МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ «НИЖЕГОРОДСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИМ. Н.И. ЛОБАЧЕВСКОГО» НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ

ИНСТИТУТ ИНФОРМАЦИОННЫХ ТЕХНОЛОГИЙ, МАТЕМАТИКИ И МЕХАНИКИ

ПРОГРАММА ПРОФЕССИОНАЛЬНОЙ ПЕРЕПОДГОТОВКИ «АНАЛИЗ ДАННЫХ ДЛЯ ПРИКЛАДНЫХ ОБЛАСТЕЙ» (252 ак.ч.)

ИТОГОВАЯ АТТЕСТАЦИОННАЯ РАБОТА НА ТЕМУ «ПРИМЕНЕНИЕ МЕТОДОВ МАШИННОГО ОБУЧЕНИЯ В ЗАДАЧЕ ДИАГНОСТИКИ АУТИЗМА У ВЗРОСЛЫХ НА ОСНОВЕ РЕЗУЛЬТАТОВ АНКЕТИРОВАНИЯ»

ПРЕДСЕДАТЕЛЬ АТТЕСТАЦИОННОЙ КОМИССИИ

_____ _____
(подпись) *(фамилия,инициалы)*

АВТОР
Панфилов Степан Николаевич (1 поток)
Дмитриев Даниил Валерьевич (1 поток)

_____ С.Н. Панфилов
(подпись) *(фамилия, инициалы)*

_____ Д.В. Дмитриев

РУКОВОДИТЕЛЬ

_____ Н.Э. Бабкин
(подпись) *(фамилия, инициалы)*

Нижний Новгород

2024

Аннотация

Данная работа направлена на разработку программного обеспечения, способного определить вероятность наличия расстройств аутистического спектра у исследуемой группы людей на основе результатов прохождения ими опросника «Коэффициент аутистического спектра-10» (Autism Spectrum Quotient, (AQ-10), созданного Национальным институтом исследований в области здравоохранения (National Institute for Health Research, NIHR) в 2012 году.

Для построения модели машинного обучения мы применили различные алгоритмы, включая Random Forest Classifier (метод случайных деревьев), Decision Tree Classifier (деревья решений) и Наивный Байесовский классификатор. С помощью этих алгоритмов были созданы модели, отличающиеся по точности и эффективности.

В итоге было разработано, оптимизировано и оценено программное обеспечение, позволяющее определить вероятность наличия расстройства аутистического спектра у опрашиваемой группы людей на основе результатов анкетирования.

Содержание

Введение

Расстройства аутистического спектра (РАС), также известные как аутизм, представляют собой группу нарушений поведения, характеризующихся качественными отклонениями в социальной коммуникации, социальных взаимодействиях и социологическом воображении. Люди с РАС часто демонстрируют ограниченность интересов, склонность к стереотипному повторяющемуся поведению, а также могут иметь сенсорные гипо- или гиперчувствительности [4].

Несмотря на то, что РАС имеет биологическую основу, его диагностика базируется на оценке поведенческих проявлений, так как причины развития аутизма разнообразны. Определение точных диагностических критериев осложняется вариативностью проявлений и интенсивности нарушений, связанных с РАС, у разных людей [5].

По данным исследований, в США аутизм диагностируется у одного из 59 детей [2], однако эта цифра может быть значительно заниженной из-за сложностей в диагностике, обусловленных разнообразием симптоматики. В связи с этим некоторые исследователи прибегают к использованию методов машинного обучения для оценки вероятности наличия РАС у испытуемых вместо применения традиционных статистических методов анализа данных.

Цель исследования:

разработка программного обеспечения, способного выявить возможность наличия расстройств аутистического спектра у исследуемой группы людей.

Задачи:

• Изучить методики диагностики РАС.

• Разработать программное обеспечение для выявления возможности наличия РАС у исследуемой группы людей.

Основная часть

1.Диагностика аутизма

Расстройства аутистического спектра (РАС) могут быть вызваны различными причинами, такими как инфекции плода (например, краснуха), отсутствие лечения метаболических нарушений (фенилкетонурия), приём матерью противосудорожных препаратов во время беременности, наличие генетических заболеваний (туберозный склероз), а также постнатальные инфекции (энцефалит). Однако конкретную причину развития аутизма удаётся установить лишь у 6-10% больных [4].

Для диагностики аутизма применяются две основные методики: ADI-R (Autism Diagnostic Interview, диагностический опрос для выявления аутизма) и ADOS-2 (Autism Diagnostic Observation Schedule, график наблюдения для диагностики аутизма). ADI-R представляет собой интервью с родителями или опекунами детей старше 2-х лет и взрослых с предполагаемым аутизмом, разработанное для систематического и стандартизированного наблюдения за аспектами поведения, имеющими диагностическое значение. Опрос состоит из 8 разделов и занимает до двух часов. ADOS-2 - это стандартизированная методика, позволяющая при наличии подозрения на РАС оценивать особенности общения, социального взаимодействия и игры. Она состоит из пяти модулей, каждый испытуемый проходит только один модуль, соответствующий его возрасту и уровню развития речи. Возможный возраст испытуемых варьируется от детей до 30 месяцев до взрослых. Прохождение модуля занимает до одного часа [10, 13].

Поскольку вышеописанные методики проводятся специалистами и требуют значительных временных затрат, были разработаны быстрые тесты для определения предрасположенности к наличию РАС [14]. Один из наиболее популярных - «Коэффициент аутистического спектра-50 (AQ-50)», опубликованный в 2001 году Саймоном Барон-Коэном и его коллегами из «Центра исследования аутизма» в Кембридже, Великобритания. Этот опросник, состоящий из пятидесяти вопросов, направлен на изучение наличия симптомов заболеваний аутистического спектра у взрослых людей со средним уровнем интеллекта [15].

Существующие опросники также легли в основу тестов в мобильных приложениях, позволяющих родителям без давления клиники заметить ранние признаки РАС. Несмотря на некоторую долю «ложноположительных» или «ложноотрицательных» результатов, использование таких приложений способствует улучшению осведомлённости родителей о проявлениях нейроотличности в раннем возрасте и поведенческих отклонениях в дальнейшем [5].

В нашей работе за основу взят опросник, составленный в 2012 году Национальным институтом исследований в области здравоохранения (National Institute for Health Research, NIHR), по данным, полученным в результате тестирования взрослых с подозрением на наличие РАС, не имеющих трудностей в обучении. Опросник состоит из 10 утверждений ("AQ-10"), отобранных из AQ-50, наиболее ярко проявляющихся у людей с РАС. Опрашиваемые выбирают степень своего согласия с каждым из утверждений. Опросник представлен в приложении 1.

Опрашиваемый получает 1 балл за каждый вопрос, если он выбрал «полностью согласен» или «частично согласен» в вопросах под номером 1, 7, 8 и 10, а также, если выбрал «полностью не согласен» или «частично не согласен» в вопросах под номером 2, 3, 4, 5, 6 и 9. Подсчитывается итоговое количество баллов, если оно равно 6 или больше, то считается, что опрашиваемый наиболее вероятно имеет РАС, и ему рекомендуется пройти специализированное диагностическое обследование для установления точного диагноза [1].

2.Использование машинного обучения в диагностике РАС

Машинное обучение - это область, посвященная пониманию и созданию методов с использованием статистики, направленных на улучшение производительности компьютера при выполнении определенного набора задач с использованием данных. Модель обучается на основе распределения данных, чтобы принимать решения на основе новых данных. Такой подход применяется для разработки сложных приложений, позволяющих делать точные классификации/прогнозы на различных данных [8, 11].

Машинное обучение подразделяется на модели с учителем (контролируемое обучение) и без учителя. Для моделей диагностики аутизма преимущественно используется первый тип [11].

Контролируемое машинное обучение включает алгоритмы, которые используют входные переменные для прогнозирования целевой классификации (зависимой переменной). В отличие от обучения без учителя (кластеризации), контролируемое обучение использует наборы данных, где целевое предсказание (например, диагноз) известно во время обучения для данных, используемых для обучения модели.

Модель контролируемого обучения считается успешной, если она может (а) точно предсказать целевой результат для обучающего набора данных с определенной степенью точности и (б) быть обобщена на новые наборы данных, помимо тех, которые использовались для обучения модели [2].

В нашей работе мы использовали методы "случайного леса", "дерева решений" и наивного байесовского классификатора.

2.1. Random Forest Classifier

Метод случайного леса (Random Forest) - это алгоритм машинного обучения, использующий ансамбль решающих деревьев. Его принцип заключается в использовании ансамбля деревьев, которые по отдельности дают не столь высокое качество классификации, однако анализируемые вместе они обеспечивают гораздо более классифицируемый паттерн.

Для создания деревьев часто используется алгоритм классификации под названием бэггинг (bagging, сокращение от bootstrap aggregation). Каждый алгоритм обучается и работает независимо от остальных на индивидуальной выборке, сформированной из исходного набора. После генерации достаточно большого количества деревьев, они голосуют за наиболее популярный класс. В итоге выбирается результат прогноза с наибольшим количеством голосов [6].

2.2. Decision Tree Classifier

Деревья решений специально разработаны для контролируемого поиска данных и представляют собой блок-схему, где каждый внутренний узел - это тест на атрибут, каждая ветвь обозначает результат теста атрибута, а каждый листовой узел обозначает метку класса. Движение по дереву продолжается, пока программа не окажется в последнем листовом узле терминального уровня, который возвращает предсказанную классификацию [9].

Decision Tree Classifier состоит из двух фаз:

1. Формирование дерева: все кортежи данных изначально находятся в корневом узле. Применяется критерий разделения и выбирается лучший атрибут разбиения для следующего уровня. Значение лучшего атрибута дает количество ветвей для этого узла. Разбиение продолжается, пока выборка в одном узле не станет очень маленькой и каждая часть будет состоять из выборки из одного класса. В итоге генерируется полное дерево, в котором никакое дальнейшее улучшение листового узла не увеличивает точность исследуемых данных.
2. "Обрезка" дерева: обрезка уменьшает размер дерева, удаляя поддеревья, которые отражают шум и выбросы. После генерации полного дерева применяется алгоритм для проверки повторяющихся и реплицированных поддеревьев. Если такое поддерево существует, то оно подвергается обрезке. Обрезка приводит к более быстрым и надежным классификаторам [12].

2.3. Наивный байесовский классификатор

Наивный байесовский классификатор является разновидностью вероятностного механизма классификации, основанного на теореме Байеса [16]. Теорема Байеса - это математическая формула, используемая для вычисления условных вероятностей, т.е. вероятности того, что результат произойдет, учитывая, что другое событие уже произошло. Байесовский классификатор включает серию вероятностных вычислений с целью нахождения наиболее подходящей классификации для заданной части данных в проблемной области. Существует два основных подхода в применении байесовского вывода к задаче классификации: вывод вероятности модели и вывод вероятности класса [7]. Этот классификатор может быть использован как общий инструментарий и применим к различным областям [16].

3. Применение методов машинного обучения

Исходными данными для анализа являлся датасет в формате CSV, полученный из открытых источников. Он содержал результаты прохождения опросника "AQ-10" и сопутствующие переменные, такие как возрастная категория, пол, национальность, наличие желтухи у испытуемого, диагноз аутизма, страна проживания и семейное положение. Всего были собраны данные 704 человек [3].

На следующем этапе производилось написание моделей машинного обучения и их оценка (Приложение 2).

Для начала были загружены библиотеки «Pandas» и «NumPY», а также сам датасет. Изучение имеющихся переменных показало, что число опрошенных составило 704 человека. Среди них 189 человек имели подтвержденный диагноз РАС, а 515 были здоровыми. Доля опрошенных с РАС от общего числа составила 26,85%.

После удаления ячеек с пропущенными значениями статистика изменилась: всего опрошенных - 702, число опрошенных с подтвержденным диагнозом РАС - 189, число здоровых опрошенных - 513, доля людей с РАС - 26.92%.

Для предварительной и статистической обработки данных были импортированы библиотеки «seaborn» и «matplotlib.pyplot». Из датасета были удалены нерелевантные переменные, такие как национальность («ethnicity»), страна проживания («country_of_res»), возрастная категория («age_desc») и сумма баллов опроса («result»).

Значения оставшихся переменных были переведены в числовой формат для удобства обработки. Затем была построена матрица корреляций с использованием цветовой схемы «cmap='RdBu_r'» и указанием диапазона цветовых кодов от -1 до 1 («vmin=-1, vmax=1»). Аннотации были добавлены вручную с помощью цикла, который проходил по каждой ячейке матрицы и добавлял текст с значением корреляции. По полученной матрице видно, что возраст, пол и наличие перенесенной в детстве желтухи слабо коррелируют с результатом (Приложение 3, рис. 2).

Датасет был разделен на две части по переменной «Class/ASD». Для стандартизации набора данных относительно возраста и результата теста (но не переменной «austim» для большей дифференциации данных) был использован класс «MinMaxScaler». Категориальные переменные были преобразованы в горячую кодировку с помощью функции «pd.get_dummies».

Для наглядности была создана гистограмма, показывающая частоту значений по переменной «Class/ASD» (Приложение 3, рис. 3).

С помощью функции «train_test_split» датасет был разделен на обучающую и тестовую части для машинного обучения. Обучающий набор содержал 561 строку, а тестовый - 141.

Для подбора модели был использован метод «дерева решений» (DecisionTreeClassifier), так как он позволяет моделировать сложные процессы и легко их интерпретировать. Параметры модели проверялись с помощью модуля «metrics» из библиотеки «sklearn». Точность составила 0.94, а доля ошибок - 0.06.

Следующим использованным методом был «случайный лес» (RandomForestClassifier), который обладает рядом преимуществ, таких как нечувствительность к выбросам, малые требования к предобработке данных и масштабированию, малая чувствительность к гиперпараметрам и меньший разброс модели. Эти параметры указывают на то, что полученная модель не склонна к переобучению.

Для оценки степени обучения модели использовался параметр «F-beta», рассчитанный с помощью функции «fbeta_score». Полученное значение 0.97 свидетельствует о хорошем качестве обучения модели.

С помощью техники «feature importance» были определены наиболее важные признаки для модели «случайного леса» в предсказании. Ими оказались:

• A9: «Мне легко понять, что думает или чувствует другой человек, просто взглянув на его лицо» (Приложение 1);

• A6: «Я знаю, как определить, что слушающему меня человеку становится скучно» (Приложение 1);

• A5: «Мне легко «читать между строк», когда со мной кто-то разговаривает» (Приложение 1);

• Возраст.

Была проверена возможность использования наивного байесовского классификатора. Несмотря на легкость интерпретации результатов и пригодность для больших выборок, не всегда выполнялось предположение о независимости характеристик, которые должны составлять полную группу событий. F-бета балл составил 0.88, что указывает на худшее качество обучения по сравнению с предыдущими моделями.

Для настройки параметров модели и оценки ее точности были использованы функции «fbeta_score», «accuracy_score», «make_scorer», «GridSearchCV», «train_test_split» и «SVC» из библиотеки «sklearn.metrics». Результаты показали, что оптимизированная модель была обучена лучше, чем неоптимизированная. Оценка точности по данным тестирования для неоптимизированной модели составила 0.9716, а F-score - 0.9635. Для оптимизированной модели итоговая оценка точности по данным тестирования составила 1.0000, а F-score - 1.0000.

В целом, применение методов машинного обучения позволило создать несколько моделей с высокой точностью и качеством предсказания наличия расстройства

аутистического спектра на основе результатов анкетирования. Наилучшие результаты показала оптимизированная модель с точностью 1.0000 и F-beta баллом 1.0000.

Заключение

Расстройства аутистического спектра (РАС), также известные как аутизм, представляют собой сложное явление, сочетающее в себе как биологические, так и поведенческие аспекты. Диагностика РАС является многоэтапным процессом, требующим наблюдения и оценки специалистов. В настоящее время проблема диагностики РАС стоит особенно остро, поэтому широкое внедрение и использование программного обеспечения для выявления РАС может помочь снизить нагрузку на медицинских работников, идентифицировать группы риска и ускорить прохождение первичных этапов постановки диагноза.

В соответствии с поставленной целью была разработана модель машинного обучения, позволяющая оценить вероятность наличия РАС у исследуемой группы людей на основе результатов заполнения опросника AQ-10 («Коэффициент аутистического спектра-10»). Программное обеспечение использует такие методы, как Random Forest Classifier (метод случайного леса) и Decision Tree Classifier (метод дерева решений). Была проведена оптимизация используемой модели и оценена ее эффективность в сравнении с неоптимизированной моделью.

Результаты показали, что оптимизированная модель достигла высокой точности (1.0000) и F-beta балла (1.0000) при предсказании наличия РАС на основе данных анкетирования. Это свидетельствует о том, что разработанное программное обеспечение может быть эффективным инструментом для предварительной диагностики РАС и выявления лиц, нуждающихся в дальнейшем обследовании специалистами.

Список литературы

1. Allison, C., Auyeung, B., Baron-Cohen, S. Toward brief “Red Flags” for autism screening: The Short Autism Spectrum Quotient and the Short Quantitative Checklist for Autism in toddlers in 1,000 cases and 3,000 controls [corrected] / C. Allison, B. Auyeung, S. Baron-Cohen // Journal of the American Academy of Child and Adolescent Psychiatry. – 2012. – № 51(2). – С. 202–212.
2. Applications of Supervised Machine Learning in Autism Spectrum Disorder Research: a Review / K. Hyde, M.N. Novack, N. LaHaye. [и др.] // Review Journal of Autism and Developmental Disorders. – 2019. – № 6. – С. 128–146.
3. Autism Screening on Adults: база данных. Данные в формате CSV. URL: https://www.kaggle.com/datasets/andrewmvd/autism-screening-on-adults (дата обращения 20.04.2023).
4. Baird, G. Diagnosis of autism / G. Baird, H. Cass, V. Slonims // BMJ. – 2003. – № 327(7413). – С. 488–493.
5. Barbaro, J. Study protocol for an evaluation of ASDetect - a Mobile application for the early detection of autism / J. Barbaro, M. Yaari // BMC Pediatrics. - 2020. - №21. - С. 1 – 20.
6. Breiman, L. Random Forests / L. Breiman // Machine Learning. - 2001. - №45. - С. 5–32.
7. Cichosz, P. Naïve Bayes classifier in Data Mining Algorithms: Explained Using R / P. Cichosz // Wiley. – 2015. – С.118-133.
8. George, G. FROM THE EDITORS: BIG DATA AND MANAGEMENT / G. George, M. R. Haas, A. Pentland // The Academy of Management Journal. - 2014. - № 57(2). - С. 321–326.
9. Goodfellow, I. Deep Learning / I. Goodfellow, Y. Bengio, A. Courville. - Cambridge: The MIT Press, 2016. - 800 с.
10. Multi-modular AI Approach to Streamline Autism Diagnosis in Young Children / H. Abbas, F. Garberson, S. Liu-Mayo [et al.] // Sci Rep. - 2020. - №10. - 5014 с.
11. Parikh, M.N.Enhancing Diagnosis of Autism With Optimized Machine Learning Models and Personal Characteristic Data / M.N. Parikh, H. Li, L. He // Frontiers in Computational Neuroscience. – 2019. – № 13(9). – С. 1–5.
12. Priyanka Decision tree classifier: a detailed survey / Priyanka, D. Kumar // International Journal of Information and Decision Sciences. – 2020. – №. 12 (3). – С. 246-269.
13. Systematic Review and Meta-Analysis of the Clinical Utility of the ADOS-2 and the ADI-R in Diagnosing Autism Spectrum Disorders in Children / J.B. Lebersfeld, M. Swanson, C.D. Clesi [и др.] // Journal of Autism and Developmental Disorders. – 2021. – № 51. – С. 4101–4114.

14. Thabtah, F. Autism Spectrum Disorder Screening: Machine Learning Adaptation and DSM-5 Fulfillment / F. Thabtah // Medical and Health Informatics. – 2017. – № 1. – С. 1–6.
15. Woodbury-Smith, M. R. Screening adults for Asperger Syndrome using the AQ: a preliminary study of its diagnostic validity in clinical practice / M. R. Woodbury-Smith, J. Robinson, S. Wheelwright, S. Baron-Cohen // Journal of autism and developmental disorders. – 2005. – № 35(3). – С. 331–335.
16. Yang, F. J. An implementation of naive bayes classifier / F. J. Yang // International conference on computational science and computational intelligence (CSCI). – 2018. – С. 301–306.

Приложение 1

	Полностью согласен	Частично согласен	Частично не согласен	Полностью не согласен
Я часто замечаю еле-уловимые звуки, которые другие не слышат				
Обычно я больше концентрируюсь на общей картине в целом, а не на мелких деталях				
Мне легко заниматься несколькими делами сразу				
Если меня отвлекли, я очень быстро могу вернуться к прежнему занятию				
Мне легко «читать между строк», когда со мной кто-то разговаривает				
Я знаю, как определить, что слушающему меня человеку становится скучно				
Когда я читаю повесть, мне сложно понять намерения персонажей				
Мне нравится собирать информацию о категориях вещей (к примеру, типах машин, видах птиц, типах поездов, видах растений и т. д.)				

Мне легко понять, что думает или чувствует другой человек, просто взглянув на его лицо				
Мне сложно понять намерения других людей				

Приложение 2

Пример кода написан *курсивом*, пример выходных данных выделен подчёркиванием

1. Сначала загружаем библиотеки *Pandas* и *NumPY*, они понадобятся нам для дальнейшей работы

```
import pandas as pd
import numpy as np
from time import time
from IPython.display import display
```

2. Загружаем сам датасет и смотрим имеющиеся в нём переменные. (Приложение 2, Таблица 1)

```
%matplotlib inline
data = pd.read_csv('autism_screening.csv')
data.head()
```

Значения переменных:

A1_Score Я часто замечаю еле уловимые звуки, которые не замечают другие люди.

A2_Score Обычно я больше концентрируюсь на картине в целом, чем на мелких деталях.

A3_Score Мне легко заниматься несколькими делами сразу.

A4_Score Если меня отвлекли, я очень быстро могу вернуться к прежнему занятию.

A5_Score Мне легко «читать между строк», когда со мной кто-то разговаривает.

A6_Score Я знаю, как определить, что слушающему меня человеку становится скучно.

A7_Score Когда я читаю повесть, мне сложно понять намерения персонажей.

A8_Score Мне нравится собирать информацию о категориях вещей (к примеру, типах машин, видах птиц, типах поездов, видах растений и т. д.).

A9_Score Мне легко понять, что думает или чувствует другой человек, просто взглянув на его лицо.

A10_Score Мне сложно понять намерения других людей.

age Возраст

gender Пол

ethnicity Национальность

jundice Болел ли испытуемый желтухой

austim Есть ли у испытуемого аутизм

contry_of_res Страна проживания

used_app_before Использовал ли испытуемый тест до этого

result Результат теста

age_desc Возрастная категория

relation Состоит ли человек в отношениях

3. Проверяем наличие пропущенных данных и выводим их количество.

```
Заменяем 'Not Available' на NaN и проверяем еще раз
print("Количество пропущенных данных по столбцам перед обработкой:")
print(data.isna().sum())
data = data.replace({'Not Available': np.nan})
print("\nКоличество пропущенных данных по столбцам после замены 'Not Available' на NaN:")
print(data.isna().sum())
```

4. Выводим базовую информацию о данных после очистки.

```
data.info()
```

5. Выводим статистику после удаления пропущенных значений.

```
print("\nСтатистика после удаления пропущенных значений:")
print_asd_statistics(data)
```

Статистика после удаления пропущенных значений:
Всего опрошенных: 702
Число опрошенных с подтвержденным диагнозом расстройства аутистического спектра: 189
Число здоровых опрошенных: 513
Доля людей в датасете с расстройством аутистического спектра: 26.92%

6. Удаление столбцов, которые не участвуют в анализе.

```
data = data.drop(columns=['ethnicity', 'contry_of_res', 'age_desc', 'result'])
```

7. Преобразование категориальных переменных в числовые.

```
mappings = {
'gender': {'f': 0, 'm': 1},
'jundice': {'no': 0, 'yes': 1},
'austim': {'no': 0, 'yes': 1},
'used_app_before': {'no': 0, 'yes': 1},
'Class/ASD': {'NO': 0, 'YES': 1},
'relation': {'Self': 0, 'Parent': 1, 'Health care professional': 2, '?': 3, 'Relative': 4, 'Others': 5}
}
```

```
for column, mapping in mappings.items():
if column in data.columns:
data[column] = data[column].map(mapping)
```

8. Удаление строк с пропущенными значениями и конвертация всех возможных текстовых полей в числа.

```
data.dropna(inplace=True)

for column in data.columns:
data[column] = pd.to_numeric(data[column], errors='coerce')

data.dropna(inplace=True)
data.reset_index(drop=True, inplace=True)
```

9. Построение корреляционной матрицы. (Приложение 2, Таблица 2)

```
corr_matrix = data.corr()

plt.figure(figsize=(14, 12))
sns.set(font_scale=1.1)

fig, ax = plt.subplots(figsize=(14, 12))
sns.heatmap(corr_matrix, ax=ax, cmap='RdBu_r', vmin=-1, vmax=1, square=True,
linewidths=0.5)

for i in range(len(corr_matrix.columns)):
for j in range(len(corr_matrix.columns)):
text = ax.text(j+0.5, i+0.5, f"{corr_matrix.iloc[i, j]:.2f}",
ha="center", va="center", color="black", size=12)

plt.show()
```

10. Разделим датасет на два: переменную Class/ASD и остальные переменные.

```
data_main = data['Class/ASD']
features_raw = data[['age', 'gender', 'jundice', 'austim',
'relation','A1_Score','A2_Score','A3_Score','A4_Score','A5_Score','A6_Score','A7_Score','A8_Score'
,'A9_Score','A10_Score']]
```

11. Стандартизируем набор данных относительно возраста.

```
scaler = MinMaxScaler()
age_columns = ['age']
features_mM_tr = features_raw.copy()
features_mM_tr[age_columns] = scaler.fit_transform(features_raw[age_columns])
display(features_mM_tr.head(3))
```

12. Используем функцию pd.get_dummies для преобразования категориальной переменной в горячую кодировку.

```
featuresfin = pd.get_dummies(features_mM_tr)
featuresfin.head(5)
```

13. Создадим гистограмму, показывающую частоту значений по переменной Class/ASD для наглядности. (Приложение 2, Таблица 3)

```
plt.hist(data_main, bins=2)
plt.xlim(0,1)
plt.title('Гистограмма Class/ASD')
plt.xlabel('Значения Class/ASD')
plt.ylabel('Частота')
plt.show()
```

14. Разделим датасет на обучающую и тестовую части для машинного обучения.

```
X_train, X_test, y_train, y_test = train_test_split(featuresfin, data_main, test_size=0.2,
random_state=1)

print(f"Обучающий набор содержит {X_train.shape[0]} строк.")
print(f"Тестовый набор содержит {X_test.shape[0]} строк.")
```

Обучающий набор содержит 561 строк.
Тестовый набор содержит 141 строк.

15. Используем метод дерева решений для подбора модели.

```
dectr_model = DecisionTreeClassifier()
dectr_model.fit(X_train, y_train)
```

16. Сравним степень соответствия в наборах.

```
y_pred = dectr_model.predict(X_test)
```

```
print('True :', y_test.values[0:25])
print('Pred :', y_pred[0:25])
```

True : [1 0 0 0 0 0 1 1 0 0 1 1 0 0 0 0 0 0 0 0 0 0 1 1 0]

Pred : [1 0 0 0 0 0 1 1 0 0 1 1 0 0 0 0 0 0 0 0 0 1 1 1 0]

17. Проверим параметры модели.

```
cm = metrics.confusion_matrix(y_test, y_pred)
print(cm)

TP = cm[1,1]
FP = cm[0,1]
TN = cm[0,0]
FN = cm[1,0]

print('\nMetrics:')
print(f'Accuracy(Точность): {(TP + TN) / float(TP + TN + FP + FN):.2f}')
print(f'Error Rate(Доля ошибок): {(FP + FN) / float(TP + TN + FP + FN):.2f}')
print(f'Precision: {metrics.precision_score(y_test, y_pred):.2f}')
print(f'Score: {dectr_model.score(X_test, y_test):.2f}')
```

18. Проверим параметры модели.

```
cm = metrics.confusion_matrix(y_test, y_pred)

print(cm)

TP = cm[1,1]

FP = cm[0,1]

TN = cm[0,0]

FN = cm[1,0]

print('\nMetrics:')

print(f'Accuracy(Точность): {(TP + TN) / float(TP + TN + FP + FN):.2f}')

print(f'Error Rate(Доля ошибок): {(FP + FN) / float(TP + TN + FP + FN):.2f}')

print(f'Precision: {metrics.precision_score(y_test, y_pred):.2f}')
```

print(f'Score: {dectr_model.score(X_test, y_test):.2f}')

[[97 4]

[7 33]]

Metrics:

Accuracy(Точность): 0.92

Error Rate(Доля ошибок): 0.08

Precision: 0.89

Score: 0.92

19. Попробуем использовать метод случайного леса.

rndm_model = RandomForestClassifier(n_estimators=5, random_state=1)

cv_score = cross_val_score(rndm_model, featuresfin, data_main, cv=10)

print(f'Средняя точность на кросс-валидации: {cv_score.mean():.2f}')

20. Находим F-beta для оценки качества модели.

rndm_model.fit(X_train, y_train)

y_pred_rf = rndm_model.predict(X_test)

print(f'F-beta score: {fbeta_score(y_test, y_pred_rf, beta=0.5):.2f}')

Модель обучена хорошо.

Средняя точность на кросс-валидации: 0.93

F-beta score: 0.90

21. Узнаем, какие признаки являются самыми важными для RF-модели в предсказании.

model_features = featuresfin.columns

data = data[model_features]

feats = {feature: importance for feature, importance in zip(model_features, rndm_model.feature_importances_)}

importances = pd.DataFrame.from_dict(feats, orient='index').rename(columns={0: 'Gini-Importance'})

importances.sort_values(by='Gini-Importance', ascending=False, inplace=True)

importances.reset_index(inplace=True)

importances.rename(columns={'index': 'Features'}, inplace=True)

22. Визуализация важности признаков. (Приложение 2, Таблица 4)

sns.set(style="whitegrid", font_scale=1.7)

plt.figure(figsize=(12, 8))

sns.barplot(x='Gini-Importance', y='Features', data=importances, palette='viridis_r')

plt.xlabel('Importance', fontsize=15, weight='bold')

plt.ylabel('Features', fontsize=15, weight='bold')

plt.title('Feature Importance', fontsize=15, weight='bold')

plt.show()

23. Выводим топ наиболее важных признаков для быстрого обзора. (Приложение 2, Таблица 5)

display(importances.head())

24. Проверим возможность использования Наивного байесовского классификатора.

nb_model = MultinomialNB()

cv_score = cross_val_score(nb_model, featuresfin, data_main, cv=10)

print(f'Средняя точность на кросс-валидации: {cv_score.mean():.2f}')

nb_model.fit(X_train, y_train)

y_pred = nb_model.predict(X_test)

print('F-beta:', fbeta_score(y_test, y_pred, average='binary', beta=0.5))

Данная модель обучена хуже, чем предыдущие.

Средняя точность на кросс-валидации: 0.87

F-beta: 0.7777777777777778

25. Проведем настройку параметров модели.

```
def f_beta_score(y_true, y_predict):
   return fbeta_score(y_true, y_predict, beta = 0.5)
clf = SVC(random_state = 1)
parameters = {'C':range(1,6),'kernel':['linear','poly','rbf','sigmoid'],'degree':range(1,6)}
scorer = make_scorer(f_beta_score)
grid_obj = GridSearchCV(estimator = clf, param_grid = parameters, scoring = scorer)
grid_fit = grid_obj.fit(X_train.values, y_train)
best_clf = grid_fit.best_estimator_
predictions = (clf.fit(X_train.values, y_train)).predict(X_test.values)
best_predictions = best_clf.predict(X_test.values)
print ("Не оптимизированная модель \n------")
print ("Оценка точности по данным тестирования: {:.4f}".format(accuracy_score(y_test, predictions)))
print ("F-score по данным тестирования: {:.4f}".format(fbeta_score(y_test, predictions, beta = 0.5)))
print ("\nОптимизированная модель\n------")
print ("Итоговая оценка точности по данным тестирования: {:.4f}".format(accuracy_score(y_test, best_predictions)))
print ("Итоговая F-score по данным тестирования: {:.4f}".format(fbeta_score(y_test, best_predictions, beta = 0.5)))
```

Не оптимизированная модель

Оценка точности по данным тестирования: 0.9716

F-score по данным тестирования: 0.9635

Оптимизированная модель

Итоговая оценка точности по данным тестирования: 1.0000

Итоговая F-score по данным тестирования: 1.0000

26. Таким образом, оптимизированная модель обучена лучше неоптимизированной.

Приложение 3

	A1_Score	A2_Score	A3_Score	A4_Score	A5_Score	A6_Score	A7_Score	A8_Score	A9_Score	A10_Score	...	gender	ethnicity	jundice	austim	contry_of_res	used_app_before	result	age_desc	relation	Class/
0	1	1	1	1	0	0	1	1	0	0	...	f	White-European	no	no	United States	no	6.0	18 and more	Self	
1	1	1	0	1	0	0	0	1	0	1	...	m	Latino	no	yes	Brazil	no	5.0	18 and more	Self	
2	1	1	0	1	1	0	1	1	1	1	...	m	Latino	yes	yes	Spain	no	8.0	18 and more	Parent	
3	1	1	0	1	0	0	1	1	0	1	...	f	White-European	no	yes	United States	no	6.0	18 and more	Self	
4	1	0	0	0	0	0	0	1	0	0	...	f	?	no	no	Egypt	no	2.0	18 and more	?	

Рис. 1. Импортированные значения переменных.

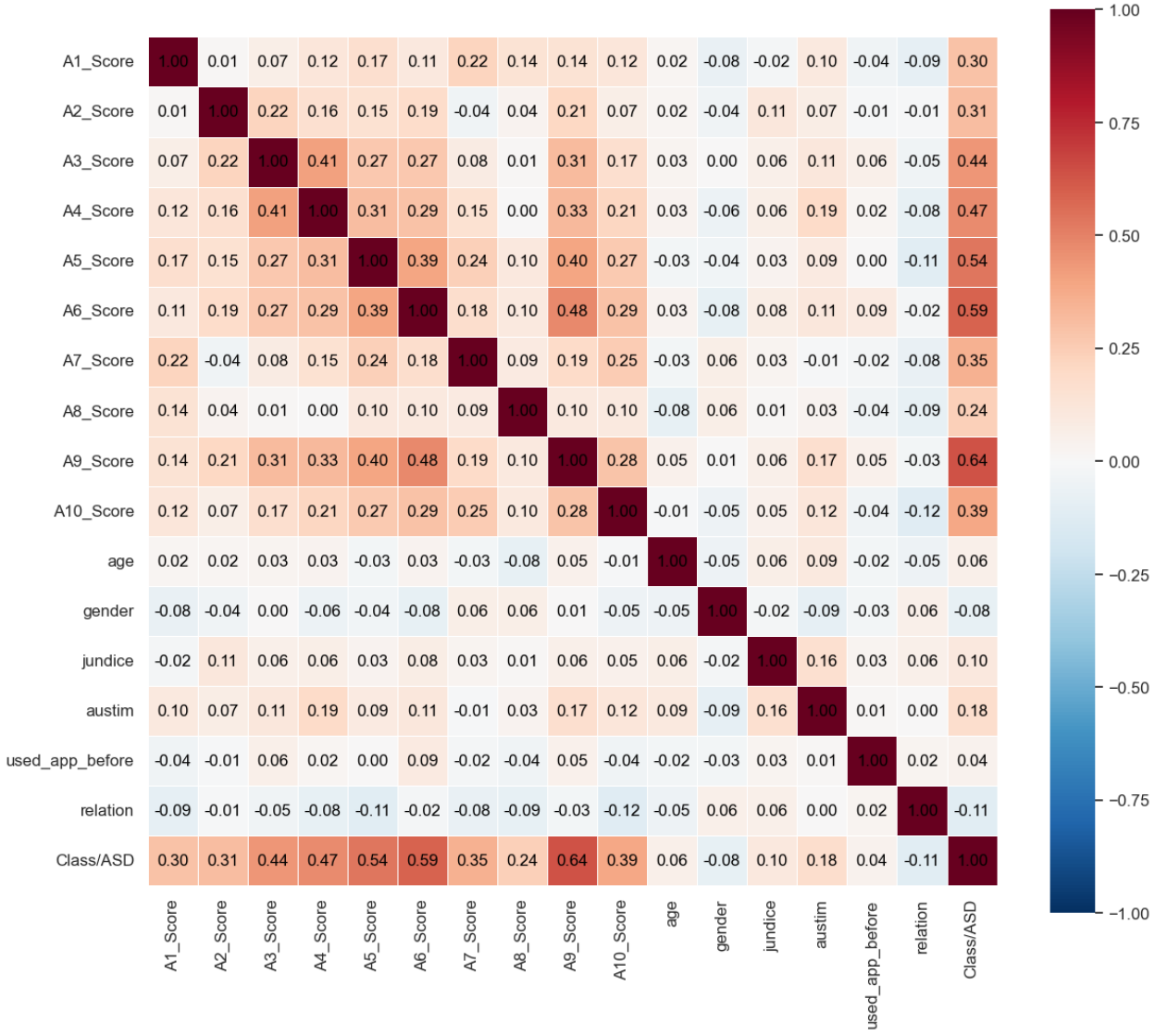

Рис. 2. Матрица корреляций

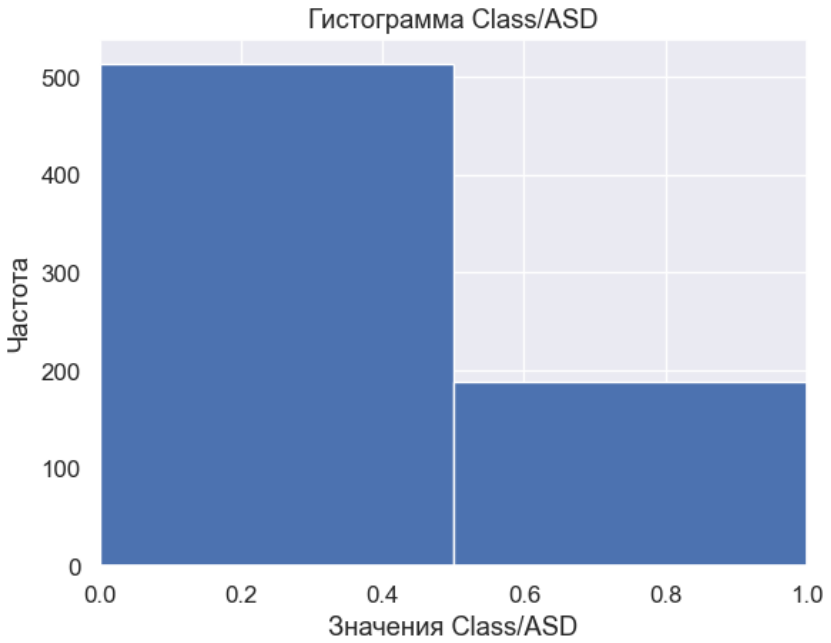

Рис. 3. Гистограмма Class/ASD

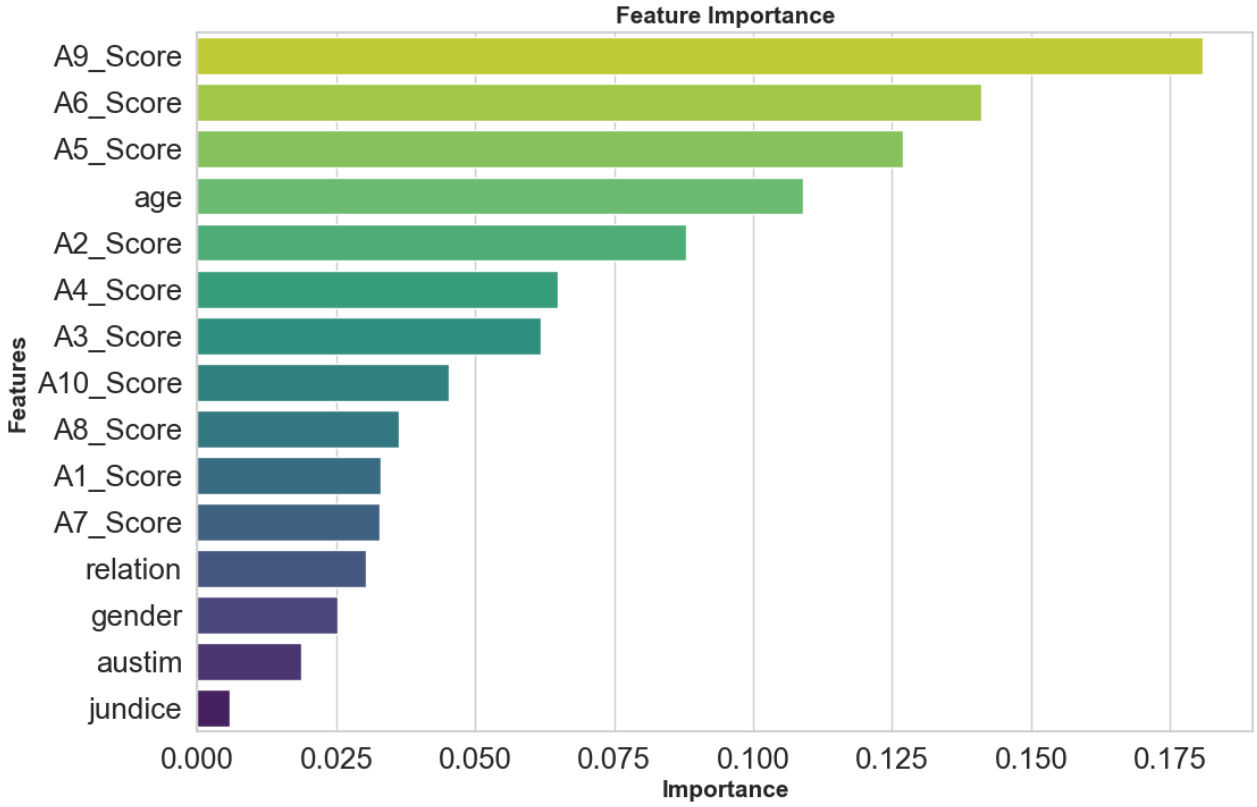

Рис. 4. Гистограмма отбор признаков по важности

	Features	Gini-Importance
0	A9_Score	0.180889
1	A6_Score	0.141027
2	A5_Score	0.126989
3	age	0.108947
4	A2_Score	0.088017

Рис. 5. Отбор признаков по важности (числовые значения).