Geekbrains

**Разработка приложения для отбора случаев стационарного лечения на экспертизу качества медицинской помощи с применением технологий искусственного интеллекта**

IT-специалист:

Инженер искусственный интеллект.

Гусев Д. Д.

Тула

2024

**СОДЕРЖАНИЕ**

ВВЕДЕНИЕ………………………………………………………………………3

1 Аналитический обзор………………………………………………………….10

2 Материалы и методики данной работы………………………………………14

2.1 Материалы данной работы…………………………………………...14

2.2 Методики данной работы…………………………………………….14

2.2.1 Общая информация о приложении…………………………14

2.2.2 Разработка модели классификатора и его выбор………….16

2.2.3 Используемые библиотеки……………………….…………19

3 Результаты работы…………………………………………………………….20

3.1 Получение обучающей выборки…………………………………….20

3.2 Обработка полученных данных……………………………………..22

3.3 Создание нейросетевой модели и ее обучение…………………….23

3.4 Результат создания приложения…………………………………….29

Заключение………………………………………………………………………30

Список используемой литературы…………………………………….……….31

**ВВЕДЕНИЕ**

Тема работы:

Разработка приложения для отбора случаев стационарного лечения на экспертизу качества медицинской помощи с применением технологий искусственного интеллекта.

Цель работы:

Создание приложения для отбора случаев стационарного лечения на экспертизу качества медицинской помощи посредством предсказания ожидаемого результата экспертизы качества медицинской помощи с помощью технологий машинного обучения

Актуальность темы:

Качество медицинской помощи – это степень, в которой услуги здравоохранения, оказываемые отдельным лицам и группам населения, повышают вероятность достижения желаемых результатов в области здравоохранения и соответствуют профессиональным знаниям, основанным на фактических данных. Это определение качества медицинской помощи распространяется на укрепление здоровья, профилактику, лечение, реабилитацию и паллиативную помощь и исходит из того, что качество медицинской помощи может измеряться и постоянно повышаться благодаря тому, что медицинская помощь оказывается на основе фактических данных и учетом потребностей и предпочтений пользователей услуг – пациентов, семей и местных сообществ.

За последние десятилетия сформулированы многочисленные элементы качества. На сегодняшний день существует четкий консенсус в отношении того, что качественные услуги здравоохранения должны быть:

- Эффективными путем предоставления основанных на фактических данных услуг здравоохранения нуждающимся в них лицам;

- безопасными путем непричинения вреда лицам, для которых предназначается медицинская помощь;

- ориентированными на потребности людей путем оказания медицинской помощи в соответствии с индивидуальными предпочтениями, потребностями и ценностями в рамках служб здравоохранения, организованных с учетом потребностей людей;

- своевременными путем сокращения времени ожидания, а иногда и задержек, причиняющих вред как тем, кто получает медицинскую помощь, так и тем, кто ее оказывает;

- справедливыми путем обеспечения одинакового качества медицинской помощи независимо от возраста, пола, гендера, расы, этнической принадлежности, географического местоположения, религии, социально-экономического статуса, а также языковой или политической принадлежности;

- интегрированными путем предоставления медицинской помощи, скоординированной на всех уровнях и среди провайдеров услуг и обеспечивающей доступ ко всему комплексу услуг здравоохранения на протяжении всей жизни;

- действенными путем получения максимальных преимуществ от имеющихся ресурсов и недопущения потерь времени и средств.

Многие из вышеупомянутых проявлений качества тесно связаны с этическими принципами.

Качественные услуги здравоохранения являются результатом как более широкого контекста систем здравоохранения, так и действий провайдеров услуг здравоохранения и отдельных лиц, работающих в системе. Всемирная Организация Здравоохранения (ВОЗ), Организация Экономического Сотрудничества и Развития (ОЭСР) и Всемирный банк предложили ряд мер для основных субъектов – правительств, систем здравоохранения, граждан и пациентов, а также работников здравоохранения, – которые должны работать сообща для достижения цели по оказанию качественных услуг здравоохранения на местах [1]:

1) Национальная стратегическая направленность на качество. Национальная политика и стратегии, направленные на повышение качества медицинской помощи, обеспечивают прочную основу для повышения качества в рамках всей системы здравоохранения и должны быть тесно увязаны с более широкой национальной политикой и планированием в области здравоохранения. В основе национальной политики и стратегии обеспечения качества лежит прагматический пакет мер, отражающих действия, которые необходимо осуществить в рамках всей системы здравоохранения для формирования системной среды, уменьшения вреда, улучшения клинической помощи и взаимодействия с пациентами, семьями и местными сообществами. Разработан примерный перечень качественных мероприятий по каждой из этих областей для рассмотрения странами [1].

2) Качество в рамках всей системы здравоохранения. Помимо благоприятной политической среды для предоставления качественных услуг здравоохранения требуются надлежащее управление; квалифицированные и компетентные трудовые ресурсы здравоохранения, которые получают поддержку и преданы своему делу; механизмы финансирования, обеспечивающие возможности и поддержку для оказания качественной медицинской помощи; информационные системы, которые обеспечивают постоянный контроль и обучение в целях улучшения медицинской помощи; лекарственные средства, устройства и технологии, которые имеются в наличии, безопасны и надлежащим образом регулируются; и доступные и хорошо оборудованные медицинские учреждения [1].

3) Высококачественная первичная медико-санитарная помощь [1]. Первичная медико-санитарная помощь играет решающую роль в обеспечении всеобщего охвата высококачественными услугами здравоохранения. Для реализации трех взаимосвязанных компонентов первичной медико-санитарной помощи – расширения прав и возможностей людей и участия местных сообществ; осуществления многосекторальных действий в интересах здоровья; и создания служб здравоохранения, в рамках которых первоочередное внимание уделяется оказанию высококачественной первичной медико-санитарной помощи и выполнению основных функций общественного здравоохранения, – необходимо тщательно учитывать аспекты качества [1].

4) Мониторинг и оценка. Качество необходимо постоянно оценивать и контролировать в целях стимулирования улучшений. Это зависит от точных, своевременных и имеющих практическую ценность данных. Интеграция глобальных и национальных усилий в области оценки имеет решающее значение для обеспечения того, чтобы страны собирали значимые данные и использовали их для преобразования и совершенствования своих систем предоставления услуг здравоохранения. Так, например, в этом контексте важное значение имеют системы отчетности о побочных эффектах и обучения [1].

5) Обмен опытом и обучение. Помимо оценки ключевых показателей улучшения имеется явная необходимость в сборе и распространении информации об извлеченных уроках и опыте в области обеспечения качества в странах и между странами в целях взаимного обогащения и активизации процесса обучения. Такие знания, поступающие из стран с разными уровнями систем здравоохранения, должны распространяться на глобальном уровне и в рамках местных систем здравоохранения.

6) Качественные устойчивые системы здравоохранения. Качество и устойчивость тесно связаны между собой. Для того чтобы системы здравоохранения были устойчивыми, необходимы качественные услуги здравоохранения, которые предоставляются до возникновения чрезвычайных ситуаций в области здравоохранения, поддерживаются в течение таких ситуаций и совершенствуются после их завершения. Качественные услуги здравоохранения играют ключевую роль во время чрезвычайных ситуаций и служат связующим звеном между местными сообществами и системой здравоохранения [1].

Наконец, во всем мире необходимо коренным образом изменить систему предоставления услуг, с тем чтобы качественные услуги предоставлялись с чувством сострадания и с учетом потребностей людей и местных сообществ, поскольку предоставление услуг с чувством сострадания, как было продемонстрировано, улучшает результаты в ряде ситуаций.

Рассмотрим, как реализуются представленные ранее условия для достижения цели по оказанию качественных услуг здравоохранения на местах на примере Российской Федерации.

В Российской Федерации (РФ) мониторинг и оценка качества медицинской помощи реализуется в том числе за счет деятельности страховых медицинских организаций. Согласно Федеральному закону от 21.11.2011 №323-ФЗ «Об основах охраны здоровья граждан в Российской Федерации» органами, организациями государственной, муниципальной и частной систем здравоохранения осуществляется внутренний контроль качества и безопасности медицинской деятельности в порядке, установленном руководителями указанных органов, организаций. Федеральный закон от 29.11.2010 №326-ФЗ «Об обязательном медицинском страховании в Российской Федерации» возлагает на страховые медицинские организации и территориальные фонды обязательного медицинского страхования функцию контроля. Она реализуется в форме медико-экономического контроля, медико-экономической экспертизы, экспертизы качества медицинской помощи, защиты прав застрахованных граждан. В Послании к Федеральному собранию РФ от 12 декабря 2013 г. Президент РФ Путин В.В. заявил о необходимости создания системы независимой оценки качества работы организаций социальной сферы, повышения заинтересованности страховых медицинских организаций в том, чтобы медицинские учреждения предоставляли качественные услуги, а страховая медицинская организация (СМО) и территориальные фонды обязательного медицинского страхования (ОМС) следили за этим и давали финансовую оценку их работе.

Таким образом, мониторинг и оценка качества медицинской помощи является важным компонентном обеспечения качества медицинской помощи. Экспертиза качества медицинской помощи проводится в целях выявления нарушений при оказании медицинской помощи, в том числе оценки своевременности ее оказания, правильности выбора методов профилактики, диагностики, лечения и реабилитации, степени достижения запланированного результата. Значительная часть экспертиз качества медицинской помощи проводится силами страховых медицинских организаций.

Существуют разнообразные виды экспертизы качества медицинской помощи – плановая, внеплановая, тематическая, целевые и это не исчерпывающий список. Значительную долю в отношении среди них занимает экспертиза методом случайной выборки. Плановая экспертиза качества медицинской помощи методом случайной выборки проводится для оценки характера, частоты и причин нарушений прав застрахованных лиц на своевременное получение медицинской помощи установленного территориальной программой обязательного медицинского страхования объема и качества, в том числе обусловленных неправильным выполнением медицинских технологий, повлекших ухудшение состояния здоровья застрахованного лица, дополнительный риск неблагоприятных последствий для его здоровья, неоптимальное расходование ресурсов медицинской организации, неудовлетворенность медицинской помощью застрахованных лиц. Предполагается, что случайный и беспристрастный отбор случаев для экспертизы качества позволяет контролировать общий ожидаемый уровень медицинской помощи. Однако существует проблема мощности страховой медицинской организации – проверить все случаи невозможно, проверять их автоматически на данном этапе развития технологии не представляется возможным, так как во-первых, юридически первичная медицинская документация обладает приоритетом над прочими формами хранения информации о пациенте что означает необходимость анализа физических носителей информации, во-вторых, согласно определению ВОЗ проявления качества медицинской помощи тесно связаны с этическими принципами.

Таким образом, для оптимизации мощностей работы страховой медицинской организации было бы идеально если бы в случайную выборку всегда попадали только случаи медицинской помощи, характеризующиеся наличием дефектов качества медицинской помощи, если среди анализируемых случаев вообще существуют случаи медицинской помощи, характеризующиеся наличием дефектов качества медицинской помощи и одновременно всегда не попадали случаи, характеризующиеся отсутствием дефектов качества медицинской помощи. Такой подход к выборке мог бы значительно увеличить мощность стартовых медицинских организаций – так согласно записям реестров счетов Тульского территориального фонда ОМС из 184 779 реестровых строк, имеющих внутренний идентификатор «Проведена экспертиза» или «Выявлены ошибки при экспертизе», менее 12% строк содержат внутренний идентификатор «Выявлены ошибки при экспертизе». Если сделать обоснованное предположение, что существуют другие строки, которые могли бы содержать внутренний идентификатор «Выявлены ошибки при экспертизе» в случае проверки, то создание алгоритма с точностью более 50% обеспечивающего отбор случаев медицинской помощи, характеризующихся наличием дефектов качества медицинской помощи позволило бы увеличить мощность работы страховых медицинских организаций в аспекте плановая экспертизы качества медицинской помощи методом случайной выборки сразу на порядок.

**1 Аналитический обзор**

Технологии искусственного интеллекта и машинного обучения широко применяются в здравоохранении во всем мире и в РФ в частности. По оценкам Markets and Markets, в 2024 г. объем глобального рынка оценивается в 20,9 млрд. долл., к 2029 г. он вырастет до 148,4 млрд. долл. Среднегодовой рост рынка составляет 48,1%. В мире около 3 тыс. стартапов предлагают свои продукты и услуги в данной сфере.

Рост интереса к ИИ обусловлен сразу несколькими трендами: появление мощных графических процессоров и рост вычислительной мощности современных компьютеров, развитие облачных вычислений, взрывной рост больших данных. Эти технологии дали возможность выполнять автоматизированное машинное обучение с высокой точностью получаемых моделей, что в свою очередь открыло многочисленные примеры успешной автоматизации процессов и перспектив цифровой трансформации с возможностью сокращения затрат на здравоохранение.

Указом Президента России №490 от 10.10.2019 утверждена Национальная стратегия развития искусственного интеллекта (ИИ) в Российской Федерации на период до 2030 г. Внедрение ИИ в российском здравоохранении является одним из ключевых направлений развития отрасли и в настоящее время существует не менее 66 разнообразных ИИ-системах для медицины и здравоохранения, созданных и продвигаемых на рынке нашей страны.

Условно существующие продукты можно объединить в несколько основных групп:

1) Анализ медицинских изображений и цифровая диагностика

2) Профилактика и лечение состояний, заболеваний и осложнений

3) Прочие направления

Однако на сегодняшний день отсутствуют развитые продукты связанные с проблематикой оценки качества медицинской помощи в целом и фактически не удаляется системное внимание проблеме отбора случаев для экспертизы медицинской помощи методом случайной выборки. Усилия разработчиков в настоящий момент направлены на попытки автоматизации самого процесса экспертизы в том числе с применением технологий машинного обучения. Так в 2019-2020 годах ТФОМС Московской области подключил систему мониторинга качества лечения и обнаружения нарушений при оказании медицинской помощи на базе искусственного интеллекта (ИИ). На создание сервиса, получившего название «Эксперт», ушло 41,6 млн рублей, а его разработкой и внедрением занимались ярославская компания «Пиклаб» и Национальный центр информатизации (НЦИ, входит в ГК «Ростех»).

Эксперты страховых компаний и специалисты ТФОМС Московской области используют систему с мая 2019 года для выявления случаев оказания медицинской помощи, имеющих признаки нарушений или ошибок в составлении документации, расходовании средств ОМС и лечении пациентов. Как утверждают в госкорпорации, «Эксперт» – это нейросеть, которая по алгоритму выдает экспертам подсказки для точечной проверки несоответствия медицинских услуг, предъявленных к оплате, записям в первичной медицинской и учетно-отчетной документации.

Кроме того, система анализирует медицинскую карту, историю болезни и делает вывод о правильном варианте лечения. По данным компании, с появлением автоматизированной системы мониторинга количество выявляемых ошибок в месяц выросло в три раза. До этого все данные медицинских учреждений отслеживались фондом и страховыми экспертами вручную.

Несмотря на то что достижения ТФОМС Московской области впечатляют, юридически процесс оценки качества медицинской помощи посредством систем искусственного интеллекта в настоящее время не является принятой или даже развивающейся практикой. Напротив – усилия федерального министерства здравоохранения РФ и федерального фонда ОМС в настоящее время направлены на увеличение компетентности врачей-экспертов СМО и расширение их штата для повышения мощности СМО в сфере контроля качества медицинской помощи. Сверх того, следует заметить, что, хотя в рамках медико-экономической экспертизы и происходит работа с первичной медицинской документацией, объем и значение этой работы совершенно несопоставимы с работой с первичной медицинской документацией в рамках экспертизы качества медицинской помощи. И даже когда ресурсы на разработку нейросетевой модели способной подобную работу осуществлять, что само по себе сложно и предполагает ансамбль взаимосвязанных нейросетей, проблемой экспертизы качества медицинской помощи с помощью искусственного интеллекта остается юридическая зыбкость обнаруживаемых дефектов. В практической деятельности СМО на сегодняшний день обнаружение подобной нейросетевой моделью дефектов в рамках автоматизированной экспертизы качества медицинской помощи приведет прежде всего к перепроверке случая лечения силами врачей-экспертов.

Таким образом, с практической точки зрения на сегодняшний день исчезающе тонка разница между отбором случаев на экспертизу с помощью технологий машинного обучения и проведением экспертизы качества медицинской помощи с помощью технологий машинного обучения. При этом задача отбора случаев с помощью технологий машинного обучения несопоставимо проще в реализации и при достаточной точности нейросетевой модели не уступает в практической целесообразности. Сверх того, важно заметить, что отбор случаев для экспертизы качества медицинской помощи методом случайной выборки с помощью приложения основанного на нейросетевой модели не противоречит заложенным в основу практики экспертизы качества медицинской помощи методом случайной выборки антикоррупционным принципам справедливости и беспристрастности, а напротив – поддерживает их – так как никакие указания на то какие именно следует отбирать случаи на проверку не существуют в виде явных правил в самой программе, а представляют собой только множество рассчитанных весов нейронных слоев.

Исходя из вышесказанного можно заключить что тема отбора случаев для проведения экспертизы качества медицинской помощи методом случайной выборки посредством технологий машинного обучения является актуальной, современной и значимой. Одновременно эта тема развита недостаточно – на рынке отсутствует разнообразие готовых решений и одновременно в литературе и исследованиях не прослеживается попытки системного подхода к ее решению. Таким образом создание приложения для отбора случаев стационарного лечения на экспертизу качества медицинской помощи посредством предсказания ожидаемого результата экспертизы качества медицинской помощи с помощью технологий машинного обучения является актуальным, перспективным и практически ориентированным научным исследованием.

**2 Материалы и методики данной работы**

**2.1 Материалы данной работы**

Обучающая выборка была сформирована на основе реестров оплаченных счетов лечения в условиях дневного и круглосуточного стационара территориального фонда ОМС Тульской области за период 01.01.2023 – 01.12.2023. Был написан SQL-запрос к базе данных для отбора интересующей меня информации и получения обезличенной выборки. В связи с тем, что территориальный реестр оплаченных счетов содержит персональную информацию, в том числе сведения составляющие врачебную тайну, не считается возможным интегрировать какие-либо функции запросов к базе данных реестра счетов или приводить описания connection string.

**2.2 Методики данной работы**

Как было описано в пункте «Введение» основной целью работы является создание приложения для отбора случаев стационарного лечения на экспертизу качества медицинской помощи посредством предсказания ожидаемого результата экспертизы качества медицинской помощи с помощью технологий машинного обучения. Это приложение и является основной методикой данной работы. Рассмотрим принцип и методы работы данного приложения подробнее.

**2.2.1 Общая информация о приложении**

1. Общий функционал приложения:

1)По запросу конечного пользователя приложение получает на вход структурированный файл данных реестра счетов, содержащий обезличенную информацию о оплаченных случаях реестра счетов в исследуемом периоде;

2) В качестве результата пользователь получает файл, содержащий идентификаторы законченных случаев и присвоенный по итогам работы программы класс законченного случая на основе обезличенной информации о оплаченных случаях реестра счетов;

3) Предсказание выполняется в рамках двух классов: ожидается выявление дефектов качества оказания медицинской помощи по итогам ЭКМП; ожидается выявление отсутствия дефектов качества оказания медицинской помощи по итогам ЭКМП.

2. Описание нефункциональных требований:

1) Безопасность. Поскольку данные реестра оплаченных счетов, с которыми работают территориальные фонды и страховые медицинские организации представляют собой одновременно персональные данные и предмет врачебной тайны, сама нейросетевая модель и приложение не должно содержать в себе никаких потенциальных угроз раскрытия персональных данных в том числе конечному пользователю.

2) Требования к клиентскому программному обеспечению. Минимальные системные требования - ПК или мобильное устройство способное использовать веб версию Телеграма или настольный клиент:

Минимальные требования для ПК.

○ ОС: Windows XP/7/8/10;

○ Процессор: Pentium II 300 МГц;

○ Оперативная память: 64 Мб;

○ Видеокарта: 4 Мб;

○ Свободное место на жёстком диске: 850 Мб;

○ Разрешение экрана: 1024\*768;

○ Клавиатура;

○ Мышь;

Поддерживаемые языки: Русский, Английский, Китайский.

3) Требования к эргономике. Приложение должно обеспечивать быструю и корректную работу. В то же время изображения, пользовательский интерфейс не должны быть слишком резкими, пестрыми и яркими, дабы не вызвать проблем со здоровьем у людей, страдающими особенными заболеваниями.

4)Требование к простоте. Приложение должно обеспечивать эффективную работу для конечного пользователя, не обладающего никакими техническими знаниями и навыками, короткой инструкции для конечного пользователя должно быть достаточно для получения от нейросетевой модели ожидаемого предсказания

**2.2.2 Разработка модели классификатора и его выбор**

Классификация - один из разделов машинного обучения, посвященный решению следующей задачи. Имеется множество объектов (ситуаций), разделённых некоторым образом на классы. Задано конечное множество объектов, для которых известно, к каким классам они относятся. Это множество называется обучающей выборкой. Классовая принадлежность остальных объектов не известна. Требуется построить алгоритм, способный классифицировать произвольный объект из исходного множества.

Классифицировать объект - значит, указать номер (или наименование класса), к которому относится данный объект.

Классификация объекта - номер или наименование класса, выдаваемый алгоритмом классификации в результате его применения к данному конкретному объекту.

В математической статистике задачи классификации называются также задачами дискриминантного анализа.

В машинном обучении задача классификации относится к разделу обучения с учителем. Существует также обучение без учителя, когда разделение объектов обучающей выборки на классы не задаётся, и требуется классифицировать объекты только на основе их сходства друг с другом. В этом случае принято говорить о задачах кластеризации или таксономии, и классы называть, соответственно, кластерами или таксонами.

Выбор классификатора и fine tuning:

В контексте машинного обучения классификация относится к обучению с учителем. Такой тип обучения подразумевает, что данные, подаваемые на входы системы, уже размечены, а важная часть признаков уже разделена на отдельные категории или классы.

После того, как данные разделены на обучающий и тестовый наборы (80% к 20%), можно использовать алгоритмы машинного обучения.

Рассматривались следующие алгоритмы:

● "Nearest Neighbors" - Метод k-ближайших соседей (k Nearest Neighbors, или kNN) – популярный алгоритм классификации, который используется в разных типах задач машинного обучения. Наравне с деревом решений это один из самых быстрых подходов к классификации;

● "Linear SVM" - Метод Опорных Векторов или SVM (от англ. Support Vector Machines) — это линейный алгоритм, используемый в задачах классификации и регрессии. Данный алгоритм имеет широкое применение на практике и может решать, как линейные, так и нелинейные задачи;

● "Decision Tree" - Дерево принятия решений (также называют деревом классификации или регрессионным деревом) - средство поддержки принятия решений, использующееся в машинном обучении, анализе данных и статистике. Структура дерева представляет собой «листья» и «ветки». На рёбрах («ветках») дерева решения записаны признаки, от которых зависит целевая функция, в «листьях» записаны значения целевой функции, а в остальных узлах - признаки, по которым различаются случаи. Чтобы классифицировать новый случай, надо спуститься по дереву до листа и выдать соответствующее значение;

● "Random Forest" - алгоритм“Случайный лес”, может использоваться как для задач регрессии, так и для задач классификации;

● "Neural Net" - многослойный персептрон;

● "AdaBoost" - Алгоритм (сокр. от adaptive boosting) — алгоритм машинного обучения, предложенный Йоавом Фройндом (Yoav Freund) и Робертом Шапиром (Robert Schapire). Является мета-алгоритмом, в процессе обучения строит композицию из базовых алгоритмов обучения для улучшения их эффективности. AdaBoost является алгоритмом адаптивного бустинга в том смысле, что каждый следующий классификатор строится по объектам, которые плохо классифицируются предыдущими классификаторами;

● "Naive Bayes" - вероятностный классификатор, основанный на применении теоремы Байеса со строгими (наивными) предположениями о независимости. Как следует из названия, этот алгоритм делает предположение, что все переменные в наборе данных "наивные", т.е. не коррелируют друг с другом;

● RF (random forest) - это множество решающих деревьев. В задаче классификации принимается решение голосованием по большинству. Все деревья строятся независимо по следующей схеме:

- Выбирается подвыборка обучающей выборки размера samplesize и по ней строится дерево (для каждого дерева — своя подвыборка);

- для построения каждого расщепления в дереве просматриваем max\_features случайных признаков (для каждого нового расщепления - свои случайные признаки);

- выбираются наилучшие признаки и расщепление по ним (по заранее заданному критерию). Дерево строится, как правило, до исчерпания выборки (пока в листьях не останутся представители только одного класса), в современных реализациях есть параметры, которые ограничивают высоту дерева, число объектов в листьях и число объектов в подвыборке, при котором проводится расщепление.

где:

N – количество деревьев;

i – счетчик для деревьев;

b – решающее дерево;

x – сгенерированная нами на основе данных выборка.

Чем больше деревьев, тем лучше качество, но время настройки и работы RF также пропорционально увеличиваются. При увеличении n\_estimators качество на обучающей выборке повышается.

**2.2.3 Используемые библиотеки**

1) Pandas

Инструмент анализа и обработки данных с открытым исходным кодом, создан на основе языка программирования Python. Необходим для работы с датасетами.

2) sklearn train\_test\_split

Подмодуль sklearn для разбиения датасета на тренировочный и тестовый.

3) sklearn RandomForestClassifier

Модуль для обучения моделей путем использования алгоритма “Случайный лес”

4) Sklearn metrics

Подмодуль sklearn для оценки качественных характеристик обученных моделей

5) Joblib

Модуль содержащий инструменты пайплайна в Python. Необходим для сохранения обученной нейросетевой модели

6) Easygui

Модуль содержащий набор инструментов, позволяющий реализовать выполнение типовых задач посредством интуитивно понятного для конечного пользователя интерфейса.

7) Tkinter

Модуль содержащий инструменты создания пользовательских форм в Python. Необходим для разработки приложения для конечного пользователя.

8) Auto PY to EXE

Проект с открытым кодом и графическим интерфейсом, созданный для полуавтоматической конвертации .py файлов в .exe файлы. Необходим для разработки приложения для конечного пользователя.

**3 Результаты работы**

Таким образом для достижения поставленной цели необходимо и достаточно реализовать следующие этапы разработки:

1. Получение обучающей выборки;

2. Обработка полученных данных;

3. Создание нейросетевой модели и ее обучение;

4. Экспорт нейросети в отдельное приложение для использования конечным пользователем.

Рассмотрим практическую реализацию каждого из указанных этапов.

**3.1 Получение обучающей выборки**

Обучающая выборка была сформирована на основе реестров оплаченных счетов лечения в условиях дневного и круглосуточного стационара территориального фонда ОМС Тульской области за период 01.01.2023 – 01.12.2023. Был написан SQL-запрос к базе данных для отбора интересующей меня информации и получения обезличенной выборки. В связи с тем, что территориальный реестр оплаченных счетов содержит персональную информацию, в том числе сведения составляющие врачебную тайну, не считается возможным интегрировать какие-либо функции запросов к базе данных реестра счетов или приводить описания connection string.

Код запроса:

select distinct a.kzip1 as ”Id”, ( a.enddate - a.begindate ) as “Days”, a.profilebed as “Prof”, a.maindiagnosis as “MKB1”, ad.ds2 as “MKB2”, p.sex as “Sex”, (a.enddate - p.birthday) as “Age”, to\_char(a.enddate, 'MM') as "Month", mu.keyrecord as “PMU”, a.CRIT as “Crit”, a.resultcode as “Result”, a.paysummamo as “Paysum”, f.expresid as “Exp” from limportal.lim\_o\_anp a left join limportal.lim\_o\_rnp r on r.oldk\_anp\_id = a.id left join limportal.LIM\_ADD\_DIAGNOSIS ad on ad.OLDK\_ANP\_ID = a.id left join limportal.lim\_o\_snp st on a.oldk\_snp\_id = st.id left join limportal.lim\_patient p on p.id = a.idpatient left join limportal.lim\_servismes s on s.oldk\_anp\_id = a.id left join limportal.lbv\_\_t\_mu mu on mu.recid=s.LIM\_SERVISKOD left join F002\_PORTAL f on f.SMOCOD=a.INSURSMO where a.conditid in (0101,0201) and f.expresid in (3,4) and to\_char(a.enddate,'YYYYMM') between 202301 and 202312 and a.statemekrepeat != 4 and st.state >5;

где:

Id – уникальный идентификатор законченного случая лечения в реестре, текстовое поле;

Days – количество койкодней в рамках законченного случая, расчетное поле, целое число;

Prof – код профиля медицинской помощи с которого подан законченный случай, целое число;

MKB1 – код основного диагноза согласно классификации MKB-X, текстовое поле;

MKB2 - код дополнительного диагноза, диагноза осложнения или конкурирующего диагноза согласно классификации MKB-X, текстовое поле;

Sex – пол застрахованного лица, получавшего лечение в рамках законченного случая, целое число, где 1 – мужской пол, 2 – женский пол;

Age – возраст застрахованного лица, получавшего лечение в рамках законченного случая на дату окончания госпитализации, расчетное поле, целое число;

Month – месяц даты окончания госпитализации, расчетное поле, целое число;

Pmu – код простой медицинской услуги, оказанной в рамках законченного случая, текстовое поле;

Crit – дополнительный критерий для отнесения законченного случая дневного и круглосуточного стационара к клинико-статистической группе, текстовое поле;

Result – исход законченного случая, согласно федеральному справочнику исходов, целое число;

Paysum – сумма оплаты, перечисленная в медицинское учреждение оказывающее лечение в рамках законченного случая, двойное с плавающей точкой;

Exp – результат проведения экспертизы качества медицинской помщи, где 3 - проведена экспертиза, 4 – выявлены ошибки в ходе экспертизы, целое число.

В результате был получен рекордсет в 184 778 строк, экспортированный в файл формата .csv средствами среды PL/SQL-developer для дальнейшей обработки. Важно заметить, что в полученной выборке менее 12% строк помечены как «выявлены ошибки в ходе экспертизы», что уже свидетельствует о целесообразности моей работы над нейросетью.

**3.2 Обработка полученных данных**

Полученный рекордсет в виде файла формата .csv далее был назван «Выгрузка.csv» и загружен в файл .ipynb для создания собственно тестовой и обучающей выборки. В файле юпитер-блокнота были проделаны необходимые этапы для подготовки данных к обучению нейросети.

import pandas as pd

data=pd.read\_csv("Выгрузка.csv")

data.keys()

Изначально были импортированы библиотека pandas для оптимизации работы с массивами данных в юпитер-блокноте, в датафрейм data загружена моя выборка и прочитаны ключи датафрейма. Так как предполагается использование стереотипного sql запроса в дальнейшем, я исхожу из стабильности названия столбцов и буду прямо к ним обращаться.

Для моделей RandomForestClassifier нельзя прямо передавать входные данные, которые не могут быть интерпретированы как числа, поэтому для каждого текстового поля нам нужно сопоставить уникальный идентификатор соответствующего значения. Это может быть сделано созданием дополнительных столбцов посредством факторизации по соответствующим исходным столбцам. В выгрузке четыре столбца с текстовыми значениями, таким образом нужно четыре дополнительных столбца с идентификаторами. Факторизация начинает счет с 0, однако я хочу зарезервировать 0 для пустых ячеек, поэтому увеличим результат вычисления на 1.

data["MKB1\_id"]=pd.factorize(data.MKB1)[0]+1

data["MKB2\_id"]=pd.factorize(data.MKB2)[0]+1

data["PMU\_id"]=pd.factorize(data.PMU)[0]+1

data["Crit\_id"]=pd.factorize(data.Crit)[0]+1

Далее в выгрузке есть большое количество пустых ячеек. Это связано с тем, что не все поля выгрузки являются обязательными для заполнения. Нейросетевая модель не должна игнорировать эпизоды отсутствия заполнения таких полей, так что заменим значения этих ячеек на 0.

data.fillna(0, *inplace*=True)

Теперь все готово для создания датафреймов, содержащих обучающую и тестовую выборки для дальнейшего обучения нейросети. Импортируем метод train\_test\_split и создаём датафреймы.

from sklearn.model\_selection import train\_test\_split

X = pd.DataFrame(data, *columns*=[ 'Days', 'Prof', 'MKB1\_id', 'MKB2\_id', 'Sex', 'Age', 'Month',

       'PMU\_id', 'Crit\_id', 'Result', 'Paysum'])

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, *test\_size*=0.3, *random\_state*=42)

Столбец Id не нужен на данном этапе, т.к. результат экспертизы несомненно не зависит от автоматически присвоенного законченному случаю ключа.

Теперь можно приступить к созданию и обучению модели.

**3.3 Создание нейросетевой модели и ее обучение**

В качестве модели нейросети была выбрана модель RandomForestClassifier из библиотеки sklearn. Начнем с автоматического поиска гиперпараметров нейросети с помощью GridSearchCV. Данный способ подбора гиперпараметров был выбран, потому что условия поиска решения поставленной задачи позволяли потратить неопределенное количество времени на перебор гиперпараметров в поисках оптимального варианта.

from sklearn.ensemble import RandomForestClassifier

from sklearn.model\_selection import GridSearchCV

n\_estimators = [600,700]

max\_features = ['sqrt']

max\_depth = [20,30,70,110,150]

min\_samples\_split = [2,3,4,22,23,40]

min\_samples\_leaf = [2,3,4,5,6,7]

bootstrap = [False]

param\_grid = {'n\_estimators': n\_estimators,

               'max\_features': max\_features,

               'max\_depth': max\_depth,

               'min\_samples\_split': min\_samples\_split,

               'min\_samples\_leaf': min\_samples\_leaf,

               'bootstrap': bootstrap}

gs = GridSearchCV(RandomForestClassifier(), param\_grid, *cv* = 3, *verbose* = 1, *n\_jobs*=-1)

gs.fit(X\_train, y\_train)

model = gs.best\_estimator\_

gs.best\_params\_

Оптимальными параметрами по данному поиску оказались следующие:

best={'bootstrap': False,

 'max\_depth': 150,

 'max\_features': 'sqrt',

 'min\_samples\_leaf': 2,

 'min\_samples\_split': 22,

 'n\_estimators': 700}

Импортируем метрики, для оценки точности.

from sklearn import metrics

print("Accuracy:", metrics.accuracy\_score(y\_test, y\_pred\_gs ))

Accuracy: 0.9264711188079517

Аккуратность в ~92% представляется достаточной, чтобы остановиться на данной модели.

Экспорт нейросети в отдельное приложение для использования конечным пользователем.

Заключительный этап практической работы – создание приложения для конечного пользователя на базе обученной нейросети. Сохраним и обучим модель для экспорта.

model = RandomForestClassifier(*bootstrap*= False,

*max\_depth*= 150,

*max\_features*= 'sqrt',

*min\_samples\_leaf*= 2,

*min\_samples\_split*= 22,

*n\_estimators*= 700)

model.fit(X\_train, y\_train)

print("Accuracy:", metrics.accuracy\_score(y\_test, y\_pred\_gs ))

Accuracy: 0.9310170653389617

Для экспорта модели использовалась библиотеку joblib. Параметр степени сжатия - максимальный.

import joblib

joblib.dump(model, "model.joblib", *compress*=9)

Также, так как приложение будет работать с данными извлеченными из реестра с использованием стереотипного sql запроса в дальнейшем, нужно экспортировать идентификаторы, созданные ранее для строковых текстовых столбцов, чтобы обеспечить их однозначное соответствие. Экспорт предпочтителен в .csv файлы.

mkb1\_values.to\_csv("mkb1\_values.csv", *index*=False)

mkb2\_values.to\_csv("mkb2\_values.csv", *index*=False)

crit\_values.to\_csv("crit\_values.csv", *index*=False)

Pmu\_values.to\_csv("Pmu\_values.csv", *index*=False)

Теперь можно начать создание отдельного .py файла. Начинаем, разумеется с экспорта необходимых библиотек. Здесь пригодятся библиотеки для создания окон и элементов пользовательского интерфейса, а также библиотеки для упрощения работы с файловой системой.

import numpy as np

import pandas as pd

from sklearn.model\_selection import train\_test\_split

from sklearn import metrics

from sklearn.ensemble import RandomForestClassifier

import joblib

import easygui

import tkinter as tk

from tkinter import \*

from tkinter import messagebox

Так как приложение крайне специализированное, на данный момент для решения поставленной в работе цели будет достаточно небольшого окна с элементом управления (рисунок 1) на который будет назначен вызов нейросети.

window = Tk()

window.title("Отбор перспектиных случаев на ЭКМП")

window.geometry('400x300')

frame = Frame(

   window,

*padx*=10,

*pady*=10

)

frame.pack(*expand*=True)

cal\_btn = Button(

   frame,

*text*='Рассчитать ИМТ',

*command*=main

)

cal\_btn.grid(*row*=5, *column*=2)

window.mainloop()

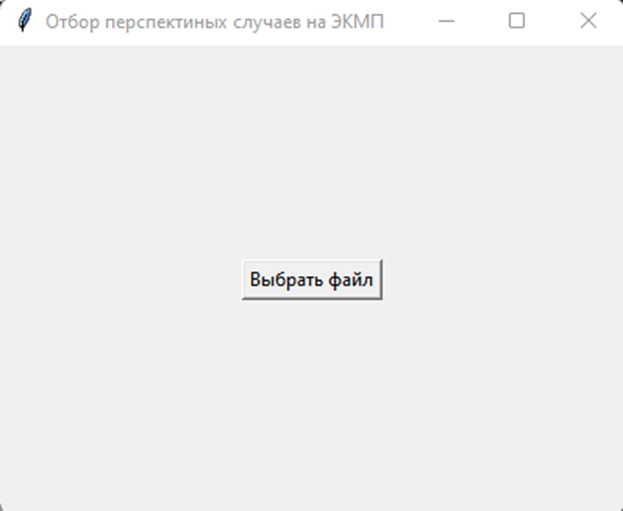


Рисунок 1 – Пример окна приложения

Результат работы кода – лаконичное, но пока не очень полезное окно.

Прежде всего хотелось получить файл с данными, по которым будет делаться предсказание. Пусть пользователь сам его выберет.

input\_file = easygui.fileopenbox()

input\_file=pd.read\_csv(input\_file)

Дальше уже сама программа загружает служебные .csv файлы, нейросетевую модель и делает предсказание. Перед предсказанием нужно добавить запомненные нейросетью идентификаторы для текстовых столбцов из служебных .csv файлов. Далее остается только экспортировать предсказанные значения. Нам очень важно экспортировать предсказания в связке с полем “Id”, что отдел экспертизы знал какие собственно строки реестра им нужно брать.

       mkb1\_values=pd.read\_csv("mkb1\_values.csv")

       mkb2\_values=pd.read\_csv("mkb2\_values.csv")

       crit\_values=pd.read\_csv("crit\_values.csv")

       Pmu\_values=pd.read\_csv("Pmu\_values.csv")

       input\_file=pd.merge(input\_file,mkb1\_values, *on*=('MKB1'), *how* = "left")

       input\_file=pd.merge(input\_file,mkb2\_values, *on*=("MKB2"), *how* = "left")

       input\_file=pd.merge(input\_file,crit\_values, *on*=("Crit"), *how* = "left")

       input\_file=pd.merge(input\_file,Pmu\_values, *on*=("PMU"), *how* = "left")

       input\_file.fillna(0, *inplace*=True)

       X = pd.DataFrame(input\_file, *columns*=[ 'Days', 'Prof', 'MKB1\_id', 'MKB2\_id', 'Sex', 'Age', 'Month',

              'PMU\_id', 'Crit\_id', 'Result', 'Paysum'])

       loaded\_model = joblib.load( "model.joblib")

       input\_file['Exp'] = loaded\_model.predict(X)

       input\_file.to\_csv("Результат.csv", *index*=False)

       window.destroy()

Итоговый код приложения выглядит следующим образом:

import numpy as np

import pandas as pd

from sklearn.model\_selection import train\_test\_split

from sklearn import metrics

from sklearn.ensemble import RandomForestClassifier

import joblib

import easygui

import tkinter as tk

from tkinter import \*

from tkinter import messagebox

*def* main():

       input\_file = easygui.fileopenbox()

       input\_file=pd.read\_csv(input\_file)

       mkb1\_values=pd.read\_csv("mkb1\_values.csv")

       mkb2\_values=pd.read\_csv("mkb2\_values.csv")

       crit\_values=pd.read\_csv("crit\_values.csv")

       Pmu\_values=pd.read\_csv("Pmu\_values.csv")

       input\_file=pd.merge(input\_file,mkb1\_values, *on*=('MKB1'), *how* = "left")

       input\_file=pd.merge(input\_file,mkb2\_values, *on*=("MKB2"), *how* = "left")

       input\_file=pd.merge(input\_file,crit\_values, *on*=("Crit"), *how* = "left")

       input\_file=pd.merge(input\_file,Pmu\_values, *on*=("PMU"), *how* = "left")

       input\_file.fillna(0, *inplace*=True)

       X = pd.DataFrame(input\_file, *columns*=[ 'Days', 'Prof', 'MKB1\_id', 'MKB2\_id', 'Sex', 'Age', 'Month',

              'PMU\_id', 'Crit\_id', 'Result', 'Paysum'])

       loaded\_model = joblib.load( "model.joblib")

       input\_file['Exp'] = loaded\_model.predict(X)

       input\_file.to\_csv("Результат.csv", *index*=False)

       window.destroy()

window = Tk()

window.title("Отбор перспектиных случаев на ЭКМП")

window.geometry('400x300')

frame = Frame(

   window,

*padx*=10,

*pady*=10

)

frame.pack(*expand*=True)

cal\_btn = Button(

   frame,

*text*='Рассчитать ИМТ',

*command*=main

)

cal\_btn.grid(*row*=5, *column*=2)

window.mainloop()

Однако, следует заметить, что это не то чтобы приложение – это скорее отдельный .py файл только для запуска которого требуются особые условия. Это не удовлетворяет поставленным задачам и цели работы и не позволяет работать конечному пользователю.

Поэтому заключительный этап практической реализации данной работы - библиотека auto-py-to-exe. Данная библиотека позволяет экспортировать из среды разработки данный .py файл как исполняемый .exe файл с набором необходимых модулей и библиотек в отдельных папках. После экспорта остается только скопировать в каталог с .exe файлом .csv файлы и нейросетевую модель.

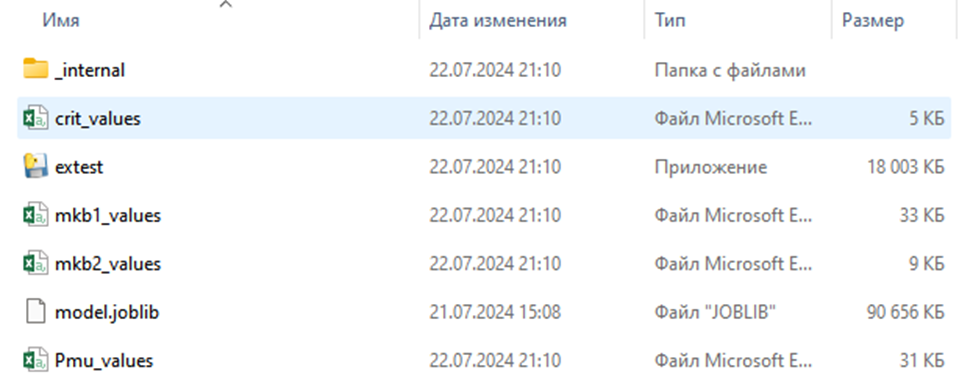


Рисунок 2 – Директория расположения приложения

В ходе данной работы был получен исполняемый файл, выполняющий поставленную задачу - отбор случаев стационарного лечения на экспертизу качества медицинской помощи посредством предсказания ожидаемого результата экспертизы качества медицинской помощи с помощью технологий машинного обучения. Теперь папку с исполняемым файлом, .csv файлами, нейросетевой моделью и модулями, и библиотеками можно упаковать в архив и передать для конечного пользователя с инструкциями по работе.

**3.4 Результат создания приложения**

В ходе данной работы был получен исполняемый файл, выполняющий поставленную задачу - отбор случаев стационарного лечения на экспертизу качества медицинской помощи посредством предсказания ожидаемого результата экспертизы качества медицинской помощи с помощью технологий машинного обучения. Теперь папку с исполняемым файлом, .csv файлами, нейросетевой моделью и модулями, и библиотеками можно упаковать в архив и передать для конечного пользователя с инструкциями по работе.

Таким образом, с помощью языка программирования Python возможно создание автономного приложения, которое посредством передачи данных предварительно обученной нейросети будет с достаточной точностью предсказывать обнаружение по результатам экспертизы ошибок в случаях законченного лечения для представленных из реестра оплаченных счетов идентификаторов законченных случаев. При этом требования к квалификации конечного пользователя минимальны, а стабильность работы приложения и защищенность персональных данных на оптимальном уровне – приложение или работает правильно, или ничего не делает, а персональных данных просто нет.

Искусственный интеллект служит для обеспечения должного уровня качества медицинской помощи и стоит на защите прав застрахованных в системе ОМС граждан РФ.

**Заключение**

Создание приложения для отбора случаев стационарного лечения на экспертизу качества медицинской помощи посредством предсказания ожидаемого результата экспертизы качества медицинской помощи с помощью технологий машинного обучения:

1. Оптимизация отбора случаев на плановую экспертизу качества медицинской помощи методом случайной выборки это путь увеличения мощности работы страховых медицинских организаций, который позволит увеличить качество оказываемой медицинской помощи в целом.

2. Создан sql запрос для получения обучающей выборки для нейросетевой модели

3. Обучена модель классификации для предсказания случаев медицинской помощи, характеризующихся наличием дефектов качества медицинской помощи

4. Создан sql запрос для получения выборки для последующего предсказания приложением и отбора случаев на плановую экспертизу качества медицинской помощи методом случайной выборки

5. Разработано приложения для предсказания в предложенной выборке случаев медицинской помощи, характеризующихся наличием дефектов качества медицинской помощи

**Список использованных источников**

1. Всемирная Организация Здравоохранения [Электронный ресурс]: Информационные бюллетени. Качественные услуги здравоохранения. URL: https://www.who.int/ru/news-room/fact-sheets/detail/quality-health-services (дата обращения 27.07.24).

2. Webiomed [Электронный ресурс]: Обзор Российских систем искусственного интеллекта для здравоохранения. URL: https://webiomed.ru/blog/obzor-rossiiskikh-sistem-iskusstvennogo-intellekta-dlia-zdravookhraneniia (дата обращения 27.07.24).

3. Кашерининов Юрий Робертович. «Особенности проведения экспертизы качества медицинской помощи в условиях многопрофильного федерального центра» Артериальная гипертензия, vol. 15, no. 2, 2009, pp. 170-180.

4. Журавлев Ю. И., Рязанов В. В., Сенько О. В. «Распознавание». Математические методы. Программная система. Практические применения. — М.: Фазис, 2006. ISBN 5-7036-0108-8.

5. Goodfellow I. Deep Learning / I. Goodfellow, Y. Bengio,A. Courville. — Cambridge: The MIT Press, 2016. — 800 p

6. Программирование на Python, том 2, 4-е издание. / Марк Лутц. — ДМК Пресс 2016, С. 72-–79.