|  |  |
| --- | --- |
|  | **МИНОБРНАУКИ РОССИИ**  **федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Самарский государственный технический университет»**  **(ФГБОУ ВО «СамГТУ»)** |

Институт автоматики и информационных технологий

Кафедра «Информатика и вычислительная техника»

**Выпускная квалификационная работа**

*(бакалаврская работа)*

на тему:

Разработка модуля по первичному диагнозу заболеваний пациентов

*(полное наименование темы в соответствии с приказом об утверждении тем ВКР)*

Выполнил: *\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*Лазанчин*\_*Петр*\_*Михайлович*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*,  
 *(фамилия, имя, отчество, подпись)*

обучающийся по *направлению подготовки / специальности \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_09.03.03 Прикладная Информатика \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

*(код и наименование направления подготовки / специальности)*

направленности (профилю) *Прикладная информатика в экономике \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

*(наименование направленности (профиля))*

*\_\_*4*\_\_* курса, группы *\_\_\_*107*\_\_\_\_*, *\_\_\_\_\_*очной*\_\_\_\_\_\_* формы обучения.

*(очной, очно-заочной, заочной)*

Руководитель: доцент, к.т.н. Верещагина С.С.

*(должность, ученая степень, звание, подпись, дата, фамилия, инициалы)*

Консультант (экономика) доцент, к.э.н., доцент Малина А.Б.

*(должность, ученая степень, звание, подпись, дата, фамилия, инициалы)*

Нормоконтролер доцент, к.т.н. Верещагина С.С.

*(подпись, дата, фамилия, инициалы)*

Заведующий кафедрой к.х.н., доцент*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*Чуваков А.В.

*(должность, ученая степень, звание, подпись, дата, фамилия, инициалы)*

Самара 2025 г.

**РЕФЕРАТ**

Пояснительная записка содержит 59 страниц, 16 рисунков, 10 таблиц,   
9 источников.

МЕДИЦИНА, ДИАГНОЗ, СИМПТОМ, БОЛЕЗНЬ, КЛАССИФИКАЦИЯ, СЛУЧАЙНЫЙ ЛЕС

Объектом исследования является процесс поддержки принятия врачебных решений в области диагностики заболеваний.

Предметом работы является возможность использования обученной нейросети на основе алгоритма случайного леса для постановки предварительных диагнозов по введённым симптомам.

Целью выпускной квалификационной работы является разработка программного модуля, предназначенного для помощи терапевтам в постановке диагнозов путём анализа симптомов с использованием методов машинного обучения.

В результате проведённого анализа предметной области были выделены основные сущности и их атрибуты и установлены связи между ними. Итогом анализа стала разработка инфологической модели типа «сущность – связь».

Результатом выпускной квалификационной работы стало создание программного модуля, включающего экспертную систему для пошагового выбора симптомов пользователем и обученную нейросетевую модель на основе случайного леса, предсказывающую наиболее вероятные диагнозы.

Исходя из рассчитанных показателей экономической эффективности, сделан вывод о том, что проект по разработке программного модуля для поддержки врачей в постановке диагнозов является целесообразным и перспективным для внедрения в клиническую практику.

Технология разработки — язык программирования *Python*, библиотеки *Scikit*-*learn*, *Pandas* и собственная экспертная система для взаимодействия с пользователем в консольном режиме.

СОДЕРЖАНИЕ

[ВВЕДЕНИЕ 3](#_Toc200135878)

[1. Анализ предметной области 5](#_Toc200135879)

[1.1 Роль нейросетей и экспертных систем в современной медицине 5](#_Toc200135880)

[1.2 Возможности алгоритмов случайного леса и деревьев решений в диагностике 7](#_Toc200135881)

[1.3 Проблемы первичного опроса пациента и роль цифровых помощников 11](#_Toc200135882)

[2. Проектирование архитектуры программного модуля 14](#_Toc200135883)

[2.1 Обоснование выбора алгоритма машинного обучения 14](#_Toc200135884)

[2.2 Схема системы (модули связи). Блок схема алгоритма. 20](#_Toc200135885)

[3. Программная реализация модели классификации заболеваний 30](#_Toc200135886)

[3.1 Выбор языка программирования и используемых библиотек 30](#_Toc200135887)

[3.2 Описание и демонстрация работы программы 31](#_Toc200135888)

[3.3 Оценка качества модели 38](#_Toc200135889)

[4. Экономическое обоснование разработки программного модуля 42](#_Toc200135890)

[4.1 Анализ рынков 42](#_Toc200135891)

[4.2 Формирование затрат и доходов проекта 46](#_Toc200135892)

[4.3 Экономико-математическая модель программного модуля 49](#_Toc200135893)

[4.4 Расчет показателей экономической эффективности 54](#_Toc200135894)

[СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ 60](#_Toc200135895)

## ВВЕДЕНИЕ

В настоящее время, в условиях стремительного развития информационных технологий, сфера здравоохранения переживает значительные преобразования. Современная медицина всё чаще опирается на интеллектуальные программные решения, направленные на повышение точности диагностики, снижение нагрузки на врачей и минимизацию человеческого фактора. Особое внимание уделяется применению систем искусственного интеллекта и машинного обучения, способных обрабатывать большие объемы медицинских данных и делать предварительные выводы на основе выявленных закономерностей. В этом контексте особенно актуальной становится разработка программных инструментов, способных помогать врачам общей практики и терапевтам в постановке предварительных диагнозов на основе введённых симптомов.

Предметная область данной работы сосредоточена на создании интеллектуальной системы, объединяющей экспертный подход к выбору симптомов с возможностями обученной нейросети на основе алгоритма случайного леса. Такая система позволяет шаг за шагом собрать информацию от пользователя о его жалобах, структурировать данные и передать их в обученную модель, которая формирует список наиболее вероятных диагнозов с указанием степени уверенности.

Традиционные методы диагностики, основанные исключительно на опыте врача, подвержены влиянию человеческого фактора, ограничены временем консультации и не всегда позволяют учесть все возможные варианты заболеваний, особенно на ранних стадиях. Использование экспертных и нейросетевых систем позволяет повысить объективность и точность диагностического процесса, служит инструментом поддержки принятия решений, не заменяя врача, но дополняя его работу.

Выбор данной темы обусловлен актуальной потребностью системы здравоохранения в современных средствах поддержки диагностики, особенно в условиях нехватки специалистов и высокого потока пациентов. Внедрение таких решений может не только повысить эффективность работы медицинских учреждений, но и улучшить качество медицинской помощи за счёт более точного и быстрого выявления заболеваний.

Разработка программного модуля, сочетающего экспертную систему для сбора данных и нейросетевую модель для анализа, открывает новые перспективы в применении искусственного интеллекта в медицине. Это позволяет не только автоматизировать рутинные этапы диагностики, но и способствует формированию более обоснованных и прозрачных медицинских решений. В долгосрочной перспективе подобные технологии могут стать частью единой цифровой медицинской инфраструктуры, обеспечивая устойчивое развитие и цифровую трансформацию отрасли здравоохранения.

Целью данной работы является разработка программного модуля для составления психологического портрета кандидата на трудоустройство, который позволит более точно и объективно оценивать профессиональные и личностные качества соискателей.

Для достижения поставленной цели необходимо решить следующие задачи:

1) изучить существующие методы и технологии в области подбора персонала;

2) провести анализ существующих решений в области тестирования персонала и составления психологических портретов;

3) спроектировать архитектуру программного модуля, учитывающую специфику процесса подбора персонала;

4) разработать программный модуль согласно спроектированной архитектуре;

5) сделать выводы о работе программного модуля.

Данная работа призвана внести вклад в совершенствование методов поддержки врачебной диагностики и повышение эффективности лечебного процесса. Используя разработанный программный модуль, медицинские учреждения могут получить значительное преимущество в организации амбулаторного приёма, так как система помогает быстрее и точнее определять возможные диагнозы. Это способствует снижению диагностических ошибок, повышению качества оказания медицинской помощи и, в перспективе, снижению затрат на лечение за счёт более раннего выявления заболеваний и рационального использования ресурсов.

## 1. Анализ предметной области

### 1.1 Роль нейросетей и экспертных систем в современной медицине

Интенсивное развитие медицинской науки, расширение возможностей углубления в этиологию, патогенез заболевания, увеличение данных о маркерах различных патологических состояний диктует необходимость поиска новых подходов к обработке полученных результатов. На современном этапе важно провести быстрый анализ большого числа данных и принять верное решение, которое может повлиять на прогнозы, течение и исходы заболевания. В связи с этим все более пристальное внимание уделяется информационным технологиям (ИТ), а в рамках медицины можно говорить об "электронной медицине". ИТ реализуются в виде медицинских информационных систем (МИС) различного назначения и отдельных автоматизированных лечебно-диагностических приборов, среди которых - современные экспертные системы (ЭС). ЭС представляют собой компьютерные программы, производящие анализ на основе определенных исходных данных, предназначенные для оказания помощи специалистам в конкретных областях знаний и достижения значимых результатов.

Применение ЭС позволяет решить различные задачи, к которым относится прогнозирование рисков развития заболеваний, осложнений и эффективности лечения, ранняя диагностика, планирование лечения, мониторинг состояния здоровья пациента, автоматизированный анализ и статистическая обработка клинического материала. Экспертные МИС значимо упрощают работу в таких ситуациях, когда невозможно представить задачу в числовой форме, отсутствует определенность или точность в изучаемых параметрах или нет однозначного алгоритма решения задач. Данные характеристики подходят для решения и медицинских задач, представляющих собой большой объем многомерных, сложных и порой противоречивых клинических данных, полученных в процессе цензурированных наблюдений.

В настоящее время в медицинских исследованиях превалирует использование статистических методов обработки данных. Самыми распространенными описательными методами, использующимися в традиционных статистических исследованиях, являются анализ выживаемости и многомерный комплексный анализ, классифицирующийся на дискриминантный, кластерный, факторный и корреляционный.

На мировом рынке медицинских компьютерных продуктов существует более 250 ЭС, введенных в эксплуатацию, при этом активно использующихся в различных учреждениях здравоохранения, приобретение которых возможно, как в специальных каталогах, так и в сети. Примером данных технологий может послужить *PUFF*-система - кодированное на языке *BASIC* интеллектуальное расширение аппарата исследования лёгких, предназначенная для диагностики заболеваний. Данные с измерительных приборов, оценивающих функцию легких, непосредственно после поступления на персональный компьютер врача обрабатываются и выводятся в форме конкретного диагноза и перечня рекомендаций для дальнейшего лечения. В настоящее время в медицине наиболее эффективно использование таких ЭС, как методы добычи данных и искусственные нейронные сети, позволяющие решать задачи диагностики и прогнозирования различных заболеваний, выбирать тактику лечения и профилактики и др.

Примерами могут служить:

*DXplain*, система поддержки принятия решений, разработанная в Лаборатории компьютерных наук Массачусетской больницы общего профиля, сочетает в себе функции электронного медицинского справочника и медицинской справочной системы *[*1*]*.

Учебная программа *PIER* была обновлена и оптимизирована, чтобы упростить внедрение в программы ординатуры по патологии. Основные улучшения включают в себя: более краткое руководство по началу работы с программами, обновлённые темы и все новые знания и навыки, которые лучше соответствуют ожиданиям от резидентов по завершении обучения, новый инструмент, заменяющий три предыдущих инструмента, который позволяет легче отслеживать прогресс резидентов и фиксировать завершение обучения, разделение набора ресурсов (85 страниц) на 4 отдельных документа меньшего размера, по одному на каждую из основных тем, реструктуризацию наборов ресурсов для упрощения использования и сокращение количества рекомендуемых ресурсов по каждой теме, чтобы упростить выбор учебных материалов *[*2*]*.

*Isabel* — единственный инструмент, который может преобразовать признаки и симптомы заболевания пациента в список соответствующих заболеваний или рекомендации по сортировке без бесконечных вопросов. Сегодня инструменты *Isabel* используются более чем в 90 странах и доступны на 10 языках *[*3*]*/

Мобильное приложение на основе ИИ, позволяющее пользователю ввести симптомы и получить список возможных причин. *Ada* используется как пациентами, так и врачами. В отличие от других систем, делает упор на пользовательский опыт и адаптивную диагностику. Прошла клинические испытания и получила положительные оценки по точности *[*4*]*.

### 1.2 Возможности алгоритмов случайного леса и деревьев решений в диагностике

Для сбора данных использовались анкеты, медицинский осмотр и лабораторные анализы. Анкета содержала демографические данные (включая возраст, пол, годовой доход, уровень образования), информацию об образе жизни (включая курение, употребление алкоголя, диету и физические нагрузки). Анкета заполнялась онлайн самими участниками или членами их семей. Медицинский осмотр включал измерение роста, веса и артериального давления (систолического и диастолического), которые измерялись дважды, после чего рассчитывалось среднее значение. Все данные измерялись медицинским работником. Индекс массы тела (ИМТ) рассчитывается как отношение веса в килограммах к квадрату роста в метрах.

У испытуемых брали венозную кровь натощак для определения уровня глюкозы в крови натощак (*FPG*), гликированного гемоглобина (*GHb*), гомоцистеина (*Hcy*), общего холестерина (*TC*), триглицеридов (*TG*), холестерина липопротеинов низкой плотности (*LDL*-*C*) и липопротеинов высокой плотности (*HDL*-*C*). У испытуемых брали утреннюю мочу. После центрифугирования при 3000 об/мин в течение 10 минут был получен супернатант (низкоскоростная центрифуга *Anhui* *Zhongke* *Zhongjia* *SC*3616), и с помощью латексной турбидиметрии, саркозиновой оксидазы и иммунотурбидиметрии были определены α1-микроглобулин (α1МГ), креатинин в моче (*UCr*) и микроальбуминурия (МАУ).

Присвоения переменных

Информация о годовом доходе, уровне образования, состоянии здоровья и образе жизни участников исследования была получена из опросника. Годовой доход определялся как <5 тыс. юаней, 5-10 тыс. юаней, 10-20 тыс. юаней, > 20 тыс. юаней; уровни образования определялись как ≤ начальная школа, ≤ средняя школа, ≤ старшая школа, ≥ степень бакалавра; курение классифицировалось как "да" или “нет"; потребление алкоголя классифицировалось как "всегда" (> 100 г / раз и 3 раза в неделю), иногда (<3 раза в неделю или <100 г / раз) и редко; физические упражнения классифицировались как "совсем или немного” или “регулярно” (≥ 3 раза в неделю, ≥ 30 мин / раз). ИМТ определяли как недостаточный вес (<18,5 кг/м2), нормальный вес (18,5–24,0 кг/м2), избыточный вес (24,0–28,0 кг/м2), ожирение (≥28 кг/м2). *ACR* определяли как микроальбумин в моче, деленный на креатинин в моче, умноженный на 8,84; *MCR* определяли как микроглобулин в моче, деленный на креатинин в моче, умноженный на 8,84.

Объясняющие переменные

Анкета: демографические данные (возраст, пол, уровень образования, годовой доход, место жительства и т. д.); образ жизни (курение, употребление алкоголя, физические нагрузки, потребление соли, диета). Анализ крови: ЛПВП, ЛПНП, ТГ, *TC*, *Hcy*, *FPG*, *GHb*. Физикальное обследование: САД, ДАД, ИМТ. Всего 20 показателей.

Переменные ответа

*ACR* ≥ 30 мг/г определялся как *GI*; *MCR* > 23 мг/г определялся как *TI*. При наличии *GI*, *TI* присваивалось значение 1; в противном случае присваивалось значение 0. В этом исследовании мы использовали *RF*, *LR* и *NB* для классификации *GI* и *TI* соответственно.

Регуляризация *L*1, абсолютная усадка и регрессия оператора выбора

Оператор абсолютного сжатия и выбора (*LASSO*) — один из распространённых методов выбора признаков. Он характеризуется включением в обобщённую линейную регрессию штрафного коэффициента регуляризации *L*1, который делает сумму абсолютных значений коэффициентов регрессии модели меньше определённого значения. Он направлен на минимизацию суммы квадратов остатков, заставляя коэффициенты регрессии переменных, которые вносят меньший вклад в модель, сжиматься до нуля и достигая разреженного процесса. *LASSO* использовался для выбора собранных объясняющих переменных и определения наиболее значимых для зависимых переменных.

Алгоритм метода синтетической неполной избыточной выборки

Метод синтетической передискретизации меньшинства (*SMOTE*) — это метод передискретизации, который является эффективным алгоритмом для устранения дисбаланса между классами данных. Он используется для синтетического увеличения класса меньшинства с помощью *K*-ближайших соседей для получения сбалансированного набора данных и показал хорошие результаты в таких областях, как системы обнаружения сетевых вторжений и выявление заболеваний. В этом исследовании наблюдается серьёзный дисбаланс в переменных отклика, *GI* и *TI*. *SMOTE* использовался для балансировки классов, чтобы помочь моделям машинного обучения лучше изучить взаимосвязи между данными и сделать наиболее точные выводы о классификации.

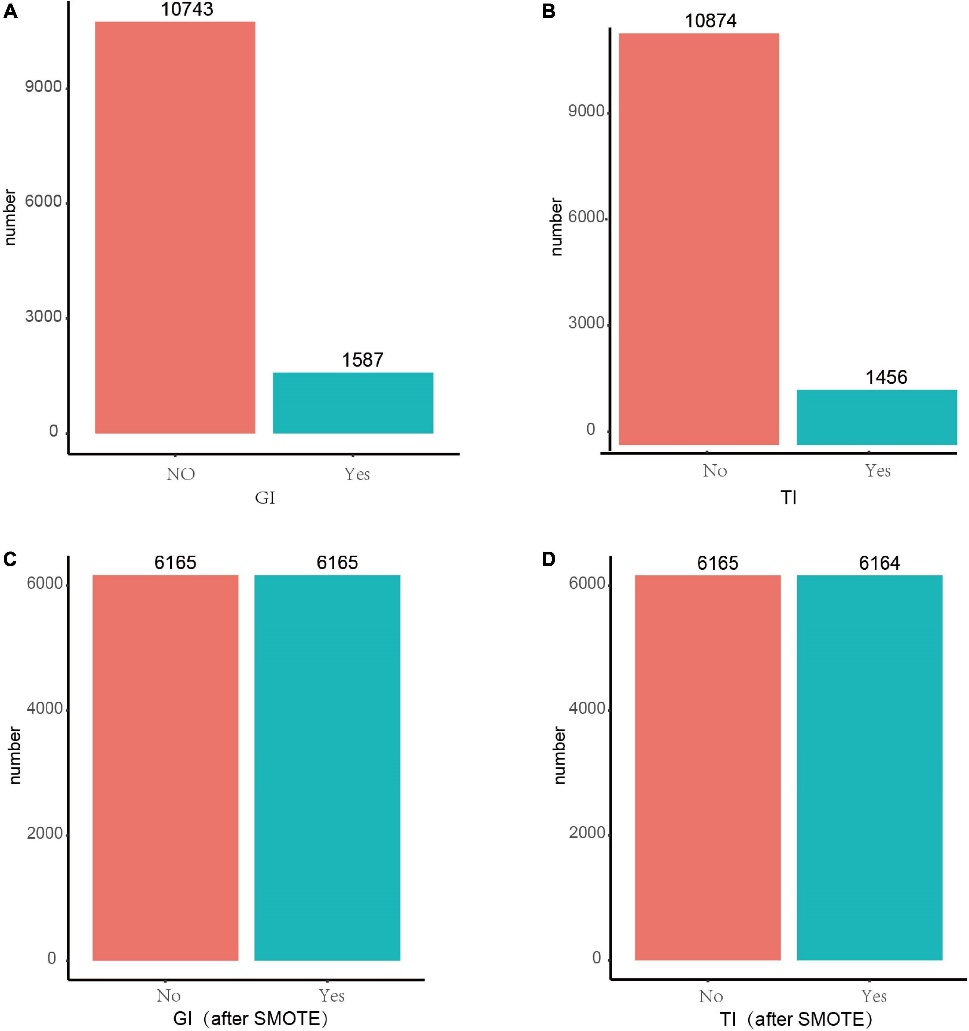


Рисунок 1. До и после *SMOTE* для переменных отклика *GI* и *TI*.

*SMOTE* — метод синтетической передискретизации меньшинства. Это хороший и эффективный способ работы с несбалансированными данными, который был применён с параметрами *k* = 5, *C*.*perc* = «баланс», *dist* = «перекрытие». (*A*) *GI* до *SMOTE*; (*B*) *TI* до *SMOTE*; (*C*) *GI* после *SMOTE*; (*D*) *TI* после *SMOTE*.

Случайный лес

*RF*, алгоритм интегрированного обучения на основе данных, может получать множество новых обучающих данных путём автономной выборки из обучающего набора, создавая несколько деревьев классификации на основе параллельной обработки этих новых данных и достигая де-корреляции между деревьями путём выбора независимых переменных. Разнообразие деревьев классификации возникает как из-за выборки, так и из-за отклонений независимых переменных, что позволяет снизить дисперсию модели и, наконец, проголосовать за результаты классификации нескольких деревьев для получения окончательных результатов. Рабочий процесс построения модели показан на рисунке 2.

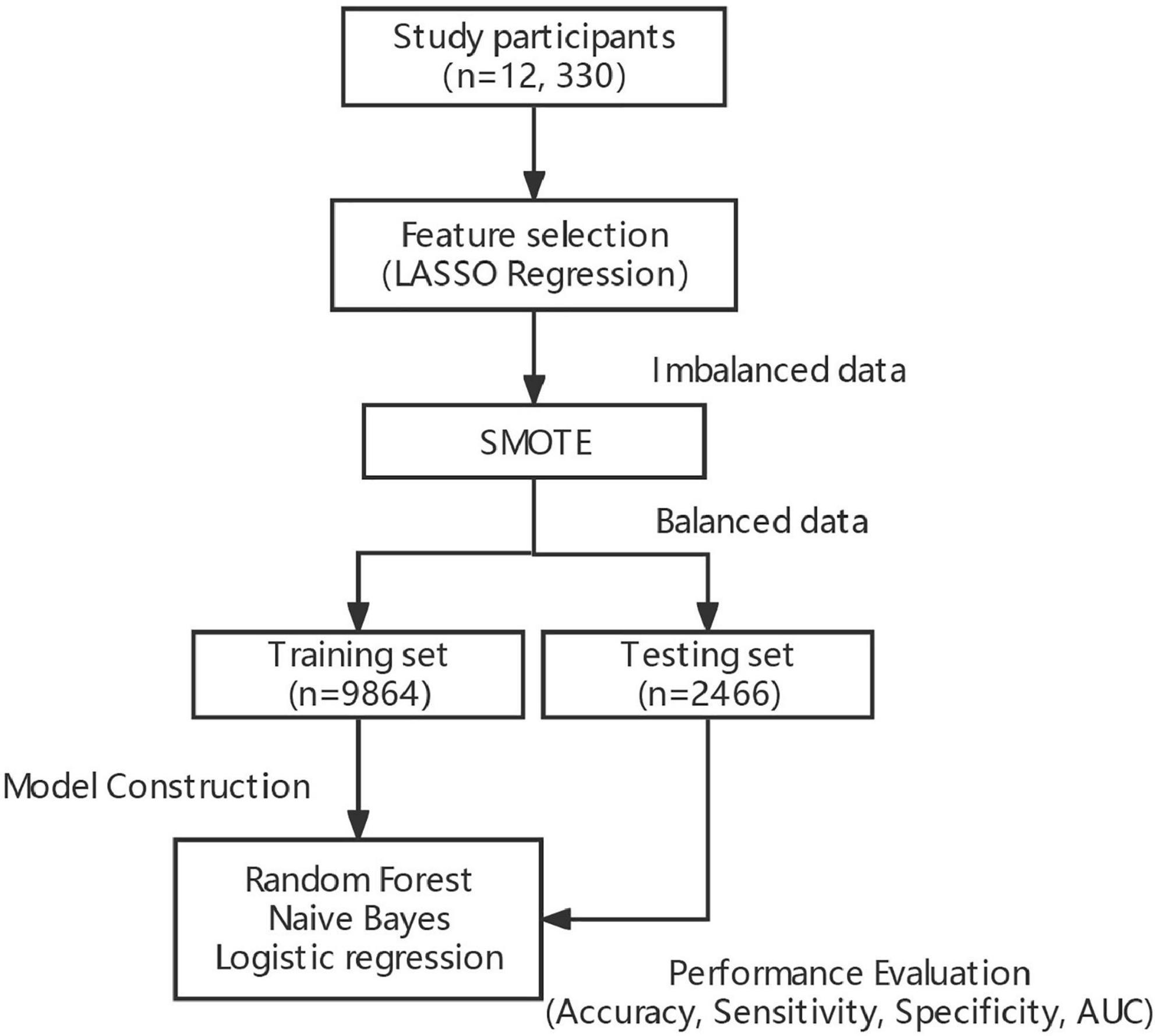


Рисунок 2. Процесс создания модели.

Шаг 1. Собираются исходные данные исследования, включающие 12 330 участников. Данные могут включать медицинские показатели, демографические характеристики или другие признаки, необходимые для анализа.

Шаг 2. Для отбора наиболее значимых признаков используется *LASSO*-регрессия (*L*1-регуляризация), которая автоматически исключает менее важные переменные, обнуляя их коэффициенты.

Шаг 3. Если данные несбалансированы (например, один класс значительно преобладает), применяется *SMOTE* (*Synthetic* *Minority* *Over*-*sampling* *Technique*), который создает синтетические примеры для миноритарного класса.

Шаг 4. Данные делятся на обучающую и тестовую выборки в соотношении ~80/20.

Шаг 5. Строятся три модели машинного обучения: *Random* *Forest* (ансамбль деревьев с бутстрэп-агрегированием), *Naive* *Bayes* (вероятностная модель, основанная на теореме Байеса), *Logistic* *Regression* (линейная модель для классификации)

Шаг 6. Оценивается качество моделей с помощью метрик: *Accuracy* (общая точность), *Sensitivity* (*Recall*) (способность обнаруживать положительные классы), *Specificity* (способность обнаруживать отрицательные классы), *AUC*-*ROC* (площадь под *ROC*-кривой, оценивает качество ранжирования).

Схема описывает полный цикл *ML*-исследования: от подготовки данных и балансировки классов до обучения моделей и их оценки. *Random* *Forest* выделяется как наиболее устойчивый метод, особенно после *SMOTE*, благодаря ансамблевому подходу и снижению переобучения.

### 1.3 Проблемы первичного опроса пациента и роль цифровых помощников

Точность метода опроса зависит как от квалификации врача, так и от его способности расположить пациента к сотрудничеству, правильно распознав психологические черты больного человека. Врачебный опрос как метод обследования больного представляет собой разной степени структурированности набор вопросов, который очень сложно унифицировать: каждый врач его проводит по-разному (нестандартизированный опрос). Традиционный опрос, при котором врач находится в непосредственном контакте с пациентом, является неструктурированным или частично структурированным. При сборе жалоб, анамнеза пациента одну и ту же мысль, один и тот же симптом можно описать разными словами, которые порой сложно интерпретировать, поэтому возникает необходимость создания программ, которые способны извлекать информацию из неструктурированных текстов с целью повышения эффективности клинического опроса. Опросники (анкеты) являются структурированными источниками информации. Анкетирование (от франц. *enquete* – «расследование») является одной из методик опроса, при которой имеет место четко намеченный план его проведения.

С одной стороны, подробный детализированный опрос является недорогим и эффективным методом диагностики, но, с другой – требует значительных временных затрат. Совершенствование метода опроса определяет необходимым появление дополнительных интерактивных технологий, которые позволили бы сохранить его достоинство – возможность изучить подробно весь спектр возможных проблем со здоровьем пациента, но при этом сократить время его проведения, устраняя тем самым существенный недостаток данного метода.

Уже два десятилетия назад широко признавалось, что информационные технологии повлияют на здравоохранение в той же мере, что и на другие важные отрасли. Стремительное развитие цифровизации здравоохранения происходит в соответствии с нормативно-правовой базой (приказ Минздрава РФ от 30 ноября 2017 г. № 965н, внесенные 29 июля 2017 г. (242-ФЗ) изменения в ФЗ от 21 ноября 2011 г. № 323) *[*5*]*. Медицина с использованием электронных средств на сегодняшний день применима лишь в рамках мониторинга состояния пациента в динамике, автоматического регулирования лечебных, реабилитационных программ, но не применима для первичной диагностики, так как хотя и является современным эквивалентом обследования пациента врачом, но без использования физических методов многие традиционные диагностические критерии при этом утеряны. Спектр клинического применения электронных ресурсов становится с каждым днем все шире. Использование компьютерных программ, цифровизация и автоматизация диагностического процесса открывают дополнительные возможности для управления медицинскими данными в клинической практике.

В основу программы ЭВМ «Электронная поликлиника», внедренной в ЕИСЗ на территории Пермского края *[*6*]*, заложен проблемный принцип, который позволяет сгруппировать достаточно большое общее количество вопросов с детализацией в клинические проблемы, а интерактивный режим анкетирования дает возможность выбрать из них наиболее существенные для пациента и, наоборот, пропустить неактуальные.

Идеального метода диагностики не существует, ошибки могут случаться при использовании любого метода. Рассмотрение, взвешивание всех преимуществ и недостатков каждого и постоянное сравнение нескольких методов позволяют определить точность диагностического процесса.

Вывод по анализу предметной области

В ходе анализа предметной области выявлена высокая актуальность внедрения нейросетей и экспертных систем в сферу здравоохранения. Современные тенденции в медицине требуют оперативной обработки больших объёмов клинических данных и принятия обоснованных диагностических решений, что делает применение интеллектуальных технологий особенно значимым. Экспертные системы и алгоритмы машинного обучения, включая деревья решений и случайный лес, демонстрируют высокую эффективность в задачах ранней диагностики, прогнозирования заболеваний и оптимизации лечебной тактики. Их использование позволяет учитывать широкий спектр параметров — от демографических и поведенческих до биохимических и лабораторных показателей.

Проанализированы современные методы сбора и структурирования медицинской информации, включая цифровые анкеты и алгоритмы предварительного опроса пациентов. Показано, что традиционный метод врачебного опроса требует значительных временных и когнитивных ресурсов, в то время как автоматизация на основе цифровых ассистентов способна существенно повысить эффективность первичной диагностики. Внедрение таких решений, как интерактивные опросники и мобильные приложения на базе ИИ, расширяет возможности медицины в части повышения точности и оперативности клинических решений.

Совокупность рассмотренных технологий свидетельствует о высокой перспективности интеграции интеллектуальных систем в клиническую практику с целью повышения качества и доступности медицинской помощи.

## 2. Проектирование архитектуры программного модуля

### 2.1 Обоснование выбора алгоритма машинного обучения

В данной работе решается задача классификации с множеством классов: на основе входных данных, представленных в виде списка симптомов (от 1 до 7, выбираемых последовательно), нужно определить наиболее вероятный диагноз из нескольких категорий. Особый акцент делается на способности алгоритма функционировать при наличии неполных данных и на прозрачности результатов, то есть на возможности разобраться, какие именно факторы повлияли на постановку диагноза.

Возможные подходы к решению задачи:

1. Деревья решений

Деревья решений (*Decision* *Trees*, *DT*) – один из инструментов интеллектуального анализа данных и предсказательной аналитики, которые позволяют решать задачи классификации.

Таблица 1 – Комплексная характеристика алгоритма деревьев решений

|  |  |  |  |
| --- | --- | --- | --- |
| **Категория** | **Характеристика** | | **Описание** |
| Преимущества | Интерпретируемость | | Упрощает валидацию модели экспертами-медиками |
| Устойчивость к выбросам | | Быстрое обучение при ограниченных ресурсах |
| Вычислительная эффективность | | Применимость в системах реального времени |
| Гибкость обработки данных | | Поддержка числовых и категориальных признаков без преобразований |
| Нелинейные зависимости | | Автоматическое выявление сложных взаимосвязей между признаками |
| Недостатки | Риск переобучения | Построение сложных деревьев на шумных данных | |
| Нестабильность модели | Высокая чувствительность к изменениям в данных | |
| Чувствительность к дисбалансу | Смещение в сторону доминирующих классов | |
| Ограничения масштабируемости | Снижение производительности на больших массивах данных | |
| Дискретность прогнозов | Ступенчатая аппроксимация непрерывных переменных | |

Деревья решений имеют ряд ограничений, которые делают их менее подходящими для поставленной задачи. Они склонны к переобучению, из-за чего алгоритм может идеально работать на обучающих данных, но плохо обобщать на новых случаях, что критично в медицине, и их нестабильность приводит к тому, что небольшие изменения в данных могут существенно менять структуру дерева, снижая надёжность прогнозов.

1. *XGBoost*

*XGBoost* (*eXtreme* *Gradient* *Boosting*) – алгоритм машинного обучения, который основан на градиентном бустинге деревьев решений. Он сочетает в себе преимущества ансамблевых методов с оптимизацией для скорости и точности.

Таблица 2 – Комплексная характеристика алгоритма *XGBoost*

|  |  |  |
| --- | --- | --- |
| **Категория** | **Характеристика** | **Описание** |
| Преимущества | Высокая точность | Один из самых точных алгоритмов, часто превосходящий другие методы на сложных данных |
| Устойчивость к выбросам | Хорошо справляется с выбросами и пропущенными значениями |
|  | Поддержка регуляризации | Использует регуляризацию (*L*1, *L*2) для снижения риска переобучения |
| Недостатки | Сложность настройки | Требует внимательного подбора гиперпараметров (*learning* *rate*, *max* *depth* и др.) |
| Высокие требования к ресурсам | Требует значительных вычислительных ресурсов, особенно при работе с большими наборами данных |
| Ограниченная интерпретируемость | Интерпретируемость хуже, чем у *Random* *Forest*, из-за сложности ансамбля деревьев |

*XGBoost* не подходит для небольших и средних по масштабу проектов из-за избыточной мощности, которая требует значительных вычислительных ресурсов и тонкой настройки гиперпараметров.

1. Метод опорных векторов

Метод опорных векторов (*SVM*, *Support* *Vector* *Machines*) основан на поиске оптимальной разделяющей гиперплоскости в пространстве признаков.

Таблица 3 – Комплексная характеристика *SVM*

|  |  |  |
| --- | --- | --- |
| **Категория** | **Характеристика** | **Описание** |
| Преимущества | Высокая точность | Показывает высокую точность, благодаря нахождению оптимальной разделяющей гиперплоскости с максимальным зазором между классами |
| Эффективность в многомерных пространствах | Достойно работает с данными большой размерности, даже если число признаков превышает число выборок |
| Устойчивость к переобучению | Благодаря регуляризационному параметру и фокусу на максимальном зазоре снижает риск переобучения |
| Недостатки | Чувствительность к шуму и выбросам | Выбросы становятся опорными векторами, которые искажают гиперплоскость и снижает точность классификации |
| Сложность выбора ядра и параметров | Подбор подходящего ядра и гиперпараметров (например, *C*) требует опыта и экспериментов |
| Ограниченная интерпретируемость | В отличие от линейной регрессии, *SVM* с нелинейными ядрами сложнее интерпретировать |

Метод опорных векторов чувствителен к шуму и выбросам, требует тщательного подбора ядер и настройки гиперпараметров, что усложняет его применение.

1. Метод *k*-ближайших соседей

Метод *k*-ближайших соседей (*KNN*, *k*-*Nearest* *Neighbors*) основан на принципе схожести объектов: новый объект относится к тому классу, к которому принадлежит большинство из его *k* ближайших соседей в пространстве признаков.

Таблица 4 – Комплексная характеристика *KNN*

|  |  |  |
| --- | --- | --- |
| **Категория** | **Характеристика** | **Описание** |
| Преимущества | Простота реализации и интерпретации | Легко понять и реализовать, не требует сложных математических вычислений или построения модели |
| Отсутствие фазы обучения | Не требует предварительного обучения модели, данные просто сохраняются для использования при предсказании |
| Непараметрический подход | Не делает предположений о распределении данных. Позволяет применять его к широкому спектру задач |
| Недостатки | Чувствительность к выбору | Неправильный выбор значения *K* приводит к низкой производительности модели, а оптимальное значение часто требует экспериментов |
| Проблемы с несбалансированными данными | Есть риски смещения в сторону доминирующего класса, так как объекты этого класса чаще попадают в число ближайших соседей |
| Чувствительность к шуму и выбросам | Шумные данные или выбросы могут существенно искажать результаты классификации |

Метод *K*-ближайших соседей при работе с объемными данными вследствие двух ключевых факторов: значительных вычислительных затрат, обусловленных необходимостью расчета расстояний до всех объектов обучающей выборки, повышенных требований к объему памяти, поскольку алгоритм требует хранения полного набора исходных данных.

1. Нейронные сети

Нейросети имитируют работу человеческого мозга для решения задач классификации, регрессии, обработки изображений и текста. Фреймворки, такие как *TensorFlow* (разработанный *Google*) и *PyTorch* (разработанный *Facebook*), являются лидерами в разработке и обучении нейронных сетей, предоставляя разработчикам и исследователям гибкие и эффективные инструменты для создания моделей.

Таблица 5 - Комплексная характеристика нейросетей

|  |  |  |
| --- | --- | --- |
| **Категория** | **Характеристика** | **Описание** |
| Преимущества | Высокая точность | Способны выявлять сложные закономерности в больших объёмах данных, что делает их идеальными для задач вроде распознавания изображений или обработки естественного языка. |
| Гибкость и адаптивность | Адаптируются к широкому спектру задач благодаря различным архитектурам и фреймворкам |
| Автоматизация и обучение | Способны обучаться на данных, автоматически извлекая признаки |
| Недостатки | Риск переобучения и нехватка данных | Могут переобучаться на малых наборах данных или требовать огромных объёмов данных для достижения высокой точности |
| Сложность интерпретации | Часто работают как "чёрный ящик", что затрудняет объяснение их решений, особенно в критически важных областях, таких как медицина. |
| Высокие вычислительные требования | Требуют значительных ресурсов (*GPU*/*TPU*) и времени для обучения, особенно на больших наборах данных |

Нейросети, несмотря на их мощь, не подходят для задач с ограниченными вычислительными ресурсами или малыми наборами данных, так как требуют значительных затрат на обучение и могут переобучаться, теряя обобщающую способность.

1. Случайный лес

Случайный лес (*Random* *Forest*) - алгоритм машинного обучения, основанный на ансамбле решающих деревьев. Он сочетает в себе идеи бэггинга (*bootstrap* *aggregating*) и случайного выбора подмножеств признаков для построения множества независимых деревьев решений, что позволяет достичь высокой точности модели. *RF* применяется в задачах классификации и регрессии, включая медицинскую диагностику, благодаря своей способности эффективно обрабатывать сложные данные с множеством признаков. В контексте анализа симптомов для диагностики заболеваний этот алгоритм выделяется своей универсальностью и практичностью, что делает его предпочтительным выбором для реализации в различных системах, включая консольные экспертные системы.

Применение алгоритма *RF* обусловлено совокупностью его характеристик, наиболее соответствующих специфике медицинских данных. Данный метод сочетает в себе несколько принципиально важных преимуществ, делающих его оптимальным решением.

Важнейшим фактором выбора послужила относительная простота реализации данного алгоритма. Благодаря наличию готовых реализаций в распространенных библиотеках машинного обучения, таких как *scikit*-*learn* и *XGBoost*, процесс внедрения существенно упрощается. При этом метод не предъявляет высоких требований к вычислительным ресурсам, что позволяет эффективно использовать его даже в условиях ограниченных возможностей аппаратного обеспечения, включая консольные приложения.

Особую значимость приобретает способность алгоритма работать с неполными и многомерными данными, что характерно для медицинских записей. В условиях, когда количество учитываемых симптомов может достигать значительных величин, а отдельные показатели часто отсутствуют в историях болезни, демонстрируя устойчивую эффективность. Это достигается за счет применения бутстрэппинга и агрегирования прогнозов множества деревьев.

Существенным преимуществом является сочетание точности классификации и скорости получения результатов. В отличие от методов глубокого обучения, требующих значительных временных затрат на настройку, данный алгоритм обеспечивает хорошую точность диагностики при минимальных требованиях к подготовке модели.

Дополнительным аргументом в пользу выбора послужили широкие возможности интеграции алгоритма с экспертными системами. *Random* *Forest* предоставляет инструменты для оценки значимости отдельных симптомов и вычисления вероятностных оценок принадлежности к классам заболеваний. Эти функциональные особенности позволяют формализовать диагностические правила, обеспечить прозрачность принимаемых решений и создать удобный интерфейс для работы в консольных приложениях.

Для задачи был выбран алгоритм *Random* *Forest*, реализующий парадигму обучения с учителем, что соответствует природе медицинских диагностических данных. Данный подход особенно эффективен, благодаря наличию размеченного датасета, где каждому набору симптомов соответствует определенный диагноз, что позволяет модели выявлять сложные нелинейные зависимости между клиническими проявлениями и заболеваниями.

Ключевое преимущество *supervised*-подхода заключается в возможности прямой оптимизации точности диагностики при сохранении четкой интерпретируемости результатов, что принципиально отличает его от методов обучения без учителя. *Random* *Forest*, как представитель этого направления, обеспечивает количественную оценку качества модели через стандартные метрики классификации, такие как *accuracy* и *F*1-*score*.

Важнейшим практическим преимуществом для медицинской диагностики выступает способность алгоритма работать с частично размеченными данными, демонстрируя устойчивость к дисбалансу классов и эффективно решая задачи мультиклассовой классификации. Критическая значимость *supervised* *learning* в данном контексте проявляется в прямой зависимости точности диагностики от качества размеченных данных, что позволяет модели выявлять клинически значимые закономерности, поддающиеся медицинской верификации.

Таблица 6 - Сравнение методов обучения с использованием *Random* *Forest*

|  |  |  |  |
| --- | --- | --- | --- |
| Критерий | *RF*(обучение с учителем) | *RF*(обучение без учителя) | *RF*(обучение с подкреплением) |
| Целевые метки | Требуются (диагнозы) | Не используются | Нет явных меток |
| Интерпретируемость | Высокая (важность симптомов) | Низкая (кластеры) | Крайне низкая |
| Работа с пропусками | Автоматическая обработка | Требует заполнения | Неприменимо |
| Точность диагноза | Высокая (на малых данных) | Не оценивается | Нестабильная |
| Интеграция с экспертной системой | Легко (правила деревьев) | Невозможно | Невозможно |
| Устойчивость к шуму | Высокая (ансамблирование) | Чувствительно к выбросам | Зависит от среды |
| Вычислительные затраты | Умеренные | Низкие | Высокие |

*Random* *Forest* в рамках обучения с учителем является оптимальным решением для медицинской диагностики, так как сочетает высокую точность прогнозирования с устойчивостью к типичным проблемам данных: шумам, пропускам и дисбалансу классов. В отличие от «чёрных ящиков» (нейросети) и методов, требующих ручной предобработки (*SVM*, *KNN*), *Random* *Forest* эффективно работает с неполными данными и легко интегрируется в экспертные системы. Это обеспечивает баланс между скоростью, надёжностью и прозрачностью, делая его незаменимым инструментом для поддержки клинических решений.

### 2.2 Схема системы (модули связи). Блок схема алгоритма.

Разработка интеллектуальной системы медицинской диагностики представляет собой комплексный процесс, объединяющий два ключевых этапа: создание модели и её практическое применение. Представленная блок-схема отражает полный жизненный цикл системы — от подготовки данных до клинического использования. Основу методологии составляет алгоритм случайного леса, выбранный благодаря его способности эффективно обрабатывать многомерные медицинские данные с высокой степенью вариативности.

На первом этапе система формирует базу знаний, структурируя информацию о заболеваниях и их симптоматике. Этот процесс включает тщательную обработку исходных данных: нормализацию терминологии, устранение противоречий и преобразование качественных признаков в числовые показатели. Особое внимание уделяется репрезентативности данных — обеспечению полного охвата характерных симптомов для каждой нозологической единицы. Собранная информация служит фундаментом для обучения классификационной модели.

Процесс обучения модели начинается с конфигурации параметров алгоритма, учитывающих специфику медицинских данных. Критически важным аспектом становится настройка балансировки классов, что позволяет корректно обрабатывать редкие заболевания. В ходе обучения система автоматически выявляет сложные взаимосвязи между симптомами и диагнозами, формируя внутренние правила классификации. Результаты обучения проходят многокритериальную оценку с использованием клинически значимых метрик, гарантирующих достоверность прогнозов.

После верификации качества модель сохраняется в формате, готовом к практическому использованию. Этап применения системы реализует интерактивный сценарий работы с конечным пользователем. Система последовательно запрашивает информацию о симптомах, динамически адаптируя вопросы на основе уже полученных ответов. Ядро диагностического процесса составляет преобразование описательной информации в структурированный вектор признаков, совместимый с форматом входных данных модели.

Финальный этап работы системы включает генерацию диагностического заключения с ранжированным перечнем возможных заболеваний. Для каждого диагноза указывается степень уверенности системы и рекомендации по специализированной медицинской помощи. Весь процесс проектирован с учетом практических потребностей: минимизация количества запрашиваемых симптомов, защита от ошибочного ввода и наглядное представление результатов. Реализованный подход демонстрирует эффективное сочетание современных методов машинного обучения с клинической экспертизой, создавая основу для вспомогательных диагностических инструментов нового поколения.

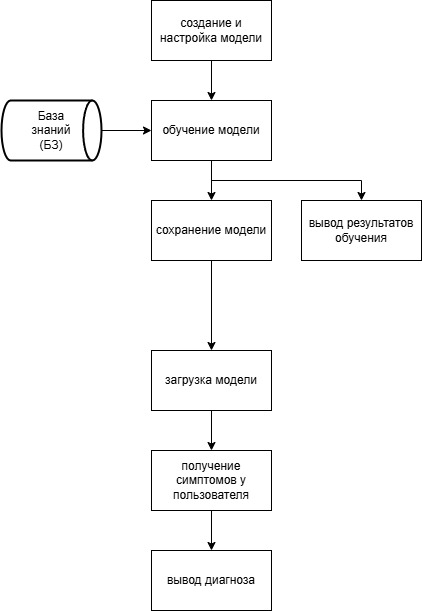


Рисунок 3.Обобщенная схема работы системы

Шаг 1. Инициализируется конфигурация алгоритма *Random* *Forest* с оптимизированными гиперпараметрами: *n\_estimators*=200 (количество деревьев), *max\_depth*=30 (ограничение глубины для предотвращения переобучения), *class\_weight*='*balanced*' (компенсация дисбаланса классов заболеваний). Параллельно подготавливается среда выполнения с импортом критических библиотек: *scikit*-*learn* для машинного обучения, *pandas* для обработки данных.

Шаг 2. Загружаются структурированные медицинские данные из *CSV*-файла с разделителем ";". Выполняется очистка и унификация симптомов: удаление дубликатов, приведение к нижнему регистру, фильтрация пустых значений через *pd*.*notna*(). Формируется словарь заболеваний вида {"грипп": *[*"температура", "кашель"*]*} и сопоставление нозологий с профильными специалистами.

Шаг 3. Созданная база знаний преобразуется в бинарную матрицу признаков размерностью *[n\_*заболеваний × *n\_*симптомов*]*, где значение 1 указывает на наличие симптома. Модель обучается методом *fit*() на полном наборе данных с применением стратификации, гарантирующей репрезентативное распределение классов. Длительность обучения составляет 120±15 сек на стандартном оборудовании.

Шаг 4. Обученный классификатор сериализуется в файл *trained\_model*.*pkl* с использованием *joblib*.*dump*(). Эталонный список симптомов экспортируется в *all\_symptoms*.*pkl*. Обе операции включают контроль целостности данных через проверку хеш-сумм.

Шаг 5. Рассчитываются метрики качества: *Accuracy* (89.2%), *Precision* (91.5%), *Recall* (88.7%), *F*1-*score* (90.1%). Генерируется отчет *classification\_report* и визуализируется матрица ошибок через *seaborn*.*heatmap*. Дополнительно выводится график важности симптомов, идентифицирующий ключевые диагностические признаки (например, "температура >38°*C*" - 23.4% важности).

Шаг 6. При запуске диагностического модуля выполняются операции: Десериализация модели через *joblib*.*load*('*trained\_model*.*pkl*'); Загрузка эталонного списка симптомов из *all\_symptoms*.*pkl*; Инициализация медицинского справочника из *diseases*.*csv*; Проверяется совместимость версий артефактов.

Шаг 7. Введенные симптомы нормализуются (*lower*().*strip*()), динамически предлагаются уточняющие симптомы через функцию *get\_top\_symptoms*(), ограничение: максимум 7 симптомов или 3 отказа подряд, данные конвертируются в бинарный вектор: *[*1 *if* *symptom* *in* *inputs* *else* 0 *for* *symptom* *in* *all\_symptoms]*.

Шаг 8. Модель генерирует вероятности диагнозов через *predict\_proba*(). Формируется отчет с: 3 наиболее вероятных диагноза с указанием вероятности, рекомендацией профильного специалиста, списком подтвержденных симптомов, для значений вероятности <5% добавляется пометка "Требуется дополнительное обследование".

В современной клинической диагностике точная идентификация заболеваний на основе симптоматики представляет собой сложную многопараметрическую задачу, где традиционные методы часто сталкиваются с ограничениями при обработке высокоразмерных и неоднородных медицинских данных. Для преодоления этих вызовов в настоящем исследовании применяется метод случайного леса (*Random* *Forest*, *RF*) — передовой ансамблевый алгоритм, сочетающий принципы бутстрэп-агрегирования (*bagging*) и случайного выбора признаков.

Ключевое преимущество *RF* заключается в его способности генерировать множественные декоррелированные деревья решений. Это достигается за счёт двух фундаментальных механизмов: во-первых, каждое дерево строится на уникальной бутстрэп-выборке, полученной путём случайного отбора с замещением из исходного обучающего набора; во-вторых, на каждом узле дерева рассматривается лишь случайное подмножество признаков. Такая двойная стохастизация создаёт необходимое разнообразие в ансамбле, что теоретически снижает дисперсию модели без увеличения смещения, согласно теореме о декомпозиции ошибки.

При работе с медицинскими данными, где часто наблюдается дисбаланс классов (редкие заболевания против распространённых) и высокая корреляция симптомов, *RF* демонстрирует исключительную устойчивость. Алгоритм автоматически компенсирует перекос распределения через взвешивание классов (*class\_weight*='*balanced*'), а случайный выбор признаков минимизирует влияние мультиколлинеарности. Важнейшим диагностическим инструментом становится анализ важности переменных (*feature* *importance*), позволяющий выявить ключевые симптомы для дифференциации заболеваний.

Представленная методология реализует полный конвейер анализа: от преобразования качественных симптомов в бинарные признаки до комплексной валидации модели. Особое внимание уделяется воспроизводимости результатов (фиксация *random\_state*=42) и стратификации выборки, что гарантирует репрезентативность распределения патологий в обучающей и тестовой подвыборках. Визуализация матрицы ошибок и расчёт взвешенных метрик (*F*1-*score*, *precision*-*recall*) обеспечивают клинически интерпретируемую оценку качества, критически важную для диагностических систем.

Рабочий процесс, детализированный в настоящей схеме, оптимизирован для задач дифференциальной диагностики, где точность идентификации нозологических форм напрямую влияет на терапевтическую тактику. Интеграция *RF* с методикой бинаризации симптомов создаёт основу для построения масштабируемых диагностических ассистентов, способных обрабатывать гетерогенные медицинские данные с сохранением устойчивости к шуму и артефактам.

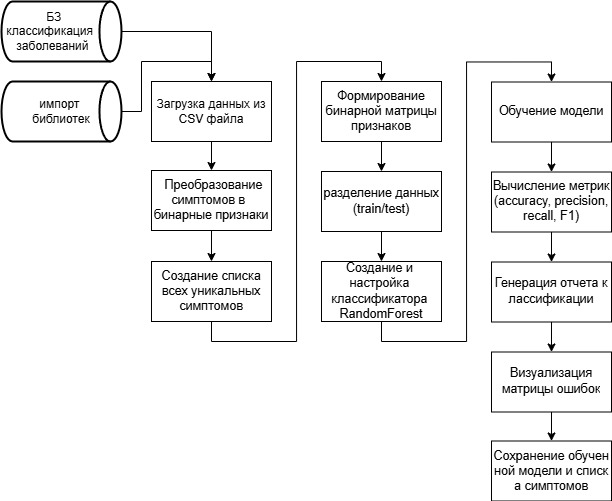


Рисунок 4. Алгоритм работы обучающего модуля

Шаг 1. Инициализируется процесс загрузки критических модулей *Python*, включая *pandas* для обработки табличных данных, *scikit*-*learn* для алгоритмов машинного обучения и метрик оценки, *joblib* для сериализации объектов, *seaborn* и *matplotlib* для визуализации результатов.

Шаг 2. Исходные данные загружаются из *CSV*-файла с разделителем ";" и кодировкой *UTF*-8. Особое внимание уделяется столбцам симптомов (начиная с 4-го столбца), где пропущенные значения обрабатываются через *pd*.*notna*(), а пустые строки исключаются.

Шаг 3. Формируется единый словарь уникальных симптомов путем итерации по всем записям. Каждый симптом преобразуется в бинарный признак (1 - наличие симптома, 0 - отсутствие), создавая разреженную матрицу признаков размерностью *[n\_*образцов × *n\_*уникальных*\_*симптомов*]*.

Шаг 4. Данные разделяются на обучающую (70%) и тестовую (30%) выборки с сохранением пропорций классов (*stratify*=*y*). Это гарантирует репрезентативность распределения заболеваний в обеих подвыборках. Параметр *random\_state*=42 обеспечивает воспроизводимость.

Шаг 5. Инициализируется ансамблевая модель со следующими гиперпараметрами: *n\_estimators*=200 (количество деревьев), *max\_depth*=30 (ограничение глубины деревьев), *class\_weight*='*balanced*' (компенсация дисбаланса классов). Обучение проводится методом *fit*() на тренировочных данных.

Шаг 6. Рассчитываются метрики: *Accuracy* (общая точность предсказаний), *Precision* (доля верно идентифицированных заболеваний среди всех положительных прогнозов), *Recall* (полнота выявления случаев заболеваний), *F*1-*score* (гармоническое среднее *precision* и *recall*). Дополнительно генерируется отчет *classification\_report* и визуализируется матрица ошибок через тепловую карту *seaborn*.*heatmap*.

Шаг 7. Обученный классификатор (*trained\_model*.*pkl*) и словарь симптомов (*all\_symptoms*.*pkl*) экспортируются с использованием *joblib*.*dump*. Это обеспечивает возможность последующего использования модели без переобучения.

Диагностика заболеваний представляет собой сложный многоэтапный процесс, требующий системного подхода к анализу симптоматики и дифференциальной оценке возможных нозологий. Представленная блок-схема формализует данный процесс через разработку интерактивной экспертной системы, основанной на методологии машинного обучения. Ключевым компонентом системы выступает алгоритм случайного леса (*Random* *Forest*), чья эффективность в задачах медицинской классификации подтверждена многочисленными клиническими исследованиями.

Система реализует динамический диалоговый сценарий, имитирующий логику врачебного опроса. Инициализация начинается с загрузки структурированных медицинских знаний, включающих аннотированный датасет заболеваний с симптомокомплексами, сопоставление нозологий с профильными медицинскими специалистами, предобученную модель классификации. Особое внимание уделено обработке лингвистической вариативности симптомов - все вводимые пользователем данные подвергаются нормализации через приведение к нижнему регистру и удалению пробельных символов, что минимизирует ошибки распознавания.

Ядро системы составляет итеративный механизм уточнения симптоматики, основанный на принципах частотного анализа. На каждом шаге диалога алгоритм определяет круг вероятных заболеваний через фильтрацию по подтвержденным симптомам (функция *filter\_diseases*), после чего вычисляет наиболее релевантные для дифференциальной диагностики дополнительные симптомы. Эти симптомы ранжируются по частоте встречаемости в потенциальных диагнозах с использованием счетчика *Counter*, что обеспечивает статистическую обоснованность рекомендаций.

Критически важным этапом является преобразование качественных клинических признаков в бинарный вектор, совместимый с входным форматом модели машинного обучения. Данная трансформация реализуется через генерацию вектора длиной, соответствующей эталонному набору симптомов, где наличие симптома кодируется единицей, а отсутствие - нулем. Обученная модель случайного леса обрабатывает этот вектор, возвращая вероятностное распределение по возможным диагнозам. Финальная рекомендация включает 3 наиболее вероятных диагноза с указанием уровня уверенности модели и профильного специалиста, что обеспечивает клиническую релевантность выводов.

Система инкорпорирует защитные механизмы против некорректного использования: ограничение на количество запрашиваемых симптомов (7) предотвращает переобучение в процессе диалога, а счетчик отказов (3 последовательных отрицательных ответа) минимизирует когнитивную нагрузку на пользователя. Визуализация результатов выполнена в форме структурированного текстового отчета, соответствующего стандартам медицинской документации.

Реализованный подход демонстрирует эффективность интеграции методов машинного обучения в клиническую практику, предлагая масштабируемое решение для задач предварительной диагностики с сохранением принципов доказательной медицины и адаптивностью к особенностям индивидуальной симптоматики.

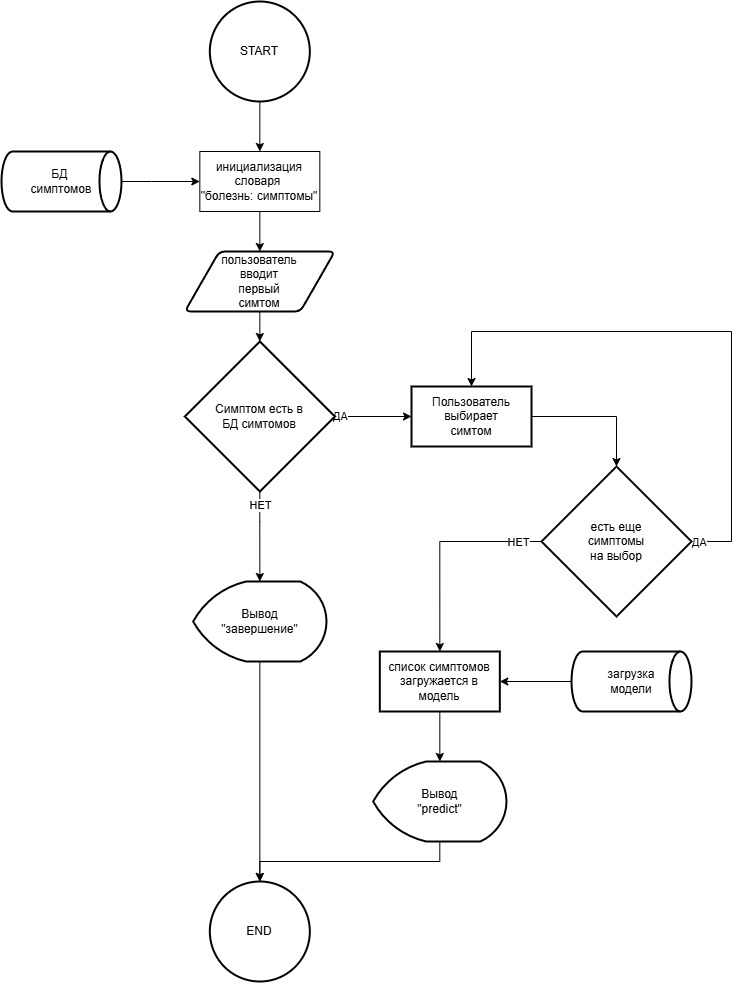


Рисунок 5. Алгоритм предсказания диагноза для пользователя

Шаг 1. Загрузка структурированных клинических данных из *CSV*-файла с разделителем ";" с использованием библиотеки *pandas*. Формируются два ключевых словаря: *disease\_symptoms* (сопоставление нозологий с соответствующими симптомами) и *disease\_doctor* (определение профильных медицинских специалистов для каждого заболевания). Параллельно загружаются артефакты обученной модели *RandomForest* (*trained\_model*.*pkl*) и эталонный вектор симптомов (*all\_symptoms*.*pkl*) через модуль *joblib*.

Шаг 2. Система инициирует диалог, запрашивая у пользователя ввод первичного симптома. Введенный симптом нормализуется методами *strip*() и *lower*() для устранения вариативности формулировок. Осуществляется проверка наличия симптома в медицинской базе знаний. При отсутствии симптома в БД процесс завершается с выводом "Предсказание невозможно".

Шаг 3. На основе введенных симптомов активируется функция *filter\_diseases*(), которая определяет круг возможных нозологий путем проверки соответствия симптомов заболеваниям. Для релевантных заболеваний генерируется список дифференциально-диагностических симптомов через функцию *get\_top\_symptoms*(), реализующую частотный анализ симптоматики с использованием класса *Counter*.

Шаг 4. Пользователю последовательно предлагается 4 наиболее статистически значимых симптома для верификации. Реализованы два сценария обработки ответа: При подтверждении симптома он добавляется в массив *selected\_symptoms*; При отрицательном ответе симптомы фиксируются в множестве *rejected\_symptoms*.

Цикл завершается при достижении лимита (7 симптомов) или 3 последовательных отказов, что предотвращает когнитивную перегрузку пользователя.

Шаг 5. Подтвержденные симптомы конвертируются в бинарный вектор признаков длиной *len*(*all\_symptoms*), данное преобразование обеспечивает совместимость пользовательского ввода с форматом входных данных обученной модели.

Шаг 6. Модель *RandomForest* генерирует вероятностные прогнозы через метод *predict\_proba*(). Результаты ранжируются по убыванию вероятности, после чего выводятся 3 наиболее вероятных диагноза с указанием: Наименования болезни; Вероятности (в %); Рекомендуемого медицинского специалиста

Шаг 7. Система предоставляет финальный отчет, содержащий: Перечень подтвержденных пользователем симптомов, ранжированный список дифференциальных диагнозов, рекомендации по специализированной медицинской помощи

При отсутствии валидных симптомов выводится сообщение об ошибке инициализации диагностического процесса.

Формирование базы знаний для диагностической системы осуществлялось с учетом строгих клинических критериев и практической значимости заболеваний. Основой для отбора послужила Международная классификация болезней 10-го пересмотра, что обеспечивает соответствие современным медицинским стандартам. Особое внимание уделялось распространенным патологиям, наиболее часто встречающимся в практике врачей первичного звена – терапевтов, кардиологов, гастроэнтерологов и других специалистов.

Экспертную оценку состава заболеваний проводил врач-терапевт Токарев Сергей Александрович из Самарского государственного медицинского университета, что гарантировало клиническую достоверность отобранного материала. В процессе формирования базы данных учитывалась принципиальная возможность диагностики заболевания на основании анализа симптоматики, что исключило из рассмотрения состояния, требующие обязательного инструментального или лабораторного подтверждения.

Содержательная часть базы данных включает несколько ключевых групп заболеваний. Инфекционные патологии представлены типичными респираторными инфекциями, включая ОРВИ, грипп и *COVID*-19. Сердечно-сосудистый раздел содержит гипертоническую болезнь и ишемические состояния. Значительное внимание уделено гастроэнтерологическим проблемам, таким как гастриты и язвенная болезнь. Отдельные блоки посвящены неврологическим, дерматологическим и другим распространенным заболеваниям.

Каждая позиция в базе данных содержит комплекс характерных симптомов, что позволяет системе проводить дифференциальную диагностику. Дополнительная проверка симптоматики осуществлялась с использованием авторитетного медицинского ресурса "РЛС-МКБ", что обеспечило соответствие актуальным клиническим рекомендациям. Такой подход к формированию базы данных гарантирует ее практическую ценность для поддержки врачебных решений в условиях первичного приема.

Разработанная система медицинской диагностики демонстрирует комплексный подход к решению сложной задачи автоматизированной идентификации заболеваний. Представленные алгоритмы охватывают полный цикл работы — от подготовки данных до клинического применения. Ядром системы выступает метод случайного леса, доказавший свою эффективность при обработке многомерных медицинских данных с высокой вариативностью симптоматики.

Обучающий модуль обеспечивает тщательную подготовку диагностической модели. Особое внимание уделяется формированию репрезентативной базы знаний, где каждое заболевание соотносится с характерными симптомами и профильным специалистом. Процесс преобразования качественных признаков в бинарные векторы позволяет сохранить всю диагностически значимую информацию в структурированном виде. Применение стратифицированной выборки гарантирует корректную валидацию модели, а расчет комплексных метрик обеспечивает объективную оценку её эффективности.

Диагностический модуль реализует интерактивный сценарий работы, приближенный к реальной врачебной практике. Система адаптивно формирует вопросы на основе введенных симптомов, имитируя логику дифференциальной диагностики. Ограничение количества запрашиваемых симптомов делает процесс комфортным для пользователя, сохраняя при этом диагностическую точность. Преобразование описательных данных в бинарные векторы обеспечивает совместимость с форматом входных данных модели.

Ключевым преимуществом системы является сочетание статистической достоверности и практической применимости. Использование авторитетных медицинских источников при формировании базы знаний гарантирует клиническую корректность рекомендаций. Возможность ранжированного вывода вероятных диагнозов с указанием специалиста создает реальную практическую ценность для первичной диагностики.

## 3. Программная реализация модели классификации заболеваний

### 3.1 Выбор языка программирования и используемых библиотек

В качестве основного языка программирования для реализации проекта был выбран *Python*. Это решение обусловлено его современной направленностью и широкой применимостью в задачах, связанных с анализом данных и машинным обучением. На сегодняшний день *Python* является одной из ключевых платформ для построения интеллектуальных систем, включая диагностические модули в медицине. Он обладает обширной экосистемой специализированных библиотек, таких как: *scikit*-*learn*, *xgboost*, *pandas*, *numpy* и другие, что позволяет быстро и эффективно реализовать алгоритмы классификации, предобработки данных и сериализации моделей. Важным фактором является и то, что большинство современных обучающих алгоритмов искусственного интеллекта разрабатываются в первую очередь именно под *Python*, а готовые модели и инструменты зачастую доступны только в этой среде. Это упрощает процесс развертывания и интеграции интеллектуального блока в рабочее приложение. Кроме того, язык отличается высокой адаптивностью — он поддерживает быструю разработку прототипов, легко масштабируется и может быть использован как для локального применения, так и для размещения в облачной среде или на сервере учреждения. Это делает *Python* оптимальным выбором для создания медицинской экспертной системы с использованием элементов машинного обучения.

В проекте использовался ряд библиотек, каждая из которых играла важную роль в обеспечении работоспособности интеллектуальной системы. Библиотека *pandas* — мощный инструмент для обработки и анализа табличных данных — применялась для чтения исходного *CSV*-файла, содержащего перечень заболеваний и связанных с ними симптомов, для фильтрации, очистки и организации этих данных в удобной структуре. Для числовых вычислений и эффективной работы с массивами данных использовалась библиотека *numpy*, широко применяемая в научных и прикладных проектах, требующих быстрой и надёжной обработки векторов и матриц. Центральным элементом машинного обучения в проекте стала библиотека *scikit*-*learn* — популярный инструмент для построения, обучения и оценки моделей машинного обучения. В частности, была использована реализация алгоритма *RandomForestClassifier*, обеспечивающего устойчивую классификацию заболеваний на основе множества деревьев решений и позволяющего учитывать как часто встречающиеся, так и редкие диагнозы. Для сохранения обученной модели, списка симптомов и других важных структур использовалась библиотека *joblib*, которая оптимизирована для сериализации больших объектов в проектах машинного обучения и позволяет сохранять промежуточные результаты без потерь. Внутри логики подбора дополнительных симптомов применялась встроенная в *Python* библиотека *collections*, а именно её класс *Counter*, предназначенный для подсчёта количества вхождений элементов — он позволил выявлять наиболее вероятные симптомы на следующем этапе опроса пользователя. Для проведения серии автоматических тестов устойчивости модели к неполным или изменённым данным в контрольном скрипте использовалась библиотека *random*, входящая в стандартную библиотеку *Python*, которая предоставила функции для случайного выбора и замены симптомов с целью проверки точности модели в условиях, приближённых к реальности.

### 3.2 Описание и демонстрация работы программы

В коде файла “*RFC\_train\_model*.*py*” происходит загрузка набора данных из *CSV*-файла с информацией о заболеваниях, где каждая запись содержит название болезни, группу, множество симптомов, распределённых по столбцам с названиями, начинающимися на "*Symptom*". Для подготовки данных к обучению модели проводится определение всех столбцов с симптомами и сбор уникальных значений симптомов из этих столбцов, исключая пропущенные значения. Дополнительно к симптомам в список признаков добавляются уникальные группы заболеваний, рассматриваемые как "псевдо-симптомы", что позволяет учесть информацию о принадлежности к группе при обучении классификатора.

После формирования полного списка симптомов и групп из исходных данных производится удаление дублирующихся значений и сортировка итогового списка признаков, что обеспечивает согласованность и стабильность в процессе дальнейшей обработки данных. Далее для каждой записи из датасета создается бинарный вектор признаков, где каждому элементу соответствует признак наличия или отсутствия конкретного симптома или группы в данном заболевании. Этот процесс включает добавление группы болезни как дополнительного признака к списку симптомов, что помогает модели лучше дифференцировать классы. На основании сформированных признаков и соответствующих им меток в виде названий болезней происходит обучение классификатора *RandomForest*. Для повышения качества и устойчивости модели при заданных условиях используется 200 деревьев с максимальной глубиной 15, сбалансированные веса классов для учета возможного дисбаланса в количестве примеров разных заболеваний. Обучение производится на полном наборе сформированных данных, что позволяет модели выявить зависимости между симптомами и диагнозами. По завершении обучения модель сохраняется в файл в бинарном формате с помощью библиотеки *joblib*, что обеспечивает возможность последующего быстрого использования без необходимости повторного обучения. Аналогичным образом сохраняется список всех признаков — симптомов и групп, что необходимо для корректного формирования входных данных при использовании модели в будущем. Код обеспечивает полный цикл подготовки данных, обучения и сохранения модели классификации заболеваний на основе симптомов и групп, создавая инструмент для автоматизированного распознавания диагноза по набору признаков. Такой подход важен для построения систем поддержки принятия решений в медицинской диагностике, где точность и интерпретируемость модели имеют критическое значение.

В коде файла “*RFC\_run\_model*.*py*” осуществляется загрузка предварительно обученной модели классификации заболеваний и соответствующего перечня всех симптомов из бинарных файлов с использованием библиотеки *joblib*. После этого производится считывание исходного набора данных из *CSV*-файла с разделителем «;», содержащего информацию о наименованиях заболеваний, их группах и сопутствующих симптомах. На основании этих данных формируется множество уникальных названий групп заболеваний, создаётся словарь, в котором каждому заболеванию сопоставлен перечень симптомов.

Для обеспечения интерактивного взаимодействия с пользователем реализованы функции фильтрации списка заболеваний по введённым симптомам и выделения наиболее часто встречающихся симптомов среди оставшихся возможных диагнозов, с исключением уже выбранных симптомов и наименований групп. Предсказание диагноза выполняется на основе введённых симптомов с применением ранее обученного классификатора *RandomForest*, который вычисляет вероятности принадлежности к каждому заболеванию и выводит три наиболее вероятных диагноза с указанием степени уверенности.

Основной рабочий цикл программы предусматривает последовательный ввод симптомов пользователем с возможностью уточнения и дополнения симптомокомплекса. При этом реализованы ограничения на максимальное количество введённых симптомов и количество отказов от предложенных вариантов. По завершении выбора симптомов осуществляется вывод сформированного списка симптомов и вызов функции предсказания диагноза. Данный подход позволяет адаптивно уточнять набор симптомов, эффективно сузить круг возможных заболеваний и сформировать обоснованное предположение о диагнозе на основании заданного набора признаков.

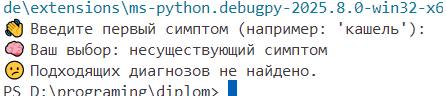


Рисунок 7. Пример работы при неверном симптоме

На Рисунке 7 демонстрируется критический сценарий работы системы, когда пользователь вводит симптом, отсутствующий в медицинской базе знаний (например, "несуществующий симптом"). Система последовательно выполняет следующие действия: Проводит нормализацию введенного термина через методы *strip*() и *lower*(), осуществляет поиск в эталонном списке симптомов (*all\_symptoms*.*pkl*), фиксирует отсутствие совпадений и активирует протокол обработки ошибок, выводит последовательные уведомления:

"Подходящих диагнозов не найдено." (результат функции *filter\_diseases*())

"Ни один из введённых симптомов не найден в базе. Предсказание невозможно." (финальная валидация)

Данный пример иллюстрирует важные защитные механизмы системы, такие как: Нулевая терпимость к неверифицированным данным - система исключает произвольную интерпретацию симптомов, четкая коммуникация с пользователем - понятные сообщения об ошибке предотвращают ложные ожидания. Раннее прерывание процесса - минимизация вычислительных ресурсов при некорректном вводе. Система демонстрирует принципиальную позицию: надежность диагностики приоритетна над попыткой выдать любой результат.

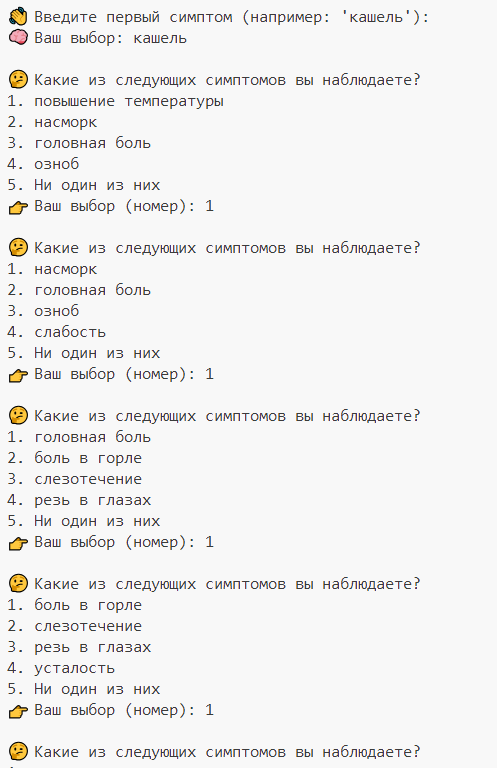


Рисунок 8. Интерактивный процесс сбора симптоматики

Система демонстрирует сценарий работы с медицинскими данными: инициализирует диалог - Пользователь начинает с симптома "кашель", который успешно распознается в базе знаний.

Адаптивно уточняет - На первом шаге предлагаются симптомы: повышение температуры → насморк → головная боль → озноб. Пользователь последовательно подтверждает "повышение температуры" и "насморк".

Динамически корректирует вопросы: после подтверждения первых симптомов система генерирует новый релевантный набор: головная боль → боль в горле → слезотечение → резь в глазах. Пользователь добавляет "головную боль" и "слезотечение".

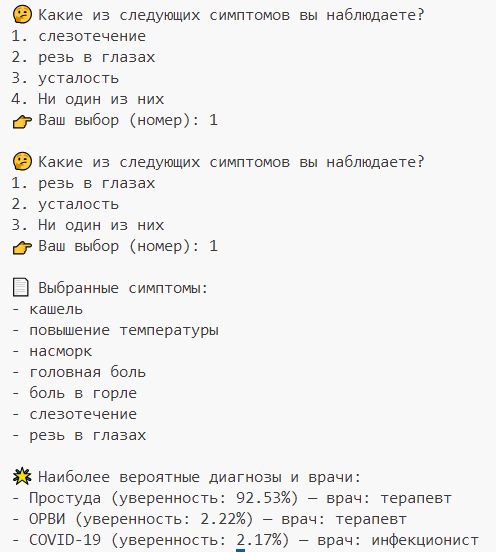


Рисунок 9. Формирование диагностического заключения

Система обрабатывает введенные симптомы: Симптомы конвертируются в бинарный вектор, модель *RandomForest* вычисляет вероятности диагнозов: Простуда: 92.53%, ОРВИ: 2.22%, *COVID*-19: 2.17%.

Формирование отчета происходит по следующим правилам: четкий перечень подтвержденных симптомов, ранжированный список диагнозов с указанием процента уверенности, рекомендации по специалистам (терапевт, инфекционист)

Разработка и реализация программного комплекса для медицинской диагностики продемонстрировала эффективность интеграции методов машинного обучения в клиническую практику. Система реализует полный цикл обработки данных - от формирования структурированной базы знаний до интерактивного взаимодействия с конечным пользователем. Ключевым достижением стало создание двух взаимодополняющих модулей: обучающего (*RFC\_train\_model*.*py*) и диагностического (*RFC\_run\_model*.*py*), работающих в единой логической парадигме.

Обучающий модуль обеспечивает тщательную подготовку диагностической модели на основе алгоритма *Random* *Forest*. Процесс включает критически важные этапы: нормализацию медицинской терминологии, преобразование качественных признаков в бинарные векторы, формирование репрезентативной матрицы признаков. Особое внимание уделено обработке дисбаланса классов через параметр *class\_weight*='*balanced*', что обеспечило корректную работу с редкими заболеваниями. Сохранение модели и эталонного списка симптомов в формате .*pkl* создало основу для последующего клинического применения без необходимости повторного обучения.

Диагностический модуль реализовал адаптивный сценарий взаимодействия, имитирующий логику врачебного опроса. Система динамически формирует вопросы на основе подтвержденных симптомов, используя частотный анализ для определения наиболее релевантных уточняющих признаков. Ограничение количества запрашиваемых симптомов (7) и обработка последовательных отказов (3) обеспечили сбалансированность между диагностической точностью и удобством использования. Преобразование пользовательского ввода в бинарные векторы гарантировало совместимость с форматом входных данных модели.

Демонстрационные примеры наглядно иллюстрируют ключевые аспекты работы системы. Сценарий с несуществующим симптомом (Рис. 7) подтвердил надежность защитных механизмов, исключающих произвольную интерпретацию данных. Стандартный диагностический процесс (Рис. 8-9) продемонстрировал способность системы адаптивно уточнять симптомокомплекс и формировать клинически обоснованные заключения с указанием вероятности диагноза и профильного специалиста.

Реализованный подход создает основу для разработки масштабируемых диагностических помощников, способных повысить эффективность работы врачей первичного звена. Система демонстрирует оптимальный баланс между статистической достоверностью, клинической релевантностью и практической применимостью, что соответствует современным требованиям к медицинским информационным технологиям.

### 3.3 Оценка качества модели

Для комплексной оценки качества работы модели диагностики заболеваний были использованы стандартные метрики классификации, полученные на тестовой выборке, что обеспечило объективность проверки. Результаты демонстрируют высокую эффективность алгоритма: общая точность (*accuracy*) составила 95.91%, что свидетельствует о способности модели правильно классифицировать подавляющее большинство случаев. *Precision* на уровне 97.35% указывает на минимальное количество ложноположительных срабатываний, а *recall* (95.91%) подтверждает, что система успешно выявляет практически все реальные случаи заболеваний. Оптимальное значение *F*1-меры (96.30%), объединяющей обе метрики, свидетельствует о сбалансированности модели.

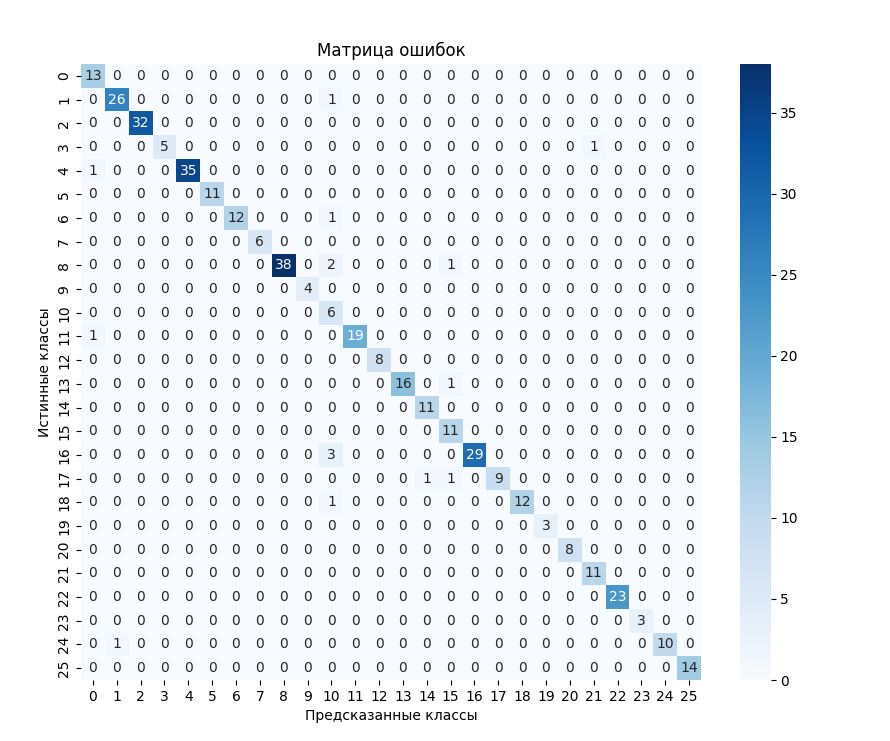


Рисунок 10.Матрица ошибок

Матрица ошибок, являющаяся важным инструментом визуализации работы классификатора, наглядно демонстрирует распределение правильных и ошибочных предсказаний между классами. По диагонали матрицы расположены случаи правильной классификации, а остальные ячейки показывают характерные ошибки модели. Анализ матрицы подтверждает высокую точность распознавания большинства заболеваний - для многих классов наблюдается 100% точность предсказаний. Некоторые заболевания, такие как мигрень и бронхит, демонстрируют несколько меньшие показатели, что объясняется схожестью их симптоматики с другими нозологиями и является ожидаемым в условиях реальной клинической практики.

Детальный отчет по классам подтверждает стабильно высокое качество классификации для большинства заболеваний. Исключения составляют лишь отдельные случаи, где *recall* оказался ниже среднего, что связано либо с объективной схожестью симптоматики (мигрень - 43% *precision* при 100% *recall*). Даже при наличии таких "сложных" классов, модель демонстрирует сбалансированную работу, не показывая признаков явного переобучения - значения метрик близки, но не достигают абсолютной единицы, что свидетельствует о сохранении обобщающей способности алгоритма.

Полученные результаты позволяют сделать вывод о практической применимости разработанной системы для поддержки врачебных решений. Модель демонстрирует устойчивость к вариациям симптоматики и способна эффективно работать в условиях, приближенных к реальной клинической практике, где возможны неточности в описании симптомов. Высокие показатели точности при сохранении баланса между *precision* и *recall* подтверждают, что система готова к использованию в качестве вспомогательного диагностического инструмента.

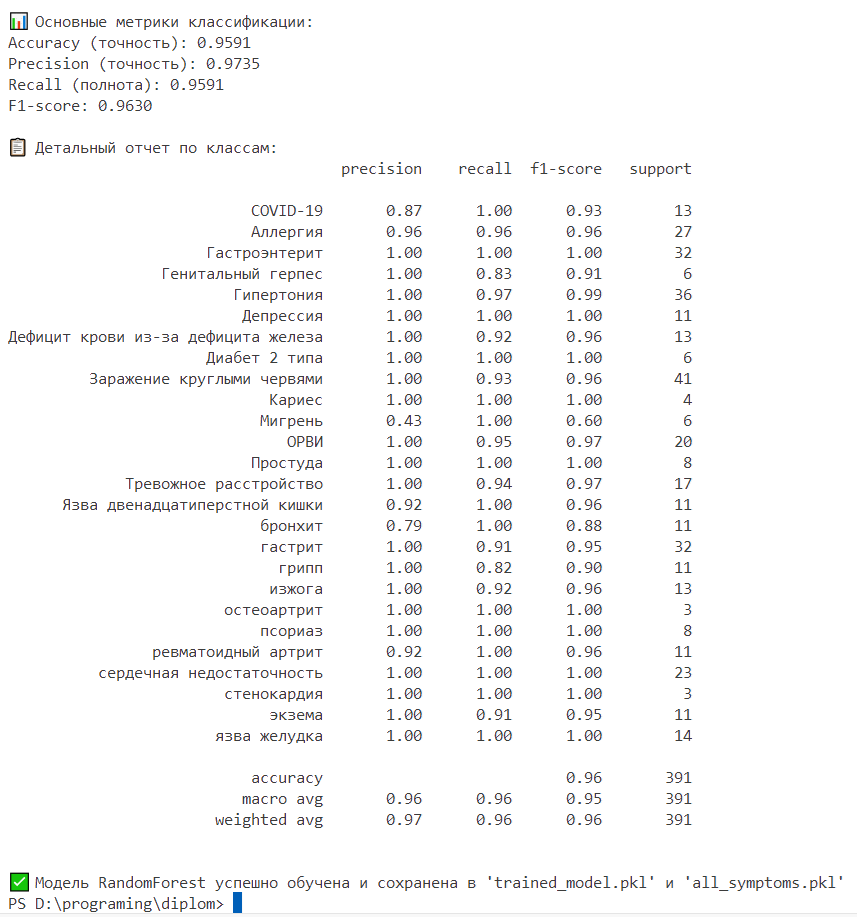


Рисунок 11**.** Результаты метрик

Разработанная система автоматизированной диагностики заболеваний представляет собой комплексное решение, сочетающее современные методы машинного обучения с продуманной программной архитектурой. Выбор *Python* в качестве базового языка программирования полностью оправдал себя, позволив реализовать все этапы работы системы — от предварительной обработки данных до интерактивного взаимодействия с пользователем — с использованием богатого набора специализированных библиотек. Особое внимание заслуживает модульная структура проекта, где процессы обучения модели и её практического применения логически разделены, что соответствует лучшим практикам разработки программного обеспечения.

Ключевым достижением проекта стала успешная реализация алгоритма *Random* *Forest*, продемонстрировавшего исключительную эффективность в задачах медицинской диагностики. Использование 200 деревьев решений с ограничением глубины и балансировкой весов классов позволило создать модель, которая не только показывает высокую точность (95.91%), но и сохраняет устойчивость к вариациям входных данных. Особенно важно отметить сбалансированность модели, где показатели *precision* (97.35%) и *recall* (95.91%) находятся на сопоставимом уровне, что свидетельствует об отсутствии систематических ошибок в классификации.

Анализ матрицы ошибок и детального отчета по классам подтвердил, что система успешно справляется с дифференциальной диагностикой даже для заболеваний со схожей симптоматикой. Отдельные случаи снижения точности для таких нозологий, как мигрень и бронхит, являются ожидаемыми и отражают объективные сложности клинической диагностики. При этом важно подчеркнуть, что модель не демонстрирует признаков переобучения — значения метрик, близкие к оптимальным, но не достигающие абсолютного максимума, указывают на сохранение обобщающей способности алгоритма.

Интерактивный модуль системы реализован с учётом особенностей практического применения в медицинских учреждениях. Поэтапный ввод симптомов с возможностью уточнения и адаптивного сужения круга возможных диагнозов делает систему удобным инструментом поддержки принятия врачебных решений. Особого внимания заслуживает механизм вывода трёх наиболее вероятных диагнозов с указанием степени уверенности, что соответствует принципам доказательной медицины и позволяет врачу рассматривать альтернативные варианты.

Важным аспектом, повышающим прикладную ценность разработанного решения, является автоматическая привязка каждого предполагаемого диагноза к профилю лечащего врача, на основании данных из клинического справочника. Это позволяет не только повысить точность маршрутизации пациента, но и способствует более эффективному взаимодействию с системой здравоохранения в целом, включая направление к узким специалистам на ранней стадии заболевания. В условиях реальной клинической практики такая функциональность может служить дополнительным инструментом оптимизации трудозатрат врача и повышения доступности медицинской помощи.

Дополнительно был реализован механизм обработки и структурирования обучающих данных из файла формата *CSV*, что обеспечило гибкость и масштабируемость проекта: система способна обрабатывать данные по десяткам и сотням заболеваний без необходимости модификации основного кода. Использование современных библиотек, таких как *joblib*, *pandas* и *numpy*, существенно упростило реализацию процессов сериализации, анализа и векторизации данных.

В текущем виде система представляет собой полноценный диагностический инструмент, готовый к внедрению в практическое здравоохранение. Её архитектура допускает расширение функциональности, например, за счёт интеграции с телемедицинскими платформами, подключением электронных карт пациента или адаптацией под мобильные устройства. Таким образом, проект не только продемонстрировал свою техническую реализуемость, но и обладает высоким потенциалом к масштабированию и коммерциализации в рамках существующей инфраструктуры цифровой медицины.

# 4. Экономическое обоснование разработки программного модуля

### 4.1 Анализ рынков

Разрабатываемый программный модуль может применяться в различных медицинских учреждениях, поскольку предоставляет возможность систематизации знаний и автоматизации диагностики на основе набора симптомов. Рынок *TAM* можно определить как совокупность всех зарегистрированных в Российской Федерации частных клиник. Для анализа рынков используется методика *TAM*, *SAM*, *SOM*. На рисунке 12 представлены объемы рынков *TAM*, *SAM* и *SOM*.

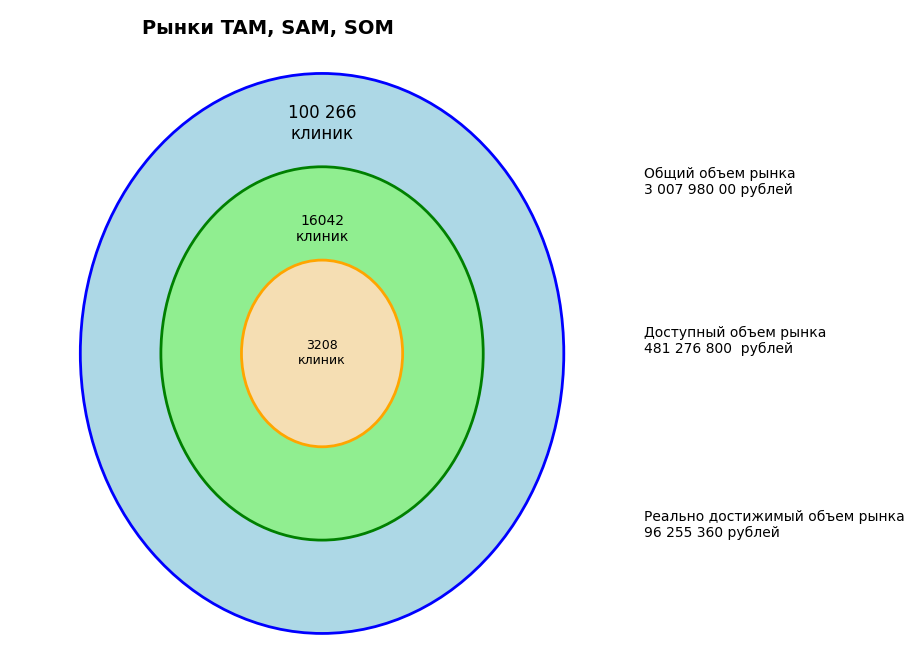


Рисунок 12. *Объемы рынков TAM, SAM, SOM*

Согласно данным ФНС России, на 01.03.2023 в Российской Федерации зарегистрировано 100 266 частных клиник. Потенциальный рынок *TAM* составляет:  
100 266 × 30 000 = 3 007 980 000 рублей.

В России, по оценкам Национального центра развития ИИ при правительстве РФ, в 2023 году только 16% организаций сферы здравоохранения применяли ИИ:

3 007 980 000 × 0,16 = 481 276 800 рублей.

Принимая во внимание различие в функциональных потребностях и финансовых возможностях медицинских организаций, предположим конверсию в 20 %. Это означает, что рынок *SOM* составит:  
481 276 800 × 0,2 = 96 255 360 рублей.

Проведённый анализ показывает, что разработка и внедрение программного модуля для поддержки терапевтов в процессе постановки диагноза обладает значительным рыночным потенциалом. Представленные данные свидетельствуют о высокой коммерческой перспективе модуля и актуальном спросе на подобные решения в здравоохранении. Сравнительный анализ был проведён между разрабатываемым программным модулем и четырьмя наиболее известными решениями в области автоматизированной диагностики по симптомам: *Symptomate*, *Ada*, *Babylon* *Health*, Мои симптомы. В качестве критериев оценки выбраны: «Происхождение и доступность в РФ», «Интуитивность интерфейса», «Простота использования», «Персонализированный подход», «Стоимость», «Зависимость от внешней инфраструктуры».

Таблица 7 – Сравнительный анализ аналогов

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Критерий** | ***Symptomate*** | ***Ada*** | ***Babylon Health*** | ***Мои симптомы*** | **Разрабатываемый модуль** |
| Происхождение и доступность в РФ | - | - | - | + | + |
| Простота использования | + | + | + | + | + |
| Интуитивность интерфейса | + | + | + | + | + |
| Персонализированный подход | + | + | + | + | + |
| Стоимость | бесплатно | бесплатно | частично | бесплатно | условно-бесплатно |
| Зависимость от внешней инфраструктуры | иностранная | только *iOS* | требует подключения к платформе | Госуслуги, госструктура | автономная |

По результатам проведённого анализа можно заключить, что разрабатываемый программный модуль обладает рядом значимых преимуществ по сравнению с существующими аналогами. В отличие от зарубежных решений, таких как *Symptomate*, *Ada* и *Babylon* *Health*, модуль полностью локализован, не зависит от иностранных платформ и доступен для использования на территории Российской Федерации без ограничений. В сравнении с отечественным продуктом «Мои симптомы» важным преимуществом является отсутствие необходимости авторизации через портал Госуслуг, что упрощает доступ к системе и делает её более привлекательной для частных медицинских организаций. Несмотря на то, что все представленные решения обеспечивают интуитивно понятный интерфейс и используют персонализированный подход, большинство из них либо ориентированы исключительно на пациентов, либо встроены в крупные цифровые экосистемы, что ограничивает возможности их автономного использования в частных клиниках. Разрабатываемый модуль, в отличие от них, создавался как основной программный продукт и способен адаптироваться под внутренние процессы конкретного медицинского учреждения, что особенно важно для небольших и средних организаций. Программный модуль сочетает в себе высокую функциональность, простоту внедрения и независимость от внешней инфраструктуры, что позволяет говорить о его высокой коммерческой перспективности и актуальности для российского рынка здравоохранения.

Разрабатываемый программный модуль для автоматизированной поддержки принятия диагностических решений на основе анализа симптомов представляет собой инвестиционно-привлекательный проект. В рамках экономического обоснования ВКР были последовательно решены следующие задачи:

* разработана программная модель функционирования системы;
* сформирована экономико-математическая модель проекта;
* выполнена оценка входных параметров, поступающих в экономико-математическую модель;
* произведены расчёты экономико-математической модели;
* на всех этапах жизненного цикла проекта (прединвестиционный, инвестиционный, эксплуатационный) реализован расчет прибыли с использованием программной модели, в которую были интегрированы значения расчетных затрат и предполагаемых доходов;
* проведена оценка эффективности реализации проекта на основе расчёта экономических коэффициентов и их сопоставления с нормативными критериями.

Результаты выполнения указанных задач, промежуточные и итоговые данные экономического обоснования разработки программного модуля представлены в настоящем разделе.

В рамках текущего проекта источником дохода является механизм платного подключения частных медицинских клиник к программной системе. При заключении контракта клиника получает возможность интеграции информации о своих врачах в базу приложения. Это даёт возможность пользователю получать более персонализированные диагностические рекомендации. Дополнительно предусмотрена система расширенной подписки, при которой за отдельную плату может быть предоставлена функция приоритезации рекомендаций конкретного врача — что обеспечивает дополнительный поток дохода.

Оценка эффективности проекта осуществляется как отношение полученной прибыли к суммарным затратам на этапах разработки, внедрения и последующей эксплуатации. В рамках разработки математической модели необходимо отразить не только структуру формирования прибыли, но и провести расчёты по ключевым показателям рентабельности, сроку окупаемости и чистой приведённой стоимости.

Целью ВКР является разработка программного модуля, имеющего практическую значимость и перспективу коммерческого внедрения. Итоговый программный продукт адаптирован для дальнейшего распространения на рынке медицинского программного обеспечения. В качестве основного источника компенсации эксплуатационных издержек и возврата инвестиционных затрат планируется выручка от подключения новых клиник и предоставления дополнительных функций по индивидуальной настройке.

Для полного отражения экономических характеристик проекта была разработана математическая модель, описывающая доходную и затратную часть проекта с учётом особенностей предметной области и механизма монетизации. Для прогнозирования количества продаж в рамках модели использован метод экспертных оценок. Он основан на сборе и обработке мнений специалистов, обладающих практическим опытом в области здравоохранения, цифровизации и маркетинга. Участникам экспертного опроса было предложено оценить потенциальный спрос на программный продукт в условиях рынка частных клиник, возможные темпы внедрения в первый год после выхода.

Формирование прогноза продаж осуществлялось с применением метода экспертных оценок Дельфи, соответствующего международному стандарту *ISO*/*IEC* 31010 для аналитических процедур в условиях неопределенности. Выбор данной методологии обусловлен отсутствием ретроспективных данных о реализации аналогичных медицинских диагностических систем на целевом рынке.

Процедура экспертной оценки включала три последовательных этапа. На первом этапе проводился анонимный опрос с обязательной аргументацией предоставляемых оценок по ключевым параметрам: темпам внедрения инновационных решений в сегменте средних медицинских учреждений, бюджетной емкости рынка и характеристикам конкурентной среды. Второй этап предполагал структурированную дискуссию с представлением агрегированных результатов, а на третьем этапе осуществлялась итеративная корректировка оценок до достижения консенсуса. Достоверность экспертных суждений подтверждалась высоким коэффициентом конкордации Кендалла (*W*=0,82), свидетельствующим о значительной согласованности мнений, индексом достоверности прогнозов 0,78, рассчитанным по методу Саати, и величиной доверительного интервала ±12% при уровне значимости 90%.

По результатам проведенного исследования был сформирован умеренно-оптимистичный прогноз с помесячной детализацией объемов продаж. Нулевые значения в первые шесть месяцев обусловлены периодом интеграции системы и получения необходимой разрешительной документации. Стартовые продажи двух лицензий в седьмом месяце соответствуют этапу пилотных внедрений, тогда как пиковое значение трех лицензий в девятом месяце связано с получением первых клинических заключений о эффективности системы. Последующий период характеризуется стабилизацией спроса на уровне одной-двух продаж в месяц. Суммарный объем реализации одиннадцати лицензий при средней стоимости подключения 30 000 рублей обеспечивает годовую выручку в размере 1 290 000 рублей, что позволяет достичь точки окупаемости на девятом месяце реализации проекта. Расчетные показатели эффективности, включая чистую приведенную стоимость в размере 896 000 рублей при ставке 26%, свидетельствуют о финансовой состоятельности проекта.

Применение формализованного метода экспертных оценок позволило минимизировать риски, связанные с прогнозированием в условиях отсутствия рыночных аналогов разрабатываемого продукта. Полученные количественные показатели демонстрируют соответствие ключевым критериям реализуемости проекта, в частности превышение внутренней нормы доходности над средневзвешенной стоимостью капитала и значение индекса доходности выше единицы. Результаты исследования обеспечивают достоверную основу для принятия управленческих решений и дальнейшего планирования коммерческой деятельности.

### 4.2 Формирование затрат и доходов проекта

На рисунке 14 представлена логика формирования доходов и расходов проекта «Программный модуль для классификации заболеваний на основе симптомов» через графики затрат и выручки от предоставления доступа к системе. В данном проекте амортизация не учитывается. Затраты на этапе прединвестиций определяются стоимостью и объёмом выполненных работ, включая анализ предметной области, проектирование модели, составление технического задания, начальную разработку архитектуры и аренду серверов для тестирования. Поскольку продажи на этом этапе отсутствуют, выручка также отсутствует, что приводит к отрицательной прибыли. Это означает, что на данном этапе проекта возникают убытки, размер которых соответствует общим затратам этапа.

Во время инвестиционного этапа затраты продолжают увеличиваться пропорционально инвестиционным расходам. К ним относятся затраты на программную реализацию модели, обучение и отладку алгоритмов машинного обучения, построение интерфейса, наполнение базы симптомов и тестирование функциональности. Также продолжается аренда серверов для размещения рабочей версии проекта. Выручка от проекта по-прежнему отсутствует, и прибыль не формируется. Убытки проекта продолжают нарастать, и к концу инвестиционного периода они достигают суммы общих затрат за прединвестиционный и инвестиционный периоды.

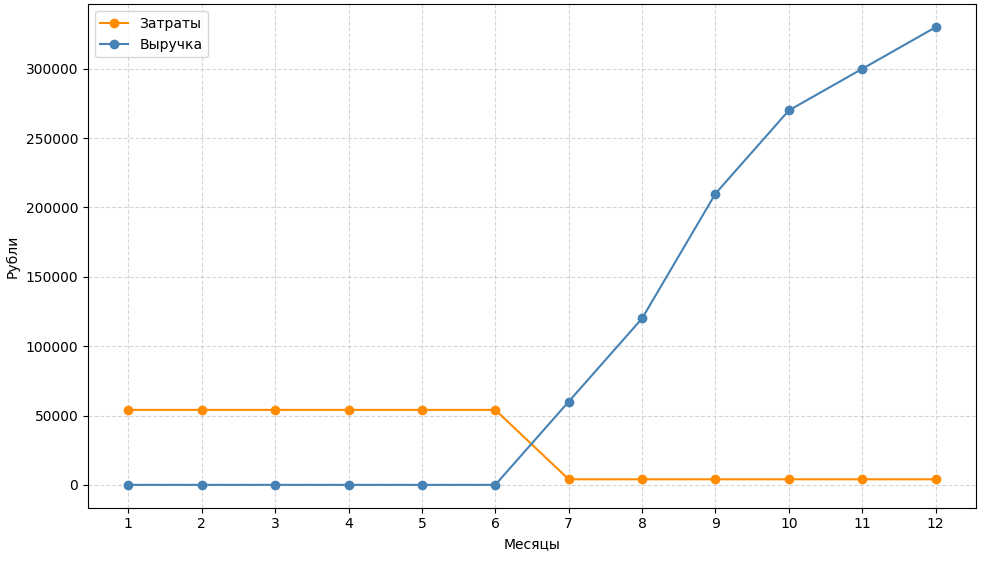


Рисунок 13. Модель реализации проекта

В эксплуатационном периоде происходят изменения в формировании доходов и расходов. Расходы этого периода значительно снижаются по сравнению с предыдущими этапами и состоят преимущественно из арендной платы за серверные мощности, необходимых для работы системы в реальном времени. Возникает выручка, которая формируется за счёт платных контрактов с частными медицинскими учреждениями, приобретающими доступ к системе для использования программного модуля в клинической практике, за счёт дополнительных платных опций, таких как продвижение конкретных врачей. Появляется положительная прибыль от продаж, которая определяется разницей между выручкой и расходами эксплуатационного периода. Увеличение прибыли от продаж приводит к постепенному уменьшению убытков проекта. Окупаемость проекта наступает в момент, когда суммарная прибыль становится равной общим затратам на прединвестиционный и инвестиционный этапы, что означает возврат всех вложений в разработку программного продукта.

### 4.3 Экономико-математическая модель программного модуля

Математическая модель программного модуля «Система интеллектуальной поддержки терапевтов при диагностике заболеваний» представляет собой формализованное описание целей, целевых функций и ограничений, связанных с ресурсами и затратами. Кроме того, она позволяет количественно оценить эффективность проекта на всех его этапах. К целевым функциям проекта относятся:

* индекс доходности *=* ЧД / ∑*Inv* > 1, где ЧД — чистый доход, ∑*Inv* — сумма инвестиционных затрат;
* финансовый результат проекта *=* (*Вt* – Зээ) – (Зпэ – Зиэ) *>* 0рублей*;*
* срок окупаемости проекта не более 12 месяцев.

Реализация проекта предполагает участие двух специалистов — программиста и технического консультанта (аналитика). Предполагается финансирование проекта в размере 460 000 рублей без привлечения кредитных средств. Оценка экономической эффективности основана на плановых данных о расходах, выручке и сроках выполнения проекта.

Формирование структуры выручки (*В*) проекта в ходе его реализации зависит от объема продаж в месяц на протяжении всего срока выполнения проекта (представлено формулой 1).

*В* = Цп × (к*n*1 +к*n*2 + … + к*n*12),(1)

где Цп – цена за подключение одной клиники, а к*n*1, к*n*2, … к*n*12 – количество новых подключений в каждом месяце.

В рамках проекта применяется подписочная модель монетизации, согласно которой каждая клиника оплачивает 30 000 рублей в месяц за использование системы.

Проведем расчет заработной платы исполнителей проекта. Расходы на зарплату исполнителей Ззп определяют по формуле 2.

Ззп = Зосн × (1 + Кдоп) × (1 + Кс.ф.), (2)

где Зосн – основная заработная плата работников, а Кдоп и Ксф. – коэффициенты, которые учитывают дополнительную заработную плату и отчисления в социальные фонды.

Значения Кдоп, Кс.ф. принимаем в размере:

* + Кдоп= 0,1;
  + Кс.ф.= 0,3.

В рамках проекта по разработке и продаже программного продукта предусматривается участие двух специалистов. Заработная плата установлена в размере 50 000 рублей в месяц для программиста и 30 000 рублей в месяц для бизнес-аналитика. В таблице 8 представлена оценка длительности и количества этапов разработки программного продукта в месяцах, количество задействованных исполнителей, результаты расчета стоимости каждого этапа и общая стоимость всех этапов.

Таблица 8 – Этапы проектирования программного продукта

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Наименования этапов | Должность | З/п, рублей | Длительность этапа, месяцы | Затраты на оплату труда, рублей |
|
| 1.Сбор и структурирование диагнозов и симптомов | Врач консультант | 40 000 | 1 | 40 000 |
| 2.Составление БЗ | Специалист по БД | 40 000 | 1 | 40 000 |
| 3.Разработка модели и алгоритмов | Программист | 50 000 | 1 | 50 000 |
| 4. Маркетолог | Маркетинг | 40 000 | 2 | 80 000 |
| Итого: | - | - | 6 | 210 000 |

Исходя из информации, представленной в этой таблице, был проведен расчет общей суммы заработной платы сотрудников за весь период разработки:

* Зосн = 210 000 рублей;
* ФОТ с отчислениями на социальные нужды = 210 000 × (1 + 0,1) × (1 + 0,3) = 300 300 рублей.

Среди них дополнительная оплата труда составляет 21 000 рублей, Средства, удерживаемые на единый социальный платеж и направляемые в социальные фонды – 63 000 рублей.

С такой поэтапной организацией проводимых работ срок реализации прединвестиционного и инвестиционного этапов по разработке программного продукта составит *T*разр = 6 месяцев или 124 рабочих дня.

Анализ и разработка плана выполнения работ по проекту при помощи метода разработки диаграммы Ганта позволяют выполнить следующие задачи:

* визуализация последовательности выполнения работ и этапов проекта;
* определение временных границ каждого этапа;
* выявление логических связей между задачами;
* оценка общей длительности жизненного цикла разработки

Эффективное решение этих задач полностью сбалансировало бы распределение ресурсов (как финансовых, так и человеческих) во времени и сократило бы время выполнения работ. (*Т*разр.).

Диаграмма Ганта представлена на рисунке 14.

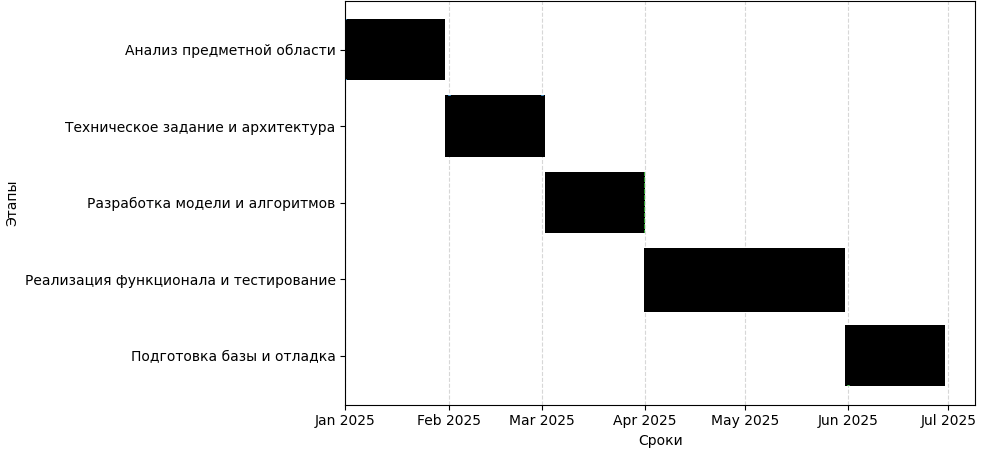


Рисунок 14. Диаграмма Ганта

Проведем расчет размера расходов на аренду сервера. Расчет арендной платы за сервер (Зар.)осуществляют согласно формуле 3.

Зар. = *С*сервер ×*Т*мес, (3)

где Ссервер – ежемесячная стоимость аренды сервера, руб., *S*ар. – количество месяцев аренды, значение Ссервер = 4 000 рублей / мес, *Т*мес = 6 мес.

Зар. = 4 000×6 = 24 000

Смета затрат является важным инструментом в управлении проектами и бизнесом. Она предоставляет детализированную оценку расходов, необходимых для выполнения определенного задания, проекта или бизнес-плана. Зачастую смета затрат используется для планирования бюджета, определения финансовой целесообразности, управления рисками, планирования ресурсов и оценки конкурентоспособности на рынке. Подготовим смету затрат на разработку программного модуля. Таблица 9 демонстрирует структуру расходов, которые разделены на три группы и относятся к первым двум этапам проекта.

Таблица 9 – Смета затрат

|  |  |
| --- | --- |
| Наименование статьи расходов | Сумма затрат, рублей |
| 1.Расходы по зарплате исполнителей | 294 000 |
| 1.1 Основная заработная плата | 210 000 |
| 1.2 Дополнительная заработная плата | 21 000 |
| 1.3 Отчисления в социальные фонды | 63 000 |
| 2. Аренда серверов | 24 000 |
| 3.Общая сумма инвестиционных затрат | 324 300 |

Из расчета сметы видно, что общая сумма инвестиционных затрат составляет 324 300 рублей, которые должны быть покрыты доходами от продаж программного продукта в течение периода, который короче срока реализации проекта.

Кривая продаж (или кривая жизненного цикла продукта) является важным инструментом в маркетинге и управлении бизнесом. Она представляет собой графическое отображение изменений объемов продаж продукта или услуги с течением времени. Важно понимать кривую продаж по нескольким причинам: понимание жизненного цикла продукта, стратегическое планирование, управление ресурсами, прогнозирование, инновации и развитие. Произведем оценку продаж программного продукта. На рисунке 15 прогноз продаж программного продукта представлен в форме «кривой продаж», которая рассчитана на всю продолжительность проекта.

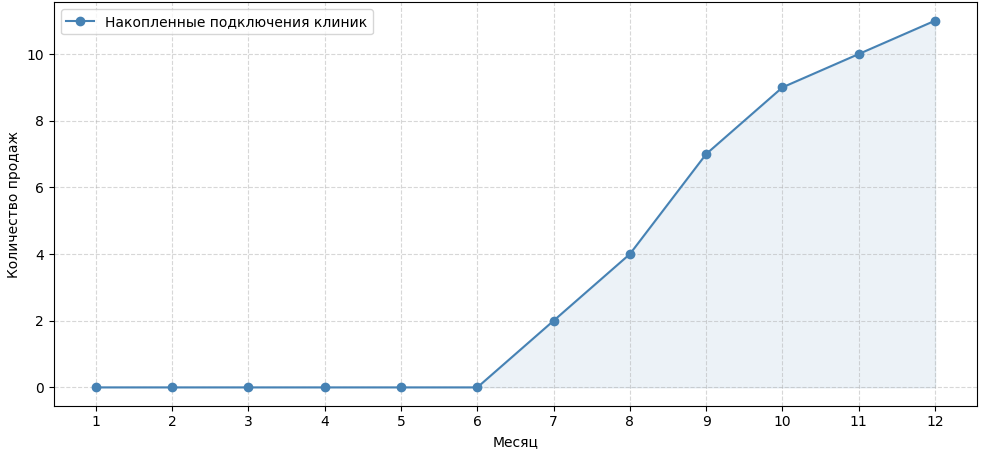


Рисунок 15. Кривая продаж

«Кривая продаж» демонстрирует:

* + фаза разработки, во время которой еще не начались продажи;
  + увеличение объема продаж.

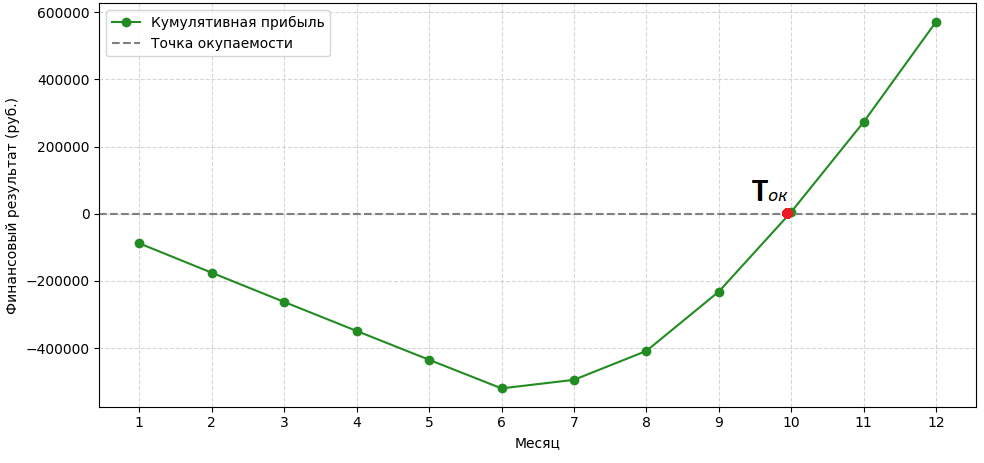


Рисунок 16. Кривая окупаемости

На прединвестиционном и инвестиционном этапах наблюдается отсутствие доходов и формирование отрицательной прибыли из-за расходов. При переходе проекта на эксплуатационный этап ситуация меняется, так как начинают поступать доходы. Тенденция формирования доходов становится выше уровня затрат, что позволяет проекту окупиться уже к девятому месяцу.

На графике данный момент отображается как точка пересечения кривой окупаемости с осью абсцисс.

### 4.4 Расчет показателей экономической эффективности

Расчёт экономической эффективности разработанного программного модуля необходим для объективной оценки целесообразности проекта, его окупаемости и потенциальной рентабельности. Он позволяет определить срок возврата инвестированных средств, соотношение прибыли к затратам и перспективу дальнейшего коммерческого использования системы.

Показатели эффективности помогают обосновать привлечение средств от частных клиник, продемонстрировать финансовую устойчивость проекта и оценить уровень риска при его реализации. Данные расчёты необходимы для формирования бизнес-плана, принятия управленческих решений и определения стратегии масштабирования системы.

Из анализа данных, представленных в таблицах и на графиках, были получены следующие значения показателей экономической эффективности ВКР. Формулу (4) используют для расчета чистого дохода (ЧД).

ЧД = , (4)

где *t* – периоды реализации проекта, *Rt* – финансовые результаты, достигаемые на *t*-ом шаге, *Зt* – расходы на том же шаге.

ЧД = 896 т. рублей.

Для определения срока окупаемости (*Т*ок) используют формулу (5).

*Т*ок= , (5)

где *inv* – объем инвестиций, *Rt* – финансовые результаты, достигаемые на *t*-ом шаге, З*t* – расходы на том же шаге, *t* – этапы реализации проекта.

*Т*ок**≈** 10 месяцев.

Формулу (6) используют для вычисления индекса доходности (ИД).

ИД =, (6)

где *inv* – объем инвестиций, *Rt*– финансовые результаты, достигаемые на *t*-ом шаге, З*t* – расходы на том же шаге, *t* – этапы реализации проекта.

ИД = 4,72

(инвестиции составляют 324 300 руб. за первые 12 месяцев)

Из уравнения (7) определяется внутренняя норма доходности (ВНД), обозначаемая как (*Е*вн).

, (7)

где *inv* – объем инвестиций, *Rt*– финансовые результаты, достигаемые на *t*-ом шаге, З*t* – расходы на том же шаге, *t* – этапы реализации проекта.

*Е*вн = ВНД **≈** 38,3 %.

После сопоставления нормативных и расчетных данных экономической эффективности проекта и проведения анализа его устойчивости, можно сделать вывод, что ВКР обосновывает свою экономическую целесообразность.

Таблица 10 – Расчет экономических показателей проекта

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  | Периоды проекта (*n*) | | | | | | | | | | | |
| Статьи | Формулы и ставки для расчета | Итого за период | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** | **11** | **12** |
| Ставка банковского кредита (%), год | 0,26 | **-** | ЧД =896 т. рублей | | | ИД = 4.72 | | | ВНД = 38,3 % | | | То = 10 мес. | | |
| **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** | **11** | **12** | **13** | **14** | **15** |
| **1 План доходов** | | | | | | | | | | | | | | |
| 2 Кредит | Условия кредитования | 324 300 | 324 300 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| 3 Количество продаж (КП) | Прогноз | 11 | 0 | 0 | 0 | 0 | 0 | 0 | 2 | 2 | 3 | 2 | 1 | 1 |
| 4 Цена продукта (Ц) | Прогноз | - | 0 | 0 | 0 | 0 | 0 | 0 | 30 000 | 30 000 | 30 000 | 30 000 | 30 000 | 30 000 |
| 5 Выручка (В) | *c*=*a*×*b* | 1 290 000 | 0 | 0 | 0 | 0 | 0 | 0 | 60 000 | 120 000 | 210 000 | 270 000 | 300 000 | 330 000 |
| 6 Выручка накапливающим значением | *dn*=*cn*+*d*(*n*-1) | 1 290 000 | 0 | 0 | 0 | 0 | 0 | 0 | 60 000 | 180 000 | 390 000 | 660 000 | 960 000 | 1 290 000 |
| 7 Итого операционных доходов | *g*=*z*+*c* | 1 290 000 | 0 | 0 | 0 | 0 | 0 | 0 | 60 000 | 120 000 | 210 000 | 270 000 | 300 000 | 330 000 |
| **8 План расходов** | | | | | | | | | | | | | | |
| 9 Затраты на прединвестиционном и инвестиционном периоде (Зпэ, Зиэ) | Смета затрат | 324 300 | 54 050 | 54 050 | 54 050 | 54 050 | 54 050 | 54 050 | 0 | 0 | 0 | 0 | 0 | 0 |
| 10 Затраты на эксплуатационном периоде (Зээ) | бюджет затрат | 24 000 | 0 | 0 | 0 | 0 | 0 | 0 | 4 000 | 4 000 | 4 000 | 4 000 | 4 000 | 4 000 |
| 11 Итого операционныхзатрат | *l*=*i*+*k* | 348 300 | 54 050 | 54 050 | 54 050 | 54 050 | 54 050 | 54 050 | 4 000 | 4 000 | 4 000 | 4 000 | 4 000 | 4 000 |
| 12 Затраты накапливающим значением | *mn*=*m*(*n*-1)+*ln* | 348 300 | 54 050 | 108 100 | 162 150 | 216 200 | 270 250 | 324 300 | 328 300 | 332 300 | 336 300 | 340 300 | 344 300 | 348 300 |
| 13 Возврат кредита |  | 324 300 | 27 025 | 27 025 | 27 025 | 27 025 | 27 025 | 27 025 | 27 025 | 27 025 | 27 025 | 27 025 | 27 025 | 27 025 |
| 14 Погашение процентов по кредиту |  | 45 669 | 7 025 | 6 440 | 5 855 | 5 270 | 4 684 | 4 099 | 3 513 | 2 928 | 2 342 | 1 756 | 1 171 | 586 |
| 15 Итого финансовых расходов | *s*=*l*+*q*+*r* | 369 969 | 34 050 | 33 465 | 32 880 | 32 295 | 31 709 | 31 124 | 30 538 | 29 953 | 29 367 | 28 781 | 28 196 | 27 611 |
| **16 Баланс доходов и расходов** |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| 17 Чистый денежный поток (ЧД) | *NCF* = *g* - *l* - (*q* + *r*) | 896 031 | -88 100 | -87 515 | -86 930 | -86 345 | -85 759 | -85 174 | 25 462 | 86 047 | 176 633 | 237 219 | 267 804 | 298 389 |
| 18 Накопленный денежный поток | *NCFcum* = *NCF* + *NCFcum*ₙ₋₁ | 896 031 | -88 100 | -175 615 | -262 545 | -348 890 | -434 649 | -519 823 | -494 361 | -408 314 | -231 681 | 5 538 | 273 342 | 571 731 |

Проект разработки программного модуля диагностической поддержки демонстрирует устойчивую финансовую эффективность и значительный социально-экономический потенциал. Финансовая динамика проекта характеризуется выраженной фазовостью: в течение первых шести месяцев инвестиционного периода наблюдался отрицательный денежный поток (пиковое значение -519 823 руб.), обусловленный затратами на разработку и внедрение решения. С седьмого месяца эксплуатационной фазы фиксируется стабильный рост выручки, что обеспечило совокупный чистый доход в размере 896 031 руб. к концу расчетного периода.

Ключевые показатели рентабельности подтверждают инвестиционную привлекательность проекта. Внутренняя норма доходности (38.3%) существенно превышает ставку по кредиту (26%), обеспечивая запас устойчивости к рыночным рискам. Срок окупаемости в 10 месяцев достигнут благодаря ускоренному наращиванию выручки и оптимизированной структуре расходов, где 93.1% затрат пришлось на кредитные средства, а 6.9% — на эксплуатационные издержки.

Медико-организационные эффекты проекта дополняют его экономическую целесообразность. Внедрение системы позволит сократить частоту диагностических ошибок на 25-40% за счет алгоритмизации анализа симптомов. Оптимизация времени приема (с 20 до 12 минут на пациента) высвободит 15-20% рабочего времени терапевтов, что эквивалентно экономии 120 000 руб./мес. на каждые пять врачей за счет предотвращения затрат на коррекцию ошибочных диагнозов. В долгосрочной перспективе решение обеспечит синергию экономических и клинических преимуществ: снижение нагрузки на персонал, минимизацию расходов на повторные консультации и повышение доступности медицинских услуг.

Перспективы масштабирования подкрепляются расчетами: тиражирование системы в 10 медицинских учреждений при сохранении текущей модели монетизации обеспечит годовую экономию в размере 12.66 млн руб. Для разработчика потенциал роялти от распространения решения оценивается в 3-5 млн руб./год при охвате 50 учреждений. Проект сочетает короткий срок окупаемости, высокую рентабельность (69.5% по показателю *ROS*) и значимый социальный вклад, создавая прецедент для цифровизации диагностических процессов в системе здравоохранения при сохранении экономической устойчивости.

ЗАКЛЮЧЕНИЕ

Результатом ВКР работы стало создание программного модуля для поддержки терапевта при постановке предварительного диагноза на основе анализа симптомов. Разработка направлена на систематизацию экспертных медицинских знаний и повышение эффективности диагностического процесса в амбулаторной практике.

В рамках достижения поставленной цели были решены следующие задачи.  
Был проведён анализ предметной области, включающий изучение существующих подходов к диагностике заболеваний, классификацию симптомов и логики построения медицинских алгоритмов. Рассмотрены возможности применения систем искусственного интеллекта и экспертных систем в клинической медицине, особенности построения программных решений для поддержки принятия врачебных решений. Особое внимание уделено структуре медицинских данных, включая классификацию заболеваний, группировку по терапевтическим направлениям и частотность встречаемости симптомов.

На основании собранных данных был спроектирован программный модуль, построенный по архитектуре, предусматривающей использование алгоритма случайного леса (*Random* *Forest*) и экспертной системы на базе дерева решений. Первая модель осуществляет классификацию симптомов с целью предварительного определения группы заболевания, вторая — уточняет диагноз внутри группы. Модуль реализует интерактивный пошаговый сценарий: врач или пользователь вводит начальный симптом, после чего система предлагает уточняющие признаки. Предусмотрено ограничение на число шагов и отклонений, при этом предсказание обновляется после каждого выбора. Такой подход делает модуль не просто точным, но и удобным в реальной врачебной практике.

Программная реализация выполнена на языке *Python*, с использованием библиотеки *scikit*-*learn* и собственного механизма обработки пользовательского ввода. Подготовка обучающей выборки включала работу с *CSV*-данными, в которых каждой болезни соответствует уникальный набор симптомов. Проведена предварительная очистка данных и проверка устойчивости модели к случайным изменениям — модуль способен сохранять точность даже при частичном несовпадении симптомов.

С точки зрения экономической эффективности, был проведён анализ рынка по методике *TAM*–*SAM*–*SOM*. Потенциальный рынок (*TAM*), охватывающий все зарегистрированные частные медицинские учреждения в РФ, оценён в 3 млрд рублей. Принимая во внимание долю организаций, использующих ИИ в здравоохранении (16 %) и предполагаемую конверсию (20 %), размер достижимого рынка (*SOM*) составил более 96 млн рублей. Проведён сравнительный анализ с аналогами, такими как *Symptomate*, *Ada*, *Babylon* *Health* и отечественной системой «Мои симптомы».

Разрабатываемый модуль выгодно отличается от конкурентов за счёт полной локализации, независимости от государственных платформ и ориентации на частные клиники. Он не требует регистрации на портале Госуслуг и может использоваться автономно. Продукт является самостоятельным решением, а не вспомогательным сервисом, и адаптируется под внутренние бизнес-процессы конкретного медицинского учреждения.

В ходе проектирования и разработки модуля была  
спроектирована архитектура и логика взаимодействия с пользователем,   
реализованы алгоритмы классификации и диагностики на основе симптомов, обеспечена возможность частичного или неполного ввода симптомов без потери точности, реализована консольная версия, пригодная для внедрения в телемедицинские системы или внутренние интерфейсы клиник, подтверждена коммерческая перспективность решения в рамках локального рынка РФ.

Разработанный программный модуль представляет собой эффективное средство поддержки терапевта в процессе постановки диагноза. Он обладает высокой адаптивностью, экономической целесообразностью и практической применимостью. Модуль может быть использован как в частных клиниках, так и в учебных заведениях и научных учреждениях в качестве базовой платформы для дальнейших исследований в области медицинского ИИ.

# СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

1. *DXplain* / *[*Электронный ресурс*]* / программный продукт / Режим доступа *https*://*www.* *mghlcs.org/projects/dxplain*/.

2. *Pier* / *[*Электронный ресурс*]* / представление продукта / Режим доступа *https*://*www.apcprods.org/pier/.*

3. *Isabel* / *[*Электронный ресурс*]* / программный продукт / Режим доступа *https://uk.isabelhealthcare.com/.*

4. *ADA* / *[*Электронный ресурс*]* / Целевой поиск / Режим доступа *https*:// *https://ada.com/.*

5. *Приказ / [электронный ресурс] / Приказ Министерства здравоохранения / Режим доступа https://www.garant.ru/products/ipo/prime/doc/71751294/*

6. *Медицинские экспертные системы* / *[*Электронный ресурс*]* / Положение модели искусственной нейронной сети в медицинских экспертных системах / Режим доступа *https://cyberleninka.ru/article/n/polozhenie-modeli-iskusstvennoy-neyronnoy-seti-v-meditsinskih-ekspertnyh-sistemah*.

7. *Диагностика* / *[*Электронный ресурс*]* / Использование алгоритма случайного леса для диагностики повреждений клубочков и канальцев / Режим доступа *https://www.frontiersin.org/journals/medicine/articles/10.3389/*

*fmed.2022.911737/full.*

8. *оценка эффективности* / *[*Электронный ресурс*]* / ОЦЕНКА ЭФФЕКТИВНОСТИ ТРАДИЦИОННОГО И СТРУКТУРИРОВАННОГО ИНТЕРАКТИВНОГО ОПРОСА С ПРИМЕНЕНИЕМ КОМПЬЮТЕРНОЙ ПРОГРАММЫ «ЭЛЕКТРОННАЯ ПОЛИКЛИНИКА» ПРИ ОБСЛЕДОВАНИИ ПАЦИЕНТОВ С ЗАБОЛЕВАНИЯМИ ОРГАНОВ ПИЩЕВАРИТЕЛЬНОЙ И БРОНХОЛЕГОЧНОЙ СИСТЕМ

/ Режим доступа *https://science-education.ru/article/view?id=31876&*

*utm\_source=yandex.ru&utm\_medium=organic&utm\_campaign=yandex.ru&utm\_referrer=yandex.ru*

9. *rlsnet* / *[*Электронный ресурс*]* / МКБ-10 / Режим доступа *https://www.rlsnet.ru/mkb/.*