МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ КАЗАХСТАН

СӘТБАЕВ УНИВЕРСИТЕТІ

Институт информационных и телекоммуникационных технологий

Кафедра «Программная инженерия»

ДОПУЩЕН К ЗАЩИТЕ
Заведующий кафедрой ПИ
Доктор Ph. D.
_____ М. Турдалыулы
«_6_» ____июня_____2021 г.

ПОЯСНИТЕЛЬНАЯ ЗАПИСКА
к дипломному проекту

На тему: «Применение методов глубокого обучения на медицинских данных»

по специальности 5В070400 – Вычислительная техника и программное обеспечение

Выполнил Еленов А.М.

Научный руководитель
Магистр технических наук,
Сениор-лектор
_____ А.Д. Куникеев
«_6_» ____июня_____2021 г.

Алматы 2021

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ КАЗАХСТАН

СӘТБАЕВ УНИВЕРСИТЕТІ

Институт информационных и телекоммуникационных технологий

Кафедра «Программная инженерия»

5В070400 – Вычислительная техника и программное обеспечение

УТВЕРЖДАЮ
Заведующий кафедрой ПИ
Доктор Ph. D.
_____ М. Турдалыулы
«_6_» _____июня_____2021 г.

ЗАДАНИЕ
на выполнение дипломного проекта

Обучающемуся *Еленову Амир Мирхатовичу*
Тема: *Применение методов глубокого обучения на медицинских данных*
Исходные данные к дипломному проекту: *Описание необходимых функций проекта.*
Перечень подлежащих разработке в дипломном проекте вопросов:
а) оценка текущих методов машинного обучения;
б) реализация теоретической части проекта и выбор нужного подхода;
в) предобработка данных полученных из открытых источников;

Перечень графического материала (с точным указанием обязательных чертежей): представлены в 22 слайдах презентации.

ГРАФИК

подготовки дипломного проекта

Наименование разделов, перечень разрабатываемых вопросов	Сроки представления научному руководителю и консультантам	Примечание
1. Анализ предметной области и релевантности проекта	09.04.21	Выполнено
2. Анализ современных и классических методов машинного обучения	17.04.21	Выполнено
3. Выбор и предобработка данных	30.04.21	Выполнено
4. Применение классических методов машинного обучения	01.05.21	Выполнено
5. Выбор оптимальных параметров и использование методов глубокого обучения	11.05.21	Выполнено
6. Оценка и использования метрик для оценки работы алгоритма	15.05.21	Выполнено
7. Написание пояснительной записки к дипломному проекту	20.05.21	Выполнено

Подписи

консультантов и нормоконтролера на законченный дипломный проект с указанием относящихся к ним разделов проекта

Наименования разделов	Консультанты, И.О.Ф. (уч. степень, звание)	Дата подписания	Подпись
Программное обеспечение	Рамазан А.Б.	31.05.2021	
Нормоконтролер	Марғұлан Қ.	06.06.2021г	

Научный руководитель _____ А.Д. Куникеев

Задание принял к исполнению обучающийся _____ Еленов А.М.

Дата «6» июня 2021 г.

АННОТАЦИЯ

Диабет является проблемой для людей по всему миру, а также в Республике Казахстан. На его лечение и своевременную диагностику уходит большое количество ресурсов. Ежегодно большое количество людей сталкиваются с этой проблемой. В данном проекте я протестировал методы диагностики сахарного диабета используя различные методы машинного обучения, сравнил различные классические методы машинного обучения, а также метод на базе глубокого обучения. Результаты сравнения показали, что нейронная сеть справляется с задачей лучше всего, в то время как остальные два метода имеют одинаковую точность, но отличаются количеством ложноположительных и ложноотрицательных предсказаний, второе может крайне негативно сказаться так как такая ошибка несет потенциальный вред и может привести к трагическим последствиям. В целом, используя метод, основанный на нейронных сетях, мы добиваемся наивысших результатов. Данную работу предполагается рассматривать как основание для будущих работ преследующих цель проверки данных методов на базе локальных данных РК с дальнейшей интеграцией.

АНДАТПА

Қант диабеті бүкіл әлемдегі, сондай-ақ Қазақстан Республикасындағы адамдар үшін проблема болып табылады. Оны емдеу мен уақтылы диагностикалауға ресурстардың көп мөлшері жұмсалады. Жыл сайын мұндай проблемаға көптеген адамдар тап болады. Бұл жобада мен қант диабетін диагностикалау әдістерін әртүрлі машиналық оқыту әдістерін қолдана отырып тексердім, әртүрлі классикалық машиналық оқыту әдістерін, сонымен қатар терең білім алуға негізделген әдісті салыстырдым. Салыстыру нәтижелері көрсеткендей, нейрон желісі тапсырманы ең жақсы шешеді, ал қалған екі әдіс бірдей дәлдікке ие, бірақ жалған оң және жалған теріс болжамдардың санымен ерекшеленеді, екіншісі өте теріс әсер етуі мүмкін, өйткені мұндай қателік ықтимал зиян келтіреді және қайғылы салдарға әкелуі мүмкін ... Жалпы, нейрон желілеріне негізделген әдісті қолданып, біз ең жақсы нәтижеге қол жеткіземіз. Бұл жұмыс әрі қарайғы интеграциямен ҚР жергілікті деректері негізінде осы әдістерді тексеруге бағытталған болашақ жұмыс үшін негіз ретінде қарастырылуы керек.

ANNOTATION

Diabetes is a problem for people around the world, as well as in the Republic of Kazakhstan. A large amount of resources is spent on its treatment and timely diagnosis. A large number of people face this problem every year. In this project, I tested methods for diagnosing diabetes mellitus using various machine learning methods, compared various classical machine learning methods, as well as a method based on deep learning. The comparison results showed that the neural network copes with the task best, while the other two methods have the same accuracy, but differ in the number of false-positive and false-negative predictions, the second can have an extremely negative effect since such an error carries potential harm and can lead to tragic consequences. In general, using the method based on neural networks, we achieve the best results. This work is supposed to be considered as the basis for future work aimed at verifying these methods based on local data of the RK with further integration.

СОДЕРЖАНИЕ

ВВЕДЕНИЕ

Сахарный диабет – заболеваний хронического типа, возникающее при недостаточной выработке инсулина поджелудочный железой или неэффективном использований его организмом.

Согласно ВОЗ, в 1980 году 108 миллионов людей страдало сахарным диабетом. К 2014 году оказалось 422 миллиона диабетиков. Сахарный диабет распространяется быстрее в странах с низким и средним доходом. Диабет является основной причиной таких болезней, как инсульт, слепота, а также почечной недостаточности и сердечных приступов и часто приводит к ампутации нижних конечностей. За 16 лет с 2000 по 2016 годы, на 5% выросла смертность от сахарного диабета. Только в 2019 году умерло 1.5 миллиона человек из-за сахарного диабета [5].

1 Исследовательский раздел

1.1 Цель разработки

Целью данной дипломной работы является выявление сахарного диабета у людей используя различные методы глубокого обучения на медицинских данных и сравнение классических методов машинного обучения и методов глубокого обучения.

1.2 Термины и сокращения

Термины и аббревиации, использованные в данной дипломной работе описаны в таблице 1 ниже.

Таблица 1 - Термины, сокращения, и их определения

Сокращение или термин	Определение
ВОЗ	(сокр. от Всемирная организация здравоохранения) – международная организация, отвечающая за медицину и здравоохранение
СД	(сокр. от сахарный диабет)
KNN	(сокр. от англ. k-nearest neighbors) Метод k-ближайших соседей
LightGBM	(сокр. от англ. Light Gradient Boosting Machine) метод градиентного бустинга
AI	(сокр. от англ. Artificial Intelligence) – искусственный интеллект
GOSS	(сокр. от англ. Gradient-based One-Side Sampling) – метод LightGBM
EFB	(сокр. от англ. Exclusive Feature Bundling) - метод LightGBM

Продолжение таблицы 1

Сокращение или термин	Определение
ML	(сокр. от англ. Machine Learning) - машинное обучение
TP	(сокр. от англ. True Positive) - истинно положительные
FP	(сокр. от англ. False Positive) - ложноположительные
FN	(сокр. от англ. False Negative) - ложноотрицательные
ReLU	(сокр. от англ. Rectified Linear Unit) – вид функции активации используемая в машинном обучении

1.3 Предметная область

Предметной областью данного дипломного проекта являются машинное обучение и медицина, в частности рассматривающая сахарный диабет.

1.4 Актуальность проблемы диагностики сахарного диабета

По определению Всемирной организации здравоохранения, диабет – это хроническое метаболическое заболевание, характеризующееся повышенным уровнем глюкозы (или сахара в крови), что со временем приводит к серьезным повреждениям сердца, кровеносных сосудов, глаз, почек и нервов. Наиболее распространенным является диабет 2 типа, обычно у взрослых, который возникает, когда организм становится устойчивым к инсулину или не вырабатывает достаточно инсулина. За последние три десятилетия распространенность диабета 2 типа резко возросла в странах с любым уровнем дохода [15]. Диабет 1 типа, когда-то известный как ювенильный диабет или инсулинозависимый диабет, представляет собой хроническое заболевание, при котором поджелудочная железа сама вырабатывает мало инсулина или не вырабатывает его совсем. Для людей, живущих с диабетом, доступ к доступному по цене лечению, включая инсулин, имеет решающее значение для их выживания. Существует глобально согласованная цель - остановить рост диабета и ожирения к 2025 году. В «Глобальном отчете о диабете» ВОЗ содержится обзор бремени диабета [1], доступных вмешательств для профилактики и лечения

диабета, а также рекомендации для правительств, отдельных лиц, гражданского общества и частного сектора.

«Глобальная стратегия ВОЗ по питанию, физической активности и здоровью» дополняет работу ВОЗ по борьбе с диабетом, сосредоточивая внимание на подходах для всего населения к пропаганде здорового питания и регулярной физической активности, тем самым уменьшая растущую глобальную проблему избыточного веса и ожирения [2]. В наше время более 400 миллионов человек живут с диабетом, как показано рисунке ниже.

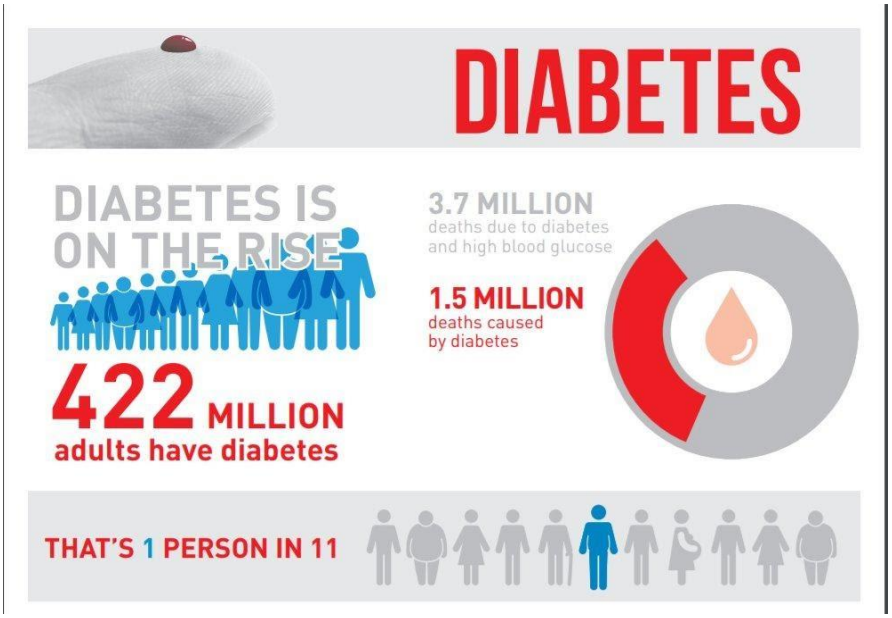


Рисунок 1.4.1. Глобальный отчет ВОЗ

В Казахстане вопрос борьбы стоит особенно остро, т. к. диабет является четвертой болезнью в стране по распространенности на текущий год. В рамках борьбы с данным заболеванием, был принят ряд законов и мер по профилактике, а также лечению диабета. Как и при любом заболевании, существуют поддающиеся изменению и не модифицируемые факторы риска, вызывающие заболевание. Генетика, пол, этническая принадлежность и возраст не поддаются контролю. Люди с диабетом 1 типа, которые имеют генетическую предрасположенность к усиленному и деструктивному аутоиммунному ответу, не могут его контролировать. Люди старше 65 лет афроамериканского происхождения подвергаются большему риску развития диабета 2 типа, но не могут контролировать эти факторы. После краткого обсуждения генетики мы хотим сосредоточиться на контролируемых факторах риска, которые мы можем изменить, чтобы предотвратить диабет [3].

Было сказано, что 90% всех хронических заболеваний можно спровоцировать или предотвратить с помощью образа жизни, особенно диеты и физических упражнений [6]. Геном человека является сильным фактором, определяющим вероятность развития диабета 2 типа. Например, если у дизиготного (разнояйцевого) близнеца разовьется диабет 2 типа, вероятность того, что у другого близнеца также разовьется болезнь, составляет около 25% [16]. Риск заболевания удваивается, если однояйцевые близнецы являются монозиготными (идентичными): если у однояйцевых близнецов развивается диабет 2 типа, вероятность того, что у другого близнеца также разовьется болезнь, составляет около 50% [7]. Однако у людей с такой генетической предрасположенностью не всегда развивается клинический диабет [4].

Определенные проблемы со здоровьем тесно связаны с развитием диабета 2 типа. Эти проблемы со здоровьем не являются ни абсолютными, ни независимыми причинами заболевания; то есть не у всех людей с этими проблемами развивается диабет 2 типа. Тем не менее, они являются основными факторами риска, поскольку помогают вызвать или усугубить диабет 2 типа у людей с предрасположенностью к нему [8].

Основные факторы риска диабета 2 типа включают ожирение, отсутствие физической активности, нездоровое питание, гипергликемию, стресс и хроническое воспаление.

Отсутствие физической активности - еще один важный фактор риска развития диабета 2 типа. Отчасти это связано с тенденцией людей, ведущих малоподвижный образ жизни, накапливать триглицериды в мышечных клетках и набирать вес [9].

Физические упражнения являются мощным противодействием инсулинорезистентности. Регулярные упражнения улучшают гликемический контроль и снижают риск развития сердечно-сосудистых осложнений у людей с диабетом 2 типа.

Стресс активирует симпатическую вегетативную нервную систему в реакции «бей или беги». Кортизол, известный как гормон стресса, вырабатываемый надпочечниками, усиливается и действует как

контррегулирующий гормон по отношению к инсулину. Кортизол повышает уровень глюкозы в крови, пытаясь доставить глюкозу к мышечным клеткам, чтобы бороться со стрессором. Хронический стресс приводит к хронической гипергликемии, которая, в свою очередь, увеличивает инсулинорезистентность и вызывает диабет 2 типа у предрасположенных людей [25].

Контролируемый фактор риска сахарного диабета – это прогрессирующие гипергликемические состояния, которые могут быть вызваны частым потреблением углеводов. Скелетные мышцы и жировая ткань, которые становятся перегруженными глюкозой, менее способны поглощать больше глюкозы, поэтому гипергликемия способствует инсулинорезистентности, преддиабету и, в итоге, диабету [10].

Постепенно развиваются длительные осложнения диабета. Чем дольше вы страдаете диабетом и чем меньше контролируете уровень сахара в крови, тем выше риск осложнений. В конце концов, осложнения диабета могут привести к потере трудоспособности или даже к угрозе для жизни [11]. Возможные осложнения включают:

Сердечно-сосудистые заболевания. Диабет резко увеличивает риск возникновения различных сердечно-сосудистых заболеваний, в том числе ишемической болезни сердца с болью в груди (стенокардия), сердечного приступа, инсульта и сужения артерий (атеросклероз). Если у вас диабет, у вас больше шансов заболеть сердечным заболеванием или инсультом [12].

Поражение нервов (невропатия). Избыток сахара может повредить стенки крошечных кровеносных сосудов (капилляров), питающих нервы, особенно в ногах. Это может вызвать покалывание, онемение, жжение или боль, которые обычно начинаются на кончиках пальцев ног или пальцев и постепенно распространяются вверх. Если не лечить, вы можете потерять чувство чувствительности в пораженных конечностях. Повреждение нервов, связанное с пищеварением, может вызвать тошноту, рвоту, диарею или запор. У мужчин это может привести к эректильной дисфункции [13].

Поражение почек (нефропатия). Почки содержат миллионы крошечных скоплений кровеносных сосудов (клубочков), которые фильтруют отходы из крови. Диабет может повредить эту хрупкую систему фильтрации. Серьезное повреждение может привести к почечной недостаточности или необратимой терминальной стадии заболевания почек, что может потребовать диализа или трансплантации почки.

Поражение глаз (ретинопатия). Диабет может повредить кровеносные сосуды сетчатки (диабетическая ретинопатия), что может привести к слепоте. Диабет также увеличивает риск других серьезных проблем со зрением, таких как катаракта и глаукома.

Повреждение стопы. Повреждение нервов стоп или плохой приток крови к стопам увеличивает риск различных осложнений со стопами. При отсутствии лечения порезы и волдыри могут привести к серьезным инфекциям, которые часто плохо заживают. Эти инфекции могут в итоге потребовать ампутации пальца, стопы или ноги.

Состояние кожи. Диабет может сделать вас более восприимчивым к кожным проблемам, включая бактериальные и грибковые инфекции [14].

Нарушение слуха. Проблемы со слухом чаще встречаются у людей с диабетом.

Болезнь Альцгеймера. Диабет 2 типа может увеличить риск деменции, такой как болезнь Альцгеймера. Чем хуже уровень сахара в крови, тем выше риск. Хотя существуют теории о том, как эти расстройства могут быть связаны, ни одна из них еще не доказана [17].

2 Технологический раздел

2.1 Python

Python – язык программирования, который включает в себя высокоуровневые структуры данных, динамическую типизацию, динамическое связывание и другие функции, которые позволяют использовать его для разработки сложных приложений, как и для написания сценариев или «связующего кода», соединяющего компоненты вместе Благодаря своей универсальности, Python часто используют для машинного обучения.

2.2 Jupiter

Jupyter – интерактивный веб-инструмент с открытым исходным кодом, выступающий в роли вычислительной записной книжки, который исследователи могут использовать для объединения программного кода, результатов вычислений, пояснительного текста и мультимедийных ресурсов в одном документе. Вычислительные ноутбуки существуют уже несколько десятилетий, но популярность Jupyter особенно возросла за последние пару лет. Этому быстрому внедрению способствовало сообщество энтузиастов-разработчиков и переработанная архитектура, которая позволяет ноутбуку говорить на десятках языков программирования.

2.3 PyTorch

PyTorch – это библиотека для языка Python, которая упрощает создание моделей глубокого обучения. PyTorch подчеркивает гибкость и позволяет выражать модели глубокого обучения на идиоматическом языке Python.
Проще говоря, это как Numpy, но с ускорением графического процессора. Более того, PyTorch поддерживает графы динамических вычислений, которые позволяют изменять поведение нейронной сети на лету, в отличие от статических графов.

2.4 Серверное оборудование

Сервер с двумя видеоускорителями NVIDIA RTX 2080ti, двумя центральными процессорами Intel xeon на 64 ядра и 128гб ОЗУ.

3 Проектная часть

3.1 Нейронные сети

Сегодня нейронные сети успешно решают огромное количество различных задач, но сегодня мы рассмотрим простую задачу классификации. Для некоторых объектов нам необходимо предсказать, к какому классу он принадлежит, например, в соответствии с характеристиками из описания пациента, чтобы предсказать, болен он или здоров, или предсказать, какая фигура изображена на картинке с фигурой. Фактически, нейронная сеть – это некоторая функция, алгоритм, который имеет что-то на входе и что-то на выходе, а на входе и выходе должны быть какие-то числовые данные. Рассмотрим простейший пример: пусть нам даны три числовые характеристики некоторого объекта, то есть всего три числа, и нам нужно сделать простое двоичное предсказание для них «да» или «нет». Например, это может быть давление, температура, возраст конкретного пациента, и нам нужно предсказать, болен он или нет. Когда нам нужно принять итоговое решение по нескольким факторам, мы обычно взвешиваем эти факторы. Давайте воспользуемся этим принципом для построения нашего предсказателя. Мы суммируем входные значения, с некоторыми весами. Таким образом, взвешенная сумма входов будет равна

Далее, чтобы определиться с полученным числом, вы можете сравнить его с каким-либо другим числом или использовать эквивалентную формулу. Результирующая взвешенная сумма плюс некоторое значение смещения сравнивается с нулем: больше или меньше. Чтобы проверить это условие, вы можете использовать следующую пороговую функцию:

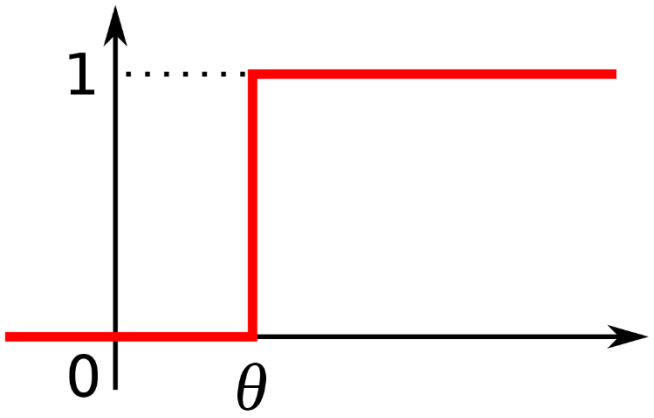


Рисунок 3.1.1 - Пороговая функция.

Все, что больше нуля, следует интерпретировать как единицу, все, что меньше нуля, следует интерпретировать как ноль. Его вывод можно напрямую интерпретировать как «да» или «нет». Часто бывает полезно смягчить эту особенность. Таким образом, на выходе мы получаем не фиксированные

значения 0 и 1, а некоторое непрерывное значение от нуля до единицы, как вероятность того, что наш пациент заболел.

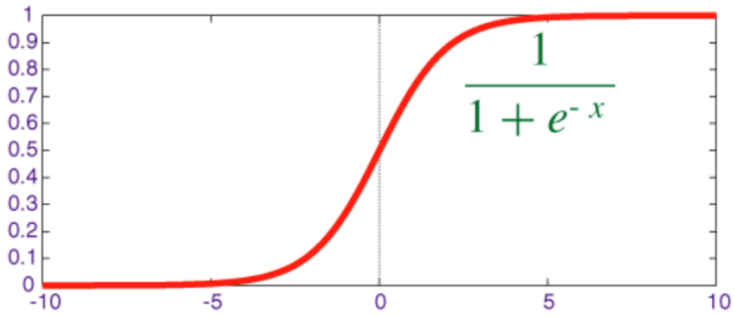


Рисунок 3.1.2 - сигмовидная функция.

Для этого, например, подойдет сигмовидная функция. При нуле у нас вероятность 0,5, т. е. мы точно не знаем, и чем дальше от нуля до положительных или отрицательных значений, соответственно, значение вероятности ближе к единице или единице. Этот простой пример соответствует тому, что происходит в одном нейроне некоторой нейронной сети, или, другими словами, его можно назвать «однослойной нейронной сетью». Этот инструмент уже можно использовать для решения различных задач классификации, но не очень сложных.

Такая модель не очень выразительна, поскольку описывает только линейную закономерность и применима только в тех случаях, когда данные линейно разделимы. Чтобы искать более сложные шаблоны, архитектура должна быть сложной. Сложные задачи имеет смысл решать по принципу «разделяй и властвуй», т. е. разбивать ее на более простые подзадачи. Таким образом, пусть один нейрон решает одну небольшую простую задачу, а сеть нейронов может решать более сложную задачу.

Итак, пусть теперь у нас есть несколько независимых нейронов, которые, согласно входным данным, решают несколько простых задач и получают несколько ответов. Каждый нейрон имеет свои собственные веса, и затем мы используем эти промежуточные данные в качестве входных значений для некоторого другого предиктора, в нашем случае, некоторого другого нейрона. Итак, мы получили простую двухслойную нейронную сеть, второй уровень также решает небольшую простую задачу, но работает с ответами первого уровня.

Давайте немного улучшим нашу нейронную сеть, если мы хотим делать классификацию не на два класса, а сразу на n классов. В таких случаях обычно строят несколько бинарных классификаторов по принципу «один против всех». В случае нейронных сетей это эквивалентно помещению ровно n нейронов в последний слой. Теперь на выходе из нейросети у нас уже есть n чисел, и эти n чисел очень удобно интерпретировать как вероятности принадлежности к тому или иному классу. Для этого достаточно в конце выполнить какое-то

нормализующее преобразование, чтобы все числа были от нуля до единицы, а сумма всех вероятностей также была равна единице.

Это, например, можно сделать с помощью функции SoftMax. Однако, конечно, это можно усложнить, добавив больше слоев, что эквивалентно увеличению глубины нейронной сети. Вы также можете добавлять нейроны к каждому отдельному слою, это иногда называют увеличением ширины нейронной сети, хотя уже двухслойная нейронная сеть является универсальным аппроксиматором, имеет смысл делать более глубокие, чтобы избежать переобучения и искать больше сложные узоры. В нейронных сетях эти промежуточные значения не обязательно должны быть значимыми или интерпретируемыми. Во время обучения нейронная сеть сама определит, что удобно поставить посередине, нас беспокоит только ввод и конечный вывод, то, что мы в итоге получили, называется полносвязной нейронной сетью или многослойным персептроном.

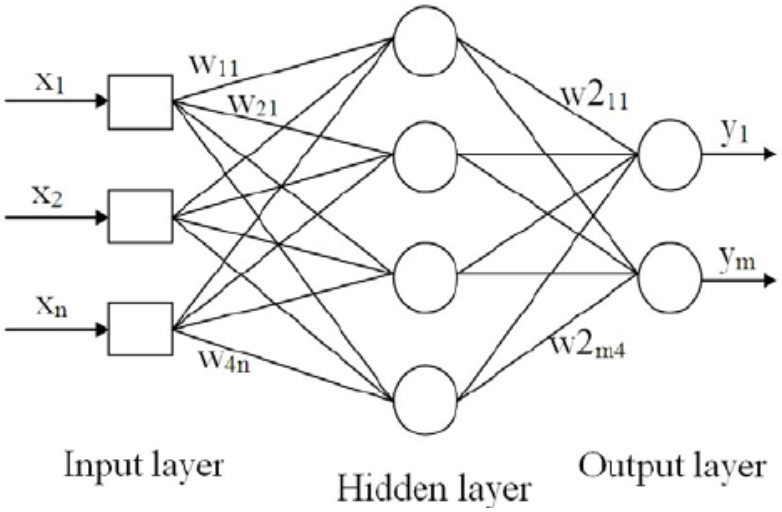


Рисунок 3.1.3 - Уровни нейронной сети

Самая распространенная и популярная функция - ReLu:

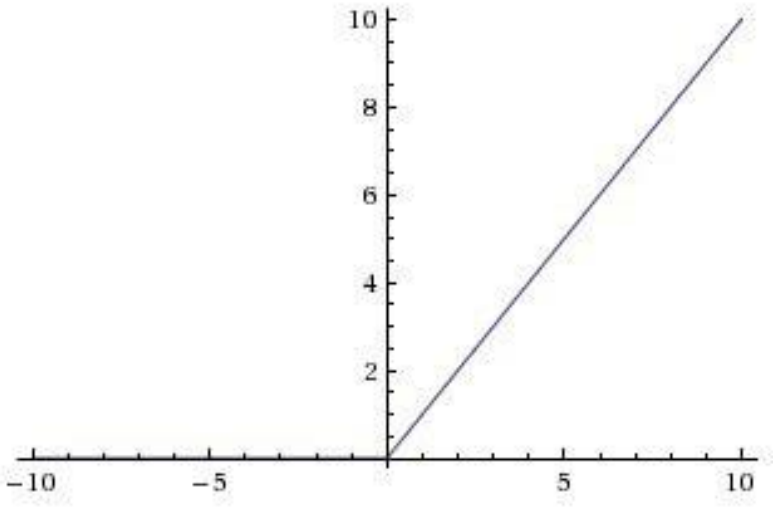


Рисунок 3.1.4 – ReLu

она оставляет все, что больше нуля, как есть, а все, что меньше нуля, становится равным нулю.

Кроме того, помимо простоты, он обладает и другими полезными свойствами, которые помогают в обучении нейронной сети. Теперь давайте более подробно рассмотрим, как работает один слой нейронной сети: взвешенная сумма в одном нейроне может быть представлена как скалярное произведение актора x, состоящего из входных значений, и вектора w, состоящего из соответствующих весовых коэффициентов, но поскольку мы есть несколько нейронов в одном слое и на выходе у нас есть целый вектор выходных значений h, получается, что каждый вход связан с каким-то выходом, и каждое такое соединение соответствует некоторому уникальному весу. Итак, у нас есть весовая матрица. Например, если было три входа и четыре выхода, матрица будет иметь размер 4 на 3. Вместе они дают нам вектор смещения или смещение. В результате вычисление одного слоя нейронной сети эквивалентно умножению входного вектора x на весовую матрицу w и добавлению вектора смещения и взятию функции активации в каждом элементе вектора. Получили вектор промежуточных скрытых значений. Затем мы подаем его на второй слой, где снова умножаем на новую матрицу весов второго слоя, добавляем вектор смещения и так далее. В последнем слое перед активацией softmax обычно не используется.

3.2 Глубокое обучение

Глубокое обучение можно рассматривать как разновидность машинного обучения. Это область, основанная на самостоятельном обучении и улучшении путем изучения компьютерных алгоритмов. В то время как машинное обучение использует более простые концепции, глубокое обучение работает с искусственными нейронными сетями, которые созданы для имитации того, как люди думают и учатся. До недавнего времени нейронные сети были ограничены вычислительной мощностью и, следовательно, были ограничены по сложности. Однако достижения в области аналитики больших данных позволили создать более крупные и сложные нейронные сети, позволяющие компьютерам наблюдать, учиться и реагировать на сложные ситуации быстрее, чем люди. Глубокое обучение помогло классифицировать изображения, языковой перевод, распознавание речи. Его можно использовать для решения любой проблемы распознавания образов и без вмешательства человека.

Искусственные нейронные сети, состоящие из многих слоев, способствуют глубокому обучению. Глубокие нейронные сети (DNN) – это такие типы сетей, в которых каждый уровень может выполнять сложные операции, такие как представление и абстракцию, которые имеют смысл изображений, звука и текста. Глубокое обучение, считающееся самой быстрорастущей областью машинного обучения, представляет собой поистине революционную цифровую

технологию, и все больше компаний используют его для создания новых бизнес-моделей.

3.3 Ансамбль

Обычный рабочий процесс машинного обучения с учителем, который частично обсуждается в недавно опубликованной статье в DSDE, существует уже более 4 десятилетий. В основном это включает в себя создание единственного экземпляра определенного метода машинного обучения. Рабочий процесс машинного обучения с учителем принимает обучающий набор данных, строит модель, оптимизирует параметры модели и использует обученную модель для прогнозирования целевой переменной из нового набора входных данных.

Метод ансамбля создает несколько экземпляров традиционных методов машинного обучения и объединяет их для выработки единого оптимального решения проблемы. Этот подход позволяет создавать более качественные модели прогнозирования по сравнению с традиционным подходом. Основные причины для использования ансамблевого метода включают ситуации, когда есть неопределенности в представлении данных, целях решения, методах моделирования или наличии случайных начальных начальных значений в модели. Экземпляры или методы-кандидаты называются базовыми учащимися. Каждый базовый ученик работает независимо, как традиционный метод машинного обучения, и конечные результаты объединяются для получения единого надежного результата. Комбинирование может быть выполнено с использованием любого из методов усреднения (простого или взвешенного) и голосования (мажоритарного или взвешенного) для методов регрессии и классификации соответственно.

Методы ансамбля также известны как «комитет машин» или «комитет экспертов», причем последний исходит из предположения, что каждый базовый обучающийся является «экспертом», а его результат - «экспертным мнением».

Ансамбль представляет собой имитацию человеческого поведения при социальном обучении, заключающегося в поиске различных мнений перед принятием решения. В психологии человека считается, что мнение комитета выше и более надежно, чем мнение отдельных лиц. Основным мотивом для применения ансамблевого метода является статистически обоснованный аргумент, что он является частью человеческих стратегий принятия решений.

Наличие разнообразия является основным требованием для реализации метода ансамбля. Модель ансамбля должна быть построена таким образом, чтобы результаты базовых учащихся были разнообразными. Это имитирует различные мнения членов комитета в человеческом случае. Проблема должна быть сложной и запутанной, чтобы комитет помогал принять решение. Если все члены комитета всегда соглашаются по всем вопросам, тогда вообще нет

необходимости в комитете. Мнение отдельного человека было бы так же хорошо. То же относится и к ансамблевому методу. Если базовые учащиеся слишком однородны, так что все они дают одинаковый результат, желаемая цель не будет достигнута.

3.4 Логистическая регрессия

Логистическая регрессия – это подходящий регрессионный анализ, который следует проводить, когда зависимая переменная является бинарной. Как и все регрессионные анализы, логистическая регрессия – это прогнозный анализ. Логистическая регрессия используется для описания данных и объяснения взаимосвязи между одной зависимой двоичной переменной и одной или несколькими номинальными, порядковыми, интервальными или пропорциональными независимыми переменными.

В логистической регрессии мы не подгоняем прямую линию к нашим данным напрямую, как в линейной регрессии. Вместо этого мы подгоняем к нашим наблюдениям кривую, называемую сигмоидой.

Прежде всего, как мы сказали ранее, модели логистической регрессии – это модели классификации; в частности, модели бинарной классификации.

3.5 Random forest

Random forest, как следует из его названия, состоит из большого количества отдельных деревьев решений, действующих как ансамбль. Каждое отдельное дерево в случайном лесу дает предсказание класса, и класс, набравший наибольшее количество голосов, становится предсказанием нашей модели. Большое количество относительно некоррелированных моделей (деревьев), действующих как комитет, превзойдут любую из отдельных составляющих моделей. Ключевым моментом является низкая корреляция между моделями. Некоррелированные модели могут давать ансамблевые прогнозы, которые более точны, чем любые индивидуальные прогнозы. Причина этого замечательного эффекта в том, что деревья защищают друг друга от своих индивидуальных ошибок (до тех пор, пока все они не ошибаются постоянно в одном и том же направлении). В то время как некоторые деревья могут быть неправильными, многие другие деревья будут правильными, поэтому деревья могут двигаться в правильном направлении как группа.

Для того чтобы понять, как устроен случайный лес, нужно понимать, как работает одно древо решений.

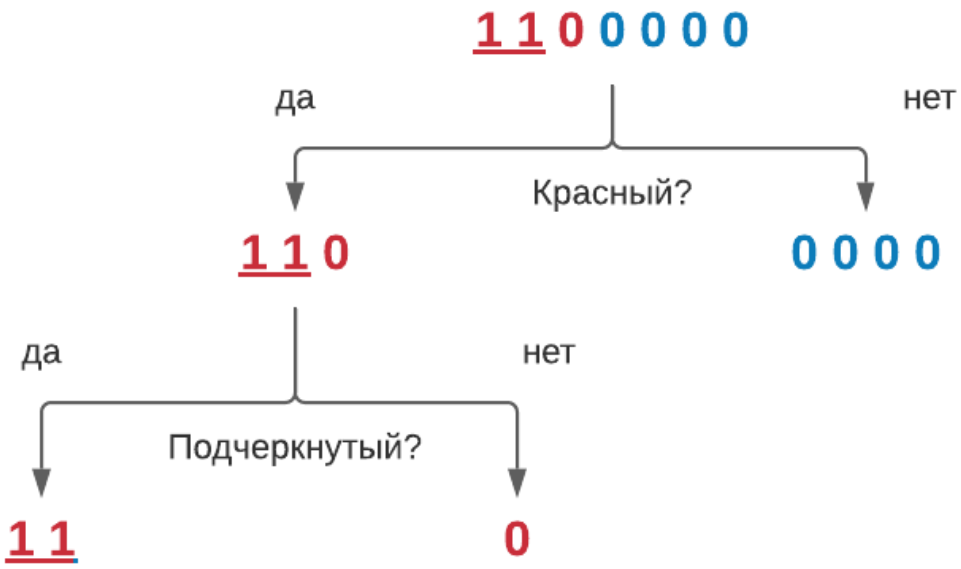


Рисунок 3.5.1 – Пример дерева решений

Представьте, что наш набор данных состоит из чисел в верхней части рисунка. У нас есть две единицы и пять нулей (1 и 0 - наши классы), и мы хотим разделить классы, используя их особенности. Характеристики окрашены в цвет (красный или синий), независимо от того, подчеркнуто наблюдение или нет. Итак, как мы можем это сделать?

Цвет кажется довольно очевидной особенностью для разделения, поскольку все нули, кроме одного, синие. Таким образом, мы можем использовать вопрос: «Это красный?» чтобы разделить наш первый узел. Вы можете представить себе узел в дереве как точку, где путь разделяется на две части: наблюдения, соответствующие критериям, идут вниз по ветви "Да", а наблюдения, которые не идут вниз, по ветви "Нет".

Ветвь «Нет» (синяя) теперь имеет нулевые значения, так что мы на этом закончили, но нашу ветвь «Да» все еще можно разделить. Теперь мы можем использовать вторую функцию и спросить: «Она подчеркнута?» чтобы сделать второй раскол. Две подчеркнутые единицы идут вниз по ответвлению "Да", а ноль, который не подчеркнут, идут по правой ветви, и все готово. Наше дерево решений могло использовать эти две функции для идеального разделения данных.

3.6 Наивный байесовский классификатор

Теорема Байеса говорит нам, как мы можем вычислить эту условную вероятность. Данная теорема представлена ниже как:

$$P(A|B) = (P(B|A)*P(A)) / P(B)$$

где A – событие 1, B – событие 2, P (A) вероятность события 1, P(B) вероятность события 2, P(B|A) вероятность события 1 при событии 2.

Что мы знаем из теории вероятностей, так это то, что если X1 и X2 являются независимыми значениями (это означает, что, например, тот факт, что погода дождливая и что сегодня выходной день, полностью независимы, между ними нет никакой условной связи), то мы можем использовать это уравнение.

$$P(X1, X2|Y) = P (X1|Y) * P(X2|Y)$$

В нашем примере это предположение верно. То, что сегодня дождливый день, никак не может быть связано с тем, что сегодня суббота. Но, вообще говоря, это предположение в большинстве случаев неверно. Если мы наблюдаем большое количество переменных для задач классификации, есть вероятность, что по крайней мере некоторые из этих переменных являются зависимыми (например, уровень образования и ежемесячный доход).

Но наивный байесовский классификатор называют наивным только потому, что он работает на основе этого предположения. Мы считаем все наблюдаемые переменные независимыми, потому что использование приведенного выше уравнения помогает нам упростить следующие шаги.

3.7 Дерево K-соседей и дерево решений

Алгоритм k-ближайших соседей – это простой, легко реализуемый алгоритм контролируемого машинного обучения, который можно использовать для решения задач классификации и регрессии.

Алгоритм KNN предполагает, что похожие вещи существуют в непосредственной близости. Другими словами, похожие вещи находятся рядом друг с другом. Алгоритм предполагает сходство между новым случаем / данными и доступными случаями и помещает новый случай в категорию, которая наиболее похожа на доступные категории.

Для реализации любого алгоритма нам понадобится набор данных. Итак, на первом этапе KNN мы должны загрузить обучающие, а также тестовые данные. Затем нам нужно выбрать значение K, то есть ближайшие точки данных. K может быть любым целым числом. Для каждой точки в тестовых данных выполните следующие действия:

Рассчитайте расстояние между тестовыми данными и каждой строкой обучающих данных с помощью любого метода, а именно: Евклидова, Манхэттенского или Хэммингового расстояния. Наиболее часто используемый метод расчета расстояния - евклидов. Теперь, исходя из значения расстояния, отсортируйте их в порядке возрастания. Затем он выберет верхние K строк из

отсортированного массива. Теперь он назначит класс контрольной точке на основе наиболее частого класса этих строк.

3.8 Метод опорных векторов

SVM ищет границу наилучшего решения, которая разделяет два класса с наивысшей способностью к обобщению. Первый вопрос, который задают, - как SVM определяет оптимальность. В отличие от логистической регрессии, которая определяет оптимальность по общей вероятности, SVM хочет, чтобы наименьшее расстояние между точками данных и границей решения было как можно большим. Другими словами, если вы представите границу принятия решения как центральную линию улицы, SVM предпочтет 8-полосное шоссе, а не проселочную дорогу. Ширина улицы называется полем.

3.9 Бустинг на основе алгоритма AdaBoost и градиентный бустинг

Алгоритм AdaBoost, сокращенно от Adaptive Boosting, – это метод повышения, который используется в качестве метода ансамбля в машинном обучении. Это называется адаптивным усилением, поскольку веса повторно назначаются каждому экземпляру, а более высокие веса - неправильно классифицированным экземплярам. Повышение используется для уменьшения систематической ошибки, а также дисперсии при обучении с учителем. Он работает по принципу, когда ученики растут последовательно. За исключением первого каждый последующий ученик вырос из уже выросших учеников. Проще говоря, слабые ученики превращаются в сильных. Алгоритм Adaboost также работает по тому же принципу, что и бустинг, но есть небольшая разница в работе.

Общая идея методов повышения заключается в последовательном обучении предикторов, каждый из которых пытается исправить своего предшественника. Два наиболее часто используемых алгоритма повышения – это AdaBoost и Gradient Boosting. На высоком уровне AdaBoost похож на Random Forest в том, что они оба подсчитывают прогнозы, сделанные каждым деревом решений в лесу, чтобы принять решение об окончательной классификации. Однако есть некоторые тонкие различия. Например, в AdaBoost деревья решений имеют глубину 1 (т. е. 2 листа). Кроме того, прогнозы, сделанные каждым деревом решений, по-разному влияют на окончательный прогноз, сделанный моделью.

Во-первых, давайте обсудим работу бустинга. Он составляет n деревьев решений в течение периода обучения данных. Когда создается первое дерево решений / модель, запись, которая неправильно классифицируется во время

первой модели, получает больший приоритет. Только эти записи отправляются в качестве входных данных для второй модели. Процесс будет продолжаться до тех пор, пока мы не укажем количество базовых учащихся, которых хотим создать.

Когда создается первая модель и ошибки из первой модели фиксируются алгоритмом, запись, которая неправильно классифицируется, предоставляется в качестве входных данных для следующей модели. Этот процесс повторяется до тех пор, пока не будет выполнено указанное условие. Существует n моделей, созданных с использованием ошибок предыдущей модели. Поскольку мы знаем принцип повышения, понять алгоритм AdaBoost будет несложно. Давайте углубимся в работу Adaboost. При использовании случайного леса алгоритм создает n деревьев. Он создает правильные деревья, состоящие из начального узла с несколькими узлами-листьями. Некоторые деревья могут быть больше других, но в случайном лесу нет фиксированной глубины. Но с Adaboost это не так. В AdaBoost алгоритм создает только узел с двумя листьями, известный как «пень» (stump на английском).

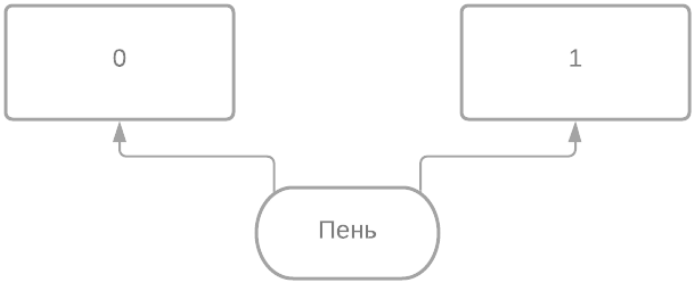


Рисунок 3.9.1 – Графическая репрезентация пня

Основная идея заключается в том, что на каждом этапе мы хотим найти лучший пень, то есть лучшее разделение данных, которое минимизирует общую ошибку. Вы можете рассматривать пень как тест, в котором предполагается, что все, что лежит с одной стороны, относится к классу 1, а все, что лежит с другой стороны, относится к классу 0.

3.10 ExtraTreesClassifier

ExtraTreesClassifier – это метод ансамблевого обучения, в основе которого лежат деревья решений. ExtraTreesClassifier, как и RandomForest, рандомизирует определенные решения и подмножества данных, чтобы минимизировать чрезмерное обучение на основе данных и переобучение.

Метод Extra Trees похож на случайный лес (Random forest) в том смысле, что он строит несколько деревьев и разбивает узлы, используя случайные

подмножества функций, но с двумя ключевыми отличиями: он не загружает наблюдения, и узлы разбиваются на случайные сплиты, а не на лучшие. Итак, вкратце, ExtraTrees:

– по умолчанию строит несколько деревьев с bootstrap = False, что означает выборку без замены

– узлы разбиваются на основе случайного разбиения среди случайного подмножества функций, выбранных в каждом узле

В Extra Trees случайность возникает не из-за начальной загрузки данных, а из-за случайного разделения всех наблюдений.

3.11 LightGBM Classifier

LightGBM – это метод бустинга градиента, в которой используются древовидные алгоритмы обучения, которые считаются очень мощным алгоритмом, когда дело доходит до вычислений. Считается, что это алгоритм быстрой обработки. В то время как деревья других алгоритмов растут по горизонтали, алгоритм LightGBM растет по вертикали, то есть растет по листам, а другие алгоритмы растут по уровням. LightGBM выбирает для роста лист с большими потерями. При выращивании одного и того же листа он может снизить больше потерь, чем алгоритмы по уровням.

В алгоритме используются два ключевых метода: GOSS (Gradient-based One Side Sampling) и EFB (Exclusive Feature Bundling).

GOSS. Различные экземпляры данных играют разную роль в вычислении получения информации. Экземпляры с большими градиентами будут больше способствовать получению информации. GOSS хранит эти экземпляры с большими градиентами (например, превышающими заранее установленный порог или среди верхних процентилей) и только случайным образом отбрасывает эти экземпляры с небольшими градиентами, чтобы сохранить точность оценки получения информации. Эта обработка может привести к более точной оценке усиления, чем равномерно случайная выборка, с той же целевой частотой дискретизации, особенно когда значение прироста информации имеет большой диапазон.

EFB. Данные высокой размерности обычно очень разрежены, что дает нам возможность разработать подход практически без потерь для уменьшения количества функций. В частности, в разреженном пространстве функций многие функции являются взаимоисключающими, т. е. они никогда не принимают ненулевые значения одновременно. Эксклюзивные функции могут быть безопасно объединены в одну функцию (называемую Exclusive Feature Bundle). Следовательно, понижая сложность построения гистограммы.

4 Экспериментальный раздел

4.1 Датасет

В качестве данных для анализа я взял два датасета из открытых источников: Pima Indians Diabetes Database и Frankfurt, Germany Dataset of diabetes.

Поскольку при применении методов глубокого обучения и нейронных сетей для успешного анализа требуется большое количество данных, я объединил эти два датасета. Данный датасет содержит данные двухтысяч пациентов, и содержит такие данные, как: количество беременностей, уровень глюкозы в крови, давление, толщина кожи, уровень инсулина, индекс массы тела, численное представление вероятности передачи диабета по наследству, возраст и имеется ли диабет у данного пациента или нет.

4.2 Метрика

В качестве метрики я использовал метрику F1. Оценка F1 сочетает в себе точность и отзывчивость по отношению к определенному положительному классу - оценку F1 можно интерпретировать как средневзвешенное значение точности и отзыва, где оценка F1 достигает своего лучшего значения при 1 и худшего при 0. Данная метрика высчитывается по формуле:

$$F1 = 2\ (точность*отзыв)/(точность-отзыв)$$

В свою очередь, точность и отзыв вычисляются по формулам:

$$Точность = TP/(TP+FP)$$

$$Отзыв = TP/(TP+FN)$$

где, TP – истинно положительные, FP – ложноположительные, FN – ложноотрицательные

4.3 Предобработка

Для предобработки данных я использовал библиотеку StandardScaler, для того чтобы стандартизировать значения путем удаления среднего и масштабирования до единичной дисперсии. Стандартизация набора данных является общим требованием для многих оценщиков машинного обучения: они

могут вести себя плохо, если отдельные функции не более или менее выглядят как стандартные нормально распределенные данные (например, по Гауссу с нулевым средним и единичной дисперсией). Например, многие элементы, используемые в целевой функции алгоритма обучения (такие как ядро RBF машин опорных векторов или регуляризаторы L1 и L2 линейных моделей), предполагают, что все функции сосредоточены вокруг 0 и имеют дисперсию в том же порядке. Если характеристика имеет дисперсию, которая на порядки больше, чем у других, она может доминировать над целевой функцией и сделать оценщик неспособным правильно учиться на других функциях, как ожидалось.

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
0	6	148	72	35	0	33.6	0.627	50	1
1	1	85	66	29	0	26.6	0.351	31	0
2	8	183	64	0	0	23.3	0.672	32	1
3	1	89	66	23	94	28.1	0.167	21	0
4	0	137	40	35	168	43.1	2.288	33	1
...	...	...	...	...	...	...	...	...	...
1995	2	75	64	24	55	29.7	0.370	33	0
1996	8	179	72	42	130	32.7	0.719	36	1
1997	6	85	78	0	0	31.2	0.382	42	0
1998	0	129	110	46	130	67.1	0.319	26	1
1999	2	81	72	15	76	30.1	0.547	25	0

2768 rows × 9 columns

Рисунок 4.3.1 – данные до стандартизации

index	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
475	-1.125851	0.489532	0.779516	0.374069	-0.719471	-0.617124	-0.737798	2.186560	0
287	-0.825208	-0.071421	0.884351	1.122321	1.260628	1.690795	1.033907	-0.352227	1
402	0.377366	0.458368	0.779516	1.247030	0.072569	0.353967	-0.568918	0.155530	1
620	-0.524564	-0.289570	0.884351	1.309384	0.720601	0.782761	-0.691740	-0.436853	0
253	-1.125851	-1.099836	-0.059165	0.685841	-0.719471	0.454860	-0.716304	-0.690732	0
...	...	...	...	...	...	...	...	...	...
1170	-1.125851	0.396040	-0.583341	-0.062411	1.899660	-0.730629	-0.366262	-1.029237	0
247	-1.125851	1.362126	1.094021	0.748195	5.400836	2.535771	-0.135971	-0.859985	0
912	0.076722	-0.133749	-0.268835	0.374069	0.360583	0.126959	-0.740868	-0.775358	0
1584	-1.125851	-0.164913	-0.268835	1.122321	1.305630	1.009770	0.763699	1.424924	0

Рисунок 4.3.2 – данные после стандартизации

4.4 Результаты обучения

В данной работе я использовал полносвязную нейронную сеть, механизмы которой были описаны мною выше. За количество входных нейронов я взял количество характеристик, равном 8. После чего посылаю данные в полносвязный скрытый слой нейронной сети, состоящий из 100 нейронов. Далее данные попадают в функцию ReLU, которые так же были описаны выше. После чего мы так же имеем еще два полносвязных слоя с 50 и 20 нейронами соответственно, как показано на рисунке. После чего мы обрабатываем данные используя сигмоевидную функцию для получения ответа – 0 или 1, что соответствует отсутствию или присутствию диабета у пациента.

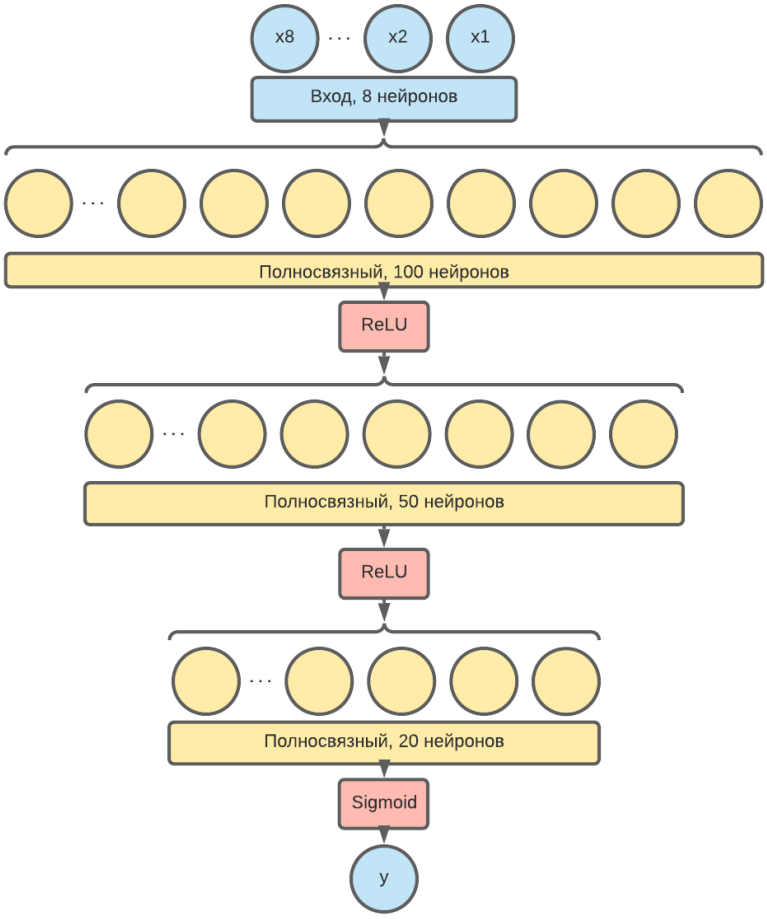


Рисунок 4.4.1 – архитектура нейронной сети

Используя датасет, описанный выше я разбил свой алгоритм на три части:

– LightGBM
– Анасамбль описанных методов
– Полносвязная нейронная сеть

И сравнил полученные результаты и получил, что наибольшей точностью обладает многослойная нейронная сеть – 99%, далее идут LightGBM и ансамбль методов – 96%. На рисунках ниже мы можем видеть матрицы ошибок для каждого из методов.

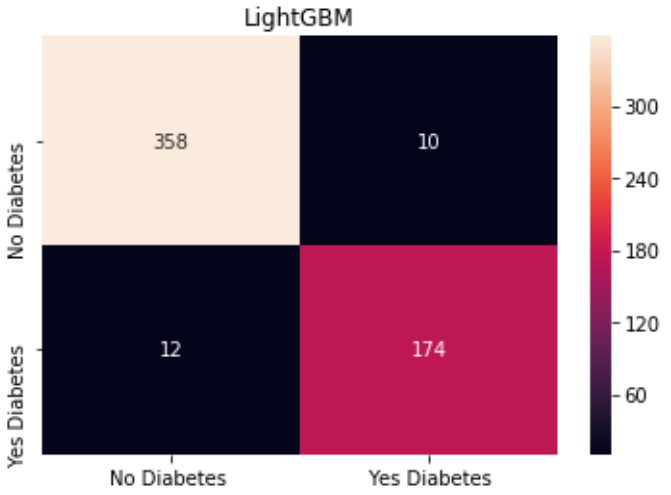


Рисунок 4.4.2 – LightGBM

В данном методе мы можем наблюдать, что количество ложноположительных результатов равно 12, а ложноотрицательных – 10.

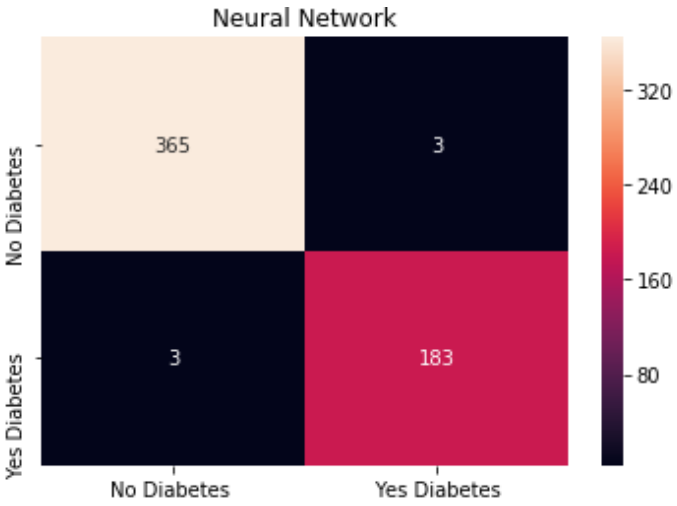


Рисунок 4.4.3 – Нейронная сеть

Здесь же, количество ложноположительных и ложноотрицательных результатов равно 3

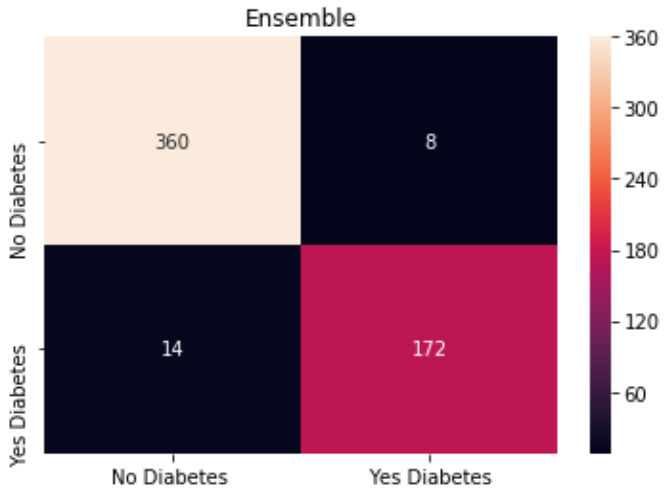


Рисунок 4.4.4 - Ансамбль

В данном же методе количество ложноположительных равно 14, а ложноотрицательных равно 8.

Из этого можно сделать вывод, что нейронные сети опережают классические методы и в данной метрике. Что интересно, так это тот факт, что в методе с использованием LightGBM мы получили больше ложноотрицательных методов, что означает что этот метод чаще говорит, что у человека нет диабета, хотя на самом деле это не так, что может привести к летальным последствиям, нежели в случае ложноположительных, где в худшем случае человек просто лишний раз сходит к доктору. Из этого можно сделать вывод, что в целом на первом месте у нас стоит нейронная сеть, после этого идет ансамбль классических методов, а на последнем месте LightGBM.

ЗАКЛЮЧЕНИЕ

Выявление диабета является очень важным вопросом в текущих реалиях в РК. С развитием машинного обучения и алгоритмов становится возможным их применение на более сложных данных в задачах классификации. На сегодняшний день в период развития доказательной медицины, такие информационные и рекомендательные системы становятся особенно важны.

В своей работе я показал, что для задачи выявления диабета являются достаточными не только глубокие нейронные сети, но и методы на основе бустинга, а также классические методы машинного обучения. Из чего можно сделать вывод, что для внедрения подобных систем в РК не требуется огромные затраты на высокопроизводительные серверные мощности, а также возможность использовать переносные и мобильные устройства.

Метод, основанный на нейронных сетях, показал себя лучшего всего, имея как самый низкий коэффициент ошибки, так и самое низкое количество ложноотрицательных результатов, что критично в случае, если речь идет о здоровье человека.

В будущем планируется провести тестирование данного приложения на данных, собранных в РК и дообучить и подготовить для внедрения и интеграцию с местными системами здравоохранения.

СПИСОК ИСПОЛЬЗОВАННОЙ ЛИТЕРАТУРЫ

1. ВОЗ. Вопросы здравоохранения: диабет. // Электронная версия на сайте https://www.who.int/diabetes/global-report/en/.

2. Диабет во всем мире. // Электронная версия на сайте https://www.diabetesaustralia.com.au/diabetes-globally.

3. Глобальные показатели диабета растут по мере распространения ожирения. // Электронная версия на сайте https://www.nytimes.com/2015/06/08/health/research/global-diabetes-rates-are-rising-as-obesity-spreads.html.

4. WHO. // Электронная версия на сайте https://www.who.int/news-room/fact-sheets/detail/diabetes.

5. ВОЗ. Ключевые сообщения. Global report on diabetes. World Health Organization. ISBN 978 92 4 156525 7 (NLM classification: WK 810).

6. Время остановить распространение диабета во всем мире. // Электронная версия на сайте https://www.raconteur.net/healthcare/time-to-halt-worldwide-spread-of-diabetes.

7. Диабет:мир в опасности. // Электронная версия на сайте https://geographical.co.uk/people/development/item/2235-the-world-at-risk.

8. Глобальная эпидемия диабета, принесенная глобальным развитием. // Электронная версия на сайте https://www.theatlantic.com/health/archive/2012/07/the-global-diabetes-epidemic-brought-to-you-by-global-development/259305/.

9. МДФ. Эпидемиология и исследования. IDF Diabetes Atlas 8th Edition. // Электронная версия на сайте https://www.idf.org/e-library/epidemiology-research/diabetes-atlas.html.

10. МДФ. Факты и цифры о диабете. // Электронная версия на сайте https://www.idf.org/aboutdiabetes/what-is-diabetes/facts-figures.html.

11. Всемирный день борьбы с диабетом. // Электронная версия на сайте https://www.calend.ru/holidays/0/0/100/.

12. Тажиева А.Е. Состояние и совершенствование организации амбулаторно-поликлинической помощи взрослым больным сахарным диабетом 2 типа в г. Алматы: дисс. на соискание степени доктора философии PhD. – Алматы, 2018. – с.13.

13. Распространенность диабета // Электронная версия на сайте http://data.worldbank.org/indicator/SH.STA.DIAB.ZS?end=2017&locations=IM-OE-TR-AU-FR-DE-GB-US-IT-LV&start=2017&view=bar.

14. Lindsay M Jaacks, Stefanie Vandevijvere, An Pan, Craig J McGowan, Chelsea Wallace, Fumiaki Imamura, Dariush Mozaffarian, Boyd Swinburn,Majid Ezzati, The obesity transition: stages of the global epidemic, Lancet Diabetes Endocrinol 2019.

15. Wang Q, Zhang X, Fang L, Guan Q, Guan L, Li Q (2018) Prevalence, awareness, treatment and control of diabetes mellitus among middleaged andelderly

people in a rural Chinese population: A cross-sectional study. PLoS ONE 13 (6): e0198343.

16. Ramachandran A, Mary S, Yamuna A, Murugesan N, Snehalatha C. High prevalence of diabetes and cardiovascular risk factors associated with urbanization in India. Diabetes Care 2008;31:893–898.

17. Raval A, Dhanaraj E, Bhansali A, Grover S, Tiwari P. Prevalence and determinants of depression in type 2 diabetes patients in a tertiary care centre. Indian J Med Res. 2010;132:195–200.

18. Misra A, Pandey RM, Devi JR, Sharma R, Vikram NK, Khanna N. High prevalence of diabetes, obesity and dyslipidaemia in urban slum population in northern India. Int J Obes Relat Metab Disord. 2001;25:1722–9.

19. Ramachandran A, Snehalatha C, Vijay V, King H. Impact of poverty on the prevalence of diabetes and its complications in urban southern India. Diabet Med. 2002;19:130–5.

20. Gutch M, Mohd Razi S, Kumar S, Gupta KK. Diabetes mellitus: Trends in northern India. Indian J Endocr Metab 2014;18:731-4/.

21. Kalra S, Kalra B, Kumar A. Social stigma and discrimination: A care crisis for young women with diabetes in India. Diabetes Voice. 2009;54:37–9.

22. Распространенность диабета 2018. // Электронная версия на сайте https://www.diabetes.org.uk/professionals/position-statements-reports/statistics/diabetes-prevalence-2018.

23. Диабет в Великобритании: факты и статистика. // Электронная версия на сайте https://mrc.ukri.org/documents/pdf/diabetes-uk-facts-and-stats-june-2015/.

24. Диабет в Великобритании. Знать диабет. Бороться с диабетом. // Электронная версия на сайте https://www.diabetes.org.uk/resources-s3/2019-02/1362B_Facts%20and%20stats%20Update%20Jan%202019_LOW%20RES_EXTERNAL.pdf.

25. Диабет: факты и статистика. version 3. revised: march 2014 next review: march 2015. // Электронная версия на сайте https://www.diabetes.org.uk/resources-s3/2017-11/diabetes- key-stats-guidelines-april2014.pdf.

26. Sanjay Basu, David Stuckler, Martin McKee, Gauden Galea. Nutritional.

27. ВОЗ - Информация о странах с диабетом, 2016. // Электронная версия на сайте https: //www.who.int/diabetes/country-profiles/kaz_ru.pdf?ua=1.

28. Туякбаева А.С. Состояние проблемы и пути профилактики сахарного диабета //Центрально-Азиатский журнал по общественному здравоохранению. – Том 11. – №2 – Алматы. – 2012.-С.8.

29. Центрально-Азиатский диабетологический форум 2015 года //Здоровье Казахстана медицинская газета. - №3(34) – Алматы. – 2015. – С.55/.

Приложение А
(обязательное)

Техническое задание

А.1.5 Техническое задание на разработку системы успеваемости студентов

Настоящее техническое задание распространяется на разработку приложения для определения сахарного диабета используя методы машинного обучения.

А.1.5.1 Основание для разработки

Редактор разрабатывается на основании устного распоряжения Заместителя Председателя Правления АО «Институт цифровой техники и технологий» по разработке программных продуктов.

А.1.5.2 Назначение

Разрабатываемое приложения создано для распознания диабета и вероятности его возникновения у пациентов.

А.1.5.3 Требования к функциональным характеристикам

Приложение должно обеспечить возможность выполнения следующих функций:

– определение сахарного диабета у пациентов

– сравнение и анализ выводов различных подходов к определению заболевания

А.1.5.4 Требования к надежности

Обеспечить высокий уровень достоверности и точности вывода

Приложение В
(обязательное)

Текст программы

1. - *Код из файла Diploma_Diabetes.ipynb*

```
import numpy as np
import pandas as pd
import torch
import torch.nn as nn
import torch.nn.functional as F
import torchvision
import torchvision.transforms as transforms
from torch.utils.data import Dataset, DataLoader
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
import lightgbm as lgb
from sklearn.ensemble import VotingClassifier
from sklearn.linear_model import LogisticRegression
from sklearn.naive_bayes import GaussianNB
from sklearn.ensemble import RandomForestClassifier, VotingClassifier
from sklearn.model_selection import StratifiedKFold
from sklearn.model_selection import cross_val_score
from sklearn.neighbors import KNeighborsClassifier
from sklearn.svm import SVC
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import AdaBoostClassifier
from sklearn.ensemble import GradientBoostingClassifier
from sklearn.ensemble import ExtraTreesClassifier
from sklearn.preprocessing import OneHotEncoder
from lightgbm import LGBMClassifier
from sklearn.ensemble import StackingClassifier
from sklearn.metrics import classification_report, confusion_matrix
import seaborn as sns
import matplotlib.pyplot as plt
clf1 = LogisticRegression(random_state=1)
clf2 = RandomForestClassifier(n_estimators=50, random_state=1)
clf3 = GaussianNB()
clf4 = KNeighborsClassifier(n_neighbors = 15)
clf5 = SVC(C= 1.7, kernel= 'linear', probability=True)
clf6 = DecisionTreeClassifier(criterion= 'gini', max_depth= 3, max_features= 2,
min_samples_leaf= 3)
clf7 = AdaBoostClassifier(learning_rate= 0.05, n_estimators= 150)
```

```
        clf8 = GradientBoostingClassifier(learning_rate= 0.01, n_estimators= 100)
        clf9 = ExtraTreesClassifier()
        clf10 = LGBMClassifier(boosting_type='gbdt', objective='binary', metric='binary_logloss', num_leaves=20,
                               learning_rate=0.1, max_depth=100,
                               )
        torch.set_num_threads(32)
        torch.cuda.set_device(0)
        device = torch.device("cuda" if torch.cuda.is_available() else "cpu")
        data = pd.read_csv("diabetes.csv")
        data2 = pd.read_csv("diabetes2.csv")
        data = data.append(data2)
        data.info()
        dataset_csv= data
        dataset_csv
        pd.value_counts(dataset_csv.Outcome)
        train_df, val_df = train_test_split(dataset_csv, test_size=0.2, random_state=1997)
        train_df.reset_index(inplace = True)
        val_df.reset_index(inplace = True)
        std=StandardScaler()
        columns = ['Pregnancies', 'Glucose', 'BloodPressure', 'SkinThickness', 'Insulin', 'BMI', 'DiabetesPedigreeFunction', 'Age']

        scaled = std.fit_transform(train_df[['Pregnancies', 'Glucose', 'BloodPressure', 'SkinThickness', 'Insulin', 'BMI', 'DiabetesPedigreeFunction', 'Age']])
        scaled = pd.DataFrame(scaled,columns=columns)

        train_df=train_df.drop(['Pregnancies', 'Glucose', 'BloodPressure', 'SkinThickness', 'Insulin', 'BMI', 'DiabetesPedigreeFunction', 'Age'], axis=1)
        train_df=train_df.merge(scaled, left_index=True, right_index=True, how = "left").set_index('index')


        scaled = std.fit_transform(val_df[['Pregnancies', 'Glucose', 'BloodPressure', 'SkinThickness', 'Insulin', 'BMI', 'DiabetesPedigreeFunction', 'Age']])
        scaled = pd.DataFrame(scaled,columns=columns)

        val_df=val_df.drop(['Pregnancies', 'Glucose', 'BloodPressure', 'SkinThickness', 'Insulin', 'BMI', 'DiabetesPedigreeFunction', 'Age'], axis=1)
        val_df=val_df.merge(scaled, left_index=True, right_index=True, how = "left").set_index('index')
```

```
        # train_df.reset_index(inplace = True)
        # val_df.reset_index(inplace = True)
        # train_df = train_df.reindex(['gender','hypertension','heart_disease','ever_married','work_type','Residence_type','Residence_type','smoking_status','avg_glucose_level','bmi','age','stroke'], axis=1)
        # val_df = val_df.reindex(['gender','hypertension','heart_disease','ever_married','work_type','Residence_type','Residence_type','smoking_status','avg_glucose_level','bmi','age','stroke'], axis=1)
        train_df = train_df[['Pregnancies', 'Glucose', 'BloodPressure', 'SkinThickness', 'Insulin', 'BMI', 'DiabetesPedigreeFunction', 'Age', 'Outcome']]
        val_df = val_df[['Pregnancies', 'Glucose', 'BloodPressure', 'SkinThickness', 'Insulin', 'BMI', 'DiabetesPedigreeFunction', 'Age', 'Outcome']]
        params={}
        params['learning_rate']=0.1
        params['boosting_type']='gbdt'
        params['objective']='binary'
        params['metric']='binary_logloss'
        params['max_depth']=100
        params['num_leaves']=20

        train_d = lgb.Dataset(train_df.iloc[:,:8],label=train_df.iloc[:,8])
        lmodel = lgb.train(params, train_d)
        y_pred=lmodel.predict(val_df.iloc[:,:8])
        #rounding the values
        y_pred=y_pred.round(0)
        #converting from float to integer
        y_pred=y_pred.astype(int)
        from sklearn.metrics import classification_report
        print(classification_report(val_df.iloc[:,8].values,y_pred))
        ax= plt.subplot()
        sns.heatmap(confusion_matrix(val_df.iloc[:,8].values, y_pred), annot=True, ax = ax, fmt="d"); #annot=True to annotate cells
        ax.set_title('LightGBM');
        ax.set_xticklabels(['No Diabetes ', 'Yes Diabetes'])
        ax.set_yticklabels(['No Diabetes ', 'Yes Diabetes'])
        class MyDataset(Dataset):
            def __init__(self, dataframe):
                self.df = dataframe
```

```
 def __len__(self):
        return len(self.df)
    def __getitem__(self, idx):
        inputs = self.df.iloc[:, : 8].values[idx]
        labels = self.df.iloc[:, 8].values[idx]
        input_tensor = torch.tensor(inputs).float()
        labels_tensor = torch.tensor(labels).float()
        return input_tensor, labels_tensor
    class MyModel(nn.Module):
    def __init__(self, input_shape, h1, h2, h3):

        super(MyModel, self).__init__()
        self.layer1 =  nn.Sequential(
        nn.Linear(input_shape, h1),
        nn.ReLU(),
    #       nn.Dropout(p=0.1),
        nn.Linear(h1, h2),
        nn.ReLU(),
        nn.Linear(h2, h3),
        nn.ReLU(),
        nn.Linear(h3, 1),
        nn.Sigmoid()
        )

    def forward(self, inputs):
        return self.layer1(inputs)
    model = MyModel(train_df.shape[1]-1, 100, 50, 20)
    model
    optimizer = torch.optim.Adam(model.parameters(), lr=0.005)
    criterion = nn.BCELoss()
    trainset    = MyDataset(train_df)
    train_loader    =    DataLoader(trainset,    batch_size=32,    shuffle=True,
num_workers=0)
    valset     = MyDataset(val_df)
    val_loader          =    DataLoader(valset,    batch_size=32,    shuffle=False,
num_workers=0)
    def train_model(epoch):
        model.train()
        avg_loss = 0.
        optimizer.zero_grad()
        for idx, (inputs, labels) in enumerate(train_loader):
            inputs_train, labels_train = inputs, labels.unsqueeze(1)
```

```
            output_train = model(inputs_train)
            loss = criterion(output_train,labels_train)
            loss.backward()
            optimizer.step()
            optimizer.zero_grad()
            avg_loss += loss.item() / len(train_loader)

        return avg_loss

    def test_model():

        avg_val_loss = 0.
        model.eval()
        with torch.no_grad():
            for idx, (inputs, labels) in enumerate(val_loader):
                inputs_vaild, labels_vaild = inputs, labels.unsqueeze(1)
                output_test = model(inputs_vaild)
                avg_val_loss += criterion(output_test, labels_vaild).item() /
len(val_loader)
        return avg_val_loss
    epochs = 1000
    best_val_loss = 99999
    for epoch in range(epochs):
        avg_loss = train_model(epoch)
        avg_val_loss = test_model()
        print('Epoch {}/{} \t loss={:.4f} \t val_loss={:.4f} \t'.format(
            epoch + 1, epochs, avg_loss, avg_val_loss))
        with open("output_final.txt", "a") as f:
            f.write('Epoch {}/{} \t loss={:.4f} \t val_loss={:.4f} \t  \n'.format(
                epoch + 1, epochs, avg_loss, avg_val_loss))
        f.close()
        if avg_val_loss < best_val_loss:
            best_val_loss = avg_val_loss
            torch.save(model.state_dict(), 'amir_model3.pth')
    model.load_state_dict(torch.load('amir_model3.pth'))
    y_preds = []
    y_true = []
    for i in val_loader:
        pred = model(i[0].float()).reshape(len(i[1]),)
        y_preds.extend([0 if x<0.5 else 1 for x in list(pred) ])
        y_true.extend([int(x) for x in i[1].numpy()])
```

```
        print(classification_report(y_true, y_preds))
        ax= plt.subplot()
        sns.heatmap(confusion_matrix(y_true, y_preds), annot=True, ax = ax, fmt="d");
#annot=True to annotate cells
        ax.set_title('Neural Network');
        ax.set_xticklabels(['No Diabetes ', 'Yes Diabetes'])
        ax.set_yticklabels(['No Diabetes ', 'Yes Diabetes'])
        estimators = [('LR',clf1), ('RF',clf2), ('GB',clf3),
                      ('KNN',clf4), ('SVC',clf5), ('DTC',clf6),
                      ('Ada',clf7), ('GBC',clf8), ('ETC',clf9), ('LBGM', clf10) ]
        kfold = StratifiedKFold(n_splits=30)
        ensemble = StackingClassifier(estimators,  final_estimator=clf10)
        results = cross_val_score(ensemble, train_df.iloc[:,:8],train_df.iloc[:,8],
cv=kfold)
        print('Accuracy on train: ',results.mean())
        ensemble_model = ensemble.fit(train_df.iloc[:,:8],train_df.iloc[:,8])
        pred = ensemble_model.predict(val_df.iloc[:,:8])
        print('Accuracy on test:' , (val_df.iloc[:,8] == pred).mean())
        y_preds = []
        y_true = []
        for i in val_loader:
           pred = model(i[0].float()).reshape(len(i[1]),)
           y_preds.extend([0 if x<0.5 else 1 for x in list(pred) ])
           y_true.extend([int(x) for x in i[1].numpy()])

        z_preds = list(ensemble_model.predict(val_df.iloc[:,:8]))
        print(classification_report(val_df.iloc[:,8], z_preds))
        ax= plt.subplot()
        sns.heatmap(confusion_matrix(val_df.iloc[:,8], z_preds), annot=True, ax = ax,
fmt="d"); #annot=True to annotate cells
        ax.set_title('Ensemble');
        ax.set_xticklabels(['No Diabetes ', 'Yes Diabetes'])
        ax.set_yticklabels(['No Diabetes ', 'Yes Diabetes'])
```

Приложение Н

Формат	Зона	Поз.	Обозначение	Наименование	Кол.	Приме-чание

					Дипломный проект	Лист
Изм.	Лист	№ докум.	Подпись	Дата		43

МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РЕСПУБЛИКИ КАЗАХСТАН

СӘТБАЕВ УНИВЕРСИТЕТІ

Институт информационных и телекоммуникационных технологий

Еленов Амир Мирхатович

5В070400 – Вычислительная техника и программное обеспечение

ОТЗЫВ НАУЧНОГО РУКОВОДИТЕЛЯ

к дипломному проекту

Тема: «Применение методов глубокого обучения на медицинских данных»

Тема дипломной работы имеет первостепенное значение для области медицины, а именно диагностике и определения диабета. Разработанная структура помогает, используя методы глубокого обучения определять диабет с большой вероятностью по общим данным пациента. Учащийся проявил обширную заинтересованность, а также компетентность при анализе текущих методов, использующихся в сфере, так и классических методов машинного обучения. Данная дипломная работа сравнивает различные подходы используя соответствующие метрики для их оценки, а также анализирует смешанные подходы. При написании данной работы были разобраны и объяснены ключевые подходы к задачам классификации, что дает гарантию того, что учащийся понимает их устройство, что доказывает применением их на практике. Учащийся в данной работе выполнил все поставленные задачи и провел анализ результатов, из которых были сделаны соответствующие выводы и показал свои знания в области машинного обучения и нейронных сетей.

Дипломный проект выполнен с учетом всех требований, предъявляемых к дипломному проекту по специальности 5В070400 – «Вычислительная техника и программное обеспечение», студент Еленов Амир Мирхатович рекомендован к защите дипломного проекта и заслуживает присвоения академической степени «бакалавра» по специальности 5В070400 – «Вычислительная техника и программное обеспечение».

Кулинкев А.Д.

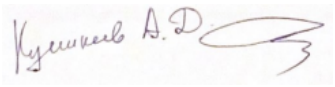
StrikePlagiarism.com

SATBAYEV UNIVERSITY

Дата отчета 6/7/2021
Дата редактирования ---

Отчет не был оценен.

Метаданные



Название
БАК 2021 Еленов Амир Мирхатович - нормконтроль.docx

Автор
Еленов Амир

Научный руководитель
Айдын Куникеев

Подразделение
ИКиИТ

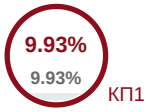
Список возможных попыток манипуляций с текстом

В этом разделе вы найдете информацию, касающуюся манипуляций в тексте, с целью изменить результаты проверки. Для того, кто оценивает работу на бумажном носителе или в электронном формате, манипуляции могут быть невидимы (может быть также целенаправленное вписывание ошибок). Следует оценить, являются ли изменения преднамеренными или нет.

Замена букв	ß	0
Интервалы	A→	0
Микропробелы	·	1
Белые знаки	ß	0
Парафразы (SmartMarks)	a	43

Объем найденных подобий

Обратите внимание!Высокие значения коэффициентов не означают плагиат. Отчет должен быть проанализирован экспертом.



КП1

КП2

КЦ

25
Длина фразы для коэффициента подобия 2

8342
Количество слов

63355
Количество символов

Подобия по списку источников

Просмотрите список и проанализируйте, в особенности, те фрагменты, которые превышают КП №2 (выделенные жирным шрифтом). Используйте ссылку «Обозначить фрагмент» и обратите внимание на то, являются ли выделенные фрагменты повторяющимися короткими фразами, разбросанными в документе (совпадающие сходства), многочисленными короткими фразами расположенные рядом друг с другом (парафразирование) или обширными фрагментами без указания источника ("криптоцитаты").

10 самых длинных фраз — Цвет текста

ПОРЯДКОВЫЙ НОМЕР	НАЗВАНИЕ И АДРЕС ИСТОЧНИКА URL (НАЗВАНИЕ БАЗЫ)	КОЛИЧЕСТВО ИДЕНТИЧНЫХ СЛОВ (ФРАГМЕНТОВ)	
1	https://galen-9.com/diabet/	336	4.03 %
2	https://coderlessons.com/tutorials/python-technologies/uznaite-mashinnoe-obuchenie-s-python/algoritm-knn-poisk-blizhaishikh-sosedei	38	0.46 %
3	https://coderlessons.com/tutorials/python-technologies/uznaite-mashinnoe-obuchenie-s-python/algoritm-knn-poisk-blizhaishikh-sosedei	29	0.35 %
4	https://pythonist.ru/chto-takoe-glubokoe-obuchenie-i-kak-ono-rabotaet/	29	0.35 %
5	https://pythonist.ru/chto-takoe-glubokoe-obuchenie-i-kak-ono-rabotaet/	27	0.32 %

6	**Metody uczenia zespołowego oparte na drzewach decyzyjnych** Szczęsna, Marta **11/26/2019** Szkoła Główna Handlowa (Szkoła Główna Handlowa)	23	0.28 %
7	https://nitin9809.medium.com/lightgbm-binary-classification-multi-class-classification-regression-using-python-4f22032b36a2	20	0.24 %
8	**Разработка конструкции безбалансирного гидроприводного станка качалки с подачей 35 м3/сут** Казиев Рустем Ерланович **5/10/2018** Satbayev University (Г_М_И)	17	0.20 %
9	https://pytorch.org/tutorials/intermediate/pipeline_tutorial.html	16	0.19 %
10	**Credit scoring jako narzędzie skutecznego systemu zarządzania ryzykiem kredytowym banku** Cichór, Joanna **11/26/2019** Szkoła Główna Handlowa (Szkoła Główna Handlowa)	14	0.17 %

из базы данных RefBooks (0.12 %)

ПОРЯДКОВЫЙ НОМЕР	НАЗВАНИЕ	КОЛИЧЕСТВО ИДЕНТИЧНЫХ СЛОВ (ФРАГМЕНТОВ)	
Источник: Paperity			
1	**Penerapan Metode Naive Bayes Dalam Pemilihan Kualitas Jenis Rumput Taman CV. Rumput Kita Landscape** Sri Rahayu, Anita Sindar RMS;	10 (1)	0.12 %

из домашней базы данных (1.35 %)

ПОРЯДКОВЫЙ НОМЕР	НАЗВАНИЕ	КОЛИЧЕСТВО ИДЕНТИЧНЫХ СЛОВ (ФРАГМЕНТОВ)	
1	**Исследование и разработка информационной системы диагностики сахарного диабета на базе инструментов BigData технологий** Мукашева А.К. **12/3/2019** Satbayev University (ИКиИТ)	38 (4)	0.46 %
2	**Разработка конструкции безбалансирного гидроприводного станка качалки с подачей 35 м3/сут** Казиев Рустем Ерланович **5/10/2018** Satbayev University (Г_М_И)	35 (4)	0.42 %
3	**Разработка установки вибропульсирующего слоя и исследование обжига сульфидов железа FeS и меди Cu2S** Сләмбеков Әсет Сләмбекұлы, Караев Адиль Маратович **5/13/2019** Satbayev University (Г_М_И)	17 (2)	0.20 %
4	**Хибибуллаев Ш. Дипломная работа (для антиплагиата).docx** Шухрат Хибибуллаев **5/3/2019** Satbayev University (ИКиИТ)	14 (2)	0.17 %
5	**Улучшение условий труда на основе аттестации рабочих мест Karachaganak Petroleum Operating b.v** Бекмухамбетова Асель Каиргалиевна **5/6/2019** Satbayev University (ИХиБТ)	9 (1)	0.11 %

из программы обмена базами данных (1.37 %)

ПОРЯДКОВЫЙ НОМЕР	НАЗВАНИЕ	КОЛИЧЕСТВО ИДЕНТИЧНЫХ СЛОВ (ФРАГМЕНТОВ)	

1	**Metody uczenia zespołowego oparte na drzewach decyzyjnych** Szczęsna, Marta **11/26/2019** Szkoła Główna Handlowa (Szkoła Główna Handlowa)	47 (4)	0.56 %
2	**Credit scoring jako narzędzie skutecznego systemu zarządzania ryzykiem kredytowym banku** Cichór, Joanna **11/26/2019** Szkoła Główna Handlowa (Szkoła Główna Handlowa)	46 (5)	0.55 %
3	**Применение технологии CLIL при преподавании цикла естественных дисциплин (на примере курса ИКТ)** Абильдинова Гульмира Маратовна **6/1/2021** S.Toraigyrov Pavlodar State University (Кафедра "Иностранная филология")	13 (1)	0.16 %
4	**SUMDU/out2019/Shalda_mag_rob.pdf** SUMDU **7/22/2019** Sumy State University (SUMDU)	8 (1)	0.10 %

из интернета (7.08 %)

ПОРЯДКОВЫЙ НОМЕР	ИСТОЧНИК URL	КОЛИЧЕСТВО ИДЕНТИЧНЫХ СЛОВ (ФРАГМЕНТОВ)	
1	https://galen-9.com/diabet/	336 (1)	4.03 %
2	https://pythonist.ru/chto-takoe-glubokoe-obuchenie-i-kak-ono-rabotaet/	108 (7)	1.29 %
3	https://coderlessons.com/tutorials/python-technologies/uznaite-mashinnoe-obuchenie-s-python/algoritm-knn-poisk-blizhaishikh-sosedei	74 (3)	0.89 %
4	https://pytorch.org/tutorials/intermediate/pipeline_tutorial.html	22 (2)	0.26 %
5	https://nitin9809.medium.com/lightgbm-binary-classification-multi-class-classification-regression-using-python-4f22032b36a2	20 (1)	0.24 %
6	https://www.diabetesatlas.org/upload/resources/material/20191217_165723_2019_IDF_Advocacy_Guide_RU.pdf	14 (2)	0.17 %
7	https://madewithml.com/courses/mlops/experiment-tracking/	12 (1)	0.14 %
8	https://massivefile.com/SVC_classification/	5 (1)	0.06 %

Список принятых фрагментов (нет принятых фрагментов)

ПОРЯДКОВЫЙ НОМЕР	СОДЕРЖАНИЕ	КОЛИЧЕСТВО ИДЕНТИЧНЫХ СЛОВ (ФРАГМЕНТОВ)

Протокол анализа Отчета подобия Научным руководителем

Заявляю, что я ознакомился(-ась) с Полным отчетом подобия, который был сгенерирован Системой выявления и предотвращения плагиата в отношении работы:

Автор: Еленов Амир

Название: БАК 2021 Еленов Амир Мирхатович - нормконтроль.docx

Координатор:Айдын Куникеев

Коэффициент подобия 1:9.9

Коэффициент подобия 2:5.5

Замена букв:0

Интервалы:0

Микропробелы:1

Белые знаки: 0

После анализа Отчета подобия констатирую следующее:

☑ обнаруженные в работе заимствования являются добросовестными и не обладают признаками плагиата. В связи с чем, признаю работу самостоятельной и допускаю ее к защите;

☐ обнаруженные в работе заимствования не обладают признаками плагиата, но их чрезмерное количество вызывает сомнения в отношении ценности работы по существу и отсутствием самостоятельности ее автора. В связи с чем, работа должна быть вновь отредактирована с целью ограничения заимствований;

☐ обнаруженные в работе заимствования являются недобросовестными и обладают признаками плагиата, или в ней содержатся преднамеренные искажения текста, указывающие на попытки сокрытия недобросовестных заимствований. В связи с чем, не допускаю работу к защите.

Обоснование:

……

07.06.2021 ……

Дата

Куникеев А.Д ……

Подпись Научного руководителя