задачи посредством классификационной модели, построенной с помощью уникальной реализации алгоритма случайного леса на объектно-ориентрованном языке программирования Java.

Классификационная модель обучалась и тестировалась на общедоступной базе данных сердечных заболеваниях Кливленда, распространяемую через репозиторий UCI Machine Learning.

Внедрение разработанной системы позволит медицинским учреждениям повысить оперативность и точность предварительной диагностики пациентов.

ABSTRACT

51 p., 17 figures, 6 tables, 30 bibliographic sources, 7 appendices.

MACHINE LEARNING, PREDICTION, CLASSIFICATION, DATA MINING, DATABASE

Cardiovascular diseases are one of the leading causes of death worldwide. Early prevention is important to reduce the risk of its occurrence. Early prevention is important to reduce the risk of their occurrence. To improve the efficiency of early diagnosis of these diseases, it was decided to develop an intelligent system that uses the random forest method to predict the risk of cardiovascular diseases.

The object of the research was the process of predicting the risk of cardiovascular diseases. The subject of the study is an intelligent system that uses the random forest method to predict the risk of cardiovascular diseases.

In this graduation work, a method is proposed for solving the problem posed by means of a classification model built using the unique implementation of the Random Forest algorithm in the Java object-oriented programming language.

The classification model was taught and tested on the Cleveland heart disease database distributed through the UCI Machine Learning repository.

The introduction of the developed system will allow medical centers to increase the efficiency and accuracy of the preliminary diagnostics of patients.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

Одной из главных причин госпитализаций и смертности в мире являются сердечно-сосудистые заболевания. В России сердечно-сосудистые заболевания являются национальной проблемой, ведь по статистике каждый 13-й гражданин Российской Федерации страдает сердечно-сосудистой патологией. В общую структуру смертности сердечно-сосудистые заболевания вносят весомый вклад – на них приходится 49,6% всех смертей, вызывая наибольшее количество социально-экономических потерь [1].

Таким образом, становится актуальным вопрос разработки интеллектуальной информационной системы для успешного прогнозирования риска сердечно-сосудистых заболеваний. Система, способная совершать подобный прогноз, смогла бы существенно повысить шанс предотвращение заболеваний данного вида у граждан Российской Федерации, а также увеличить эффективность работы отечественных поликлиник.

Целью выпускной квалификационной работы является создание интеллектуальной информационной системы, предназначенной для прогнозирования риска сердечно-сосудистых заболеваний.

Для достижения указанной цели были поставлены следующие исследовательские задачи:

- выявить актуальность разработки;
- предложить метод классификации для поставленной задачи;
- спроектировать интеллектуальную информационную систему и её практически реализовать;
- провести анализ достигнутых показателей выбранного классификатора с помощью различных метрик.

Объектом исследования стал процесс прогнозирования риска сердечно-сосудистых заболеваний. **Предметом исследования работы** является интеллектуальная система, использующая метод случайного леса для прогнозирования риска сердечно-сосудистых заболеваний.

Новизна заключается в проведении исследования решении задачи прогнозирования риска сердечно-сосудистых заболеваний методом случайного леса.

Работа состоит из введения, трех разделов, заключения, списка используемой литературы и приложений. Объем работы составляет 51 страниц, объем библиографии – 30 источников, количество рисунков - 17, количество таблиц – 6, количество приложений – 7.

Первая глава бакалаврской работы посвящена теоретической части, содержащей описание исследуемой задачи, обзор подходов для создания интеллектуальных системы, выбран оптимальный подход для решения поставленной задачи и составлен перечень требований к программному продукту.

Вторая глава сконцентрирована на проектировке и разработке информационной технологии для решения исследуемой задачи.

Заключительная третья глава посвящена тестированию разработанного программного решения.

ГЛАВА 1 ИССЛЕДОВАНИЕ ПРОГНОЗИРОВАНИЯ РИСКА СЕРДЕЧНО-СОСУДИСТЫХ ЗАБОЛЕВАНИЙ

1.1 Актуальность предметной области

Сердце является самой трудолюбивой мышцей в теле человека, выполняющей, по существу, очень простую функцию, сравнимую с насосом, продвигающим кровь по организму. Здоровое сердце просто необходимо для поддержания жизни человека [2]. Однако, существует ряд болезней, именуемые сердечно-сосудистыми заболеваниями, которые могут, непосредственно, повредить его работе.

Сердечно-сосудистые заболевания представляют собой группу болезней сердца и системы кровообращения, в которую входят [3]:

- нарушения ритма и проводимости, к которым относятся аритмия сердца, блокада ножек пучка Гиса, фибрилляция сердца и другие;
- заболевания сердца, связанные с воспалением разных частей сердца: внутренней оболочки – эндокарда, сердечной мышцы – миокарда и соединительной оболочки сердца – перикарда;
- клапанные пороки, делящиеся на врожденные, возникающие из-за генетических нарушений, и приобретенные, чаще всего связанные с инфекционными поражениями организма или аутоиммунными реакциями;
- артериальная гипертензия, являющиеся подгруппой заболеваний связана со стойким повышением артериального давления;
- ишемические поражения связаны с полным или частичным уменьшением притока крови к сердечной мышце;
- поражение сосудов сердца: кардиосклероз, коронарные заболевания сердца, атеросклероз;
- патологические изменения – заболевания, относящиеся к необратимым изменениями в работе сердца, например, сердечная астма и недостаточность, гипертрофия разных частей сердца

Ежегодно от проблем с сердцем в мире умирают 17,5 млн человек [20]. Среди развитых стран Российская Федерация лидирует по этому печальному показателю. По данным Федеральной службы государственной статистики только в 2016 году по причине сердечно-сосудистых заболеваний скончалось около 904,1 тысяч граждан России [4].

Такие пугающие цифры остро ставят необходимость в разработке интеллектуальной информационной системы, основанной на знаниях, полученных в процессе анализа работы специалистов-кардиологов. Благодаря такой системе будет возможно уменьшить количество жертв, способных пострадать от сердечно-сосудистых заболеваний, своевременно спрогнозировав риск их развития.

1.2 Постановка задачи прогнозирования риска сердечно-сосудистых заболеваний

Проблему прогнозирования риска сердечно-сосудистых заболеваний сформулируем следующим образом. Пусть имеется некое множество пациентов $P = p_1, \dots, p_P$, каждый из которых характеризуется вектором признаков $X = x_1, \dots, x_X$, а также множество групп риска возникновения сердечно-сосудистых заболеваний $Y = 0{,}1$. Существует неизвестная целевая функция $\Phi: X \to Y$, значения которой известны только на объектах конечной обучающей выборки $D = (X_i, Y_i)_{i=1}^{D}$.

Целью является построение классификатора $\Phi': X \to Y$ максимально близкого к неизвестной целевой функции Ф [5]. Значения функции в данном случае интерпретируются, как риск развития у пациентов сердечно-сосудистых заболеваний: 0 – низкий, 1 - высокий.

Для того, чтобы классификация приносила удовлетворительные результаты, необходимо в первую очередь добиться высокой точности, что непосредственно зависит от выбора алгоритма классификации, его параметров и количества обучающих примеров.

1.3 Обзор существующих решений поставленной задачи

Рассмотрим и проанализируем схожие работы других авторов для формирования набора методов построения классификационных моделей с целью их исследования в решении прогнозирования риска сердечно-сосудистых заболеваний.

В своей исследовательской работе «Application of Data Mining Techniques in Healthcare and Prediction of Heart Attacks» K. Srinivas и соавторы провели ряд исследований в области здравоохранения и прогнозировании сердечных приступов с применением технологий интеллектуального анализа данных. Рассматривались такие методы, как деревья решений, наивный байесовский классификатор и метод k-ближайших соседей. Набор обучающих данных состоял из 3000 экземпляров с 14 различными атрибутами. По атрибутам набор данных делился на две части: 70% для обучающей выборке, 30% для тестовой. В работе проводилось сравнение выше озвученных алгоритмов классификации. Лучшие результаты в диагностики сердечно-сосудистых заболеваний показал наивный байесовский классификатор [21].

В статье «A Review on Heart Disease Prediction using Machine Learning and Data Analytics Approach» M. Marimuthu и S.Deivarani рассматриваются методы машинного обучения, такие как опорные вектора, наивный байесовский алгоритм, дерево решений и k-ближайших соседей. Используя атрибуты, как кровяное давление, холестерин, диабет, предсказывалась вероятность ишемического заболевания сердца у пациента в ближайшие 10 лет. Также учитывалась семейная история сердечных заболеваний. Лучшую точность у авторов по сравнению с другими рассматриваемыми алгоритмами продемонстрировал метод k-ближайшего соседа [22].

Доктор D. Raghu и другие разработали систему поддержки принятия решений для прогнозирования сердечных заболеваний с использованием наивного байесовского алгоритма в своём труде «Probability: based Heart Disease Prediction using Data Mining Techniques». Для прогноза вероятности ишемической болезни использовались такие атрибуты, как пол, возраст,

уровень кровообращения и глюкозы. Медицинский набор данных был получен из базы данных о сердечных заболеваний Кливленда. В ходе рассмотрения ряда методов наилучшие результаты оказались у наивного байесовского классификатора [23].

В статье «A Comparative Study for Predicting Heart Diseases Using Data Mining Classification Methods. International Journal of Computer Science and Information Security (IJCSIS)» от Isra'a Ahmed Zriqat и его соавторов исследовались пять классификаторов: линейный дискриминантный анализ, деревья решений, случайный лес, наивный байесовский классификатор, метод опорных векторов. Реализация классификаторов происходила с использованием инструментов MATLAB для имитации принятия решений в области здравоохранения с повышенной точностью. Результаты показали, что все алгоритмы классификации могут давать относительно достоверный ответ. По всем показателям оценки качества лучшим оказался метод доревев принятия решений. Второе место занял метод случайного лес [24].

На основании результатов рассмотренных выше работ было решено исследовать такие методы машинного обучения, как деревья решений, случайный лес, метод k-ближайшего соседа и наивный байесовский классификатор для выявления среди них наиболее эффективного в решении поставленной задачи.

1.4 Интеллектуальный анализ данных как способ решения задачи прогнозирования риска сердечно-сосудистых заболеваний

Зачастую, когда возникает необходимость в прогнозировании, имеется некий объём накопленной исторической информации. Медицинская среда обычно содержит приличное количество данных о пациентах, хранящихся в системах управления здравоохранением. Методы интеллектуального анализа данных могут помочь в извлечении ценных знаний и выявлении скрытых зависимостей из данных прошлого для выработки прогноза на будущее. О

Предположение состоит в том, что рассматриваемые предыдущие взаимосвязи и характеристики данных не потеряют свою актуальность и в будущем [6].

Методы классификации являются наиболее широко используемыми алгоритмами в секторе здравоохранения, поскольку они помогают прогнозировать состояние пациента путем классификации, отнеся его на основании значений атрибутов к какому-либо классу. Классификация, как правило, принадлежит к контролируемым методам машинного обучения, требующая от исходных данных, чтобы они были изначально классифицированы к неким классам. При работе с этими данными алгоритм обнаруживает скрытые связи между атрибутами, влияющие на результат прогнозирования. Результатом работы алгоритма является классификатор, построенный на основании обучающего набора, состоящего из кортежей наборов данных и связанных с ними меток классов (рисунок 1.1) [17].

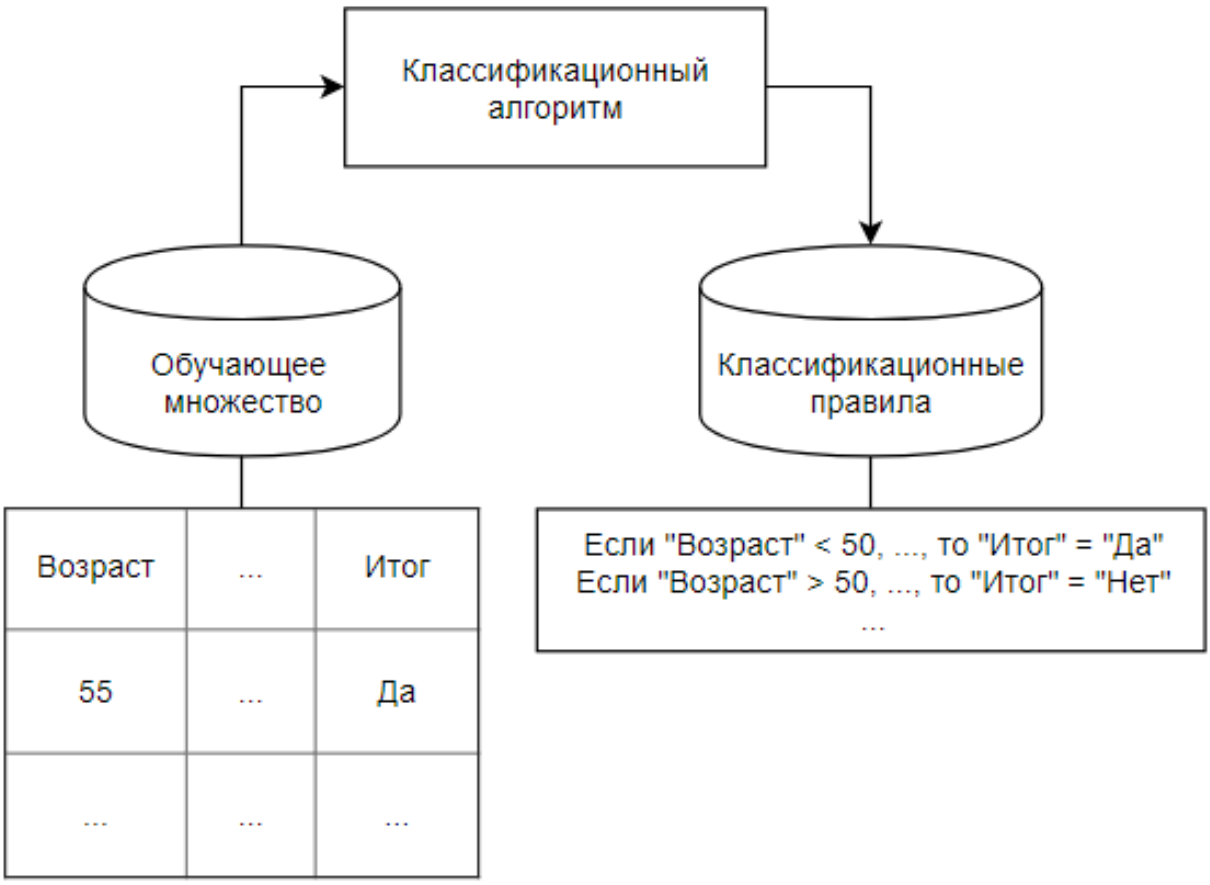


Рисунок 1.1 – Построение классификационной модели

Когда на вход поступает новый случай, построенная классифицирующая модель классифицирует его в один из предопределенных классов, как показано на рисунке 1.2.

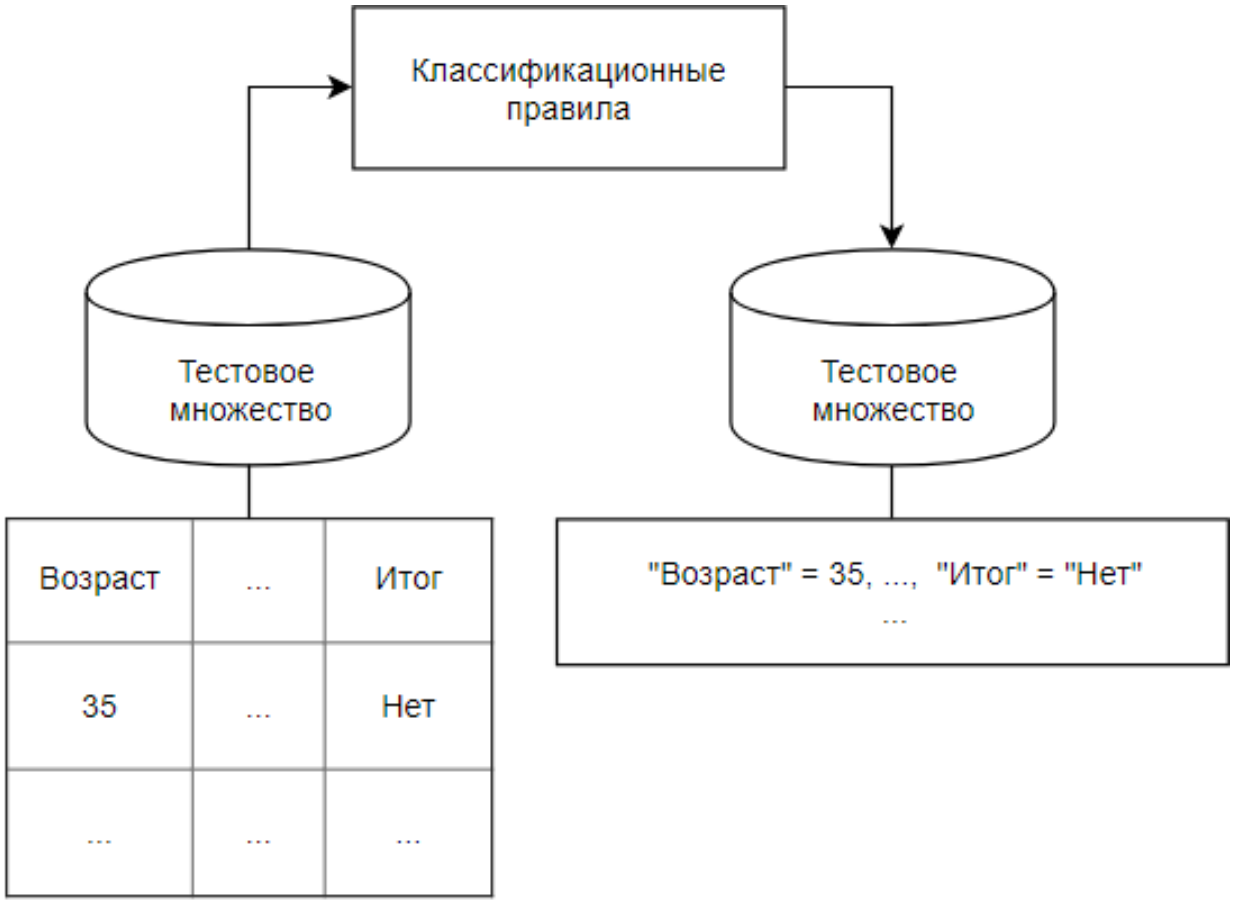


Рисунок 1.2 – Использование классификационной модели

В литературе существует множество алгоритмов классификации, которые можно использовать для решения поставленной задачи. Как было сказано ранее в подразделе «обзора существующих решений» в данной работе было решено рассмотреть такие методы машинного обучения, как деревья решений, случайный лес, метод опорных векторов, метод k-ближайшего соседа и наивный байесовский классификатор.

1.4.1 Деревья принятия решений

Один из самых известных методов классификации, заключающийся в построении деревьев принятия решений.

Дерево решений (decision trees) представляет собой структуру данных, состоящую из узлов и листьев, где узлом называется структурный элемент дерева, в котором происходит оценка значения одного из атрибутов, влияющей на решение переходить к какой из выходящих из этого узла ветвей переходить, листом же именуется конечная вершина одной из ветвей с возможным значением целевого атрибута. Данные, подлежащие классификации, находятся в так называемом «корневом узле» дерева [7]. Схематичное представление дерева решений можно наблюдать на рисунке 1.3.

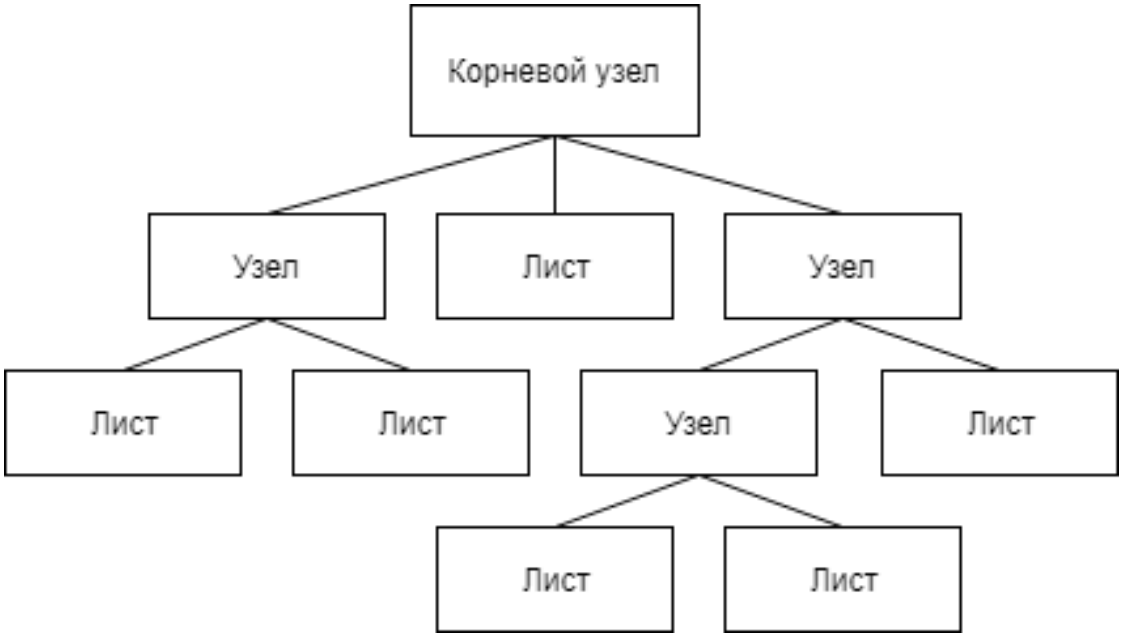


Рисунок 1.3 - Схематичное представление деревьев решений

Если целевой атрибут принимает только категориальные значения, то такое дерево называется деревом классификации, в случае если только количественные, то деревом регрессии.

Существует большое количество алгоритмов, реализующих деревья принятия решений. Все они отличаются способами построения, критериями выбора разбивающих признаков, правилами остановки и т.д.

Самыми широко используемыми алгоритмами на сегодняшний день можно назвать: ID3, CART (Classification and Regression trees) и C4.5.

Рассмотрим подробнее алгоритм CART. Целью данного алгоритма является построение бинарного дерева решений, у которого из каждого узла

дерева выходят только две ветки. На каждом шаге построения дерева правило, формируемое в узле, делит обучающую выборку на часть, в которой выполняется правило (правый потомок) и часть, в которой правило не выполняется (левый потомок). Для выбора оптимального правила применяется функция оценки качества разбиения [25].

Обучение дерева решений относится к классу обучения с учителем, то есть обучающая и тестовая выборки содержат классифицированный набор примеров.

Поскольку на вход дереву может передаваться значение атрибута любого типа, то существует несколько правил разбиения: в случае переменных числового типа проверяется больше ли текущее значение порогового, а для атрибутов категориального типа строится правило равенства входного значения и одного из категориальных. Для выбора атрибута, на основе которого строится правило, используется функция оценки качества разбиения, выражающаяся с помощью индекса $Gini$, которая базируется на идеи о уменьшении нечистоты в узле. Индекс $Gini$, представлен формулой (1.1).

$$Gini\ T\ = 1 - \ \sum_{i=1}^{n} p_i^2, \tag{1.1}$$

где T – множество объектов обучающей выборки;

n – количество классов;

p_i – вероятность встречаемости класса i в множестве T.

В случае, когда множество T делится на две части T_1 и T_2 с числом объектов в каждой N_1 и N_2, соответственно, тогда индекс $Gini$ может быть представлен формулой 1.2.

$$Gini_{split}(T) = \frac{N_1}{N} \cdot Gini(T_1) + \frac{N_2}{N} \cdot Gini(T_2) \tag{1.2}$$

Наилучшим разбиением считается то, при котором значение $Gini_{split}(T)$ минимально.

После построения дерева решений выполняется этап отсечения ветвей, который является компромиссом между получением дерева оптимального размера и получением точной оценки ошибочной классификации. Кроме того,

данный этап помогает решить проблему переобучения. Механизм отсечения ветвей дерева называется «minimal cost-complexity tree pruning» и заключается в отсечении тех поддеревьев, стоимость которых наименьшая, где стоимость вычисляется по формуле (1.3).

$$C_\alpha\ T\ = R(T) + \alpha \cdot\ T\ , \tag{1.3}$$

где T – количество листов дерева T, $R(T)$ – ошибка классификации дерева, то есть доля неправильных ответов.

Таким образом, получается последовательность деревьев с разными стоимостями. В основном выбирают то дерево, у которого минимальная ошибка классификации.

1.4.2 Метод случайного леса

Метод случайного леса (random forest) основан на построении большого числа (ансамбля) деревьев решений, каждое из которых строится по выборке, получаемой из исходной обучающей. Предназначением метода является решение задач классификации и регрессии. Как инструмент анализа данных случайный лес был впервые предложен Лео Брейманом и Адель Катлер в 2001 году [26].

В методе случайного леса в отличие от алгоритма построения классических деревьев решений при построении каждого дерева на стадии построения узлов используется строго фиксированное число случайно отбираемых признаков, помимо этого дерево является полным, в результате чего каждый лист дерева содержит наблюдения только одного класса.

Следует также отметить, что в основании формирования обучающих выборок для деревьев ансамбля используется «bootstrap» подход, в соответствии с которым каждая подвыборка строится из обучающей выборки с возвращением таким образом, что некоторые объекты попадают в подвыборку несколько раз, а некоторые не попадают ни в одну подвыборку.

Алгоритм построения случайного леса можно описать следующим образом. Пусть обучающая выборка состоит из N объектов, размерность

пространства признаков равна M, и задан параметр m – количество признаков, из которых происходит выбор признаков для разбиения в узлах деревьев (в задачах классификации обычно $m \cong \overline{M}$). Все деревья строятся независимо друг от друга по следующей процедуре:

1) генерируется случайная подвыборка с повторяющимися записями того же размера, что и обучающая выборка (размерностью N), называемая также бутстреп выборка;

2) строится решающее дерево, классифицирующее объекты данной подвыборки, при этом в ходе создания очередного узла дерева признак, на основе которого производится разбиение, выбирается не из всех M признаков, а лишь из m случайно выбранных. Выбор наилучшего из этих m признаков может осуществляться с помощью индекса $Gini$, как в алгоритме построения решающих деревьев CART;

3) дерево строится до полного исчерпания подвыборки и не подвергается процедуре отсечения.

Классификация объектов проводится путём голосования: каждое дерево относит классифицируемый объект к одному из классов, и побеждает класс, за который проголосовало наибольшее число деревьев. В задачах регрессии оценка регрессии производится усреднением оценок регрессии всех деревьев.

1.4.3 Наивный байесовский классификатор

Наивный байесовский классификатор (naive bayes classifier) представляет собой простой вероятностный классификатор, основанный на применении Теоремы Байеса со строгими предположениями о независимости [27].

В основе байесовской классификации лежит гипотеза максимальной вероятности, то есть объект d считается принадлежащим классу c_j $(c_j \in C)$, если достигается наибольшая апостериорная вероятность: $\min_c P(c_j|d)$. Формула Байеса выглядит следующим образом 1.4.

$$P(c_j|f) = \frac{P(c_j)P(d|C_j)}{P(d)} \approx P(c_j)P(d|c_j), \quad (1.4)$$

где $P(d|c_j)$ – вероятность встретить объект d среди объектов класса c_j;

$P(c_j)$ – априорные вероятности класса c_j;

$P(d)$ – априорные вероятности объекта d (не влияет на выбор класса и может быть опущена).

Если сделать «наивное» предположение, что все признаки, описывающие классифицируемые объекты, абсолютно равноправны и не имеют связи друг с другом, то $P(d|c_j)$ можно вычислить по формуле 1.5 как произведение вероятностей встретить признак x_i $(x_i \in X)$ среди объектов класса c_j.

$$P(d|c_j) = \prod_{i=1}^{X} P(x_i|c_j), \quad (1.5)$$

где $P(x_i|c_j)$ – вероятностная оценка вклада признака x_i в то, что $d \in c_j$.

На практике при умножении очень малых условных вероятностей может происходить потеря значащих разрядов, ввиду этого вместо самих оценок вероятностей $P(x_i|c_j)$ применяют логарифмы этих вероятностей. Поскольку логарифм – монотонно возрастающая функция, то класс c_j с наибольшим значением логарифма вероятности останется наиболее вероятным. В этом случае решающее правило наивного байесовского классификатора можно представить в виде формулы 1.6.

$$c^* = arg\max_{c_j \in C} \log P(c_j) + \sum_{i=1}^{X} P(x_i|c_j) \quad (1.6)$$

Несмотря на простоту, наивный байесовский алгоритм может быть достаточно точным.

1.4.5 Метод k-ближайших соседей

Метод k-ближайших соседей (k-nearest neighbors algorithm) основан на принципе классификации объекта к классу, которому принадлежит большинство из k его ближайших соседей в многомерном пространстве признаков. Собственно, для вычисления ближайших соседей объекта используются метрики схожести, например, Евклидово расстояние, рассчитывающееся по формуле 1.7.

$$d(x, y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}, \quad (1.7)$$

где x – точка объекта;

y – сосед из списка.

Число k – это количество соседних объектов в пространстве признаков, с которыми сравнивается классифицируемый объект. Иначе говоря, если $k = 10$, то каждый объект сравнивается с 10-ю соседями [28].

Выбор параметра k достаточно противоречив, ведь с одной стороны увеличение его значения повышает достоверность классификации, с другой же границы между классами становятся более размытыми. На практике хорошие результаты дают эвристические методы выбора параметра k, например, перекрестная проверка.

Несмотря на свою относительную алгоритмическую простоту метод показывает хорошие результаты. Главным его недостатком является высокая вычислительная трудоемкость, которая увеличивается квадратично с ростом числа записей в наборе данных.

1.5 Выбор метода классификации для решения поставленной задачи

1.5.1 Метрики для определения качества классификационных моделей

Для определения качества построенных классификационных моделей применялись следующие метрики:

- доля правильно классифицированных записей (accuracy);
- точность (precision);
- полнота (recall);
- F-мера (F-measure).

Для вычисления метрик качества введём такое понятие, как матрица ошибок, представляющую собой способ разбиения объектов на четыре категории в зависимости от комбинации истинного ответа и ответа алгоритма (таблица 1.1).

Таблица 1.1 – Матрица ошибок

Категория	Фактическая оценка	
	Положительная	Отрицательная

Оценка системы	Положительная	Истинно положительный (TP)	Ложно положительный (FP)
	Отрицательная	Ложно отрицательный (FN)	Истинно отрицательный (TN)

Доля правильно классифицированных записей является самой простой оценкой классификации, высчитывающей вероятность того, что класс будет предсказан правильно (1.7).

$$accuracy = \frac{P}{N}, \tag{1.7}$$

где P – количество правильно классифицированных записей;

N – размер обучающей выборки.

Также данную метрику можно выразить через элементы матрицы ошибок, получив формулу 1.8.

$$accuracy = \frac{TP+TN}{TP+TN+FP+FN} \tag{1.8}$$

Точность системы в пределах класса обозначает долю записей, действительно принадлежащих данному классу относительно всех объектов, которые система отнесла к этому классу (1.9).

$$precision = \frac{TP}{TP+FP} \tag{1.9}$$

Полнота системы отображает долю найденных классификатором записей, принадлежащих классу относительно всех записей этого класса в тестовой выборке (1.10).

$$recall = \frac{TP}{TP+FN} \tag{1.10}$$

Очевидно, что чем выше точность и полнота, тем лучше. Однако в реальной жизни максимальная точность и полнота не достижимы одновременно, что вынуждает искать некий баланс. Именно для это используется F-мера, которая объединяет в себе информацию о точности и полноте рассматриваемого алгоритма, что облегчает принятие решения о том какую реализацию использовать. F-мера вычисляется по формуле 1.11.

$$\text{Fmeasure} = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{1.11}$$

Данная формула придает одинаковый вес точности и полноте, поэтому F-мера будет падать одинаково при уменьшении и точности и полноты [8].

1.5.2 Данные для прогнозирования риска сердечно-сосудистых заболеваний

В качестве входных данных для обучения и тестирования построенных классификационных моделей, а также дальнейшего использования с целью прогнозирования риска сердечно-сосудистых заболеваний, было решено воспользоваться общедоступным набором из базы данных сердечных заболеваний Кливленда, распространяемого через репозиторий UCI Machine Learning [29]. Данный набор содержит информацию об исследованиях триста трёх пациентов, подозреваемых на наличие сердечно-сосудистых заболеваний.

В таблице 1.2 можно наблюдать атрибуты, используемые в наборе, с их типом и описанием.

Таблица 1.2 – Атрибуты базы данных сердечных заболеваний Кливленда

№	Название	Описание	Значения
1	Age	Возраст в годах	Непрерывные
2	Sex	Пол пациента	0 = женщина 1 = мужчина
3	Cp	Тип боли в груди	1 = типичная стенокардия 2 = атипичная стенокардия 3 = неангинозные боли 4 = бессимптомный
4	Trestbps	Артериальное давление в покое (в мм рт. ст.)	Непрерывные значение в мм рт.
5	Chol	Сывороточный холестерин в мг/дл	Непрерывные значение в мм / дл
6	Fbs	Уровень сахара в крови натощак	0 = <120 мг/дл 1 = >120 мг/дл

№	Название	Описание	Значения
7	Restecg	Результат электрокардиографии	0 = нормальный 1 = наличие аномалии ST-T 2 = гипертрофия левого желудочка
8	Thalach	Максимальная частота сердечных сокращений	Непрерывные значение
9	Exang	Наличие стенокардии	0 = нет 1 = да
10	Oldpeak	Депрессия сегмента ST на электрокардиограмме	Непрерывные значение
11	Slope	Наклон сегмента ST на электрокардиограмме	1 = восходящий 2 = ровный 3 = спуск
12	Ca	Количество крупных сосудов, окрашенных с помощью флюороскопии	Значения от 0 до 3
13	Thal	Результаты стресс-теста таллия, измеряющего приток крови к сердцу	1 = нормальный 2 = фиксированный дефект 3 = обратимый дефект
14	Target	Результат диагностики	0 = низкий риск 1 = высокий риск

Проверим равномерность целевого атрибута в данных, так как это является достаточно важным моментом в обучении классификаторов. Чрезвычайно несбалансированный набор может быть неэффективным в обучении классификационных моделей. На рисунке 1.4 продемонстрировано количество записей, которые относятся к тому или иному целевому атрибуту.

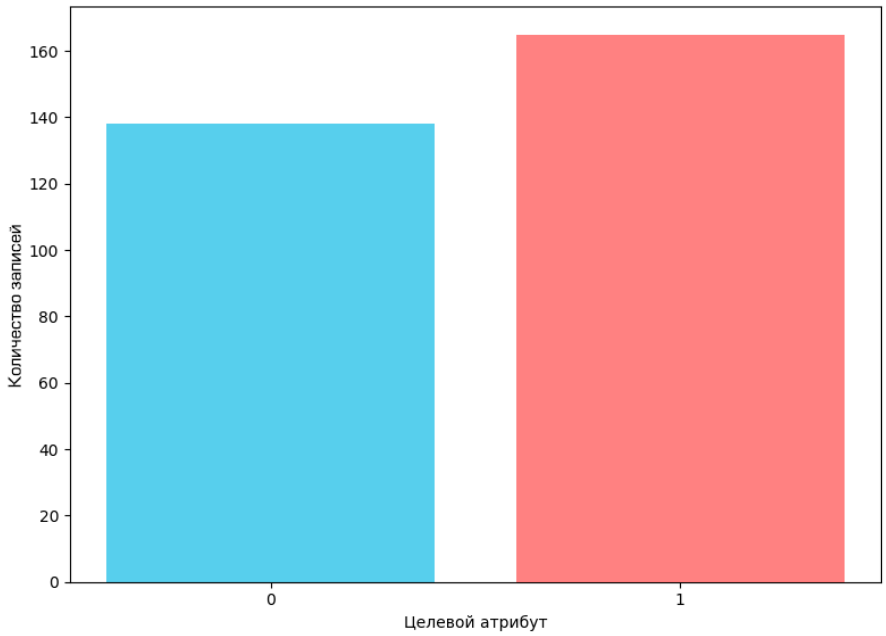


Рисунок 1.4 – Равномерность целевого атрибута в наборе данных

Из рисунка 1.4 видно, что классы в наборе практически сбалансированы.

1.5.3 Тестирование классификаторов

С целью выбора классификатора, показывающего наилучшие результаты в решении поставленной задачи, было проведено тестирование, суть которого заключалось в следующем:

1. Набор данных о сердечных заболеваниях Кливленда случайным образом разбивался на обучающую (67%) и тестовую выборку (33%).
2. Далее на основе обучающей выборке строились классификационные модели.
3. После чего на тестовом множестве проверялось качество построенных моделей, соотнеся их решение с заведомо известным правильным.

Схема проведения тестирования представлена на рисунке 1.5.

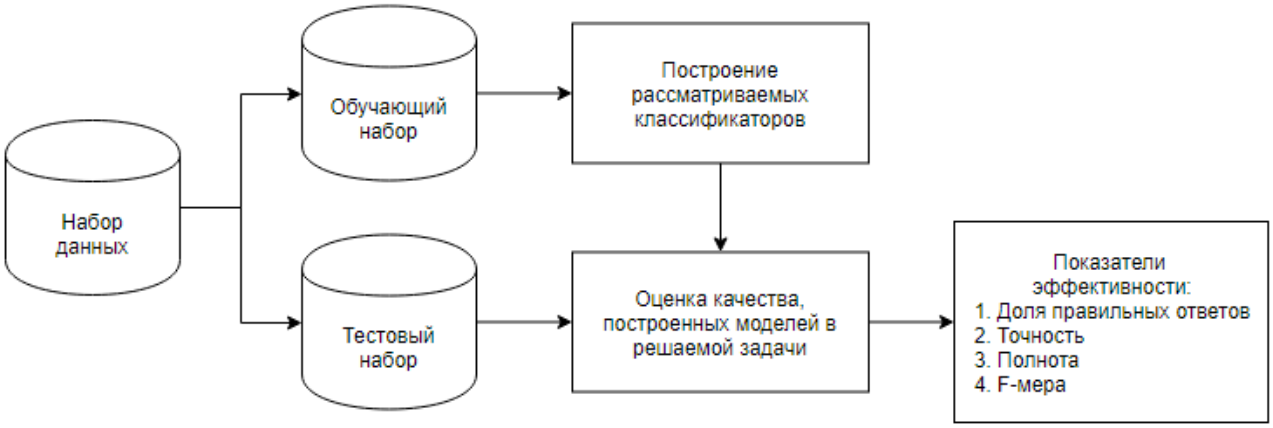


Рисунок 1.5 – Схема тестирования классификаторов

Для тестирования алгоритмов классификации в решении задачи прогнозирования риска сердечно-сосудистых заболеваний использовалась библиотека scikit-learn, реализованная на языке Python [9].

Библиотека была выбрана, так как:

- предоставляет реализации всех необходимых алгоритмов классификации;
- предоставляет средства для визуализации и анализа полученных данных.

В экспериментах использовались следующие модули из библиотеки scikit-learn для построения классификационных моделей разными методами:

1. DecisionTreeClassifier. Модуль, реализующий алгоритм CART, который, соответственно, строит дерево принятия решений. Дерево строилось без отсечений.

2. RandomForestClassifier. Модуль, реализующий случайный лес. Оптимальное количество деревьев в случайном лесу выбиралось с помощью класса GridSearchCV, осуществляющего поиск наилучшего набора параметров, доставляющих минимум ошибки перекрестного контроля. По умолчанию рассматривался 3-кратный перекрестный контроль. Поиск наилучшего показателя параметра проводился среди значений [10, 50, 100, 150, 200]. Лучшим оказалось число в 50 деревьев. Ошибка перекрестного контроля составила 0.197%.

3. GaussianNB. Модуль, ответственный за реализацию наивного байесовского классификатора.

4. KNeighborsClassifier. Модуль, реализующий алгоритм k ближайших соседей. Основной параметр метода k ближайших соседей – это k. Выбор оптимального значения данного параметра осуществлялся с помощью того же класса, что и при поиске оптимального количества деревьев. Поиск наилучшего значения k проводился среди значений [1, 3, 5, 7, 10, 15, 20]. В качестве оптимального значения метод выбрал k равное 10. Ошибка перекрестного контроля составила 0.365%.

Для получение объективной картины тестирование проводилось десять раз при разных значениях параметра random_state метода train_test_split, что в свою очередь при каждом эксперименте влияло на случайное разбиение исходной выборки на обучающий и тестовый набор данных. Размерности обучающего и тестового множества при разбиении исходного были неизменно равны 0.67 и 0.33, соответственно. Среднеарифметические результаты метрик оценки качества классификационных моделей можно наблюдать в таблице 1.3.

Таблица 1.3 – Сравнение моделей

Алгоритм	**Доля правильных ответов**	**Точность**	**Полнота**	**F-мера**
Дерево решений	0.74	0.6875	0.75	0.7174
Случайный лес	0.83	0.7647	0.886	0.821
Наивный байесовский классификатор	0.79	0.725	0.840	0.779
Метод k-ближайших соседей	0.61	0.533	0.727	0.615

На рисунке 1.6 изображено графическое представление таблицы 1.3 в виде столбчатой диаграммы.

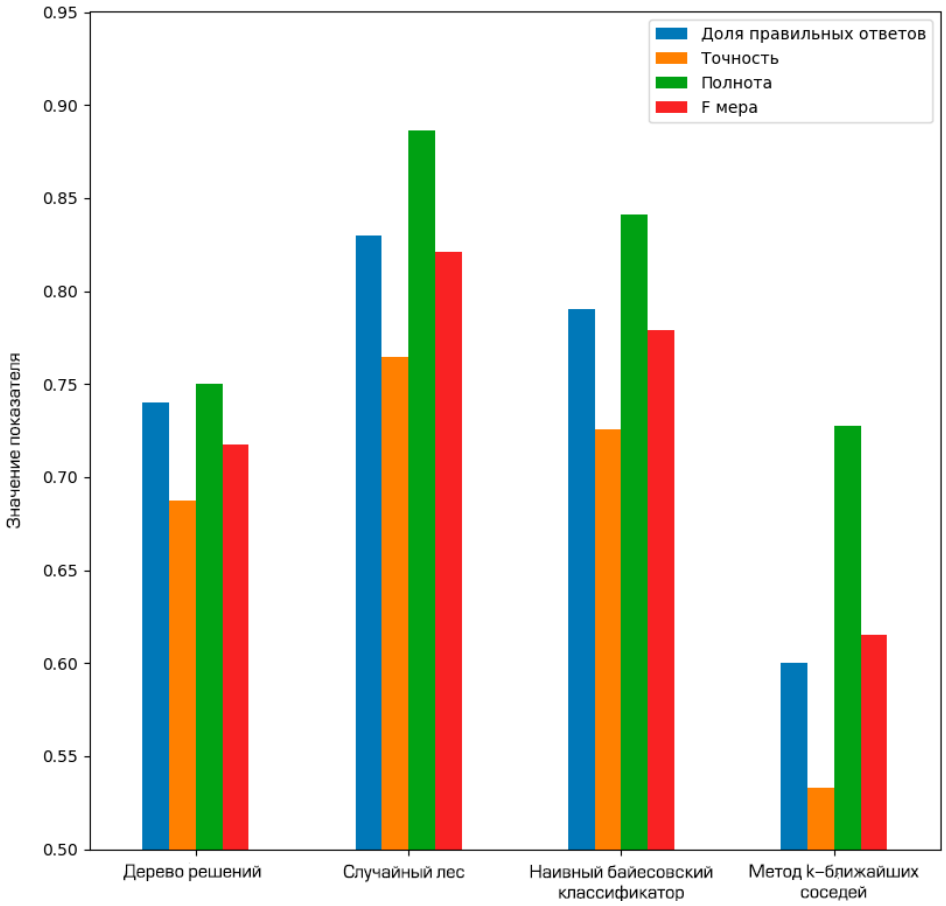


Рисунок 1.6 – Графическое сравнение моделей в решении поставленной задачи

В результате тестирования качества алгоритмов классификации в решении поставленной задачи наиболее эффективным оказался алгоритм

случайного леса. Таким образом, именно данный метод машинного обучения было решено использовать в прогнозировании риска сердечно-сосудистых заболеваний.

1.6 Формализация требований к разрабатываемой системе

Проведем декомпозицию задачи построения системы, способной прогнозировать риск сердечно-сосудистых заболеваний, рассмотрев разрабатываемую интеллектуальную информационную систему в виде отдельных взаимосвязанных подсистем.

Было выделено три ключевых компонента системы:

1. Подсистема для общения с пользователем. Система пользовательского интерфейса, обеспечивающая взаимодействие между интеллектуальной информационной системой и пользователем. Подсистема должна обеспечивать:

- эффективную обработку ввода и вывода, в ходе которой вводимые и выводимые данные обрабатываются быстро и в выразительной форме;
- распознавание непонимания между пользователем и системой, возникшее от неверных действий человека при пользовании последней, а также реагирование системы соответствующим образом на такие ситуации;
- «дружелюбие» к пользователю и интуитивность.

2. Подсистема построения классификационной модели. В качестве базового механизма вывода для поддержки принятия решений используется метод машинного обучения случайный лес.

3. Подсистема приобретения знаний. Предназначением данной подсистемы является добавление в базу знаний новых данных и модификация уже имеющихся.

Сформируем ряд требований к разрабатываемой системе. В первую очередь начнём с функциональных требований:

1. Входные переменные. К входным данным прототипа системы прогнозирования сердечно-сосудистых заболеваний относятся данные пациентов, являющиеся факторами, приводящих к возникновению инфаркта.

2. Результат работы системы. Результат работы системы должен быть представлен в виде одного из значений целевой переменной (Низкий риск или Высокий риск), которое принимает целевая переменная в зависимости от рассматриваемого пациента.

3. Создание модели взаимодействия с пользователями системы. Должен быть определен непосредственный интерфейс взаимодействия с системой, в данном случае графический пользовательский интерфейс.

4. Требования к надежности. Программный продукт должен быть надежным, хранить всю информацию в базах данных. При вводе данных должна осуществляться проверка на корректность введённых данных.

Нефункциональные требования:

1. Язык разработки. Система должна быть разработан на языке программирования Java.

2. Совместимость с Windows.

1.7 Выводы по первой главе

В данной главе была сформулирована задача, которую необходимо решить, которая заключается в прогнозировании риска сердечно-сосудистых заболеваний. Выявлена актуальность и злободневность выбранной темы бакалаврской работы, состоящая в том, что Российская Федерация находится на первых позициях в списках по смертности от вида данных заболеваний, что вызывает потребность в решении данной проблемы.

Проведен поиск и анализ смежных статей на тему прогнозирования сердечно-сосудистых заболеваний разными методами машинного обучения, среди которых были выбраны наиболее эффективные. В ходе тестирования выбранных методов в поставленной задачи с лучшей стороны показал себя метод случайного леса, собственно, его и было решено использовать для решения поставленной задачи.

Также были выявлены требования к разрабатываемой интеллектуальной информационной системе.

ГЛАВА 2 ПРОГРАММНАЯ РЕАЛИЗАЦИЯ СИСТЕМЫ ПРОГНОЗИРОВАНИЯ РИСКА СЕРДЕЧНО-СОСУДИСТЫХ ЗАБОЛЕВАНИЙ

2.1 Проектирование интеллектуальной информационной системы

Основной задачей данного подраздела является проектирование многопользовательского приложения для прогнозирования риска сердечно-сосудистых заболеваний методом случайного леса, пользователями которого являются врачи-кардиологи и администраторы системы.

2.1.1 Проектирование архитектуры системы прогнозирования риска сердечно-сосудистых заболеваний

При разработке системы прогнозирования риска сердечно-сосудистых заболеваний было решено использовать двухуровневую «клиент-серверную» архитектуру разновидности «толстый клиент», где логика представления данных и бизнес-логика размещаются на клиенте, который общается с логикой хранения и накопления данных на сервере, используя язык структурированных запросов SQL [10] (рисунок 2.1). Для обеспечения независимого соединения клиентской части и сервера базы данных был выбран стандарт JDBC [11].

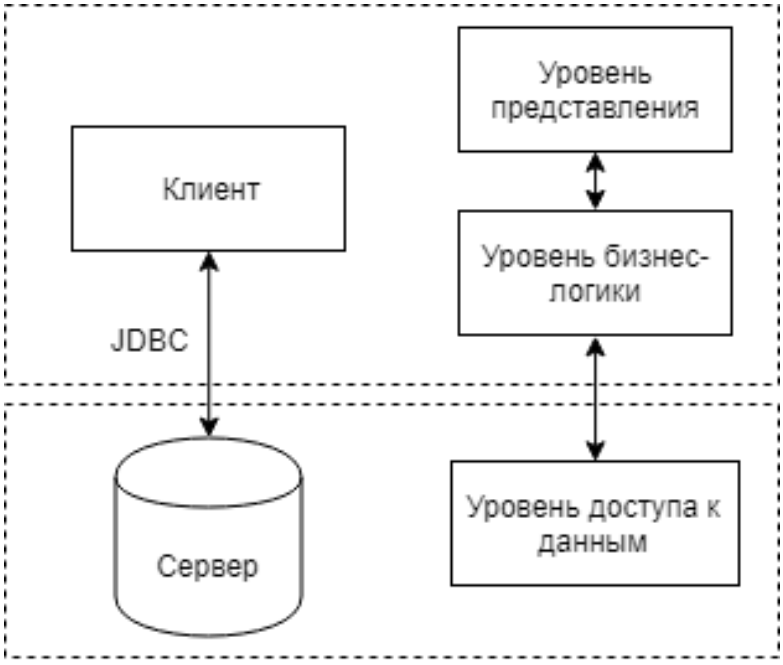


Рисунок 2.1 – Двухуровневая архитектура «клиент-сервер»

Уровень представления содержит механизмы, обеспечивающие возможность взаимодействия между пользователем и интеллектуальной информационной системой. На данном логическом уровне выполняются процессы, связанные с управлением системой ввода/вывода, обработкой входных данных и формированием представлений.

Уровень бизнес-логики содержит реализацию бизнес-моделей, инкапсулирующих всю вычислительную логику приложения, а также объекты, полученные на уровне доступа к данным и передаваемые уровнем представления. Именно на этом уровне выполняется прогнозирование риска сердечного-сосудистых заболеваний путём построения классификационной модели методом машинного обучения случайного леса.

Уровень доступа к данным содержит модели сущностей, хранящиеся в базе

При разработке архитектуры приложения было решено воспользоваться шаблоном проектирования Model-View-Controller (MVC). Данный шаблон позволяет отделить бизнес-логику (модели) от её визуализации (представления, вида). MVC подразделяет архитектура кода на три основных части [12]:

- модель (model) – является объектной моделью какой-либо предметной области, включающей в себя данные и методы работы с этими самыми данными, отвечает на запросы из контроллера, возвращая данные и/или изменяя своё состояние, при этом модель не содержит в себе информации, как эти данные можно отобразить, а также не «контактирует» с пользователем лично.
- представление (view) - отвечает непосредственно за визуализацию информации;
- контроллер (controller) - обеспечивает взаимодействие между пользователем системы и самой системой: контролирует ввод данных пользователем, используя модель и представление для реализации необходимой реакции.

Схема шаблона проектирования MVC представлена на рисунке 2.2.

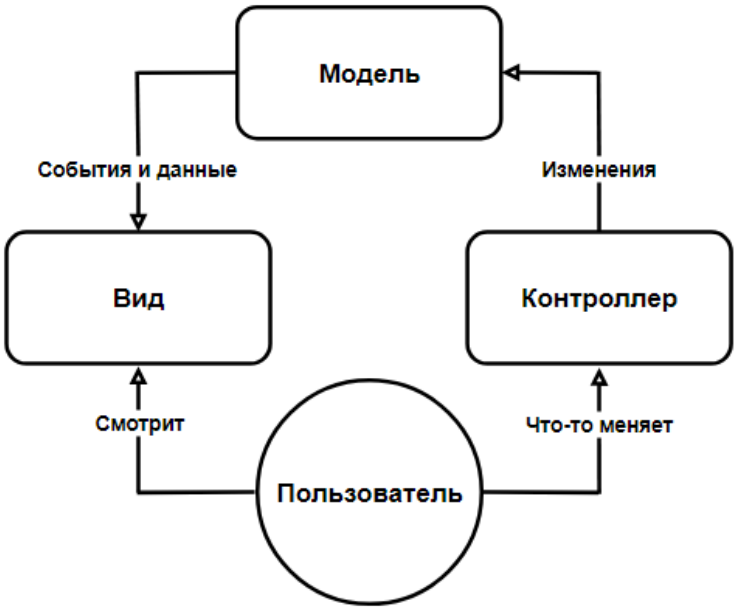


Рисунок 2.2 – Схема шаблона проектирования MVC

После проектирования архитектуры приложения и её особенности в виде использования шаблона проектирования MVC, было решено перейти к работе над пользовательским интерфейсом системы.

2.1.2 Проектирование пользовательского интерфейса системы

Определим действующие лица и как именно они взаимодействуют с системой, используя диаграмму вариантов использования системы [13], изображенную на рисунке 2.3.

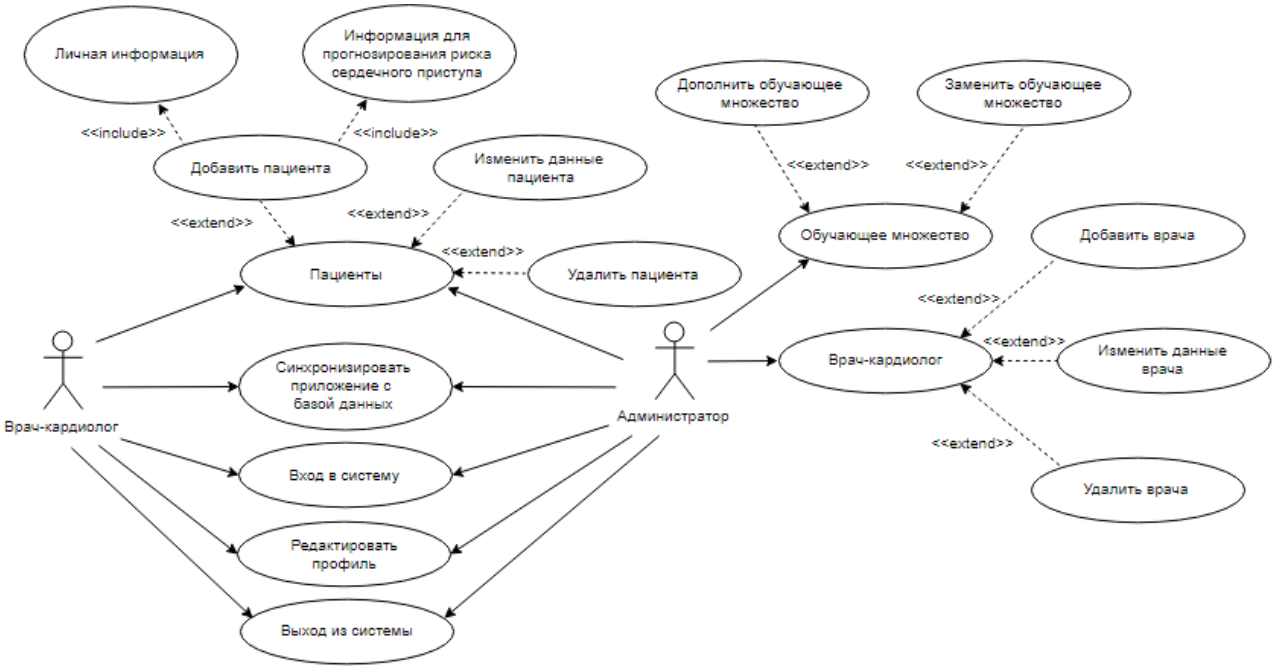


Рисунок 2.3 – Диаграмма вариантов использования системы

Построив диаграмму прецедентов, мы можем перейти к созданию структурной схемы интеллектуальной информационной системы**,** определяющей функциональные блоки приложения (рисунке 2.4).

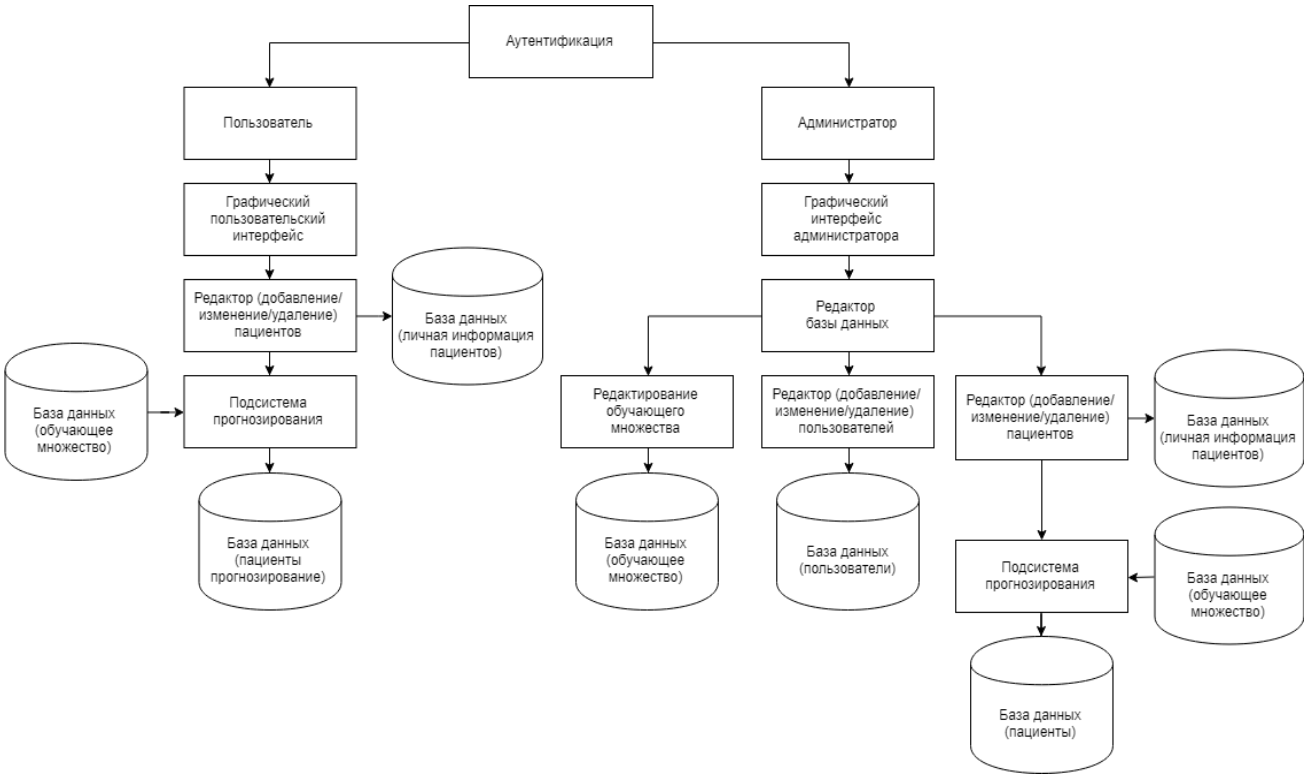


Рисунок 2.4 – Структурная схема разрабатываемой системы

Как видно на рисунке выше от приложения требуется наличие двух интерфейсов, выводящихся на экран в соответствии с положенными физическому лицу правами после его удачной аутентификации в системе.

Пользовательский интерфейс позволяет врачу-кардиологу добавлять в систему новых пациентов, а также изменять и удалять информацию уже имеющихся в системе, привязанных к нему. Вдобавок данный интерфейс предоставляет возможность изменять личные данные самого пользователя.

Интерфейс администратора имеет все те же самые функции, что и пользовательский интерфейс, плюс, ряд ещё некоторых, такие как редактирование обучающей выборки, а также добавление новых пользователей в систему, изменение данных или удаление уже числящихся в системе.

2.1.3 Проектирование базы данных

База данных является основным источником получения необходимой информации, а также средством для хранения и изменения необходимых данных. Вся информации в БД разделена на таблицы, которые являются сущностями того или иного бизнес-объекта или процесса [14].

Исходя из анализа предметной области можно выделить такие сущности как: пользователи системы, обучающее множество, личная информация пациентов, данные пациентов для прогнозирования риска сердечно-сосудистых заболеваний, краткая информация о пациентах. Определим атрибуты для выделенных выше сущностей, описание которых представлено в таблице 2.1.

Таблица 2.1 – Атрибутивный состав сущностей

Сущность	**Атрибуты**
Пользователи системы	Идентификационный номер
	Имя
	Фамилия
	Отчество
	Электронная почта
	Логин
	Пароль
	Права в системе
Обучающее множество	Идентификационный номер
	Возраст
	Пол
	Тип боли в груди
	Артериальное давление в покое
	Сывороточный холестерин
	Уровень сахара в крови натощак
	Результат электрокардиографии
	Максимальная частота сердечных сокращений
	Стенокардия
	Депрессия сегмента ST на электрокардиограмме
	Наклон сегмента ST на электрокардиограмме

Сущность	Атрибуты
	Количество крупных сосудов, окрашенных с помощью флюороскопии
	Результаты стресс-теста таллия, измеряющего приток крови к сердцу
	Риск сердечно-сосудистых заболеваний
Личная информация пациента	Идентификационный номер
	Имя
	Отчество
	Фамилия
	Год рождения
	Телефон
	Электронная почта
	Идентификационный номер, лечащего врача
Данные пациентов для прогнозирования риска сердечно-сосудистых заболеваний	Содержит все те же самые атрибуты, что и сущность «Обучающее множество»
Краткая информация о пациентах	Идентификационный номер
	Имя
	Отчество
	Фамилия
	Риск сердечно-сосудистых заболеваний

База данных для интеллектуальной информационной системы спроектирована и готова к следующему этапу – разработке.

2.2 Разработка интеллектуальной информационной системы

Перейдём непосредственно к программной реализации выше спроектированного приложения, используя для написания исходного кода клиентской части объектно-ориентированный язык программирования Java и интегрированную среду разработки NetBeans [15].

2.2.1 Реализация собственного классификатора

Как было установлено, в ходе выбора классификационной стратегии из первой главы, наиболее эффективным методом машинного обучения в решении задачи прогнозирования риска сердечно-сосудистых заболеваний является метод случайного леса. Таким образом, было решено разработать собственную программную реализацию случайного леса на объектно-ориентированном языке программирования Java.

При работе над программной реализацией случайного леса в арсенале присутствовали только стандартные библиотеки Java, в результате чего были созданы следующие классы, представленные в таблице 2.2.

Таблица 2.2 – Классы, реализующие метод случайного леса

Класс	Предназначение
RandomForest	Главный класс, содержащий методы для построения случайного леса.
DecisionTreeClassifier	Класс, ответственный за построение классификационного дерева. Использует критерий расщепления Gini.
DecisionTreeRegressor	Класс, ответственный за построение регрессионного дерева. Использует для критерий расщепления Gini.
GeneralMethods	Класс, содержащий методы необходимые для построения деревьев классами DecisionTreeClassifier, DecisionTreeRegressor.

Для визуализации выше рассмотренных классов использовались UML диаграммы (Приложения А, Б, В, Г).

Общий алгоритм разработанной программной реализации метода случайного леса для прогнозирования риска сердечно-сосудистых заболеваний изображен на рисунке 2.5.

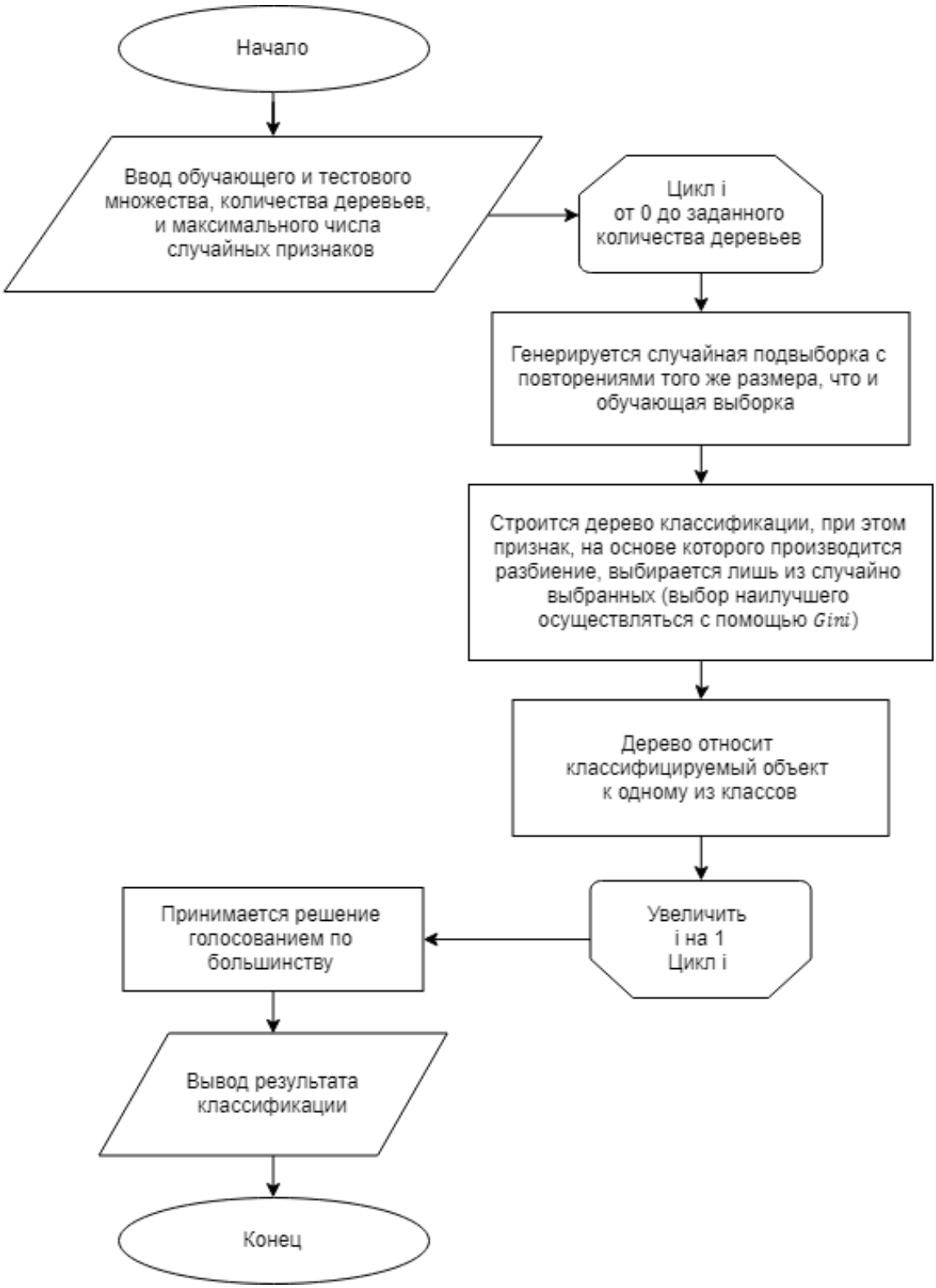


Рисунок 2.5 – Блок-схема прогнозирования риска сердечно-сосудистых заболеваний

В качестве обучающего множества, для построения классификационной модели, с целью прогнозирования риска сердечно-сосудистых заболеваний в

разработанной интеллектуальной информационной системы, выступил набор о сердечных заболеваниях из базы данных о сердечных заболеваниях Кливленда.

2.2.2 Создание графического представления программы

Для реализации графического интерфейса была выбрана платформа JavaFX для создания программных обеспечений с насыщенным графическим интерфейсом [16]. С целью удобства и быстроты создания графического интерфейса использовался инструмент JavaFX SceneBuilder [17].

Графическое часть представлена в файлах формата fxml, сформированных с помощью SceneBuilder, которые содержат сцены, являющиеся контейнерами для всех графических элементов. Созданные сцены соединяются и управляются с помощью контроллеров, которые в свою очередь взаимодействуют с основными классами программы.

В результате работы было создано три сцены с событиями для взаимодействия с пользователем

- сцена «Login» открывается при запуске приложения, с целью аутентификации и регистрации пользователя в системе (Приложение Д);
- сцена «Doctor» выводится на экран для пользователей, имеющих в системе права «доктора» (Приложение Е). Основным предназначением данной сцены является прогнозирование риска сердечно-сосудистых заболеваний;
- сцена «Admin» выводится на экран для пользователей, имеющих в системе права «администратора» (Приложение Ж). Данная сцена предоставляет ряд возможностей администрирования системы.

Также для обрабатывания событий элементов, размещенных в сценах выше были созданы три контроллера «LoginController», «DoctorController», «AdminController», обрабатывающие сцены «Login», «Doctor» и «Admin», соответственно.

2.2.4 Создание базы данных

Для хранения обучающего множества классификационной модели, а также других необходимых для работы системы данных было решено использовать реляционную базу данных [18]. Выбор для создания базы данных пал на MySQL, систему управления реляционной базы данных. Вся работа, связанная с базой данных, выполнялась в средстве визуализации базы данных для проектирования и моделирования баз данных для СУРБД MySQL под названием «MySQLWorkbench» [19].

На основании сущностей и их атрибутов, рассмотренных на стадии проектирования, было создано пять таблиц для хранения необходимых данных. Структурная схема разработанной базы данных изображена на рисунке 2.6.

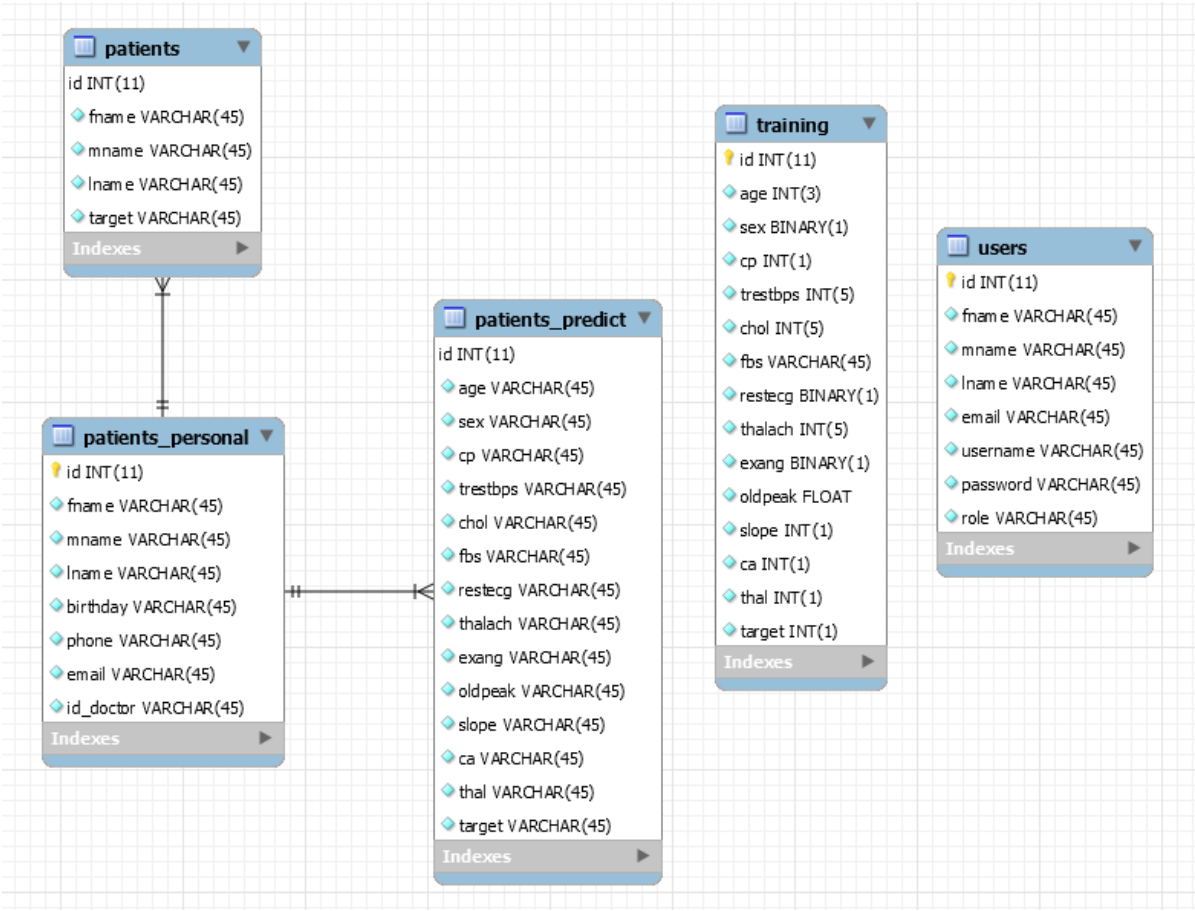


Рисунок 2.6 – Диаграмма базы данных

На рисунке 2.6 представлены следующие таблицы:

1) данные пользователей системы (users).
2) обучающее множество или база знаний (training);

3) личная информация пациентов (patients_personal);

4) данные пациентов для прогнозирования риска сердечно-сосудистых заболеваний или рабочая память (patients_predict);

5) краткая информация о пациентах (patients)

Работа с базой данный MySQL внутри приложения была реализована посредством запросов, посылаемых через специальный драйвер JDBC.

2.3 Выводы по второй главе

В данной главе проведено проектирование и разработка интеллектуальной информационной системы, решающей задачу прогнозирования риска сердечно-сосудистых заболеваний методом случайного леса.

В ходе этапа проектирования была спроектирована архитектура приложения, рассмотрена клиентская часть в виде диаграммы вариантов использования системы и структурной схемы, а также выделены сущности и их атрибуты для построения требуемых таблиц в базе данных.

В ходе этапа разработки была выполнена собственная реализация метода случайного леса, создан графический интерфейс, а также построена база данных необходимая для работы приложения. Вся клиентская часть писалась на объектно-ориентированном языке программирования Java.

ГЛАВА 3 ТЕСТИРОВАНИЕ РАЗРАБОТАННОЙ СИСТЕМЫ

3.1 Тестирование классификатора

Испытания классификатора проводились с использованием данных Кливленда о сердечных заболеваниях.

Для оценки качества классификационной модели, построенной методом случайного леса, применялись такие метрики, как доля правильно классифицированных записей (accuracy), точность (precision), полнота (recall) и F-мера. Данные метрики рассчитывались по формулам 1.8 - 1.11.

Суммарно было проведено десять экспериментов, результаты которых можно наблюдать в таблице 3.1.

Таблица 3.1 - Экспериментальные исследования

№	Доля правильно классифицированных объектов	Точность	Полнота	F-мера
1	0,84	0,839	0,87	0,854
2	0,88	0,894	0,894	0,894
3	0,81	0,843	0,796	0,819
4	0,84	0,83	0,892	0,862
5	0,67	0,60	0,98	0,746
6	0,78	0,753	0,890	0,816
7	0,81	0,765	0,924	0,837
8	0,85	0,892	0,847	0,869
9	0,83	0,7647	0,886	0,821
10	0,79	0,767	0,843	0,803

На рисунке 3.1 изображен график, основывающийся на данных, полученных по результатам экспериментов.

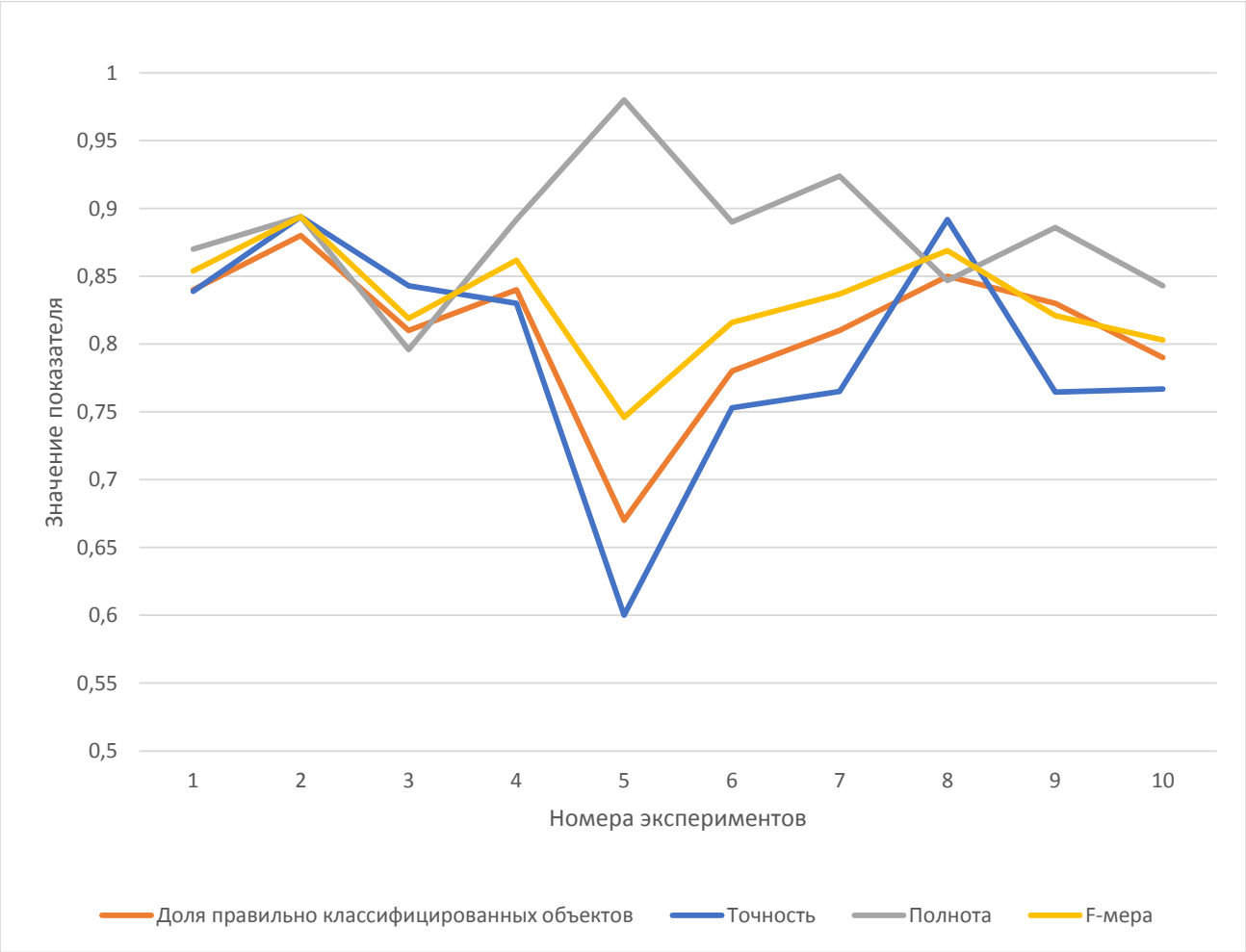


Рисунок 3.1 – График экспериментальных исследований

Как видно из результатов, реализованная модель имеет достаточно высокий показатель охвата, что даёт понять параметр полноты. Показатель точности имеет показатели чуть меньше, что означает, что присутствовали ложные срабатывания, однако, благодаря метрики F-мера можно сделать вывод, что классификатор в виде случайного леса эффективен в решении поставленной задачи.

3.2 Функциональное тестирование

Проведем тестирование фрагмента разработанной интеллектуальной информационной системы, а именно – прогнозирование риска сердечно-сосудистых заболеваний.

В первую очередь зайдём в систему с правами врача-кардиолога, после чего откроем раздел «пациенты», где выберем опцию «добавить» (рисунок 3.2).

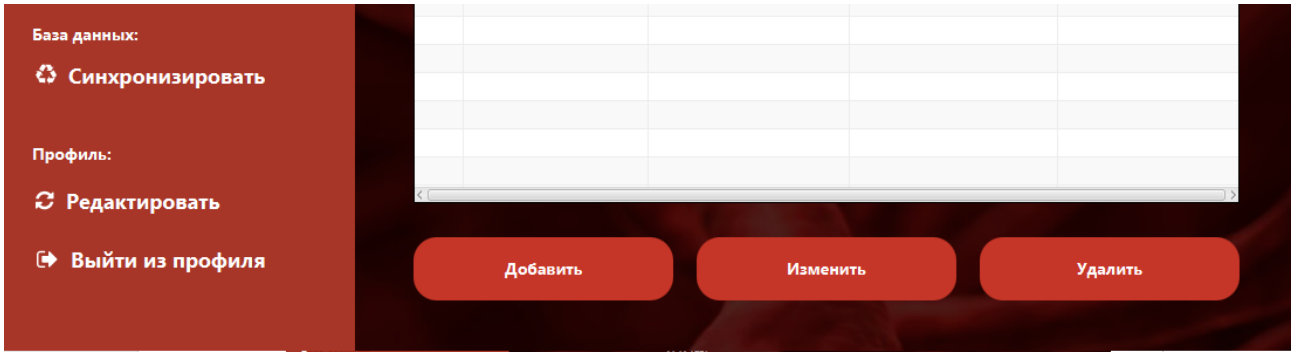


Рисунок 3.2 – Опции окна «Пациенты»

После чего открывается форма для ввода личной информации пациента, которую можно наблюдать на рисунке 3.3.

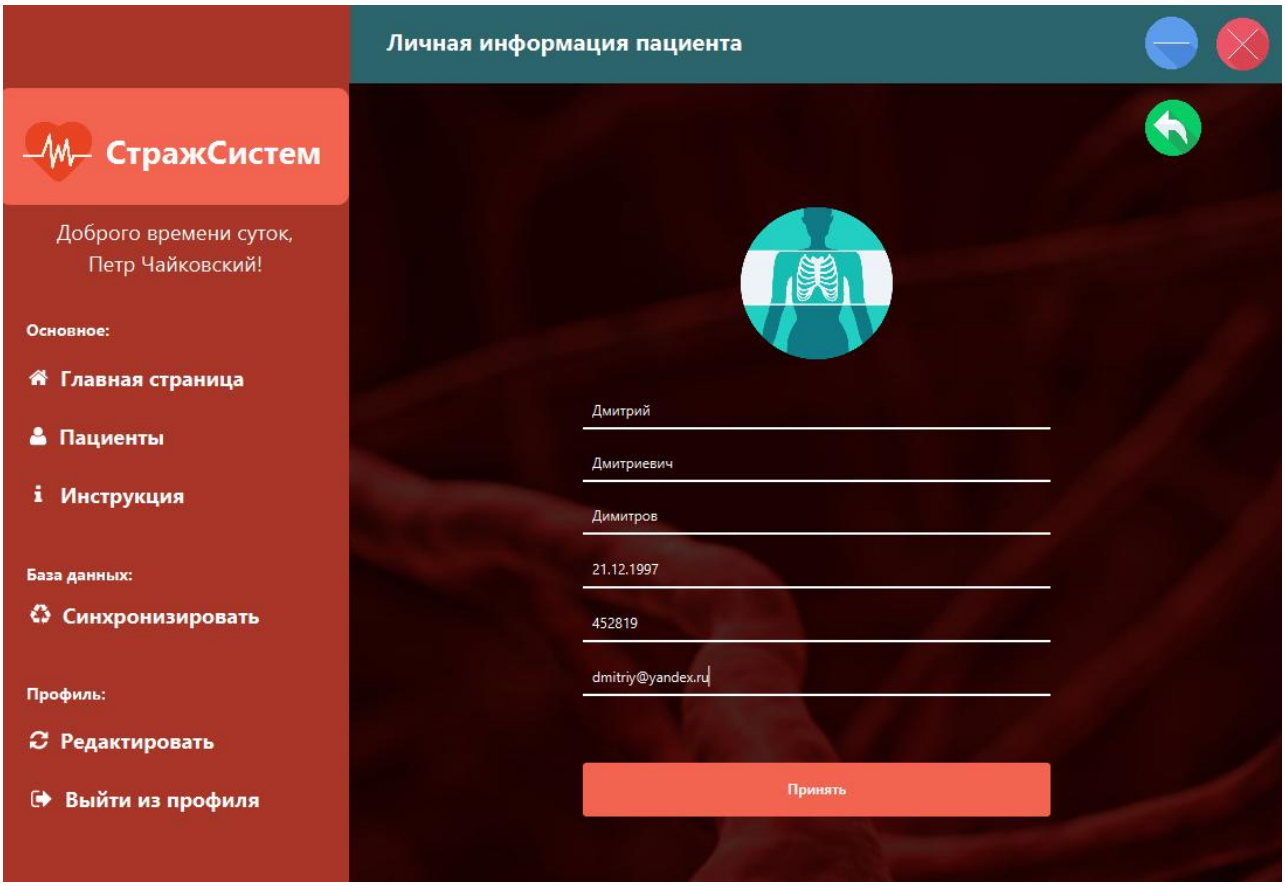


Рисунок 3.3 – Форма «личная информация пациента»

Подтвердив заполнение всех текстовый полей, нажатием кнопки «Подтвердить», открывается следующая форма ввода необходимой

информации для совершения прогноза риска возникновения у пациента сердечно-сосудистых заболеваний (рисунок 3.4).

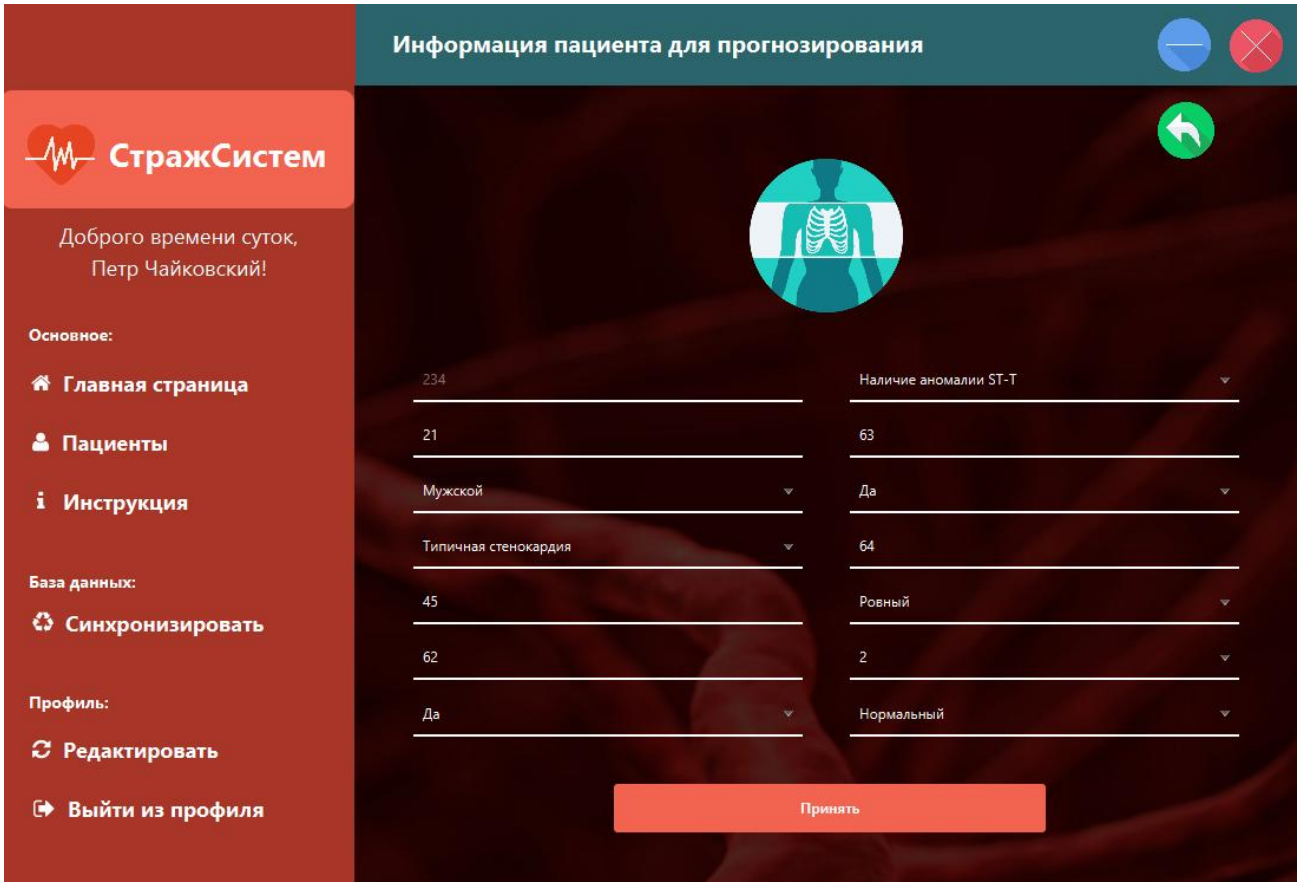


Рисунок 3.4 – Информация для прогнозирования риска заболеваний сердца

В очередной раз подтверждаем ввод данных после чего разработанное приложение снова возвращает пользователя в раздел «пациенты», где можно увидеть добавленного в систему пациента с уже вынесенным ему прогнозом риска возникновения сердечно-сосудистых заболеваний. Полученный результат представлен на рисунке 3.5.

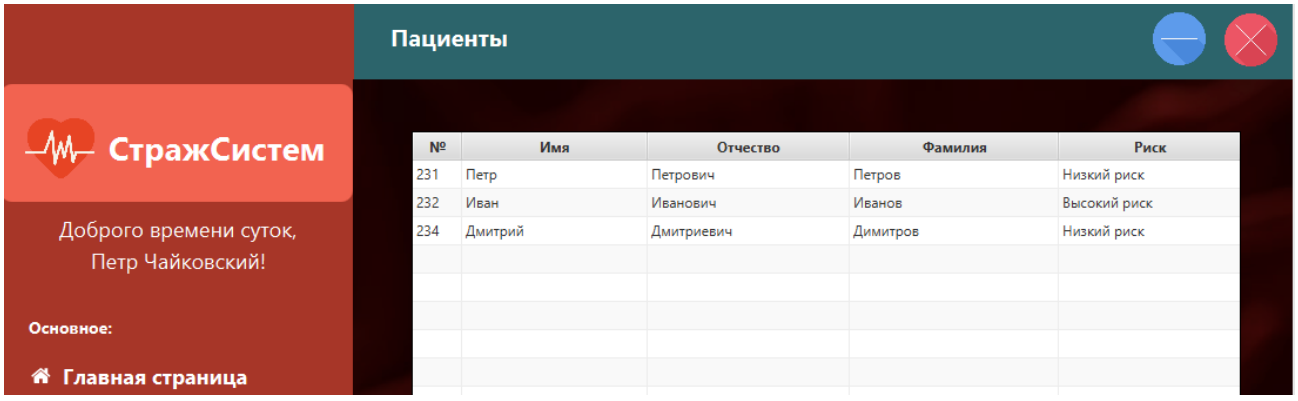


Рисунок 3.5 – Результат прогнозирования

Функциональное тестирование показало, что разработанное приложения для прогнозирования риска сердечно-сосудистых заболеваний, работает исправно.

3.3 Выводы по третьей главе

В ходе тестирования классификационной модели, построенной методом случайного леса, были проведены экспериментальные исследования, посвященные оценки её качества. В результате ряда экспериментов получены следующие среднеарифметические результаты: точность – 0,79, полнота – 0,88 F-мера – 0,83 и доля правильно классифицированных записей – 0,81, что позволяет судить о эффективности классификатора в решении задачи прогнозирования риска сердечно-сосудистых заболеваний.

При функциональном тестировании программных сбоев не было обнаружено, все разработанные модули работоспособны.

ЗАКЛЮЧЕНИЕ

В результате выполнения бакалаврской работы для решения задачи прогнозирования риска сердечно-сосудистых заболеваний был предложен метод случайного леса. Реализация предложенного мной в данной работе способа решения поставленной задачи представляет собой построение классификационной модели, с помощью метода случайного леса. В ходе работы была разработана собственная программная реализация данного метода на объектном-ориентированном языке Java. Доля правильно классифицированных записей на тестовой выборке данным классификатором составила и F-мера составили 0,81 и 0,83, соответственно, что позволяет сделать выводы об эффективности данного способа решения исследуемой задачи.

В первой части бакалаврской работы было приведено обоснование необходимости и цели создания интеллектуальной информационной системы для решения поставленной задачи, проведен обзор и анализ технологий искусственного интеллекта, а также была выбрана стратегия решения задачи прогнозирования риска сердечно-сосудистых заболеваний.

Вторая часть включает в себя проектирование интеллектуальной информационной системы и её программную реализацию.

В третьей части произведено тестирование работоспособности и эффективности разработанной интеллектуальной информационной системы.

СПИСОК ИСПОЛЬЗУЕМОЙ ЛИТЕРАТУРЫ

1. Об актуальных проблемах борьбы с сердечно-сосудистыми заболеваниями // Совет Федерации Федерального Собрания РФ. – М., 2015 – 108 с.

2. Всё о сердце [Электронный ресурс] // Ассоциация сердечно-сосудистых хирургов России Секция "Кардиология и визуализация в кардиохирургии" – Режим доступа: http://heart-master.com/for_patients/about_heart/ (Дата обращения: 12.05.2019).

3. Заболевания сердечно-сосудистой системы: виды и особенности [Электронный ресурс] // Самоздрав Дыхательный Тренажер – Режим доступа: https://samozdrav.ru/blog/zabolevaniya-serdechno-sosudistoy-sistemy/ (Дата обращения: 07.05.2019).

4. Здравоохранение в России. 2017 // Стат.сб./Росстат. – М., 2017. – 21 с.

5. Прикладная статистика: Классификация и снижение размерности / Айвазян С.А. и др. - М.: Финансы и статистика, 1989. - 607 с.

6. Барсегян А.А. Технологии анализа данных: Data Mining, Visual Mining, Text Mining, OLAP/ А.А. Барсегян – СПб.: БХВ-Петербург, 2007. – 284 с.

7. «Деревья решений – общие принципы работы» [Электронный документ] // BaseGroup Labs Systems – Режим доступа: http://www.basegroup.ru/library/analysis/tree/description/ (Дата обращения: 14.04.2019).

8. Информационный поиск [Электронный ресурс] // Википедия – Режим доступа: https://ru.wikipedia.org/?oldid=93657750 (Дата обращения: 29.05.2019).

9. Документация по библиотеке scikit-learn для машинного обучения с Python [Электронный ресурс] — Режим доступа: http://scikit-learn.org/stable/ (Дата обращения: 14.04.2019).

10. Избачков Ю.С. Информационные системы: Учебник для вузов/ Ю.С.Избачков, Петров В.Н. - Санкт-Петербург, 2006. - 656 с.

11. Соломон М., Мориссо-Леруа Н., Басу Дж. Oracle. Программирование на языке Java. — М.: Издательство «Лори», 2010. — 512 с.

12. Примеры объектно-ориентированного проектирования. Паттерны проектирования. / Э. Гамма, Р. Хелм, Р. Джонсон и др.; пер. с анг. А. Слинкина. – СПб.: Питер, 2001 – 368с.

13. Розенберг Д., Скотт К. Применение объектного моделирования с использованием UML и анализ прецедентов - М.: "ДМК Пресс", 2002. - 160 с.

14. Голицына, О.Л. Базы данных: Учебное пособие / О.Л. Голицына, Н.В. Максимов, И.И. Попов. - М.: Форум, 2012. - 400 с.

15. Монахов, В. Язык программирования Java и среда NetBeans / В. Монахов. - М.: БХВ-Петербург, 2011. - 704 с.

16. Машнин, Тимур JavaFX 2.0. Разработка RIA-приложений / Тимур Машнин. - М.: БХВ-Петербург, 2017. - 320 с.

17. Загоруйко Н.Г. Прикладные методы анализа данных и знаний. Новосибирск: ИМ СО РАН, 1999. - 270 с.

18. Scene Builder [Электронный ресурс] // Gluon – Режим доступа: http://gluonhq.com/open-source/scenebuilder (дата обращения: 01.06.2019).

19. Информационные технологии. Основы работы с реляционной БД Oracle. - М.: McGraw-Hill, 2002. - 200 с.

20. Яргер, Р.Дж. MySQL и mSQL: Базы данных для небольших предприятий и Интернета / Р.Дж. Яргер, Дж. Риз, Т. Кинг. - М.: СПб: Символ-Плюс, 2015. - 560 с.

21. Cardiovascular disease [Электронный ресурс] // World Health Organization – Режим доступа: https://www.who.int/cardiovascular_diseases/ru/ (Дата обращения: 05.06.2019).

22. K. Srinivas, B. Kavitha Rani and Dr. A. Govrdhan, “Application of Data Mining Techniques in Healthcare and Prediction of Heart Attacks”, International Journal on Computer Science and Engineering, Vol. 02, No. 02, pp. 250 - 255, 2011.

23. M. Marimuthu, M. Abinaya, K. S. Hariesh, K. Madhankumar and V. Pavithra. A Review on Heart Disease Prediction using Machine Learning and Data

Analytics Approach. International Journal of Computer Applications 181(18):20-25, September 2018.

24. Dr. D. Raghu. T. Srikanth Ch. Raja Jacub,"Probability: based Heart Disease Prediction using Data Mining Techniques" IJCST Vol. 2, Issue 4, Oct - Dec. 2011, ISSN: 0976-8491 (Online) | ISSN: 2229-4333 (Print).

25. Isra'a Ahmed Zriqat, Ahmad Mousa Altamimi, Mohammad Azzeh. A Comparative Study for Predicting Heart Diseases Using Data Mining Classification Methods. International Journal of Computer Science and Information Security (IJCSIS), Vol. 14, No. 12, December 2016.

26. Breiman, L., Friedman, J., Stone, C.J. and Olshen, R.A. Classification and Regression Trees. – 1984. – Taylor & Francis. – 368 c.

27. Hastie, T., Tibshirani R., Friedman J. Chapter 15. Random Forests // The Elements of Statistical Learning: Data Mining, Inference, and Prediction. — 2nd ed. – Springer-Verlag, 2009. – 746 p.

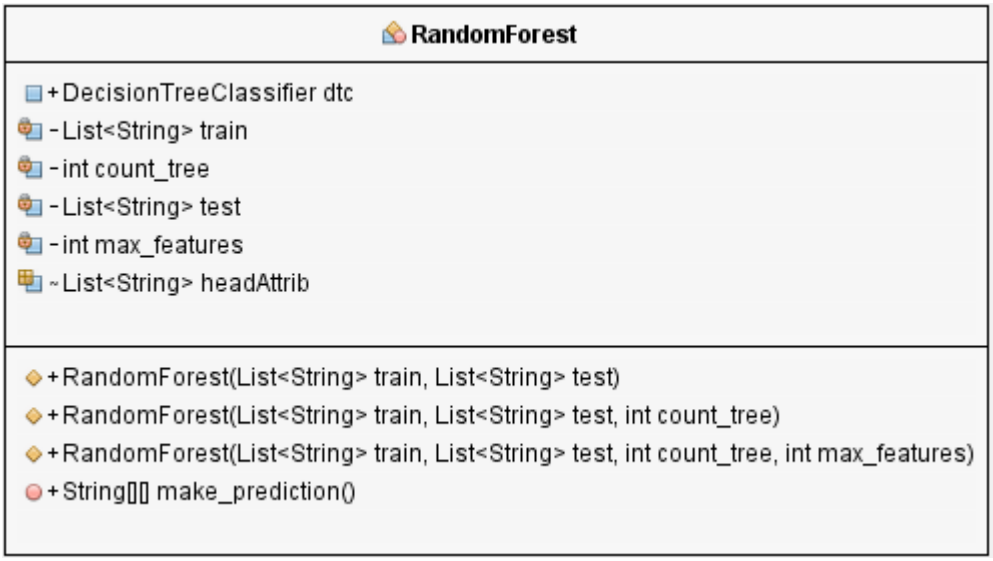
28. Rish, I. An empirical study of the naive Bayes classifier // IJCAI 2001 Workshop on Empirical Methods in Artificial Intelligence. – т.3. № 22. – 2001. – с. 41-46.

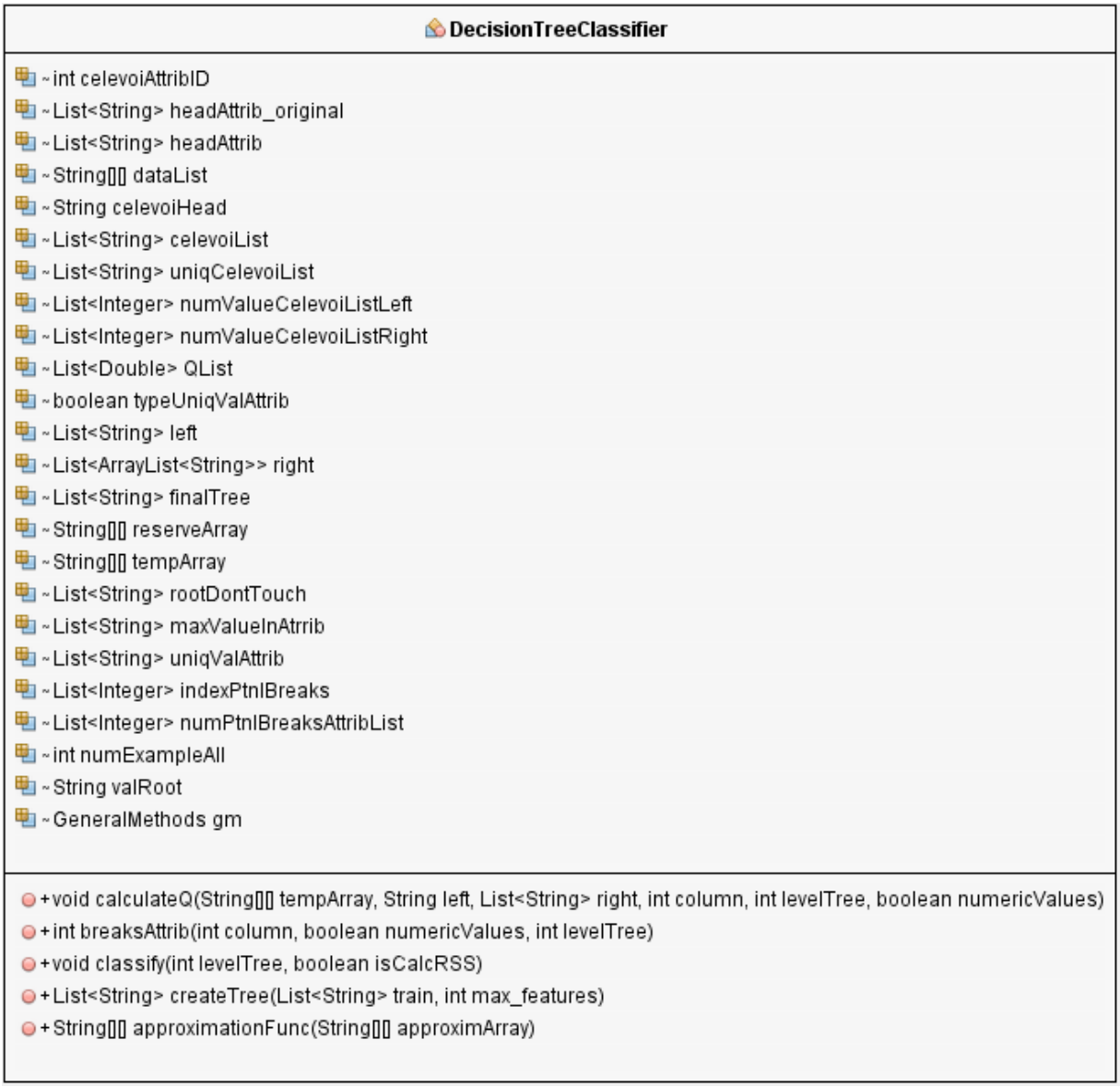
29. Brett Lantz. Machine Learning with R. Pack Publishing. – 2013. – 375 с.

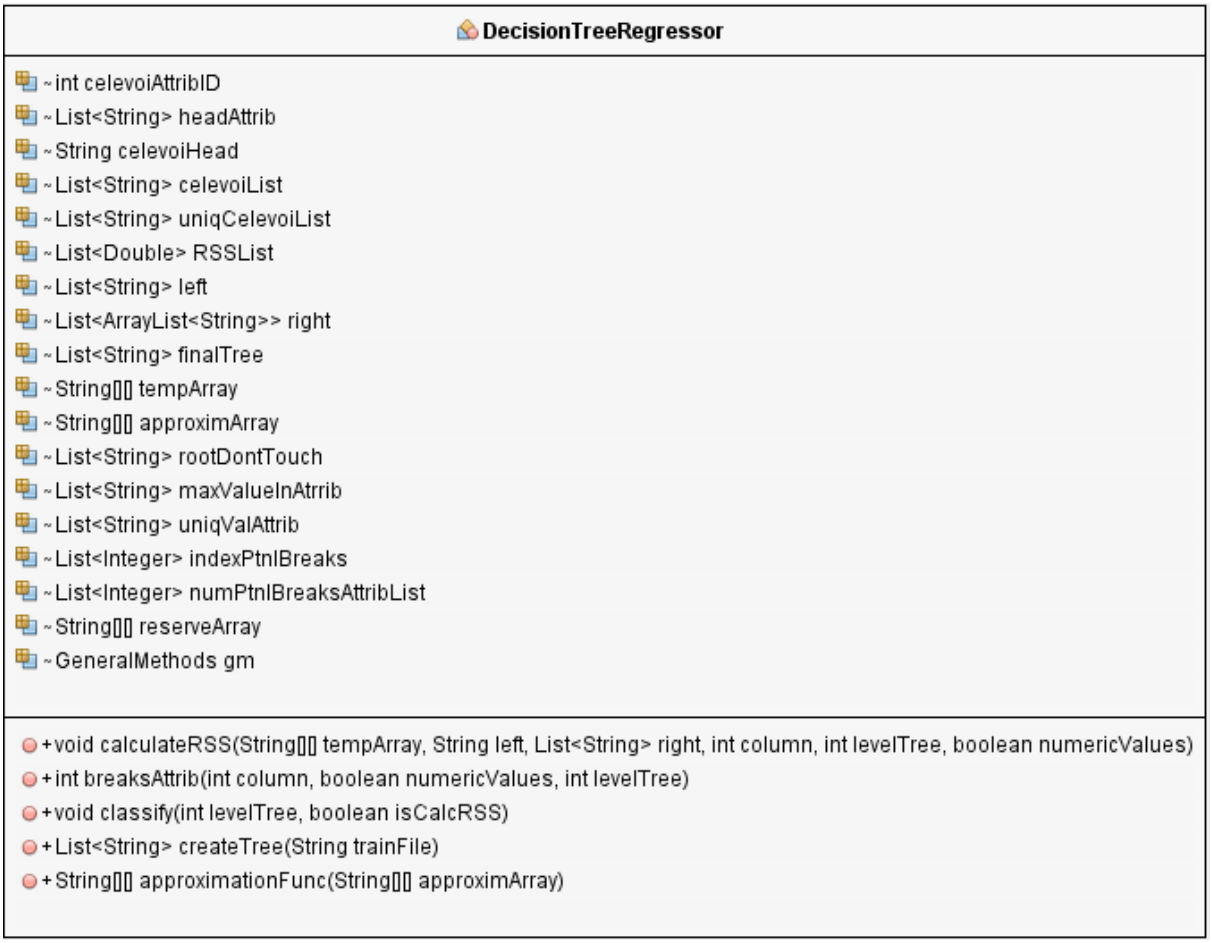
30. Statlog (Heart) Data Set [Электронный документ] // UCI Machine Learning Repository. Center for Machine Learning and Intelligent Systems – Режим доступа: http://archive.ics.uci.edu/ml/datasets/statlog+(heart) (Дата обращения: 28.05.2018).

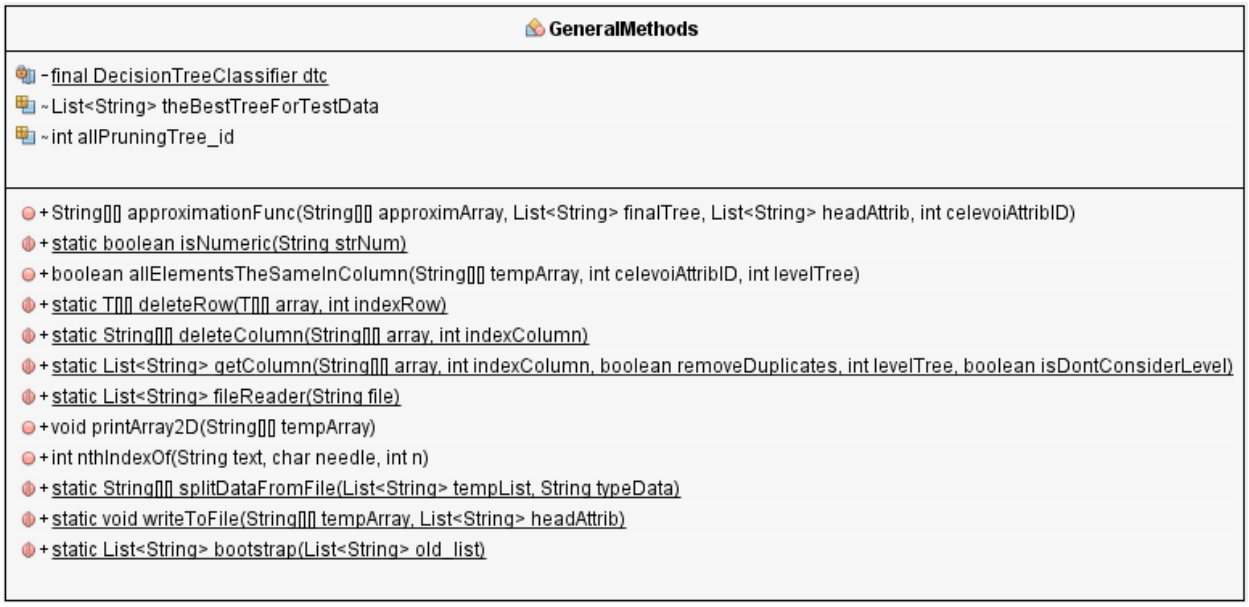
ПРИЛОЖЕНИЕ А

Основные классы приложени



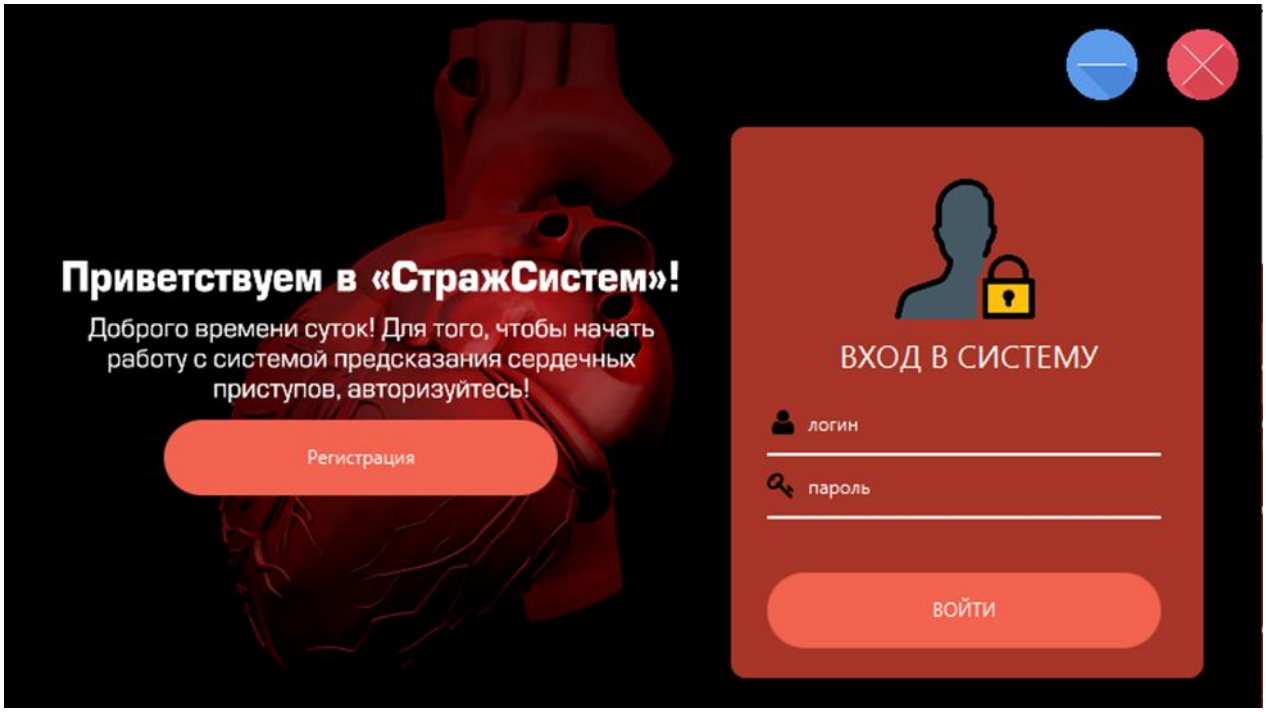


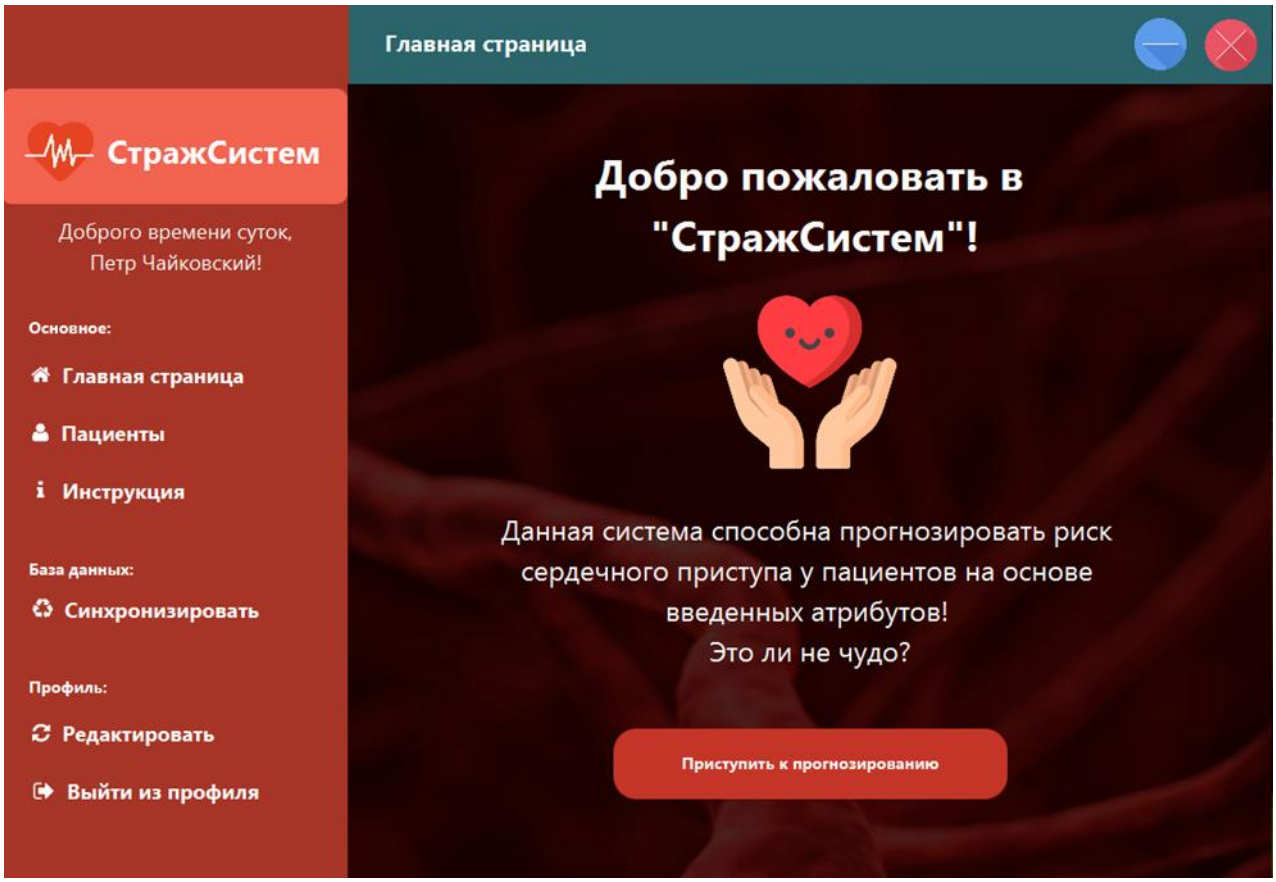

DecisionTreeRegressor
~int celevoiAttribID
~List<String> headAttrib
~String celevoiHead
~List<String> celevoiList
~List<String> uniqCelevoiList
~List<Double> RSSList
~List<String> left
~List<ArrayList<String>> right
~List<String> finalTree
~String[][] tempArray
~String[][] approximArray
~List<String> rootDontTouch
~List<String> maxValueInAtrrib
~List<String> uniqValAttrib
~List<Integer> indexPtnlBreaks
~List<Integer> numPtnlBreaksAttribList
~String[][] reserveArray
~GeneralMethods gm
+void calculateRSS(String[][] tempArray, String left, List<String> right, int column, int levelTree, boolean numericValues)
+int breaksAttrib(int column, boolean numericValues, int levelTree)
+void classify(int levelTree, boolean isCalcRSS)
+List<String> createTree(String trainFile)
+String[][] approximationFunc(String[][] approximArray)


GeneralMethods
-final DecisionTreeClassifier dtc
~List<String> theBestTreeForTestData
~int allPruningTree_id
+String[][] approximationFunc(String[][] approximArray, List<String> finalTree, List<String> headAttrib, int celevoiAttribID)
+static boolean isNumeric(String strNum)
+boolean allElementsTheSameInColumn(String[][] tempArray, int celevoiAttribID, int levelTree)
+static T[][] deleteRow(T[][] array, int indexRow)
+static String[][] deleteColumn(String[][] array, int indexColumn)
+static List<String> getColumn(String[][] array, int indexColumn, boolean removeDuplicates, int levelTree, boolean isDontConsiderLevel)
+static List<String> fileReader(String file)
+void printArray2D(String[][] tempArray)
+int nthIndexOf(String text, char needle, int n)
+static String[][] splitDataFromFile(List<String> tempList, String typeData)
+static void writeToFile(String[][] tempArray, List<String> headAttrib)
+static List<String> bootstrap(List<String> old_list)

ПРИЛОЖЕНИЕ Б

Экранные формы работы интеллектуальной системы






Главная страница
СтражСистем
Доброго времени суток,
Клим Калугин!
Основное:
Главная страница
Врачи
Пациенты
Обучающее множество
База данных:
Синхронизировать
Профиль:
Редактировать
Выйти из профиля
Врачи
Пациенты
Обучение
Профиль