МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ КАЗАХСТАН
SATBAYEV UNIVERSITY

Институт автоматики и информационных технологий

УДК 004.852

На правах рукописи

Абжанова Камила Ниязбековна

МАГИСТЕРСКАЯ ДИССЕРТАЦИЯ

На соискание академической степени магистра

Название диссертации	Выявление сердечных аномалий с помощью методов машинного обучения
Направление подготовки	7M06102 - Machine Learning & Data science

Научный руководитель
PhD, ассис.-проф
_____Б.С. Омаров
«18» ____04____ 2022 г.

Оппонент
канд. тех. наук, ст.препод-ль
_____ И.М. Уалиева
«30» ____05____ 2022 г.

Нормоконтроль
PhD
_____А.Т.Ахмедиярова
«03» ____05____ 2022 г.

ДОПУЩЕН К ЗАЩИТЕ
Заведующий кафедрой ПИ
канд. ф.–м.наук, ассис.-проф
_____
А.Н.Молдагулова
«01» ____06____ 2022 г.

Алматы 2022

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ КАЗАХСТАН

SATBAYEV UNIVERSITY

Институт автоматики и информационных технологий

Кафедра Программная инженерия
Специальность: 7М06102 - Machine Learning & Data science

УТВЕРЖДАЮ
Заведующий кафедрой ПИ
канд. ф.–м.наук, ассис.-проф
_____
А.Н.Молдагулова
«01» 06 2022 г.

ЗАДАНИЕ
на выполнение магистерской диссертации

магистранту, обучающемуся Абжановой Камиле Ниязбековне

Тема диссертации: «Выявление сердечных аномалий с помощью методов машинного обучения»

Срок сдачи законченной диссертации «15» апреля

Исходные данные к магистерской диссертации. Дан анализ моделей и методов машинного обучения. Объектами исследования выступают прогнозирование риска заболеваемости сердечно-сосудистыми заболеваниями. Предметами исследования являются оценка рисков и неопределенности в прогнозных показателях заболеваемости.

Перечень подлежащих разработке в магистерской диссертации вопросов или краткое содержание магистерской диссертации: а) определение требований – анализ потребностей, целей, задач и назначения разработки; б) исследование особенностей современных систем управления здравоохранения; в) сравнительный анализ существующих методов г) разработка системы прогнозирования выявления сердечных аномалий с помощью методов машинного обучения.

Рекомендуемая основная литература

1. Андреас Мюллер, Сара Гвидо «Введение в машинное обучение с помощью Python». 2020. DOI: 10459039/02023-1

2. С.А.Шумский. Машинный интеллект. Очерки по теории машинного обучения и искусственного интеллекта. М., РИОР, 2019. DOI: 10.29039/02011-1

3. А.Мюллер, С.Гвидо. Введение в машинное обучение с помощью Python. Руководство для специалистов по работе с данными. Вильямс, 2017, 480 с. ISBN 978-5-99089-108-1

ГРАФИК

подготовки магистерской диссертации

Наименование разделов, перечень разрабатываемых вопросов	Сроки представления научному руководителю	Примечание
Раздел 1. Машинное обучение и возможность его применения в здравохранении	30.10.2020 г.	Выполнено
Раздел 2. Методы и средства машинного обучения для выявления сердечных аномалий	31.03.2021 г.	Выполнено
Раздел 3. Разработка системы прогнозирования выявления сердечных аномалий с помощью методов машинного обучения	28.02.2022 г.	Выполнено

Консультации по проекту с указанием относящихся к ним разделов проекта

Раздел	Консультант, (уч. степень, звание)	Сроки	Подпись
Нормоконтроль	А.Т.Ахмедиярова, PhD	26.04.2022-07.05.2022	
Антиплагиат	Әубәкіров Б.С., Ассистент	09.05.2022-14.05.2022	

Дата выдачи задания "10" 01 2022 г.

Заведующий кафедрой _____ А.Н. Молдагулова

Научный руководитель _____ Б.С. Омаров

Задание принял к исполнению магистрант _____ К.Н. Абжанова

Дата "25" 04 2022 г.

СОДЕРЖАНИЕ

Введение

Актуальность. Болезнь сердца является одним из сложных заболеваний, и во всем мире от этого заболевания страдает множество людей. Своевременное и эффективное выявление заболеваний сердца играет ключевую роль в здравоохранении, особенно в области кардиологии. Сердечно-сосудистые заболевания (ССЗ) являются основной причиной смерти во всем мире, унося примерно 17,9 миллиона жизней ежегодно, что составляет 32% от всех смертей. ССЗ представляют собой группу заболеваний сердца и кровеносных сосудов и включают ишемическую болезнь сердца, цереброваскулярные заболевания, ревматические заболевания сердца и другие состояния. Более четырех из пяти смертей от сердечно-сосудистых заболеваний связаны с сердечными приступами и инсультами, и одна треть этих смертей происходит преждевременно среди людей моложе 70 лет.

День ото дня число случаев сердечных заболеваний растет быстрыми темпами, и очень важно заранее прогнозировать любые такие заболевания. Эта диагностика является сложной задачей, т.е. она должна быть выполнена точно и эффективно.

Если ничего не предпринимать, ожидается, что к 2030 году общее число погибших в мире возрастет до 22 миллионов. Бляшки на стенках артерий могут препятствовать кровотоку, что приводит к сердечному приступу или инсульту. Заболевания сердца вызываются различными факторами риска, такими как отсутствие физической активности, нездоровое питание, активное употребление алкоголя и табака [1, 2]. Вышеуказанные факторы снижаются за счет правильного образа жизни, а именно: уменьшения потребления соли в рационе, потребления овощей и фруктов, регулярных занятий физической активностью, отказа от употребления алкоголя и табака, что способствует минимизации риска сердечных заболеваний [3]. Решением этих проблем является использование сбора историй болезни пациентов из различных медицинских центров и больниц. Для получения результатов и запроса другого мнения опытного врача используется система поддержки принятия решений. *Эта методика позволяет избежать ненужных тестов для диагностики, тем самым экономя деньги и время [4, 5]. В последнее время система управления больницами использовалась для управления медицинскими данными или данными пациентов, что означает, что эти системы производят больше данных. Для прогнозирования сердечных заболеваний был разработан DSS с использованием алгоритма NB (Naive Bayes). Веб-приложение создано для получения приложения и пользовательского ввода, и оно извлекает ключевые характеристики, относящиеся к сердечным заболеваниям, из исторической базы данных (набор данных Кливленда) [1, 7].

На начальных стадиях сердечной недостаточности (ХСН) запускается большее количество нейрогормональных регуляторных механизмов. В краткосрочной перспективе эти компенсаторные механизмы могут вызвать последствия диеты с высоким содержанием жиров (ДВСЖ), приводящие к выраженной дисфункции желудочков, одышке при нагрузке, периферическим

отекам, ремоделированию легких и сердца, что может вызвать постоянные изменения постнагрузки и преднагрузки. Пациенту предоставляется больше вариантов лечения с ДВСЖ, включая изменение образа жизни и имплантацию или медицинские устройства, такие как дефибриллятор или кардиостимулятор.

Главной задачей является обеспечение последующего наблюдения за этой популяцией, учитывая, что госпитализация в связи с острой декомпенсацией ДВСЖ является основной причиной расходов на здравоохранение. Статистика и исследования показывают, что сердечные заболевания являются наиболее серьезной проблемой, с которой сталкиваются люди, особенно с ДВСЖ [3, 9]. При различных заболеваниях ранняя диагностика и обнаружение сердечных заболеваний является первым шагом в уходе и лечении. ДВСЖ в настоящее время является новым расстройством для таких заболеваний, как гипертония, бессонница и болезни сердца, среди прочих. Выявление ДВСЖ на ЭКГ завершается выявлением вариаций длительности сердечных сокращений от временного интервала от 1 волны выпадения сердечного цикла (PQRST) до следующей волны PQRST. Для раннего выявления ИБС новым и перспективным неинвазивным диагностическим методом является МКГ (магнитокардиография). В то время как МКГ меньше подвержена влиянию контактной интерференции электрод-кожа по сравнению с ЭКГ, она очень чувствительна к вихревым токам и тангенциальным воздействиям через ткань ишемизированного сердца. Несмотря на высокое качество сигнала, интерпретация МКГ занимает много времени, сильно зависит от опыта перевода и имеет ограниченное применение в клиниках. В результате клиницисты получат пользу от автономной системы, которая может обнаруживать и локализовать ишемию на ранней стадии [10].

Раннее выявление сердечно-сосудистых заболеваний у лиц с улучшенной диагностикой и у лиц с высоким риском с использованием прогностической модели может быть рекомендовано в целом для снижения уровня смертности, а принятие решений улучшается для дальнейшего лечения и профилактики. В системе поддержки принятия клинических решений (СППКР или CDSS) модель прогнозирования реализована и используется для поддержки клиницистов в оценке риска сердечных заболеваний, а также предоставляются соответствующие методы лечения для управления дальнейшим риском. Кроме того, в многочисленных исследованиях также сообщается, что внедрение CDSS может улучшить качество принятия решений, принятие клинических решений и профилактическую помощь соответственно [3, 11]. Ишемическая болезнь сердца (далее ИБС), является ведущей причиной смерти взрослых старше 35 лет в разных странах. За тот же промежуток времени он стал самой большой причиной смерти в Китае. При снижении притока крови к сердцу вследствие стеноза коронарных артерий возникает ИБС. Повреждение миокарда может иметь серьезные последствия, включая желудочковую аритмию или даже внезапную сердечную смерть вследствие инфаркта миокарда.

Выявление лиц с самым высоким риском сердечно-сосудистых заболеваний и обеспечение их надлежащего лечения может предотвратить

преждевременную смерть. Доступ к лекарствам от неинфекционных заболеваний и основным медицинским технологиям во всех учреждениях первичной медико-санитарной помощи имеет важное значение для обеспечения того, чтобы нуждающиеся получали лечение и консультирование. Но наиболее эффективным в снижении сердечно-сосудистых заболеваний является применение цифровых IT технологий, а именно применение методов машинного обучения, которые помогут наиболее точно спрогнозировать сердечно-сосудистого риска за счет нелинейных взаимосвязей их более глубокой настройки между факторами риска и результатами заболеваний. В медицине для управления рисками сердечно-сосудистых заболеваний используются рискометры – шкалы, полученные в результате длительных наблюдений и исследований. Но практика показало их ограниченность в прогнозировании рисков ССЗ.

Цель диссертационной работы – разработка системы прогнозирования сердечных аномалий с помощью методов машинного обучения.

Для реализации этой цели были поставлены следующие **задачи:**

– рассмотрение теоретических основ машинного обучения;

– сравнительный анализ методов и средств Machine learning для выявления сердечных аномалий;

– разработка системы прогнозирования и принятия решений сердечно-сосудистых заболеваний, чтобы выяснить будет ли у пациента диагностировано сердечное заболевание или нет используя историю болезни пациента.

Новизна Подготовлена система прогнозирования сердечных заболеваний, чтобы предсказать, будет ли у пациента диагностировано сердечное заболевание или нет используя историю болезни пациента. Были использованы различные алгоритмы машинного обучения, такие как логистическая регрессия и KNN, для прогнозирования и классификации пациента с сердечными заболеваниями. Весьма полезный подход был использован для регулирования того, как модель может быть использована для повышения точности прогнозирования сердечного приступа у любого человека. Сила предложенной модели была достаточно удовлетворительной и позволяла прогнозировать наличие сердечных заболеваний у конкретного человека с помощью KNN и логистической Регрессии, которая показала хорошую точность. Таким образом, с помощью данной модели было снято довольно значительное количество измерений давления при определении вероятности того, что классификатор правильно и точно идентифицирует заболевание сердца. Данная система прогнозирования сердечных заболеваний улучшает качество медицинской помощи и снижает ее стоимость.

Методы выполнения исследования по теме диссертации: Диссертационной работе предложена система диагностики рисков сердечно-сосудистых заболеваний, основанная на методах машинного обучения. Система разработана на основе алгоритмов классификации, включающих в себя логистическую регрессию, K-ближайшего соседа. Рассмотрены и проанализированы методы и алгоритмы машинного обучения для решения проблемы выбора признаков. Алгоритмы выбора признаков используются для

выбора признаков, чтобы повысить точность классификации и сократить время выполнения системы классификации. Кроме того, метод перекрестной проверки с исключением одного субъекта использовался для изучения передового опыта оценки моделей и настройки гиперпараметров. Метрики измерения производительности используются для оценки производительности классификаторов. Производительность классификаторов была проверена на выбранных функциях, выбранных алгоритмами выбора функций. Реализовано программное обеспечение для выявления сердечных аномалий на базе методов машинного обучения

Результаты Основные результаты диссертационной работы докладывались на семинарах и заседаниях кафедры «Программная инженерия» Института Автоматики и информационных технологий Satbayev University, на конфереции «Сатпаевские чтения», которая проходила в Satbayev University в 2022 году.

Публикации результатов исследования

- **Статья, опубликованная в трудах:**

Международная научно-практическая конференция «Сатпаевские чтения – 2022. Тренды современных научных исследований» на тему «Выявление сердечных аномалий с помощью методов машинного обучения», объем которой составляет 8 страниц (см. Приложение)

Сердечно-сосудистые заболевания часто встречаются и являются основной причиной внезапной смерти в настоящее время. Эта болезнь атакует людей мгновенно. Большинство людей не знают о симптомах сердечно-сосудистых заболеваний. Своевременное внимание и правильная диагностика заболеваний сердца снизят уровень смертности. Интеллектуальный анализ медицинских данных заключается в изучении скрытых закономерностей в наборах данных. Контролируемые алгоритмы используются для раннего прогнозирования сердечных заболеваний. Ближайший сосед (КБС в англоязычных изданиях, на русском метод «к ближайших соседей или КБС) — это широко используемый алгоритм ленивой классификации. КБС — самый популярный, эффективный и действенный алгоритм, используемый для распознавания образов. Наборы медицинских данных содержат большое количество функций. Производительность классификатора будет снижена, если наборы данных содержат зашумленные признаки. Для решения этой проблемы предлагается выбор подмножества признаков. Выбор функции повысит точность и сократит время работы. Оптимизация роя частиц (ОРЧ) — это метод эволюционных вычислений (ЕС), используемый для выбора признаков. ОРЧ не требует больших вычислительных затрат и быстро сходится. В этой диссертации исследуется возможность применения КБС и ОРЧ для прогнозирования сердечных заболеваний. Экспериментальные результаты показывают, что алгоритм работает очень хорошо со 100% точностью при использовании ОРЧ в качестве выбора признаков.

Исследовательская работа в основном посвящена тому, у какого пациента больше шансов заболеть сердечным заболеванием, основываясь на различных медицинских признаках. Подготовлена система прогнозирования сердечных

заболеваний, чтобы предсказать, будет ли у пациента диагностировано сердечное заболевание или нет используя историю болезни пациента. Были использованы различные алгоритмы машинного обучения, такие как логистическая регрессия и КБС, для прогнозирования и классификации пациента с сердечными заболеваниями. Весьма полезный подход был использован для регулирования того, как модель может быть использована для повышения точности прогнозирования сердечного приступа у любого человека. Сила предложенной модели была достаточно удовлетворительной и позволяла прогнозировать наличие сердечных заболеваний у конкретного человека с помощью КБС и логистической Регрессии, которая показала хорошую точность. Таким образом, с помощью данной модели было снято довольно значительное количество измерений давления при определении вероятности того, что классификатор правильно и точно идентифицирует заболевание сердца. Данная система прогнозирования сердечных заболеваний улучшает качество медицинской помощи и снижает ее стоимость. Этот проект дает значительные знания, которые могут помочь прогнозировать пациентов с заболеваниями сердца. Он реализован в формате .ipynb.

1 Машинное обучение и возможность его применения в здравохранении

1.1 Текущее состояние и сферы использования машинного обучения

Ишемическая болезнь сердца (ИБС) представляет собой закупорку коронарных артерий с такими симптомами, как стенокардия, боль в груди и сердечные приступы. Артерии снабжают кровью сердечную мышцу. ИБС является ведущей причиной смерти во многих странах. На его долю приходится более 30% всех смертей. Если ничего не предпринимать, ожидается, что к 2030 году общее число погибших в мире возрастет до 22 миллионов. Бляшки на стенках артерий могут препятствовать кровотоку, что приводит к сердечному приступу или инсульту. Заболевания сердца вызываются различными факторами риска, такими как отсутствие физической активности, нездоровое питание, активное употребление алкоголя и табака [1, 2]. * Вышеуказанные факторы снижаются за счет правильного образа жизни, а именно: уменьшения потребления соли в

рационе, потребления овощей и фруктов, регулярных занятий физической активностью, отказа от употребления алкоголя и табака, что способствует минимизации риска сердечных заболеваний [3]. *Решением этих проблем является использование сбора историй болезни пациентов из различных медицинских центров и больниц. Для получения результатов и запроса другого мнения опытного врача используется система поддержки принятия решений. *Эта методика позволяет избежать ненужных тестов для диагностики, тем самым экономя деньги и время [4, 5-8]. В последнее время система управления больницами использовалась для управления медицинскими данными или данными пациентов, что означает, что

эти системы производят больше данных. Для прогнозирования сердечных заболеваний был разработан DSS с использованием алгоритма NB (Naive Bayes). Веб-приложение создано для получения приложения и пользовательского ввода, и оно извлекает ключевые характеристики, относящиеся к сердечным заболеваниям, из исторической базы данных (набор данных Кливленда) [1, 7].

На начальных стадиях сердечной недостаточности (ХСН) запускается большее количество нейрогормональных регуляторных механизмов. В краткосрочной перспективе эти компенсаторные механизмы могут вызвать последствия HFD, приводящие к выраженной дисфункции желудочков, одышке при нагрузке, периферическим отекам, ремоделированию легких и сердца, что может вызвать постоянные изменения постнагрузки и преднагрузки. Пациенту предоставляется больше вариантов лечения с HFD, включая изменение образа жизни и имплантацию или медицинские устройства, такие как дефибриллятор или кардиостимулятор.

*Главной задачей является обеспечение последующего наблюдения за этой популяцией, учитывая, что госпитализация в связи с острой декомпенсацией HFD является основной причиной расходов на здравоохранение. Статистика и исследования показывают, что сердечные

заболевания являются наиболее серьезной проблемой, с которой сталкиваются люди, особенно с HFD [3, 9]. При различных заболеваниях ранняя диагностика и обнаружение сердечных заболеваний является первым шагом в уходе и лечении.

*е HFD в настоящее время является новым расстройством для таких заболеваний, как гипертония, бессонница и болезни сердца, среди прочих.

*д Выявление HFD на ЭКГ завершается выявлением вариаций длительности сердечных сокращений от временного интервала от 1 волны PQRST до следующей волны PQRST. Для раннего выявления ИБС новым и перспективным неинвазивным диагностическим методом является МКГ (магнитокардиография). В то время как МКГ меньше подвержена влиянию контактной интерференции электрод-кожа по сравнению с ЭКГ, она очень чувствительна к вихревым токам и тангенциальным воздействиям через ткань ишемизированного сердца. Несмотря на высокое качество сигнала, интерпретация МКГ занимает много времени, сильно зависит от опыта перевода и имеет ограниченное применение в клиниках. В результате клиницисты получат пользу от автономной системы, которая может обнаруживать и локализовать ишемию на ранней стадии [10].

Раннее выявление сердечно-сосудистых заболеваний у лиц с улучшенной диагностикой и у лиц с высоким риском с использованием прогностической модели может быть рекомендовано в целом для снижения уровня смертности, а принятие решений улучшается для дальнейшего лечения и профилактики. В CDSS модель прогнозирования реализована и используется для поддержки клиницистов в оценке риска сердечных заболеваний, а также предоставляются соответствующие методы лечения для управления дальнейшим риском. Кроме того, в многочисленных исследованиях также сообщается, что внедрение CDSS может улучшить качество принятия решений, принятие клинических решений и профилактическую помощь соответственно [3, 11]. ИБС является ведущей причиной смерти взрослых старше 35 лет в разных странах. За тот же промежуток времени он стал самой большой причиной смерти в Китае. При снижении притока крови к сердцу вследствие стеноза коронарных артерий возникает ИБС. Повреждение миокарда может иметь серьезные последствия, включая желудочковую аритмию или даже внезапную сердечную смерть вследствие инфаркта миокарда. В Индии ежегодно насчитывается около 3 миллионов пациентов с сердцем, и ежегодно проводится 2 миллиона операций на открытом сердце [1]. ИБС является ведущей причиной смертности, ежегодно унося почти 17,3 миллиона человек. Причиной тому является курение, высокий уровень холестерина, сахарный диабет. Раннее прогнозирование сердечно-сосудистых заболеваний необходимо для снижения уровня смертности. Интеллектуальный анализ данных обеспечивает ориентированный на пользователя подход к извлечению новых и непокрытых шаблонов в наборе данных. Интеллектуальный анализ данных предназначен для извлечения полезных знаний из медицинских данных для медицинской диагностики [2]. Интеллектуальный анализ данных широко применяется в медицинской сфере. Интеллектуальный анализ медицинских данных используется для вывода

диагностических правил и помогает врачам сделать процесс диагностики более точным [3]. K-ближайший сосед является наиболее широко используемым алгоритмом ленивой классификации, поскольку он уменьшает ошибку ошибочной классификации [4]. Выбор подмножества признаков (FSS) широко используется в интеллектуальном анализе данных и машинном обучении. FSS — это метод уменьшения размерности, используемый для повышения точности [1]. Оптимизация роя частиц является эффективным методом EC, используемым для выбора признаков [1]. ОРЧ быстро сходится и не требует больших вычислительных затрат.

Машинное обучение может использоваться для диагностики, обнаружения и прогнозирования многих заболеваний в медицинской отрасли.

В этой диссертации исследуется применение КБС и ОРЧ для прогнозирования сердечных заболеваний. ОРЧ используется как мера выбора признаков.

1.2 Анализ проведенных исследований

Из-за увеличения числа смертей от сердечных заболеваний организациям здравоохранения нужны инновационные подходы, чтобы знать, контролировать и управлять своими действиями, чтобы повысить качество обслуживания и помочь врачам и персоналу выполнять свои задачи в нужное время. , в хорошем состоянии. Одним из инновационных подходов является Интернет вещей (IoT), который в последние годы широко используется в сердечно-сосудистых областях для измерения, мониторинга и сбора данных. В этом контексте система, предложенная Сафой и Пандианом [9], фокусируется на определении уровня стресса в четырех классифицированных категориях путем получения физиологических параметров от наблюдаемого человека. Датчики оксиметра протестированы с использованием модели классификатора, обученной на уже сохраненном наборе кардиологических данных. Результаты показали, что качество классификации K-соседей значительно выше, чем у методов SVM и дерева решений. Балакришнанд и др. [10] создали решение с использованием интеллектуального датчика частоты сердечных сокращений на кончиках пальцев, который удаленно и непрерывно контролирует артериальное давление и частоту сердечных сокращений пациентов, требуется безопасность устройств IoT в этой классификации. Для защиты этой системы был предложен легкий метод шифрования. Классификация аритмичных сердечных сокращений осуществляется с помощью линейной регрессии. Заман и др. [11] предложили метод прогнозирования состояния сердца, основанный на IoT и машинном обучении. Данные, собранные с человеческого тела, были нормализованы перед тем, как их использовали алгоритмы машинного обучения для расчета и прогнозирования общего состояния сердца пациента, результаты оказались вполне удовлетворительными. Основываясь на Интернете вещей (IoT) и биосенсорах, Ислам и др. [12] предлагают недорогую систему здравоохранения для пациентов с сердечно-сосудистыми заболеваниями в Бангладеш. Это

позволит врачам удаленно наблюдать за состоянием пациента с сердечным заболеванием в больнице или дома.

Интеллектуальный анализ данных — это междисциплинарная область, широко используемая в клинической области, такой как прогнозирование сердечных заболеваний. Исследователи разработали различные методы прогнозирования сердца с помощью интеллектуального анализа данных. Криттанавонг и др. [13] представили метаанализ предсказания машинного обучения при сердечно-сосудистых заболеваниях. Сионтис и др. [14] обобщают текущее и будущее состояние ЭКГ (электрокардиограмма) с усилением ИИ в выявлении сердечно-сосудистых заболеваний в группах риска, обсуждают его значение для принятия клинических решений у пациентов с сердечно-сосудистыми заболеваниями и оценивают его потенциальные ограничения. в их обзоре. Линда и др.

[15] разработали новую систему поддержки принятия клинических решений для назначения упражнений пациентам, страдающим сердечными заболеваниями. В своем предварительном анализе они обнаружили, что клиницисты не знают, как составить рецепт упражнений для пациентов с множественными факторами риска сердечно-сосудистых заболеваний. Предоставленная система представляет собой простой в использовании, управляемый и эффективный по времени подход к пациентам, основанный на фактических данных.

Техника извлечения ассоциативных правил также использовалась во многих работах для поиска частых наборов элементов среди больших наборов данных пациентов для диагностики наличия сердечных заболеваний. Али и др. [16] представили трехэтапный подход PB-FARM для оценки факторов риска, связанных с заболеваниями. Он также был реализован в наборе данных Z-Alizadeh Sani для оценки факторов, влияющих на заболеваемость этим заболеванием. Результаты показали сильную корреляцию между уровнем заболеваемости ИБС и пожилым возрастом и типичной болью в груди. Джесмин и др. [17] представили эксперимент по извлечению правил для данных о сердечных заболеваниях с использованием различных алгоритмов извлечения правил. Из набора здоровых правил следует, что принадлежность к женскому полу является одним из факторов здорового состояния сердца, у них больше шансов избавиться от ишемической болезни сердца, чем у мужчин.

М. Анбараси и соавт. использовали генетические алгоритмы в [18]; оптимизировать размер информации и найти достаточное подмножество среди значений атрибутов пациентов для прогнозирования состояния сердца. Оптимизационные преимущества генетического алгоритма были использованы в [19]; где Питер и Сомасундарам реализовали гибридную систему для инициализации весов нейронной сети, которая поддерживала группу факторов риска, таких как гипертония, высокий уровень холестерина, ожирение и т. д. В другом исследовании [20] Амин использовал многоуровневый нейро-нечеткий подход, чтобы обеспечить ужасно низкий уровень ошибок при проведении анализа возникновения сердечных заболеваний.

Методы вменения данных использовались для заполнения недостающих данных на этапе предварительной обработки. Худрифи и Бахадж [21] сравнивают алгоритмы с различными показателями производительности, используя машинное обучение. Каждый алгоритм работал лучше в одних ситуациях и хуже в других. K-NN, RF и многослойный персептрон (MLP) с гибридной оптимизацией роя частиц (PSO) и оптимизацией колонии муравьев (ACO) — модели, которые, вероятно, будут работать лучше всего в наборе данных, использованном в этом исследовании. Сетиаван и др. [22] внедрили ANN с Rough Set Theory (RST), (ANN-RST) уменьшением атрибутов для прогнозирования реальных отсутствующих значений атрибутов в данных о сердечных заболеваниях, этот метод хорошо работает по сравнению с другими методами, такими как ANN, кусочно-линейная сеть. -Выбор ортонормированных признаков методом наименьших квадратов

Пурушоттам и др. разделил набор данных о сердечных заболеваниях (HDD) на два раздела и оценил производительность в каждом разделе, внедрив правила принятия решений для модифицированного набора данных [1]. Это может быть выбор признаков и извлечение признаков [1]. Ришаб Саксена и др. реализовали алгоритмы KNN и дерева решений, используя HDD (набор данных о сердечных заболеваниях), в котором уменьшаются нерелевантные атрибуты, и тем самым исследуя проблемы с точки зрения временной сложности и точности [2]. Фэн-Джанг-Джан и др. рассказывает, как наивный байесовский классификатор используется для различных проблемных областей для классификации, объясняя вероятностные вычисления, используемые в наивном байесовском классификаторе [3]. Мариам Бенлларк и др. предложил две модели дерева решений, такие как Очень быстрое дерево решений, которое связано с границей Хеффдинга, которая определяет отсутствие выборок, необходимых для наилучшего разделения в узле, Чрезвычайно быстрое дерево решений является улучшением VFDT, где оно проверяет разделение в узле. Они реализованы в Cleveland HDD Dataset, и результаты показывают, что EFDT имеет большую точность, чем VFDT [4]. S.Manikandan et.al разработали веб-интерфейс с 81,25% для прогнозирования ранних сердечных заболеваний с использованием наивного байесовского классификатора [1]. Шадаб Адам Паттекари и др. использовали наивный байесовский метод для диагностики риска сердечно-сосудистых заболеваний. Эта система обеспечивает эффективные результаты для прогнозирования сердечных заболеваний [1]. Ali Haghpanah Jahromi et.al использовали гауссовский наивный байесовский метод для классификации 12 наборов данных UCI и сравнили их с некоторыми сильными классификаторами. Результаты показывают, что производительность улучшается по сравнению с соответствующей работой других классификаторов [2].

Чайтрали и др. внедрил и сравнил нейронные сети, дерево решений, наивный байесовский алгоритм для прогнозирования сердечной недостаточности, рассмотрев набор данных о сердечной недостаточности и добавив к нему еще два атрибута, таких как ожирение и курение, и, таким образом, оценил производительность трех алгоритмов. Результаты показывают,

что нейронная сеть дает наилучшую точность для набора данных с новыми добавленными атрибутами [3]. Канак Саксена и др. действительно работал над обнаружением данных и внедрил эффективную систему прогнозирования, которая генерирует правила принятия решений для классификации записи из набора данных Кливленда по сердечно-сосудистым заболеваниям. Результаты показывают, что эта система более точна, чем другие алгоритмы машинного обучения для рассматриваемого набора данных [3]. Методы обработки данных используются для получения значимых данных из информации. Методы классификации, использованные в Эксперименте для анализа результатов и точности [10]. Аджит Соланки со своей исследовательской группой провели обзор нескольких алгоритмов интеллектуальной обработки данных (ИОД) и сделали анализ потенциал алгоритмов для прогнозирования ССЗ. Результаты показывают, что из всех методов ИОД алгоритмы классификации лучше всего подходят для прогнозирования сердечных заболеваний, а многослойный персептрон достиг наибольшей точности [11]. Д-р Т.Картикеян и др. проанализировали некоторые алгоритмы классификации и свели в таблицу результаты по определенным критериям, таким как точность, скорость, надежность, масштабируемость, интерпретируемость. Говорят, что у каждого алгоритма есть свои плюсы и минусы, выбор правильного алгоритма зависит от набора данных, времени, продолжительности [12].

Большая часть предыдущей работы над набором данных UCI дает ценные результаты, и все еще делается много улучшений в методах классификации для повышения точности прогнозирования. Но во время нашего опроса мы заметили, что набор данных под названием «Набор данных о сердечной недостаточности», над которым не проводилась предыдущая работа. Итак, наша мотивация для этой статьи — реализовать эффективную и точную диагностику сердечных заболеваний с использованием алгоритмов машинного обучения в наборе данных о сердечной недостаточности.

Прогнозирование сердечных заболеваний с помощью нейронной сети было предложено Dangare et al. в [2]. Выбор признаков используется для прогнозирования заболевания. Их метод получил точность 92,5% для 13 функций и 100% точность для 15 функций. Улучшение на 7,5% после отказа от 2 функций с 15 по 13.

Джаббар и др. предложил метод, использующий ассоциативную классификацию и выбор подмножества признаков для оценки риска заболевания [2]. Авторы использовали прирост информации, симметричную неопределенность и генетический алгоритм в качестве мер отбора признаков. Их метод получил точность 95% при выборе гибридных признаков. Набор данных о сердечных заболеваниях, собранный с 11 функциями для экспериментального анализа.

Предлагается диагностика заболеваний сердца с помощью нечетких методов [3]. Авторы классифицировали пациентов на основе характеристик, полученных в терапевтической области. Авторы использовали нечеткий и КБС и достигли точности 97%.

Обнаружение болезней сердца на основе нечеткой логики предложено в [3]. Авторы рассматривали 6 параметров для своих экспериментов. Их подход достиг точности 92%. Этот метод дает меньшую точность по сравнению с [3]. Нечеткий подход с использованием КБС дал 97%. Дискретизация и другие фильтры могут повысить производительность алгоритма. В [10] авторы предложили прогнозирование заболеваний сердца с помощью генетических нейронных сетей. Эксперименты проводились на наборе данных Американской кардиологической ассоциации. Их подход показал точность 96,2%.

Масете и др. [11] предложили модель, использующую дерево решений для прогнозирования сердечных заболеваний. Авторы сравнили свой подход с другими подходами к классификации. RepTree и J48 достигли точности 99,07%.

Оценка факторов риска ишемической болезни сердца была предложена karaolis et al. [12]. Авторы исследовали 2 типа факторов риска, а именно модифицируемые и немодифицируемые. Было собрано 528 проб и проведен анализ данных с использованием C4.5. Наивысшая точность, полученная их моделью, составила 75% для моделей ЧКВ и АКШ. Авторы использовали классификатор C4.5 без мер по отбору признаков. Точность, полученная этим подходом, меньше по сравнению с другими подходами.

Диагностика сердечных заболеваний с использованием деревьев регрессии была предложена Амиром [13]. Авторы собрали 116 наборов данных звуковых сигналов сердца и применили дерево регрессии. Их модель предлагается для классификации данных фонокардиограммы (ФКГ). Авторы рассчитали коэффициент капюшона для классификации болезни. Их метод получил точность 99%. В 2014 году авторы [14] предложили основу для прогнозирования коронарной болезни с использованием многослойного персептрона. Их метод использует 13 клинических элементов в качестве входных данных и достиг точности 98%. Литература, упомянутая в этой связанной работе, не использовала эффективные меры выбора признаков для повышения точности. Авторы использовали слабые классификаторы для прогнозирования заболевания. В этой диссертации мы интегрировали ОРЧ с классификатором КБС для получения эффективных результатов.

Ришаб Саксена и др. реализовали алгоритмы KNN и дерева решений, используя HDD (набор данных о сердечных заболеваниях), в котором уменьшаются нерелевантные атрибуты, и тем самым исследуя проблемы с точки зрения временной сложности и точности [2]. Фэн-Джанг-Джан и др. рассказывает, как наивный байесовский классификатор используется для различных проблемных областей для классификации, объясняя вероятностные вычисления, используемые в наивном байесовском классификаторе [3]. Мариам Бенлларк и др. предложил две модели дерева решений, такие как Очень быстрое дерево решений, которое связано с границей Хеффдинга, которая определяет отсутствие выборок, необходимых для наилучшего разделения в узле, Чрезвычайно быстрое дерево решений является улучшением VFDT, где оно проверяет разделение в узле. Они реализованы в Cleveland HDD Dataset, и результаты показывают, что EFDT имеет большую точность, чем VFDT [4]. S.Manikandan et.al разработали веб-интерфейс с 81,25% для прогнозирования

ранних сердечных заболеваний с использованием наивного байесовского классификатора [1]. Шадаб Адам Паттекари и др. использовали наивный байесовский метод для диагностики риска сердечно-сосудистых заболеваний. Эта система обеспечивает эффективные результаты для прогнозирования сердечных заболеваний [1]. Ali Haghpanah Jahromi et.al использовали гауссовский наивный байесовский метод для классификации 12 наборов данных UCI и сравнили их с некоторыми сильными классификаторами. Результаты показывают, что производительность улучшается по сравнению с соответствующей работой других классификаторов [2].

Чайтрали и др. внедрил и сравнил нейронные сети, дерево решений, наивный байесовский алгоритм для прогнозирования сердечной недостаточности, рассмотрев набор данных о сердечной недостаточности и добавив к нему еще два атрибута, таких как ожирение и курение, и, таким образом, оценил производительность трех алгоритмов. Результаты показывают, что нейронная сеть дает наилучшую точность для набора данных с новыми добавленными атрибутами [3]. Канак Саксена и др. действительно работал над обнаружением данных и внедрил эффективную систему прогнозирования, которая генерирует правила принятия решений для классификации записи из набора данных Кливленда по сердечно-сосудистым заболеваниям. Результаты показывают, что эта система более точна, чем другие алгоритмы машинного обучения для рассматриваемого набора данных [3]. Методы обработки данных используются для получения значимых данных из информации. Методы классификации, использованные в Эксперименте для анализа группа результатов и точности [10]. Аджит Соланки и его научно-исследовательская провели обзор некоторых алгоритмов интеллектуального анализа данных и проанализировали потенциал алгоритмов для прогнозирования сердечных заболеваний. Результаты показывают, что из всех методов интеллектуального анализа данных алгоритмы классификации лучше всего подходят для прогнозирования сердечных заболеваний, а многослойный персептрон достиг наибольшей точности [11]. Д-р Т.Картикеян и др. проанализировали некоторые алгоритмы классификации и свели в таблицу результаты по определенным критериям, таким как точность, скорость, надежность, масштабируемость, интерпретируемость. Говорят, что у каждого алгоритма есть свои плюсы и минусы, выбор правильного алгоритма зависит от набора данных, времени, продолжительности [12].

1.3 Выводы по первому разделу

Сердечно-сосудистые заболевания являются одной из серьезных проблем в современном мире и одной из ведущих причин многих смертей во всем мире. Недавнее развитие приложений машинного обучения (МО) демонстрирует, что с помощью электрокардиограммы (ЭКГ) и данных пациентов можно выявлять сердечные заболевания на ранней стадии. Тем не менее, как ЭКГ, так и данные пациентов часто несбалансированы, что в конечном итоге затрудняет непредвзятую работу традиционного машинного обучения. За прошедшие годы

многие исследователи и практики предложили несколько решений на уровне данных и алгоритмов. Чтобы обеспечить более широкий взгляд на существующую литературу, в этом исследовании используется подход систематического обзора литературы (SLR), чтобы выявить проблемы, связанные с несбалансированными данными в прогнозах сердечных заболеваний. До этого, мы провели метаанализ с использованием 40 ссылочной литературы, полученной из авторитетных журналов в период с 2012 г. по 15 ноября 2021 г. Для углубленного анализа было рассмотрено и изучено 40 ссылочной литературы с учетом следующих факторов: тип заболевания сердца, алгоритмы , приложения и решения. Анализ показал, что текущие подходы сталкиваются с различными открытыми проблемами/проблемами при работе с несбалансированными данными, что в конечном итоге препятствует их практическому применению и функциональности.

2. Методы и средства машинного обучения для выявления сердечных аномалий

2.1 Алгоритмы классификации машинного обучения

Классификация является одним из наиболее важных аспектов контролируемого обучения .

В этом разделе мы обсудим различные алгоритмы классификации, такие как логистическая регрессия , наивный байесовский алгоритм , деревья решений, Оптимизация роя частиц, случайные леса и многие другие. Мы рассмотрим каждое из свойств классификации алгоритма и то, как они работают.

Машинное обучение (МО) — это обширная междисциплинарная область, основанная на концепциях компьютерных наук, статистики, когнитивных наук, инженерии, теории оптимизации и многих других дисциплин математики и естественных наук [1]. Существует множество приложений для машинного обучения, но наиболее важным из них является интеллектуальный анализ данных [2]. Машинное обучение в основном можно разделить на две широкие категории, включая машинное обучение с учителем и машинное обучение без учителя.

Неконтролируемое машинное обучение используется для получения выводов из наборов данных, состоящих из входных данных без помеченных ответов [3], или мы можем сказать, что при неконтролируемом обучении желаемый результат не дается. Методы контролируемого машинного обучения пытаются выяснить взаимосвязь между входными атрибутами (независимыми переменными) и целевым атрибутом (зависимой переменной) [4]. Методы под наблюдением можно разделить на две основные категории; классификация и регрессия. В регрессии выходная переменная принимает непрерывные значения, в то время как в классификации выходная переменная принимает метки классов [1].

Классификация — это подход к интеллектуальному анализу данных (машинному обучению), который используется для прогнозирования членства в группе для экземпляров данных [1]. Хотя существует множество доступных методов машинного обучения, классификация является наиболее широко используемым методом [2]. Классификация — любимая задача в машинном обучении, особенно в будущем планировании и открытии знаний.

Классификация классифицируется как одна из важнейших проблем, изучаемых исследователями в области машинного обучения и интеллектуального анализа данных [3]. Общая модель контролируемого обучения (методы классификации) показана на рисунке **1**.

Хотя классификация является хорошо известным методом машинного обучения, она страдает от таких проблем, как обработка отсутствующих данных. Отсутствующие значения в наборе данных могут вызвать проблемы как на этапах обучения, так и на этапах классификации. Некоторые из возможных причин отсутствия данных представлены в [3]: Невнесение записи из-за неправильного понимания, признание данных неактуальными на момент

ввода, удаление данных из-за отклонения от других документированных данных и неисправности оборудования.

Проблема отсутствия данных может быть решена с помощью таких подходов [10], как; Специалисты по анализу данных могут игнорировать пропущенные данные, заменять целые пропущенные значения отдельной глобальной константой, заменять пропущенное значение его средним значением признаков для данного класса, вручную наблюдать выборки с пропущенными значениями и вставлять возможное или вероятное значение. В этой работе мы сосредоточимся только на некоторых избранных методах классификации.

2.2. Классификация машинного обучения

Обнаружение аномалий связан с проблемой обнаружения, когда наблюдаемое значение значительно отличается от прогнозируемого.

Был проведен поиск литературы с использованием баз данных, включая IEEE xplore, google Scholar, Science Direct и некоторые соответствующие веб-страницы, написанные на английском языке. Ключевые слова, используемые для поиска литературы, включают в себя; Машинное обучение, интеллектуальный анализ данных, классификация, обзор классификации, приложения классификации и алгоритмы классификации. Эти ключевые слова использовались по отдельности и в комбинации для первоначального сбора исследовательского материала. В этот обзор были включены только те статьи, которые содержат соответствующие данные о применении методов классификации, проблемах и решениях. Трудно предоставить исчерпывающий обзор всех методов классификации контролируемого машинного обучения в одной статье, поэтому мы сосредоточились только на часто используемых методах классификации, включая дерево решений (ID3 и C4.5), байесовскую сеть, K-ближайших соседей и машины опорных векторов. Применение различных методов классификации представлено в рисунке 1. Дерево решений обеспечивает легко понятный метод моделирования, а также упрощает процесс классификации [12]. Дерево решений — это прозрачный механизм, который позволяет пользователям легко следовать древовидной структуре, чтобы увидеть, как принимается решение [13]. В этом разделе обсуждалась основная философия методов дерева решений с их сильными сторонами, ограничениями и приложениями.

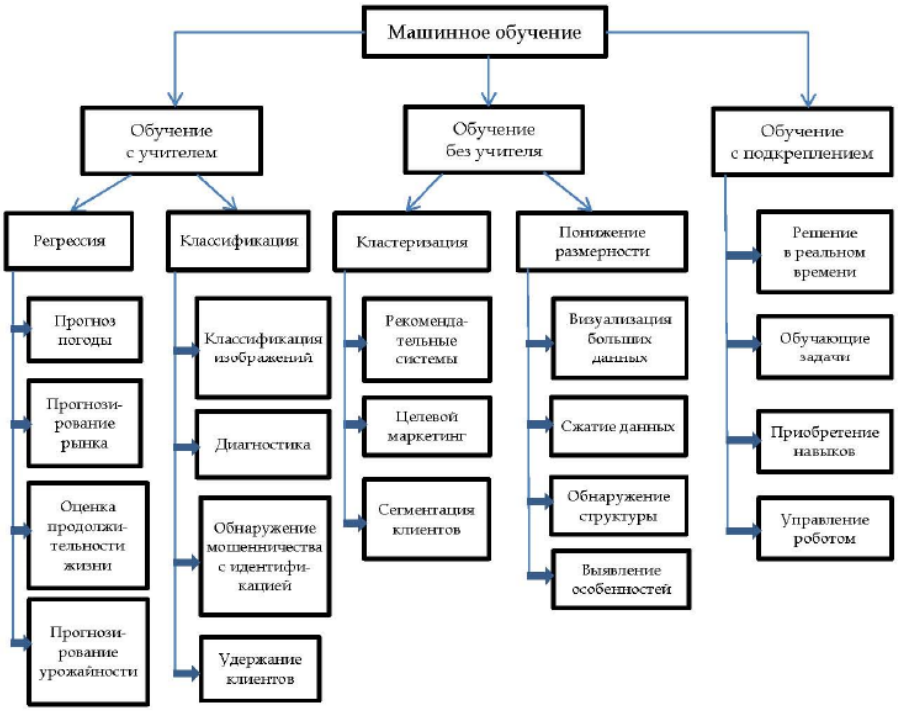


Рисунок 1. Классификация МО

2.2.1 Дерево решений

Основной целью дерева решений является создание модели, которая вычисляет значение требуемой переменной на основе многочисленных входных переменных [1]. Обычно все алгоритмы дерева решений строятся в два этапа:

(i) рост дерева; в котором обучающий набор, основанный на локальных оптимальных критериях, рекурсивно разбивается до тех пор, пока большая часть записей, принадлежащих разделу, не будет иметь одинаковую метку класса [14] (ii) обрезка дерева; в котором размер дерева уменьшен, что упрощает понимание [15]. В этом разделе мы сосредоточимся на алгоритме дерева решений ID3 и C4.5.

(ii) Алгоритм дерева решений ID3 (Iterative Dichotomiser 3) был представлен в 1986 г. [16, 17]. Это один из широко используемых алгоритмов в области интеллектуального анализа данных и машинного обучения благодаря своей эффективности и простоте [16]. Алгоритм ID3 основан на получении информации. Некоторые сильные и слабые стороны дерева решений ID3 представлены в [18], включая сильные стороны; легко понять, и в окончательном решении рассматривается весь учебный пример, включая слабые стороны; без обратного поиска, невозможности обработки отсутствующих значений и глобальной оптимизации.

C4.5 — известный алгоритм построения деревьев решений. Это расширение алгоритма ID3 и минимизирует свои недостатки вызванный ID3. На этапе обрезки C4.5 пытается устранить неудобные ветви, заменяя их конечными узлами, возвращаясь к дереву после его создания [19]. Сильные стороны C4.5 заключаются в обработке обучающих данных с отсутствующими

значениями признаков, работе как с дискретными, так и с непрерывными функциями, а также в обеспечении возможности как предварительной, так и последующей обрезки [18, 20]. Слабые стороны включают в себя; не подходит для небольшого набора данных [18] и большого времени обработки по сравнению с другими деревьями решений.

2.2.2. Байесовские сети

Байесовская сеть (BN) относится к графической модели вероятностных ассоциаций между набором переменных [21]. Структура BN S состоит из направленного ациклического графа (DAG), а узлы в S находятся во взаимно однозначной связи с функциями X. Дуги иллюстрируют неожиданные столкновения между узлами, в то время как нехватка возможных дуг в S кодирует условные свободы [2]. Обычно задачи обучения байесовской сети можно разделить на две подзадачи; (а) изучение сетевой структуры DAG, (б) определение параметров.

Одна из проблем с классификатором байесовских сетей заключается в том, что он обычно требует дискретизации непрерывных атрибутов. Процесс преобразования непрерывного атрибута в дискретный атрибут привел к проблемам классификации [22, 23]. Эти проблемы могут включать в себя шум, недостающую информацию и сознание изменения атрибутов в сторону переменных класса [24]. Другой метод байесовского сетевого классификатора, в котором непрерывный атрибут не преобразуется в дискретный атрибут, требует оценки условной плотности атрибута [23].

Чтобы преодолеть проблему условной оценки плотности атрибутов, в [24] использовалась функция ядра Гаусса с устойчивыми ограничениями для оценки плотности атрибутов. Затем был проведен эксперимент, выполненные на наборе данных, предоставленном в репозитории машинного обучения UCI, показывают, что непрерывные атрибуты обеспечивают лучшую точность классификации по сравнению с другими методами с использованием функции ядра Гаусса в классификаторах байесовской сети.

Некоторые из преимуществ байесовской сети, представленные в [25], включают (i) свойства гладкости; незначительные изменения в модели байесовской сети не влияют на работу системы (ii) Гибкая применимость; идентичная модель байесовской сети может использоваться для решения проблем как регрессии, так и классификации (iii) обработки отсутствующих данных; Байесовская сеть имеет возможность восполнять недостающие данные, ассимилируя все возможности недостающих значений.

2.2.3. K-ближайший сосед

K-ближайший сосед (КБС) — это простой, ленивый и непараметрический классификатор. КБС предпочтительнее, когда все функции непрерывны. КБС также называется рассуждениями на основе прецедентов и используется во многих приложениях, таких как распознавание образов, статистическая

оценка. Классификация получается путем идентификации ближайшего соседа для определения класса неизвестного образца. КБС предпочтительнее других алгоритмов классификации из-за его высокой скорости сходимости и простоты [17]. **На рисунке 2** показана классификация ближайших соседей. Классификация КБС имеет два этапа

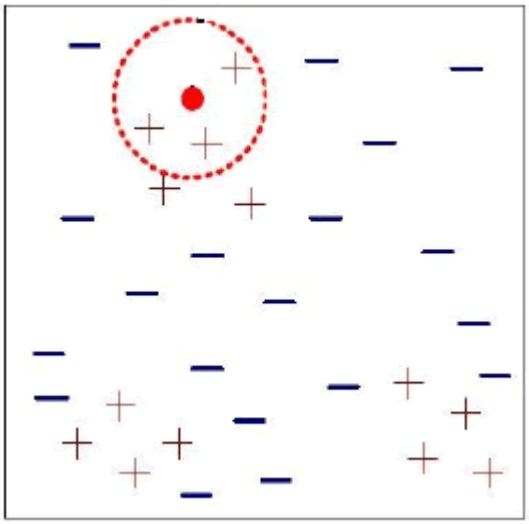


Рисунок 2: Классификация K-ближайших соседей.

В методе K-ближайшего соседа (KNN) ближайший сосед измеряется относительно значения k, которое определяет, сколько ближайших соседей необходимо проверить, чтобы описать класс выборочной точки данных [26]. Метод ближайшего соседа делится на две категории, т. е. KNN на основе структуры и KNN без структуры. Метод, основанный на структуре, имеет дело с базовой структурой данных, где структура имеет меньше механизмов, связанных с обучающими выборками данных [27]. В бесструктурном методе все данные классифицируются на точки выборки и данные обучения, расстояние вычисляется между точками выборки и всеми точками обучения, а точка с наименьшим расстоянием называется ближайшим соседом [28].

Одним из основных преимуществ метода KNN является то, что он эффективен для больших обучающих данных и устойчив к зашумленным обучающим данным [29]. Масштабирование запросов KNN по огромным многомерным наборам мультимедийных данных является стимулирующей проблемой для классификаторов KNN. Чтобы решить эту проблему, была введена высокопроизводительная мультимедийная система обработки запросов KNN [30], в этой системе методы сокращения на основе быстрого расстояния сочетаются с предлагаемой структурой индекса R-дерева (DPR-Tree) на основе вычислений Distance-Pre. Стоимость ввода/вывода снижается за счет этой эксклюзивной связи, но увеличивает вычислительную работу поиска KNN.

Два важных препятствия с классификаторами на основе ближайших соседей выделены в [19], в том числе; потребность в пространстве и время его классификации. Были введены различные методы для решения проблемы

нехватки места. Классификатор K-ближайших соседей (k-NNMC) был введен в [19]. K-NNMC независимо ищет k ближайших соседей для каждого класса шаблонов обучения и вычисляет среднее значение для всех заданных k-соседей. Экспериментально с использованием многочисленных стандартных наборов данных показано, что точность классификации предлагаемого классификатора лучше по сравнению с другими классификаторами, такими как взвешенный классификатор k-ближайших соседей (Wk-NNC) [10], и он может эффективно комбинироваться с любым пространством. Методы сокращения и индексации.

Преимущества KNN включают простоту, прозрачность, устойчивость к зашумленным обучающим данным, простоту понимания и реализации, а недостатки включают сложность вычислений, ограничение памяти, низкую производительность во время выполнения для большого обучающего набора и нерелевантные атрибуты, которые могут вызвать проблемы [28, 61].

1) Найдите k экземпляров в наборе данных, ближайший к экземпляру S.

2) Эти k экземпляров затем голосуют, чтобы определить класс экземпляра S.

Точность КБС зависит от метрики расстояния и значения K. Различные способы измерения расстояния между двумя экземплярами — косинусное, евклидово расстояние. Чтобы оценить новую неизвестную выборку, КБС вычисляет ее ближайших соседей и присваивает класс большинством голосов.

2.2.4. Логистическая регрессия

Логистическая регрессия, иногда называемая логистической моделью или логит-моделью, анализирует взаимосвязь между несколькими независимыми переменными и категориальной зависимой переменной и оценивает вероятность возникновения события путем подгонки данных к логистической кривой. Существуют две модели логистической регрессии: бинарная логистическая регрессия и полиномиальная логистическая регрессия. Бинарная логистическая регрессия обычно используется, когда зависимая переменная дихотомична, а независимые переменные либо непрерывны, либо категоричны. Когда зависимая переменная не является дихотомической и состоит из более чем двух категорий, можно использовать полиномиальную логистическую регрессию.

В качестве наглядного примера рассмотрим, как ишемическая болезнь сердца (ИБС). С помощью ЛР можно предсказать по уровню холестерина в сыворотке крови. Вероятность ИБС увеличивается с увеличением уровня холестерина в сыворотке крови. Однако между ИБС и холестерином в сыворотке является нелинейным, и вероятность ИБС очень мало меняется при низких или высоких крайних значениях холестерина Поскольку логистическая регрессия вычисляет вероятность наступления события по сравнению с вероятностью того, что событие не произойдет, влияние независимых переменных обычно объясняется с точки зрения шансов.

В исследовании мы обычно моделируем взаимосвязь между двумя переменными, а именно переменной X (независимой) и переменной Y (зависимой). Метод, обычно используемый в подобных исследованиях, представляет собой линейную регрессию, простую или множественную. Однако иногда линейная регрессия с использованием метода МНК (Обычный метод наименьших квадратов) не подходит для использования. Часто используемая линейная регрессия иногда нарушает предположение Гаусса-Маркова. Например, в случае, когда зависимая переменная (Y) имеет номинальный тип данных, а независимая переменная/предиктор (X) — интервальный или относительный тип данных.

Хотите знать, являются ли студенты финансово грамотными в зависимости от пола, выбранного факультета и совокупного среднего балла. В этом случае есть только 2 возможных ответа студентов, а именно: финансово грамотные студенты и финансово неграмотные студенты.

Из приведенного выше примера видно, что тип данных переменной ответа (Y) номинальный, а именно категоризация решений студентов быть финансово грамотными или нет (например, финансовая грамотность — это номер 1, а не финансовая грамотность — номер 1). 0), а тип данных для независимой переменной (X.) не менее интервала (шкала Лайкерта). Если к такому случаю применить обычный метод линейной регрессии, согласно Катнеру и соавт. (2004), будет 2 нарушения предположения Гаусса-Маркова и 1 нарушение пределов подогнанного значения переменной отклика (Y), а именно:

1. Ошибка полученной регрессионной модели не имеет нормального распределения.

2. Дисперсия ошибки возникает гетероскедастичность).

3. Между тем, нарушение установленного заключается в том, что оценочное значение, сгенерированное из обычной модели линейной регрессии, выходит за пределы диапазона от 0 до 1. Это явно необоснованно, поскольку предельное значение переменной Y (в данном случае это понимание грамотности) у высокая финансовая грамотность = 1 и низкая финансовая грамотность = 0. Для решения этой проблемы был введен метод логистической регрессии. Логистическая регрессия (иногда называемая логистической моделью или логит) в статистике используется для прогнозирования вероятности возникновения события путем подгонки данных к логит-функции логистической кривой.

Логистическая регрессия — это подход к созданию прогностических моделей, таких как линейная регрессия или обычно называемая регрессией по методу наименьших квадратов (МНК). Разница в том, что в логистической регрессии исследователь предсказывает зависимую переменную по дихотомической шкале. Рассматриваемая дихотомическая шкала представляет собой номинальную шкалу данных с двумя категориями, например: «Да» и «Нет», «Хорошо» и «Плохо» или «Высоко» и «Низко».

Если для МНК требуются условия или допущения, что дисперсия ошибки (остаток) имеет нормальное распределение. С другой стороны, логистическая

регрессия не нуждается в этом допущении, потому что логистическая регрессия следует логистическому распределению.

Предположения, которые должны выполняться в логистической регрессии, включают:

1. Логистическая регрессия не требует линейной зависимости между независимой переменной и зависимой переменной.
2. Независимая переменная не требует предположения о многомерной нормальности.
3. Предположение о гомоскедастичности не требуется
4. Независимые переменные не нужно преобразовывать в метрическую форму (интервальную или шкалу отношений).
5. Зависимая переменная должна быть дихотомической (2 категории, например: высокая и низкая или хорошая и плохая)
6. Независимые переменные не должны иметь одинаковое разнообразие между группами
7. переменных
8. Минимум требуется до 50 выборок данных для предикторной переменной (независимой).
9. Логистическая регрессия может выбирать отношения, потому что она использует подход нелинейного логарифмического преобразования для прогнозирования отношения шансов. Шансы в логистической регрессии часто выражаются как вероятности.

Модель алгебраического уравнения, подобная МНК, которую мы обычно используем, выглядит следующим образом: Y = B0 + B1X + e. Где e — дисперсия ошибки или невязка. В логистической регрессии не используется та же интерпретация, что и в уравнении регрессии МНК. Модель Образованное уравнение отличается от уравнения МНК.

Как и обычный метод регрессии, логистическую регрессию можно разделить на 2, а именно:

1. Бинарная логистическая регрессия (Binary Logistic Regression).

Бинарная логистическая регрессия используется, когда есть только 2 возможные переменные ответа (Y), например покупка и отказ от покупки.

2. Полиномиальная логистическая регрессия (многочленная логистическая регрессия).

Полиномиальная логистическая регрессия используется, когда переменная ответа (Y) имеет более двух категорий.

С помощью логистической регрессии среднее значение переменной отклика p в терминах объясняющей переменной x моделируется, связывая p и x с помощью уравнения $p = \alpha+\beta x$.

К сожалению, это не очень хорошая модель, потому что экстремальные значения x будут давать значения $\alpha+\beta x$, которые не попадают между 0 и 1. Решением этой проблемы с помощью логистической регрессии является преобразование шансов с использованием натурального логарифма (Peng , Ли и Ингерсолл, 2002). С помощью логистической регрессии мы моделируем натуральные логарифмические шансы как линейную функцию независимой

переменной: logit (y) =ln (шансы) = a + βχ (1), где p — вероятность интересующего исхода, а x — объясняющая переменная. Параметрами логистической регрессии являются α и β. Это простая логистическая модель.

Взяв антилогарифм уравнения (1) с обеих сторон, можно вывести уравнение для предсказания вероятности наступления интересующего исхода, поскольку логистическая регрессия вычисляет вероятность наступления события по сравнению с вероятностью того, что событие не произойдет, влияние независимых переменных обычно объясняется с точки зрения шансов. С помощью логистической регрессии среднее значение переменной отклика p в терминах объясняющей переменной x моделируется, связывая p и x с помощью уравнения p = α+βx.

К сожалению, это не очень хорошая модель, потому что экстремальные значения x будут давать значения α+βx , которые не попадают между 0 и 1. Решением этой проблемы с помощью логистической регрессии является преобразование шансов с использованием натурального логарифма (Peng , Ли и Ингерсолл, 2002).

Логистическая регрессия — это метод подбора кривой регрессии, y = f(x), когда y состоит из двоично-кодированных (0, 1—неудача, успех) данных. Когда ответ представляет собой бинарную (дихотомическую) переменную, а x является числовым, логистическая регрессия подгоняет логистическую кривую к взаимосвязи между x и y. Логистическая кривая представляет собой S-образную или сигмовидную кривую, часто используемую для моделирования роста населения (Eberhardt & Breiwick, 2012). Логистическая кривая начинается с медленного линейного роста, за которым следует экспоненциальный рост, который затем снова замедляется до стабильной скорости.

Ниже приведено уравнение логистической регрессии:

$$\ln\left(\frac{\hat{p}}{1-\hat{p}}\right) = B_0 + B_1 X$$

Где:

Ln : Натуральный логарифм.

$B_0 + B_1X$: Уравнение, широко известное в МНК.

В то время как P Accent - это логистическая вероятность, полученная по формуле вероятности логистической регрессии следующим образом:

$$\hat{p} = \frac{\exp(B_0 + B_1 X)}{1 + \exp(B_0 + B_1 x)} = \frac{e^{B_0 + B_1 x}}{1 + e^{B_0 + B_1 x}}$$

Где:

exp или написанное «e» - это экспоненциальная функция.

(Имейте в виду, что показатель степени противоположен натуральному логарифму. В то время как натуральный логарифм является логарифмической формой, но с постоянным значением 2,71828182845904 или обычно округляется до 2,72).

Простая логистическая функция определяется формулой

$$\text{logit}(y) = \ln\left(\frac{p}{1-p}\right) = a + \beta_1 \chi_1 + \ldots + \beta_k \chi_k$$

где α и β определяют логистическую точку пересечения и наклон.

Логистическая регрессия соответствует α и β, коэффициентам регрессии. Логистическая или логит-функция используется для преобразования S-образнойкривой в приблизительно прямую линию и для изменения диапазона пропорции от 0 – 1 до -∞ - + ∞ как

logit (y) =ln (шансы) = ln (п) = α + βχ

где p — вероятность интересующего исхода, α — параметр пересечения, β — коэффициент регрессии, а χ — предиктор.

2.2.5. Машины опорных векторов

Вапник предложил основанный на статистической теории обучения метод машинного обучения, известный как машина опорных векторов (SVM) [24]. SVM считается одним из самых популярных и экономичных методов решения задач, связанных с классификацией данных [25], обучением и прогнозированием [26]. Опорные векторы — это точки данных, ближайшие к области принятия решений [27]. Выполняет классификацию векторов данных над гиперплоскостью в безразмерном пространстве [28]. Максимально маргинальный классификатор — это более простая или базовая форма SVM, которая помогает решить более простую задачу классификации линейно разделимых обучающих данных с использованием бинарной классификации [27]. Классификатор максимального поля используется для поиска гиперплоскости максимального поля в реальных осложнениях [29]. Основным преимуществом SVM является его способность решать множество задач классификации, включая разделимые многомерные и нелинейные задачи. Основным недостатком SVM является то, что некоторые ключевые параметры

должны быть правильно установлены для отличных результатов классификации [20].

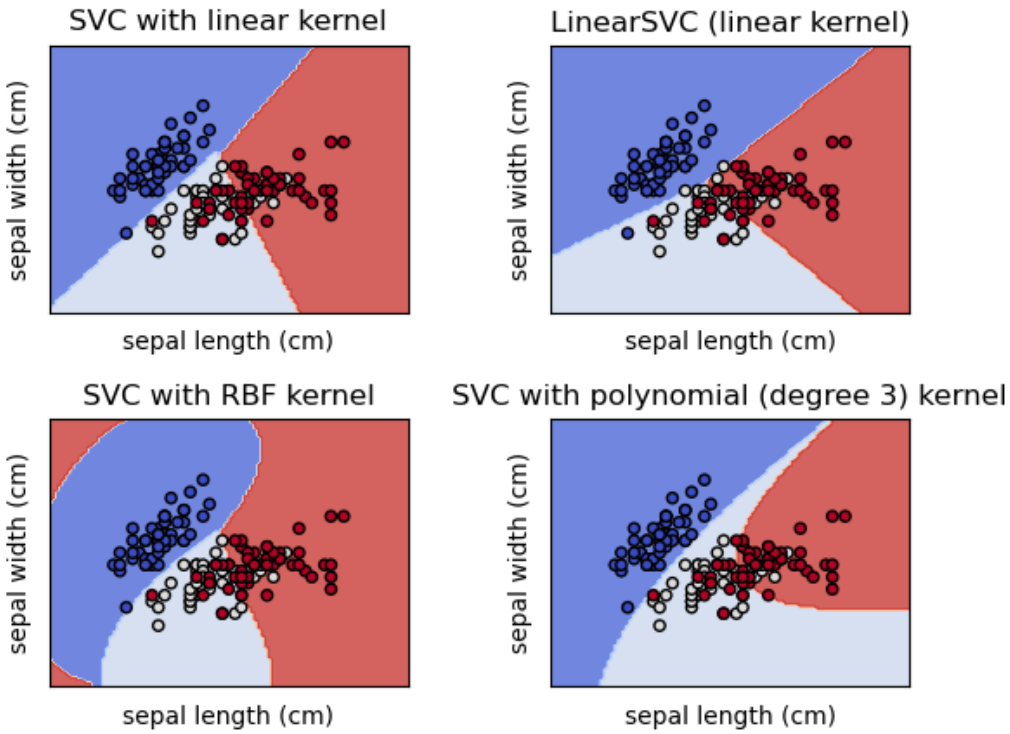


Рисунок 5. SVC

SVC и NuSVC — похожие методы, но они принимают несколько разные наборы параметров и имеют разные математические формулировки (см. раздел «Математическая формулировка»). С другой стороны, LinearSVC — это еще одна (более быстрая) реализация классификации опорных векторов для случая линейного ядра. Обратите внимание, что LinearSVC не принимает параметр , так как он считается линейным. В нем также отсутствуют некоторые атрибуты SVC и NuSVC, например .kernelsupport_.Как и другие классификаторы, SVC, NuSVC и LinearSVC принимают в качестве входных данных два массива: массив формы, содержащий обучающие образцы, и массив меток классов (строки или целые числа) формы: X(n_samples, n_features)y(n_samples)

```
>>> from sklearn import svm
>>> X = [[0, 0], [1, 1]]
>>> y = [0, 1]
>>> clf = svm.SVC()
>>> clf.fit(X, y)
SVC()
```

После подгонки модель можно использовать для прогнозирования новых значений:

```
>>> clf.predict([[2., 2.]])
array([1])
```

Функция принятия решений SVM (подробно описанная в математической формулировке) зависит от некоторого подмножества обучающих данных, называемых опорными векторами. Некоторые свойства этих векторов поддержки можно найти в атрибутах и :support_vectors_support_n_support_

```
>>> # get support vectors
>>> clf.support_vectors_
array([[0., 0.],
       [1., 1.]])
>>> # get indices of support vectors
>>> clf.support_
array([0, 1]...)
>>> # get number of support vectors for each class
>>> clf.n_support_
array([1, 1]...)
```

Мультиклассовая классификация

SVC и NuSVC реализуют подход «один против одного» для многоклассовой классификации. Всего строятся классификаторы, и каждый из них обучает данные из двух классов. Чтобы обеспечить согласованный интерфейс с другими классификаторами, опция позволяет монотонно преобразовывать результаты классификаторов «один против одного» в решающую функцию «один против остальных» формы .n_classes * (n_classes - 1) / 2decision_function_shape (n_образцов, n_классов)

```
>>> X = [[0], [1], [2], [3]]
>>> Y = [0, 1, 2, 3]
>>> clf = svm.SVC(decision_function_shape='ovo')
>>> clf.fit(X, Y)
SVC(decision_function_shape='ovo')
>>> dec = clf.decision_function([[1]])
>>> dec.shape[1] # 4 classes: 4*3/2 = 6
6
>>> clf.decision_function_shape = "ovr"
>>> dec = clf.decision_function([[1]])
>>> dec.shape[1] # 4 classes
4
```

С другой стороны, LinearSVC реализует многоклассовую стратегию «один против остальных», таким образом обучая модели.

```
>>> lin_clf = svm.LinearSVC()
>>> lin_clf.fit(X, Y)
LinearSVC()
>>> dec = lin_clf.decision_function([[1]])
>>> dec.shape[1]
4
```

Полное описание решающей функции см. в разделе «Математическая формулировка».

Обратите внимание, что LinearSVC также реализует альтернативную мультиклассовую стратегию, так называемую мультиклассовую SVM, сформулированную Краммером и Сингером 16, с использованием параметра . На практике обычно предпочтительнее использовать классификацию «один

против остальных», так как результаты в основном схожи, но время выполнения значительно меньше.multi_class='crammer_singer'

Для LinearSVC «один против остальных» атрибуты и имеют форму и соответственно. Каждая строка коэффициентов соответствует одному из классификаторов «один против остальных» и аналогичным для перехватов в порядке «одного» class.coef_intercept_(n_classes, n_features)(n_classes,)n_classes

В случае SVC и NuSVC «один на один» расположение атрибутов немного сложнее. В случае линейного ядра атрибуты и имеют вид и соответственно. Это похоже на макет для LinearSVC, описанный выше, где каждая строка теперь соответствует двоичному классификатору. Порядок классов от 0 до n следующий: «0 против 1», «0 против 2», … «0 против n», «1 против 2», «1 против 3», «1 против n», . . . «n-1 против n».coef_intercept_(n_classes * (n_classes - 1) / 2, n_features) (n_classes * (n_classes - 1) / 2)

Форма с несколько трудным для понимания макетом. Столбцы соответствуют опорным векторам, участвующим в любом из классификаторов «один против одного». Каждый опорный вектор имеет двойной коэффициент в каждом из классификаторов, сравнивающих класс с другим классом. Обратите внимание, что некоторые, но не все, из этих двойных коэффициентов могут быть равны нулю. Записи в каждом столбце представляют собой эти двойные коэффициенты, упорядоченные по противоположному классу.

2.2.6. Оптимизация роя частиц

Оптимизация роя частиц, именуемая здесь и далее ОРЧ, представляет собой алгоритм оптимизации на основе ЕС, предложенный Кеннеди и Эберхартом [15]. ОРЧ вдохновлен социальным поведением, таким как стайка рыб и стаи птиц. В ОРЧ популяция (рой) кодируется как частицы. Поиск начинается со случайной инициализации популяции. Весь рой перемещается в пространстве поиска лучшего решения, обновляя положение каждой частицы. Позиция каждой частицы определяется на основе ее собственной позиции, а также на основе соседних частиц. Текущее положение частицы представлено как $X_i = \{x_{i1}, x_{i2} - x_{iD}\}$, где D - размерность пространства поиска. Скорость частицы представлена как $V=(V_{i1}, V_{i2} ---- V_{iD})$. Скорость частицы ограничена V_{max} и V_{tid} $[-V_{max}, V_{max}]$. Наилучшая предыдущая позиция и наилучшая полученная позиция представлены как p_{best} и g_{best}. ОРЧ ищет оптимальное решение, обновляя положение и скорость на основе $p_{наилучшего}$ и $g_{наилучшего}$ [16]. ОРЧ используется в качестве метода выбора функций из-за его преимуществ, таких как

1. Простота реализации
2. Может быстрее сходиться
3. Вычислительно дешевле и проще в реализации

2.3. Сравнительный анализ выбранных методов МО на конкретном примере

Набор медицинских данных содержит большое количество избыточных и нерелевантных признаков. Производительность классификатора может снизиться, если набор данных содержит признаки такого типа. За счет удаления избыточных функций точность классификатора повышается и сокращается время работы. Методы выбора признаков в целом классифицируются как:

1. Фильтр
2. Обертка
3. Гибридные подходы

Выбор признаков для большого набора данных является сложной задачей. Многие методы поиска, используемые для выбора признаков, страдают локальными оптимумами и высокими вычислительными затратами. Следовательно, для разработки метода выбора признаков требуется дешевый алгоритм глобального поиска. В предлагаемом нами подходе мы использовали ОРЧ для задачи выбора признаков.

Метод направлен на повышение эффективности классификатора КБС для прогнозирования заболеваний. Алгоритм предлагаемого нами метода показан ниже как Алгоритм 1.

Шаг 1: Ввод: набор данных о сердечных заболеваниях

Шаг 2: Результат: классификация набора данных на пациентов с сердечными заболеваниями и нормальных пациентов.

Шаг 3: Введите набор данных

Шаг 4. Примените методы предварительной обработки. Заполните пропущенные значения.

Шаг 5: выберите признаки на основе значений, полученных после применения ОРЧ в качестве FSS.

Шаг 6. Откажитесь от избыточных функций (функций с низкими значениями ОРЧ)

Шаг 7: Применить (КБС+Межквартильный размах) к преобладающим функциям

Шаг 8. Измерьте производительность модели КБС+ОРЧ.

Алгоритм 1. Прогнозирование заболеваний сердца с использованием КБС и ОРЧ.

Алгоритм берет набор данных о сердечных заболеваниях и классифицирует, есть ли у человека сердечные заболевания или нет. Приведенный выше алгоритм разбит на 2 части. Часть 1 (строки 3-6) выполняет обработку и выбор подмножества признаков. Эта часть выбирает только преобладающие признаки для дальнейшего процесса. В части 2 (строки 7-8) КБС применяется к предварительно обработанному набору данных и измеряется производительность. Мера выбора признаков ОРЧ используется для выбора лучших признаков для получения высокой точности.

Результаты экспериментов

Для прогнозирования сердечных заболеваний из репозитория UCI был собран набор данных, содержащий 270 случаев [18]. Информация о наборе данных о сердечных заболеваниях показана в таблице 1 .

Таблица 1

Набор данных о сердечных заболеваниях

Набор данных	Экземпляры	Функции
Болезнь сердца	270	14

WEKA используется в качестве основного пакета. Чтобы найти точность, мы запускаем 10 перекрестных проверок. Технические характеристики КНН и ПСО приведены в таблицах 2 и 3 .

Таблица 2

Технические характеристики КНН

Сл.номер	Технические характеристики КНН
1	КНН=2
2	Перекрестная проверка = 2
3	NN Поиск=линейный
4	Средний квадрат = ложь

Таблица 3

Технические характеристики ОРЧ

Сл. нет	Технические характеристики
1	Численность: 100
2	Количество поколений: 50
3	Частота отчетов: 50
4	Случайное семя=1

Из 14 функций поиск ОРЧ выбирает 8 функций (включая класс). Остальные 6 признаков не будут учитываться при классификации болезней сердца. Эти 7 функций являются преобладающими функциями, которые повысят точность классификатора. **В таблице 5** показана точность, полученная нашей моделью для данных о сердечных заболеваниях.

Функции, выбранные ОРЧ (доминирующие функции), перечислены в Таблице 4 .

Таблица 4

Функции, выбранные ОРЧ

Сл.номер	Название функции
1	Грудь
2	Отдых_электрокардиографические_результаты
3	Максимальная_частота сердечных сокращений_достигнута
4	Упражнение_индуцированная_стенокардия
5	Старый пик
6	Number_of_major_vessels
7	Таль

Таблица 5

Точность, полученная нашей моделью

Метод	значение К				
	К=1	К=2	К=3	К=4	К=5
До ФСС	75,18	77.03	78	78	78,14
После нормализации	77,7	78,8	81,1	81,4	81,4
После дискретизации	79,2	79,25	81,1	81,1	80,3
После ПСО+ КБС	78,8	81,1	81,4	81,4	81,4
КБС+ОРЧ+Межквартильный размах	100	100	100	100	100

Из 14 функций поиск ОРЧ выбирает 8 функций (включая класс). Остальные 6 признаков не будут учитываться при классификации болезней сердца. Эти 7 функций являются преобладающими функциями, которые повысят точность классификатора. В таблице 5 показана точность, полученная нашей моделью для данных о сердечных заболеваниях.

Точность получена для различных значений К. Мы протестировали четыре метода записи точности классификатора. Межквартильный диапазон (Межквартильный размах) является мерой изменчивости. Он делит набор данных на квартили.

Q1: В ранжированном наборе данных среднее значение в первой половине. Q2: Среднее значение в наборе данных.

Q3: это «среднее» значение во второй половине набора данных.

Межквартильный размах=Q3-Q1 → (1)

Точность, зарегистрированная нашей моделью до выбора подмножества признаков, составляет 75,18 для k = 1 и 78,14 для k = 5. Фильтр дискретизации в WEKA повысил точность с 75,18 до 79,2 для k=1 и с 78,14 до 80,3 для k=5. Фильтр Межквартильный размах вместе с ОРЧ повысили точность до 100%. На рисунке 5 показана точность, зарегистрированная нашей моделью для различных значений К. Результаты, полученные с помощью модели КБС+ОРЧ, показывают, что предлагаемая нами модель улучшит точность на хорошем уровне. Эксперименты для предлагаемого нами подхода были проведены на четырех различных наборах данных. Болезни сердца-1 и Болезни сердца-2 — это реальные наборы данных, собранные в различных больницах Индии.

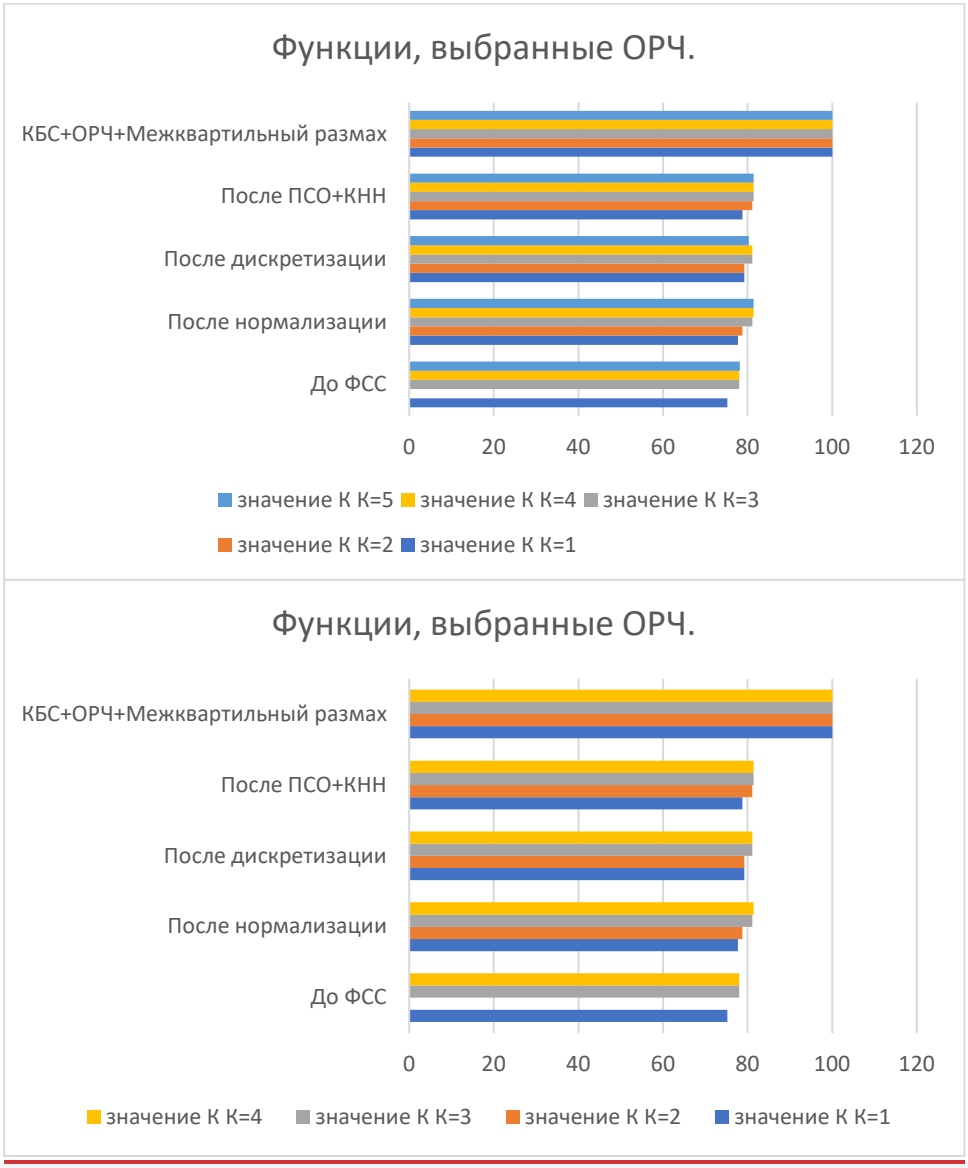


Рисунок 5. Точность, зафиксированная предложенной моделью для различных значений k.

Результаты предлагаемого подхода для различных наборов данных показаны в таблице 6 . Значения значений параметров производительности занесены в Таблицу 7 . Истинная положительная скорость и чувствительность записываются как 100%.

Таблица 6

Точность, полученная для различных наборов данных

Сл. нет	Набор данных	Экземпляры	Атрибуты	Точность
1	Болезнь сердца-1	40	10	97,5
2	Болезнь сердца-2	75	12	100
3	Данные о рабочей силе	57	17	100
4	Соевый боб	683	36	100

Таблица 7

Значения различных параметров. (Набор данных сталлога сердца)

Имя параметра	Ценность
Чувствительность	100%
целевая цена	100%
Точность	100%

Предлагаемый подход (КБС+ОРЧ) сравнивается с КБС+GA. При использовании GA точность составляет 77,7%, что показано в таблице 8 .

Таблица 8

Сравнение точности с GA и ОРЧ

Имя набора данных	Подход	Точность
Болезнь сердца	КНН+ГА	77,7
	КНН+ПСО	100

В таблице 6 и на рисунке 3 показано сравнение точности обучения нашей модели с другими моделями, упомянутыми в опросе.

Из результатов моделирования, полученных из таблицы 9 и рисунка 6 , наш подход достиг повышенной точности за счет учета только преобладающих признаков. Наш подход помогает врачам предложить пациентам пройти диагностический тест для прогнозирования заболевания.

Таблица 9

Сравнение точности

Сл. нет	Метод	Точность (%)
1	Марал[7]	92,5
2	Арай [8]	97
3	Кумар [9]	92
4	Амина[10]	96,2
5	Серик[11]	99
6	Арман [14]	98
7	Наш подход	100

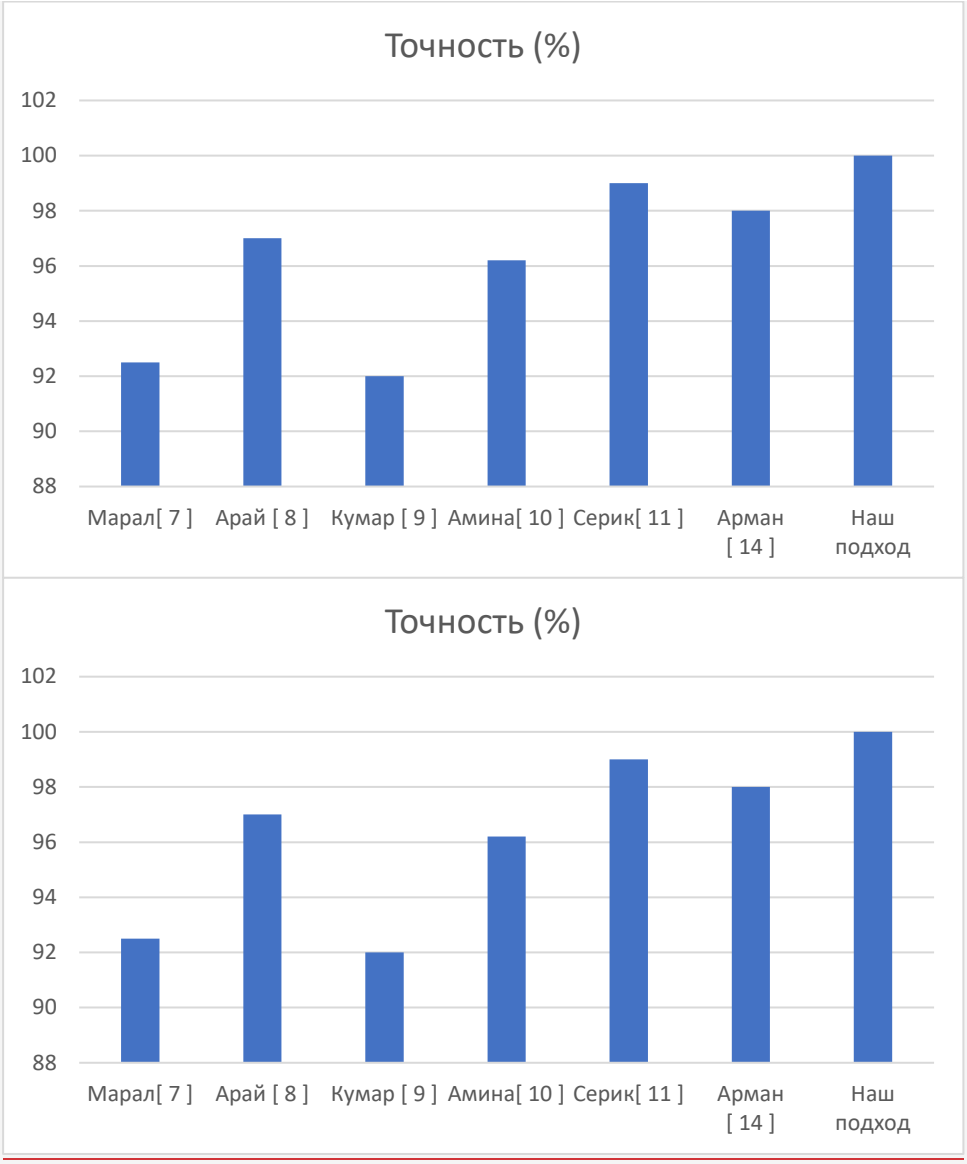


Рисунок 6. Сравнение точности.

Полученная точность выбора подмножества признаков составляет 75%. Поиск ОРЧ фильтрует количество функций и выбирает функции, которые вносят больший вклад в классификацию. За счет применения КБС с ОРЧ точность повысилась до 100%. Точность увеличилась почти на 25%. Мы протестировали НАШУ модель для различных значений k. Наш эксперимент ограничен k = 5, так как нет большого прироста точности. Предлагаемый подход хорошо подходит для многомерного набора данных. Из Таблицы 9 видно, что ОРЧ играет ключевую роль в повышении точности в качестве меры выбора признаков.

2.4. Программные пакеты на Python для реализации методов машинного обучения

Алгоритмы машинного обучения можно разделить на несколько категорий. С точки зрения того, имеет ли алгоритм параметры для оптимизации, его можно разделить на:

- параметрический. В этой группе мы сводим задачу к оптимизации параметров. Мы предполагаем, что задача может быть представлена функцией некоторого вида (например, линейной, полиномиальной и т. д.). Примером этой группы является линейная модель.

- непараметрический. В этой группе мы не предполагаем, что задача может быть представлена функцией определенного вида. Примерами этой группы являются Наивный Байес, дерево решений (ID3) и K-ближайшие соседи.

С другой точки зрения, типы алгоритмов можно разделить на:

Линейные модели, примеры линейной регрессии, логистическая регрессия, метод опорных векторов.

Вероятностные модели, например, Наивный Байес, скрытые марковские модели.

Нелинейная модель, а именно (обычно) искусственная нейронная сеть. ЛР – базовый метод машинного обучения, но бывают случаи (особенно с очень сложными моделями), когда логистическая регрессия не справляется. В таких обстоятельствах можно использовать другие методы классификации:

- k-ближайшие соседи
- Наивные байесовские классификаторы
- Опорные векторные машины
- Деревья решений
- Случайные леса
- Нейронные сети

К счастью, существует несколько подробных библиотек Python для МО, которые реализуют эти методы. Например, пакет scikit-Learn, который вы видите в действии, активирует все вышеперечисленные методы, кроме нейронных сетей.

Для всех этих методов scikit-learn предоставляет подходящие классы с методами, такие как model.fit(), model.predict_proba(), model.predict(), model.score(). Можно комбинировать train_test_split(), confusion_matrix(), classification_report() и многое другое.

Нейронные сети (включая глубокие нейронные сети) стали очень популярными для задач классификации. Следующие библиотеки TensorFlow, PyTorch или Keras, обеспечивают подходящую, функциональную и мощную поддержку таких моделей.

Два наглядных примера логистической регрессии, решенной с помощью scikit-learn

Один концептуальный пример, решенный с помощью StatsModels

Один реальный пример классификации рукописных цифр

Пакеты Python для логистической регрессии

Есть несколько пакетов, которые вам понадобятся для логистической регрессии в Python. Все они бесплатные и с открытым исходным кодом, с большим количеством доступных ресурсов. Во- первых, вам понадобится NumPy — основной пакет для научных и числовых вычислений в Python.

NumPy полезен и популярен, поскольку позволяет выполнять высокопроизводительные операции с одномерными и многомерными массивами.

В NumPy есть много полезных процедур работы с массивами. Он позволяет писать элегантный и компактный код и хорошо работает со многими пакетами Python. Если вы хотите изучить NumPy, то можете начать с официального руководства пользователя. Справочник по NumPy также содержит исчерпывающую документацию по его функциям, классам и методам.

Следующий пакет Python, которого можно использовать, — это scikit-learn. Считается популярным пакетом, которого используют в машинном обучении. Также можно применить scikit-learn для выполнения различных функций:

- Предварительно обработать данные
- Уменьшить размерность проблем
- Проверка моделей
- Выберите наиболее подходящую модель
- Решение задач регрессии и классификации
- Реализовать кластерный анализ

Вы найдете полезную информацию на официальном веб- сайте scikit-learn, где вы, возможно, захотите прочитать об обобщенных линейных моделях и реализации логистической регрессии. Если вам нужна функциональность, которую scikit-learn не может предложить, вам могут пригодиться StatsModels . Это мощная библиотека Python для статистического анализа. Дополнительную информацию мы можем найти на официальном сайте .

Можно применить Matplotlib для рендеринга результатов вашей классификации. Это полная библиотека Python, широко используемая для создания высококачественной графики. Для получения дополнительной информации мы можем посетить официальный веб-сайт и руководство пользователя. Существует несколько учебных ресурсов Matplotlib, которые могут быть полезны, например, официальные учебники, «Анатомия Matplotlib» и «Питограммирование с помощью Matplotlib» (руководство).

Интеллектуальный анализ данных — это процесс анализа больших баз данных для прогнозирования тенденций . Этот процесс сложен. Специалисты по данным исследуют большие объемы информации и делают определенные оценки на основе этих данных. Интеллектуальный анализ данных включает в себя анализ социальных сетей, создание криминальной картины и т. д.

Еще одна вещь, которая входит в полезность Python, — это управление данными и их очистка . Python считается одним из лучших языков программирования для работы. Кроме того, машинное обучение с помощью Python упрощает анализ данных за счет использования алгоритмов.

Python известен своим широким разнообразием фреймворков, предоставляющих большое количество предварительно написанных фрагментов кода, которые позволяют разработчикам улучшать качество своих

проектов. То же самое касается интеллектуального анализа данных. Вот список самых популярных фреймворков для анализа данных:

Numpy — это ведущая платформа, предназначенная для числовых вычислений в Python.

SciPy — это модуль для науки, математики и инженерии.

Scikit-Learn — это платформа машинного обучения Python для продуктивного интеллектуального анализа данных, позволяющая выполнять процессы регрессии, кластеризации, выбора модели, предварительной обработки и классификации.

Dask — это платформа для расширенного параллелизма для аналитики и масштабирования кластеров с тысячей узлов.

Настольная программа с графическим интерфейсом

Графический пользовательский интерфейс (GUI) также является примером полезности Python. Графические интерфейсы позволяют людям взаимодействовать с компьютерами, используя визуальные элементы, такие как значки или изображения, вместо текстовых команд. В Python доступно множество модулей для создания графических интерфейсов. Сказав это, мы покажем вам некоторые из наиболее часто используемых:

Tkinter — это встроенный интерфейс Python. Этот набор инструментов/инструментов с графическим интерфейсом работает на всех самых популярных платформах, таких как Microsoft, Linux и Mac OS X.

PyGTK — это бесплатный набор инструментов, который помогает создавать графические интерфейсы.

wxPython — это связующее звено для кроссплатформенного инструментария GUI и wxWidgets. Изначально разработчики делали wxPython на C++. Однако Python заменил C++.

Kivy — это библиотека Python для создания мобильных приложений и прикладного программного обеспечения с поддержкой мультитач. Это отличный выбор для разработки пользовательских интерфейсов и взаимодействий.

Мы должны создать наборы тестов и создать их до проверки ваших данных. Набор данных MNIST разделен на два набора: один для обучения и один для тестирования. x_tr, x_tes, y_tr, y_te = x [:60000], x[60000:], y[:60000], y[60000:]

Давайте поиграем с вашим набором упражнений следующим образом, чтобы сделать перекрестную проверку похожей (без каких-либо цифр сопоставляется) отсутствует)

Импортировать numpy как np

myData = np.radom.permutaion(500000) x_tr, y_tr = x_tr[myData], y_tr[myData]

Теперь пришло время сделать это довольно просто, мы попробуем идентифицировать только одну цифру, например число 6. «6detector» это будет пример бинарного классификатора, чтобы различать 6 и не 6, поэтому мы создадим вектор для этой задачи:

Y_tr_6 = (y_tr == 6) // это означает true для 6 секунд, и false для любого другого числа Y_tes_6 = (Y_tes == 6)

После этого мы можем выбрать классификатор и обучить его. Начните сстохастический градиентный спускклассификатораПреимущество класса ScikitLearn заключается в обработке очень больших наборов данных. В этом примере SGD будет обрабатывать экземпляры отдельно следующим образом.

```
from sklearn.linear_model import SGDClassifier mycl = SGDClassifier (random_state = 42) mycl.fit(x_tr, y_tr_6)
обнаружения 6
>>>mycl.prdict([any_digit)
```

используйте его для классификатора, это будет сложнее, чем регрессор, поэтому давайте объясним, как оценивать классификатор. В этом примере мы будем использовать перекрестную проверку для оценки нашей модели.

Мы используем класс StratifiedFold для выполнения стратифицированной выборки, в результате чего получается складка, содержащая выделение для каждого класса. Затем каждая итерация в коде будет создавать клон классификатора, чтобы делать прогнозы на тестовом сгибе. И, наконец, рассчитает количество правильных прогнозов и их соотношение.

Теперь мы будем использовать функцию cross_val_score для оценки SGDClassifier с перекрестной проверкой Kfold. K-кратная перекрестная проверка разделит обучающий набор на 3 сгиба, затем для каждого сгиба будут сделаны прогнозы и оценки из sklearn.model_selection import cross_val_score cross_val_score (sgd_clf, x_tr, y_tr_6, cv = 3, scoring = «accuracy»)

Вы получите коэффициент точности «правильного прогноза» для всех складок. Давайте классифицируем каждый классификатор в каждом изображении в not6

Проверим точность этой модели с помощью следующего кода:

На выходе вы получите не менее 90%: только 10% изображений являются 6s, поэтому мы всегда можем представить, что изображение не 6 Мы будем правы примерно в 90% случаев. Имейте в виду, что точность — не лучшая мера производительности классификаторов, если вы работаете с искаженными наборами данных.

Матрица смешаний

Существует лучший метод для оценки производительности вашего классификатора: матрица путаницы. Легко измерить производительность с помощью матрицы путаницы, просто подсчитав, например, сколько раз экземпляр класса X классифицируется как класс Y. Чтобы получить количество раз, когда классификатор изображений 6s на 2s, вы должны посмотреть в 6-ю строку и 2-й столбец матрицы путаницы. Давайте вычислим матрицу путаницы, используя функцию cross_val_predict().

```
from sklearn.model_selection import cross_Val_predict y_tr_pre =
cross_val_predict (sgd_cl, x_tr, y_tr_6, cv = 3)
```

Эта функция, как и функция cross_val_score(), выполняет перекрестную проверку k сгибов, а также возвращает прогноз для каждого сгиба. Он также возвращает чистый прогноз для каждого экземпляра в вашем тренировочном наборе. Теперь мы готовы получить матрицу, используя следующий код.

из sklearn.metrics import путаница_матрица путаница_матрица (y_tr_6, y_tr_pred)

Вы получите массив из 4 значений, «числа».

Каждая строка представляет класс в матрице, а каждый столбец представляет прогнозируемый класс. Первая строка — отрицательная: «содержит не 6 изображений». Из матриц можно многому научиться. Но есть и хорошая, интересная вещь, с которой можно работать, если вы хотите получить положительную точность прогнозирования, а именно точность классификаторов, использующих это уравнение.

Точность = (TP)/(TP+FP) TP: количество истинных положительных результатов FP: количество ложных срабатываний

Отзыв = (TP) /(TP+FN) «чувствительность»: измеряет соотношение положительных образцов (рис.7).

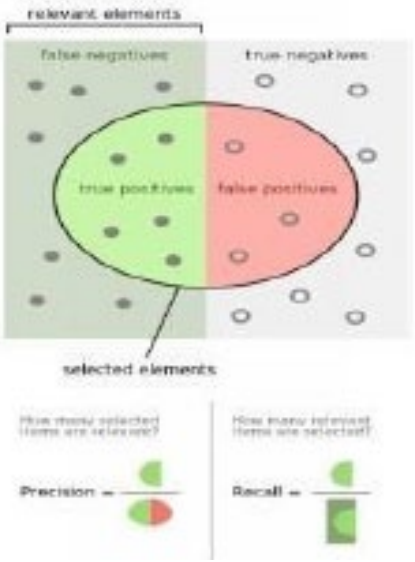


Рисунок 7. Расчет

Очень часто точность и полнота объединяются только в одном показателе, а именно в балле F1. F1 — среднее значение точности и полноты. Мы можем рассчитать оценку F1 с помощью следующего уравнения:

F1 = 2 / ((1/точность) + (1)/отзыв)) = 2 * (точность * отзыв) / (точность + отзыв) = (TP) / ((TP)

+ (FN+FP)/2)

Чтобы рассчитать балл F1, просто используйте следующую функцию Рисунок 8.:

```
>>> from sklearn.metrics import f1_score
>>>f1_score (y_tr_6, y_pre)
```

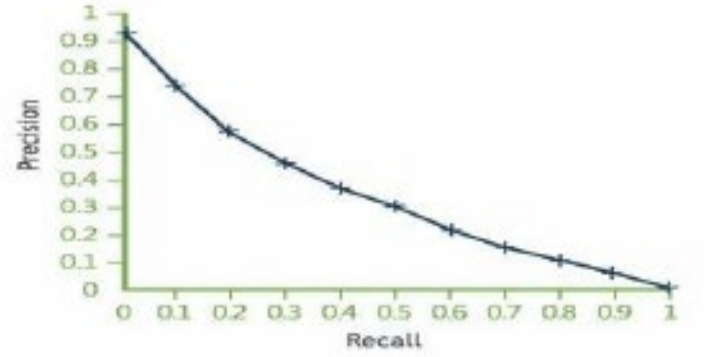


Рисунок 8. График чувствительности

Чтобы добраться до этого момента, вы должны посмотреть на SGDClassifier и на то, как SGDClassifier принимает решения относительно классификации. . Он вычисляет оценку на основе функции принятия решения, а затем сравнивает оценку с порогом. Если он больше этой оценки, он установит для экземпляра значение «положительное или отрицательное». Например, если порог решения находится посередине, вы найдете 4 истинных + справа от порога и только один ложный. Таким образом, коэффициент точности будет составлять всего 80%. (Рисунок 9).

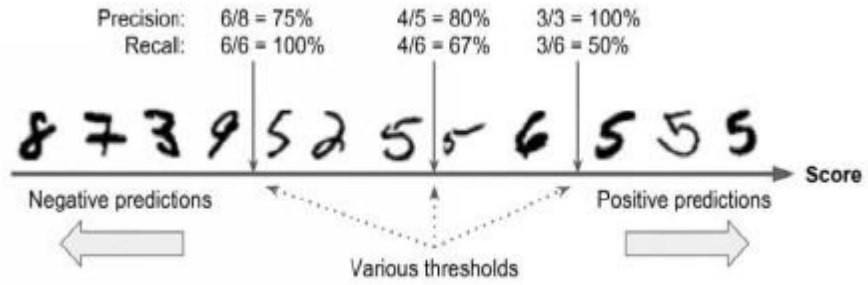


Рисунок 9. Установка результата прогнозирования

В ScikitLearn вы не можете установить порог напрямую. Вы должны получить доступ к оценке решения, которая использует предсказание, и вызвав функцию принятия решения ().

```
>>> y_sco = sgd_clf.decision_funciton([any digit])
>>> y_sco
>>> threshold = 0
>>>y_any_digit_pre = (y_sco > threshold)
```

В этом коде SGDClassifier содержит порог = 0, чтобы вернуть тот же результат, что и функция прогнозирования().

```
>>> threshold = 20000
>>>y_any_digit_pre = (y_sco > threshold)
>>>y_any_digit_pre
```

Этот код подтвердит, что по мере увеличения порога вывод уменьшается.

y_sco = cross_val_predict (sgd_cl, x_tr, y_tr_6, cv = 3, method = «функция решения»)

Пришло время вычислить все возможные точности и повторения для порога, вызвав функцию precision_recall_curve()

из sklearn.metrics import precision_recall_curve

точность, отзыв, порог, y_sco) и теперь давайте построим график точности и полноты с y_tr_6 помощью

Matplotlib

. Этот инструмент похож на кривую полноты, но не отображает точность и полноту: он отображает положительные и ложные частоты. Вы также будете работать с FPR, то есть с отрицательной частотой дискретизации. Можно представить что-то вроде (1 — отрицательный показатель. Другое понятие — TNR и его специфичность. Вспомним = 1 — специфичность. Давайте поиграем с ROC Curve. Сначала нам нужно рассчитать TPR и FPR, просто вызвав функцию roccurve() ,

После этого вы построите графики FPR и TPR с помощью Matplotlib в инструкциями соответствии со следующими как классификатор случайного леса или байесовский классификатор, который может сравнивать более двух. Однако, с другой стороны, SVM (машина опорных векторов) и линейные классификаторы работают как двоичные классификаторы. Если вы хотите разработать систему, которая классифицирует цифровые изображения на 12 классов (от 0 до 11), вам нужно обучить 12 бинарных классификаторов и создать по одному для каждого классификатора (например, 4 — de детектор, 5детектор, 6детектор и так далее), а затем вы должны получить DS, «оценку решения» каждого классификатора для изображения. Затем вы выберете классификатор с наивысшим баллом. Мы называем это стратегией OvA: «один против всех».

Другой метод заключается в обучении двоичного классификатора для каждой пары цифр; например, один для 5 и 6, а другой для 5 и 7. — мы называем этот метод OvO, «один к одному» — чтобы рассчитать, сколько классификаторов вам нужно, исходя из количества классов, используя следующее уравнение: «N = количество классов».

N*(N1)/2. Если бы вы хотели использовать этот метод с MNIST 10 * (101)/2, на выходе было бы 45 классификаторов, «бинарных классификаторов».

В ScikitLearn вы автоматически запускаете OvA при использовании алгоритма бинарной классификации.

Кроме того, вы можете вызвать функцию solution_function(), чтобы вернуть оценку «10 баллов для одного класса».

```
def plot_pre_re(pre, re, thr):
plt.plot(thr, pre[:-1], "b—", label = "precision")
plt.plot(thr, re[:1], "g-", label="Recall")
plt.xlabel("Threshold")
plt.legend(loc="left")
plt.ylim([0,1])
plot_pre_re(pre, re, thr)
plt.show
```

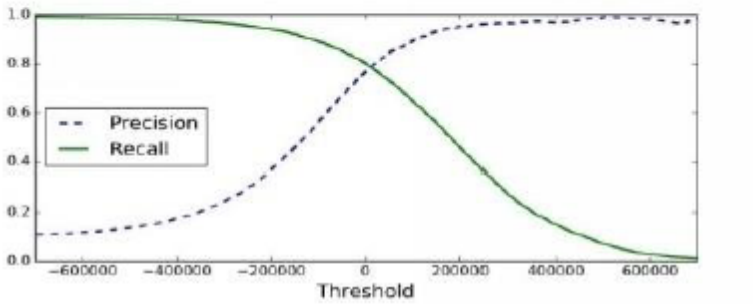
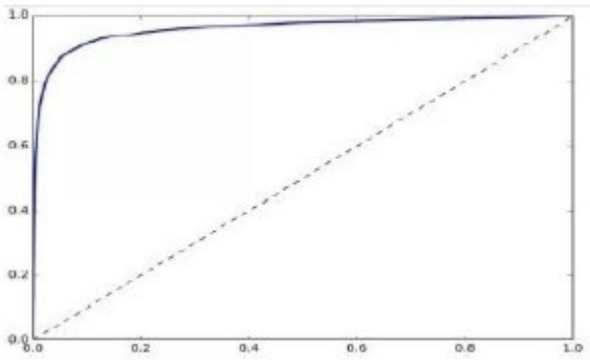


Рисунок 10. Обучение классификатора случайного леса

Как видите, обучить классификатор случайного леса всего двумя строками кода очень просто. ScikitLearn не выполняет функции OvA или OvO, потому что алгоритмы такого рода — «классификатор случайного леса» — могут автоматически работать с несколькими классами. Если вы хотите увидеть список возможных классификаторов, вы можете вызвать функцию predict_oroba().

```
def_roc_plot (fp, tp, label=none):
plt.plot(fp, tp, linewidth=2, label = label)
plt.plot([0,1)], [0,1], "k--")
plt.axis([0,1,0,1])
plt.xlabel('This is the false rate')
plt.ylabel('This is the true rate')
roc_plot (fp, tp)
plt.show
```



11. Кривая результатов

Классификатор очень точен в своих прогнозах, как вы можете видеть на выходе; под номером индекса 5 0,8. Оценим классификатор с помощью функции cross_val_score().

Вы получите на 84% больше. При использовании случайного классификатора вы получите в этом случае 10% за оценку точности. Имейте в виду, что чем выше это значение, тем лучше.

Анализ ошибок

Прежде всего, при разработке проекта машинного обучения:

определите проблему;

Соберите свои данные;

Работайте над своими данными и исследуйте;

Чистые данные

Работайте с несколькими моделями и выбирайте лучшую;

Включите свои модели в решение;

Покажите свое решение;

Запустите и проверьте свою систему.

Во-первых, вам нужно работать с матрицей путаницы и делать прогнозы с помощью функции crossval. Затем вы вызовете функцию матрицы путаницы:

```
>>> y_tr_pre = cross_val_prediciton(sgd_cl, x_tr_scaled, y_tr, cv=3)
>>> cn_mx  = confusion_matrix(y_tr, y_tr_pre)
>>> cn_mx
```

```
array([[5625, 2, 25, 8, 11, 44, 52, 12, 34, 6],
[ 2, 2415, 41, 22, 8, 45,  10, 10, 9],
[ 52, 43, 7443, 104, 89, 26, 87, 60, 166, 13],
[ 47, 46, 141, 5342, 1, 231, 40, 50, 141, 92],
[ 19, 29, 41, 10, 5366, 9, 56, 37, 86, 189],
[ 73, 45, 36, 193, 64, 4582, 111, 30, 193, 94],
[ 29, 34, 44, 2, 42, 85, 5627, 10, 45, 0],
[ 25, 24, 74, 32, 54, 12, 6, 5787, 15, 236],
[ 52, 161, 73, 156, 10, 163, 61, 25, 5027, 123],
[ 50, 24, 32, 81, 170, 38, 5, 433, 80, 4250]])
```

```
Plt.matshow(cn_mx, cmap=plt.cm.gray)
Plt.show()
```

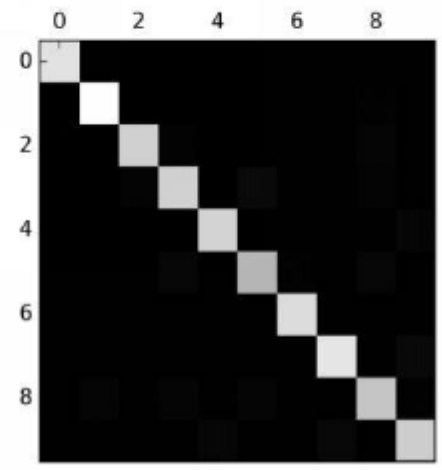


Рисунок 12 Результаты. Массив отображаемых результатов

Сначала вы должны разделить каждое значение в матрице на количество изображений в классе, а затем сравнить коэффициенты ошибок.

Следующий шаг — сделать все нули по диагонали, и это предотвратит появление ошибок.

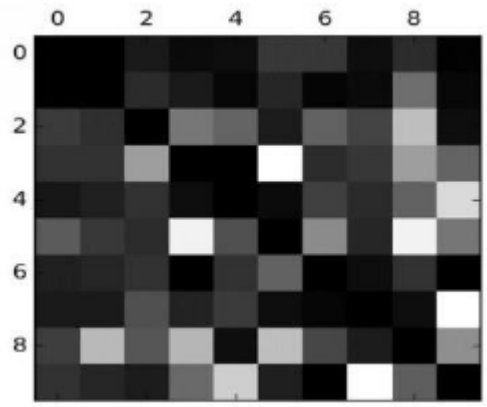


Рис. 13 . Результаты. Массив отображаемых результатов

В приведенном выше примере каждый класс имеет только один экземпляр. Но что, если мы хотим присвоить экземпляр какому-то классу — например, распознаванию лиц. Предположим, вы хотите найти более одного лица на одной фотографии. Для каждого лица будет одна метка. Давайте потренируемся на простом примере.

```
y_tr_big = (y_tr >= 7)
y_tr_odd = (y_tr %2 ==1)
y_multi = np.c [y_tr_big, y_tr_odd]
kng_cl = KNeighborsClassifier()
kng_cl.fit (x_tr, y_m,ulti)
```

В этом руководстве мы создали массив y_mullti, содержащий две метки для каждого изображения. Причем первый содержит информацию о том, является ли число «большим» (8,9,.), а второй проверяет на нечетность или нет. Далее мы будем делать прогнозы, используя следующий набор инструкций.

```
>>>kng_cl.predict([any-digit])
Array([false, true], dataType=bool)
```

Прямо здесь означает странный и неправильный, а не большой.

На этом этапе мы можем обсудить последний тип задач классификации, а именно многовыходную классификацию. Это всего лишь общий случай классификации с несколькими метками, но каждая метка будет иметь мультикласс. Dengan kata lain, itu akan memiliki lebih dari satu nilai. Mari kita perjelas dengan contoh ini, menggunakan gambar MNIST, dan menambahkan beberapa noise ke gambar dengan fungsi NumPy.

```
No = rnd.randint (0, 101, (len(x_tr), 785)))
No = rnd.randint(0, 101, (len(x_tes), 785))
```

Рисунок 14. Результаты

2.5 Выводы по разделу

В данном разделе обсуждаются различные популярные методы классификации машинного обучения с указанием их основного рабочего механизма, сильных и слабых сторон.

Также были выделены потенциальные приложения и проблемы с их доступными решениями. Методы классификации обычно сильны в моделировании взаимодействий. Обсуждаемые методы классификации могут применяться к разным типам наборов данных, например, к данным о здоровье, финансах и т. д.

Трудно определить, какой метод лучше другого, потому что каждый метод имеет свои достоинства, недостатки и проблемы с реализацией. Выбор метода классификации зависит от предметной области пользователя.

Тем не менее, в области классификации была проделана большая работа, но она по-прежнему требует формального внимания исследовательского сообщества для преодоления проблем классификации, которые возникают из-за решения новых проблем классификации, таких как проблемы классификации больших данных.

В этом разделе также рассматривается прогноз сердечно-сосудистых заболеваний на основе ОРЧ и КБС. Наш подход использует КБС в качестве классификатора, чтобы уменьшить количество ошибочных классификаций. В этой диссертации также исследуется мера выбора признаков на основе ОРЧ, чтобы выбрать небольшое количество признаков и повысить эффективность классификации. По результатам моделирования делается вывод, что выбор признаков на основе ОРЧ важен для классификации болезней сердца. Эта модель помогает врачам эффективно прогнозировать заболевания с преобладающими признаками.

3. Прогнозирование сердечных заболеваний с помощью машинного обучения

В этом разделе мы будем тесно работать с прогнозированием сердечных заболеваний, и для этого мы будем изучать набор данных о сердечных заболеваниях из этого набора данных, мы получим различные идеи, которые помогут нам узнать вес каждой функции и то, как они взаимосвязаны с друг друга, но на этот раз наша единственная цель - определить вероятность человека, который будет затронут проблемой сердца спасителя или нет.

Прогноз сердечно-сосудистых заболеваний будет иметь следующие ключевые выводы:

1. Анализ данных: как упоминалось здесь, мы будем работать с набором данных для выявления сердечных заболеваний и будем делать интересные выводы из данных, чтобы получить некоторые значимые результаты.
2. EDA: Исследовательский анализ данных является ключевым шагом для получения значимых результатов.
3. Разработка функций: после получения информации из данных мы должны изменить функции, чтобы они могли перейти к этапу построения модели.
4. Построение модели: на этом этапе мы будем строить нашу модель машинного обучения для выявления сердечных заболеваний.

Классификация может быть методом, который хочет назначить точки данных для сбора целевых категорий. Основная цель методов классификации заключается в том, насколько точно алгоритм может классифицировать все входные данные по меткам целевого класса. В нашем случае методы классификации используются, чтобы предсказать, склонен ли человек к сердечной недостаточности или не учитывая некоторые факторы риска.

3.1 Подготовка и сбор данных

Проект фокусируется в основном на трех методах интеллектуального анализа данных, а именно: (1) Логистическая регрессия, (2) КБС и (3) Метод случайных лесов. Точность проекта составляет 87,5%.

Цель проекта состоит в том, чтобы проверить, могут ли у пациента быть диагностированы какие-либо сердечно-сосудистые заболевания сердца на основе их медицинских характеристик, таких как пол, возраст, боль в груди, уровень сахара натощак и т.д. Набор данных выбирается из хранилища UCI с историей болезни и атрибутами пациента. Используя этот набор данных, проект предсказывает, может ли у пациента быть заболевание сердца или нет. Чтобы предсказать это, используется 14 медицинских характеристики пациента и классифицируются, если у пациента, вероятнее всего, есть заболевание сердца. Эти медицинские атрибуты обучаются с помощью трех алгоритмов: Логистической регрессии, КБС и классификатора случайных лесов. Наиболее эффективным из этих алгоритмов является КБС, который дает точность 88,52%.

Эффективное прогнозирование сердечно-сосудистых заболеваний было сделано с использованием различных алгоритмов, некоторые из которых включают логистическую Регрессию, КБС, Классификатор Случайных Лесов и т.д. Из результатов видно, что каждый алгоритм обладает своей силой для достижения определенных целей [1].

Организованный набор данных людей был отобран с учетом их истории проблем с сердцем и в соответствии с другими заболеваниями [2]. Болезни сердца - это разнообразные состояния, при которых поражается сердце. По данным Всемирной организации здравоохранения (ВОЗ), наибольшее число смертей у людей среднего возраста происходит из-за сердечно-сосудистых заболеваний. Берется источник данных, который состоит из истории болезни 304 разных пациентов разных возрастных групп. Этот набор данных дает необходимую информацию, т.е. медицинские характеристики, такие как возраст, кровяное давление в состоянии покоя, уровень сахара натощак и т.д. пациента, которые помогают в выявлении пациента, у которого диагностировано какое-либо заболевание сердца или нет. Набор данных содержит 13 медицинских характеристик 304 пациентов и берется из репозитория UCI. Записи разделены на две части: Обучение и тестирование. Также, набор данных содержит 303 строки и 14 столбцов, где каждая строка соответствует одной записи. Все атрибуты перечислены в "Таблице 1".

S. №	Наблюдение	Описание	Значения
1.	Возраст	Возраст в годах	Непрерывно
2.	Пол	Пол	Муж/Жен
3.	CP	Боль	Четыре
4.	Артериальное	давление в состоянии покоя	Непрерывно
5.	Chol	Холестерин в сыворотке	Непрерывно
6.	FBS	сахара	< или> 120 мг/дл
7.	Restecg	Электрокардиограмма	Пять значений
8.	Талаховая частота	максимум	Непрерывно
9.	Exang	Стенокардия	Нет
10.	Старая пиковая	во время тренировки по сравнению с продолжительностью отдыха	Непрерывно
11.	Да	ST	/Вниз
12.	Ca	Указывает количество крупных сосудов, окрашенных при рентгеноскопии	0-3

13.	Thal	тип	дефекта Обратимый/фиксированный/нормальный
14.	Num(расстройство)	Заболевание сердца отсутствует	/присутствует в четырех основных типах.

Таблица 1. Используемые атрибуты.

3.2 Модель для выявления сердечных аномалий

В исследовании показан анализ различных алгоритмов машинного обучения, которые используются в этой диссертации. Из них метод ближайших соседей (КБС), Логистическая регрессия и классификаторы случайных лесов, которые могут быть полезны практикующим врачам или медицинским аналитикам для точной диагностики ССЗ. Документация включает: изучение журналов, опубликованных статей и данных о ССЗ за последнее время. Методология дает основу для предлагаемой модели [3]. Методология - это процесс, включающий шаги, которые преобразуют данные в распознанные шаблоны данных для ознакомления пользователей. Предлагаемая методология (рис. 9.) включает этапы, где первый этап сбор данных, на втором этапе извлекаются значимые значения, на 3-м этапе осуществляется предварительная обработка, где исследуются сами данные. Предварительная обработка данных имеет дело с недостающими значениями, очисткой данных и нормализацией в зависимости от используемых алгоритмов [10]. После предварительной обработки данных, идет их классификация. Классификатором, используемым в предлагаемой модели, являются КБС, Логистическая регрессия, классификатор Случайных лесов. Наконец, идет предпринятие предлагаемая модель, где идет оценка модели на основе точности и производительности с использованием различных показателей производительности. Модель использует 13 медицинских параметров, таких как боль в груди, кровяное давление, уровень холестерина, возраст, пол и т.д. для прогнозирования [11] (рисунок 15).

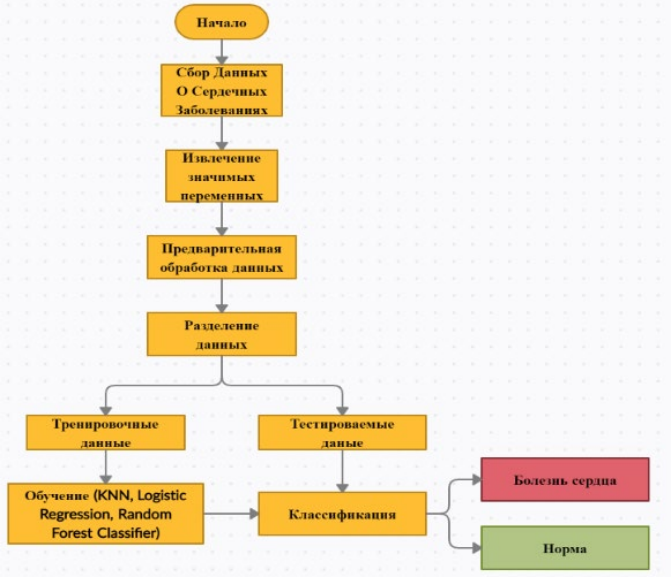


Рисунок 15. Предлагаемая Модель

3.3 Импорт необходимых библиотек

Библиотеки чертежей

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import cufflinks as cf
%matplotlib inline
```

Метрики для метода классификации

```
from sklearn.metrics import classification_report,confusion_matrix,accuracy_score
```

Скалер

```
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import  RandomizedSearchCV, train_test_split
```

Построение модели

```
from xgboost import XGBClassifier
from catboost import CatBoostClassifier
from sklearn.ensemble import RandomForestClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.svm import SVC
```

3.4 Загрузка данных

Здесь мы будем использовать функцию pandas read_csv для чтения набора данных . Укажите расположение набора данных и импортируйте их.

Импорт данных

```
  data = pd.read_csv("heart.csv")
  data.head(6) # Указать количество строк, которые
должны отображаться сверху в аргументе
```

Выход (рисунок 16):

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	thal	target
0	63	1	3	145	233	1	0	150	0	2.3	0	0	1	1
1	37	1	2	130	250	0	1	187	0	3.5	0	0	2	1
2	41	0	1	130	204	0	0	172	0	1.4	2	0	2	1
3	56	1	1	120	236	0	1	178	0	0.8	2	0	2	1
4	57	0	0	120	354	0	1	163	1	0.6	2	0	2	1
5	57	1	0	140	192	0	1	148	0	0.4	1	0	1	1

Рисунок 16. результаты

3.5 Исследовательский анализ данных

Теперь давайте посмотрим размер набора данных

```
  data.shape
```

Выход:

(303, 14)

Вывод: у нас есть набор данных с 303 строками, что указывает на меньший набор данных.

Как и выше, мы видели размер нашего набора данных, теперь давайте посмотрим тип каждой функции, которую содержит наш набор данных.

```
data.info()
```

Выход:

```
RangeIndex: 303 entries, 0 to 302
Data columns (total 14 columns):
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   age       303 non-null    int64
 1   sex       303 non-null    int64
 2   cp        303 non-null    int64
 3   trestbps  303 non-null    int64
 4   chol      303 non-null    int64
 5   fbs       303 non-null    int64
 6   restecg   303 non-null    int64
 7   thalach   303 non-null    int64
 8   exang     303 non-null    int64
 9   oldpeak   303 non-null    float64
 10  slope     303 non-null    int64
 11  ca        303 non-null    int64
 12  thal      303 non-null    int64
 13  target    303 non-null    int64
dtypes: float64(1), int64(13)
memory usage: 33.3 KB
```

Вывод: вывод, который мы можем сделать из приведенного выше вывода:

- Из 14 функций у нас есть 13 типов int и только одна с типами данных float.
- К счастью, в этом наборе данных нет пропущенных значений.

Поскольку мы получаем некоторую информацию от каждой функции, давайте посмотрим, как статистически распространяется набор данных.

```
data.describe()
```

Выход (рисунок 17):

	age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca
count	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000
mean	54.366337	0.683168	0.966997	131.623762	246.264026	0.148515	0.528053	149.646865	0.326733	1.039604	1.399340	0.729373
std	9.082101	0.466011	1.032052	17.538143	51.830751	0.356198	0.525860	22.905161	0.469794	1.161075	0.616226	1.022606
min	29.000000	0.000000	0.000000	94.000000	126.000000	0.000000	0.000000	71.000000	0.000000	0.000000	0.000000	0.000000
25%	47.500000	0.000000	0.000000	120.000000	211.000000	0.000000	0.000000	133.500000	0.000000	0.000000	1.000000	0.000000
50%	55.000000	1.000000	1.000000	130.000000	240.000000	0.000000	1.000000	153.000000	0.000000	0.800000	1.000000	0.000000
75%	61.000000	1.000000	2.000000	140.000000	274.500000	0.000000	1.000000	166.000000	1.000000	1.600000	2.000000	1.000000
max	77.000000	1.000000	3.000000	200.000000	564.000000	1.000000	2.000000	202.000000	1.000000	6.200000	2.000000	4.000000

Рисунок 17. Результаты

Всегда лучше проверять корреляцию между функциями, чтобы мы могли проанализировать, какая функция имеет отрицательную корреляцию, а какая положительную, поэтому давайте проверим корреляцию между различными функциями.

```
plt.figure(figsize=(20,12))
sns.set_context('notebook',font_scale = 1.3)
sns.heatmap(data.corr(),annot=True,linewidth =2)
plt.tight_layout()
```

Выход (рисунок 18) :

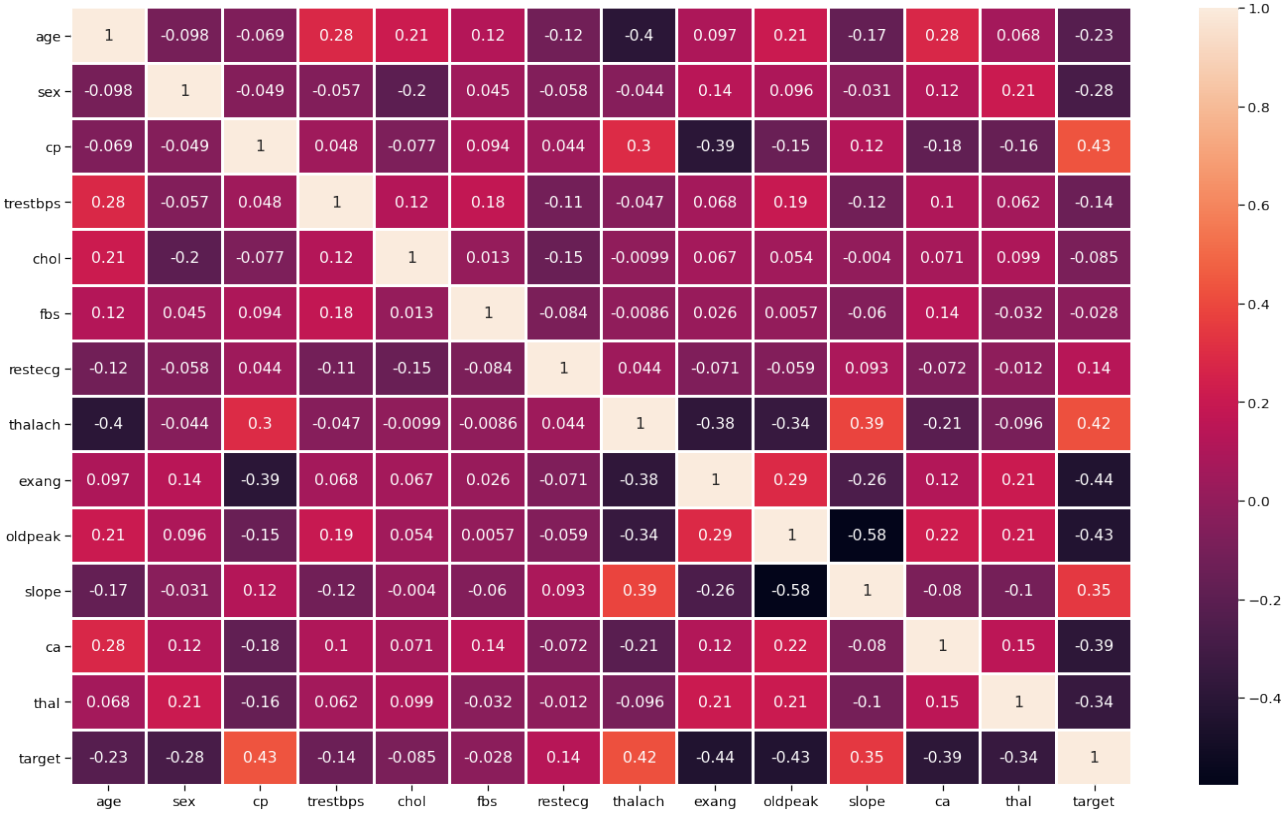


Рисунок 18. Результаты

До сих пор мы проверяли корреляцию между функциями, но также рекомендуется проверять корреляцию целевой переменной.

```
sns.set_context('notebook',font_scale = 2.3)
data.drop('target', axis=1).corrwith(data.target).plot(kind='bar', grid=True, figsize=(20, 10),
                                                       title="Correlation with the target feature")
plt.tight_layout()
```

Выход:

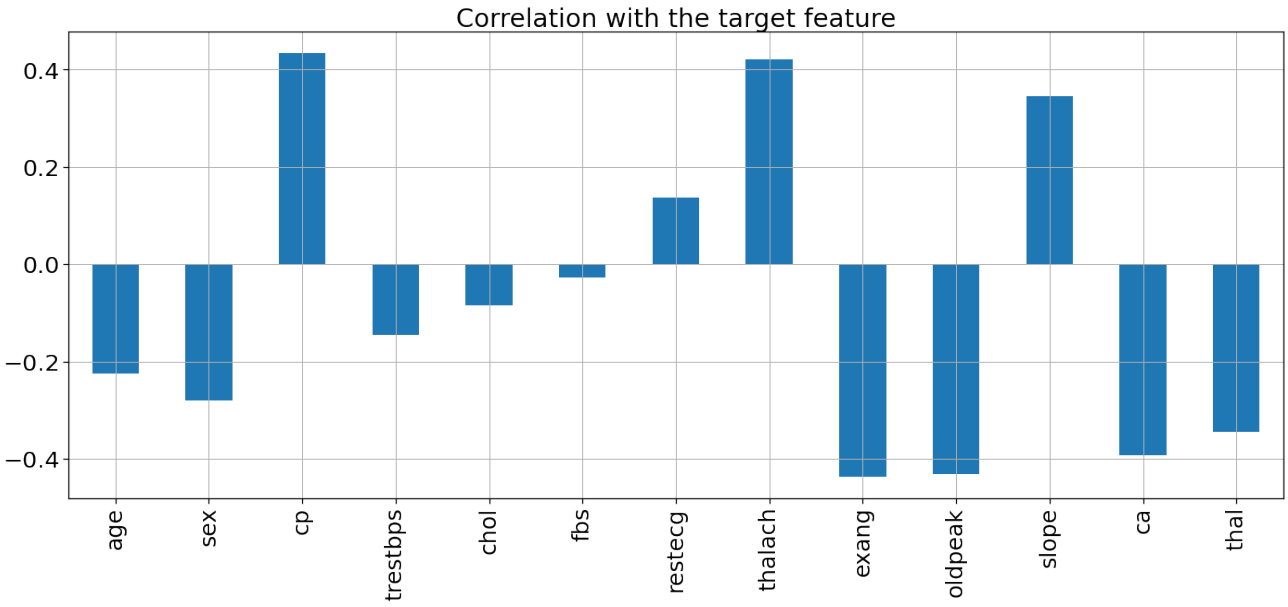


Рисунок 19. Результаты

Возраст («возраст») Анализ

Здесь мы будем проверять 10 возрастов и их количество.

```
plt.figure(figsize=(25,12))
sns.set_context('notebook',font_scale = 1.5)
sns.barplot(x=data.age.value_counts()[:10].index,y=data.age.value_counts()[:10].values)
plt.tight_layout()
```

Выход:

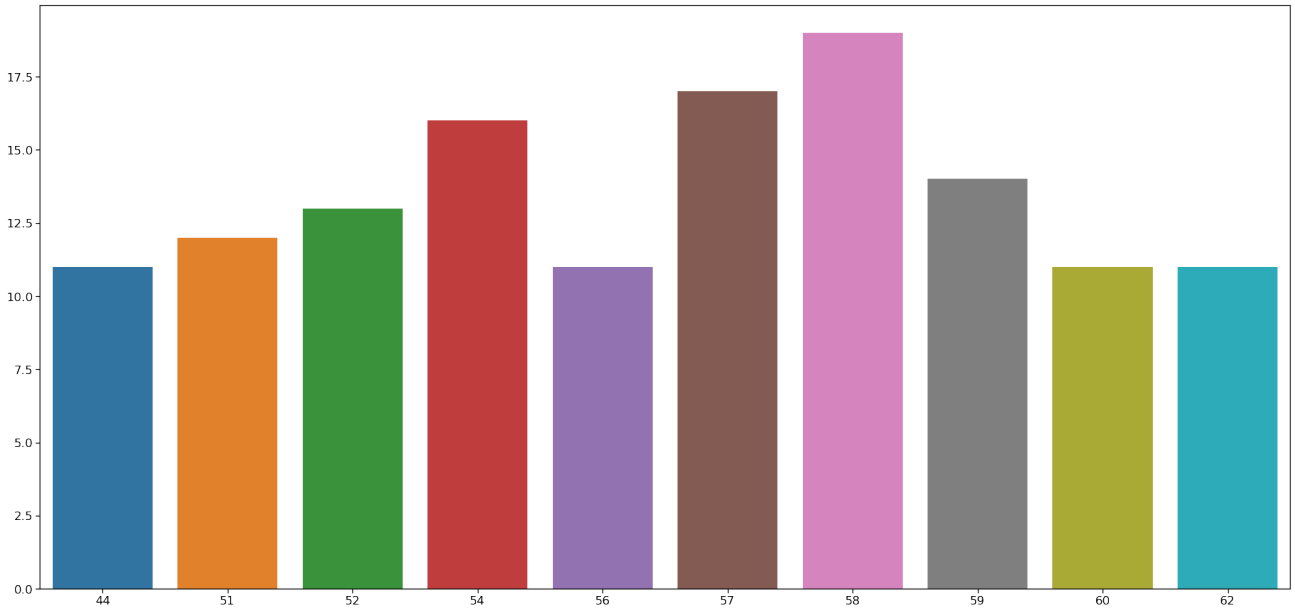


Рисунок 20. Результаты

Вывод: здесь мы видим, что столбец 58 лет имеет самую высокую частоту.

Давайте проверим возрастной диапазон в наборе данных.

```
minAge=min(data.age)
maxAge=max(data.age)
meanAge=data.age.mean()
print('Min Age :',minAge)
print('Max Age :',maxAge)
print('Mean Age :',meanAge)
```

Выход:

```
Min Age : 29 Max Age : 77 Mean Age : 54.366336633663366
```

Мы должны разделить функцию «Возраст» на три части – «Молодой», «Средний» и «Старший».

```
Young = data[(data.age>=29)&(data.age<40)]
Middle = data[(data.age>=40)&(data.age<55)]
Elder = data[(data.age>55)]

plt.figure(figsize=(23,10))
sns.set_context('notebook',font_scale = 1.5)
sns.barplot(x=['young ages','middle ages','elderly ages'],y=[len(Young),len(Middle),len(Elder)])
plt.tight_layout()
```

Выход (рисунок 21):

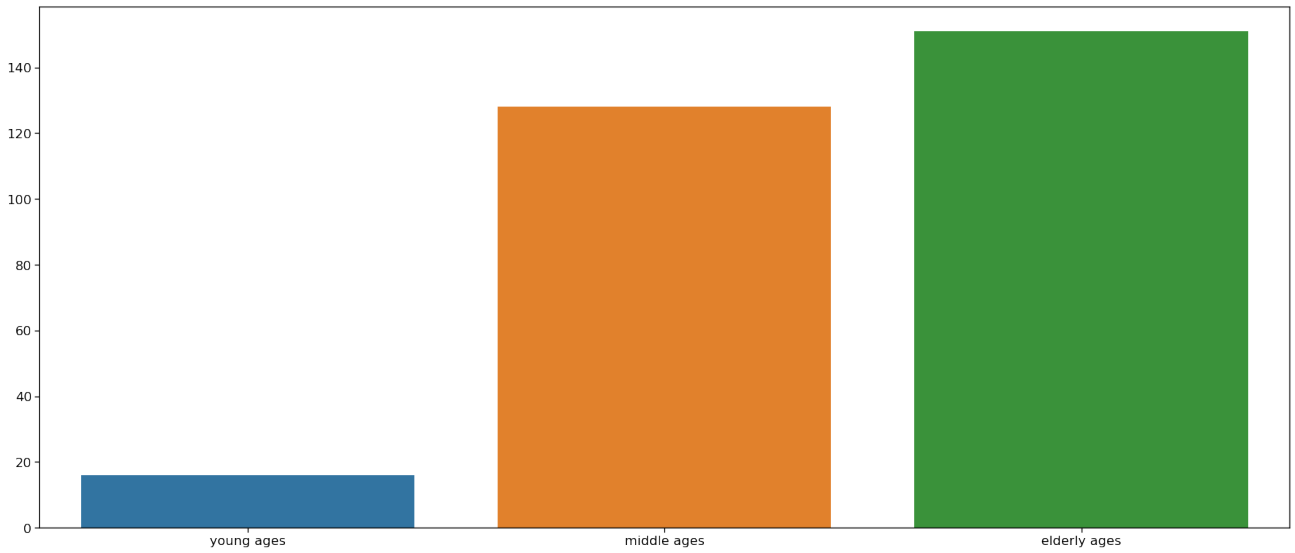


Рисунок 21. Результаты

Вывод: здесь мы видим, что пожилые люди больше всего страдают от сердечных заболеваний, а молодые — меньше всего.

Чтобы доказать приведенный выше вывод, мы построим круговую диаграмму.

```
    colors = ['blue','green','yellow']
    explode = [0,0,0.1]
    plt.figure(figsize=(10,10))
    sns.set_context('notebook',font_scale = 1.2)
    plt.pie([len(Young),len(Middle),len(Elder)],labels=['young ages','middle ages','elderly
ages'],explode=explode,colors=colors, autopct='%1.1f%%')
    plt.tight_layout()
```

Выход (Рисунок 22):

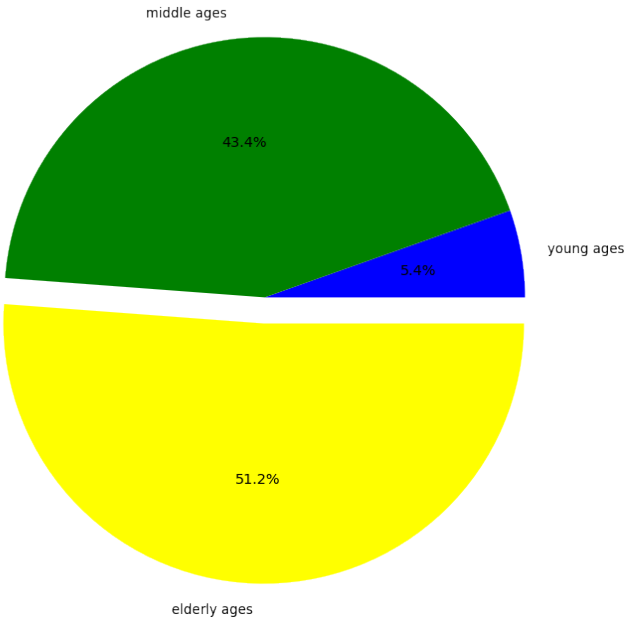


Рисунок 22. Результаты

Пол («пол») Анализ признаков

```
plt.figure(figsize=(18,9))
sns.set_context('notebook',font_scale = 1.5)
sns.countplot(data['sex'])
plt.tight_layout()
```

Выход (Рисунок 23):

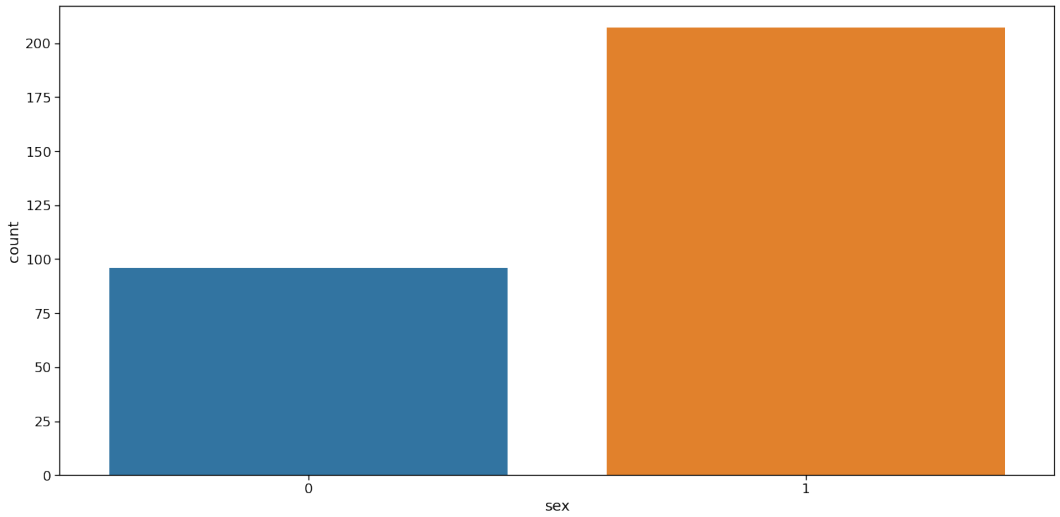


Рисунок 23. Результаты

Вывод : здесь ясно видно, что соотношение мужчин и женщин составляет примерно 2:1.

Теперь давайте построим зависимость между полом и наклоном.

```
plt.figure(figsize=(18,9))
sns.set_context('notebook',font_scale = 1.5)
sns.countplot(data['sex'],hue=data["slope"])
plt.tight_layout()
```

Выход (Рисунок 24):

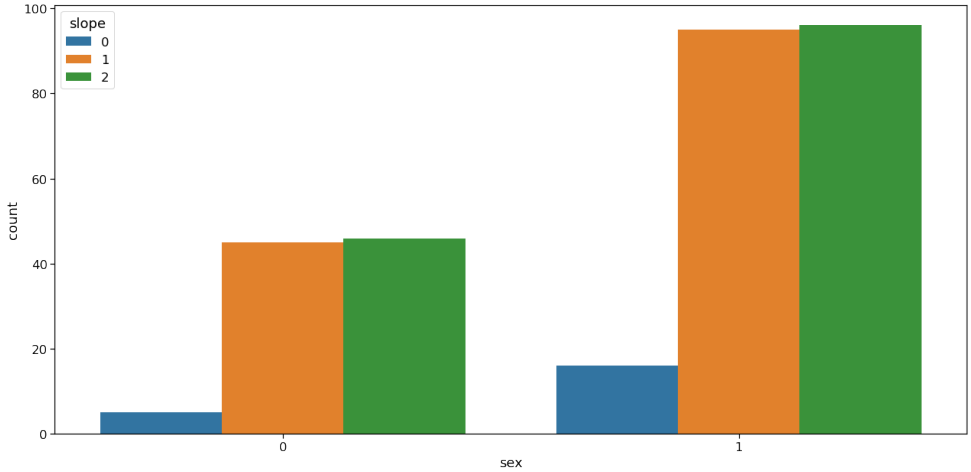


Рисунок 24. Результаты в виде групповой гистограммы

Вывод: здесь ясно видно, что значение наклона выше у самцов (1).

Анализ типа боли в груди («cp»)

```
plt.figure(figsize=(18,9))
sns.set_context('notebook',font_scale = 1.5)
sns.countplot(data['cp'])
plt.tight_layout()
```

Выход (Рисунок 25):

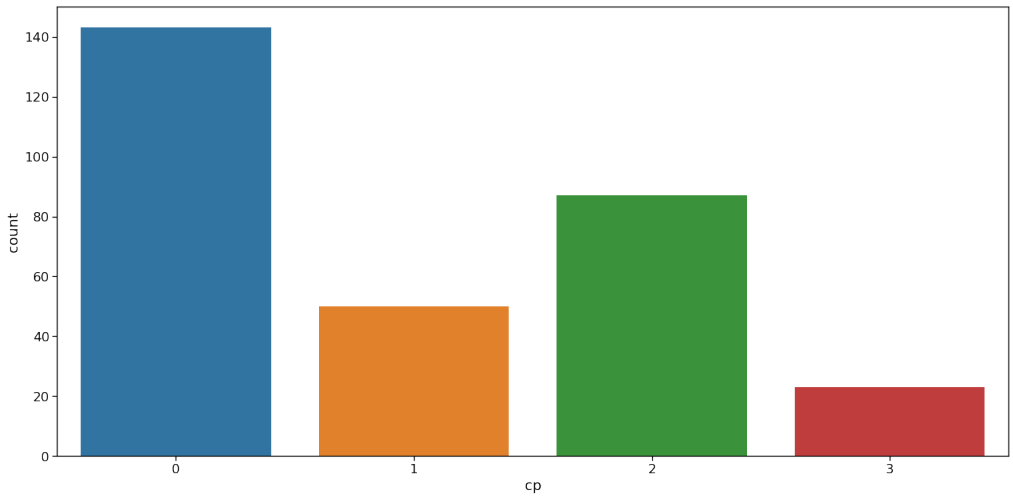


Рисунок 25. Результаты в виде гистограммы

Вывод: как видно, существует 4 типа боли в груди.

1. статус по крайней мере
2. состояние слегка угнетенное
3. состояние средняя проблема
4. состояние слишком плохое

Анализ ср по сравнению с целевым столбцом (рисунок 26)

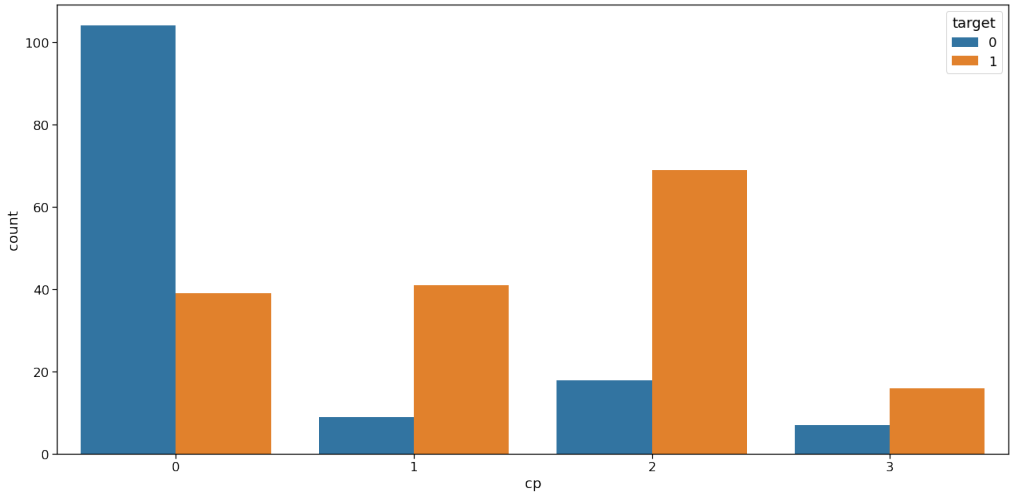


Рисунок 26. Результаты в виде графиков

Вывод: Из приведенного выше графика мы можем сделать некоторые выводы,

- Люди, испытывающие наименьшую боль в груди, скорее всего, не страдают сердечными заболеваниями.
- Люди, испытывающие сильную боль в груди, скорее всего, имеют сердечные заболевания.

Пожилые люди чаще испытывают боль в груди.

Анализ Таля

```
plt.figure(figsize=(18,9))
sns.set_context('notebook',font_scale = 1.5)
sns.countplot(data['thal'])
plt.tight_layout()
```

Выход Рисунок 27:

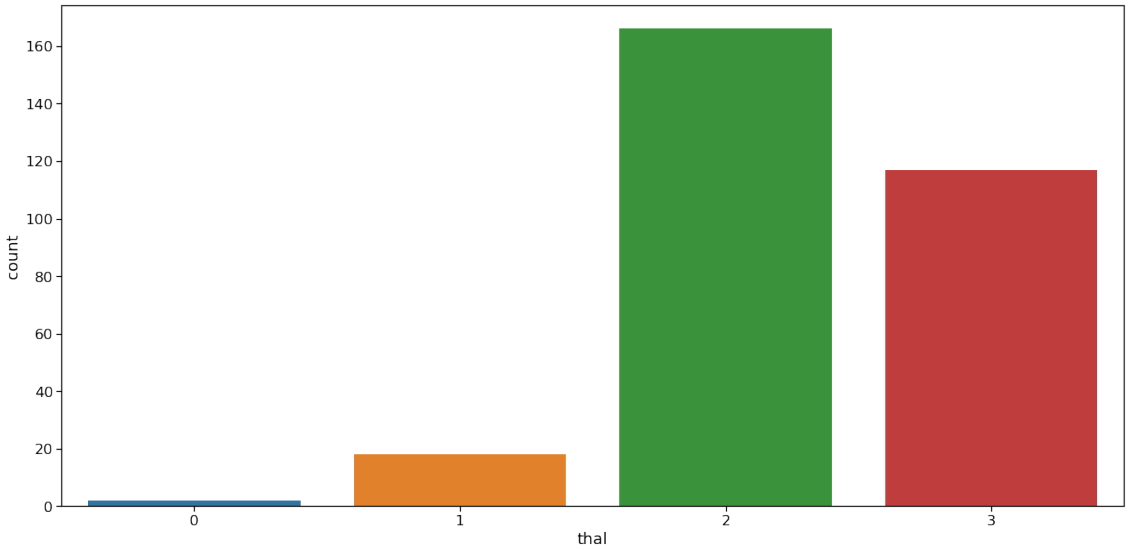


Рисунок 27. Результаты в виде графиков

Цель

```
plt.figure(figsize=(18,9))
sns.set_context('notebook',font_scale = 1.5)
sns.countplot(data['target'])
plt.tight_layout()
```

Выход Рисунок 28:

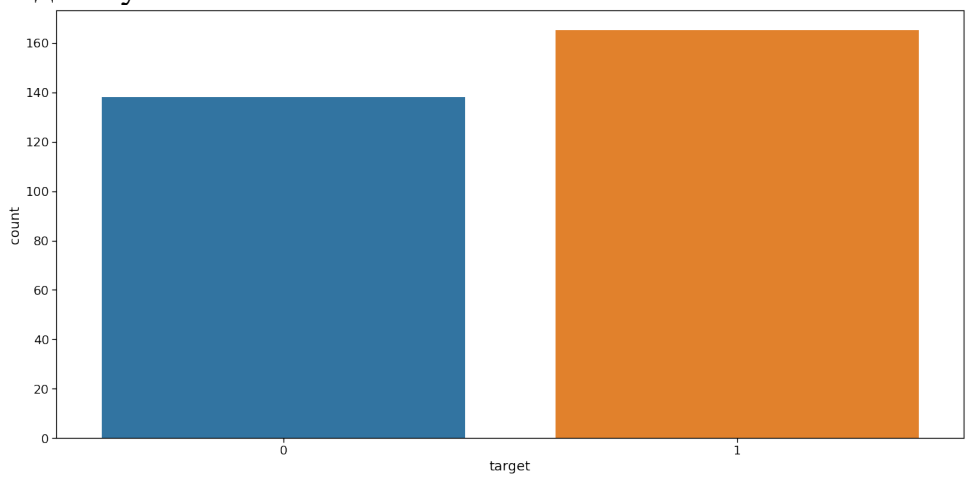


Рисунок 28. Результаты в виде графиков

Вывод: отношение между 1 и 0 намного меньше 1,5, что указывает на то, что целевая функция не несбалансирована. Таким образом, для сбалансированного набора данных мы можем использовать точность_оценки в качестве оценочной метрики для нашей модели.

3.6 Разработка функций

Теперь мы увидим полное описание непрерывных данных, а также категорийных данных.

```
categorical_val = []
continous_val = []
for column in data.columns:
    print("--------------------")
    print(f"{column} : {data[column].unique()}")
    if len(data[column].unique()) <= 10:
        categorical_val.append(column)
    else:
        continous_val.append(column)
```

Выход (рисунок 29):

```
--------------------
age : [63 37 41 56 57 44 52 54 48 49 64 58 50 66 43 69 59 42 61 40 71 51 65 53
 46 45 39 47 62 34 35 29 55 60 67 68 74 76 70 38 77]
--------------------
sex : [1 0]
--------------------
cp : [3 2 1 0]
--------------------
trestbps : [145 130 120 140 172 150 110 135 160 105 125 142 155 104 138 128 108 134
 122 115 118 100 124  94 112 102 152 101 132 148 178 129 180 136 126 106
 156 170 146 117 200 165 174 192 144 123 154 114 164]
--------------------
chol : [233 250 204 236 354 192 294 263 199 168 239 275 266 211 283 219 340 226
 247 234 243 302 212 175 417 197 198 177 273 213 304 232 269 360 308 245
 208 264 321 325 235 257 216 256 231 141 252 201 222 260 182 303 265 309
 186 203 183 220 209 258 227 261 221 205 240 318 298 564 277 214 248 255
 207 223 288 160 394 315 246 244 270 195 196 254 126 313 262 215 193 271
 268 267 210 295 306 178 242 180 228 149 278 253 342 157 286 229 284 224
 206 167 230 335 276 353 225 330 290 172 305 188 282 185 326 274 164 307
 249 341 407 217 174 281 289 322 299 300 293 184 409 259 200 327 237 218
 319 166 311 169 187 176 241 131]
--------------------
fbs : [1 0]
--------------------
restecg : [0 1 2]
--------------------
thalach : [150 187 172 178 163 148 153 173 162 174 160 139 171 144 158 114 151 161
 179 137 157 123 152 168 140 188 125 170 165 142 180 143 182 156 115 149
 146 175 186 185 159 130 190 132 147 154 202 166 164 184 122 169 138 111
 145 194 131 133 155 167 192 121  96 126 105 181 116 108 129 120 112 128
 109 113  99 177 141 136  97 127 103 124  88 195 106  95 117  71 118 134
  90]
```

Рисунок 29 Выходные данные

Теперь здесь сначала мы удалим целевой столбец из нашего набора функций , затем мы классифицируем все категориальные переменные, используя метод get фиктивных, который создаст отдельный столбец для каждой категории, предположим, что переменная X содержит 2 типа уникальных значений, тогда она создаст 2 разных столбца для переменной X.

```
categorical_val.remove('target')
dfs = pd.get_dummies(data, columns = categorical_val)
dfs.head(6)
```

Выход (рисунок 30):

	age	trestbps	chol	thalach	oldpeak	target	sex_0	sex_1	cp_0	cp_1	...	slope_2	ca_0	ca_1	ca_2	ca_3	ca_4	thal_0	thal_1	thal_2	thal_3
0	63	145	233	150	2.3	1	0	1	0	0	...	0	1	0	0	0	0	0	1	0	0
1	37	130	250	187	3.5	1	0	1	0	0	...	0	1	0	0	0	0	0	0	1	0
2	41	130	204	172	1.4	1	1	0	0	1	...	1	1	0	0	0	0	0	0	1	0
3	56	120	236	178	0.8	1	0	1	0	1	...	1	1	0	0	0	0	0	0	1	0
4	57	120	354	163	0.6	1	1	0	1	0	...	1	1	0	0	0	0	0	0	1	0
5	57	140	192	148	0.4	1	0	1	1	0	...	0	1	0	0	0	0	0	1	0	0

6 rows × 31 columns

Рисунок 30. Выходные данные

Теперь мы будем использовать стандартный метод масштабирования для уменьшения данных, чтобы он не увеличивал выбросы, а набор данных, масштабированный до общих единиц, обеспечивает лучшую точность.

```
sc = StandardScaler()
col_to_scale = ['age', 'trestbps', 'chol', 'thalach', 'oldpeak']
dfs[col_to_scale] = sc.fit_transform(dfs[col_to_scale])
dfs.head(6)
```

Выход (рисунок 31):

	age	trestbps	chol	thalach	oldpeak	target	sex_0	sex_1	cp_0	cp_1	...	slope_2	ca_0	ca_1	ca_2	ca_3	ca_4	thal_0	thal_1	thal_2
0	0.952197	0.763956	-0.256334	0.015443	1.087338	1	0	1	0	0	...	0	1	0	0	0	0	0	1	0
1	-1.915313	-0.092738	0.072199	1.633471	2.122573	1	0	1	0	0	...	0	1	0	0	0	0	0	0	1
2	-1.474158	-0.092738	-0.816773	0.977514	0.310912	1	1	0	0	1	...	1	1	0	0	0	0	0	0	1
3	0.180175	-0.663867	-0.198357	1.239897	-0.206705	1	0	1	0	1	...	1	1	0	0	0	0	0	0	1
4	0.290464	-0.663867	2.082050	0.583939	-0.379244	1	1	0	1	0	...	1	1	0	0	0	0	0	0	1
5	0.290464	0.478391	-1.048678	-0.072018	-0.551783	1	0	1	1	0	...	0	1	0	0	0	0	0	1	0

6 rows × 31 columns

Рисунок 31. Выходные данные

Моделирование

Разделение нашего набора данных

```
X = dfs.drop('target', axis=1)
y = dfs.target

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0,3, random_state=42)
```

Алгоритм машинного обучения KNN

```
knn = KNeighborsClassifier (n_neighbors = 10)
knn.fit (X_train, y_train)
y_pred1 = knn.predict (X_test)
print (accuracy_score (y_test, y_pred1))
```

Выход:

0,8571428571428571

Из результатов видно, что КБС, Классификатор Случайных лесов и логистическая регрессия дают лучший результат, превосходящий остальные [14]. Алгоритмы, которые использовались, более точны, экономят много денег, т.е. они экономически эффективны и быстрее. Более того, максимальная точность, полученная с помощью КБС и логистической регрессии, равна 88,5%, что больше или почти равно точности, полученной в результате предыдущих исследований. Итак, итог показывает, что точность улучшилась благодаря увеличению медицинских характеристик, которые были использованы из взятого набора данных. Проект также сообщает, что Логистическая регрессия и КБС превосходят классификатор случайных лесов в прогнозировании пациента с диагнозом сердечного заболевания. Это доказывает, что КБС и логистическая регрессия лучше подходят для диагностики сердечных заболеваний. На следующих "рисунок 27, ‘рисунок 28’, ‘рисунок 29’ показан график из числа пациентов, которые были разделены и предсказаны классификатором в зависимости от возрастной группы, артериального давления в состоянии покоя, пола, боли в груди:

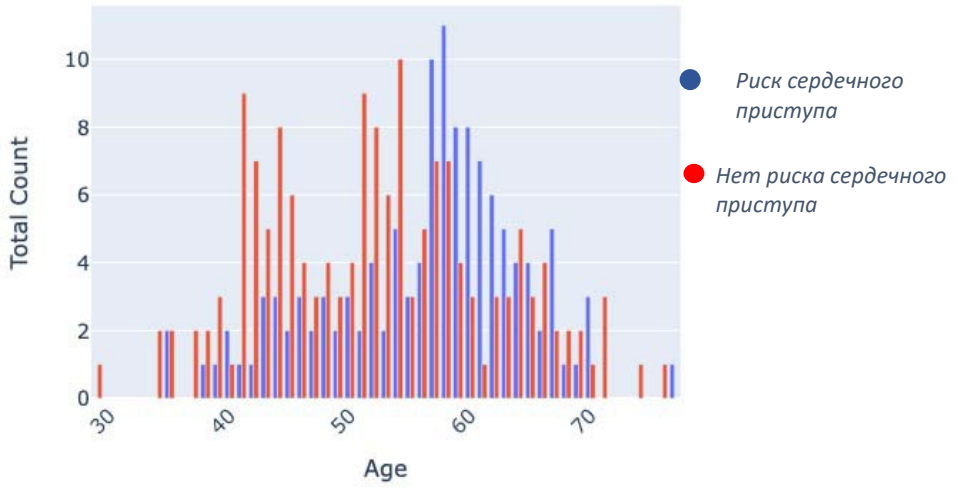


Рисунок 32. Риск сердечного приступа в зависимости от возраста.

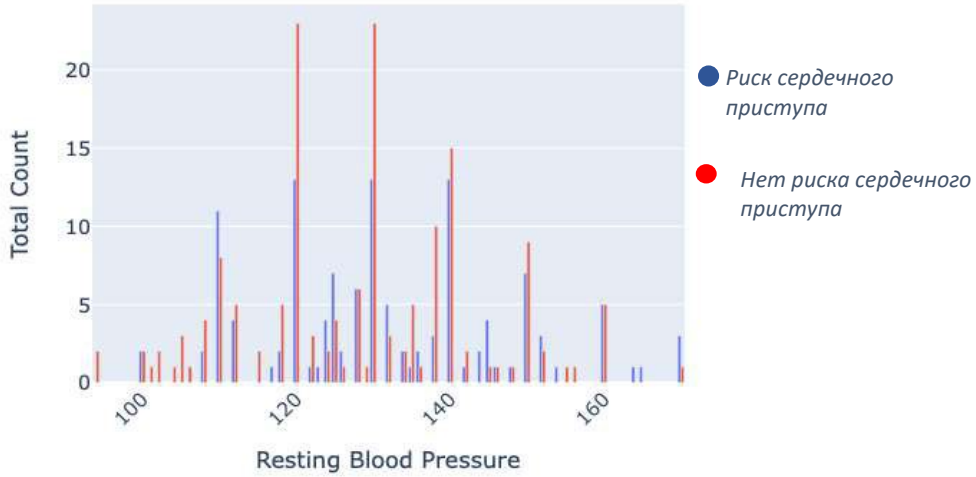


Рисунок 33. Риск сердечного приступа на основе данных кровяного давления.

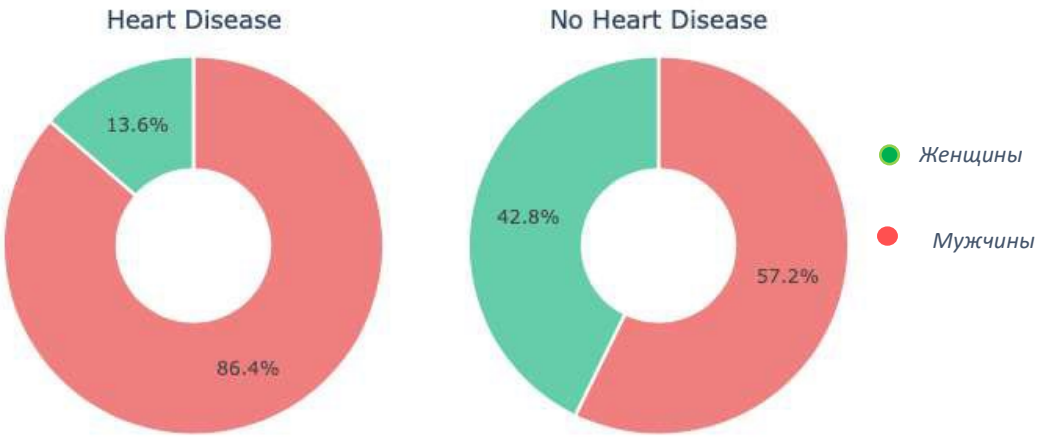


Рисунок 34. Пациенты, имеющие/не имеющие сердечные заболевания по гендеру

3.7 Выводы по третьему разделу

Заключение о прогнозировании сердечно-сосудистых заболеваний

1. Мы провели визуализацию данных и анализ данных целевой переменной, возрастных особенностей и многого другого вместе с одномерным и двумерным анализом.

2. В этом разделе мы также сделали полную часть разработки функций, в которой перечислены все допустимые шаги, необходимые для дальнейших шагов, например , для построения модели.

3. Исходя из приведенной выше точности модели, KNN дает нам точность, которая составляет 89%

Заключение

В диссертационной работе были решены следующие **задачи:**

– рассмотрены теоретические основы машинного обучения

– произведён сравнительный анализ методов и средств машинного обучения для выявления сердечных аномалий

– разработана система прогнозирования сердечных заболеваний, чтобы предсказать, будет ли у пациента диагностировано сердечное заболевание или нет используя историю болезни пациента.

Сердечно-сосудистые заболевания являются одной из серьезных проблем в современном мире и одной из ведущих причин многих смертей во всем мире. Недавнее развитие приложений машинного обучения (МО) демонстрирует, что с помощью электрокардиограммы (ЭКГ) и данных пациентов можно выявлять сердечные заболевания на ранней стадии. Тем не менее, как ЭКГ, так и данные пациентов часто несбалансированы, что в конечном итоге затрудняет непредвзятую работу традиционного машинного обучения. За прошедшие годы многие исследователи и практики предложили несколько решений на уровне данных и алгоритмов. Чтобы обеспечить более широкий взгляд на существующую литературу, в этом исследовании используется подход систематического обзора литературы (SLR), чтобы выявить проблемы, связанные с несбалансированными данными в прогнозах сердечных заболеваний. До этого, мы провели метаанализ с использованием 40 ссылочной литературы, полученной из авторитетных журналов в период с 2012 г. по 15 ноября 2021 г. Для углубленного анализа было рассмотрено и изучено 40 ссылочной литературы с учетом следующих факторов: тип заболевания сердца, алгоритмы , приложения и решения. Наше исследование SLR показало, что текущие подходы сталкиваются с различными открытыми проблемами/проблемами при работе с несбалансированными данными, что в конечном итоге препятствует их практическому применению и функциональности.

Обсуждены различные популярные методы классификации машинного обучения с указанием их основного рабочего механизма, сильных и слабых сторон. Также были выделены потенциальные приложения и проблемы с их доступными решениями. Методы классификации обычно сильны в моделировании взаимодействий. Обсуждаемые методы классификации могут применяться к разным типам наборов данных, например, к данным о здоровье, финансах и т. д. Трудно определить, какой метод лучше другого, потому что каждый метод имеет свои достоинства, недостатки и проблемы с реализацией. Выбор метода классификации зависит от предметной области пользователя. Тем не менее, в области классификации была проделана большая работа, но она по-прежнему требует формального внимания исследовательского сообщества для преодоления проблем классификации, которые возникают из-за решения новых проблем классификации, таких как проблемы классификации больших данных.

Рассмотрены прогноз сердечно-сосудистых заболеваний на основе ОРЧ и КБС. Наш подход использует КБС в качестве классификатора, чтобы уменьшить количество ошибочных классификаций. В этой диссертации также исследуется мера выбора признаков на основе ОРЧ, чтобы выбрать небольшое количество признаков и повысить эффективность классификации. По результатам моделирования делается вывод, что выбор признаков на основе ОРЧ важен для классификации болезней сердца. Эта модель помогает врачам эффективно прогнозировать заболевания с преобладающими признаками.

Модель выявления сердечно-сосудистых заболеваний была разработана с использованием трех методов моделирования классификации МО. Проект прогнозирует людей с сердечно-сосудистыми заболеваниями путем извлечения истории болезни пациента, которая приводит к смертельному заболеванию сердца, из набора данных, включающего историю болезни пациентов, такую как боль в груди, уровень сахара и т.д. Данная система выявления сердечных заболеваний помогает на основе клинической информации диагностировать заболевание сердца. Алгоритмы, используемые при построении данной модели, представляет собой логистическую регрессию, метод случайных лесов и КБС [13]. Точность модели составляет 87,5%. Использование большего количества обучающих данных обеспечивает более высокие шансы модели точно предсказать, есть ли у данного человека заболевание сердца или нет [1]. Используя эти методы, можно быстро и лучше прогнозировать состояние пациента, а затраты могут быть значительно снижены. Таким образом, в заключение этого, проект помогает прогнозировать пациентов, у которых диагностированы сердечные заболевания, путем очистки набора данных и применения логистической регрессии и КБС, чтобы получить точность в среднем 87,5% по данной модели. Кроме того, вывод, что точность КБС является самой высокой среди трех алгоритмов, которые были использованы, т.е. 88,52%. "Рисунок 13" показывает, что 44% людей, перечисленных в наборе данных, страдают сердечными заболеваниями (рисунок 35).

Рисунок 35. Статистика пациентов, имеющих/не имеющих сердечные заболевания.

Заключение о прогнозировании сердечно-сосудистых заболеваний. Мы провели визуализацию данных и анализ данных целевой переменной, возрастных особенностей и многого другого вместе с одномерным и двумерным анализом. мы также сделали полную часть разработки функций, в которой перечислены все допустимые шаги, необходимые для дальнейших шагов, например, для построения модели. Исходя из приведенной выше точности модели, KNN дает нам точность, которая составляет 89%

Использованная литература

1 Сони Дж., Ансари У., Шарма Д. и Сони С. (2011). Прогностический анализ данных для медицинской диагностики: обзор прогнозирования сердечных заболеваний. Международный журнал компьютерных приложений, 17 (8), 43–8.

2 Дангаре К.С. и Апте С.С. (2012). Улучшенное исследование системы прогнозирования сердечных заболеваний с использованием методов классификации интеллектуального анализа данных. Международный журнал компьютерных приложений, 47(10), 44-8.

3 Башир С., Камар У. и Джавед М.Ю. (2014 г., ноябрь). Система поддержки принятия решений на основе ансамбля для интеллектуальной диагностики заболеваний сердца. В Международной конференции по информационному обществу (i-Society 2014) (стр. 259-64). IEEE.

4 Джи Ш., Чан И, О Ди Джей, О Би Х, Ли Ш, Пак С В и Юн И Д (2014). Модель прогнозирования ишемической болезни сердца: Корейское исследование сердца. BMJ открытый, 4(5), e005025.

5 Ганна А., Магнуссон П.К., Педерсен Н.Л., де Файр Ю., Рейли М., Эрнлёв Дж. и Ингельссон Э. (2013). Мультилокусные оценки генетического риска для прогнозирования ишемической болезни сердца. Артериосклероз, тромбоз и биология сосудов, 33(9), 2267-72.

6 Дангаре Чайтрали С. и Сулабха С. Апте. «Улучшенное исследование системы прогнозирования сердечных заболеваний с использованием методов классификации интеллектуального анализа данных». Международный журнал компьютерных приложений 47.10 (2012): 44-8.

7 Сони Джиоти. «Прогнозный анализ данных для медицинской диагностики: обзор прогнозирования сердечных заболеваний». Международный журнал компьютерных приложений 17.8 (2011): 43-8.

8 Chen A H, Huang S Y, Hong P S, Cheng CH & Lin E J (2011, сентябрь). HDPS: система прогнозирования сердечных заболеваний. В 2011 г. Компьютеры в кардиологии (стр. 557–60). IEEE.

9 Вольгаст Г., Эренборг С., Исраэльссон А., Хеландер Дж., Йоханссон Э. и Манефьорд Х. (2016). Беспроводная сеть на теле для обнаружения сердечного приступа [Образовательный уголок]. Антенны IEEE и журнал распространения, 58 (5), 84–92.

10 Чжан Ю., Фогорос Р., Томпсон Дж., Кеннайт Б.Х., Педерсон М.Дж., Патангей А. и Мазар С.Т. (2011). Патент США № 8,014,863. Вашингтон, округ Колумбия: Ведомство США по патентам и товарным знакам.

11 Бюхлер К.Ф. и Макферсон П.Х. (1999). Патент США № 5,947,124. Вашингтон, округ Колумбия: Ведомство США по патентам и товарным знакам.

12 Ачарья У. Р., Фудзита Х., О С. Л., Хагивара Ю., Тан Дж. Х. и Адам М. (2017). Применение глубокой сверточной нейронной сети для автоматизированного обнаружения инфаркта миокарда по сигналам ЭКГ. Информационные науки, 415, 190-8.

13 Пиллер Л.Б., Дэвис Б.Р., Катлер Дж.А., Кушман В.К., Райт Дж.Т., Уильямсон Дж.Д. и Хейвуд Л.Дж. (2002). Валидация событий сердечной недостаточности у участников антигипертензивного и гиполипидемического лечения для предотвращения сердечного приступа (ALLHAT), назначенных доксазозину и хлорталидону. Текущие контролируемые испытания в сердечно-сосудистой медицине, 3(1), 10.
14 ФМНКом А.Р., Принеас Р.Дж., Кей С.А. и Солер Дж.Т. (1989). Распределение телесного жира и самооценка распространенности гипертонии, сердечного приступа и других сердечных заболеваний у пожилых женщин. Международный журнал эпидемиологии, 18(2), 361-7.
15 Буй А.Л., Хорвич Т.Б. и Фонароу Г.К., «Эпидемиология и профиль риска сердечной недостаточности», Nature Reviews Cardiology, vol. 8, нет. 1, стр. 30–41, 2011 г.
16 Манприт Сингх, Леви Монтейро Мартинс, Патрик Джоанис и Виджай К. Маго, 2016 г., «Построение прогностической модели сердечно-сосудистых заболеваний с использованием модели структурного уравнения и нечеткой когнитивной карты», Международная конференция IEEE по нечетким системам (FUZZ), стр. 1377-1382 гг.
17 Рустам и Т. В. Рамписела, Классификация данных о шизофрении с использованием метода опорных векторов (SVM), Journal of Physics: Conference Series 1108(1), 012038 (2018 г.).
18 Шейх Абдул Ханнан, А.В. Мане, Р. Р. Манза и Р. Дж. Рамтеке, декабрь 2010 г., «Прогнозирование назначения врача по болезни сердца с использованием функции радиального базиса», Международная конференция IEEE по вычислительному интеллекту и компьютерным исследованиям (ICCIC), DOI: 10.1109/ICCIC.2010.5705900, 28-29
19 Н. Баракат, А. П. Брэдли и М. Н. Баракат, Интеллектуальные машины опорных векторов для диагностики сахарного диабета, IEEE Engineering in Medical and Biology Society (2010)
20 Д. Б. Манурунг, Б. Дирганторо и К. Сетианингсих, Распознавание говорящего для цифрового криминалистического анализа звука с использованием метода квантования обучающего вектора, Международная конференция IEEE
21 20-е место в Интернете вещей и интеллектуальной системе (IoTaIS) (2018 г.)
22 О. В. Самуэль, Г. М. Асогбон, А. К. Сангайах, П. Фанг и Г. Ли, «Интегрированная система поддержки принятия решений на основе ANN и Fuzzy_AHP для прогнозирования риска сердечной недостаточности», Expert Systems with Applications, vol. 68, стр. 163–172, 2017.
23 Херон, М. Смерти: основные причины в 2017 г. . Национальные статистические отчеты о естественном движении населения;68(6). По состоянию на 19 ноября 2019 г.
24 Бенджамин Э.Дж., Мантнер П., Алонсо А., Биттенкур М.С., Каллауэй К.В., Карсон А.П. и другие. Статистика сердечных заболеваний и инсультов — обновление 2019 г.: отчет Американской кардиологической ассоциации. Тираж. 2019;139(10):e56–528.

25 Fryar CD, Chen T-C, Li X. Распространенность неконтролируемых факторов риска сердечно-сосудистых заболеваний: США, 1999–2010 гг. Сводка данных NCHS, нет. 103. Hyattsville, MD: Национальный центр статистики здравоохранения; 2012.
26 Fryar CD, Chen T-C, Li X. Prevalence of uncontrolled risk factors for cardiovascular disease: United States, 1999–2010 . NCHS data brief, no. 103. Hyattsville, MD: National Center for Health Statistics; 2012. Accessed May 9, 2019.
27 Dua, D. and Graff, C. (2019). UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science.
28 Hungarian Institute of Cardiology. Budapest: Andras Janosi, M.D.
29 University Hospital, Zurich, Switzerland: William Steinbrunn, M.D.
30 University Hospital, Basel, Switzerland: Matthias Pfisterer, M.D.
31 V.A. Medical Center, Long Beach and Cleveland Clinic Foundation:Robert Detrano, M.D., Ph.D.
32 MHKon, Randal S. et al. "Data-driven advice for applying machine learning to bioinformatics problems." Pacific Symposium on Biocomputing. Pacific Symposium on Biocomputing 23 (2017): 192–203.
33 **SciPy.** Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, CJ Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E.A. Quintero, Charles R Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors. (2019) SciPy 1.0–Fundamental Algorithms for Scientific Computing in Python. preprint arXiv:1907.10121
34 **Python.** a) Travis E. Oliphant. Python for Scientific Computing, Computing in Science & Engineering, 9, 10–20 (2007) b) K. Jarrod Millman and Michael Aivazis. Python for Scientists and Engineers, Computing in Science & Engineering, 13, 9–12 (2011)
35 Hazra, A., Mandal, S., Gupta, A. and Mukherjee, A. (2017) Heart Disease Diagnosis and Prediction Using Machine Learning and Data Mining Techniques: A Review. Advances in Computational Sciences and Technology, 10, 2137-2159.
36 Patel, J., Upadhyay, P. and Patel, D. (2016) Heart Disease Prediction Using Machine learning and Data Mining Technique. Journals of Computer Science & Electronics, 7, 129-137.
37 Chavan Patil, A.B. and Sonawane, P. (2017) To Predict Heart Disease Risk and Medications Using Data Mining Techniques with an IoT Based Monitoring System for Post-Operative Heart Disease Patients. International Journal on Emerging Trends in Technology (IJETT), 4, 8274-8281.
38 Weng, S.F., Reps, J., Kai, J., Garibaldi, J.M. and Qureshi, N. (2017) Can Machine-Learning Improve Cardiovascular Risk Prediction Using Routine Clinical Data? PLoS ONE, 12, e0174944.

39 https://doi.org/10.1371/journal.pone.0174944
40 Zhao, W., Wang, C. and Nakahira, Y. (2011) Medical Application on Internet of Things. IET International Conference on Communication Technology and Application (ICCTA 2011), Beijing, 14-16 October 2011, 660-665.
41 Chiuchisan, I. and Geman, O. (2014) An Approach of a Decision Support and Home Monitoring System for Patients with Neurological Disorders Using Internet of Things Concepts. WSEAS Transactions on Systems, 13, 460-469.
42 Soni, J., Ansari, U. and Sharma, D. (2011) Intelligent and Effective Heart Disease Prediction System Using Weighted Associative Classifiers. International Journal on Computer Science and Engineering (IJCSE), 3, 2385-2392.

SATBAYEV UNIVERSITY

СӘТБАЕВ ОҚУЛАРЫ
САТПАЕВСКИЕ ЧТЕНИЯ
SATBAYEV CONFERENCE

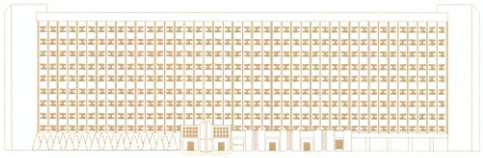
Сертификат
Certificate

«СӘТБАЕВ ОҚУЛАРЫ-2022. ҚАЗІРГІ ҒЫЛЫМИ ЗЕРТТЕУЛЕРДІҢ ТРЕНДТЕРІ»
«SATBAYEV CONFERENCE-2022. TRENDS IN MODERN SCIENTIFIC RESEARCH»

Халықаралық ғылыми-практикалық конференцияға жоғарғы деңгейлі
International research and Practice conference on the topic

«Выявление сердечных аномалий с помощью методов машинного обучения»
Detection of cardiac abnormalities using machine learning methods
(Абжанова К.Н.)

атты мазмұнды баяндама ұсынғаны үшін беріледі.
for providing a high-level semantic report.



Басқарма мүшесі -
Ғылым және халықаралық
ынтымақтастық жөніндегі проректор
Ә. Шоқпаров

АЛМАТЫ | 2022

Приложение
Сертификат участия в международной научно-практической конференции