

МИНОБРНАУКИ РОССИИ
Федеральное государственное бюджетное образовательное учреждение
высшего образования
**«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИМЕНИ Н. Г. ЧЕРНЫШЕВСКОГО»**

Кафедра математической кибернетики и компьютерных наук

**РАЗРАБОТКА ПРИЛОЖЕНИЯ ДЛЯ ОЦЕНКИ РИСКОВ
ЗАБОЛЕВАНИЯ ПНЕВМОНИЕЙ**

АВТОРЕФЕРАТ БАКАЛАВРСКОЙ РАБОТЫ

студента 4 курса 411 группы
направления 02.03.02 — Фундаментальная информатика и информационные
технологии
факультета КНиИТ
Остроухова Даниил Григорьевича

Научный руководитель
доцент, к. ф.-м. н.

А. С. Иванов

Заведующий кафедрой
к. ф.-м. н.

С. В. Миронов

Саратов 2022

ВВЕДЕНИЕ

Актуальность темы. В настоящее время из-за пандемии Covid-19 возросла нагрузка на лечащих врачей, от их действий зависят жизни и здоровье большого числа людей. В связи с этим создается множество программных средств, призванных облегчить решения и увеличить эффективность действий докторов. Методы машинного обучения применяются все чаще для того, чтобы своевременно обратить внимание медика на риск развития какого-либо осложнения у пациента. Такие приложения помогают докторам увеличить эффективность лечения и снизить негативные последствия заболеваний. Разработка веб-приложения, обеспечивающего оценку рисков осложнения в виде пневмонии у пациентов с Covid-19 на основе методов машинного обучения, представляется актуальной задачей. Подобное приложение обеспечит удобный пользовательский интерфейс для описания анамнеза и позволит повысить объективность процесса диагностирования.

Цель бакалаврской работы — создание веб-приложения для ввода сведений о пациенте и для классификации пациентов по заболеваемости на основе машинного обучения.

Поставленная цель определила **следующие задачи**:

- анализ методов машинного обучения, подходящих для определения рисков заболевания пневмонией;
- построение классификационной модели на основе деревьев решений для оценки рисков заболевания; реализация подбора параметров модели на основе точности классификации;
- проектирование и разработка программного обеспечения для диагностирования заболеваемости пневмонией;
- апробация приложения на примере новых пациентов.

Методологические основы применения машинного обучения для решения классификационной задачи представлены в работах таких ученых, как: В. Вьюгин, П. Флах, Р. Саттон, Н. Нилссон, Т. Рашид, С. Рассел, Т. Хастис, Ш. Бастиан, Д. Барбер.

Теоретическая значимость бакалаврской работы. Теоретическая значимость бакалаврской работы заключается в рассмотрении различных методов машинного обучения, которые могут быть применены для решения поставленной задачи классификации медицинских данных, выборе наиболее подходящего

метода для конкретного набора данных — дерева решений, применении процесса метаоптимизации для увеличения точности работающего алгоритма.

Практическая значимость бакалаврской работы. Приложение, разработанное в ходе выполнения бакалаврской работы, позволит обеспечить удобный пользовательский интерфейс для описания анамнеза и повысить объективность процесса диагностирования для помощи работы лечащих врачей посредством машинного обучения, которое способно определять риски возникновения пневмонии у нового пациента.

Структура и объем работы. Бакалаврская работа состоит из введения, трех разделов, заключения, списка использованных источников и двух приложений. Общий объем работы — сто тридцать три страницы, из них пятьдесят три страницы — основное содержание, включая двадцать рисунков, цифровой носитель в качестве приложения, список использованных источников информации — двадцать наименований.

КРАТКОЕ СОДЕРЖАНИЕ РАБОТЫ

Первый раздел «Анализ возможностей методов машинного обучения для определения рисков заболеваемости пневмонией» посвящен разбору различных доступных методов машинного обучения для применения в решении поставленной задачи классификации.

В настоящее время существует множество задач, требующих способность определять принадлежность объекта к какому-либо классу. Такие проблемы называются задачами классификации. Обычно они состоят в том, чтобы с помощью конечного множества объектов, разделенных на классы, построить алгоритм, который способен корректно отнести произвольный объект из исходного множества к какому-либо из этих классов.

Для решения задач классификаций существуют следующие алгоритмы:

- наивный байесовский классификатор;
- метод опорных векторов;
- метод релевантных векторов;
- линейная регрессия;
- логистическая регрессия;
- искусственная нейронная сеть;
- метод k-ближайших соседей;
- метод условных случайных полей;
- скрытые модели Маркова;
- бустинг;
- случайный лес;
- дерево решений.

Дерево решений является алгоритмом машинного обучения с учителем. Однако в отличие от других подобных алгоритмов, данный метод применим для решения задач регрессии и классификации. Цель использования дерева решений — создать модель, способную путем изучения выведенных из данных обучения правил принятия решений прогнозировать принадлежность объекта к одному из представленных классов. Результат работы алгоритмов построения деревьев решений представляется пользователю в легко интерпретируемой форме, что положительно влияет на понимание обоснованности принятых алгоритмом решений.

Для начала построения дерева решений необходимо разделить классы от

целостного набора входных данных. Далее процесс конструирования разделяется на два этапа: «построение» и «сокращение» дерева. Во время процесса построения определяется каким будет критерий расщепления. Во время этапа сокращения принимается решение отсечения некоторых невостребованных ветвей дерева.

Разделяют следующие алгоритмы построения дерева решений:

- ID3;
- C4.5;
- C5.0;
- CART.

CART (Classification and Regression Trees) — метод деревьев классификации и регрессии схож с алгоритмом C4.5, но отличается тем, что поддерживает числовые целевые переменные и не вычисляет наборы правил. CART строит бинарные деревья, используя функцию и порог, которые дают наибольший прирост информации в каждом узле.

Библиотека машинного обучения «scikit-learn» для построения деревьев решений использует оптимизированную версию алгоритма CART.

Алгоритм CART использует один из двух критериев расщепления:

- индекс Gini;
- мера энтропии

Если целью является результат классификации, принимающий значения $0, 1, \dots, K - 1$ для узла m , будет верна следующая формула:

$$p_{mk} = \frac{1}{n_m} * \sum_{y \in Q_m} I(y = k),$$

которая является пропорцией наблюдений класса k в узле m . Если m является конечным узлом, то в качестве значения *predict_proba* выступает p_{mk} .

Индекс Gini представляется следующей формулой:

$$H(Q_m) = \sum_k p_{mk}(1 - p_{mk}).$$

Мера энтропии определяется следующей формулой:

$$H(Q_m) = - \sum_k p_{mk} \log(p_{mk}).$$

В приведенных формулах p_{mk} является вероятностью того, что узел m принимает одно из значений существующего набора из k классов.

Применение деревьев решений в поставленной задаче классификационной проблемы определения рисков заболевания пневмонией новых пациентов на основе реальных исторических данных переболевших Covid-19 обусловлено следующими преимуществами данного метода машинного обучения:

- интуитивность деревьев решений;
- отсутствие требования к определению входных атрибутов;
- высокая результирующая точность, сравнимая с точностью нейронных сетей;
- масштабируемость алгоритма, обеспечивающая линейное увеличение времени обучения при росте числа примеров;
- быстрый процесс обучения модели.

Второй раздел «Выбор инструментальных средств для разработки приложения» посвящен выбору и описанию программных средств, необходимых для построения приложения.

Для разработки приложения были использованы такие программные средства, как фреймворки «JMX», «Apache Spark», «Spring». Для применения алгоритмов машинного обучения использована библиотека «scikit learn» для языка программирования «Python».

JMX — фреймворк высокого уровня для корпоративных веб-приложений. Он поставляется с расширенными инструментами и богатым набором функциональных модулей.

Spring — мощный и гибкий фреймворк, ориентированный на создание веб-приложений с помощью языка Java. Одним из основным преимуществ данного фреймворка является то, что он заботится о большинстве низкоуровневых задачах генерации приложения для того, чтобы разработчики могли сконцентрировать свое внимание на создании основной логики программы. Еще один плюс заключается в том, что Spring уже существуют длительный период времени и активно поддерживается сообществом разработчиков, делая его актуальным и согласованным с экосистемой языка Java.

Apache Spark — фреймворк с открытым исходным кодом для реализации распределенной обработки неструктурированных и слабоструктурированных данных, входящий в экосистему продуктов Hadoop. В отличие от классиче-

ского обработчика из ядра Nadoop, реализующего двухуровневую концепцию MapReduce с хранением промежуточных данных на накопителях, Spark работает в парадигме резидентных вычислений, то есть обрабатывает данные в оперативной памяти, благодаря чему позволяет получать значительный выигрыш в скорости работы для некоторых классов задач.

Третий раздел «Практическая часть» посвящен разработке веб-приложения и построению моделей деревьев решений для анализа рисков возникновения пневмонии.

На рынке IT-технологий нет открытых и бесплатных программных средств, способных собирать исторические медицинские данные о переболевших пациентах, а затем с их помощью определять риски заболевания пневмонией у новых больных. Поэтому был создан проект сбора медицинских данных. Он является веб-ресурсом, предоставляющим возможность врачу удобно записывать, изменять и удалять необходимую информацию для прогнозирования на основании анализа крови пневмонии у больных с covid-19. Данный сервис обладает удобным пользовательским интерфейсом с возможностью настраивания фильтра поиска записей под определенные задачи.

Для функционирования приложения сбора и анализа данных пациентов необходимо создать объектную модель. Для поставленной задачи должны присутствовать такие объекты как:

- класс «Patient» — для хранения обезличенной информации о пациентах;
- класс «Diagnosis» — для хранения информации о поставленных диагнозах;
- класс «BloodTest» — для хранения информации об общем анализе крови;
- класс «CoagulationSystem» — для хранения информации о свертываемости крови;
- класс «GeneralState» — для хранения информации об общем состоянии пациента;
- класс «Nose» — для хранения информации о состоянии носовой полости пациента;
- класс «Pain» — для хранения информации о болевых ощущениях пациента;
- класс «RespiratorySystem» — для хранения информации о состоянии дыхательной системы пациента;

- класс «Skin» — для хранения информации о состоянии кожного покрова пациента;
 - класс «Pharynx» — для хранения информации о состоянии зева пациента;
 - класс «PeripheralLymphNodes» — для хранения информации о состоянии периферических лимфоузлов;
 - класс «Therapy» — для хранения информации о проведенном курсе лечения пациента;
 - класс «CardiovascularSystem» — для хранения информации о состоянии сердечно-сосудистой системы пациента;
 - класс «Risks» — для хранения анализов пациентов, для которых необходимо рассчитать возможные риски развития пневмонии;
 - класс «CalculatedRisks» — для хранения просчитанных приложением рисков развития пневмонии;
 - класс «User» — для хранения сведения о пользователях данной системы;
- Классы «Diagnosis», «BloodTest», «CoagulationSystem», «GeneralState», «Nose», «Pain», «RespiratorySystem», «Skin», «Pharynx», «PeripheralLymphNodes», «Therapy», «CardiovascularSystem» относятся к классу «Patient» как «многие-к-одному». Данная объектная модель проиллюстрирована на Рис. 1.

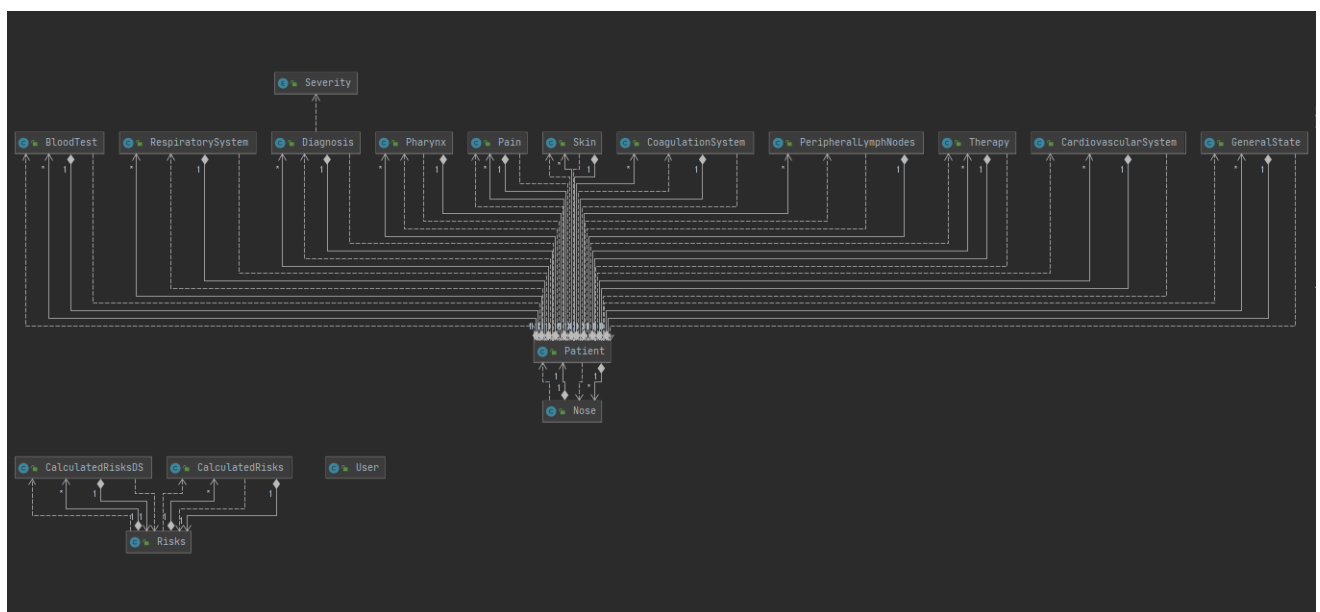


Рисунок 1 – Иллюстрация объектной модели

При нажатии кнопки «Вычислить риски» в окне сущности «Диагностика нового пациента» выполняется вызов Python скрипта, которые производит вычисление рисков возникновения пневмонии у пациентов, чьи медицинские

данные были внесены в данном окне сущности.

Для апробации работоспособности созданного программного обеспечения следует провести демонстрацию работы на примере одного нового пациента. Для данной цели внесем медицинские данные больного во вкладке «Диагностика нового пациента», как проиллюстрировано на Рис. 2. Для запуска исполнения скрипта оценки рисков пневмонии необходимо нажать на кнопку «Вычислить риски». Данная кнопка также находится на текущей вкладке.

The screenshot shows a web application interface for medical data entry. The top navigation bar includes tabs for 'Оценка риска (кровь)' and 'Диагностика нового пациен...'. The left sidebar contains a menu with items like 'Приложение', 'Анамнез', 'Диагностика нового пац...', 'Оценка риска (кровь)', 'Оценка риска (ДС)', and 'Administration'. The main area features a table with columns: Id, Возраст (лет), CP6 1, RBC, HGB, HTC, MCV, MCH, MCHC, RDW, PLT, and MPV. The table contains two rows of data. To the right of the table is a vertical list of input fields for various medical parameters, including Id, Возраст (лет), CP6 1, RBC, HGB, HTC, MCV, MCH, MCHC, RDW, PLT, MPV, WBC, NEU %, Neu abs, Lymph %, Lymph abs, mono %, Mono abs, eo %, Eo abs, baso %, and Baso abs. At the bottom left, there is a status bar showing 'GMT+04:00' and a user profile icon for 'admin'.

Id	Возраст (лет)	CP6 1	RBC	HGB	HTC	MCV	MCH	MCHC	RDW	PLT	MPV
1,501	15	0	4.15	133	33	81	32	353	10.1	177	7
1,601	1	1	1	1	1	1	1	1	1	1	1

Рисунок 2 – Иллюстрация ввода медицинских данных нового пациента

Чтобы увидеть результат выполнения программы, необходимо перейти во вкладку «Оценка риска». После исполнения процесса подсчета рисков на данной странице появится результат работы, который включает в себя информацию об «ID» анализируемого пациента и полученном результате анализа. Результат работы алгоритма для введенного пациента во вкладке «Диагностика нового пациента» представлен на Рис. 3.

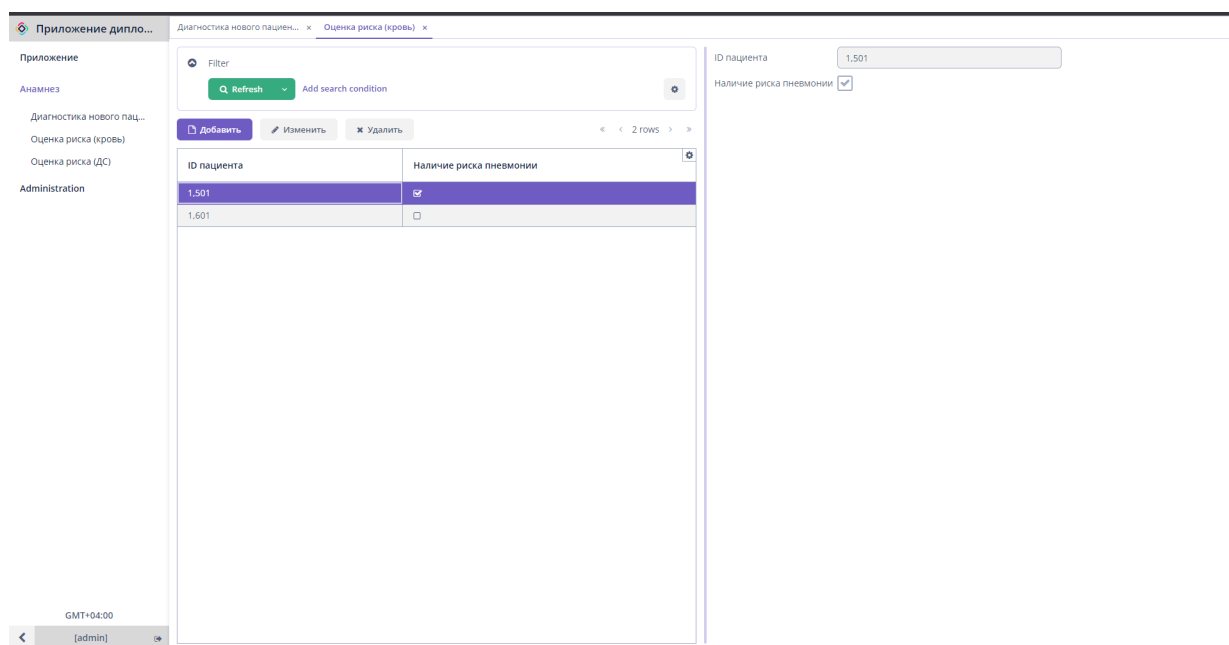


Рисунок 3 – Иллюстрация результата работы приложения

ЗАКЛЮЧЕНИЕ

В настоящее время является актуальной задачей применение методов машинного обучения в медицинской диагностике, что позволит повысить объективность процесса принятия медицинских решений. В дипломной работе была рассмотрена прикладная задача классификации на примере диагностирования больных пневмонией. Имеющаяся выборка отличалась относительно небольшим объемом при наличии 103 атрибутов. Кроме того часть данных в выборке отсутствовала.

Были проанализированы такие методы машинного обучения, как наивный байесовский классификатор, метод опорных векторов, логистическая регрессия, искусственная нейронная сеть, дерево решений. В процессе выполнения дипломной работы была выбрана и построена модель дерева решений в связи с выявленными особенностями исходной выборки. Данный метод машинного обучения оказался эффективным для применения в решении поставленной задачи классификации медицинских данных пациентов на классы наличия или отсутствия пневмонии. Был произведен подбор параметров модели на основе точности классификации, в подборе участвовали такие параметры, как глубина дерева и критерий расщепления. Были созданы две модели деревьев решений для анализа рисков пневмонии по анализу крови и состоянию дыхательной системы пациента. Оценка точности полученных моделей осуществлялась кросс-валидационными методами. Веб-приложение разработано на фреймворке JMX, который эффективно объединяет средства разработки как бэкенд, так и фронтенд части. В качестве СУБД использовалась PostgreSQL. Анализ данных производился при помощи Apache Spark и библиотеки scikit-learn для Python. Функциональные возможности данного приложения включают: регистрацию нового пользователя, добавление пациента, описание анамнеза, внесение данных больного для диагностики пневмонии, оценка риска пневмонии. Для демонстрации работоспособности приложения была произведена апробация действующего функционала на примере нового пациента.

Дальнейшее улучшение функциональности и эффективности приложения может заключаться в увеличении выборки медицинских данных пациентов, уже проанализированных на наличие или отсутствие пневмонии. Также по мере увеличения выборки данных существует возможность включать новые атрибуты в процесс построения моделей деревьев решений.

Основные источники информации

1. *Вьюгин, В.* Математические основы теории машинного обучения и прогнозирования / В. Вьюгин — Москва: МЦНМО, 2013. — Яз. рус.
2. *Barber, D.* Bayesian Reasoning and Machine Learning / D. Barber — Cambridge: Cambridge University Press, 2012. — Яз. англ.
3. *Флах, П.* Машинное обучение. Наука и искусство построения алгоритмов, которые извлекают знания из данных / П. Флах — Москва: ДМК-Пресс, 2016. — Яз. рус.
4. *Саттон, Р.* Обучение с подкреплением / Р. Саттон — Москва: Бином. Лаборатория знаний, 2017. — Яз. рус.
5. *Nilsson, N.* Introduction to Machine Learning / N. Nilsson — Stanford: Stanford University Press, 2015. — Яз. англ.
6. *Rashid, T.* Make Your Own Neural Network / T. Rashid — South Carolina: CreateSpace Independent Publishing Platform, 2016. — Яз. англ.
7. *Hastie, T.* The Elements of Statistical Learning / T. Hastie — Stanford: Springer, 2009. — Яз. англ.
8. *Бастиян, Ш.* Крупномасштабное обучение вместе с Python / Ш. Бастиян — Москва: ДМК-Пресс, 2017. — Яз. рус.
9. *Russel, S.* Artificial Intelligence: A Modern Approach / S. Russel — London: Pearson, 2013. — Яз. англ.